Facultad 2

SOLUCIÓN DE INTEROPERABILIDAD DEL SISTEMA AUTOMATIZADO PARA LA SUPERACIÓN PEDAGÓGICA EN LA UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas

Autores: Pedro Rafael Cutiño Sánchez
Gianni Martínez Reina

Tutores: Dr.C. Febe Angel Ciudad Ricardo
Ing. Walfrido Serrano Pérez

La Habana, Junio 2014
"Año 56 de la Revolución"

Por este medio declaramos que: Pedro Rafael Cutiño Sánchez y Gianni Martínez Reina, ambos estudiantes de la facultad 2 de la Universidad de las Ciencias Informáticas, con números de identidad 90092841389 y 90101046785 respectivamente, son los únicos autores del resultado que se expone en la presente tesis titulada Solución de Interoperabilidad del Sistema Automatizado para la Superación Pedagógica (SASPED) en la Universidad de las Ciencias Informáticas (UCI).

Este trabajo fue desarrollado en el período comprendido entre el mes de septiembre del año 2013 y el mes de junio del año 2014, en colaboración con los colegas del equipo de desarrollo del Sistema Automatizado Para la Superación Pedagógica Yanet Crespo Díaz, Eric Rodríguez Ochoa, Harold García Alfonso, Yuniel Alamino Placeres, Leonardo Eloy Saavedra Remón y Yuniel Rodríguez Palmero.

A todos ellos, el agradecimiento más profundo. En especial agradecer al Dr.C. Febe Angel Ciudad Ricardo y al Ing. Walfrido Serrano Pérez, quienes fungieron como tutores y a los profesores Jommy Alexis Barban, Orestes Febles Díaz, Yoan Carlos Machado Espinosa, Pedro Manuel Nogales, Yordanys Piñeiro Gómez y Yurisbel Vega Ortiz.

Por todo lo anteriormente planteado, con todo el derecho y la responsabilidad que se nos otorga al ser los principales autores del presente trabajo, autorizamos al Centro de Innovación y Calidad de la Enseñanza (CICE) de la Universidad de las Ciencias informáticas a hacer uso del mismo en su beneficio.

Y para que así conste, se procede a firmar la presente declaración jurada de autoría en la provincia de La Habana, a los **30** días del mes de **mayo** del año **2014**.

_____

Autor: Pedro Rafael Cutiño Sánchez

_____

Autor: Gianni Martínez Reina

Dr.C. Febe Angel Ciudad Ricardo.

Correo electrónico: fciudad@uci.cu

Teléfono: 837-2579

Graduado como Ingeniero Informático en el año 2004 por la Universidad de Holguín "Oscar Lucero Moya" (UHOLM) y el Instituto Superior Politécnico "José Antonio Echeverría" (CUJAE). Titulado como Máster en Informática Aplicada en el año 2007 por la Universidad de las Ciencias Informáticas (UCI) y obtuvo el grado científico de Doctor en Ciencias de la Educación – Especialidad Tecnología Educativa en el año 2012 por la Universidad de La Habana (UH). Imparte su docencia de pregrado en las disciplinas de Ingeniería y Gestión de Software, Metodología de la Investigación Científica y Formación Pedagógica. Es miembro de los claustros de las maestrías de Informática Aplicada, Informática Avanzada, Gestión de Proyectos y Educación a Distancia de la UCI. Desarrolla sus investigaciones en las temáticas de Ingeniería y Gestión de Software, con énfasis en el área del Software Educativo; así como en la Tecnología e Informática Educativas. Ha publicado diversos artículos científicos y ha participado en diferentes eventos nacionales e internacionales en estas áreas del conocimiento. Ha sido arquitecto, analista y líder de proyectos de desarrollo de software, jefe de departamento docente y asesor técnico – docente. Actualmente se desempeña como Director del Centro de Innovación y Calidad de la Educación (CICE) de la UCI.

Ing. Walfrido Serrano Pérez.

Correo electrónico: wserrano@uci.cu

Teléfono: 837-8126

Graduado como Ingeniero en Ciencias Informáticas en la 3ra graduación de la UCI. Forma parte del colectivo del Centro de Innovación y Calidad de la Educación desde 2009 e imparte clases en la facultad 1 desde entonces, en las asignatura de Programación Web y el curso optativo Elementos de Hardware. Además trabajó vinculado al proyecto de Gestión de Archivos de la misma facultad, ha participado como expositor de los productos de la universidad en los eventos internacionales Universidad 2010 y Universidad 2012, así como en el salón de exposición de la UCI. Labora en la administración del Sistema de Encuestas UCI apoyando el proceso de caracterización de los estudiantes. Ha tutorado varias tesis en los años anteriores. Ha obtenido siempre la calificación de excelente en la evaluación profesoral.

El avance de las Tecnologías de la Información y las Comunicaciones ha tenido un impacto significativo en la sociedad. Los procesos de enseñanza y aprendizaje no están exentos a estos cambios. Con el apoyo de las TIC, el viejo modelo de superación pedagógica se actualiza acorde a las necesidades del sistema socialista cubano de crear personal altamente capacitado en la formación de profesionales competentes. A raíz de esto, en la Universidad de las Ciencias Informáticas se creó una Propuesta de Estrategia de Superación Pedagógica. Para la realización de esta nueva estrategia se propone un sistema informático que gestione los procesos asociados con la superación pedagógica. Dicho sistema está a su vez compuesto por varios subsistemas, cada uno de los cuales cumple una función fundamental para la resolución de la estrategia, pero que se encuentran aislados de los demás. Luego del estudio de la problemática, se llegó a la conclusión de que para optimizar el apoyo de los subsistemas a la Estrategia de Superación Pedagógica, se necesita que se garanticen las interoperaciones entre ellos. Durante el desarrollo de la investigación se revisaron los principales conceptos, estándares, tecnologías y herramientas asociadas a la interoperabilidad. Haciendo uso de la metodología OpenUP, se llevó a cabo un proceso ágil de desarrollo de software que tuvo como resultado la implementación de una solución de interoperabilidad que garantizara la interconexión, así como controlará la autenticación y la autorización de los subsistemas antes mencionados.

Information and Communication Technologies advancements have made a meaningful impact on society. The instruction and learning processes are not excluded from this changes. The old pedagogic improvement model is becoming a new, more effective one, in relation to the needs of the Cuban Socialist System to instruct highly qualified personnel, in competent professional capacitation techniques. To materialize this new strategy, a computing system which manages processes associated with pedagogic improvement, is required. This system is composed of a set of subsystems, each one managing a core function which is needed to the implementation of the strategy. A big issue in the situation is that each of those subsystems are isolated from the others. During the investigation, main concepts, standards, technologies and tools associated with interoperability were studied. Making use of the OpenUP methodology, an agile development process took place, which brought as a result an interoperability solution which guarantees interconnection between subsystems, and also controls authentication and authorization.

Índice

INTRODUCCIÓN 1

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA PROBLEMÁTICA 6

Interoperabilidad 6

Modelo de interoperabilidad propuesto 8

Beneficios de la interoperabilidad 11

Diseño del diagnóstico tecnológico a los subsistemas de SASPED 11

Soluciones de interoperabilidad en el ámbito internacional 13

Soluciones de interoperabilidad en el ámbito nacional 16

Principales métodos de interoperabilidad 18

Estándar 20

Protocolo SOAP 24

Metodología de desarrollo 24

Arquitectura de Software 25

Herramientas para el desarrollo de la solución 27

Conclusiones parciales 33

CAPÍTULO NO. 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN DE INTEROPERABILIDAD PARA LOS SUBSISTEMAS DE SASPED 34

Definición conceptual y operacional de las variables 34

Caracterización de la concepción de interoperabilidad de subsistemas de SASPED 35

Modelo conceptual de SASPED 36

Requisitos del sistema 38

Definición de casos de uso 40

Arquitectura de SASPED 44

Diagramas de clases del diseño 46

Patrones GRASP 48

Conclusiones parciales 50

CAPÍTULO NO. 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN DE INTEROPERABILIDAD DE LOS SUBSISTEMAS DE SASPED 51

Definición de los contratos de los Servicios Web implementados. 51
Definición de servicios web implementados....52
Gestión de seguridad para los servicios definidos 55
Gestión de control de acceso 56
Modelo de datos 57
Estilo de codificación 57
Tratamiento de errores 58
Pruebas 59
Validación de la solución de interoperabilidad para subsistemas de SASPED....66
Conclusiones parciales 67
CONCLUSIONES FINALES 68
REFERENCIAS BIBLIOGRÁFICAS 70
GLOSARIO DE TÉRMINOS 74
ANEXOS 75

Índice de tablas

Tabla 1. Operacionalización de la variable interoperabilidad entre subsistemas de SASPED 34
Tabla 2. Listado de requisitos funcionales 38
Tabla 3. Actores del sistema 40
Tabla 4. Descripción del caso de uso Autenticar usuario. 40
Tabla 5. Descripción del caso de uso Insertar Rol. 40
Tabla 6. Descripción del caso de uso Modificar Rol. 41
Tabla 7. Descripción del caso de uso Insertar Usuario. 41
Tabla 8. Descripción del caso de uso Eliminar Usuario. 41
Tabla 9. Descripción del caso de uso Verificar Privilegios de Usuario. 42
Tabla 10. Descripción del caso de uso Mostrar Detalles de Rol. 42
Tabla 11. Descripción del caso de uso Validar Petición de Servicio. 42
Tabla 12. Descripción del caso de uso Obtener Encuestas de Superación Pedagógica. 43
Tabla 13. Descripción del caso de uso Obtener Encuestados. 43
Tabla 14. Descripción del caso de uso Obtener Resultados de Encuestados. 43
Tabla 15. Descripción de las clases del servicio Encuestas_WS 47
Tabla 16. Descripción de las clases del servicio web Encuestados_WS 47
Tabla 17. Descripción de las clases del servicio web Respuestas_WS. 48
Tabla 18. Descripción de parámetros de entrada y salida del servicio Encuestas_WS. 53
Tabla 19. Descripción de parámetros de entrada y salida del servicio Encuestados_WS. 54
Tabla 20. Descripción de parámetros de entrada y salida del servicio Respuestas_WS. 55
Tabla 21. Resultado de las pruebas de rendimiento al servicio Encuestados_WS. 61
Tabla 22. Resultado de las pruebas de rendimiento al servicio Encuestas_WS. 62
Tabla 23. Clases de Equivalencia 65
Tabla 24. Casos de Prueba para Técnica de Caja Negra 65

Índice de figuras

Figura 1: Niveles de interoperabilidad (autor y año), [tomado de (Manso, 2008)]. 9
Figura 2: Modelo conceptual 38
Figura 3: Diagrama de casos de uso del sistema. 44
Figura 4: Arquitectura de SASPED. 46
Figura 5: Diagrama de clases del servicio web Encuestas_WS. 46
Figura 6: Diagrama de clases del servicio web Encuestados_WS. 47
Figura 7: Diagrama de clases del servicio web Respuestas_WS. 48
Figura 8: Ejemplo de encabezado WSDL del servicio Encuestados_WS con Username-Token 56
Figura 9: Modelo de Datos del Esquema Seguridad 57
Figura 10. Captura de errores al realizar la conexión a la base de datos 59
Figura 11. Captura de errores al llenar la lista de encuestados. 59
Figura 12: Cantidad de peticiones en función del tiempo del servicio Encuestados_WS. 63
Figura 13: Cantidad de peticiones en función del tiempo del servicio Encuestas_WS. 64

INTRODUCCIÓN

La enseñanza superior, según la Real Academia Española (RAE), es la que comprende los estudios especiales que cada profesión o carrera requiere. Partiendo de este concepto, su puede deducir que una de las principales necesidades de la sociedad es la de elevar la calidad de los procesos de enseñanza y aprendizaje. De acuerdo con lo planteado en la Conferencia mundial sobre la Educación Superior de la UNESCO (UNESCO, 1998), «*es necesario establecer estructuras, mecanismos y programas adecuados de formación del personal docente*».

En Cuba, las universidades no están ajenas a estos problemas, sino que persisten dificultades dentro del proceso de superación pedagógica. Según (Ciudad & otros, 2013), en un estudio realizado a diferentes diseños de superación pedagógica en el país, se constataron:

- Diferencias en el reconocimiento de necesidades en el área de conocimiento.
- Variedad de contextos en que se desarrolla la preparación pedagógica.
- Diversidad de objetivos y contenidos declarados en el área de conocimiento para organizar la superación profesional y la formación académica.

En la Universidad de las Ciencias Informáticas (UCI), a raíz de la evaluación institucional realizada por el Ministerio de Educación Superior (MES), disponible en el IFAI-UCI-MES (UCI, 2010) se comprobó que es necesario elevar la preparación pedagógica del claustro. Con tal objetivo, el Centro de Innovación y Calidad de la Educación (CICE) de esta universidad, elaboró la Estrategia de Superación Pedagógica del Claustro de la UCI (ESP) para elevar la calidad del proceso de formación integral de los profesores. La automatización de esta estrategia requiere un conjunto de datos que se obtienen de subsistemas y herramientas informáticas, algunos de éstos existentes y otros en construcción.

Estas herramientas informáticas son el Sistema Analizador Semántico de Texto, el Sistema Experto y el Sistema de Gestión de la Información, los cuales se encuentran actualmente en fase de desarrollo. Estos sistemas poseen la característica de operar por separado; es decir, son sistemas autosuficientes. Cada uno de estos brinda una funcionalidad necesaria para el funcionamiento de la concepción de la automatización de la ESP, a la cual se hará referencia en lo adelante como Sistema Automatizado para la Superación Pedagógica (SASPED).

El Analizador Semántico de Texto debe realizar el proceso de análisis de preguntas abiertas almacenadas en la base de datos de encuestas. Sin embargo, este proceso se ve imposibilitado, debido a la

incapacidad de interactuar que poseen ambos subsistemas. Por otra parte, el Sistema Experto se centra en la predicción de las rutas de superación pedagógica del claustro, necesitando para esto, datos que son generados por el Analizador Semántic*o* y a partir de encuestas realizadas a profesores, que son almacenadas en una base de datos externa a SASPED, viéndose afectado este proceso por la falta de colaboración entre estas aplicaciones. Por último, el sistema de Gestión de la Información es el encargado de administrar la evaluación profesoral, los controles a clases de los profesores y los Grupos de Trabajo de Formación Pedagógica, dependiendo de estrategia generada por el Sistema Experto, con el cual no tiene ninguna cooperación. Estos problemas provocan la falta de colaboración entre estas aplicaciones.

Al no existir una vía de comunicación oportuna, íntegra y segura de los datos, se recurre en muchos casos a la gestión de forma manual de la información, incluyendo la posibilidad de error humano; lo cual puede generar problemas como acceso de personal no autorizado a los datos, la dispersión de la información y valores diferentes para los mismos datos, afectando el éxito del funcionamiento de SASPED.

A partir de lo antes expuesto, se constata que a pesar de la relación que existe entre los procesos que realizan estos subsistemas, todos establecen esfuerzos individuales para dar solución a los diferentes aspectos de la problemática general que presenta la ESP; así como también se pudo confirmar que se encuentran aislados, pues no existe ninguna vía de comunicación e intercambio de datos, al no existir una estructura que los agrupe en un único sistema.

La falta de comunicación entre los sistemas de información está estrechamente relacionada con el desarrollo y la evolución de las tecnologías. Esto se evidencia en el cambio de paradigma de la utilización de aplicaciones sencillas para dar solución a problemas simples, en complejos sistemas distribuidos y no distribuidos, que se encargan de dar respuesta a problemáticas de mayor envergadura. La interoperabilidad de la información y las funcionalidades, se ha convertido en un aspecto clave en el terreno de las telecomunicaciones y la Informática, ya que a medida que crecen los volúmenes de datos en cualquier área, la heterogeneidad de la información se convierte en un problema global.

La concepción del sistema SASPED no está exenta a este tipo de problemas, sino que debido a la falta de cooperación anteriormente descrita entre los subsistemas que lo conformarán, se imposibilita la automatización del proceso de superación pedagógica en la UCI, en lo fundamental por no existir interoperabilidad a lo interno de la estructura de subsistemas.

A partir de lo expuesto, se puede concluir que el proceso de interacción entre los subsistemas de SASPED está caracterizado por:

- Insuficiencia en el intercambio de datos, así como en la comunicación entre los subsistemas.
- Insuficiencias tanto en la centralidad, consistencia, como en la integridad de la información.
- Alto acoplamiento entre los subsistemas.

Esto constata que el proceso de interacción entre los subsistemas de SASPED no facilita la comunicación y el intercambio de datos e información, impidiendo el objetivo fundamental de este.

En consecuencia, se identificó como **problema científico**: Las insuficiencias en la concepción de la interoperabilidad de los subsistemas de SASPED producen deficiente integración entre éstos para el cumplimiento de las funcionalidades deseadas.

Este problema se enmarca en el **objeto de estudio** concepción de la interoperabilidad de los subsistemas de SASPED; y el **campo de acción** lo comprenden las soluciones de interoperabilidad entre sistemas web.

Para resolver el problema identificado se propone el siguiente **objetivo general:**

Desarrollar una solución de interoperabilidad que permita la integración de los datos y las funcionalidades de los subsistemas de SASPED para la automatización de la Estrategia de Superación Pedagógica de la UCI.

Para dar cumplimiento al objetivo anteriormente planteado se definen las siguientes **tareas de investigación**:

1. Establecimiento de los fundamentos teórico-metodológicos para la comunicación de los subsistemas web en lo referente a la definición de interoperabilidad, sus principales modelos, tecnologías para lograrla, así como soluciones a otras problemáticas semejantes.
2. Diagnóstico e identificación de las insuficiencias de la concepción de interoperabilidad de los subsistemas informáticos de SASPED.
3. Desarrollo de la solución de interoperabilidad para el sistema SASPED.
4. Validación del grado de interoperabilidad ofrecido por la solución desarrollada para el sistema SASPED.

Como **hipótesis** en la investigación se plantea que: La implementación de una solución de interoperabilidad garantizará la integración de los datos y las funcionalidades de los subsistemas web que conforman SASPED.

Con el fin de dar cumplimiento a las tareas de investigación propuestas, fue necesario hacer uso de los siguientes métodos científicos:

Métodos Teóricos:

- Histórico-Lógico: Este método se utiliza con el objetivo de estudiar la evolución y trayectoria de las distintas soluciones de interoperabilidad existentes en el mundo y las más utilizadas en nuestro país, prestando especial atención a las herramientas que nos permitan desarrollar una solución de interoperabilidad para integrar los subsistemas de SASPED.
- Analítico-Sintético: Se emplea con el fin de revelar y comprender los factores que inciden directa e indirectamente sobre la interoperabilidad entre sistemas, así como para describir relaciones e interacciones que existen entre dichos factores.
- Hipotético-Deductivo: Este método se emplea para deducir y explicar las leyes y los patrones que rigen la interoperabilidad entre sistemas así como para la verificación del presupuesto hipotético enunciado en la investigación.

Métodos Empíricos:

- Observación: Se utiliza para conocer aspectos esenciales del comportamiento de la interoperabilidad y como método auxiliar para ampliar la información obtenida a través de los métodos Histórico Lógico, Analítico Sintético e Hipotético Deductivo.
- Experimento: Este método se utiliza para esclarecer las propiedades y/o cualidades de la integración de los subsistemas de SASPED, para que una vez revelada su realidad, se pueda actuar sobre ella con el fin de transformarla. Además, se utiliza para verificar el presupuesto hipotético enunciado en la investigación mediante la creación de las condiciones o la adaptación de las ya existentes.
- Análisis documental: Se emplea con el fin de adquirir todos los conocimientos que se puedan utilizar como base para el desarrollo de la creación científica a partir de la recolección, la selección y el análisis de documentos relacionados con el tema de estudio de la presente investigación.

La información del presente trabajo se encuentra dividida en tres capítulos. Quedando estructurado el mismo de la siguiente manera:

En el primer capítulo se realiza un estudio para determinar el modelo de interoperabilidad que presentará la solución. También se analiza la existencia de soluciones que operan bajo condiciones de uso similares, garantizando la interoperabilidad entre distintos sistemas. Se describen los beneficios de la interoperabilidad, como también los principales métodos para lograrla, los estándares que la especifican, los principales protocolos que la apoyan y las tecnologías y metodologías a utilizar en el desarrollo de la solución.

El segundo capítulo se refiere al análisis y diseño de solución de interoperabilidad para los subsistemas de SASPED. Este aborda la definición conceptual y operacional de la variable de la investigación, como también la caracterización del objeto de estudio, describiendo su estado actual. También se describen elementos relacionados con el análisis como los son el modelo conceptual de la problemática, los requisitos del sistema y la definición de casos de uso. Para el diseño se hace una propuesta de arquitectura del sistema, los candidatos a servicios web y los diagramas de clases.

El tercer capítulo describe la fase de implementación y validación de la solución. Se realiza una descripción de los contratos de los servicios web que formaran parte de la solución, pasando luego a la descripción de la implementación de dichos servicios, la gestión de la seguridad de dichos servicios, la descripción del intercambio de información entre los subsistemas de SASPED, la gestión del control de acceso al sistema, el modelo de datos del esquema de seguridad, la vista de despliegue, el estilo de codificación, el manejo de excepciones y las pruebas realizadas. Por último se procede a la validación de la solución con respecto a la modificación del estado del objeto de estudio.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA DE LA PROBLEMÁTICA

La capacidad de comunicación entre aplicaciones y sistemas web es de vital importancia para su completo éxito. No pocos han sido los estudios realizados en el terreno de las Ciencias Informáticas para vencer un enemigo común de dichos sistemas: el tiempo de respuesta y la seguridad de la información. El éxito de estas aplicaciones y sistemas web recae fundamentalmente sobre el esfuerzo que se realiza para aumentar su rapidez y fiabilidad a la hora de intercambiar y/o almacenar información.

Este capítulo agrupa un conjunto de conceptos fundamentales que muestran una visión general de la interoperabilidad entre sistemas, los cuales permitirán estudiar y realizar una propuesta de solución en la que los subsistemas informáticos de SASPED sean capaces de interactuar, logrando así una mayor eficiencia en todo el proceso de superación pedagógica. También se realiza un análisis en este capítulo sobre las herramientas, tecnologías y metodologías que se utilizan en el mundo en la actualidad que permitan la interoperabilidad entre sistemas heterogéneos.

