

Universidad de las Ciencias Informáticas
Facultad 5



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

Título: Servicio Web del Centro de Contenido de AsixMec para la integración de los procesos CAD.

Autor: Marelys Mayté Samada Vernier

Tutor: Ing. Yerandi Marcheco Díaz

La Habana, Junio 2015

Nunca consideres el estudio como una obligación, sino como una oportunidad para penetrar en el bello y maravilloso mundo del saber.

Albert Einstein

DECLARACIÓN DE AUTORÍA

Declaro ser el autor de la presente tesis que tiene por título: **Servicio Web del Centro de Contenido de AsiXMec para la integración de los procesos CAD** y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmo la presente a los ____ días del mes de Junio del año 2015.

Marelys Mayté Samada Vernier

Ing. Yerandi Marcheco Díaz

Firma del Autor

Firma del Tutor

DATOS DEL CONTACTO

Tutor:

Ing. Yerandi Marche Díaz

- Ingeniero en Ciencias Informáticas, Universidad de las Ciencias Informáticas 2008
- Se ha desempeñado en el campo del Desarrollo de Software, Bases de Datos, Tecnologías de Internet (IT).
- Profesor Asistente

E-mail: ymarcheco@uci.cu

DEDICATORIA

A mis padres, por ser mi razón de ser.

A mi hermano, por ser mi fuente de inspiración.

A mis sobrinas, por ser mi alegría.

A mi tía Zaida en donde quiera que esté, por ser la estrella que ilumina mis días.

A toda mi familia, por apoyarme y creer en mí cuando yo pensaba en rendirme.

AGRADECIMIENTOS

A mis padres por cada segundo invertido en mí, y por todo el amor que me han dado.

A mi mamá por apoyarme en cada una de mis decisiones, por estar siempre conmigo, por ser mi guía en el camino de la vida, por sus consejos y por sus regaños, pero sobre todo por retarme a seguir adelante.

A mi papá por ser más que un padre un amigo, por cumplir todos mis caprichos, por todo el sacrificio realizado para que nunca me faltase nada, por nunca dejarme sola y darme todo su apoyo, pero sobre todo por su amor incondicional.

A mi tía Zaida, que aunque no se encuentre conmigo hoy, sé que me está cuidando desde el cielo, y que guía cada uno de mis pasos, por protegerme y amarme mientras pudo y por convertirme en la persona que soy hoy.

A mi hermano por ser mi ejemplo a seguir, por ayudarme en los años de mi carrera, por pasar conmigo los momentos dulces y amargos, pero sobre todo por soportarme.

A mi cuñada Aysel, por todo su apoyo, por cada consejo que me dio para mi vida personal y profesional, pero sobre todo gracias por darme las sobrinas más bellas del mundo. A mis peques Alexa y Alis, por ser la luz de mis ojos, por cada una de sus sonrisas, por sus abrazos, por sus cariños, por todos los momentos de felicidad que me han dado desde que llegaron a mi vida.

A mi tía Xiomara, por cada una de sus locuras, que hacían que hasta en los momentos más triste siempre tuviese una sonrisa en la cara. A mi primo Ray por siempre tomarme como su niña cuando yo era pequeña, y por seguir queriéndome tanto.

A mis amigas Ailin, Kathy y Yeily, por todos los momentos que pasamos juntas. Por nunca abandonarme ni siquiera en los momentos más malos, por darme su hombro para llorar y llorar junto conmigo, por compartir no solo los momentos malos, sino los momentos de alegría, de amor. A Kathy y a Yeily que aún en la distancia, siguen preocupándose por mí y dándome consejo cuando ellas creen que lo necesito. A Ailin por apoyarme en mis decisiones incluso

cuando ella no está de acuerdo, por estar siempre al pendiente de mí. A ellas tres, gracias por ser más que mis amigas mis hermanas.

A Yosva por ser mi amigo, y mi compañero de estudio, por estar compitiendo conmigo haciendo que me esforzara más en mis estudios para ganarle.

A todos los amigos que conocí, tanto dentro como fuera de la universidad, aquellos que me tendieron la mano cuando más lo necesite, a los que hicieron más ameno mis momentos de universidad, a mis compañeros de aula aquellos que lucharon a mi lado soportando mis imperfecciones.

A todos los profesores que tuve a lo largo de mi carrera, a los buenos, y a los no tan buenos. A los que me pusieron obstáculos y a los que me ayudaron a vencerlos y lograr mi meta.

A mi tutor, mi profesor, mi amigo, mi pareja, por soportar cada una de mis malcriadeces, mi forma de ser, mis stress. Gracias porque a pesar de todas las cosas vividas juntos, siempre estuviste conmigo, en las buenas, en las malas y en las regulares. Gracias por todos y cada uno de los consejos dados, que aunque no lo creas siempre los tomaba en cuenta, gracias por hacer de mis problemas tus problemas, en fin gracias por ser mi cosi.

RESUMEN

“AsiXMec” es un software de Diseño Asistido por Computadora (CAD) basado en tecnologías libres que se desarrolla como parte del proyecto Diseño y Simulación de Estructuras Mecánicas del centro Vertex, perteneciente a la Universidad de las Ciencias Informáticas. Uno de los módulos que se desarrolla es el Centro de Contenido, que permite a los diseñadores e ingenieros utilizar modelos CAD de partes y piezas normalizadas según los estándares internacionales. Además, en el proyecto se han desarrollado varios visualizadores de modelos CAD: para la aplicación principal, para aplicaciones de dispositivos móviles y para aplicaciones web. Cada uno de ellos fue desarrollado con tecnologías independientes e incompatibles entre sí, lo que impide realizar un acceso homogéneo y eficaz a la información de los modelos CAD almacenados en el Centro de Contenido. En esta investigación se implementa una capa de servicios web (desarrollada con lenguaje PHP y el marco de trabajo Yii) para proveer a cada uno de los visualizadores una interfaz estándar de acceso a los modelos CAD. De esta forma se garantiza un correcto modelo de integración entre las aplicaciones clientes y el Centro de Contenido.

PALABRAS CLAVES: Diseño Asistido por Computadoras (CAD), Integración, Servicios Web, Centro de Contenidos.

ABSTRACT

“AsiXMec” is a Computer-Aided Design software (CAD) based on free technologies that is being developed as part of the project "Design and Simulation of Mechanical Structures", being performed at Vertex, a Center of the University of Informatics Sciences (UCI). One of the modules already developed is the Content Center, which allows designers and engineers to use standard CAD models as defined in international norms. Besides, there were developed several CAD viewers: for the main, mobile and web applications. Each of them was developed with different technologies, incompatible each other; so, it was not possible to make them access CAD models stored in the Content Center in a standard way. In this investigation, a web service layer was developed (using PHP programming language and Yii framework) in order to provide a standard access to CAD models for each visualization application. That way a correct integration model for client applications and Content Center is guaranteed.

KEYWORDS: Computer-Aided Design (CAD), Integration, Web Services, Content Center.

Contenido

INTRODUCCIÓN	15
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	18
1.1 Introducción	18
1.2 Conceptos asociados al dominio del problema	18
1.2.1. Sistemas de Diseño Asistido por Computadora	18
1.2.2. Servicios Web.....	19
1.3 Descripción de los sistemas desarrollados por el proyecto DISEM	19
1.3.1. Base de Datos de Piezas Normadas	20
1.3.2. Centro de Contenido.....	20
1.3.3. AsiXMec	20
1.3.4. CADViewer	21
1.3.5. MecViewer.....	21
1.3.6. Sitio Web	21
1.4 Integración de aplicaciones.....	22
1.4.1. Tendencia hacia la integración de aplicaciones	22
1.5 Servicios Web.....	23
1.5.1. Estándares y protocolos de los Servicios Web.....	24
1.5.2. Necesidad de usar los Servicios Web.....	25
1.5.3. Ventajas de los Servicios Web.....	26
1.5.4. Desventajas de los Servicios Web	27
1.6 Metodología de Desarrollo	27
1.6.1. Proceso Racional Unificado, RUP.....	27
1.6.2. Extreme Programming, XP	28
1.7 Tecnologías y herramientas a utilizar.....	29
1.7.1. Lenguaje de Programación	29
1.7.1.1. PHP, versión 5.4	29
1.7.1.2. Java	30
1.7.2. Herramientas a utilizar	30
1.7.2.1. Apache HTTP Server, versión 2.4.....	30
1.7.2.2. Visual Paradigm, versión 8.0.....	31
1.7.2.3. DbDesigner, versión 4.0.....	31
1.7.3. Framework de desarrollo	32
1.7.3.1. Framework Yii, versión 1.10.....	32
1.7.3.2. NuSOAP	32
1.7.3.3. Extensión de SOAP para PHP	32
1.7.4. IDEs.....	33
1.7.4.1. NetBeans, versión 7.0.....	33
1.7.4.2. Zend Studio, versión 8.0.....	33

1.7.4.3.	PhpStorm, versión 7.0.....	34
1.8	Conclusiones parciales del capítulo.....	34
CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA		35
2.1.	Introducción	35
2.2.	Descripción de los procesos fundamentales del Centro de Contenido.....	35
2.3.	Descripción de la arquitectura utilizada.....	36
2.4.	Implementación de Servicios Web con el framework Yii	37
2.5.	Análisis y requerimientos de los Servicios Web	39
2.5.1.	Listado de requerimientos.....	39
2.5.2.	Fase de exploración.....	40
2.5.2.1.	Historias de Usuario	40
2.5.3.	Fase de planificación	42
2.5.3.1.	Estimación del esfuerzo	42
2.5.3.2.	Plan de Iteraciones.....	43
2.5.3.3.	Plan de Entrega	43
2.6.	Conclusiones parciales del capítulo.....	44
CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA		45
3.1.	Introducción	45
3.2.	Implementación	45
3.2.1.	Diagrama de despliegue	45
3.2.2.	Estándar de codificación.....	46
3.2.2.1.	Conversión de nomenclatura.....	46
3.2.2.2.	Indentación.....	46
3.2.2.3.	Comentarios.....	47
3.2.2.3.1.	Formatos de los comentarios en la implementación.....	47
3.2.2.3.1.1.	Bloque de comentario.....	47
3.2.2.3.1.2.	Comentarios de una línea.....	48
3.2.2.3.1.3.	Comentario de fin de línea.....	48
3.2.3.	Tareas de implementación.....	48
3.3.	Descripción de los servicios web	51
3.3.1.	Servicio Web CategorySoap.....	51
3.3.2.	Servicio Web SubCategorySoap.....	52
3.3.3.	Servicio Web FamilySoap.....	53
3.3.4.	Servicio Web PieceSoap	53
3.4.	Aplicación web para el consumo de la capa de servicios web	54
3.4.1.	Consumo del servicio web CategorySoap.....	54
3.4.2.	Consumo del servicio web SubCategorySoap	55
3.4.3.	Consumo del servicio web FamilySoap.....	55
3.4.4.	Consumo del servicio web PieceSoap	56

3.5. Validación de la propuesta.....	57
3.5.1. Pruebas unitarias.....	58
3.5.1.1. Framework de Prueba PHPUnit	58
3.5.2. Pruebas de aceptación	60
3.5.3. Resultados de las pruebas.....	65
3.6. Conclusiones parciales del capítulo	65
CONCLUSIONES GENERALES.....	67
RECOMENDACIONES	68
BIBLIOGRAFÍA.....	69
ANEXOS.....	72

ÍNDICE DE TABLAS

Tabla 2.1 Descripción de la historia de usuario: Listar Categorías.	41
Tabla 2.2 Descripción de la historia de usuario: Listar Subcategorías.....	41
Tabla 2.3 Descripción de la historia de usuario: Listar Familias de Piezas.....	41
Tabla 2.4 Descripción de la historia de usuario: Listar Piezas.....	41
Tabla 2.5 Descripción de la historia de usuario: Detalle de la Pieza.....	42
Tabla 2.6 Descripción de la historia de usuario: Buscar Pieza.	42
Tabla 2.7 Distribución de la estimación por cada historia de usuario.	43
Tabla 2.8 Distribución de historias de usuario por iteración.....	43
Tabla 2.9 Fecha de entrega por cada iteración.	44
Tabla 3.1 HU planificada para la 1ra Iteración.....	49
Tabla 3.2 Descripción de la tarea de desarrollo #1.	49
Tabla 3.3 Descripción de la tarea de desarrollo #2.	49
Tabla 3.4 HU planificada para la 2da Iteración.....	50
Tabla 3.5 Descripción de la tarea de desarrollo #3.	50
Tabla 3.6 Descripción de la tarea de desarrollo #4.	50
Tabla 3.7 HU planificada para la 3ra Iteración.....	51
Tabla 3.8 Descripción de la tarea de desarrollo #5.	51
Tabla 3.9 Descripción de la tarea de desarrollo #6.	51
Tabla 3.10 Pruebas unitarias realizadas.	59
Tabla 3.11 Caso de prueba: Listar Categorías.....	60
Tabla 3.12 Caso de prueba: Listar Subcategorías.	61
Tabla 3.13 Caso de prueba: Listar Subcategorías dada una Categoría.	61
Tabla 3.14 Caso de prueba: Listar Familias de Piezas.....	62
Tabla 3.15 Caso de prueba: Listar Familias de Piezas dada una Subcategoría.	62
Tabla 3.16 Caso de prueba: Listar Piezas en Vista de Lista.....	62
Tabla 3.17 Caso de prueba: Listar Piezas en Iconos.	63
Tabla 3.18 Caso de prueba: Listar Piezas dada una Familia de Piezas.....	63
Tabla 3.19 Caso de prueba: Información de la Pieza.	64
Tabla 3.20 Caso de prueba: Detalle de la Pieza.	64
Tabla 3.21 Caso de prueba: Buscar Pieza.....	65

ÍNDICE DE FIGURAS

Figura 1 Visualización en el software MecViewer.....	21
Figura 2 Diagrama de la Arquitectura Modelo-Vista-Controlador de la capa de servicios web.	36
Figura 3 Arquitectura de Yii para Servicios Web.	37
Figura 4 Ejemplo de un método SOAP disponible en un Controlador de Yii.....	38
Figura 5 Ejemplo de acceso a un Servicio Web.	39
Figura 6 Diagrama de Despliegue.....	46
Figura 7 Ejemplo de estándar de codificación (Conversión de nomenclatura e Indentación).	47
Figura 8 Ejemplo de Bloque de comentario.....	47
Figura 9 Ejemplo de comentarios de fin de línea.....	48
Figura 10 Prueba Unitaria para la función categoryList.	59
Figura 11 Pruebas de Aceptación realizadas en cada iteración.	65

INTRODUCCIÓN

El avance de las tecnologías de la informática y las comunicaciones, ha estado marcado por la constante utilización del poder de cálculo de las computadoras, en busca de mejorar el proceso de desarrollo de las actividades de innovación y creación en diferentes campos de la ciencia. En áreas, como la arquitectura y la mecánica, el uso de los sistemas computacionales ha sido la respuesta a la complejidad de los análisis que se realizan en algunas de sus ramas. Esto se ha logrado en gran medida por la creación y mejora de los programas de cálculo y a su vez por la mejora en el rendimiento de los equipos informáticos. Debido al uso de estas herramientas, se ha obtenido en la mayoría de los casos resultados más precisos y sobre todo una reducción de costos, ya que permiten analizar virtualmente el comportamiento de nuevas soluciones.

En este contexto han surgido tecnologías que permiten simplificar, optimizar y elevar la calidad de los procesos de diseño; siendo el caso de los sistemas de Diseño Asistido por Computadora (CAD, del inglés *Computer Aided Design*), software que se encargan de asistir a ingenieros, arquitectos y diseñadores en sus respectivas actividades, es decir, son sistemas informáticos destinados a asistir al diseñador en su tarea específica, ya sea el dibujo técnico como la documentación del mismo, aunque también permite realizar tareas relacionadas con la presentación y el análisis del diseño que se realiza.

Con el propósito de vincularse a la utilización de los sistemas CAD y buscando alcanzar soberanía tecnológica en el campo del diseño y modelación de estructuras mecánicas, surge el proyecto Diseño y Simulación de Estructuras Mecánicas (DISEM) de la Universidad de las Ciencias Informáticas (UCI). Uno de los módulos que se desarrolla en este proyecto es el Centro de Contenido, conocido en los sistemas CAD por su equivalente en inglés *Content-Center*, el cual constituye la biblioteca de partes y piezas estandarizadas, que permite importar los modelos que están definidos en los estándares internacionales hacia los software CAD. En este proyecto también se desarrollan diferentes sistemas para el diseño y simulación de estructuras mecánicas, los cuales son: Base de Datos de Piezas Normadas, AsiXMec, CADViewer, Sitio Web del proyecto DISEM y MecViewer, sistemas que trabajan de forma independiente y con funciones específicas.

Los sistemas están desarrollados con lenguajes de programación diferentes, lo que trae como consecuencia que se haga una implementación específica para la conexión a la base de datos de cada producto. Esto conlleva que para cada software se implemente la conexión con un lenguaje determinado, provocando así que se derroche esfuerzo innecesario. Los mismos también se encuentran desarrollados para plataformas diferentes (Windows, Linux y Android), obligando a que se hagan implementaciones específicas para la conexión con la base de

datos de cada producto, provocando, que existan desarrolladores trabajando para lograr la conexión en Windows con bibliotecas diferentes a las que utilizan otros desarrolladores en Linux, sucediendo lo mismo con Android.

Las aplicaciones implementan la conexión en el Sistema Gestor de Bases de Datos (SGBD) PostgreSQL, lo que imposibilita la migración de la base de datos hacia otras tecnologías como el SGBD MySQL. Trayendo consigo que si en un futuro cambia el SGBD, debería ser implementada nuevamente la conexión para cada una de las aplicaciones.

Teniendo en cuenta los elementos anteriores se plantea el siguiente **problema de la investigación**:

¿Cómo brindar un mecanismo único de acceso a la Base de Datos de Piezas Normadas del proyecto DISEM?

A partir del problema que se presenta se define como **objeto de estudio**: “el proceso de desarrollo de servicios web para la integración de aplicaciones”, sobre esta base y con el fin de dirigir el resultado de la investigación hacia la solución del problema, se define como **objetivo general**: desarrollar un servicio web para acceder a la Base de Datos de Piezas Normadas que pueda ser utilizado por las aplicaciones del proyecto DISEM, y como **campo de acción**: “el desarrollo y diseño de servicios web para arquitecturas orientadas a servicios”.

Con el fin de orientar el trabajo hacia un resultado alcanzable se definen las siguientes **tareas de investigación**:

- Elaborar el marco teórico de la investigación científica a partir del estado del arte.
- Caracterizar las metodologías, herramientas y tecnologías a utilizar en el desarrollo de servicios web.
- Identificar los requisitos funcionales y no funcionales para la creación de servicios web.
- Caracterizar los principales servicios web.
- Diseñar la aplicación web para el consumo de los servicios web.
- Implementar las funcionalidades identificadas.
- Diseñar los casos de pruebas para determinar el correcto funcionamiento de los requisitos.
- Realizar pruebas funcionales a la herramienta.

Para llevar a cabo la investigación se utilizaron los siguientes métodos:

Métodos teóricos: permiten estudiar las características del objeto de estudio para identificar las relaciones que fluyen alrededor de este, contribuyendo al desarrollo de teorías científicas.

- **Análisis-Síntesis:** se descomponen los diferentes elementos teóricos que giran alrededor de las arquitecturas de servicios web, donde se ven las características del servidor de servicios web, de los clientes que consumirán dicho servicio, y del servidor de base de datos. Por esto se sintetizan las variables importantes que influyen en la solución del problema: los sistemas operativos de los dispositivos, el tipo de base de datos empleado, el formato exigido para cada aplicación y la arquitectura cliente-servidor.
- **Inductivo-Deductivo:** se deduce, mediante los datos arrojados por el estudio de diferentes arquitecturas de servicios web existentes en la actualidad, el tipo de arquitectura que más se ajuste a las características planteadas. Con esta estructura se induce que es capaz de solucionar las necesidades de los servicios web del sistema CAD.
- **Histórico-lógico:** se analizan las tendencias y evolución de los sistemas CAD, que evidencia la importancia actual de los servicios web en proyectos de esta índole. Se determinan bajo qué tecnologías y condiciones se obtienen mejores beneficios y las diferentes variables que influyen.

Métodos empíricos: representan un nivel del estudio que precede de la experiencia adquirida y se desarrolla una elaboración racional que describe y explica las características del fenómeno que se investiga.

- **Análisis documental:** se realiza un análisis de los documentos que rigen los estándares internacionales de partes y piezas mecánicas; que permite generar una descripción de los datos que se almacenan en la base de datos y que serán consultados por el servicio web.
- **Observación científica:** se realiza una observación de las características de los dispositivos del proyecto DISEM que deberán acceder a la base de datos, donde se recogen las características del sistema operativo, y las prestaciones de hardware y de servicios existentes.
- **Análisis estático:** método observacional de investigación informática. Se inspecciona y recorre la aplicación en búsqueda de errores.

A partir de los conocimientos empíricos y teóricos adquiridos se puede definir como **Hipótesis:** “la aplicación del servicio web para la Base de Datos de Piezas Normadas hace posible su consulta desde las aplicaciones de diferentes aplicaciones, desde plataformas y dispositivos de diferentes características de forma precisa”.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.1 Introducción

Este capítulo constituye la base teórica para la comprensión del trabajo que se desarrolla, se analizan los principales conceptos y definiciones asociados al dominio del problema, que sirven de apoyo durante el desarrollo de la investigación. Se describen las principales características relacionadas con los componentes de software, y se aborda además lo relacionado a los servicios web, así como en qué consisten y para qué se utilizan, exponiendo las razones por las cuales es necesario su empleo en la integración de los sistemas que forman parte del proyecto DISEM.

1.2 Conceptos asociados al dominio del problema

Con el objetivo de lograr una mayor comprensión acerca de los temas que se abordarán a lo largo de este capítulo y en los próximos capítulos, se mostrarán algunos conceptos que están relacionados con la investigación realizada.

1.2.1. Sistemas de Diseño Asistido por Computadora

Algunos autores definen los sistemas CAD desde diferentes puntos de vista, mostrando a continuación algunos identificados durante el proceso de la investigación.