Interoperabilidad

El término Interoperabilidad, tiene hoy un amplio abanico de conceptos y definiciones aplicadas desde los más variables puntos de vista de los diferentes autores que han hecho alusión al tema, sin embargo, todos y cada uno de ellos llegan a la misma conclusión literalmente.

De acuerdo con los autores estudiados, la interoperabilidad se define como:

- La capacidad de dos o más sistemas o componentes para intercambiar la información y utilizarla (IEEE, 1990).
- La capacidad de un sistema o componente de un sistema para proporcionar intercambio de información y el procesamiento cooperativo entre aplicaciones (ISO, 2011).
- La disponibilidad de mecanismos que permitan intercambiar procesos y/o datos entre sistemas heterogéneos (Nogales & Galvan, 2011).
- La capacidad de las organizaciones y sistemas diversos y dispares de trabajar juntos eficazmente en el cumplimiento de los objetivos comunes en beneficio mutuo mediante el intercambio de datos entre sistemas de las TIC (EIF, 2010).

Para (Chen, 2008), existen tres categorías de barreras que impiden la interoperabilidad: las barreras de naturaleza conceptual, que están relacionadas con las diferencias sintácticas y semánticas que sufre la información a intercambiar; las barreras tecnológicas, que son incompatibilidades con respecto a las

Tecnologías de la Información, ya sean arquitecturas, plataformas, infraestructura, etc., que comprenden los estándares para presentar, almacenar, intercambiar, procesar y comunicar datos e información a través del uso de sistemas de software; las barreras de tipo organizacional, que se encuentran relacionadas con la definición de responsabilidades y autoridades de forma que la interoperabilidad se efectúe bajo condiciones favorables, que pueden ser vistas como factores humanos y que están dirigidas a los comportamientos humanos y de la organización que pueden obstaculizar las interoperaciones.

Para la evaluación de sistemas informáticos interoperables, (Chen, 2008) propone que existen tres tipos de mediciones: potencialidad, compatibilidad y rendimiento. La potencialidad es la existencia en un sistema de características relacionadas con la interoperabilidad, que facilitan las interacciones con otro sistema. El propósito que persigue es aumentar la capacidad de interoperación, así como disminuir el riesgo de encontrar problemas durante una interacción. Por cada categoría de barrera, se definen 5 niveles para caracterizar la potencialidad:

- **Aislado**: El sistema no posee ninguna capacidad para interoperar.
- **Inicial**: La interoperabilidad requiere grandes esfuerzos que afectan la interacción.
- **Ejecutable**: La interoperabilidad es posible, pero el riesgo de encontrar problemas es grande.
- **Conectable**: La interoperación se realiza con facilidad incluso si aparecen problemas en la interacción.
- **Interoperable**: La interoperabilidad ha evolucionado al punto que la probabilidad de aparición de problemas es baja.

La compatibilidad es la medida con la cual se identifican las barreras en la interoperabilidad. A mayor compatibilidad se empequeñecen las barreras. Mediante la medición de compatibilidad se abordan interrogantes relativas a cada categoría de barreras de interoperabilidad. Para establecer un estado sobre la compatibilidad a nivel conceptual, se analiza si la información a ser transmitida se expresa con la misma sintaxis y/o significado. A nivel tecnológico, se observa si las tecnologías sobre las que están construidos los sistemas son compatibles y si se utilizan los mismos protocolos de intercambio de datos. En la dimensión organizacional, es necesario analizar entre dos sistemas se encuentran definidas claramente las autoridades y responsabilidades, además, si las estructuras de sus organizaciones son compatibles.

La medida de rendimiento de aporta el grado de interoperabilidad entre los sistemas, a través de la evaluación del tiempo, la calidad y el costo de la interoperación. También (Chen, 2008) propone una descomposición del tiempo de interoperación, la cual se efectuará a través de la suma de los tiempos de petición, tratamiento de la petición, retorno y uso de la información.

Para la calidad, se toman en consideración tres tipos: la calidad de intercambio, que refleja si el proceso de intercambio de datos se lleva a cabo sin problemas; la calidad de uso, que representa la cantidad de información recibida por un sistema en comparación con la que pidió, lo cual, al estar desbalanceado puede traer como consecuencias dificultad para procesar la información excedente o la falta de ésta; la calidad de conformidad corresponde al hecho de si la información recibida por un sistema puede ser utilizable o no. La medición de la calidad de la interoperación se dará a través de la suma de la calidad de intercambio, el valor absoluto de uso y la calidad de conformidad.

El costo de interoperación se define a través de los costos asociados a la eliminación de las barreras de interoperabilidad y la modificación de los sistemas para obtener un tiempo y calidad de interoperación aceptables. Se define como la suma del costo de intercambiar la información y el de hacer que los datos sean utilizables.

Modelo de interoperabilidad propuesto

El análisis bibliográfico realizado arrojó que la interoperabilidad puede estar representada en varios niveles. Según (Manso & otros, 2008), hasta la fecha se han identificado 14 niveles diferentes de interoperabilidad: Semántica, Técnica, Sintáctica, Pragmática, Organizacional, Conceptual, Dinámica, Legal, Social, Esquemática o Estructural, Intracomunitaria, Política-Humana, Internacional, Empírica y Física (**Figura 1**).

Autor	Año	Técnica	Esquemática/estructural	Semántica	Organizacional	Física	Empírica	Sintáctica	Pragmática	Social	Política / Humana	Legal	Internacional	Dinámica	Conceptual
ISO	1990	x		x				x							
Goh [23]	1997		x												
Goodchild [1]	1997	x	x		x			x	x			x		x	X
Bishr [24]	1998		x	x				x							
Shanzhen [25]	1999			x	x			x	x					x	
Ouksel & Sheth [26]	1999	x	x	x				x	x	x				x	
Miller [21]	2000	x		x							x	x	x		
Tolk [2]	2003				x										
Tolk, Muguira [27]	2003	x		x				x	x						x
Bermudez [28]	2004			x											
Shekhar [29]	2004		x	x				x							
Schekkerman [30]	2004	x	x	x											
Stroetmann [31]	2005			x				x							
Ding [32]	2005			x				x							
Nowak [33]	2005		x	x				x							
Mohammadi [34]	2006	x			x					x	x	x			
Kalantari [35]	2006	x		x								x			
Assche [36]	2006					x	x	x	x	X					
Turnitsa &Tolk [12]	2006	x		x				x	x					x	X
Dekkers [37]	2007	x		x	x										

Figura 1: Niveles de interoperabilidad (autor y año), [tomado de (Manso, 2008)].

Luego de examinar lo planteado por (Manso, y otros, 2008), se realiza una descripción de dichos niveles la cual se ofrece a continuación:

- Los niveles *internacionales, políticos/humanos, sociales y/o culturales, intercomunitarios, legales* y *de políticas*, pueden ser englobados como aspectos que afectan a usuarios, instituciones y organizaciones. Los autores citados agrupan estos aspectos en un único nivel llamado *organizacional.* Concretamente, en este nivel se abordan aspectos como las políticas de acceso, el uso de datos y servicios, responsabilidades personales o institucionales, los objetivos y los fines perseguidos por la organización al crear un dato o proporcionar un servicio.

- Los aspectos *físico* y *empírico* de la interoperabilidad atienden aspectos referentes al contexto de la enseñanza electrónica e inciden en la interacción hombre máquina y en la cantidad de información expuesta al estudiante.
- La interoperabilidad *técnica* posibilita la interconexión de los sistemas a nivel de protocolos y el intercambio de información en su nivel más básico: los bits. Es necesaria para interconectar sistemas computacionales y servicios, incluyendo aspectos clave como interfaces abiertas, servicios de conexión, integración de datos, presentación e intercambio de datos, accesibilidad y servicios de seguridad.
- La interoperabilidad *sintáctica* posibilita el intercambio de información, utilizando un formato común como lo representan los formatos estandarizados de intercambio de datos.
- La interoperabilidad semántica posibilita el intercambio de información, utilizando un vocabulario común y compartido que evite las inexactitudes en la interpretación del significado de los términos.
- La interoperabilidad *pragmática* es la que posibilita que los sistemas conozcan y exploten los métodos y procedimientos de los demás sistemas.
- La interoperabilidad *dinámica* permite a los sistemas autocorregir su funcionamiento ante los cambios en la transferencia de la información. Ofrece la capacidad de cambiar un servicio utilizado por otro, si este no cumple con las necesidades o si es inaccesible.
- La interoperabilidad *conceptual* es la que aborda la descipción de los modelos de datos y sistemas en forma de documentación estandarizada e intercambiable sin depender del modelo utilizado para describirla.
- La interoperabilidad esquemática o estructural hace referencia a las diferencias en los modelos de datos en cuanto al uso de lenguajes de modelado que independicen al modelo de su implementación.

Debido a que SASPED es un sistema compuesto por varios subsistemas heterogéneos que se encuentran aislados, y que presentan la necesidad de establecer interoperaciones en cuanto a la transmisión de información para hacer funcionar los procesos de la ESP, se analizará el aspecto **técnico** de la interoperabilidad. Para garantizar que los datos transmitidos tengan un significado común en cuanto a su formato y significado, se analizarán los aspectos **sintáctico** y **semántico** de la interoperabilidad.

La concepción de SASPED también tiene definidas personas en calidad de operadores como parte inherente al sistema, las cuales se rigen por políticas y reglas organizacionales, como también

regulaciones de la UCI, que según (Sommerville, 2005), son características de un sistema socio-técnico y por lo tanto, responden al aspecto **organizacional** de la interoperabilidad.

A partir de lo expuesto, se propone utilizar en la presente solución un modelo de interoperabilidad que abarcará los niveles **técnico**, **sintáctico**, **semántico** y **organizacional.**

Beneficios de la interoperabilidad

La capacidad de multiplicar los rendimientos que se obtienen con la automatización de procesos de las distintas esferas de la sociedad a través de sistemas informáticos, es una de las utilidades más destacadas de la interoperabilidad. Según (Criado & otros, 2010) esto se debe a la posibilidad que ésta brinda de cooperar entre distintos niveles como: Gobiernos, Agencias de Administración o simples Empresas. Todo esto sin distinción del nivel de desarrollo tecnológico, ya que combina los diferentes tipos de tecnologías de los organismos implicados. La interoperabilidad también simplifica la actividad administrativa y los procesos de negocios, lo que tiene como consecuencia un aumento en la eficiencia del quehacer de las instituciones públicas. Por otra parte, brinda la posibilidad de utilizar estándares abiertos, aplicaciones tecnológicas, reutilización de datos y funcionalidades que pueden influir de manera positiva en el costo de desarrollo de sistemas de información.

La interoperabilidad también permite la cooperación internacional en ámbitos en los que se requieran nuevas herramientas, a la vez que aumenta la sencillez con la que el ciudadano o usuario realiza sus trámites. Visto de otra manera, es posible aumentar la calidad de vida de las personas, ya que esta posee la capacidad de proveer servicios de salud adecuados, de mitigar el impacto de los desastres naturales, de vigilar la seguridad de nuestro país o facilitar el negocio exterior de nuestras empresas.

Lograr la integración de los subsistemas que conforman SASPED, garantizaría la rapidez y eficiencia en la comunicación de la información necesaria para la automatización de los procesos definidos en la ESP. A través del funcionamiento en conjunto de dichos procesos automatizados, y habiendo desplegado SASPED en la UCI, se contribuiría a la superación pedagógica continua de los profesionales que intervienen en la formación del Ingeniero en Ciencias Informáticas, siendo esto el problema a resolver de la estrategia propuesta por el CICE.

Diseño del diagnóstico tecnológico a los subsistemas de SASPED

Con la intención de determinar tanto las especificaciones técnicas como los procesos internos que llevan a cabo cada uno de los subsistemas que intervienen y aportan datos a SASPED, se realiza un grupo de

reuniones con los integrantes de cada uno de los equipos de desarrollo de estos. Los principales objetivos de dichas reuniones son:

- Determinar el estado actual de cada subsistema en cuanto a la interacción y el intercambio de información con los demás subsistemas que contribuyen al proceso de Superación Pedagógica.
- Realizar un análisis e identificar los principales problemas que actualmente presenta SASPED en materia de intercambio de información.
- Determinar el estado deseado de cada subsistema en cuanto a la interacción y el intercambio de información de manera que posibilite el correcto funcionamiento de SASPED.
- Identificar, evaluar y organizar un grupo de iniciativas que permitan solucionar los problemas interacción e intercambio de información identificados y que garanticen a la vez la transición de los subsistemas SASPED desde el estado actual al estado deseado.

En los análisis realizados durante estas reuniones, se constató que el estado actual de la interacción e intercambio de información entre los subsistemas es un completo aislamiento de cada uno de estos sistemas, que dificulta el correcto funcionamiento del proceso de Superación Pedagógica. También este problema ocasiona la lentitud en la toma de desiciones para dar una propuesta de estrategia de superación que debe seguir cada profesor.

Los problemas que ponen barreras a las interoperaciones de los subsistemas son los siguientes:

- El sistema que proporciona las encuestas se encuentra implementado en el lenguaje PHP y utiliza un Sistema Gestor de Bases de Datos MySQL, mientras que los subsistemas de SASPED utilizan código Java y Bases de Datos PostgreSQL, lo cual presenta incompatibilidades en los tipos de datos.
- Los subsistemas de SASPED no tienen ningún requisito dirigido a la interoperar.

Idealmente, el intercambio de información entre los subsistemas se debe realizar de la siguiente forma:

- El Subsistema Consultor de Información debe obtener datos de una encuesta, un encuestado, o las respuesta de un encuestado, datos que están almacenados en la base de datos del sistema de encuestas.
- Dicho Subsistema también debe realizar un intercambio de datos con el Subsistema Experto, donde este último requiere características de un profesor que deben ser obtenidas en una encuesta como son la evaluación de profesor y los controles a clase.
- El Subsistema Experto debe proveer una lista de cursos de un profesor al Subsistema Consultor de Información.

- El Subsistema Consultor de Información debe proporcionar al Subsistema de Análisis Semántico de Texto las respuestas a preguntas abiertas obtenidas de las respuestas de los encuestados.
- El Subsistema de Análisis Semántico de Texto debe proporcionar valores numéricos al Subsistema Consultor de Información a partir del procesamiento de las respuestas a las preguntas abiertas.

A raíz de los resultados arrojados por estos encuentros, se definen los principales pasos a seguir para el diseño y posterior implementación de una solución de interoperabilidad de los subsistemas de SASPED.

Soluciones de interoperabilidad en el ámbito internacional

A nivel internacional se han realizado investigaciones con el fin de dar respuesta a la creciente demanda de intercomunicación entre sistemas. A continuación se muestran algunas de las herramientas que se han desarrollado como producto de estas investigaciones.

Programa ISA

En el documento: “Los servicios públicos europeos de Administración electrónica: Programa ISA y estado de situación de la integración de la Administración”, disponible en (ISA, 2014), se expresa la necesidad del desarrollo de servicios transfronterizos en la región, con el objetivo de aumentar la movilidad de sus ciudadanos y las empresas en el mercado interior. En tal sentido se propone el uso del Programa ISA para el logro de estos objetivos.

Según el documento, el Programa ISA se centra en apoyar aquellos servicios que dan soporte a actos y políticas comunitarias y que cuenta con una dimensión esencialmente transfronteriza, basándose en el uso eficaz y eficiente de las tecnologías de la información. Sus principales esfuerzos estarían encaminados a la identificación de brechas en materia de interoperabilidad y reconocimiento mutuo, especialmente en el terreno de la identificación, firma electrónica y documento electrónico. Con tales objetivos, ISA adopta los principios de neutralidad con respecto a la tecnología y adaptación, apertura y reutilización, privacidad y protección de los datos personales y la seguridad.

Este programa sienta sus bases sobre los resultados obtenidos por sus antecesores: Entrega Interoperable de Servicios de Gobierno Electrónico Europeos a Administraciones Públicas, Negocios y Ciudadanos (IDABC), Intercambio de Datos entre Administraciones (IDA) e Intercambio de Datos entre Administraciones II (IDA II), los cuales son explicados a continuación.

El programa IDABC vio la luz en el periodo comprendido entre el año 2005 y el 2009. En esta etapa constituyó la base legal para el programa y entre sus objetivos fundamentales estuvieron: la identificación,

promoción y desarrollo los servicios paneuropeos de administración electrónica. Estuvo dirigido a las personas naturales o ciudadanos, a las empresas y a las administraciones, así como de las infraestructuras y servicios necesarios para el despliegue de los primeros. Por otra parte, el IDA II prestó sus servicios a la comunidad en el periodo de 1999 al 2004, constituyendo en esta etapa la base legal para el establecimiento de los servicios transeuropeos entre administraciones, denominados proyectos de interés común, para el soporte a la aplicación de políticas y actos comunitarios, el apoyo a la comunicación interinstitucional de la Unión Europea y el apoyo al proceso de decisión comunitario, así como el establecimiento de las denominadas Acciones y Medidas Horizontales, para interoperabilidad de infraestructuras, servicios y contenidos en apoyo del despliegue de los citados proyectos.

El programa IDA prestó sus servicios entre 1995 y 1999. Su principal objetivo fue abordar el establecimiento efectivo de la interoperabilidad de los sistemas de información de los estados miembros. IDA persiguió, en su momento, la contribución comunitaria determinados proyectos de ámbito sectorial, así como el establecimiento efectivo de la interoperabilidad de los sistemas de información de los estados miembros.

En este caso se observa el beneficio que reporta el uso de sistemas encaminados a garantizar la interoperabilidad. Aunque es necesario tener en cuenta que el Programa ISA está enfocado al desarrollo de la comunidad pública en específico, y no a los sistemas de superación pedagógica. Además, responde única y exclusivamente a los objetivos e intereses de la Unión Europea, financiado por empresas privativas que regulan su uso y distribución. También, como es de suponer, se desarrolla bajo condiciones y regulaciones muy diferentes a las de Cuba. Es válido destacar que aunque este sistema no opere en un marco de superación pedagógico y sus objetivos y metas no sean los mismos que las de la presente investigación, sí logra la interoperabilidad entre sus partes, logrando entrelazar los sistemas de información de: el Ministerio de la Industria, Turismo y Comercio, los Departamentos Ministeriales y las Comunidades Autónomas de la región.

Proyecto AGESIC

El proyecto AGESIC tiene como objetivo general facilitar y promover la implementación de servicios de Gobierno Electrónico en Uruguay. Con tal fin, esta herramienta brinda mecanismos que apuntan a simplificar integración entre los organismos del estado y a posibilitar un mejor aprovechamiento de sus activos. Técnicamente AGESIC posibilita que los organismos provean sus funcionalidades de negocio a través de servicios de software de forma independiente a la plataforma en la que fueron implementados.

Esto se debe a la implementación de una SOA a nivel de estado, en la cual los servicios ofrecidos por los organismos son descritos, publicados y descubiertos, invocados y combinados a través de interfaces y protocolos estandarizados. De esta forma, al facilitarse la reutilización de servicios en base a otros ya existentes, reduciéndose los tiempos de implementación de nuevos requerimientos. Por otro lado, el bajo acoplamiento entre servicios promovidos por la SOA, permite la evolución autónoma de los servicios de software implementados en los organismos.

Esta plataforma de interoperabilidad está basada en una arquitectura basada en servicios (SOA) e integrada por un sistema de control de acceso, un sistema de gestión de metadatos y una plataforma middleware. Estos componentes facilitan la provisión, búsqueda e invocación de servicios web que son brindados por los organismos, así como la interoperabilidad e interacción segura entre los mismos.

El sistema de control de acceso, integrado por una suite de productos que permiten la autenticación y autorización para el consumo de servicios basados en XML, oficia de puerta de entrada a la Plataforma.

El sistema de gestión de metadatos provee una especificación de alto nivel de los conceptos relativos a servicios públicos, de forma de evitar, o eventualmente resolver, ambigüedades en el manejo de estos conceptos por parte de los organismos.

Finalmente, el componente de middleware de la PDI cuenta con mecanismos para facilitar el desarrollo, despliegue e integración de servicios y aplicaciones. Está integrado por dos ESB (Enterprise Service Bus), uno de tecnología Microsoft y otro de tecnología Java, con el fin de obtener lo mejor de ambos y ampliar el espectro de posibilidades en cuanto a los métodos de conexión. Los organismos pueden utilizar esta plataforma para publicar y descubrir servicios, así como utilizar las diferentes capacidades de mediación, las cuales permiten desacoplar clientes y servicios.

Esta propuesta de cierto modo abarca algunas de las tecnologías en las que se pretende enfocar esta investigación, pero es válido señalar que no se desenvuelve en un entorno pedagógico, por lo que no constituye una respuesta a la problemática que se pretende resolver en la presente investigación. Además, su desarrollo está basado en gran medida en el uso de tecnologías privativas, las cuales requieren el pago de cierta cantidad de dinero para acceder a ellas, contrastando esto con las actuales necesidades económicas de nuestro país. No obstante, hay que destacar que posee variantes que son factibles incorporar a la solución que se pretende desarrollar, pues como se puede apreciar posee un sistema de control de acceso único mediante autenticación y autorización, utiliza servicios web y estándares abiertos para su desarrollo.

Proyecto de implementación del Decreto-Ley sobre el acceso e intercambio electrónico de datos, información y documentos entre órganos y entes del Estado (PNSII)

El propósito de este proyecto es promover la capacidad de colaboración entre las instituciones del Estado, permitiendo agilizar la realización de trámites administrativos, disminuyendo costos y evitando que el mismo dato sea solicitado a un ciudadano en diferentes ocasiones, reconociendo a las instituciones públicas como brazos articuladores que sirven efectivamente a los ciudadanos.

Según el documento "Implementación del Decreto-Ley sobre el Acceso e Intercambio Electrónico de Datos, Información y Documentos entre Órganos y Entes del Estado" en su primera fase (CNTI, 2012), la implementación de la interoperabilidad generará beneficios directos a los ciudadanos, simplificando trámites anteriormente burocráticos, y posibilitando a los órganos y entes del Estado a dedicar tiempo al cumplimiento de los lineamientos del ejecutivo, afianzando el desarrollo social, político y económico de la nación.

Con la definición de estándares y normativas, se proporcionará sustentabilidad a los componentes de plataformas de servicios de Gobierno Electrónico, bajo el empleo de Tecnologías de la Información Libres en las instituciones públicas. En este documento también se resalta que el proyecto se realizará en dos fases, la primera asociada a la atención de las necesidades mencionadas en el planteamiento del problema y la generación de insumos para la implementación de la Plataforma Nacional de Servicios de Información Interoperables (PNSII). Y una segunda fase para la puesta en producción de esta PNSII, una vez sea asignado por el Presidente de la República el ente ejercerá las funciones de la Interoperabilidad.