“Los sistemas CAD: es el uso de un amplio rango de herramientas computacionales que asisten a ingenieros, arquitectos y a otros profesionales del diseño en sus respectivas actividades [1].

“Conjunto de aplicaciones informáticas que permiten al ingeniero u otros profesionales en el diseño, “definir” un producto a fabricar” [2].

“Es un sistema electrónico que permite diseñar nuevas partes o productos o modificar otros ya existentes, en sustitución del dibujo tradicional a mano” [20].

Teniendo en cuenta los conceptos anteriormente expuestos, se plantea para el presente documento, que los sistemas CAD son herramientas que permiten crear modelos de piezas, circuitos integrados, ensamblajes y otros.

1.2.2. Servicios Web

Acercas de los servicios web (SW) diferentes autores plantean sus definiciones, a continuación se muestran algunos de los conceptos que más se ajustan al contexto de esta investigación:

“Aplicaciones modulares y auto-descriptivas que se pueden publicar, ubicar e invocar desde cualquier punto de red o desde el interior de una red local, basados en estándares abiertos de Internet. No es necesario que el proveedor y el usuario de un servicio web tengan el mismo sistema operativo y utilicen el mismo lenguaje de programación, dado que se basan en estándares aceptados plenamente por la industria como, Lenguaje Extensible de Etiquetas (XML, del inglés *Extensible Markup Language*), Lenguaje de Etiquetas de Hipertextos (HTML¹, del inglés, *HyperText Markup Language*) y el Protocolo para la Transferencia de Correo (SMTP, del inglés *Simple Mail Transfer Protocol*)” [3].

“Una biblioteca de clases accesible mediante el Protocolo de Transferencia de Hipertextos (HTTP del inglés, *HyperText Transfer Protocol*). Su interfaz está descrita en XML, de forma estándar e independiente del lenguaje de implementación subyacente” [4].

“Uno de los usos principales de los servicios web es permitir la comunicación entre las empresas y entre las empresas y sus clientes. Los servicios web permiten a las organizaciones intercambiar datos sin necesidad de conocer los detalles de sus respectivos sistemas de información” [5].

Teniendo en cuenta las definiciones anteriores, se puede decir que un servicio web es básicamente una función o procedimiento que puede ser accedida vía web por cualquier programa o aplicación, sin importar en que plataforma reside el servicio o en que lenguaje ha sido desarrollado, describiendo una forma estandarizada de integración web mediante el uso de estándares sobre los protocolos de Internet.

1.3 Descripción de los sistemas desarrollados por el proyecto DISEM

En el proyecto DISEM, se desarrollan diferentes sistemas para el diseño y simulación de estructuras mecánicas, estos son: Base de Datos de Piezas Normadas, Centro de Contenido, AsiXMec, CADViewer, MecViewer y el sitio web del proyecto DISEM.

A continuación se hace una descripción de los principales sistemas que componen el proyecto DISEM:

¹ Estándar que define una estructura básica y un código para la definición de contenido de una página web.

1.3.1. Base de Datos de Piezas Normadas

La Base de Datos de Piezas Normadas surge buscándole solución al problema de modelar una base de datos capaz de almacenar partes estandarizadas, sin importar la diversidad y complejidad que tengan las familias y piezas que se guardan en las mismas. Por lo tanto esta base de datos, almacena prototipos de piezas correspondientes a la norma ISO², aunque cuenta con la estructura para almacenar otras normas como la ANSI³, DIN⁴, entre otras. Esta base de datos almacena la información relacionada con las familias de piezas. Esta familia contiene las piezas, y cada pieza tiene diferentes características de diferentes tipos de datos, por lo que para el desarrollo de esta base de datos se usó una variante del patrón de diseño llamado Entidad-Atributo-Valor.

Esta Base de Datos Objeto-Relacional, se encuentra alojada en el Sistema Gestor de Bases de Datos libre PostgreSQL, siendo esta base de datos la que brinda los modelos necesarios para ser utilizada por los diferentes sistemas del proyecto.

1.3.2. Centro de Contenido

El Centro de Contenido es la aplicación encargada de gestionar las piezas estandarizadas por las diferentes normas internacionales como la ISO, ANSI, DIN, entre otras, teniendo como componente básico una familia de piezas. Las familias se organizan en categorías en una biblioteca del Centro de Contenido y una categoría es una agrupación lógica de los tipos de piezas. Una familia de piezas contiene miembros que tienen las mismas propiedades de familia, y que representan variaciones de tamaño u operación. Es la interfaz gráfica de usuario que permite explorar la Base de Datos de Piezas Normadas. Es una aplicación de escritorio que se encuentra desarrollada con el lenguaje de programación C++ utilizando el marco de trabajo (en inglés, *framework*) de Qt.

1.3.3. AsiXMec

AsiXMec es un sistema informático basado en tecnologías libres. Es un modelador geométrico y paramétrico basado en *Feature e Historial* de Operaciones, que orienta el diseño a partir de bocetos 2D. Permite a sus usuarios manipular, crear, editar, importar y exportar archivos .step, .iges, .stl, compatibles con otros sistemas libres como DraftSight y LibreCAD y sistemas propietarios como AutoCAD, SolidWorks e Inventor, entre otros de los más conocidos. Es una aplicación de escritorio desarrollada con lenguaje C++, basado en la plataforma de desarrollo de la línea CAD/CAE, utiliza el *framework* (OCE)⁵, *framework* Qt 4,

² Organización Internacional de Normalización. Organismo encargado de promover el desarrollo de normas internacionales de fabricación, comercio y comunicación para todas las ramas industriales

³ Instituto Nacional Estadounidense de Estándares.

⁴ Instituto Alemán de Normalización.

⁵ Edición para la Comunidad de OpenCASCADE, en inglés: open CASCADE Community Edition. Open CASCADE es un kit de desarrollo de software para aplicaciones CAD.

es multilinguaje y tiene soporte para *plugins*. Ofrece además una biblioteca de piezas estandarizadas que cubre la norma ISO y provee funcionalidades que ayudan al usuario en el proceso de modelado. Este software es integrado con el Centro de Contenido para importar prototipos de la Base de Datos de Piezas Normadas.

1.3.4. CADViewer

CADViewer es un visor de modelos CAD, con soporte para los formatos STEP, IGES y STL. Este software permite visualizar los modelos en un formato y guardarlo en otro, además permite cambiar la estructura de como se desea ver la pieza. Este software actualmente no permite visualizar los modelos almacenados en la base de datos del proyecto, aunque este es un requisito funcional de alta importancia.

1.3.5. MecViewer

MecViewer es un visor de modelos CAD, que permite la visualización de estructuras mecánicas en el formato STL. Es una aplicación para sistemas móviles con Android. MecViewer permite descargar las piezas al teléfono desde un servidor web de prueba, conectándose mediante la dirección del host, facilitando que se permita interactuar con la pieza, como puede ser cambiarle el color, la textura, acercarla o alejarla, cambiarle el modo a *wireframe*, rotarla, entre otras operaciones, como se muestra en la Figura 1. Entre los requisitos funcionales aún no implementados se encuentra el de visualizar modelos CAD almacenados en la Base de Datos de Piezas Normadas. Está desarrollado con el motor Unity y el lenguaje C#.



Figura 1 Visualización en el software MecViewer.

1.3.6. Sitio Web

El sitio web del proyecto DISEM es una aplicación web que está en desarrollo, será la cara del proyecto para Internet y tiene como propósito promocionar los productos del proyecto.

Entre los requisitos funcionales que aún no tienen respuesta se encuentra el requisito de visualizar modelos CAD almacenados en la Base de Datos de Piezas Normadas. Este software se está desarrollando con el lenguaje php.

1.4 Integración de aplicaciones

La integración de aplicaciones consiste en conectar en una misma aplicación llegando a verse como un sistema único, a todas las aplicaciones independientes, para lograr así una mayor organización. Permitiendo principalmente el intercambio de datos entre diferentes aplicaciones y que las mismas trabajen de manera conjunta.

1.4.1. Tendencia hacia la integración de aplicaciones

En ocasiones se cuenta con diferentes aplicaciones desplegadas en el transcurso del tiempo que son desarrolladas con diferentes lenguajes, que usan diferentes tecnologías, que se despliegan en diferentes plataformas de hardware y sistemas operativos con interfaces de usuario inconsistentes. Esto trae como resultado que existan funcionalidades aisladas, múltiples instancias de los mismos datos, actividades manuales redundantes, costos más altos y respuestas ineficientes para sus clientes. De ahí surge la necesidad de la integración de aplicaciones, capaz de capturar los datos, las herramientas y funciones de una variedad de aplicaciones y combinarlas con la información proporcionada en otras aplicaciones, para que se pueda acceder y utilizar toda la información en un sencillo cliente, en lugar de perder tiempo y esfuerzo mediante la ejecución de varios programas y la diversificación de sus datos.

La integración de aplicaciones no solo se hace de gran necesidad en las aplicaciones internas de una organización, sino que se ha vuelto de gran importancia y necesidad a la hora de integrar aplicaciones entre organizaciones asociadas, y esto ha sido producto de la evolución constante de los mercados, debido a la incorporación del Internet en las organizaciones. Este proceso de integración es de gran costo, teniendo además un gran nivel de complejidad y demanda de tiempo y suele requerir soluciones informáticas distribuidas.

Adolfo de Soto en su artículo *“Nuevas Tendencias en Sistemas de Información: Procesos y Servicios”* del año 2006 expresa que: “para lograr la construcción rápida de sistemas distribuidos se necesita una nueva capa software que realice la abstracción de la comunicación entre sistemas heterogéneos, esta capa se coloca entre las aplicaciones distribuidas y los servicios de red, los sistemas operativos y el hardware de comunicaciones” [7]. En busca de resolver el problema de la comunicación entre los procesos, independientemente del lenguaje y la plataforma que se desarrollen surge la capa *middleware*, que son servicios y abstracciones que facilitan el diseño, desarrollo, integración y despliegue de aplicaciones y sistemas distribuidos en entornos de redes heterogéneas [8].

El desarrollo del Internet y el uso masivo de la web en las organizaciones ha traído como consecuencia que se utilicen nuevas técnicas a la hora de integrar aplicaciones, ya que los mecanismos de los *middleware* no pueden ser utilizados para la integración de aplicaciones entre diferentes organizaciones.

Con el avance del Internet aparecieron protocolos de interacción estándar como el HTTP y formatos de datos como el XML, que las compañías fueron tomando rápidamente y adaptándose a los mismos, creándose así un estándar donde establecer una infraestructura de *middleware* común que minimiza la heterogeneidad. Aquí es donde logran tomar un mayor auge los servicios web, aplicaciones auto-contenidas, auto-descriptivas y modulares, que pueden ser publicadas, localizadas e invocadas a través de la web y que cuentan con un mecanismo estándar para establecer la comunicación con otros tipos de software a través de la red [9].

1.5 Servicios Web

Los servicios web nos ofrecen una sencilla forma de extender la funcionalidad de nuestras aplicaciones empresariales, ya que los mismos se basan en una tecnología, que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. Distintas aplicaciones desarrolladas en lenguajes de programación diferentes, y ejecutadas sobre cualquier plataforma, pueden utilizar los servicios web para intercambiar datos en redes de ordenadores como Internet.

Son muchos los autores que han dado su definición acerca de los servicios web, algunos de los cuales se muestran a continuación.

La W3C⁶ lo define como: “un servicio web es una aplicación software identificada por un Identificador de Recursos Uniforme (URI ⁷del inglés, *Uniform Resource Identifier*), cuyas interfaces se pueden definir, describir y descubrir mediante documentos XML. Los servicios web hacen posible la interacción entre “agentes” software (aplicaciones) utilizando mensajes XML intercambiados mediante protocolos de Internet” [11].

Jorge Armando Portilla Galviz en su artículo *Arquitecturas Orientadas a Servicios como soporte a modelos de educación virtual (módulos: académico, investigación, entorno y los servicios a estudiantes)* plantea: “los servicios web son pequeños programas formados por varios componentes, que permiten ser publicados en directorios e invocados para su ejecución por otros programas vía HTTP, generando una respuesta en XML” [12].

⁶ World Wide Web Consortium. Comunidad internacional para el desarrollo de estándares de la web.

⁷ Cadena de caracteres que identifica los recursos de una red de forma unívoca.

1.5.1. Estándares y protocolos de los Servicios Web

Un servicio web está caracterizado por cuatro estándares tales como: XML, el Protocolo Simple de Acceso a Objetos (SOAP del inglés, *Simple Object Access Protocol*), la Descripción Universal, Descubrimiento e Integración (UDDI del inglés, *Universal Description, Discovery and Integration*) y el Lenguaje de Descripción de Servicios Web (WSDL del inglés, *Web Services Description Language*), los cuales al trabajar juntos proporcionan una funcionalidad básica de tipo "solicitud/respuesta". Estos estándares permiten que las industrias desarrollen componentes con las garantías suficientes de interacción, funcionalidad y calidad.

1.5.1.1. Lenguaje Extensible de Etiquetas

XML no es más que un conjunto de reglas para definir etiquetas semánticas que nos organizan un documento en diferentes partes. Es lo que se conoce como un metalenguaje usado en servicios web para especificar otros lenguajes de etiquetas estructurados y protocolos necesarios. Es un lenguaje que permite jerarquizar y estructurar la información y describir los contenidos dentro del propio documento, así como la reutilización de partes del mismo.

La W3C lo define como: "un lenguaje con una importante función en el proceso de intercambio, estructuración y envío de datos en la web. Describe los datos de tal manera que es posible estructurarlos utilizando para ello etiquetas, como lo hace HTML pero que no están predefinidas, delimitando de esta manera los datos, a la vez que favoreciendo la interoperabilidad de los mismos" [13].

El formato XML es muy parecido al del HTML, lo que no significa que sea una extensión ni un componente de este; permite a los desarrolladores crear sus propias etiquetas (*tags*) para habilitar definiciones, transmisiones, validaciones, e interpretación de los datos entre aplicaciones y entre organizaciones [14].

1.5.1.2. Protocolo Simple de Acceso a Objetos

SOAP es un formato de mensajes común y extensible que describe como se organiza en forma de documentos XML, la información a intercambiar.

EL protocolo SOAP tiene como objetivo, especificar como organizar la información de forma estructurada usando XML, para que sea intercambiada entre los extremos de la invocación, básicamente se usa para recoger la información de los mensajes de petición y respuesta de los servicios web que se envían a través de una red, en un entorno distribuido y descentralizado. Los mensajes SOAP pueden ser utilizados independientemente del sistema operativo y los protocolos que se utilicen, los mismos pueden ser transportados usando diversidad de protocolos de Internet, incluyendo SMTP y HTTP.

SOAP define precisamente como realizar la codificación de las llamadas a los métodos de un servicio web, y como debe el mismo codificar el resultado para que pueda ser interpretado [12].

1.5.1.3. Descripción Universal, Descubrimiento e Integración

El estándar UDDI provee un mecanismo para que los negocios se "describan" así mismos y los tipos de servicios que proporcionan, y luego se puedan registrar y publicar en un registro UDDI. Por lo que el mismo tiene como objetivo el de ser accedido por los mensajes SOAP y dar paso a documentos WSDL, en los que se describen los requisitos del protocolo y los formatos del mensaje solicitado para interactuar con los servicios web del catálogo de registros. La especificación UDDI define un método estándar para la publicación y el descubrimiento de los componentes de software basados en la red de una arquitectura orientada a servicios" [15].

Por tanto UDDI es una especificación para un registro distribuido de información acerca de los servicios web. Define la forma en la cual se publica y descubre información acerca de estos.

1.5.1.4. Lenguaje de Descripción de Servicios Web

WSDL es el lenguaje común utilizado para la descripción de los servicios web. Es un lenguaje basado en XML que describe totalmente la forma en la cual los clientes externos pueden interactuar con los servicios web existentes en una máquina dada, los métodos que soportan y la sintaxis de los protocolos de comunicación (HTTP, SOAP) [9].

Según el consorcio W3C, encargado de la definición del estándar WSDL, lo define como: "un formato XML para la descripción de servicios de red, o un conjunto de puntos finales operando sobre mensajes que contienen información orientada a documentos u orientada a procesos" [29].

Tomando en cuenta las definiciones anteriores se puede decir que WSDL es el equivalente de un resumen en XML, describiendo los servicios web, donde se ubican, y como se pueden invocar. Por cada servicio web tendremos un documento wsdl que nos describe el servicio.

1.5.2. Necesidad de usar los Servicios Web

Los servicios web son la revolución informática de la nueva generación de aplicaciones que trabajan colaborativamente, en los cuales el software está distribuido en diferentes servidores.

A continuación se citan los beneficios más importantes de usar los servicios web [9]:

1. **promueven la interoperabilidad:** la interacción entre un proveedor y un solicitante de servicio está diseñada para que sea completamente independiente de la

plataforma y el lenguaje. Esta interacción requiere un documento WSDL para definir la interfaz y describir el servicio, junto con un protocolo de red.

2. **permiten la integración “justo-a-tiempo”**: el proceso de descubrimiento se ejecuta dinámicamente, a medida que los solicitantes de servicios utilizan a los agentes para encontrar proveedores de servicios. Una vez que el solicitante y el proveedor de servicio se han ubicado, se utiliza el documento WSDL del proveedor para enlazar al solicitante con el servicio. Esto significa que los solicitantes, los proveedores y los agentes actúan en conjunto para crear sistemas que son auto-configurables, de fácil adaptación y robustos.
3. **reducen la complejidad por medio del encapsulamiento**: los solicitantes y los proveedores del servicio se preocupan por las interfaces necesarias para interactuar. Como resultado, un solicitante de servicio no sabe como fue implementado el servicio por parte del proveedor, y este a su vez, no sabe como utiliza el cliente el servicio. Estos detalles se encapsulan en los solicitantes y proveedores.
4. **dan una “nueva vida” a las aplicaciones de legado**: es relativamente correcto tomar una aplicación, generar un wrapper SOAP, y luego generar un documento WSDL para moldear la aplicación como un servicio web.
5. **abren la puerta a nuevas oportunidades de negocio**: los servicios web facilitan la interacción con socios de negocios, al poder compartir servicios internos con un alto grado de integración.
6. **disminuyen el tiempo de desarrollo de las aplicaciones**: gracias a la filosofía de orientación a objetos utilizada, el desarrollo se convierte más bien en una labor de composición.

1.5.3. Ventajas de los Servicios Web

Los servicios web traen consigo grandes ventajas, las cuales según Santiago Márquez son [16]:

- Aumentan la interoperabilidad entre aplicaciones de software, independientemente de la plataforma en donde estén instalados o de sus propiedades.
- Fomentan los estándares y protocolos basados en texto, haciendo más fácil acceder y entender su contenido y funcionamiento.
- Al emplear HTTP, pueden utilizar un sistema firewall sin cambiar las reglas de filtrado.

- Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos, puedan ser combinados fácilmente para proveer servicios integrados.
- Al apoyarse en HTTP, pueden esquivar medidas de seguridad basadas en firewall cuyas reglas tratan de bloquear la comunicación entre programas.

1.5.4. Desventajas de los Servicios Web

Como es cierto que los servicios web tienen grandes ventajas que resultan incuestionables, los mismos también traen consigo cierta cantidad de desventajas, las más significativas e importantes son [16]:

- No son tan desarrollados para realizar transacciones comparadas a otros sistemas como CORBA⁸ (del inglés, *Common Object Request Broker Architecture*).
- Su rendimiento es bajo comparado con otros modelos de computación distribuida como es el caso de los sistemas CORBA, especialmente por el uso de protocolos y estándares basados en texto. Y es que entre los objetivos de XML no se encuentra la eficacia del procesamiento.
- Dependencia de la disponibilidad de servidores y comunicaciones.

1.6 Metodología de Desarrollo

Dada la necesidad de aumentar la calidad del software, la eficacia y eficiencia de su proceso de desarrollo, surgen las metodologías de desarrollo de software. Las mismas son un conjunto de herramientas, técnicas y procesos que facilitan el desarrollo de productos de software.

Las metodologías de desarrollo de software tienen como objetivo, presentar un conjunto de técnicas tradicionales y modernas de modelado de sistemas, que permitan desarrollar software de calidad, incluyendo heurísticas de construcción y criterios de comparación de modelos de sistemas. Las más utilizadas son: Programación Extrema (XP del inglés, *Extreme Programming*) y el Proceso Racional Unificado (RUP del inglés, *Rational Unified Process*).

1.6.1. Proceso Racional Unificado, RUP

Metodología de desarrollo de software la cual está guiada por casos de uso y centrada en la arquitectura, iterativa e incremental, utiliza el Lenguaje Unificado de Modelado (UML del inglés, *Unified Modeling Language*), lo que constituye la metodología estándar más utilizada para el análisis, implementación y documentación de sistemas orientados a objetos. RUP es un conjunto de metodologías adaptables al contexto y necesidades de cada organización.

⁸ Estándar definido por Object Management Group (OMG, que facilita el desarrollo de aplicaciones distribuidas en entornos heterogéneos).

Está basado en componentes, lo que quiere decir, que el sistema de software en construcción está formado por componentes de software interconectados a través de interfaces [17].

Pressman la define como: “la metodología de desarrollo más popular y una de las más utilizadas a nivel mundial en grandes proyectos por su robustez. Muchas metodologías de surgimiento posterior, son derivadas o adaptaciones de esta metodología tradicional. RUP es el resultado de la evolución e integración de diferentes metodologías de desarrollo de software. Permite sacar el máximo provecho de los conceptos asociados a la orientación a objetos y al modelado visual. Cuenta con las mejoras prácticas del modelo de desarrollo de un software en particular” [18].

1.6.2. Extreme Programming, XP

La metodología de desarrollo de software XP es una estrategia de desarrollo de software creada hace aproximadamente unos diez años.

La metodología XP presenta varias características como:

Retroalimentación con el cliente: conceptualmente, al menos uno de los miembros del equipo de trabajo del proyecto, es un cliente. Esto propicia una constante interacción del mismo con el producto en desarrollo.

Cortas iteraciones: en cada iteración, se obtiene un producto listo para entregar y que tiene valor para el cliente. La entrega continua de resultados compromete a ambas partes en la evolución del proyecto e indirectamente influye de forma positiva en la calidad del producto final.