Esta propuesta de interoperabilidad, aunque posee objetivos muy similares al objetivo general y está comprendida dentro del campo de acción de esta investigación, no se desenvuelve en un entorno pedagógico. Además, no está orientada a las necesidades que apremia la situación del Sistema de Superación Pedagógico en Cuba. También presenta inconvenientes en cuanto a su puesta en marcha.

Soluciones de interoperabilidad en el ámbito nacional

En el contexto nacional también se han desarrollado hasta la fecha algunos proyectos que tributan a la integración de sistemas. Algunos de los más destacados por sus logros y resultados obtenidos en tal sentido se muestran a continuación.

Estrategia de interoperabilidad para la transferencia de datos entre sistemas ERP en Cuba

Mediante esta investigación, (Nogales, y otros, 2011) proponen un proceso de interoperabilidad que garantice el intercambio de información entre los sistemas integrales de gestión usados en Cuba. Como producto de la ejecución de la misma se puede garantizar el desarrollo de una plataforma de interoperabilidad que permita optimizar, organizar y estandarizar el proceso de interoperabilidad en el país.

La propuesta permite que se garantice el intercambio de información entre los diferentes sistemas actualmente desplegados en el país y el sistema integral de gestión CEDRUX, como escenario de aplicación de la propuesta. Aportó la construcción de una Plataforma de Interoperabilidad, además de proveer una comunicación en un lenguaje común a través de un estándar de intercambio, permitiendo absorber toda la información generada y evitando la pérdida, redigitalización y reprocesamiento de los datos.

Esta estrategia ha posibilitado estandarizar el proceso de interoperabilidad en cuanto a protocolos de comunicación, procesos a interoperar y orden de los mismos, estándares de intercambio y esquemas de seguridad.

Desde el punto de vista técnico, la solución se da en varios aspectos, como la construcción del ambiente de configuración para la gestión de los procesos a interoperar, donde se adicionan, modifican, eliminan, y se define la estructura que finalmente puede tener el estándar de intercambio de información. También se observa el desarrollo de un componente donde se gestiona el trabajo con XML y demás estándares que se utilicen, (IDABC, 2009), el cual es usado por los desarrolladores para organizar los elementos que forman las taxonomías. Otra característica de la solución es la creación de un ambiente de interoperabilidad, que cuenta con las funciones Exportar e Importar información, donde el usuario interactúa directamente con la plataforma y ordena la interoperabilidad del proceso que desee.

Propuesta de arquitectura de interoperabilidad para el intercambio de datos en el Órgano de Justicia-MININT

(Hernández, y otros, 2011) desarrollaron esta investigación con el objetivo de realizar una propuesta de arquitectura de interoperabilidad para el intercambio de datos en el Órgano de Justicia-MININT, a partir de la cual se pudiera implementar una plataforma informática que contribuyera a la eficiencia en el proceso de toma de decisiones en el dicho órgano.

Tecnológicamente estos sistemas no ofrecen una solución a la presente problemática, pues el primero de los casos se da respuesta a las necesidades de interoperabilidad de una gran cantidad de Sistemas de

Planificación de Recursos Empresariales a través del país, mediante la implementación de una Arquitectura Orientada a Servicios, lo cual no es aplicable en el desarrollo de la presente solución, debido a que SASPED se compone por un reducido número de subsistemas, para lo cual establecer dicha arquitectura sería excesivo.

Principales métodos de interoperabilidad

El intercambio de información entre sistemas es una problemática que se puede resolver de varias formas. Entre las más comunes se encuentran las que se mencionan a continuación.

Replicación de datos

Para (Mazilu, 2010), la replicación de datos es el proceso de crear y mantener múltiples instancias de la misma base de datos, y el proceso de compartir datos o cambios en los diseños de bases de datos en diferentes lugares sin tener que copiar la base de datos entera. Entre sus muchos beneficios se puede observar el alto rendimiento al procesar la lectura de los datos, ya que éstos se encuentran repartidos en diferentes nodos, que se encargan de responder a las múltiples peticiones de lectura, reduciendo la carga que suponen. También se puede apreciar que los sistemas que implementan estas soluciones son fiables hasta cierto nivel, pues al presentar problemas un nodo específico, otros pueden reemplazar la información solicitada, con lo cual se obtiene una mayor disponibilidad del servicio.

Por otra parte, optar por la replicación de datos tiene como consecuencia que al actualizarse la información, es necesario un mecanismo de propagación que actualice todos los nodos, con lo cual se acarrea un tiempo mayor que en otras soluciones para la interoperabilidad, como también se necesitará de mayor espacio de almacenamiento. También es notable que las técnicas de control de la concurrencia de recuperación de datos serán más complejas, lo cual puede aumentar los costos de desarrollar la solución.

Ya que una de sus características fundamentales es que se utiliza en entornos distribuidos, no es aplicable a la actual problemática.

Bases de datos centralizadas

En este tipo de soluciones se almacena la información de los distintos sistemas en una sola base de datos física. Como ventajas tiene que se evita la redundancia y la inconsistencia en los datos, además de que con un control central de la base de datos, un administrador puede garantizar que se cumplan normas establecidas para la gestión de la información y aplicar restricciones de seguridad.

Como desventajas, las bases de datos centralizadas tienen que, al encontrarse en un sólo lugar físico, en caso de ocurrir una eventualidad puede verse comprometida la disponibilidad de la información o perderse ésta completamente. También, como consecuencia de la centralización, las cargas de trabajo no se pueden repartir entre más computadoras, por lo cual se puede ver afectado el rendimiento.

Servicios Web

Luego de revisar lo planteado en (W3C, 2004), (W3C, 2012) y (Zapater, 2005), se pudo observar un consenso entre los diversos criterios de los autores que abordan el tema de la interoperabilidad. Todos concuerdan con que los servicios web constituyen la principal herramienta para combatir las insuficiencias en el intercambio de datos, así como en la comunicación entre sistemas heterogéneos. En la presente investigación, la interacción entre los subsistemas y el intercambio de datos entre estos se garantizará mediante documentos XML utilizando servicios web, ya que se necesita la estandarización de las comunicaciones para establecer interoperaciones desacopladas.

Según W3C, los servicios web proporcionan mecanismos de comunicación estándares entre diferentes aplicaciones, que interactúan entre sí para presentar información dinámica al usuario. Estos hacen posible la interacción entre aplicaciones software utilizando mensajes XML intercambiados mediante protocolos de internet. También plantean que para proporcionar interoperabilidad y extensibilidad entre estas aplicaciones, y que al mismo tiempo sea posible su combinación para realizar operaciones complejas, es necesaria una arquitectura de referencia estándar.

Por otro lado, (Zapater, 2005) plantea que un servicio web no es más que un sistema de software diseñado para soportar la interacción entre dos máquinas a través de una red. Estos poseen una interfaz descrita en un formato que puede ser procesada por una máquina (específicamente WSDL). Los servicios web pueden interactuar con otros sistemas gracias a la utilización de los estándares relacionados con la Web como son: SOAP para el envío de mensajes, HTTP para el transporte y el serializado con XML.

Debido a que el Subsistema de Gestión de Información y la base de datos del sistema de encuestas presentan tecnologías diferentes se hace visible la necesidad de estandarizar la información transmitida. Debido a esto, los servicios web son la preferencia cuando se establece una comparación con otras formas de resolver la interoperabilidad, en lo referente a la idoneidad para el desarrollo de la presente solución.

Estándar

Según (Tortosa, 2006), los estándares son acuerdos internacionales documentados o normas establecidas por consenso mundial. Contienen especificaciones técnicas y de calidad que deben reunir todos los productos y servicios para cumplir satisfactoriamente con las necesidades para las que han sido creados y para poder competir internacionalmente en condiciones de igualdad. Según la (ISO, 2014), un estándar es un documento que provee requisitos, puntos de guía o características que pueden ser utilizadas consistentemente para asegurar que materiales, productos, procesos y servicios cumplan con su propósito.

En otras palabras, los estándares no son más que un conjunto de reglas o de normas que deben cumplir todos los productos o servicios destinados al consumo de la comunidad nacional o internacional, teniendo en cuenta que estas reglas no son elaboradas por cualquier persona o institución, sino que son analizadas, documentadas, reguladas, justificadas y certificadas por un amplio grupo de expertos a nivel mundial en dicha esfera, rama de la ciencia o la sociedad.

La informática como ciencia al fin, no escapa a estas regulaciones. Los estándares son requerimientos obligatorios destinados a establecer un enfoque uniforme y disciplinado en el desarrollo de software. Para Microsoft, los estándares aseguran la interoperabilidad, el intercambio de datos y la portabilidad a través de la más amplia gama de productos y servicios, y traen grandes beneficios para los clientes (IEEE, 1990).

A nivel mundial existe un importante grupo de organizaciones que centran sus objetivos en el desarrollo y proposición de estándares para el desarrollo de software. Ejemplo de estas instituciones son OASIS (Organization for the Advancement of Structured Information Standards), World Wide Web Consortium (W3C) y GS1 (fusión de Uniform Code Council y European Article Number) las cuales han aprobado y puesto en marcha varios estándares de gran relevancia. Muchos se encuentran actualmente al servicio y disposición de la comunidad internacional como: UDDI (Universal Description, Discovery and Integration), XML (eXtended Markup Languaje), HTML (HyperText Markup Languaje), SOAP (Simple Object Access Protocol), WSDL (Web Services Description Language), BPEL (Business Process Execution Languaje), Open Document (OASIS Open Document Format for Office Applications), GS1 BarCodes y GS1 eCom por solo citar algunos.

Los estándares juegan un papel fundamental en el proceso de intercomunicación de sistemas. De ahí a que gran parte de la problemática de interoperabilidad entre sistemas, encuentre respuestas a sus

interrogantes en la aplicación de estándares de comunicación. Estos establecen las reglas generales que señalan a donde se debe llegar, aunque nunca muestran el camino ni como se debe llegar.

Como herramientas activas para la comunicación y la compatibilidad entre sistemas, los estándares se engloban en dos grandes grupos: estándares abiertos y estándares cerrados o propietarios. Según la organización de estándares W3C, los estándares abiertos son aquellos que se pueden implementar en una base libre de derechos y deben garantizar que cualquiera los pueda usar sin necesidad de pagar regalías o rendir condiciones a ningún otro, facilitando así, las comunicaciones entre distintos sistemas. Por otra parte, los estándares cerrados o propietarios, son aquellos que pertenecen a las compañías que los desarrollan, los cuales cierran o entorpecen las comunicaciones entre sistemas distintos.

Por las bondades que nos brinda y las características propias de la presente investigación, se decide trabajar con los estándares abiertos para dar solución al objetivo general de la misma. Resaltando la capacidad que poseen ambos grupos de ser ampliamente aceptados.

De acuerdo con lo planteado por (Harold & Means, 2002), **XML**, el Lenguaje Extensible de Marcado, es un estándar para marcado de documentos aprobado por el W3C. Define una sintaxis genérica utilizada para marcar datos con etiquetas simples y legibles. Provee un formato estándar para documentos digitales. Este formato es lo suficientemente flexible para ser modificado para dominios tan diversos como sitios web, intercambio de datos electrónicos, gráficas vectoriales, genealogías, serialización de objetos, llamadas de procedimientos remotos, sistemas de buzones de voz y más.

En la página oficial de XML, (O'Reilly Media, 2014) expresa que XML fue creado para que documentos enriquecidos y estructurados pudieran ser utilizados en toda la Web. Las únicas alternativas viables, HTML y SGML (Standard Generalized Markup Language), no eran prácticas para este propósito. HTML está atado a un grupo de semánticas y no provee una estructura arbitraria. SGML provee esta estructura, pero es muy difícil de implementar solo para un navegador Web. Sistemas SGML completos resuelven problemas largos y complejos que justifican el esfuerzo de utilizar esta herramienta.

Según el (W3C, 1999), **HTTP** (Hypertext Transfer Protocol), es un protocolo a nivel de aplicación para sistemas informáticos distribuidos, colaborativos o de hipermedios. Es un protocolo genérico y sin estado que puede ser utilizado en varias tareas más allá de su uso por hipertexto, como nombres de servidores y sistemas gestores de objetos distribuidos, a través de extensiones de sus métodos de peticiones, códigos de error y encabezados.

Dada la heterogeneidad de los subsistemas de SASPED con respecto a la base de datos del sistema de encuestas, debido a que presentan diferentes maneras de almacenar la información, es necesario seleccionar un grupo de estándares que garanticen formas de transmitir datos en un formato común que sea entendible para ambas partes y que se abstraigan de la forma en que se encuentran implementados dichos sistemas.

En el documento Arquitectura para los Sistemas que Conforman la Intranet Universitaria (UCI, 2007) se describen algunos estándares cuya utilización será necesaria en la presente investigación. Sus definiciones aparecen a continuación.

WSDL o Web Services Description Language, es un formato XML que se utiliza para describir una interfaz pública para servicios Web, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

El (W3C, 2001) define WSDL como un formato XML para describir servicios como un conjunto de puntos de destino que operan en mensajes que contienen información orientada tanto a documentos como a procedimientos. Las operaciones y mensajes se describen abstractamente, luego se utilizan en un protocolo de red concreto y un formato de mensaje para definir un punto de destino. Puntos de destino concretos relacionados son combinados en puntos de destino abstractos o servicios. WSDL es extensible para permitir descripciones de puntos de destino y sus mensajes sin necesidad de conocer que formatos de mensajes o protocolos de red se utilizan para la comunicación.

UDDI, o Universal Description, Discovery, and Integration, es una iniciativa industrial abierta, en el contexto de los servicios Web. El registro de un negocio en una UDDI cuenta con tres partes:

- Páginas blancas: dirección de contacto y otros identificadores conocidos.
- Páginas amarillas: categorización industrial basada en taxonomías.
- Páginas verdes: información técnica sobre los servicios que aportan las propias empresas.

Acerca de UDDI, la (OASIS, 2005), expresa que estas especificaciones definen una un servicio de registros para servicios Web y para otros servicios electrónicos y no electrónicos. Un servicio de registro UDDI es un servicio Web que gestiona información acerca de proveedores de servicios, implementaciones de servicios y metadatos de servicios. Los proveedores de servicios pueden utilizar UDDI para dar a conocer los servicios que ofrecen. Los consumidores de servicios pueden utilizar UDDI para descubrir

servicios que cumplan con sus requerimientos y para obtener los metadatos de los servicios necesitados para consumirlos.

Las especificaciones UDDI definen:

- APIs de SOAP que utilizan las aplicaciones para consultar y publicar información a un registro UDDI.
- Esquemas de XML del modelo de datos del registro y los formatos de mensaje de SOAP.
- Definiciones WSDL de las APIs de SOAP.
- Definiciones de registros de UDDI (modelos técnicos) de varios identificadores y sistemas de categoría que puedan ser utilizados para identificar y categorizar registros de UDDI.

(Berners-Lee & otros, 2005) Plantean que un Identificador Universal de Recurso (**URI**), es una secuencia compacta de caracteres que identifica un recurso abstracto o físico. Permite la identificación uniforme de recursos a través de un grupo extensible de esquemas de nombres definidos.

Un URI puede ser clasificado como un localizador, un nombre, o ambos. El término Localizador Uniforme de Recurso (URL) se refiere a un subgrupo de URI que, además de identificar un recurso, provee medios para localizar el recurso describiendo su mecanismo de acceso primario. El término Nombre de Recurso Uniforme (URN) ha sido utilizado históricamente para referirse tanto a los URI bajo el esquema URN, de los cuales se requiere que permanezcan únicos globalmente y persistentes incluso cuando el recurso deja de existir o se hace no disponible, así como cualquier otro URI con las propiedades de un nombre.

Para el envío de los datos estandarizados, es necesario un protocolo que permita la transmisión de los servicios web, así como una esquemática para su construcción. Según el documento Arquitectura de Servicios Web(W3C, 2004), para la construcción de servicios web se pueden establecer dos arquitecturas: **REST** (Representational State Transfer), que de acuerdo con (Flanders, 2009), es un estilo arquitectónico que está caracterizado fundamentalmente por el establecimiento mediante una interfaz uniforme de los comandos estándares de HTTP (GET, POST, PUT y DELETE); o **servicios web arbitrarios,** en los cuales el servicio presenta un grupo arbitrario de operaciones. Dado que la base de datos del sistema de encuestas, de la cual se deben obtener datos mediante servicios, presenta una estructura para la cual se necesita implementar métodos de extracción específicos, se construirán dichos servicios de forma arbitraria.

Para la transmisión de los XML que contienen la información, se necesitan protocolos estándares, es decir que sean reconocibles por los sistemas involucrados en la problemática. Para la interconexión entre los

subsistemas de SASPED se seleccionará **HTTP** (HyperText Transfer Protocol) por su alta compatibilidad con las tecnologías utilizadas en el desarrollo. Para la comunicación entre los subsistemas consultores de información y la Base de Datos del Sistema de Encuestas, dada la heterogeneidad constatada, acuerdo con el W3C (W3C, 2004), se escogerá el protocolo **SOAP**.

Protocolo SOAP

De acuerdo con (W3C, 2000), Simple Object Access Protocol (SOAP) es un protocolo ligero para intercambio de información en un ambiente distribuido y descentralizado. Está basado en XML que consiste en 3 partes: una capa que define un framework para describir qué contiene un mensaje y como procesarlo, un grupo de reglas de codificación para expresar instancias de tipos de datos definidos por la aplicación y una convención para representar llamadas y respuestas a procedimientos remotos (RPC).

Por otra parte, (Skonnard, 2003) establece que SOAP provee un simple, extensible y rico framework de mensajes XML que define protocolos de aplicaciones de alto nivel, incrementando la interoperabilidad en sistemas distribuidos y heterogéneos.

Definidos ya los objetivos a implementar, se hace necesario la utilización de un marco de trabajo para estructurar y controlar el proceso de desarrollo, es decir, una **metodología de desarrollo de software**.

Metodología de desarrollo

De acuerdo con (Avison, y otros, 1995), “una metodología es una colección de procedimientos, técnicas, herramientas y documentos auxiliares que ayudan a los desarrolladores de software en sus esfuerzos por implementar nuevos sistemas de información. Una metodología está formada por fases, cada una de las cuales se puede dividir en sub-fases, que guiarán a los desarrolladores de sistemas a elegir las técnicas más apropiadas en cada momento del proyecto y también a planificarlo, gestionarlo, controlarlo y evaluarlo”. A lo largo del tiempo, una gran cantidad de métodos han sido desarrollados diferenciándose por su fortaleza y debilidad. El marco de trabajo para metodología de desarrollo de software consiste en una filosofía de desarrollo de aplicaciones informáticas, un enfoque del proceso de desarrollo de software, así como herramientas, modelos y métodos para asistir al proceso de desarrollo de software.

Dados los factores presentes en el desarrollo de la investigación, como por ejemplo el tamaño del equipo de desarrollo, la complejidad de la arquitectura y una necesidad de agilidad, para establecer una guía para un proceso de desarrollo del software que satisfaga dichas características, se asumirá la metodología de desarrollo de software **OpenUP.**

Metodología de desarrollo de software OpenUP.

OpenUP es un proceso de desarrollo de software mínimamente suficiente, es decir, que solo se incluye contenido fundamental. Aunque no provee guía en varios temas como grandes tamaños de equipo, o a una tecnología específica, es completo en el sentido de que se puede manifestar como un proceso completo para construir un sistema.

OpenUP es un proceso ágil que contiene las características fundamentales de un Proceso Unificado que aplica enfoques iterativos e incrementales dentro de un ciclo de vida estructurado. Está basado en casos de uso, gestión de riesgos, y un enfoque centrado en la arquitectura para guiar el desarrollo.

La factibilidad de esta metodología para el desarrollo de la presente se evidencia al analizar aspectos como el equipo de desarrollo, que está compuesto por dos personas, para lo cual OpenUP permite incrementar las probabilidades de éxito; las interacciones en dicho equipo y con otros relacionados en el desarrollo de SASPED, la prioridad de que exista una solución en funcionamiento, que interconecte los subsistemas para facilitar sus respectivos desarrollos, con prioridad sobre una extensa documentación explicativa, así como el tiempo disponible para construir dicha solución, son aspectos que según (Pressman, 2010), implican la necesidad de un proceso ágil.

El proceso tendrá definidas las fases del Proceso Unificado: Inicial, Elaboración, Construcción y Transición, como también las disciplinas del desarrollo de software: requisitos, arquitectura, desarrollo y pruebas. (Balduino, 2007).

Luego de establecer la metodología de desarrollo para la solución, es necesario identificar la arquitectura del sistema, las herramientas y tecnologías que facilitan la construcción. A continuación se aborda dicha selección.

Arquitectura de Software

Una arquitectura de software es la organización fundamental de un sistema representado en sus componentes, las relaciones entre éstos y el medio, así como los principios que guían su diseño y evolución. (ISO, 2011). La arquitectura de software como disciplina persigue crear técnicas para describir sistemas de software existentes, prescribiendo como construir un nuevo sistema de software y evaluar la configuración de otros (Kiewelekar, 2014).

Dada la necesidad de desarrollar una solución que intercomunique los subsistemas de SASPED, de forma que garantice la transmisión segura y desacoplada de la información entre dichos subsistemas, así como

que permita a los usuarios percibir la interacción con los subsistemas como un todo para que puedan resolver los problemas de la ESP, se propone utilizar una arquitectura **Cliente Servidor**.

Según (IBM, 2010), mediante la utilización de esta arquitectura se proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo y/o a través de la organización, en múltiples plataformas. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones clientes, resultan en un trabajo realizado por otras computadoras llamadas servidores. Es un modelo para el desarrollo de sistemas de información en el que las transacciones se dividen en procesos independientes que cooperan entre sí para intercambiar información, servicios o recursos. Se denomina cliente al proceso que inicia el diálogo o solicita los recursos y servidor al proceso que responde a las solicitudes.

El cliente es usualmente la parte de la aplicación que es vista por el usuario. Debido a esto, la parte del cliente en una aplicación cliente servidor generalmente reside en una estación de trabajo, donde el usuario puede interactuar con la aplicación a través de la interfaz gráfica del sistema operativo de la estación de trabajo.

Los servidores, por otra parte, son usualmente transparentes al usuario, lo cual significa que la persona que utiliza la estación de trabajo solo percibe la parte del cliente de la aplicación, la parte que muestra la información. Debido a que solo un servidor se encarga de un grupo definido de clientes, los servidores proveen una manera ideal de gestionar el acceso compartido a los recursos del sistema. Por esta razón, los servidores generalmente se encuentran en máquinas con mejor rendimiento.

Entre sus características fundamentales se puede observar la facilidad de mantenimiento, pues es posible intercambiar, reparar, mejorar e incluso relocalizar un servidor sin que los clientes se percaten o sean afectados por el cambio. Todos los datos se almacenan en los servidores, los cuales generalmente tienen mejores controles de seguridad que la mayoría de los clientes. Dado que el almacenamiento de datos es centralizado, las modificaciones a esos datos son mucho más sencillas, ya que los datos no necesitan ser distribuidos y aplicados a cada aplicación en la red, lo cual consume más tiempo y es proclive a errores. Existen tecnologías avanzadas disponibles que implementan cliente servidor, las cuales han sido diseñadas para proporcionar seguridad, interfaces intuitivas y facilidad de uso.

Patrón Modelo-Vista-Controlador

Para el actual proceso de desarrollo existe la necesidad de dividir las responsabilidades asociadas al almacenamiento, la entrada y salida de información y el control del flujo de datos. Se necesita una

esquemática para la utilización de las herramientas de generación de código, que utilicen arquitecturas dirigidas a la separación de las responsabilidades asociadas a las vistas, el control de la aplicación y el almacenamiento de los datos. Para satisfacer todas estas condiciones, el patrón arquitectónico con mayor probabilidad de ofrecer resultados positivos es el Modelo Vista Controlador. Tomando en cuenta las características de las herramientas que se propone construir, la arquitectura de las clases vista de dicho patrón, responde al problema de la navegación; la arquitectura de las clases controladoras responde al problema del negocio del sistema; y la arquitectura de las clases modelo responde al problema del almacenamiento y el flujo de los datos.

También para establecer una guía en la construcción de aplicaciones web, (Pressman R. , 2010) sugiere el diseño de una arquitectura en 3 capas que desacople la interfaz del comportamiento de navegación y el de aplicación, lo cual simplifica la implementación y facilita la reutilización. Uno de los patrones arquitectónicos que desacopla la interfaz de usuario, las funcionalidades y el contenido informacional es el Modelo-Vista-Controlador (MVC). El modelo posee el contenido específico de la aplicación y la lógica de procesamiento, incluyendo los objetos del contenido, el acceso a las fuentes externas de datos, así como las funcionalidades de procesamiento específicas de la aplicación. La vista contiene las funciones específicas de las interfaces y provee la presentación del contenido y la lógica de procesamiento que requiere el usuario. Por su parte, el controlador administra el acceso al modelo y a la vista y coordina el flujo de datos entre ellos.

Luego de establecer la el estilo y el patrón arquitectónico de la solución, es necesario definir las herramientas que se utilizarán en el proceso de desarrollo. Dichas herramientas se muestran a continuación.

Herramientas para el desarrollo de la solución

Lenguaje de programación

Según (Ralston, y otros, 2000) un lenguaje de programación es un conjunto de caracteres, reglas para combinarlos, y normas que especifican sus efectos cuando son ejecutadas por un ordenador, que tienen las cuatro características siguientes:

- No requiere ningún conocimiento de código de máquina por parte del usuario.
- Define un proceso que se ejecuta en un computador.
- Es de alto nivel, cercano a los problemas que se quieren resolver, permitiendo la abstracción.

- Permite construir nuevas abstracciones que se adapten al dominio que se programa.

Un potente lenguaje de programación es algo más que un medio para instruir a una computadora para realizar tareas. El lenguaje también sirve como un marco dentro del cual organizamos nuestras ideas acerca de los procesos. Así, cuando se describe un idioma, hay que prestar especial atención a los medios que la lengua ofrece para combinar ideas simples para formar ideas más complejas.

Otros autores como (Berthet, 1988), plantean que un lenguaje de programación es un lenguaje exterior o evolucionado (de alto nivel), en oposición a los lenguajes de máquina y lenguajes ensambladores (bajo nivel), que se utiliza para describir un conjunto de acciones consecutivas, permitiendo crear programas mediante instrucciones, operadores y reglas de sintaxis para comunicarse con los dispositivos hardware y software de una máquina. Dispone de formas adecuadas que permiten ser leídas y escritas por personas, que a su vez resultan independientes del modelo de computador a utilizar. Está formado por símbolos y reglas sintácticas y semánticas que definen su estructura y el significado de sus elementos y expresiones.

Dentro de los lenguajes de programación de alto nivel que más se utilizan en el mundo para desarrollar aplicaciones informáticas están: Hipertext Markup Languaje (HTML), Cascading Style Sheets (CSS), Java, C, C++, C#, PHP, Pascal, entre otros. Cada uno de estos con sus propias ventajas y desventajas con respecto a los demás, pero todos con capacidad de ser aceptados.

Para la implementación de la solución de interoperabilidad se determina utilizar el **lenguaje de programación Java**, que se describe a continuación.

Java

Java es un lenguaje de programación orientado a objetos. Para (Flanagan, 1999), esto significa que el programador debe poner atención especial a los datos de aplicación y a los métodos que manipulan los datos, y que no debe pensar estrictamente en términos de procedimiento.

Java es un lenguaje intérprete, pues su compilador genera bytecode para la máquina virtual (Java Virtual Machine - JVM). Como los bytecode de Java no dependen de plataforma, sus programas se pueden ejecutar en cualquier plataforma en la que la JVM haya sido instalada (el intérprete y el entorno en tiempo de ejecución).

Como los programas de Java se compilan en un formato de bytecode de arquitectura neutral, una aplicación de Java se puede ejecutar en cualquier sistema, siempre y cuando dicho sistema instrumente la máquina virtual de Java. Esto resulta muy importante para las aplicaciones distribuidas en Internet u otras redes heterogéneas.

Otras características de Java son que es un lenguaje dinámico, ya que en todo momento es posible cargar una clase en un intérprete en ejecución, así como su simplicidad, fácil aprendizaje, robustez, seguridad y alto desempeño.

Dicho lenguaje se utilizó para el desarrollo de la solución de interoperabilidad luego de tomar en cuenta las características de simplicidad, la robustez, el alto desempeño y fundamentalmente la experiencia del equipo de desarrollo con el lenguaje.

Para la implementación en la solución del control de acceso a SASPED, se adoptó el framework **JBoss Seam**, que se describe a continuación.

JBoss Seam

Para centrarse en resolver la intercomunicación de los subsistemas de SASPED, se necesita de herramientas que se abstraigan de problemas generales de aplicaciones empresariales como la concurrencia, las transacciones, la persistencia y la seguridad.

JBoss Seam es un framework del lenguaje Java que combina a los dos frameworks Enterprise JavaBeans EJB3, que atiende los problemas generales anteriormente descritos, y JavaServerFaces JSF, que simplifica el desarrollo de interfaces de usuario en aplicaciones Java EE. Se puede acceder a cualquier componente Enterprise JavaBeans (EJB) desde la capa de presentación refiriéndose a él mediante su nombre de componente seam.

Seam introduce el concepto de contextos. Cada componente de Seam existe dentro de un contexto. El contexto conversacional por ejemplo captura todas las acciones del usuario hasta que éste sale del sistema o cierra el navegador - inclusive puede llevar un control de múltiples pestañas y mantiene un comportamiento consistente cuando se usa el botón de regresar de el navegador.

Este framework facilita que se pueda automáticamente generar una aplicación web a partir de una base de datos existente utilizando una herramienta de línea de comandos llamada seam-gen incluida con el framework. También contiene las JBoss Tools, que es un conjunto de plug-ins diseñados para el entorno integrado de desarrollo Eclipse. Seam puede ser integrado con las librerías de componentes RichFaces, JSF, Seam User Interface (UI) y ajax4jsf.

Para el desarrollo de la solución utilizando las tecnologías mencionadas, se necesita un **Entorno de Desarrollo Integrado** que facilite la implementación.

Eclipse

La implementación de la presente solución requiere de un Entorno de Desarrollo Integrado que sea versátil, dada la variedad de tecnologías con que se necesita trabajar. Otros factores como el código abierto, la amplia documentación existente y el conocimiento previo del equipo de desarrollo, llevaron a la selección de Eclipse.

Eclipse es un entorno de desarrollo integrado, de Código abierto y Multiplataforma. Según el proyecto que lo desarrolla, es «una especie de herramienta universal - un IDE abierto y extensible para todo y nada en particular». Mayoritariamente se utiliza para desarrollar lo que se conoce como "aplicaciones de cliente enriquecido", opuesto a las aplicaciones "cliente-liviano" basadas en navegadores. Es una potente y completa plataforma de programación, desarrollo y compilación de elementos tan variados como sitios web, programas en C++ o aplicaciones Java. Entre sus ventajas principales se observa que emplea módulos para proporcionar toda su funcionalidad al frente de la plataforma de cliente enriquecido, a diferencia de otros entornos monolíticos donde las funcionalidades están todas incluidas, las necesite el usuario o no. Este mecanismo de módulos es una plataforma ligera para componentes de software. Adicionalmente a permitirle a Eclipse extenderse usando otros lenguajes de programación como son C/C++ y Python, le permite trabajar con lenguajes para procesado de texto como LaTeX, aplicaciones en red como Telnet y sistemas de gestión de base de datos.

Servidor de aplicaciones JBoss

JBoss es un servidor de aplicaciones JavaEE de código abierto implementado en Java puro. Al estar basado en Java, JBoss puede ser utilizado en cualquier sistema operativo para el que esté disponible la máquina virtual de Java. JBoss Application Server es el primer servidor de aplicaciones de código abierto, preparado para la producción y certificado J2EE 1.4, disponible en el mercado, ofreciendo una plataforma de alto rendimiento para aplicaciones de comercio electrónico. Está caracterizado por presentar licencia de código abierto sin coste adicional, es confiable a nivel de empresa, presenta flexibilidad consistente, funciona como middleware para cualquier objeto de Java y provee soporte completo para Java Management eXtensions (JMX).

Sistema gestor de bases de datos

El estilo arquitectónico seleccionado en la investigación establece una capa encargada de la persistencia a los datos, por lo cual se hace necesaria la selección de un Sistema de Gestión de Bases de Datos que se ocupe del almacenamiento.

Los Sistemas Gestores de Bases de Datos (SGBD), son un conjunto de programas que permiten el almacenamiento, modificación y extracción de información de una base de datos. Estos proveen al usuario un grupo de herramientas que le permiten añadir, borrar, modificar y analizar los datos. Los SGBD también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe. A nivel mundial existe un importante grupo de herramientas que se dedican a gestionar bases de datos, tales como: Microsoft SQL Server, MySQL y PostgreSQL. Se escoge esta última para la implementación de la solución de interoperabilidad por ser de código abierto y por la experiencia previa que tiene el equipo de desarrollo con esta tecnología.

PostgreSQL

PostgreSQL, es un Sistema de Gestión de Bases de Datos Objeto-Relacionales con énfasis en la extensibilidad y el cumplimiento de estándares. Como servidor de bases de datos, su función primaria es almacenar información con seguridad y utilizando buenas prácticas, y recuperarla luego, si se otras aplicaciones lo requieren, estén en la misma computadora o en otra a través de una red. Puede manejar cargas que van de aplicaciones de una sola computadora a grandes aplicaciones de internet con muchos usuarios concurrentes. Versiones recientes también proveen replicación de la base de datos para garantizar seguridad y escalabilidad. De este gestor de bases de datos también se puede decir que se encuentra bastante diseminado a través de varios sistemas operativos incluyendo Linux, FreeBSD, Solaris, Microsoft Windows y Mac OS X.

Servidor de aplicaciones

Para poder ejecutar las aplicaciones que componen SASPED, se necesita un servidor que las proporcione a un cliente.

Un servidor de aplicaciones, o servidor web, como también se le conoce, es un programa que gestiona cualquier aplicación del lado del servidor realizando conexiones bidireccionales y/o unidireccionales y sincrónicas o asincrónicas con el cliente, generando una respuesta en cualquier lenguaje o aplicación en el lado del cliente. El código recibido por el cliente suele ser compilado y ejecutado por un navegador web. Entre los principales servidores de aplicaciones que se existen en el mundo se encuentran: JBoss, Nginx, Cherokee, Tomcat y Apache.

Por la necesidad de la presente investigación de desarrollar servicios web de forma sencilla y eficiente en lenguaje Java, se escoge **WSO2 AS** para la implementación ya que presenta las siguientes características.

WSO2 AS

El Web Service Application Server o AS, es una herramienta para el almacenamiento de servicios desarrollados bajo diferentes frameworks de desarrollo como Axis, Axis2, JAX-WS, Servicios Spring y JAR. Brinda amplias funcionalidades en materia de configuración de los servicios, generando los WSDL de los mismos, así como permitiendo la realización de pruebas funcionales, soporta la validación del cumplimiento de WS-I para verificar la interoperabilidad de los servicios desplegados. Además permite adoptar mecanismos de restricción de acceso a servicios a través de filtros por direcciones IP o por subredes y dominios. Permite además la exposición de EJB como servicios así como un completo cumplimiento con los estándares WS. Posee una interfaz gráfica sumamente amigable como el resto de los productos de WSO2 que permite una fácil utilización de la herramienta; así como integración con un IDE de desarrollo, en este caso el Eclipse.

Para ejecutar los servicios web que se implementarán, es necesario un motor de servicios web que sea eficiente en cuanto a la rapidez, el consumo de memoria y que soporte WSDL. A partir de estos indicadores, se escoge **Axis2** como motor de servicios web.

Axis2

Apache Axis2 es un motor de servicios web SOAP. La arquitectura de Axis2 es más flexible, eficiente, modular, más orientada a XML y configurable en comparación con la de Axis1. No solo ofrece soporte para SOAP 1.1 y SOAP 1.2, sino que también soporta servicios web REST. Se diseñó teniendo en mente el soporte a módulos adicionales que extienden su funcionalidad para características como seguridad y confiabilidad. Entre sus características más notables se puede observar su rapidez, necesidad de poco espacio en memoria, capacidad de despliegue en tiempo de ejecución, es decir, que es posible desplegar nuevos servicios web sin tener que apagar el servidor. También se puede apreciar su flexibilidad, pues el desarrollador tiene la posibilidad de insertar varias extensiones en el motor. Axis2 También soporta WSDL, en su versión 1.1 y 2.0.

UML

Para el modelado de la presente solución se utiliza el Lenguaje Unificado de Modelado (UML). UML es el lenguaje de modelado de sistemas de software más conocido y utilizado en la actualidad. Es un lenguaje útil para visualizar, especificar, construir y documentar un sistema. Se puede aplicar en el desarrollo de software en gran variedad de formas para dar soporte a una metodología de desarrollo, pero no especifica en sí mismo qué metodología o proceso usar. UML cuenta con varios tipos de diagramas, los cuales

muestran diferentes aspectos de las entidades representadas como lo son los diagramas de clases, componentes, casos de uso, secuencia, actividades, etc.

Conclusiones parciales

- La necesidad de establecer interoperaciones de los subsistemas de SASPED, así como de obtener datos vitales para la automatización de la ESP a través de la base de datos del sistema de encuestas, constata de que existe un problema de interoperabilidad.
- La selección de un modelo que considere los niveles técnico, sintáctico, semántico y organizacional, garantiza la evaluación de la interoperabilidad entre los subsistemas de SASPED.
- Mediante la utilización de una arquitectura Cliente - Servidor y herramientas como el lenguaje Java con su framework Jboss Seam, MySql, PostgreSQL, y el servidor de aplicaciones WSO2 AS, siguiendo la metodología de desarrollo de software OpenUP, se posibilita la creación de una solución que logre la interoperabilidad entre los sistemas de SASPED.

CAPÍTULO NO. 2: ANÁLISIS Y DISEÑO DE LA SOLUCIÓN DE INTEROPERABILIDAD PARA LOS SUBSISTEMAS DE SASPED

En el presente capítulo se procede a caracterizar la concepción de interoperabilidad de los subsistemas informáticos de SASPED. Debido a que la metodología orientada a servicios, establecida como guía para trazar el camino en la presente investigación, se enmarca en este caso en el proceso de desarrollo ágil, se procederá en el presente flujo de trabajo a realizar las disciplinas de requisitos y arquitectura. Como objetivo se tiene entender el negocio de SASPED, el problema de dominio que se observa en el mismo y determinar una solución viable para resolverlo.

Definición conceptual y operacional de las variables

Luego de revisar lo planteado por (Hernández & otros, 2006) y (Hernández & Coello, 2002), en la investigación se realizó la definición conceptual y operacional de las variables. La variable dependiente en la investigación es la **interoperabilidad entre subsistemas de SASPED**, cuya definición se presentó anteriormente. A continuación se muestra su operacionalización:

Tabla 1. Operacionalización de la variable interoperabilidad entre subsistemas de SASPED

Dimensión	Subdimensión	Indicador	Escala de valoración
Técnica	Compatibilidad	Grado de integración en la implementación	Alto: si todos los subsistemas están implementados con las mismas tecnologías. Medio: si 2 o 3 de los subsistemas están implementados con las mismas tecnologías. Bajo: si todos los subsistemas están implementados con tecnologías diferentes
	Desempeño	Tiempo de Interoperación	Eficiente: si TI <= 1 segundo. Medio: si TI >1 segundo, TI < 2 segundos. Deficiente: si TI >2 segundos.
		Calidad del intercambio de datos	Alta: si el proceso de intercambio de datos se realiza sin problemas. Media: Si el proceso de intercambio de datos presenta problemas. Baja: Si el proceso de intercambio de datos no se realiza.
		Calidad de uso	Alta: si se envía el 100% de los datos pedidos por un subsistema. Baja: si el subsistema recibe más o menos información de la que pidió.

Dimensión	Subdimensión	Indicador	Escala de valoración
		Calidad de conformidad	Alta: si la información recibida por un sistema es utilizable. Baja: si la información recibida por un sistema no es utilizable.
Compa tibilidad Sintáctica	-	Compatibilidad en la sintaxis de la información a intercambiar	Compatible: si la sintaxis es la misma. Incompatible: si la sintaxis no es la misma.
Compatibili dad Semántica	-	Compatibilidad en el significado de la información a intercambiar	Compatible: si el significado es el mismo. Incompatible: si el significado no es el mismo.
Compatibili dad Organizacio nal	-	Grado de definición de las autoridades y responsabilidad es en los subsistemas	Alto: Los roles del sistema se encuentran bien definidos. Bajo: No están definidos los roles del sistema

Caracterización de la concepción de interoperabilidad de subsistemas de SASPED

Luego de realizar la operacionalización de la variable de la investigación, se utilizan sus indicadores para establecer el estado actual del objeto de estudio concepción de interoperabilidad de subsistemas de SASPED.

Para la constatación empírica de la integración de los subsistemas de SASPED a través de la implementación de la solución de interoperabilidad, se recolectaron datos a través de reuniones con el equipo de desarrollo de SASPED, con el objetivo de obtener información referente a las dimensiones Técnica y Compatibilidad Semántica. Para la recolección de información relacionada con la Compatibilidad Organizacional, se sostuvieron encuentros con el cliente y se analizó lo planteado en la Propuesta de Estrategia de Superación del Claustro (Ciudad, y otros, 2013) con respecto a las responsabilidades del personal docente relacionado con la ESP.

El análisis cualitativo y cuantitativo de los datos recolectados, permitió establecer las siguientes características de la concepción de interoperabilidad de subsistemas de SASPED:

- **Grado medio de integración en la implementación**, evidenciado en la utilización de código Java en los tres subsistemas de SASPED, con el Gestor de Bases de Datos PostgreSQL, mientras que el Sistema de Encuestas se construyó en PHP y utiliza para la persistencia de los datos MySQL.
- **Tiempo de interoperación deficiente**, puesto que la comunicación entre subsistemas no está automatizada.
- **Compatibilidad sintáctica entre los subsistemas de SASPED**, evidenciada también en la utilización de Java como lenguaje de implementación.
- **Incompatibilidad sintáctica entre el Subsistema Consultor de Información y el Sistema de Encuestas**, debido a que los datos almacenados en el Sistema de Encuestas son incompatibles con PostgreSQL.
- **Compatibilidad semántica entre los subsistemas de SASPED y el Sistema de Encuestas**, pues al presentar semejantes modelos conceptuales de la problemática, sus modelos de datos también tienen un alto grado de semejanza.
- **Autoridades y responsabilidades no definidas en los subsistemas**, existiendo libre acceso a sus datos y funcionalidades.

No fue posible evaluar las calidades de intercambio, de uso y de conformidad, puesto que la presente solución no se encontraba implementada.

Modelo conceptual de SASPED

A continuación se representan los principales conceptos asociados al dominio de la problemática.

- **Universidad:** Universidad de las Ciencias Informáticas (UCI).
- **CICE:** Centro de Innovación y Calidad de la Educación de la UCI.
- **Facultad:** Subgrupo organizacional de profesores y estudiantes de la universidad.
- **Departamento:** Subgrupo organizacional de profesores que pertenecen a una facultad.
- **Vicedecano de Investigación y postgrado:** Cargo de la persona que dirige la superación pedagógica en la facultad.
- **Jefe de Departamento Docente:** Cargo de la persona que dirige un departamento de profesores.
- **Categoría Docente:** Condición profesional del profesor.

- **Vías de obtención del conocimiento:** Maneras en las que se puede obtener información acerca de los profesores.
- **Autoevaluación:** Evaluación que un profesor hace de sí mismo.
- **Evaluación Profesoral:** Examen que se realiza al profesor.
- **Control a Clases:** Comprobación o inspección que se realiza por parte de un directivo a un profesor.
- **Elemento Evaluativo:** Medio en que se desarrolla una evaluación.
- **Característica del Profesor:** Cualidad que posee un profesor.
- **SASPED:** El objetivo del Sistema Automatizado de Superación Pedagógica, es proveer una solución automatizada a las exigencias de la Estrategia de Superación Pedagógica.
- **Sistema de Encuestas:** Sistema para la gestión de encuestas en la universidad.
- **Encuesta:** Encuesta de superación pedagógica.
- **Pregunta:** Una pregunta de una encuesta de superación pedagógica.
- **Pregunta Abierta:** Pregunta que arroja una respuesta que no puede ser determinada.
- **Pregunta Cerrada:** Pregunta que arroja una respuesta que puede ser determinada.
- **Análisis Cualitativo:** Análisis que denota una cualidad.
- **Análisis Cuantitativo:** Análisis que denota una cantidad.
- **Profesor:** Principal beneficiario de SASPED. Se beneficia con una estrategia de superación pedagógica que provee el sistema.
- **Estrategia de Superación Pedagógica:** Propone un cambio al modelo actual de superación pedagógica.