Muy flexible a cambios: la constante retroalimentación con los clientes permite prever futuros cambios y evita llegar a momentos que paralizan el desarrollo del producto.

La metodología XP define su ciclo de desarrollo con los siguientes pasos:

1. El cliente define el valor de negocio a implementar.
2. El programador estima el esfuerzo necesario para su implementación.
3. El cliente selecciona qué construir, de acuerdo con sus prioridades y las restricciones de tiempo.
4. El programador construye ese valor de negocio.
5. Vuelve al paso 1.

Cada iteración del ciclo de vida incluye las siguientes fases:

- exploración,
- planificación de la entrega,
- iteraciones,
- producción,

- mantenimiento,
- muerte del proyecto.

Consideraciones de la metodología de desarrollo de software

Luego del estudio realizado sobre las metodologías de desarrollo más difundidas a nivel internacional se decidió utilizar la metodología XP, por ser la metodología adecuada al marco de la solución propuesta teniendo en cuenta los siguientes factores:

- Se requiere un corto plazo de tiempo para la culminación del producto, como máximo 6 meses.
- El equipo de trabajo es pequeño. En este caso de un solo desarrollador y el cliente.
- El cliente conoce ampliamente el funcionamiento del proceso del negocio.
- El cliente forma parte activa del equipo de desarrollo.
- Permite la creación de servicios web robustos y de alta calidad.

1.7 Tecnologías y herramientas a utilizar

1.7.1. Lenguaje de Programación

Un lenguaje de programación, es un lenguaje diseñado para describir el conjunto de acciones consecutivas que un equipo debe ejecutar. En la actualidad existen diversos lenguajes de programación para desarrollar en la web, estos han ido surgiendo debido a las tendencias y necesidades de las plataformas. Para la programación de los servicios web existen dos lenguajes de programación que son los más utilizados estos son PHP y Java.

1.7.1.1. PHP, versión 5.4

El lenguaje PHP es un lenguaje que se acopla al HTML (páginas web) para definir procedimientos que ha de realizar el servidor web. PHP soporta el trabajo con Interfaz de Entrada Común (CGI, en inglés *Common Gateway Interface*)⁹, permite procesar la información de formularios, generar páginas con contenidos dinámicos, y el trabajo con cookies. Es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles y funciones. El resultado es normalmente una página HTML. Permite la posibilidad de ejecutarlo en diferentes tipos de servidores, entre ellos Apache. Además incorpora un gran número de extensiones, aumentando las potencialidades del lenguaje. Con una sintaxis similar a los lenguajes de programación C, Java y Perl, este lenguaje hace

⁹ Tecnología de la World Wide Web que permite a un cliente solicitar datos de un programa ejecutado en un servidor web.

posible el desarrollo de páginas web que se generen dinámicamente y de forma rápida, y que pueden ejecutarse en distintos tipos de servidores web [19].

Una de las ventajas de este lenguaje es que tiene la capacidad de conexión con la mayoría de los manejadores de base de datos como: MySQL, PostgreSQL, Oracle, MS SQL Server, entre otras, además de que es un lenguaje multiplataforma.

PHP se caracteriza por ser un lenguaje muy rápido y fácil de aprender, que ofrece una solución simple y universal para las paginaciones dinámicas de la web de fácil programación.

1.7.1.2. Java

Java es un lenguaje de programación desarrollado por la compañía Sun Microsystems. En la actualidad es un lenguaje muy extendido y cada vez cobra más importancia tanto en el ámbito del Internet como en la informática en general.

Una de las ventajas de Java es que es un lenguaje de programación de distribución libre, no es necesario pagar una licencia para poder comenzar a desarrollar en él. Así mismo es un lenguaje muy completo y poderoso, pues posee librerías y utilidades muy completas que facilitan la programación.

Consideraciones de los lenguajes de programación

Después de un estudio realizado sobre los lenguajes de programación anteriormente mencionados y analizar las características y ventajas de cada uno, se decidió utilizar PHP, en su versión 5.4. Principalmente por ser un lenguaje que posee un conjunto de funciones que permiten la construcción de servicios web.

1.7.2. Herramientas a utilizar

1.7.2.1. Apache HTTP Server, versión 2.4

El servidor HTTP Apache es un servidor web HTTP de código abierto y multiplataforma. La arquitectura del servidor Apache es modular. El servidor consta de diversos módulos que aportan muchas de las funcionalidades que podrían considerarse básicas para un servidor web.

Está diseñado para ser un servidor web potente y flexible, que pueda funcionar en la más amplia variedad de plataformas y entornos. Las diferentes plataformas y entornos, hacen que a menudo sean necesarias diversas características o funcionalidades.

Apache es usado principalmente para enviar páginas web estáticas y dinámicas en la *World Wide Web (WWW)*. Muchas aplicaciones web están diseñadas asumiendo como ambiente de implantación a Apache, o que utilizarán características propias de este servidor web.

También es usado para muchas otras tareas donde el contenido necesita ser puesto a disposición de una forma segura y confiable.

Características

Apache es un servidor web flexible, rápido y eficiente, continuamente actualizado y adaptado a los nuevos protocolos. Entre sus características se destacan:

- Multiplataforma.
- Modular: puede ser adaptado a diferentes entornos y necesidades, con los diferentes módulos de apoyo que proporciona, y con la API de programación de módulos, para el desarrollo de módulos específicos.
- Basado en hilos en la versión 2.0.
- Incentiva la realimentación de los usuarios, obteniendo nuevas ideas, informes de fallos y parches para la solución de los mismos.
- Se desarrolla de forma abierta.
- Extensible: gracias a ser modular se han desarrollado diversas extensiones entre las que destaca PHP, un lenguaje de programación del lado del servidor.

1.7.2.2. Visual Paradigm, versión 8.0

Visual Paradigm es una herramienta de Ingeniería de Software Asistida por Computación (CASE del inglés, *Computer Aided Engineering*), que propicia un conjunto de ayudas para el desarrollo de programas informáticos, desde la planificación, pasando por el análisis y el diseño, hasta la generación del código fuente de los programas y la documentación.

Visual Paradigm surge con la idea de guiar el ciclo de vida completo del proceso de desarrollo del software a través de la representación de todo tipo de diagramas. Se caracteriza por la disponibilidad que tiene en diversos sistemas operativos como es el caso de Windows y Linux. El mismo utiliza un lenguaje estándar común a todo el equipo de desarrollo que facilita la comunicación entre ellos. Visual Paradigm se basa en un diseño centrado en casos de uso y enfocado al negocio que trae consigo generar un software de mayor calidad.

1.7.2.3. DbDesigner, versión 4.0

DbDesigner es una aplicación para el diseño visual de bases de datos, que combina características y funciones profesionales con un diseño simple, claro y fácil de usar, a fin de ofrecer un método efectivo para gestionar bases de datos. Es capaz de trabajar con MySQL, Oracle, MSSQL y cualquier estándar de acceso a las base de datos.

DbDesigner permite guardar el proyecto en su formato original (XML) para mantener toda la información. Permite administrar la base de datos, diseñar tablas, hacer peticiones SQL (del inglés, *Structured Query Language*), manuales y mucho más. El mismo puede conectarse

directamente a una base de datos de *backend* y construir una base de datos allí a partir del diseño, como también puede importar a partir de bases de datos existentes.

1.7.3. Framework de desarrollo

Un *framework* puede considerarse como una estructura de software, compuesta de componentes personalizables e intercambiables para el desarrollo de una aplicación.

Básicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado para ayudar a desarrollar y unir los diferentes componentes de un proyecto.

1.7.3.1. Framework Yii, versión 1.10

Yii es un *framework* para PHP, de código abierto, utilizado para el desarrollo de todo tipo de aplicaciones web. Debido a que es ligero y está equipado con sofisticados mecanismos de cacheo, es especialmente adecuado para el desarrollo de aplicaciones de gran tráfico, tales como: portales, foros, sistemas de gestión de contenidos (en inglés, *Content Management System*, CMS), sistemas de comercio electrónico, entre otros [21]. Yii es un *framework* totalmente basado en Programación Orientada a Objetos (POO, del inglés *Object-Oriented Programming*). Se basa en el patrón de diseño Modelo-Vista-Controlador (MVC, del inglés *Model-View-Controller*). Tiene soporte de autenticación incorporado y además soporta autorización a través del control de acceso basado en roles jerárquico (RBAC).

1.7.3.2. NuSOAP

NuSOAP es un conjunto de herramientas para desarrollar servicios web bajo el lenguaje PHP, no es el único soporte de servicios web en PHP que existe. Está compuesto por una serie de clases que hacen mucho más fácil el uso de las librerías para el desarrollo de servicios web. Provee soporte para el desarrollo de clientes y servidores. NuSOAP está basado en SOAP 1.1, WSDL 1.1 y HTTP 1.0/1.1. NUSOAP ofrece tres características fundamentales que la hacen ideal para el desarrollo de servidores, en el lenguaje PHP [23]:

1. Está en una fase madura de desarrollo.
2. No necesita módulos adicionales.
3. Es muy fácil su instalación y uso.

1.7.3.3. Extensión de SOAP para PHP

La extensión SOAP recientemente fue agregada dentro de PHP5. Su soporte nativo implica una mejora en la eficiencia y la rapidez. La extensión SOAP puede ser utilizada para escribir servidores y clientes SOAP. Es compatible con los subconjuntos de SOAP 1.1, SOAP 1.2 y WSDL 1.1. Posee ventajas en cuanto a su fácil uso y no requiere de la instalación de módulos adicionales para su utilización.

Consideraciones de los frameworks

Analizando las características del *framework* Yii, se decide utilizar este para el desarrollo de la capa de servicios web, ya que posee un alto rendimiento, permitiendo la máxima reusabilidad en la programación web y acelerando el proceso de desarrollo de software. Además, se decide usar la extensión SOAP ya que no necesita ningún tipo de complemento adicional, solo utilizar la versión 5 de PHP. Por lo que se propone utilizar esta extensión como herramienta para exponer y consumir servicios web.

1.7.4. IDEs

Un entorno de desarrollo integrado o entorno de desarrollo interactivo, es una aplicación de software que proporciona servicios integrales para los programadores informáticos para el desarrollo de software. Un IDE normalmente consiste en un editor de código, un compilador, un constructor de interfaz gráfica y un depurador.

Uno de los objetivos del IDE, es reducir la configuración necesaria para reconstruir múltiples utilidades de desarrollo, en lugar de proporcionar el mismo conjunto de capacidades como una unidad cohesiva.

1.7.4.1. NetBeans, versión 7.0

NetBeans es una herramienta para programadores pensada para escribir, compilar, depurar y ejecutar programas. Está escrito en Java, pero puede servir para cualquier otro lenguaje de programación. NetBeans es un producto libre y gratuito sin restricciones de uso. La misma es capaz de generar el código necesario para la creación, tanto de servicios web como de clientes de servicios web para Java y PHP.