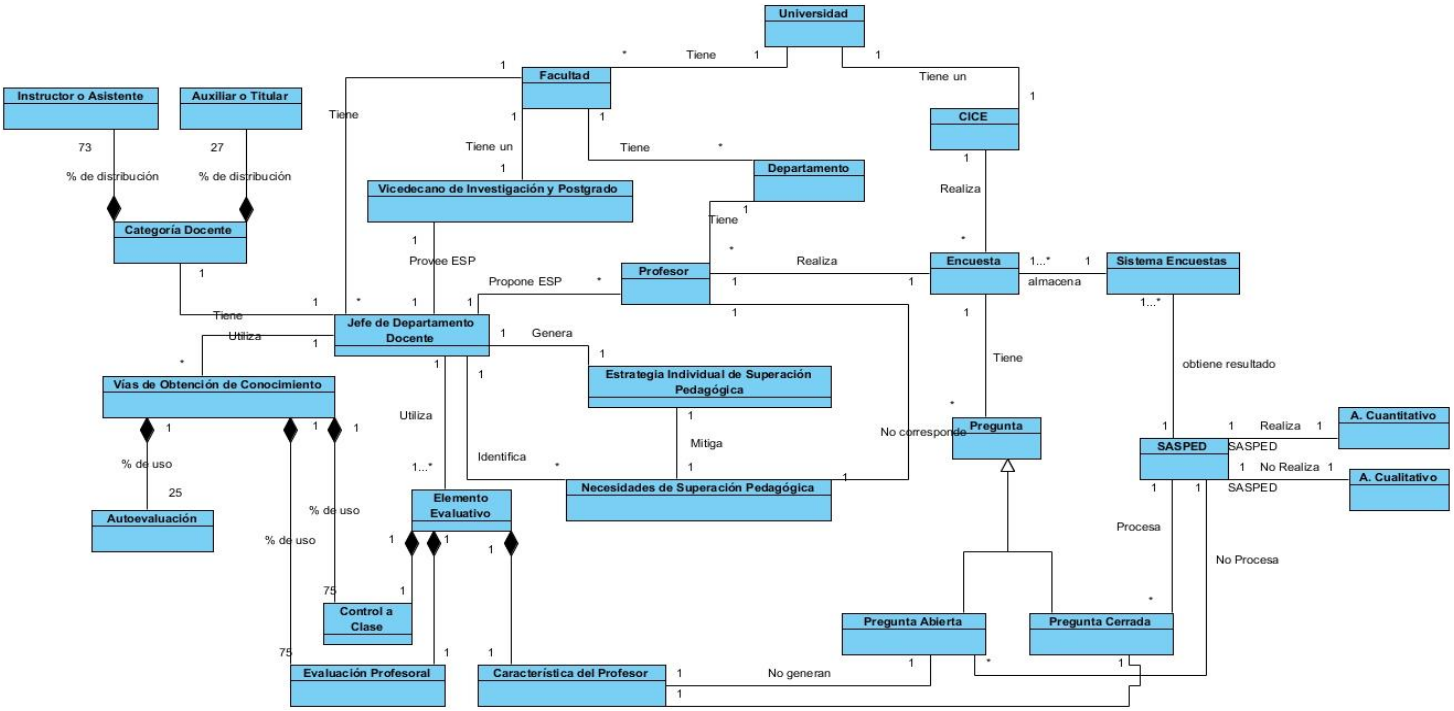


Figura 2: Modelo conceptual

Requisitos del sistema

Requisitos Funcionales

Para resolver la problemática establecida, se establecen los siguientes requisitos funcionales:

Tabla 2. Listado de requisitos funcionales

Número	Nombre	Descripción
1	Obtener listado de encuestas	Permite obtener un listado de las encuestas de superación pedagógica almacenada en el sistema Lime Survey
2	Obtener listado de encuestados	Permite obtener un listado de los encuestados almacenados en el sistema Lime Survey dada una encuesta determinada
3	Obtener resultados de encuestado	Permite obtener los resultados almacenados en el sistema Lime Survey de un encuestado dada una encuesta determinada

4	Validar petición de servicio	Permite validar la petición de un servicio determinado
5	Insertar Usuario	Permite registrar los datos de un usuario
6	Eliminar Usuario	Permite eliminar un usuario
7	Autenticar Usuario	Permite autenticarse para acceder al sistema
8	Verificar privilegios	Permite verificar qué privilegios tiene un usuario determinado
9	Insertar rol	Permite la inserción de un rol en el sistema de seguridad
10	Modificar rol	Permite la modificación de un rol en el sistema de seguridad
11	Mostrar rol	Permite mostrar un rol del sistema de seguridad
12	Eliminar Rol	Permite eliminar un rol del sistema de seguridad
13	Mostrar detalles de rol	Permite mostrar todos los usuarios que tienen un rol determinado

Requisitos no funcionales

Los requisitos no funcionales son una especificación de las características del sistema para garantizar su rendimiento. A continuación se presentan los requisitos no funcionales que deberán regir la solución de interoperabilidad.

- RNF1- Disponibilidad. El sistema debe estar disponible el 99% del tiempo en un curso escolar.
- RNF2- Escalabilidad. El sistema debe ser capaz de mantener su funcionamiento al insertar un nuevo subsistema.
- RNF3- Usabilidad. El sistema podrá ser utilizado por personas que posean conocimientos básicos de informática.
- RNF4- Tiempo de respuesta. El sistema tendrá un tiempo de respuesta promedio menor de 1 segundo por cada 100 peticiones concurrentes.
- RNF5- Confiabilidad. El sistema contará con un tratamiento de errores que permita que el funcionamiento continúe en caso de un fallo.
- RNF6- Flexibilidad. La solución deberá garantizar que se puedan realizar cambios cada capa de la arquitectura sin que estos afecten a las capas adyacentes.
- RNF7- Software. Se deberá contar con el servidor de aplicaciones de la plataforma WSO2 para manejar el uso de los servicios implementados.

- RNF8- Hardware. Se deberá contar con un procesador a 1.7 GHz y 2 Gb de memoria RAM como mínimo.
- RNF9- Compatibilidad. La solución deberá ser multiplataforma.
- RNF10- Eficiencia. Cuando un usuario permanece inactivo durante 20 minutos, se cierra la sesión automáticamente.
- RNF11- Seguridad. Los servicios web deben ser accedidos a través de un proceso de autenticación.

Definición de casos de uso

A continuación, se presentan los actores del sistema, así como los casos de uso que generan.

Definición de actores

Tabla 3. Actores del sistema

Actores	Descripción
Usuario	El operario principal del Sistema. Luego de autenticarse tiene acceso a un listado de funcionalidades que proveen los subsistemas de SASPED.
Administrador	Gestiona el sistema en términos de usuarios, sus roles, la implementación de la seguridad, así como los servicios web.
Subsistema consultor de Información de SASPED	Necesita información almacenada en una base de datos de encuestas externa a SASPED para procesar información relacionada a la gestión de la superación pedagógica.

Listado de casos de uso

Tabla 4. Descripción del caso de uso Autenticar usuario.

CASO DE USO:	Autenticar usuario
Resumen:	El caso de uso inicia cuando el actor accede a la interfaz de usuario principal de SASPED, el sistema proporciona campos de autenticación, el actor introduce los datos necesarios para ingresar en el sistema y luego accede, terminando así el caso de uso.
Complejidad:	Alta
Prioridad:	Alta
REFERENCIAS	
Actores:	Jefe de Departamento, Vicedecano de Información, Experto, Profesor
Requisitos:	**7**

Tabla 5. Descripción del caso de uso Insertar Rol.

CASO DE USO:	Insertar Rol
Resumen:	El caso de uso inicia cuando el actor procede a relacionar un rol determinado con un usuario de SASPED. Al realizarse la operación termina el caso de uso.
Complejidad:	Media
Prioridad:	Media
REFERENCIAS	
Actores:	Administrador
Requisitos:	**9**

Tabla 6. Descripción del caso de uso Modificar Rol.

CASO DE USO:	Modificar Rol
Resumen:	El caso de uso inicia cuando el actor procede a modificar un rol relacionado con un usuario de SASPED. Al realizarse la operación termina el caso de uso.
Complejidad:	Baja
Prioridad:	Media
REFERENCIAS	
Actores:	Administrador
Requisitos:	**10**

Tabla 7. Descripción del caso de uso Insertar Usuario.

CASO DE USO:	Insertar Usuario
Resumen:	El caso de uso inicia cuando el actor intenta insertar un nuevo usuario en el sistema. Este procede a insertar todos los datos requeridos del usuario. El caso de uso termina cuando el sistema inserta el nuevo usuario y muestra un mensaje de confirmación.
Complejidad:	Baja
Prioridad:	Baja
REFERENCIAS	
Actores:	Administrador
Requisitos:	**5**

Tabla 8. Descripción del caso de uso Eliminar Usuario.

CASO DE USO:	Eliminar Usuario
Resumen:	El caso de uso inicia cuando el actor intenta eliminar un usuario existente en el sistema. Este procede a seleccionar el usuario que desea eliminar. El caso de uso termina cuando el sistema elimina el usuario y muestra un mensaje de confirmación.
Complejidad:	Baja
Prioridad:	Baja
REFERENCIAS	
Actores:	Administrador
Requisitos:	**6**

Tabla 9. Descripción del caso de uso Verificar Privilegios de Usuario.

CASO DE USO:	Verificar Privilegios de Usuario
Resumen:	El caso de uso inicia cuando el actor realiza un chequeo de los privilegios de un usuario determinado. El caso de uso termina cuando el sistema devuelve dichos privilegios.
Complejidad:	Baja
Prioridad:	Baja
REFERENCIAS	
Actores:	Administrador
Requisitos:	**8**

Tabla 10. Descripción del caso de uso Mostrar Detalles de Rol.

CASO DE USO:	Mostrar Detalles de Rol
Resumen:	El caso de uso inicia cuando el actor realiza una búsqueda sobre todos los usuarios que tienen un rol determinado. El caso de uso termina cuando el sistema devuelve dichos usuarios.
Complejidad:	Baja
Prioridad:	Baja
REFERENCIAS	
Actores:	Administrador
Requisitos:	**13**

Tabla 11. Descripción del caso de uso Validar Petición de Servicio.

CASO DE USO:	Validar Petición de Servicio
Resumen:	El caso de uso inicia cuando el actor intenta consumir algún servicio. Presenta las credenciales al sistema y este comprueba las mismas. En caso de ser correctas el sistema le da acceso al servicio requerido, terminando así el caso de uso.
Complejidad:	Media
Prioridad:	Alta
REFERENCIAS	
Actores:	Subsistema consultor de Información de SASPED
Requisitos:	**4**

Tabla 12. Descripción del caso de uso Obtener Encuestas de Superación Pedagógica.

CASO DE USO:	Obtener Encuestas de Superación Pedagógica
Resumen:	El caso de uso inicia cuando el actor realiza una petición de todas las encuestas de superación pedagógica. El sistema realiza una consulta a la base de datos de encuestas para extraer estas y luego devolverlas al actor, terminando así el caso de uso.
Complejidad:	Media
Prioridad:	Alta
REFERENCIAS	
Actores:	Subsistema consultor de Información de SASPED
Requisitos:	**1**

Tabla 13. Descripción del caso de uso Obtener Encuestados.

CASO DE USO:	Obtener Encuestados
Resumen:	El caso de uso inicia cuando el actor realiza una petición de todos los encuestados que respondieron una determinada encuesta de superación pedagógica. El sistema realiza una consulta a la base de datos de encuestas para dicha información y luego devolverla al actor, terminando así el caso de uso.
Complejidad:	Alta
Prioridad:	Alta
REFERENCIAS	
Actores:	Subsistema consultor de Información de SASPED
Requisitos:	**2**

Tabla 14. Descripción del caso de uso Obtener Resultados de Encuestados.

CASO DE USO:	Obtener Resultados de Encuestados
Resumen:	El caso de uso inicia cuando el actor realiza una petición de las respuestas de un determinado encuestado a una encuesta de superación pedagógica. El sistema realiza una consulta a la base de datos de encuestas para encontrar las respuestas y luego devolverlas al actor, terminando así el caso de uso.
Complejidad:	Alta
Prioridad:	Alta
REFERENCIAS	
Actores:	Subsistema consultor de Información de SASPED
Requisitos:	**3**

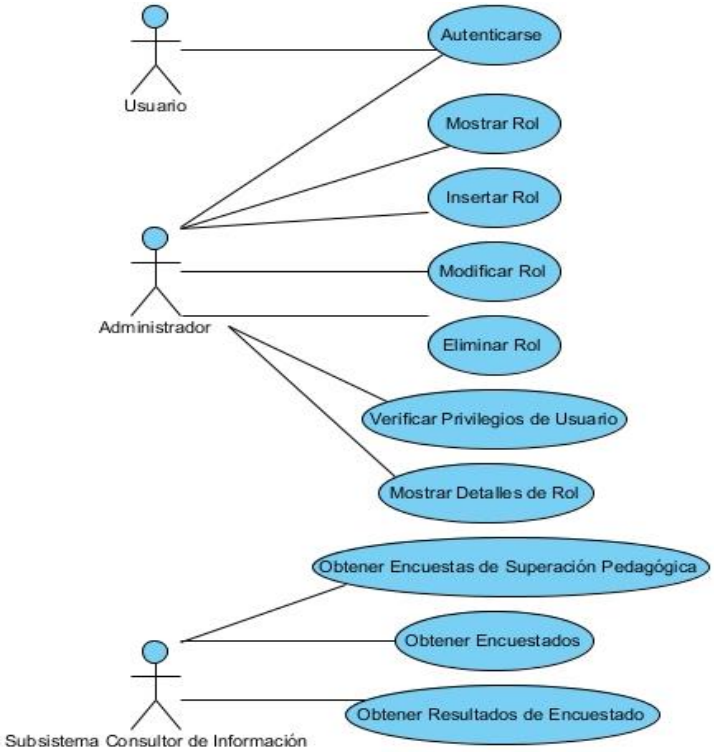


Figura 3: Diagrama de casos de uso del sistema.

Arquitectura de SASPED

Dada la necesidad de desarrollar una solución que intercomunique los subsistemas de SASPED, de forma que garantice la transmisión segura y desacoplada de la información entre dichos subsistemas, así como que permita a los usuarios percibir la interacción con los subsistemas como un todo para que puedan resolver los problemas de la ESP, se propone utilizar una arquitectura **Cliente Servidor**.

Esta arquitectura estará compuesta por un grupo importante de componentes que, en su accionar como conjunto, ofrecerán una respuesta eficiente a cada una de las necesidades que presente dicha ESP en cada una de sus etapas. En el lado del cliente, y en consecuencia con lo que en él se realiza, se encontrará la interfaz gráfica del sistema. A través de esta, los funcionarios y encargados del proceso de superación podrán gestionar e interactuar con la misma y realizar las acciones necesarias que contribuyan positivamente en el proceso de toma de decisiones. Dicho componente permitirá acceder a las funcionalidades que brindan los subsistemas a través de peticiones por el protocolo HTTP utilizando cualquier aplicación o dispositivo que soporte dicho protocolo. En general, este componente será el encargado de mostrar toda la información que pueda ser generada durante el proceso de superación pedagógica.

En el lado del servidor se encontrarán los subsistemas que apoyan el proceso de Superación Pedagógica. El Subsistema de Gestión de Información, maneja la información referente a la superación pedagógica del claustro y persiste su información en el esquema que le corresponde en la base de datos de SASPED. El Subsistema Experto realiza propuestas de estrategia de superación pedagógica, y que al ser un software de inteligencia artificial, almacena su información en una base de casos, la cual se encontrará también en un esquema de la base de datos de SASPED. El Subsistema de Análisis Semántico de Texto, analiza las respuestas a preguntas abiertas generadas en las encuestas de superación pedagógica y almacena datos en una ontología que se encuentra físicamente en un archivo.

La transmisión de datos entre estos tres subsistemas se realizará mediante la utilización de las conversaciones del framework JBoss Seam. Dicha transmisión se realizará de la siguiente forma: el Subsistema de Gestión de Información establece un contexto conversacional con el Subsistema Experto mediante la cual le envía una lista de características de un profesor a través del protocolo HTTP; para el envío de las respuestas de preguntas abiertas al Subsistema de Análisis Semántico de Texto, el Subsistema de Gestión de Información también crea un contexto conversacional, utilizando HTTP como protocolo de transporte.

También en el lado del servidor el sistema contará con un repositorio de servicios web basado en el estándar UDDI. En dicho repositorio serán registrados cada uno de los servicios web y sus interfaces correspondientes, los cuales serán accedidos por el subsistema consultor de información que necesite obtener determinadas encuestas de superación pedagógica, encuestados asociados a éstas, así como sus respuestas a las encuestas. En la base de datos de SASPED se encontrará un esquema que

corresponderá a la gestión de la seguridad del sistema, en el cual se almacenará la información asociada a los usuarios que se registren en el sistema y sus respectivos roles.

En la siguiente figura se muestra la representación gráfica de la arquitectura de los componentes que conforman a SASPED, siguiendo el patrón arquitectónico **MVC**.

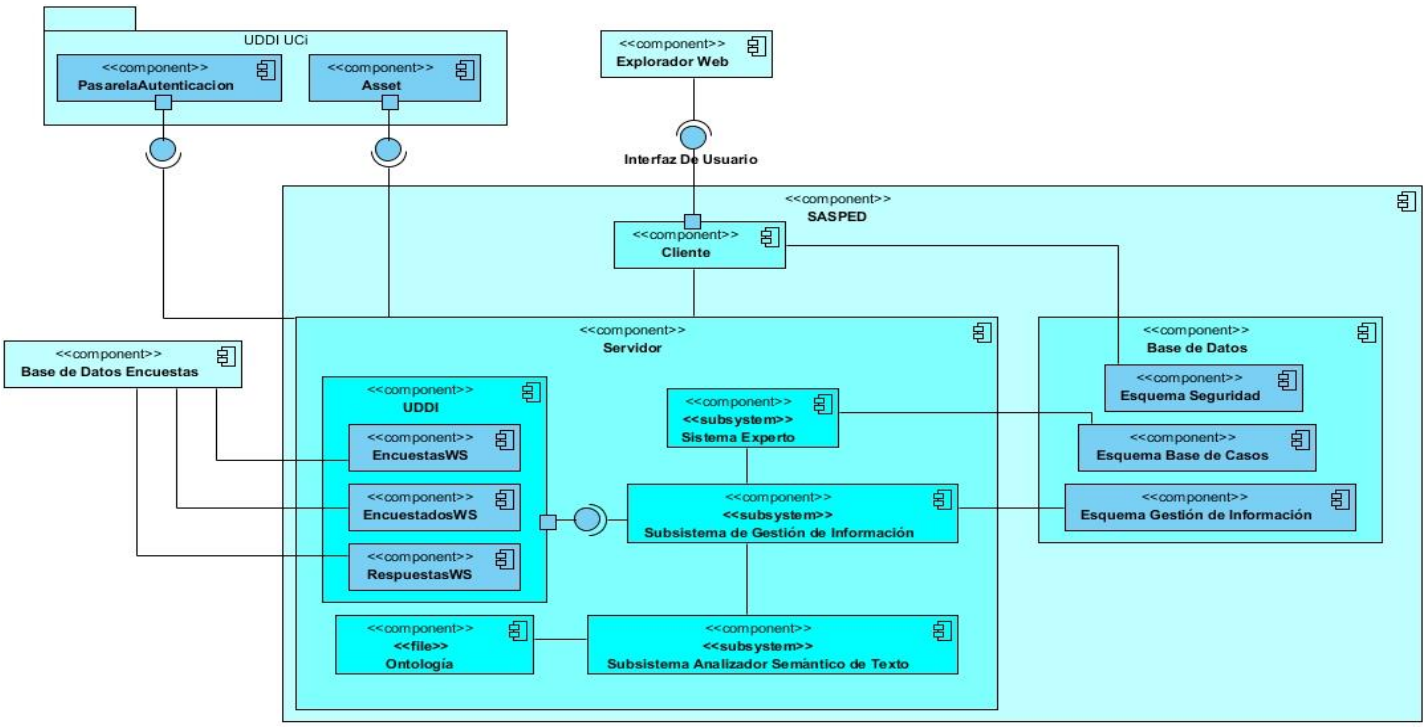


Figura 4: Arquitectura de SASPED.

Diagramas de clases del diseño

A continuación se representan los diagramas de clases correspondientes a la presente investigación.

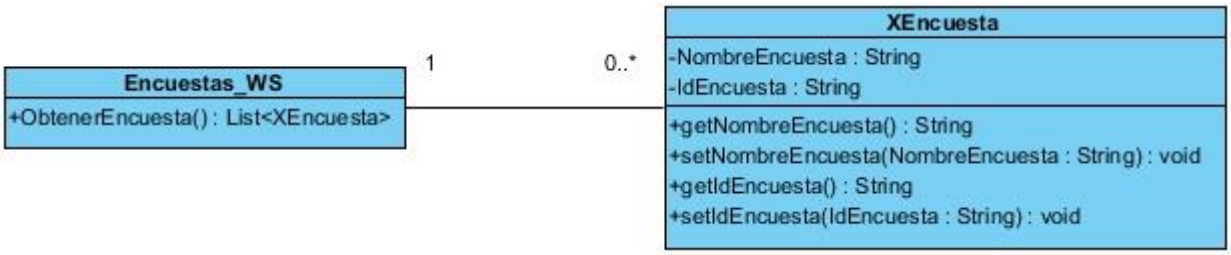


Figura 5: Diagrama de clases del servicio web Encuestas_WS.

Tabla 15. Descripción de las clases del servicio Encuestas_WS

Clase	Descripción
XEncuesta	Esta clase es la que contiene todos los atributos que pertenecen a una encuesta.
Encuestas_WS	Esta clase es la encargada de realizar la conexión con la base de datos y obtener todos los datos referentes a la encuesta.