1.7.4.2. Zend Studio, versión 8.0

Zend Studio o *Zend Development Environment*, es una herramienta propietaria que permite el desarrollo de aplicaciones web mediante PHP. Está escrito en Java, y está disponible para las plataformas Microsoft Windows, Mac OS X y GNU/Linux. Junto con su contraparte *Zend Platform*, son la propuesta de *Zend Technologies* para el desarrollo de aplicaciones web utilizando PHP, actuando *Zend Studio* como la parte cliente y *Zend Platform* como la parte servidora. Se trata en ambos casos de software comercial, lo cual contrasta con el hecho de que PHP es software libre. Permite generar el WSDL correspondiente a una clase o función escrita en PHP, basándose en elementos de configuración previamente definidos por el desarrollador, como la localización del WSDL, convención de nombres y opciones de enlaces, entre otros.

1.7.4.3. PhpStorm, versión 7.0

PhpStorm proporciona un editor para PHP, HTML y JavaScript con el análisis de código sobre la marcha, la prevención de errores y refactorizaciones automáticas para PHP y JavaScript. PhpStorm se basa en IntelliJ IDEA¹⁰, que está escrito en Java. Los usuarios pueden ampliar el IDE mediante la instalación de *plugins* creados para la plataforma IntelliJ o escribir sus propios *plugins*.

Consideraciones de los IDEs

Luego de analizados los entornos de desarrollo integrados se concluye que el más conveniente para implementar la propuesta es el PhpStorm, pues además de estar entre los más potentes para el desarrollo de aplicaciones web, adquirirlo reportaría beneficios y un costo mínimo al ser de código abierto.

1.8 Conclusiones parciales del capítulo

Después de ser analizadas las técnicas de integración de aplicaciones, se llega a la conclusión que la técnica de servicios web es la más factible para dar solución al problema planteado. Así mismo como metodología de desarrollo se seleccionó la metodología XP. Para implementar la aplicación se decidió utilizar PHP 5, por las funciones que presenta para la creación de servicios web y la extensión de SOAP para PHP que permite crear servicios web con mayor rapidez y eficacia. También se eligió PhpStorm 7.0 como IDE, por ser de código abierto y brindar las herramientas para crear servicios web.

¹⁰ Entorno de desarrollo Java creado por Jet Brains, existen dos distribuciones: Community Edition (open source) y Ultimate (comercial).

CAPÍTULO 2: DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

2.1. Introducción

El proyecto DISEM necesita de un servicio web, para lograr la interoperabilidad entre las aplicaciones que se desarrollan en el mismo, debido a que las mismas se encuentran desarrolladas en lenguajes y para plataformas diferentes. Para dar solución a esta problemática, se realiza en el presente capítulo, una descripción general de los requisitos de servicios web, especificados mediante historias de usuarios y se realiza una descripción acerca de cómo implementar los servicios web utilizando el *framework* Yii.

Como solución al problema planteado, se propone desarrollar una capa de servicios web para la integración de las aplicaciones desarrolladas en el proyecto DISEM. Esta capa permite a los sistemas, que puedan obtener información acerca de los modelos almacenados en la base de datos del proyecto, sin tomar en cuenta la tecnología o plataforma de implementación empleada.

La capa de servicios web debe servir como interfaz de comunicación entre todas las aplicaciones, brindando la posibilidad de acceso a una serie de operaciones, las cuales serán de lectura de información.

2.2. Descripción de los procesos fundamentales del Centro de Contenido

El Centro de Contenido es la aplicación encargada de gestionar las piezas estandarizadas por las diferentes normas internacionales como la ISO, ANSI, DIN, entre otras. El componente básico del Centro de Contenido es una familia de piezas, que se encuentra estructurado mediante categorías, subcategorías y piezas.

Categoría: es una agrupación lógica de los tipos de piezas, la misma tiene como atributos el nombre de la categoría, la descripción de la categoría y el lenguaje que puede ser español o inglés. Una categoría tiene una o varias subcategorías.

Subcategoría: una subcategoría tiene como atributos nombre de la subcategoría, descripción de la subcategoría y el lenguaje que puede ser español o inglés. Una subcategoría tiene una o varias familias de piezas.

Familia de piezas: las familias se organizan en categorías. Una familia de piezas contiene miembros que tienen las mismas propiedades de familia, y que representan variaciones de tamaño u operación, y a una familia pertenecen una o varias piezas. Los archivos paramétricos, textos de descripción y vista preliminar de imágenes, son comunes en todos los tamaños de una familia de piezas del Centro de Contenido.

Piezas: una pieza tiene diferentes características de diferentes tipos de datos, como son el nombre de la pieza, la descripción de la pieza y la organización estándar a la que pertenece.

El Centro de Contenido cuenta con una aplicación cliente para la interacción con la base de datos para las familias de piezas de la norma ISO; la misma se encuentra alojada en el servidor de Bases de Datos PostgreSQL.

2.3. Descripción de la arquitectura utilizada

El patrón MVC, es una arquitectura de diseño de software para separar los componentes de aplicación en tres niveles, interfaz de usuario, lógica de control y lógica de negocio.

El modelo es el objeto que representa los datos del programa, maneja los datos y controla todas sus transformaciones, es decir, es quien contiene la lógica del negocio de la aplicación.

La vista es el objeto que maneja la presentación visual de los datos representados por el modelo, genera una representación visual del modelo y muestra los datos al usuario. El controlador es el objeto que recibe e interpreta la interacción del usuario, actuando sobre los datos representados por el modelo y sobre la vista para provocar cambios de estado en la representación interna de los datos, así como en su visualización [26].

En la Figura 2 se muestra el diagrama que describe la arquitectura del patrón MVC de la capa de servicios web.

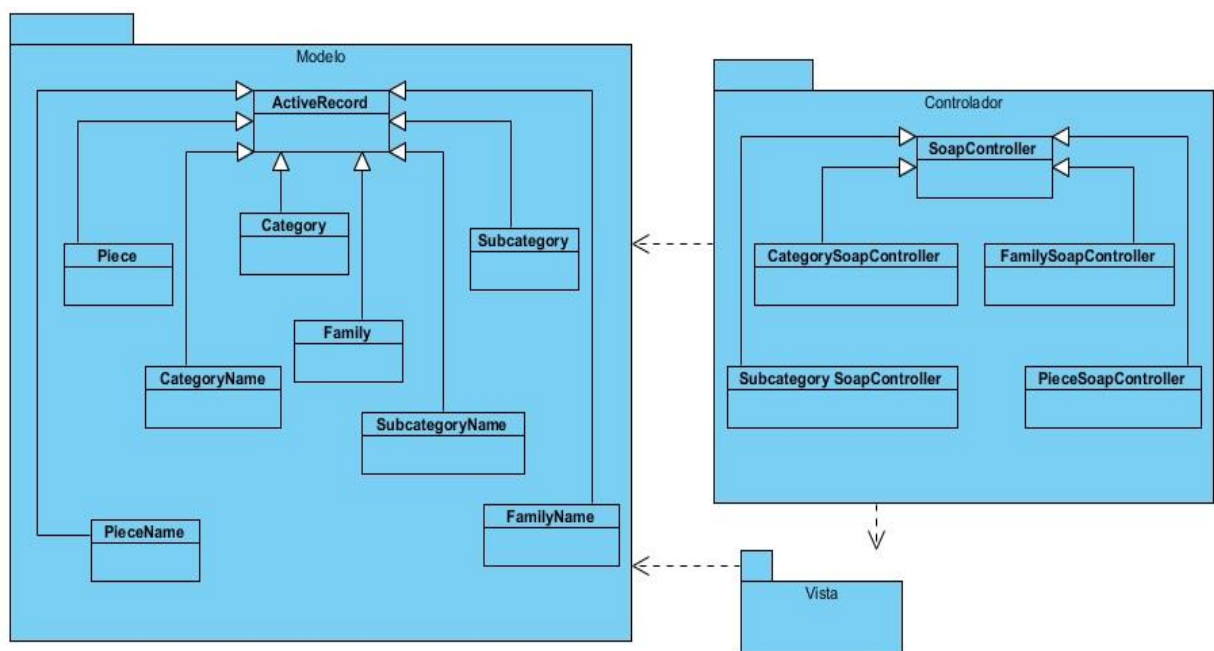


Figura 2 Diagrama de la Arquitectura Modelo-Vista-Controlador de la capa de servicios web.

En este diagrama, el cual representa la arquitectura de la capa de servicios web no se trabaja en la capa de interfaz de usuario, puesto que no se hace necesario para el desarrollo de los servicios web.

2.4. Implementación de Servicios Web con el framework Yii

Un servicio web es un sistema de software, diseñado para soportar la interacción interoperable de máquina a máquina en una red. En el contexto de aplicaciones web, usualmente se refiere a un conjunto de Interfaces de Programa de Aplicación, (APIs del inglés, *Application Program Interface*) que se pueden acceder a través de Internet y ejecutar en un sistema remoto que aloja el servicio solicitado.

El *framework* Yii proporciona una clase *CWebService* que encapsula *SoapServer* y proporciona un servicio basado en WSDL y una clase *CWebServiceAction* que implementa una acción que ofrece servicios web, logrando simplificar el trabajo de la aplicación del servicio web en una aplicación web. Las APIs se agrupan en clases llamadas proveedores de servicios. Yii generará para cada clase una especificación WSDL que describe qué APIs están disponibles y como debe ser invocada por el cliente. Cuando un cliente invoca una API, Yii creará una instancia del proveedor correspondiente y llamará a la API solicitada para completar la solicitud [25].

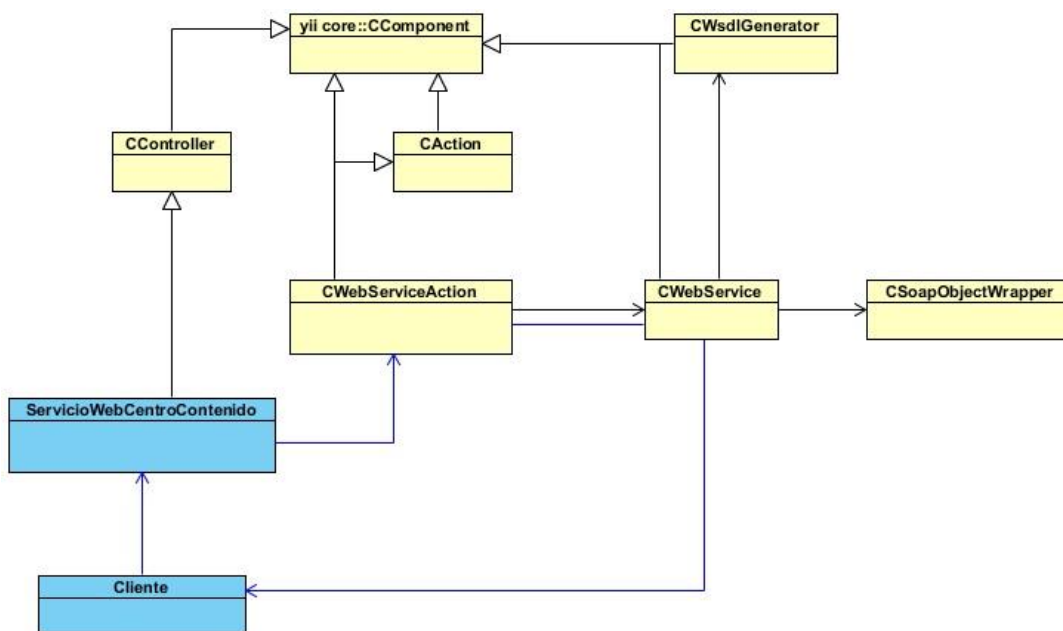


Figura 3 Arquitectura de Yii para Servicios Web.