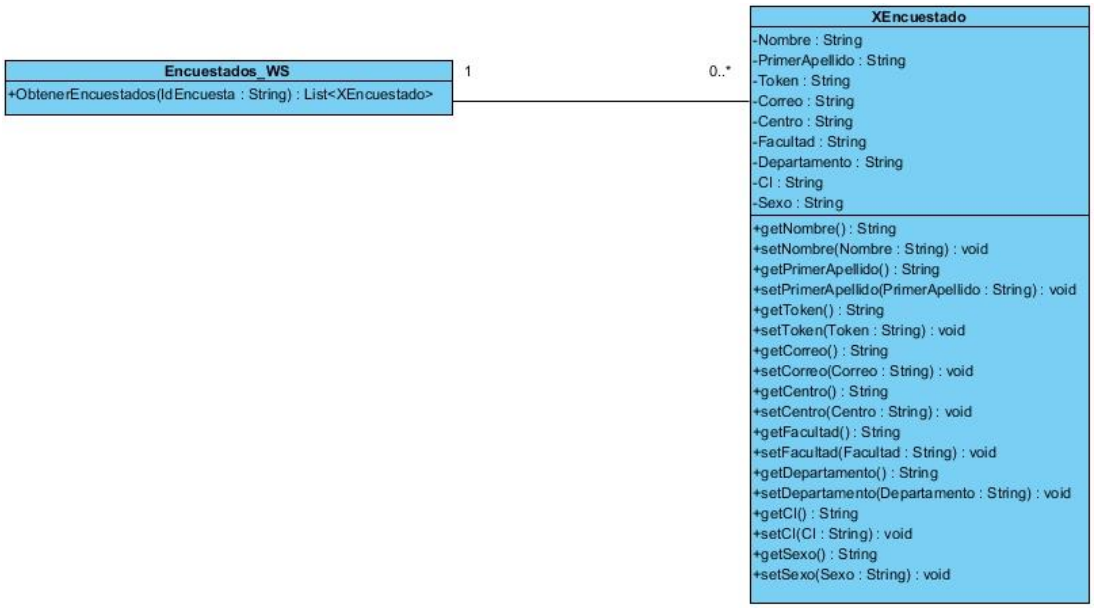


Figura 6: Diagrama de clases del servicio web Encuestados_WS.

Tabla 16. Descripción de las clases del servicio web Encuestados_WS

Clase	Descripción
XEncuestado	Esta clase es la que contiene todos los atributos que pertenecen a un encuestado.
Encuestas_WS	Esta clase es la encargada de realizar la conexión con la base de datos y obtener todos los datos referentes a un encuestado.

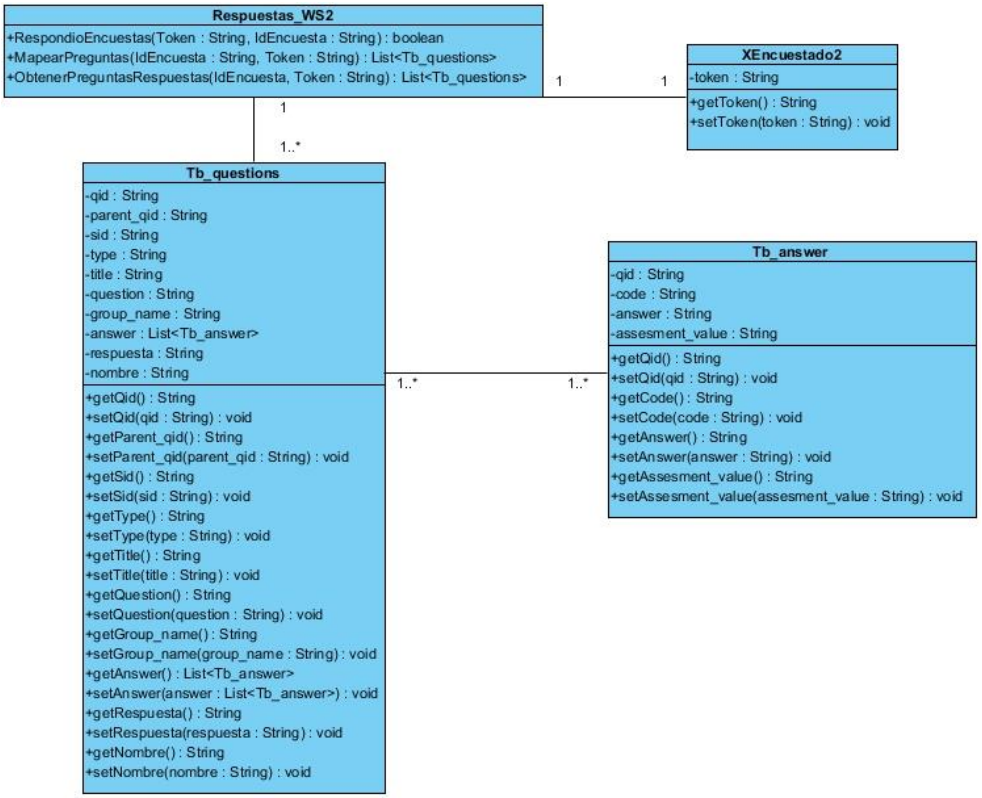


Figura 7: Diagrama de clases del servicio web Respuestas_WS.

Tabla 17. Descripción de las clases del servicio web Respuestas_WS

Clase	Descripción
Tb_questions	Esta clase contiene todos los atributos que pertenecen a una pregunta. Además contiene una instancia de la clase Tb_answer.
Tb_answer	Esta clase contiene todos los atributos que pertenecen a una respuesta.
XEncuestado	Esta clase contiene todos los atributos que pertenecen a un encuestado.
Respuestas_WS	Esta clase es la encargada de realizar la conexión con la base de datos, determinar si la persona ha respondido la encuesta, mapear todas las preguntas y obtener todos los datos referentes a las respuestas de un encuestado

Patrones GRASP

A continuación se describen la utilización de los patrones de asignación de responsabilidades a las clases del código.

- **Creador**: Este patrón ayuda a identificar quién debe ser el responsable de la creación de objetos, teniendo en cuenta que esta debe poseer la información necesaria para la creación de estas instancias. Una ventaja de la utilización de este patrón es el bajo acoplamiento si se diseña correctamente, lo cual permite la facilidad para realizar cambios, mayor claridad y reutilización. El uso de este patrón en los servicios a implementar se evidencia en las clases Encuestas_WS, Encuestas_WS y Respuestas_WS.
- **Controlador**: Este patrón se encarga de que una clase actúe como intermediaria entre una determinada interfaz y el algoritmo que la implementa. Este patrón sugiere que la lógica de negocios debe estar separada de la capa de presentación, esto es para aumentar la reutilización de código y a la vez tener un mayor control. El uso de este patrón en la gestión de la seguridad a implementar se evidencia en las clases CCRoles, CCUsuarioUCI y CCPerfil.
- **Experto**: Se basa en asignar una responsabilidad al experto en información, se refiere a la clase que cuenta con la información o atributos necesarios para cumplir con la responsabilidad o responsabilidades para la que fue implementada. Una clase se considera experta si contiene todos los datos necesarios para realizar la labor que tiene encomendada. Con este patrón se logra la encapsulación y una alta cohesión. El uso de este patrón en la gestión de la seguridad a implementar se evidencia en las clases CCRoles, CCUsuarioUCI y CCPerfil.
- **Bajo Acoplamiento**: Este patrón permite mantener las clases lo menos ligadas o dependientes posible. De tal forma en caso de producirse una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de las clases. La solución es asignar las responsabilidades de forma tal, que las clases se comuniquen con el menor número de entidades que se pueda, sin comprometer su funcionalidad. El uso de este patrón en la gestión de la seguridad a implementar se evidencia en las clases CLRoles, CMRoles y CLPersonaUCI. En los servicios web se evidencia en las clases XEncuesta, XEncuestado y Tb_answer.
- **Alta Cohesión**: Este patrón se refiere a cuán relacionadas están las clases. El mismo indica que cada elemento del diseño realiza una labor única dentro del sistema, lo cual expresa que la información que almacena una clase debe de ser coherente y en la mayor medida de lo posible relacionada con la clase, esto permite liberar de responsabilidades a algunas clases, pero exceptúa el bajo acoplamiento. El uso de este patrón en la gestión de la seguridad a implementar se evidencia en las clases CCRoles,

CCUsuarioUCI y CCPerfil. En los servicios web se evidencia en las clases Encuestas_WS, Encuestas_WS y Respuestas_WS.

Conclusiones parciales

- El estado actual de la concepción de la interoperabilidad entre los subsistemas de SASPED es poco favorable, dado que esta se caracteriza por deficiencias en el rendimiento, la compatibilidad entre los subsistemas y la indefinición de autoridades y responsabilidades.
- El diseño de una arquitectura cliente-servidor que agrupe los subsistemas de SASPED, así como la UDDI donde se almacenan los servicios web de dicho sistema y su base de datos, haciendo uso del patrón MVC garantizará la guía para automatizar los procesos definidos en la ESP.

CAPÍTULO NO. 3: IMPLEMENTACIÓN Y VALIDACIÓN DE LA SOLUCIÓN DE INTEROPERABILIDAD DE LOS SUBSISTEMAS DE SASPED

En el presente capítulo se describe la implementación de la solución perteneciente a la fase de elaboración del proceso de desarrollo de software, dados los artefactos generados previamente. Luego de terminar la construcción de la solución, se procedió a realizar pruebas de validación utilizando métodos adecuados para los software del campo de acción de soluciones de interoperabilidad para sistemas web.

Definición de los contratos de los Servicios Web implementados.

Todos los servicios web implementados bajo el protocolo SOAP generan un documento en formato WSDL. WSDL son las siglas de Web Services Description Language, un formato XML que se utiliza para describir los servicios web. Este formato describe la interfaz pública de los servicios y su forma de comunicación, es decir, los requisitos y formatos de los mensajes necesarios para interactuar con los servicios listados en su catálogo. En la estructura de un documento WSDL figuran las secciones de tipos de datos, mensajes, tipos de puerto, bindings y servicios. En estas secciones se define principalmente los tipos de datos usados en los mensajes, los elementos que conforman un mensaje, las operaciones permitidas y los mensajes intercambiados, así como también los protocolos de comunicación usados y el conjunto de puertos y dirección de cada servicio respectivamente. Otra de las características que poseen estos documentos es que emplean XML como lenguaje de notación, lo que los convierte en documentos neutrales al lenguaje de programación y accesibles desde diversas plataformas. Para el desarrollo de la presente investigación fue utilizado wsdl en su versión 1.1 para la implementación de los servicios, la cual constituye la segunda versión recomendada por el W3C para la realización de este tipo de contratos.

Con el objetivo de lograr la estandarización de estos documentos y la familiarización de los desarrolladores de los subsistemas que los utilizaran, y con el apollo de las herramientas que brinda el Aplication Server de la plataforma WSO2, fueron definidas a continuación un grupo de pautas para la descripción de los servicios mediante WSDL:

- Los nombres de los WSDL deben seguir el siguiente formato: *Servicio_WS.wsdl* ejemplo: ***Encuestas_WS.wsdl.***
- El atributo targetNamespace debe tener como valor un elemento que se corresponda con su verdadero espacio de nombres. Para estos casos se seguirá el siguiente formato *http://servicio*. Ejemplo: ***http://encuestados***.

- El uso del lenguaje de programación Java condiciona algunos de los atributos que se representan en el WSDL del servicio. En caso de que algún método del servicio reciba o devuelva una lista o un arreglo, estos serán representados en el WSDL como un tipo de dato complejo. Si la lista o el arreglo posee una estructura bien definida la etiqueta *complexType* contendrá el tipo de datos de la misma. Ejemplo: ***XRespuesta***.
- La estructura interna de estas listas o arreglos será definida por los elementos que los conforman, estos se encontrarán ubicados dentro de la etiqueta *sequence* en una etiqueta *element* y cada uno debe tener un nombre identificativo y un tipo de dato. Ejemplo: ***group_name string***.
- Cada servicio posee dos documentos WSDL en la plataforma, uno para la version 1.1 de WSDL y otro para la version 2.0. Estos a su vez cuentan con un solo puerto. Los puertos deben llamarse *Servicio_WSPortType*. Ejemplo: ***Encuestados_WSPortType***.
- Cada documento WSDL contará además con tres binding que se llamarán: *Servicio_WSSoap11Binding*, *Servicio_WSSoap12Binding* y *Servicio_WSHttpBinding* respectivamente. Cada uno de estos se enlazará con el puerto creado. Además, cada binding tendrá un estilo Document y utilizara HTTP como protocolo de transporte. Ejemplo: ***Encuestados_WSSoap11Binding***.
- Por último, se debe crear un servicio que tendrá como nombre *Servicio_WS*, el mismo tendrá tres puertos los cuales serán: *Servicio_WSHttpsSoap11Endpoint*, *Servicio_WSHttpsSoap12Endpoint* y *Servicio_WSHttpsSoapEndpoint*. Cada puerto tendrá un binding y un address con respecto a su version.

Es de vital importancia la implementación de estos servicios para el funcionamiento de SASPED, ya que estos brindan la posibilidad al subsistema de gestión de la información y, a travez de este a todos los demás, de obtener la información relacionada con las encuestas de superación pedagógica. A continuación se describen estos servicios, haciéndose énfasis en los parámetros de entrada y en las estructuras de salida básicamente.

Definición de servicios web implementados

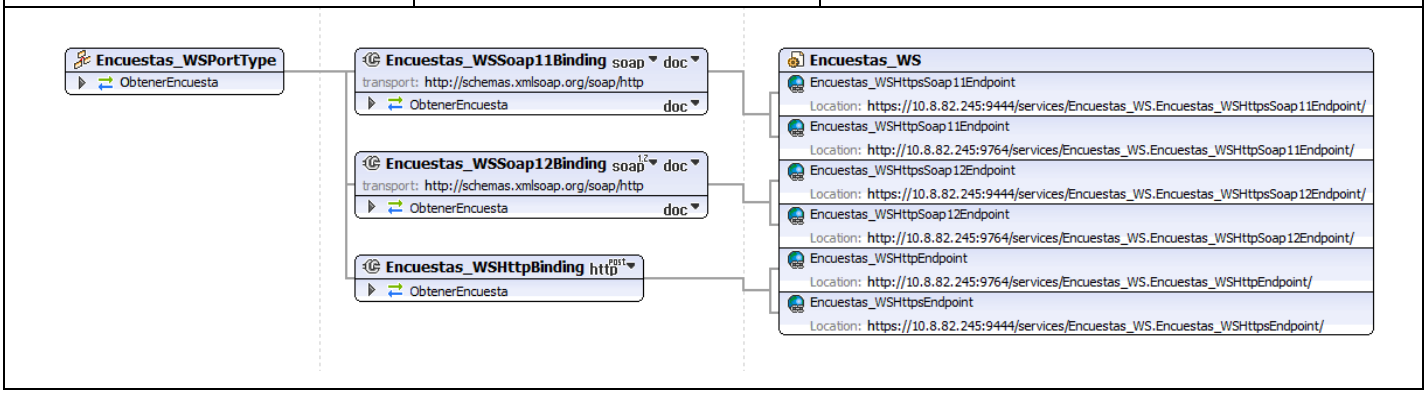
Obtener Listado de Encuestas de Superación Pedagógica.

Mediante este servicio, el Subsistema de Gestión de la Información podrá obtener el nombre y el identificador de todas las encuestas de superación pedagógica realizadas hasta el momento, los cuales serán utilizados posteriormente para obtener los participantes en una encuesta determinada. Este servicio

no recibe parámetros de entrada y devuelve un arreglo con los parámetros de salida que se definen en la siguiente tabla.

Tabla 18. Descripción de parámetros de entrada y salida del servicio Encuestas_WS.

Nombre del Servicio: **Encuestas_WS**		
Parámetros de Entrada		
Tipo		**Descripción**
-		-
Parámetros de Salida		
Tipo		**Descripción**
List<XEncuesta>	string	IdEncuesta
	string	NombreEncuesta

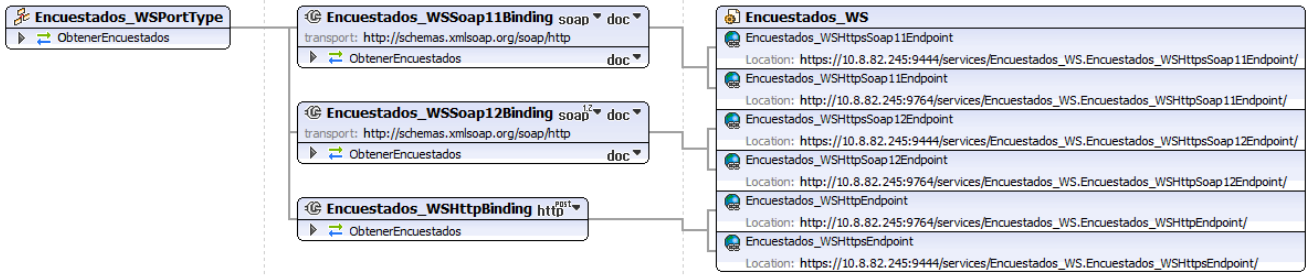


Obtener listado de encuestados

Este servicio garantiza la obtención de la información relacionada con los encuestados que han realizado una encuesta determinada, dado el identificador de la misma. Permite obtener el nombre de los encuestados, el primer apellido, la dirección de correo electrónico, el centro en que labora, la facultad a la que pertenece, el departamento, el carné de identidad, el sexo y un token que funcionará a manera de identificador del encuestado que posteriormente se utilizara para la obtención de las respuestas del mismo en una encuesta determinada. Este servicio recibe como parámetros de entrada el identificador de la encuesta de la que se desea obtener los encuestados y devuelve un arreglo con los parámetros de salida que se definen en la siguiente tabla.

Tabla 19. Descripción de parámetros de entrada y salida del servicio Encuestados_WS.

Nombre del Servicio: **Encuestados_WS**		
Parámetros de Entrada		
Tipo		**Descripción**
string		IdEncuesta
Parámetros de Salida		
Tipo		**Descripción**
List<XEncuestado>	string	Nombre
	string	PrimerApellido
	string	Token
	string	Correo
	string	Centro
	string	Facultad
	string	Departamento
	string	CI
	string	Sexo

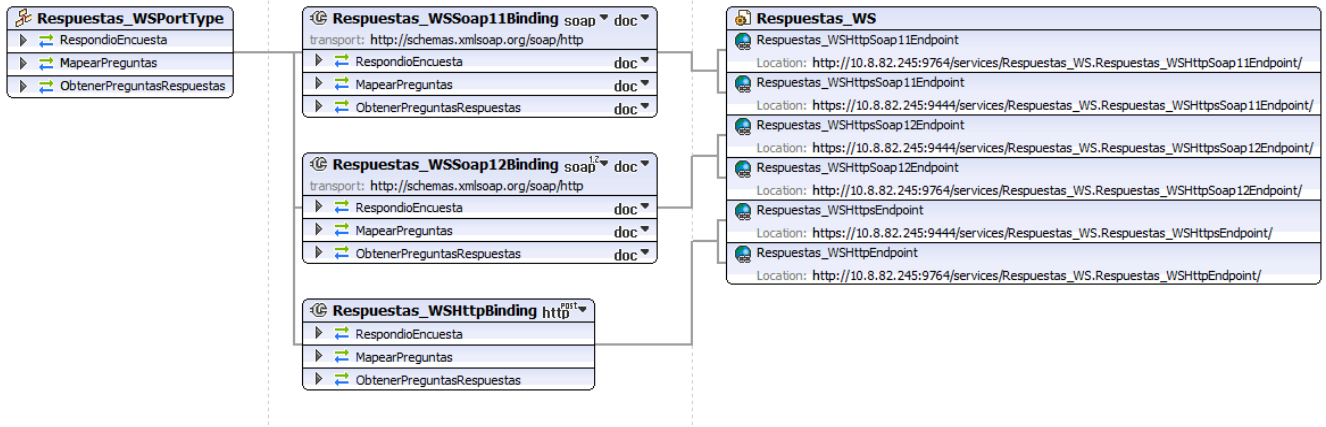


Obtener resultado de encuestados

A través de este servicio se obtendrán todas las secciones, preguntas y respuestas ofrecidas por un encuestado en una determinada encuesta de superación pedagógica, a partir del identificador de la encuesta y el token del encuestado. Este servicio recibe como parámetros de entrada el identificador de la encuesta y el token del encuestado del que se desea obtener sus resultados y devuelve una lista de tipo Respuesta con los parámetros que se definen en la siguiente tabla.

Tabla 20. Descripción de parámetros de entrada y salida del servicio Respuestas_WS.

Nombre del Servicio: **Respuestas_WS**		
Parámetros de Entrada		
Tipo		**Descripción**
String		IdEncuesta
String		Token
Parámetros de Salida		
Tipo		**Descripción**
Lista<XRespuesta>	String	Session (Sesión de la encuesta)
	String	Question
	String	Respuesta



Gestión de seguridad para los servicios definidos

Con el objetivo de garantizar la integridad y la confidencialidad de toda la información que se maneja a través de los servicios anteriormente descritos, se utiliza el primer nivel de seguridad que ofrece la herramienta WSO2 Application Server, el cual provee de un proceso de autenticación para el acceso a cada servicio seleccionado. Este método permite asegurar el mensaje en la capa de transporte, un token de seguridad y un nombre de usuario se utilizan para la autenticación. Primeramente, el cliente envía el token y el nombre del usuario dentro del encabezado del mensaje WSDL donde se realiza la petición, como se muestra en la ***Figura 8***, luego se validan las entradas en el almacén de usuarios del servicio y si ambos parámetros son correctos se devuelve un mensaje XML con las respuestas.

```
− <sp:SignedSupportingTokens>
  − <wsp:Policy>
      <sp:UsernameToken sp:IncludeToken="http://schemas.xmlsoap.org/ws/2005/07/securitypolicy/IncludeToken/AlwaysToRecipient"/>
    </wsp:Policy>
  </sp:SignedSupportingTokens>
```

Figura 8: Ejemplo de encabezado WSDL del servicio Encuestados_WS con Username-Token

Gestión de control de acceso

La gestión del acceso y la seguridad de SASPED, estará comprendida por los proceso de autenticación, mediante el cual se determinará la legitimidad de la identidad del usuario, así como la autorización, la cual regirá las funcionalidades a las cuales tendrá acceso el usuario. Se decidió utilizar el modelo de seguridad RBAC (Role Based Access Control), mediante el cual, según (Ferraiolo, Kuhn, & Chandramouli, 2007), se definen los usuarios, que son las personas que interactúan con el sistema, los sujetos, que será el proceso que actúa como mediador entre el usuario y el sistema, los objetos, que son los recursos que se encuentran accesibles en el sistema, ya sean archivos, periféricos, impresoras, bases de datos y entidades de mayor granularidad como campos en una base de datos, las operaciones, las cuales se definen como procesos activos invocados por el usuario, así como los permisos o privilegios, que son autorizaciones para realizar una acción en el sistema.