En la Figura 3 la clase Cliente es quien representa las aplicaciones del proyecto DISEM, las cuales hacen la petición a la capa de servicios web. ServicioWebCentroContenido es quien agrupa a las clases que constituyen la capa de servicios web. Las clases restantes son las que propone el *framework* Yii.

Yii depende de los *doc comment* y de la clase *reflection* para identificar los métodos que se pueden invocar remotamente y cuáles son sus parámetros y valor de retorno.

Cada método remoto alojado en una clase proveedora, basada en la clase *CController* de Yii, deberá tener la etiqueta *@soap* como se muestra en la Figura 4, para que el componente de Yii encargado en proveer el servicio web sepa cuáles serán los métodos a exportar y cuáles serán sus argumentos y tipo de retorno usando las anotaciones *@param* y *@return*

```
class CategorySoapController extends SoapController {
    /**
     * Retorna los datos de todas las categorías
     * @param string language
     * @return array Arreglo con los datos de las categorías o en caso de error (status, mensaje)
     * @soap
     */
    public function categoryList($language)
    {
        try
        {
            $criteria = new CDbCriteria(array('order'=>'name'));
            $criteria->compare('language',$language, false);

            $result = CategoryName::model()->findAll($criteria);

            if(count($result) == 0)
            {
                return array('status'=>'error', 'message'=>'No existen categorías para este lenguaje');
            }
            else
            {
                return $result;
            }
        }
        catch (Exception $exc)
        {
            return array('status'=>'error', 'message'=>$exc->__toString());
        }
    }
}
```

Figura 4 Ejemplo de un método SOAP disponible en un Controlador de Yii.

Teniendo definido el proveedor de servicios se debe poner a disposición de los clientes. Por lo que se crea un "action" especial denominado *CWebServiceAction* que se encuentra alojado en una clase *Controller*. Este *action* especial al ser consultado, devolverá un esquema XML, el cual puede ser usado para saber qué se está exportando en ese servicio web. Por tanto ese esquema XML es el que será utilizado por el cliente PHP SOAP para saber como comunicarse con el servicio web.

Luego de creado el servicio web se crea un cliente, que consume de dicho servicio, al cual se accede mediante una URL, como se evidencia en la Figura 5, la cual dará el acceso al mecanismo del servicio web de la aplicación, esta URL que se provee, es la dirección del *action* creado anteriormente, el cual mostrará el contenido XML, que es realmente el WSDL del servicio web que ha sido definido.

```
public function actionGetCategory()
{
    $this->wsdlUrl = 'http://localhost/web-services-tesis/index.php?r=categorySoap/ws';
    ...
}
```

Figura 5 Ejemplo de acceso a un Servicio Web.

2.5. Análisis y requerimientos de los Servicios Web

2.5.1. Listado de requerimientos

Entre los elementos más importantes de un proceso de desarrollo se encuentran los requerimientos, los cuales proporcionan la comunicación entre el equipo de desarrollo y los usuarios finales, con el propósito de alcanzar un entendimiento claro de lo que se debe realizar.

R1. Listar Categorías: la capa de servicios web debe permitir mostrar el listado de las categorías de las piezas con su descripción.

R2. Listar Subcategorías dada una Categoría: la capa de servicios web debe permitir seleccionar una categoría y mostrar el nombre y la descripción de las subcategorías pertenecientes a la categoría seleccionada.

R3. Listar Subcategorías: la capa de servicios web debe permitir mostrar el nombre de todas las subcategorías con su descripción.

R4. Listar Familias de Piezas dada un Subcategoría: la capa de servicios web debe permitir mostrar los datos de todas las familias de piezas pertenecientes a una subcategoría seleccionada.

R5. Listar Familias de Piezas: la capa de servicios web debe permitir mostrar el nombre de las familias de piezas y la descripción de la misma.

R6. Listar Piezas dada una Familia: la capa de servicios web debe permitir seleccionar una familia de piezas y mostrar el nombre de las piezas pertenecientes a dicha familia.

R7. Listar Piezas: la capa de servicios web debe permitir mostrar el nombre de la pieza con su descripción.

R8. Detalle de la Pieza: la capa de servicios web debe permitir mostrar la información de la pieza, y los detalles de la misma.

R9. Buscar Piezas: la capa de servicios web debe permitir buscar piezas y mostrar su nombre y descripción.

2.5.2. Fase de exploración

La metodología XP se basa en un ciclo de vida, el cual consta de diversas fases, iniciando el ciclo con la fase de exploración. En la misma, el cliente define lo que necesita mediante la redacción de sencillas “historias de usuarios”, las cuales son de interés para la primera entrega del producto. Al mismo tiempo el equipo de desarrollo se familiariza con las herramientas, tecnologías y prácticas que se utilizarán en el proyecto. Esta fase toma el tiempo dependiendo del tamaño y familiaridad que tengan los programadores con la tecnología [22].

2.5.2.1. Historias de Usuario

La metodología XP, escogida para el proceso de desarrollo de la solución, utiliza las Historias de Usuario (HU) para describir y organizar las funciones que debe proporcionar el sistema, las cuales se encuentran escritas en un lenguaje no técnico. Las mismas apoyan la cooperación y la colaboración entre los miembros del equipo, ya que se crean y evolucionan a medida que el proyecto avanza.

La estructura de las HU es la siguiente:

Número: A cada HU se le asigna un número para facilitar su identificación.

Nombre: Nombre descriptivo de la HU.

Prioridad: Grado de prioridad que se le asigna a la HU en dependencia de las necesidades del cliente. Los valores que puede tomar son (Alta, Media o Baja).

Complejidad: Grado de complejidad que se le asigna a la HU luego de ser analizada. (Alta, Media o Baja).

Estimación: Unidades de tiempo estimadas por el equipo de desarrollo para darle cumplimiento a la HU.

Iteración: Número de la iteración en la cual será implementada la HU.

Descripción: Descripción simple sobre lo que debe hacer la funcionalidad en cuestión.

Información Adicional: Breve información que ayude a comprender algún dato no especificado antes.

Historia de Usuario		Número: 1		
Nombre:	Listar Categorías			
Prioridad: Alta	Complejidad: Media	Estimación: 3	Iteración: 1	
Descripción:				
Listar Categorías: Obtener un listado con las categorías de las piezas y su descripción.				
Listar Subcategorías dada una Categoría: Mostrar un listado con la información de las subcategorías pertenecientes a una categoría dada.				

Información Adicional:

Da cumplimiento al requisito R1 y R2.

Tabla 2.1 Descripción de la historia de usuario: Listar Categorías.

Historia de Usuario		Número: 2		
Nombre:	Listar Subcategorías			
Prioridad: Alta	Complejidad: Media	Estimación: 3	Iteración: 1	
Descripción:				
Listar Subcategorías: Obtener un listado con las subcategorías de las piezas y su descripción.				
Listar Familias de Piezas dada una Subcategoría: Mostrar un listado con los datos de las familias de piezas a partir de una subcategoría seleccionada.				
Información Adicional:				
Dar cumplimientos a los requisitos R3 y R4.				

Tabla 2.2 Descripción de la historia de usuario: Listar Subcategorías.

Historia de Usuario		Número: 3		
Nombre:	Listar Familias de Piezas			
Prioridad: Alta	Complejidad: Media	Estimación: 3	Iteración: 2	
Descripción:				
Listar Familias de Piezas: Obtener un listado con las familias de piezas y su descripción.				
Listar Piezas dada una Familia: Mostrar un listado con la información de las piezas de acuerdo a la familia de piezas a la que pertenece.				
Información Adicional:				
Da cumplimiento a los requisitos R5 y R6.				

Tabla 2.3 Descripción de la historia de usuario: Listar Familias de Piezas.

Historia de Usuario		Número: 4		
Nombre:	Listar Piezas			
Prioridad: Alta	Complejidad: Media	Estimación: 1	Iteración: 2	
Descripción:				
Listar Piezas: Obtener un listado con las piezas y su descripción en vista de iconos o en vista de lista.				
Información Adicional:				
Da cumplimiento a los requisitos R7.				

Tabla 2.4 Descripción de la historia de usuario: Listar Piezas.

Historia de Usuario		Número: 5		
Nombre:	Detalle de la Pieza			
Prioridad: Alta	Complejidad: Media	Estimación: 2	Iteración: 3	
Descripción:				
Detalle de la Pieza: Muestra la información de una pieza como: su nombre, descripción, estándar de la pieza y organización estándar a la que pertenece. Además se muestran todas las propiedades de la pieza.				
Información Adicional:				
Da cumplimiento a los requisitos R8.				

Tabla 2.5 Descripción de la historia de usuario: Detalle de la Pieza.

Historia de Usuario		Número: 6		
Nombre:	Buscar Pieza			
Prioridad: Alta	Complejidad: Media	Estimación: 1	Iteración: 3	
Descripción:				
Buscar Pieza: Permite buscar piezas y mostrar su nombre y descripción.				
Información Adicional:				
Da cumplimiento a los requisitos R9.				

Tabla 2.6 Descripción de la historia de usuario: Buscar Pieza.

2.5.3. Fase de planificación

La fase de planificación es la segunda fase que contiene el ciclo de vida de un proyecto, es una fase corta. En esta fase el cliente establece la prioridad que debe tener cada HU, y correspondientemente, los programadores realizan una estimación del esfuerzo necesario que deben realizar en cada una de ellas. En la misma se toman acuerdos sobre el contenido de la primera entrega y se determina un cronograma en conjunto con el cliente. Esta fase dura unos pocos días y una de las entregas debe realizarse en no menos de 3 meses.

2.5.3.1. Estimación del esfuerzo

Las estimaciones de esfuerzo asociado a la implementación de las HU, se establecen por los programadores utilizando como medida el punto. Un punto, equivale a una semana ideal de programación. Las historias generalmente valen de 1 a 3 puntos.

No	Historia de Usuario	Puntos de Estimación
1	Listar Categorías	3
2	Listar Subcategorías	3
3	Listar Familias de Piezas	3
4	Listar Piezas	1
5	Detalle de la Pieza	2
6	Buscar Pieza	1

Tabla 2.7 Distribución de la estimación por cada historia de usuario.

2.5.3.2. Plan de Iteraciones

Esta es la fase más importante del ciclo de desarrollo de XP, la misma incluye varias iteraciones sobre el sistema antes de ser entregado. Las funcionalidades son desarrolladas en esta fase, generando al final de cada una, un entregable funcional que implementa las HU asignadas a la iteración. Todo el trabajo de la iteración es expresado en tareas de programación, cada una de ellas es asignada a un programador como responsable, pero llevadas a cabo por parejas de programadores. Al final de la última iteración el sistema estará listo para entrar en producción.

Iteración	Orden de implementación de la HU	Duración Total
1	Listar Categorías	6 semanas
	Listar Subcategorías	
2	Listar Familias de Piezas	4 semanas
	Listar Piezas	
3	Detalle de la Pieza	3 semanas
	Buscar Piezas	

Tabla 2.8 Distribución de historias de usuario por iteración.