Debido a que SASPED se enmarca en un contexto de superación pedagógica, su utilización solo será efectuada por parte de personal docente de la Universidad de las Ciencias Informáticas. En tal sentido, para verificar la legitimidad de los datos del usuario, el sistema utiliza el servicio de autenticación que provee la UDDI de la UCI, luego procede a buscar en la base de datos si el usuario existe y tiene un rol asociado. En caso contrario, se procede a verificar que el usuario pertenece al dominio de la aplicación y automáticamente se crea y se le asigna un rol a partir de su cargo.

Para la interacción con los diferentes procesos automatizados de SASPED, se cuenta con un grupo de roles, que están definidos en la Propuesta de Estrategia de Superación Pedagógica, lo cuales son el Profesor, Jefe de Departamento Docente, Jefe del Grupo de Trabajo de Formación Pedagógica, Vicedecano de Formación, Vicedecano de Investigación y Postgrado, Decano, Director del CICE, Director del Departamento Docente Metodológico, Vicerrector de Formación, Rector y el Administrador del Sistema.

Modelo de datos

La base de datos de la presente solución de interoperabilidad se encuentra dividida por esquemas, pues cada subsistema necesita procesar la información relacionada con sus procesos por separado. Como se planteó anteriormente al definir la arquitectura de SASPED, existen tres esquemas, los cuales son la base de datos del Subsistema de Gestión de Información, la base de casos del Subsistema Experto y el esquema de seguridad. Mediante la interacción de la propuesta de solución con dicho esquema de seguridad se garantizará el control de acceso de SASPED. A continuación se muestra el diagrama del modelo de datos.

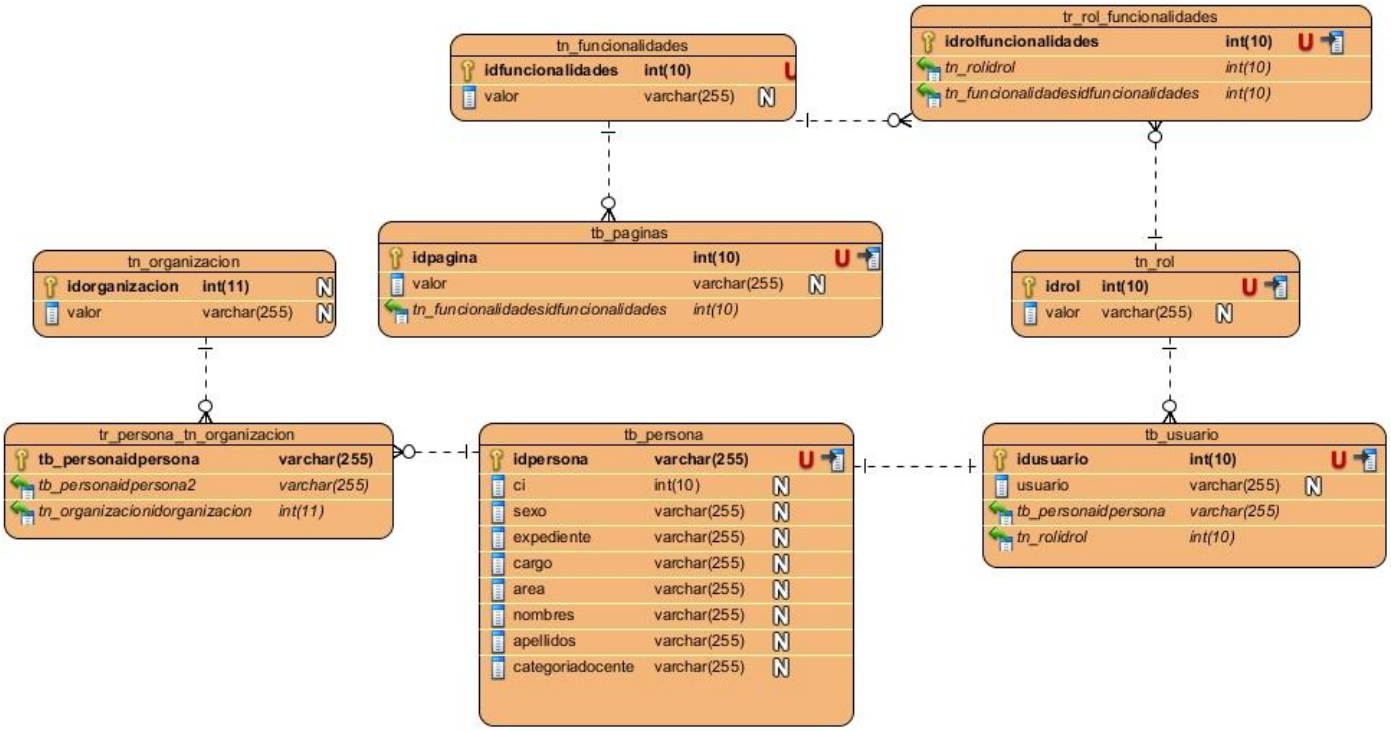


Figura 9: *Modelo de Datos del Esquema Seguridad*

Estilo de codificación

Los estilos o estándares de codificación son parte de las buenas prácticas que facilitan la comprensión del código por otras personas. Para facilitar la posterior comprensión de la lógica de programación se tomaron medidas como la indentación de código, la notación camella para las variables y métodos, y la descripción de los métodos y clases mediante comentarios.

Para la indentación de código, se utilizaron 4 espacios en blanco. También se evitaron líneas de más de 80 caracteres. Para romper una expresión que no entre en una línea, se decidió hacerlo después de las comas, antes de un operador y alinear la línea nueva con el comienzo de la expresión al mismo nivel que la línea anterior.

Con respecto a los comentarios, se consideró utilización de comentarios de bloque para descripciones de métodos, estructuras de datos y algoritmos, que se indentan al mismo nivel del código escrito, con el objetivo de facilitar información adicional que no es legible en el código mismo y solo para mostrar datos relevantes para la lectura y el entendimiento del programa.

Tratamiento de errores

Los errores del sistema son manejados dentro de la propia aplicación, lo que permite que el usuario no pierda de vista la interfaz gráfica del sistema en ningún momento. Por otra parte, los errores de entrada de datos de los usuarios o validaciones típicas del negocio son resueltos a través de mecanismos que explican y guían al mismo sobre los pasos que originaron el error. Algunos de estos mecanismos son los conocidos como excepciones.

Una excepción es un evento que ocurre durante la ejecución del programa, que interrumpe el flujo normal de las sentencias. El manejo de excepciones permite al usuario crear aplicaciones tolerantes a fallas y robustas para controlar estas excepciones y que pueda seguir ejecutando el programa sin verse afectado por el problema. Para el tratamiento de excepciones se utilizaron las instrucciones Try y Catch, que contienen el código que puede presentar errores. A continuación se muestran dos casos en que se utilizan dichas instrucciones.

Para capturar errores en la conexión a la base de datos, se utiliza el bloque de instrucciones que se muestra en la ***Figura 10***, en el cual primero se trata de realizar la conexión a la base de datos indicada y en caso de no existir se muestra un error.

```
try {
    Class.forName("com.mysql.jdbc.Driver");
    conexion = DriverManager.getConnection(
            "jdbc:mysql://localhost/encuestas", "root", "");
} catch (SQLException e) {
    e.printStackTrace();
    // TODO: handle exception
} catch (ClassNotFoundException ex) {
    ex.printStackTrace();
    // TODO: handle exception
}
```

Figura 10. Captura de errores al realizar la conexión a la base de datos

Otro ejemplo se muestra en la ***Figura 11***, al tratar de identificar si la persona ha respondido la encuesta. En este caso se utilizan las siguientes instrucciones.

```
try {
    PreparedStatement statement = conexion.prepareStatement(SQL);
    ResultSet result = statement.executeQuery();
    List<XEncuestado> list = new ArrayList<XEncuestado>();
    while (result.next()) {
        XEncuestado encuestado = new XEncuestado();
        encuestado.setToken(result.getString("token"));
        list.add(encuestado);
    }
    return list.size() > 0;
} catch (SQLException e) {
    // TODO: handle exception
    e.printStackTrace();
}
```

Figura 11. Captura de errores al llenar la lista de encuestados.

Pruebas

Para (Sommerville, 2006), el proceso de pruebas de software se realiza con el objetivo de demostrarle al desarrollador y al cliente que la aplicación cumple con los requisitos, como también para descubrir fallas o defectos donde el comportamiento del software es incorrecto, indeseable o no cumple su especificación. Dada la necesidad de comprobar el desempeño de la solución propuesta, se evaluará su funcionamiento mediante el uso de casos de pruebas que reflejen su manejo esperado, asi como pruebas de defectos a traves de casos diseñados para descubrir los posibles problemas. Tambien para comprobar si el sistema

alcanza el éxito en su funcionamiento o para descubrir los problemas que causan el malfuncionamiento del mismo en caso contrario. En la presente investigación se hace uso de las pruebas unitarias y de caja negra.

Pruebas unitarias

Con el objetivo de comprobar si cada uno de los módulos implementados está correctamente codificado, así como para probar su desempeño separado del resto del sistema, se realizan pruebas unitarias.

A continuación se muestran un conjunto de resultados pertenecientes a la realización de este tipo de pruebas a los servicios implementados para la propuesta de solución de interoperabilidad, haciendo énfasis, en el análisis de los niveles de eficiencia y en los tiempos de respuesta de los mismos. Para el desarrollo de los casos de prueba se consideraron diferentes cantidades de peticiones concurrentes, debido a que los servicios implementados, se encontraran disponibles de para todos los subsistemas de SASPED que necesiten utilizarlos.

Además, debido a la importancia del proceso en el que se encuentran enmarcados, es necesario tener en cuenta la capacidad de respuesta de esta solución bajo distintas condiciones de uso. Se debe tener en cuenta inicialmente que la generación de los contratos de servicios o documentos WSDL de estos servicios fueron generados con la suite de soluciones para el trabajo con servicios web WSO2, en específico con la herramienta Application Server versión 5.2.0. Esta herramienta, además de generar el WSDL en las versiones 1.1 y 2.0, provee la posibilidad de analizar cada servicio por separado con el objetivo de determinar si los mismos están correctamente implementados y bien formados. También permite hacer peticiones SOAP al servidor para analizar los resultados, almacenando las estadísticas para cada una de estas, como son: cantidad de peticiones realizadas al servidor, cantidad de respuestas enviadas desde el servidor, el tiempo máximo y mínimo de cada respuesta, cantidad de fallos ocurridos y el promedio del tiempo de respuesta.

Para corroborar todos estos resultados también se hará uso de la herramienta SOAP UI en su versión 4.5.1, la que permite elaborar casos de pruebas de carga y rendimiento a los servicios. Esta herramienta permite simular a varios clientes consumiendo el mismo servicio a la misma vez, añadiendo más hilos y por consiguiente más consumo de CPU.

Para el desarrollo de las pruebas se utilizó una computadora personal DELL, con 2GB de memoria RAM y un procesador Core i3 a 1.70 GHz de frecuencia, donde fue desplegada la plataforma WSO2 y montado los servicios, incluyendo su base de datos.

Con el objetivo de realizar un correcto análisis a los resultados obtenidos en las pruebas realizadas se tendrán en cuenta las aserciones que permitan validar contra el XSD definido dentro del servicio y también se validará que el mensaje de respuesta sea un mensaje SOAP y no un error o excepción de tipo SOAPFault. Para estas pruebas se realizaran 10 iteraciones en las cuales se medirán variables como el tiempo mínimo de respuesta (TMín), tiempo máximo de respuesta (TMáx) y tiempo promedio (TP), así como la cantidad de errores (CE) y la cantidad de bytes (CB) transferidos para cada iteración.

El primer caso se efectuará con 10 hilos que simularán la concurrencia de igual número de usuarios (Nu), con una demora de una unidad de tiempo entre peticiones y 100 peticiones por cada hilo. Para cada iteración se aumentará en 10 el número usuarios que concurren a los servicios. Todos los tiempos de respuesta serán analizados en milisegundos (ms).

Tabla 21. Resultado de las pruebas de rendimiento al servicio Encuestados_WS.

	Servicio Encuestados_WS				
CU	**TMín**	**TMáx**	**TP**	**CE**	**CB**
10	12	1878	198.12	0	1249957
20	7	2455	212.07	0	2114296
30	7	2978	408.82	0	3087448
40	12	3358	538.67	0	3863973
50	4	2871	636.2	0	5005117
60	5	4190	733.14	0	6052720
70	6	3312	746.85	0	6665071
80	4	4787	766.27	0	7481685
90	6	3864	837.15	0	7927953
100	5	3852	682.94	0	8452490
Tiempo medio de respuesta			**576.023**		

Tabla 22. Resultado de las pruebas de rendimiento al servicio Encuestas_WS.

	Servicio Encuestas_WS				
CU	**TMín**	**TMáx**	**TP**	**CE**	**CB**
10	8	2169	215.73	0	730556
20	7	2528	310.89	0	1173917
30	9	3349	370.74	0	1749954
40	9	3551	397.42	0	2316283
50	6	3068	366.71	0	2756408
60	5	4245	601.85	0	2917399
70	6	3560	655.19	0	3940033
80	4	3601	642.24	0	4493418
90	5	5553	737.49	0	4842126
100	4	3467	632.37	0	5300858
Tiempo medio de respuesta			**493.063**		

Los resultados arrojados por los casos de prueba realizados para ambos servicios, indican que el tiempo medio de respuesta del servicio Encuestados_WS es mayor que el tiempo medio de respuesta del servicio Encuestas_WS, lo cual es algo lógico, analizando que el primero de estos maneja un mayor volumen de información. Si se analiza la cantidad de bytes transferidos en cada iteración por cada servicio se puede corroborar lo anteriormente expuesto. Lo mismo sucede con el servicio Respuestas_WS, el cual procesa más información y también posee una mayor lógica de negocio que los anteriormente analizados.

Estos resultados también confirman que a mayor número de peticiones realizadas al servidor, disminuyen los tiempos mínimos y aumentan los tiempos máximos y promedios de respuestas. A pesar de todo esto, en ambos casos el tiempo de respuesta del servidor fue muy satisfactorio, ya que los tiempos promedios de respuestas oscilan en un rango muy pequeño de tiempo, además, se debe tener en cuenta que las pruebas no se están realizando en el escenario de hardware real donde debe ser desplegada esta solución en forma centralizada.

A continuación se realiza un estudio más detallado de los resultados obtenidos en las pruebas anteriores. En las **Figuras 12** y **13** se muestran dos modelos matemáticos que representan la relación existente entre

las variables de tiempo promedio de respuesta y la cantidad de usuarios concurrentes para cada servicio. Estas gráficas responden a un proceso de ajuste de curvas de tipo polinomial, el cual permite estudiar el comportamiento de estos sistemas en las distintas situaciones bajo las que operaron en los casos de pruebas ejecutados anteriormente. Para el desarrollo de las mismas se utilizó el software MATLAB en su versión 7.6.0.

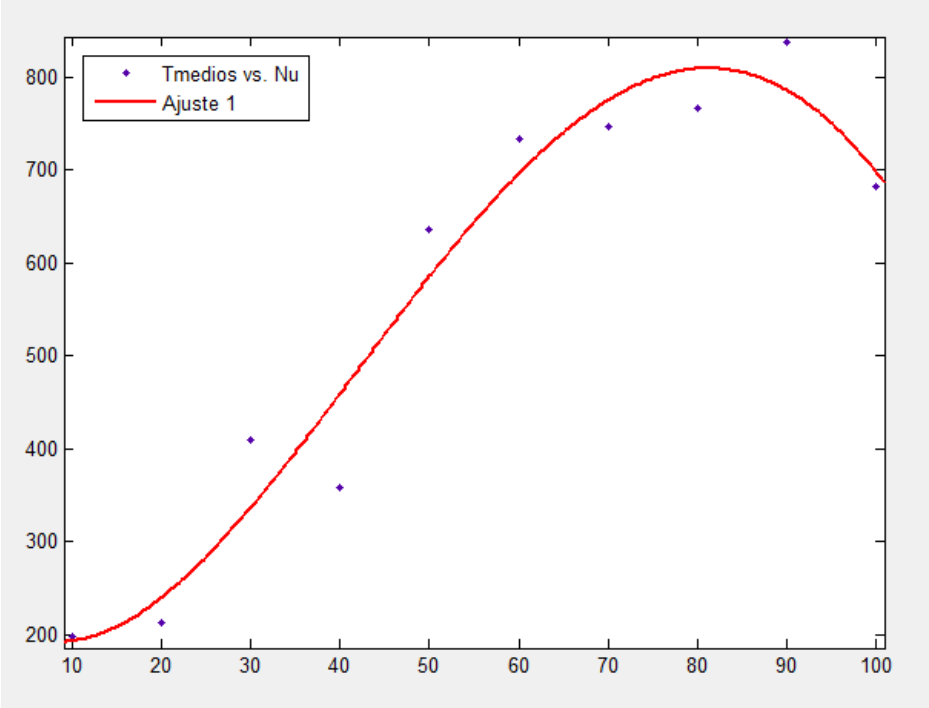


Figura 12: Cantidad de peticiones en función del tiempo del servicio Encuestados_WS

Esta gráfica responde a la función $f(x)=0.0001542x^4-0.005188x^3+0.5332x^2-7.826x+222.9$ y posee los siguientes valores como bondad de ajuste: SSE=2.552, R-square=0.951, Adjusted R-square=0.9118, RMSE=71.44. Si tenemos en cuenta que las estadísticas:

- **SSE** es la que mide la desviación total de los valores de respuesta, un valor cercano a 0 indica que el modelo tiene una componente de error más pequeño.
- **R-square** es la que mide el éxito del ajuste y es conocida como el cuadrado del coeficiente de correlación múltiple, puede tomar cualquier valor entre 0 y 1 aunque para obtener un buen modelo debe estar cercano a 1.

- **Adjusted R-square** es la que ajusta la estadística definida anteriormente sobre la base de los grados de libertad del residuo, con valor cercano a 1 indica un mejor ajuste.
- **RMSE** es el error estándar del ajuste y de regresión, un valor cercano a 0 indica que el ajuste es más útil para la predicción.

Al comparar los valores obtenidos en la gráfica anterior con los valores modelos descritos anteriormente, se puede deducir que estamos en presencia de un buen modelo.

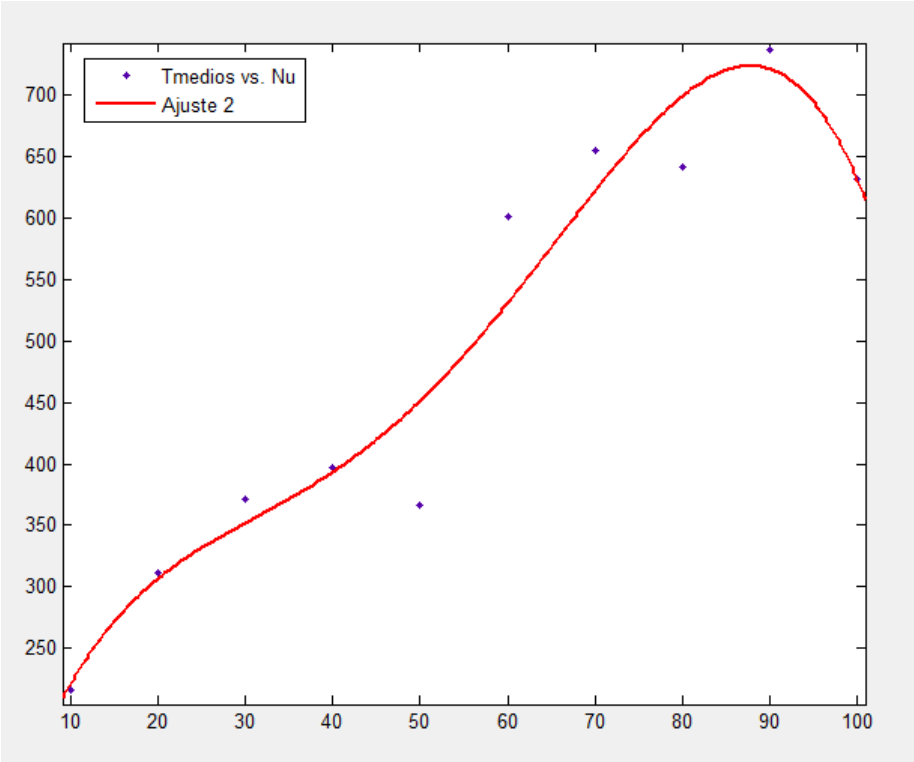


Figura 13: Cantidad de peticiones en función del tiempo del servicio Encuestas_WS.

Esta gráfica responde a la función $f(x)=-0.000182x^4+0.01248x^3-0.7893x^2+24.49x+43.14$ y posee los siguientes valores como bondad de ajuste: SSE=1.72, R-square=0.9406, Adjusted R-square=0.8931, RMSE=58.65, lo que indica que se está en presencia de un buen modelo.

A partir de los valores de tiempos medios de respuesta, obtenidos como resultados de los casos de prueba realizados a los servicios **Encuestados_WS** y **Encuestas_WS**, se puede concluir con un grado de certidumbre de 95%, que ambos poseen buenos indicadores de eficiencia.

Pruebas de Caja Negra

Las pruebas de caja negra o de comportamiento, se basan en la especificación del programa o componente a ser probado para crear los casos de prueba. Este conjunto de técnicas analiza el comportamiento del componente, estudiando las entradas y salidas generadas a partir de su uso. Para su uso, se tiene en cuenta que en todo programa existe un conjunto de entradas que causan un comportamiento erróneo, que como consecuencia produce una serie de salidas que revelan la presencia de defectos. El objetivo final es encontrar una serie de datos que pertenezcan al conjunto de entradas que causen una mayor probabilidad de comportamiento erróneo (Juristo, Moreno, & Vegas, 2006).

Para la confección de los casos de prueba de Caja Negra, se utilizó la técnica de particiones de equivalencia. Mediante la partición de equivalencia se divide el campo de entrada de un programa en clases de datos válidos e inválidos para el funcionamiento del código, a través de las cuales se derivan casos de prueba.

A continuación se muestran las clases de equivalencia y los casos de prueba generados para la realización de las pruebas de caja negra al sistema de Autenticación y Autorización de SASPED.

Tabla 23. Clases de Equivalencia

Condición de Entrada	Tipo	Clase de Equivalencia Válida	Clase de Equivalencia no Válida
Usuario	String	1: Cadena de letras 4: Que el usuario sea parte del personal docente (profesores).	2: Cadena de caracteres no letras. 3: Entrada vacía 5: Que el usuario no sea parte del personal docente.
Contraseña	String	6: Cadena de caracteres ASCII de 5 o más posiciones	7: Cadena de caracteres ASCII de menos de 5 posiciones

Tabla 24. Casos de Prueba para Técnica de Caja Negra

No. Caso	Clase de Equivalencia	Usuario	Contraseña	Resultado
1	1, 4, 6	“mfguzman”(Profesor)	“¡blackboXt3st123”	El usuario tiene acceso al sistema
2	2, 6	“12*3?”	“¡blackboXt3st123”	El usuario no tiene acceso al sistema
3	3, 6	“ ”	“¡blackboXt3st123”	El usuario no tiene acceso al sistema
4	5, 6	prcutino (Estudiante)	“estenoesmiPaSS23*”	El usuario no tiene acceso al sistema
5	1, 4, 7	“mfguzman”	“1234”	El usuario no tiene acceso al sistema

Validación de la solución de interoperabilidad para subsistemas de SASPED

Para la validación de la propuesta de solución, se procedió a utilizar las medidas de interoperabilidad presentadas en la operacionalización de la variable de la investigación. Haciendo uso de los mismos métodos de recolección de datos definidos para determinar el estado del objeto de estudio de la investigación, se presenta la modificación de dicho estado mediante la implementación de la solución de interoperabilidad.

A continuación se ofrecen las características de la concepción de interoperabilidad de subsistemas de SASPED:

- **Alto grado de integración en la implementación**, lo que se evidencia con que los subsistemas de SASPED están construidos con las mismas tecnologías (Java, JBoss Seam y PostgreSQL), y con el uso de los servicios web implementados, se elimina la barrera de incompatibilidad entre estos subsistemas y el sistema de encuestas.
- **Tiempo de interoperación eficiente**, constatado con el valor de tiempo de respuesta medio de los servicios, que se encuentra alrededor de los 500 milisegundos.
- **Existencia de compatibilidad sintáctica entre los subsistemas de SASPED**, evidenciada también en la utilización de Java como lenguaje de implementación.
- **Existencia de compatibilidad sintáctica entre el Subsistema Consultor de Información y el Sistema de Encuestas**, lo cual logran los servicios web, al estandarizar los datos intercambiados.
- **Compatibilidad semántica entre los subsistemas de SASPED y el Sistema de Encuestas**, pues al presentar semejantes modelos conceptuales de la problemática, sus modelos de datos también tienen un alto grado de semejanza.
- **Autoridades y responsabilidades bien definidas en los subsistemas**, lo cual se constata a mediante la gestión del control de acceso propuesta, la cual se adopta en el sistema SASPED.
- **Alta calidad de intercambio,** pues las vías de comunicación implementadas establecen el proceso de intercambio libre de problemas.
- **Alta calidad de uso,** pues se garantiza que el 100% de los datos requeridos en las peticiones se responden con exactitud.
- **Alta calidad de conformidad,** ya que todos los datos recibidos por cada subsistema son utilizables.

Las características expuestas constatan que la implementación de la solución garantiza la interoperabilidad entre los subsistemas de SASPED.

Conclusiones parciales

- La implementación de los servicios web soluciona el problema de la extracción de datos de la base de datos del sistema de encuestas.
- La construcción de un sistema de acceso regula la autenticación y autorización de los usuarios de SASPED.
- La solución de interoperabilidad muestra un correcto funcionamiento, dado que las pruebas de carga a los servicios web arrojaron resultados favorables, así como se validó que la gestión de la seguridad y el acceso a SASPED funciona correctamente.
- La solución de interoperabilidad modificó el estado del objeto de estudio *concepción de interoperabilidad de subsistemas de SASPED,* el cual presenta a un estado favorable.

CONCLUSIONES FINALES

- El Sistema Automatizado de Superación Pedagógica de la UCI apoya la implementación de una nueva Estrategia de Superación pedagógica, y está compuesto por un grupo de aplicaciones que resuelven determinados problemas asociados a ésta. Dada la necesidad de integración de dichos sistemas, en términos de interoperabilidad, se establecieron los fundamentos tecnológicos que la definen, los cuales se materializan en el sistema de dimensiones técnica, semántica y organizacional. Haciendo uso de la medición de estas dimensiones, se presentó un estado del objeto de estudio concepción de interoperabilidad de subsistemas de SASPED, el cual era poco favorable.
- El planteamiento de una arquitectura cliente servidor para SASPED, entendida y aceptada por el equipo de desarrollo encargado de construir el sistema, provee una vía eficiente para definir cuáles son las relaciones entre los subsistemas y cómo se establecen.
- La implementación de la solución de interoperabilidad de subsistemas de SASPED, conformada por una UDDI que contiene servicios web que apoyan el proceso de automatización de la ESP, así como un sistema de autenticación y autorización de usuarios, permiten modificar el estado observado en el objeto de estudio de la investigación con condiciones que realmente proporcionan la cooperación de dichos subsistemas.
- Al concluir la presente investigación, se cuenta con un conjunto de conocimientos que permitieron afirmar que la implementación de la presente solución garantiza la integración de los subsistemas de SASPED, en términos de interoperabilidad.

RECOMENDACIONES

Para el desarrollo de trabajos futuros asociados a los resultados obtenidos en la presente investigación se recomienda:

- Incorporar métodos de cifrado de contenido, certificados y firmas digitales, a los servicios web y al contenido de los mensajes SOAP para elevar el nivel de seguridad.
- Crear servicios web para el intercambio de información entre SASPED y otros sistemas de la UCI, como por ejemplo, el sistema de Gestión Universitaria, con el objetivo de enriquecer el proceso de superación pedagógica.

REFERENCIAS BIBLIOGRÁFICAS

- **AVISON, D.E Y FITZGERALD, G. 1995 .** *Information systems development: Methodologies, techniques, and tools.* s.l. : McGraw-Hill (New York) , 1995 . 0077092333 .
- **BALDUINO, RICARDO. 2007.** *Introduction To Open Up.* 2007.
- **BERNERS-LEE, TIM Y OTROS. 2005.** tools.ietf.org. [En línea] Enero de 2005. http://tools.ietf.org/rfc/rfc3986.txt.
- **BERTHET, C. 1988.** *Manual de Informática.* Buenos Aires : s.n., 1988. 950-02-5251-1.
- **BIANCO, PHIL, KOTERMANSKI, RICK Y MERSON, PAULO. 2007.** *Evaluating A Service-Oriented Architecture.* 2007.
- **CHEN, DAVID & OTROS. 2008.** *An Approach For Enterprise Interoperability Measurement.* 2008.
- **CIUDAD, FEBE Y OTROS, Y. 2013.** *Propuesta de Estrategia de Superación del Claustro.* 2013.
- **CNTI. 2012.** *Formulación Del Proyecto Implementación del Decreto-Ley sobre el Acceso e Intercambio Electrónico de Datos, Información y Documentos entre Órganos y Entes del Estado.* Caracas : s.n., 2012.
- **CRIADO, J Y OTROS, Y. 2010.** *Bases para una estrategia Iberoamericana de Interoperabilidad. Buenos Aires: Centro Latinoamericano da Administración para el Desarrollo.* 2010.
- **MAZILU, M. 2010.** *Database Replication.* Bucarest : s.n., 2010.
- **DATE, C.J. 2001.** *Introducción a los Sistemas de Bases de Datos.* 2001.
- **DELGADO, ANDREA Y OTROS. 2005.** *Desarrollo de aplicaciones con enfoque SOA.* Montevideo : s.n., 2005.
- **DOD. 1998.** *Levels of Information Systems Interoperability.* 1998.
- **DONATIEN, ARIAGNA Y RAMÍREZ, CARLOS. 2009.** *Sistema Para La Identificación De Aguas En Pozos Petroleros (Siapp).* La Habana : s.n., 2009.
- **FERRAIOLO, D, KUHN, RICHARD Y CHANDRAMOULI, RAMASWAMY. 2007.** *Role-Based Access Control.* s.l. : Artech, 2007.
- **FLANAGAN, DAVID. 1999.** *Java En Pocas Palabras.* 1999.
- **FLANDERS, JOHN. 2009.** MSDN Magazine. [En línea] Enero de 2009. http://msdn.microsoft.com/en-us/magazine/dd315413.aspx#id0070003.
- **GÓMEZ, KAREL. 2010.** *Propuesta De Procesos De Autenticación, Autorización Y Auditoría Para Aplicaciones Basadas En Servicios Web XML.* La Habana : s.n., 2010.

- **HAROLD, ELLIOTTE RUSTY Y MEANS, W. SCOTT. 2002.** *XML in a Nutshell, 2nd Edition.* s.l. : O'Reilly, 2002.
- **HERNÁNDEZ, M. Y HOGGUIT, O. 2011.** *Propuesta De Arquitectura De Interoperabilidad Para El Intercambio De Datos En El Órgano De Justicia-Minint.* La Habana : s.n., 2011.
- **HERNÁNDEZ, R. Y COELLO, S. 2002.** *El Paradigma Cuantitativo de la Investigación Científica.* La Habana : s.n., 2002.
- **HERNÁNDEZ, ROBERTO Y OTROS. 2006.** *Metodología de la Investigación Cuarta Edición.* s.l. : McGraw-Hill Interamericana, 2006.
- **HURWITZ, JUDITH Y OTROS, Y. 2009.** *Service Oriented Architecture For Dummies.* s.l. : Wiley Publishing, Inc., 2009.
- **IBM. 2010.** IBM. [En línea] 2010. http://pic.dhe.ibm.com/infocenter/zos/v1r11/index.jsp?topic=/com.ibm.zos.r11.ieak500/iea2k50004.htm.
- *IDABC.* **EUROPEAN INTEROPERABILITY FRAMEWORK FOR PAN-EUROPEAN eGOVERNMENT SERVICES.** 2009.
- **IEEE. 1990.** *IEEE Standard Glossary of Software Engineering Terminology.* 1990.
- **ISA. 2014.** *ISA.* [En línea] 2014. http://ec.europa.eu/isa/library/index_en.htm.
- **ISO. 2014.** ISO. [En línea] 2014. http://www.iso.org/iso/home/standards.htm.

  —. 2011. Systems and software engineering — Architecture description. [En línea] 14 de july de 2011. http://www.iso-architecture.org/ieee-1471/defining-architecture.html.
- **JURISTO, N., MORENO, A. Y VEGAS, S. 2006.** *Técnicas de Evaluación de Software.* 2006.
- **KIEWELEKAR, ARVIND. 2014.** Software Engineering Institute. [En línea] 2014. http://www.sei.cmu.edu/architecture/start/glossary/community.cfm.
- **LARMAN, CRAIG. 1999.** *UML y Patrones. Introducción al Análisis Orientado a Objetos.* 1999.
- **MANSO, M. A.,OTROS. 2008.** *Modelo de Interoperabilidad Basado en Metadatos.* 2008.
- **MAZILU, M. 2010.** *Database Replication.* Bucarest : s.n., 2010.
- **MINISTERIO DEL PODER POPULAR PARA CIENCIA, TECNOLOGÍA E INNOVACIÓN . 2012.** *Implementación del Decreto-Ley sobre el Acceso e Intercambio Electrónico de Datos, Información y Documentos entre Órganos y Entes del Estado.* Caracas : s.n., 2012.
- **NOGALES, P.M. Y GALVAN, M. 2011.** Estrategia De Interoperabilidad Para La Transferencia De Datos Entre Sistemas Erp En Cuba. [En línea] 15 de Abril de 2011. http://www.cyta.com.ar/ta1002/v10n2a3.htm. 1666-1680.

- **NOGALES, PEDRO Y GALVAN, MAGDANIS. 2011.** *Estrategia De Interoperabilidad Para La Transferencia De Datos Entre Sistemas Erp En Cuba .* 2011.
- **OASIS. 2005.** OASIS. [En línea] Febrero de 2005. https://www.oasis-open.org.
- **O'REILLY MEDIA. 2014.** xml.com. *xml.com.* [En línea] 2014. http://www.xml.com/pub/a/98/10/guide0.html?page=2#AEN58.
- **PRESSMAN, R. 2010.** *Software Engineering. A Practitioner´s Approach.* 2010.
- **PRESSMAN, ROGER. 2005.** *Ingeniería del Software, Un enfoque Práctico.* 2005.
- **RALSTON, A Y OTROS, Y. 2000.** *Encyclopedia of Computer Science.* Londres : s.n., 2000.
- **RALSTON, A., REILLY, E. Y OTROS. 2000.** *Encyclopedia of Computer Science, 4th Edition.* s.l. : Wiley, 2000.
- **ORT, ED. 2005.** *Service-Oriented Architecture and Web Services: Concepts, Technologies, and Tools***.** 2005.
- **SKONNARD, AARON. 2003.** Microsoft Developer Network. [En línea] Marzo de 2003. http://msdn.microsoft.com/en-us/library/ms995800.aspx.
- **SOMMERVILLE, IAN. 2006.** *Ingeniería del Software.* s.l. : Addison Wesley, 2006.
- **SPROTT, DAVID Y LAWRENCE, WILKES. 2004.** msdn.microsoft.com. [En línea] Enero de 2004. http://msdn.microsoft.com/en-us/library/aa480021.aspx.
- **2013. THE OPEN GROUP**. [Online] 2013. http://www.opengroup.org/soa/source-book/soa/soa.htm#soa_definition.
- **TORTOSA, S. 2006.** *Propuesta De Una Arquitectura De Software Basada En Servicios Para La Implementación De Repositorios De Objetos De Aprendizaje Distribuidos .* Alcalá : s.n., 2006.
- **UCI. 2007.** *Arquitectura Para Los Sistemas Que Conforman La Intranet Universitaria.* La Habana : Universidad de las Ciencias Informáticas, 2007.
- **UNESCO. 1998.** Declaracion Mundial Sobre La Educacion Superior en el Siglo XXI y Marco de Acción Prioritaria para el Cambio y el Desarrollo de la Educación Superior. [En línea] 1998. http://www.unesco.org/education/educprog/wche/declaration_spa.htm.
- **UNIVERSIDAD DE LAS CIENCIAS INFORMÁTICAS. 2010.** *Informe Final de Autoevaluación Institucional de la UCI al MES.* La Habana : s.n., 2010.
- **W3C. 2000.** *W3C.* [En línea] 8 de Mayo de 2000. http://www.w3.org/TR/2000/NOTE-SOAP-20000508/.

—. 2004. *W3C.* [En línea] 2 de Febrero de 2004. http://www.w3.org/TR/ws-arch/#wsdossoa.

—. 2001. W3C. [En línea] 15 de Marzo de 2001. http://www.w3.org/TR/2001/NOTE-wsdl-20010315.

—. 2012. W3C Architecture Domain. [En línea] 2012. http://www.w3.org/2002/ws/Activity.

—. 1999. www.w3.org. [En línea] 1999. http://www.w3.org/Protocols/rfc2616/rfc2616.html.

- **ZAPATER, J.J. 2005.** *Ontologías Para Servicios Web Semánticos De Información De Tráfico: Descripción Y Herramientas De Explotación.* Valencia : s.n., 2005.

GLOSARIO DE TÉRMINOS

Interoperabilidad: La habilidad de dos o más sistemas informáticos de intercambiar información y utilizar los datos intercambiados.

SASPED: Sistema Automatizado de Superación Pedagógica.

Servicio Web: Tecnología informática mediante la cual se permite transmitir información entre sistemas informáticos de forma desacoplada.

Acoplamiento: Nivel de dependencia entre entidades.

Arquitectura: Combinación predefinida de componentes de software y sus relaciones.

Encuesta de superación pedagógica: Cuestionario realizado al personal docente relacionado con el proceso de superación pedagógica.

Encuestado de superación pedagógica: Sujeto del personal docente que forma parte del proceso de superación pedagógica.

Respuesta de encuestado: Las respuestas de un encuestado de superación pedagógica correspondientes a una encuesta.

CICE: Centro de Innovación y Calidad de la Educación de la Universidad de las Ciencias Informáticas.

Estrategia de Superación Pedagógica: Nuevo modelo de superación pedagógica presentado por el CICE en el año 2013.

Subsistema: Programa de software pertenece a un sistema mayor, y que a su vez, presenta una relativa independencia al punto de que puede reutilizarse en otros sistemas.

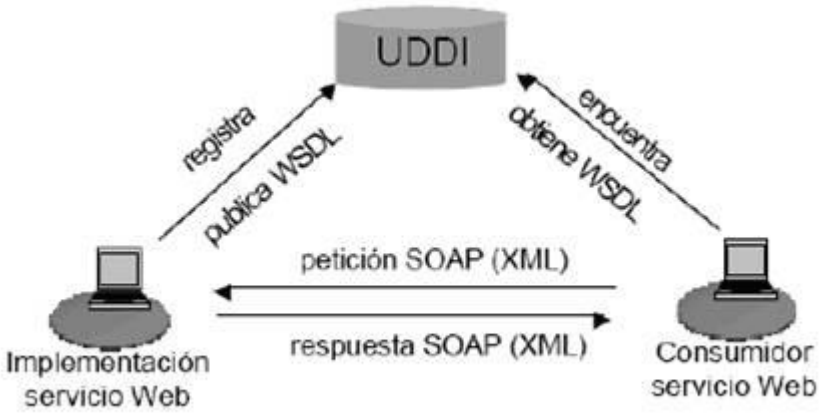
Subsistema de Gestión de Información: Sistema que gestiona información relacionada a la Estrategia de Superación Pedagógica.

Subsistema de Análisis Semántico de Texto: Programa que extrae información de las preguntas abiertas generadas en las encuestas.

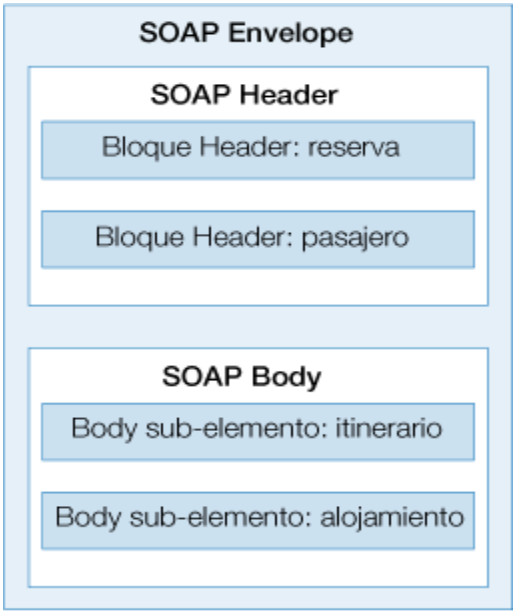
Subsistema Experto: Programa que da propuestas de estrategias de superación pedagógica utilizando técnicas de inteligencia artificial.

ANEXOS

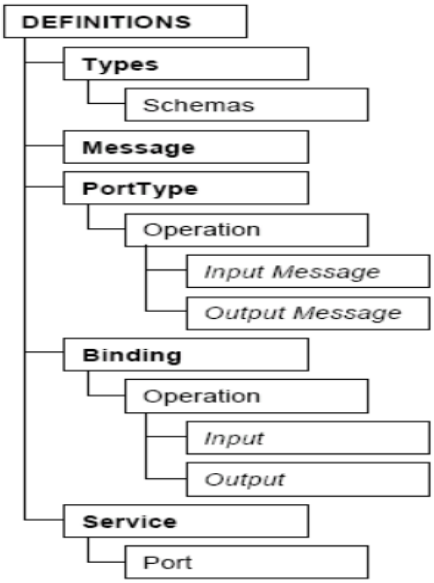
Anexo 1: Flujo de un servicio web SOAP.



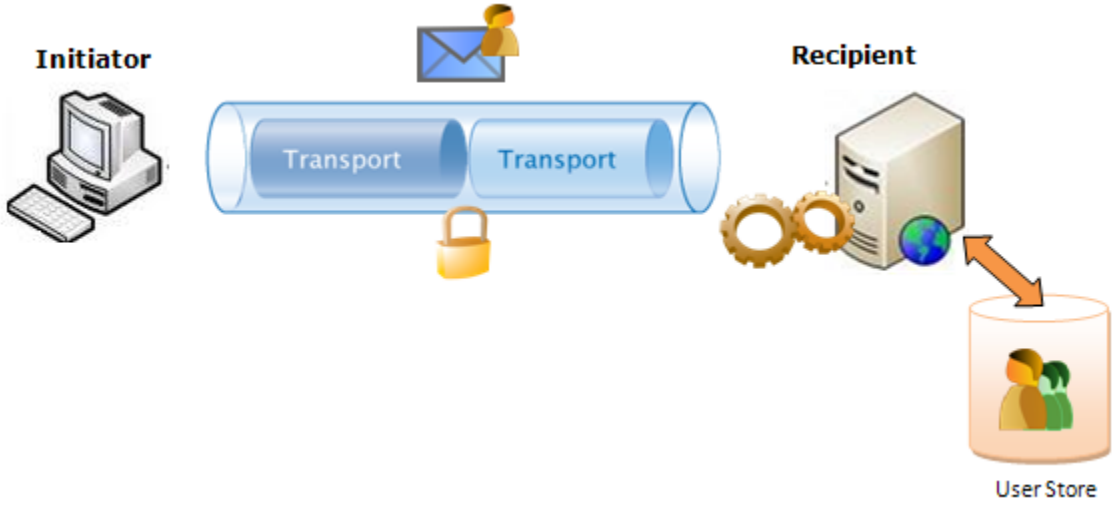
Anexo 2: Estructura de los mensajes SOAP.



Anexo 3: Estructura de los mensajes SOAP.



Anexo 4: Escenario de seguridad de los servicios web implementados.


Initiator
Transport
Transport
Recipient
User Store