2.5.3.3. Plan de Entrega

El cronograma de entregas establece qué HU serán agrupadas para conformar una entrega, y el orden de las mismas, con el objetivo de que los programadores obtengan una estimación del desarrollo de las iteraciones en cuanto a tiempo, es decir, el plan de entregas se realiza en base a las estimaciones de tiempos de desarrollo realizadas por los desarrolladores. Se toma como inicio el 2 de febrero del 2015.

Iteración	Entrega
1	16 de marzo
2	13 de abril
3	4 de mayo

Tabla 2.9 Fecha de entrega por cada iteración.

2.6. Conclusiones parciales del capítulo

A partir de las especificaciones de los requisitos mediante las historias de usuarios, se identificaron diferentes funcionalidades que fueron agrupadas en varios servicios web según el contenido a devolver. Se realizó una descripción de los procesos del proyecto DISEM, logrando un mayor entendimiento de los mismos. Se pudo realizar el diseño de la propuesta de solución, permitiendo definir la arquitectura y sirviendo como antesala a su implementación.

CAPÍTULO 3: IMPLEMENTACIÓN Y PRUEBA

3.1. Introducción

En el presente capítulo se hace una descripción de las funcionalidades implementadas en la capa de servicios web, y su organización una vez desplegado el mismo. Se describe el proceso de creación de una aplicación web para el consumo de los servicios web. Se realiza un análisis de los casos de prueba, validando la aceptación de la solución propuesta, para la integración funcional entre los sistemas del proyecto DISEM.

Durante el proceso de desarrollo de los servicios web se utilizarán pautas de codificación con vistas a generar un código de alta calidad, siendo de gran importancia para la calidad de los mismos.

3.2. Implementación

En la fase de implementación, la metodología XP propone que la misma se debe hacer teniendo en cuenta los estándares de codificación ya creados. Programar bajo estándares mantiene el código consistente y facilita su comprensión y escalabilidad.

En esta fase se desglosan las HU en tareas de desarrollo, con el objetivo de facilitar y optimizar el trabajo de los desarrolladores. Durante el proceso de desarrollo de los servicios web se llevaron a cabo 3 iteraciones, permitiendo según XP, que se obtenga una versión del producto en cada iteración finalizada.

3.2.1. Diagrama de despliegue

Un diagrama de despliegue es la forma de mostrar la configuración de nodos de procesamientos en tiempo de ejecución y los componentes que en ellos residen. Estos nodos forman la topología de hardware sobre la que se ejecuta el sistema. Este diagrama se preocupa principalmente de la distribución, entrega e instalación de las partes que constituye el sistema físico [24].

El diagrama mostrado en la Figura 6, permite capturar los elementos de configuración del procesamiento de la información y las conexiones que existen entre estos elementos. Está formado por nodos, dispositivos y conectores. Los nodos son elementos de procesamiento con al menos un procesador, memoria y posiblemente otros dispositivos; los dispositivos a su vez son nodos estereotipados sin capacidad de procesamiento en el nivel de abstracción que se modela, y los conectores expresan el protocolo o conector utilizado entre los elementos del diagrama.



Figura 6 Diagrama de Despliegue.

Nodo Cliente_PC: representa las aplicaciones del proyecto DISEM. Establece relación con el servidor de aplicaciones a través del protocolo HTTP.

Nodo Servidor de Aplicaciones: en este nodo se encuentra la capa de servicios web.

Nodo Servidor de Datos: en este nodo se encuentra la base de datos que contiene toda la información de los estándares de piezas.

3.2.2. Estándar de codificación

Un estándar de codificación completo comprende todos los aspectos de la generación de código, incluyendo pautas sobre la nomenclatura de las variables, clases y paquetes; la correcta indentación del código, entre otros aspectos. La correcta elaboración y utilización de los estándares de codificación permiten a todos los desarrolladores, que realicen la implementación siguiendo las mismas pautas y así el resto de los desarrolladores puedan entender el código, facilitando así que un proyecto de software se convierta en un sistema legible, uniforme y fácil de mantener. En la Figura 7 se muestra un ejemplo del estándar de codificación utilizado en la capa de servicios web.

3.2.2.1. Conversión de nomenclatura

Se siguió el estándar de codificación utilizado por los desarrolladores del proyecto DISEM. El nombre de las variables, clases y métodos se encuentra escrito en inglés. Las variables siempre comienzan con minúscula y en caso de nombres compuestos, la primera letra de la segunda palabra comienza con mayúscula. Las clases siempre comienzan con mayúscula, en caso de nombres compuestos, las palabras que componen el nombre de la clase comienzan con mayúsculas. En las funciones se utilizó el estilo *lowerCamelCase*, donde siempre comienzan con minúscula y en caso de nombres compuestos la primera letra de cada palabra comienza con mayúscula, exceptuando la primera palabra. Los parámetros son separados por espacio, luego de la coma que los separa.

3.2.2.2. Indentación

La indentación se realiza a un **TAB**, con un equivalente a dos espacios para los bloques de instrucciones. El uso de las llaves de apertura y de cierre se realizó en una nueva línea. Se utilizó un máximo de 100 caracteres por cada línea de código. Cuando una expresión no cabe en una línea, se separa de acuerdo con los siguientes principios:

- Romper después de una coma,
- Romper antes de un operador,
- Alinear la nueva línea con el comienzo de la expresión al mismo nivel de la línea anterior.

```
public function familyListBySubCategory($language, $subCategoryId)
{
    try
    {
        $join = " INNER JOIN family fa ON (t.\"familyId\" = fa.id) "
            . "INNER JOIN subcategory sc ON (fa.\"subcategoryId\" = sc.id)";

        ...
    }
}
```

Figura 7 Ejemplo de estándar de codificación (Conversión de nomenclatura e Indentación).

3.2.2.3. Comentarios

Todo el código se encuentra comentado en idioma **inglés**, y no deben nunca incluir caracteres especiales.

3.2.2.3.1. Formatos de los comentarios en la implementación

Los comentarios de implementación se utilizan para dar descripciones de código y facilitar información adicional que no es legible en el código mismo. Los comentarios de implementación se delimitan entre `/*...*/` o se utiliza la variante `//`. Se utilizan tres formatos de comentarios: Bloque de comentario, comentario de una línea y comentario de fin de línea.

3.2.2.3.1.1. Bloque de comentario

Los bloques de comentarios fueron utilizados al inicio de cada fichero y al inicio de cada método para proveer descripciones de los mismos, como se muestra en la Figura 8.

```
/**
 * Funcion encargada de consultar el servicio web CategorySoap
 * Se consulta la funcion listCategory, que devuelve el listado de las categorias de acuerdo al idioma pasado por parametro
 */
```

Figura 8 Ejemplo de Bloque de comentario.

3.2.2.3.1.2. Comentarios de una línea

Utilizado para hacer comentarios cortos de una única línea al nivel del código al que se hace referencia.

3.2.2.3.1.3. Comentario de fin de línea

El delimitador de comentario // puede convertir en comentario, una línea completa o una parte de una línea, como se muestra en la Figura 9. No es utilizado para hacer comentarios de varias líneas consecutivas, aunque podrá utilizarse en líneas consecutivas para comentar secciones de código.

```
public function actionGetSubcategoryByCategory()
{
    //$this->wsdlUrl = 'http://localhost/web-services-tesis/index.php?r=categorySoap/ws';
    $this->wsdlUrl = 'http://'.$this->pathHost.'/web-services-tesis/index.php?r=categorySoap/ws';

    //If it is ajax validation request
    if(Yii::app()->request->isAjaxRequest) {
        $categoryId = $_POST['id'];
    }
    else
        $categoryId = $_GET['id'];
}
```

Figura 9 Ejemplo de comentarios de fin de línea.

3.2.3. Tareas de implementación

Las HU congregadas en cada iteración, se van implementando a medida que se vaya desarrollando la iteración a la cual pertenecen, por lo que al iniciar cada iteración se revisa el plan de iteraciones y se realizan las modificaciones que sean necesarias. Las HU se descomponen en tareas de desarrollo, y son asignadas a una persona o a un grupo como responsable de su implementación. Estas tareas pueden ser escritas en lenguaje técnico, ya que las mismas son para el uso estricto de los programadores.

A continuación quedan detalladas las tareas de desarrollo realizadas en cada una de las iteraciones.

Iteración 1

La primera iteración tendrá como objetivo darle cumplimiento a las HU 1 y 2. Al final se deben tener todas las funcionalidades propuestas en cada HU, teniéndose así una versión del producto para mostrar el resultado al cliente.

Historia de usuarios	Tiempo de Implementación(Semana)	
	Estimado	Real
Listar Categorías	3	3
Listar Subcategorías	3	3

Tabla 3.1 HU planificada para la 1ra Iteración.

Tareas de las historias de usuarios desarrolladas en la primera iteración

Luego de relacionar las HU pertenecientes a esta iteración, se procede a la especificación de las principales tareas de desarrollo que se realizaron para cumplir el propósito de la misma. A continuación se muestran las tareas de desarrollo correspondiente a la primera iteración.

Tarea:	
Número de la Tarea: 1	Número de la HU: 1
Nombre de la Tarea: Listar Categorías	
Tipo de Tarea: Desarrollo	
Descripción: En esta tarea se crea un servicio web que permite mostrar el nombre y la descripción de las categorías y permite además dada una categoría mostrar las subcategorías pertenecientes a dicha categoría.	

Tabla 3.2 Descripción de la tarea de desarrollo #1.

Tarea:	
Número de la Tarea: 2	Número de la HU: 2
Nombre de la Tarea: Listar Subcategorías	
Tipo de Tarea: Desarrollo	
Descripción: En esta tarea se crea un servicio web que permite mostrar el nombre y la descripción de las subcategorías, y permite también dada una subcategoría mostrar las familias de piezas pertenecientes a esta subcategoría.	

Tabla 3.3 Descripción de la tarea de desarrollo #2.

Iteración 2

La segunda iteración está centrada en desarrollar las HU 3 y 4, con el fin de obtener una segunda versión, de la capa de servicios web, con nuevas funcionalidades.

Historia de usuarios	Tiempo de Implementación(Semanal)	
	Estimado	Real
Listar Familias de Piezas	3	3
Listar Piezas	1	1

Tabla 3.4 HU planificada para la 2da Iteración.

Tareas de las historias de usuarios desarrolladas en la segunda iteración

Tarea:	
Número de la Tarea: 3	Número de la HU: 3
Nombre de la Tarea: Listar Familias de Piezas	
Tipo de Tarea: Desarrollo	
Descripción: En esta tarea se crea un servicio web que permite mostrar la información acerca de las familias de piezas, y permite además, dada una familia de piezas mostrar todas las piezas que forman parte de esta familia.	

Tabla 3.5 Descripción de la tarea de desarrollo #3.

Tarea:	
Número de la Tarea: 4	Número de la HU: 4
Nombre de la Tarea: Listar Piezas	
Tipo de Tarea: Desarrollo	
Descripción: En esta tarea se crea un servicio web que permite mostrar el nombre y la descripción de las piezas.	

Tabla 3.6 Descripción de la tarea de desarrollo #4.

Iteración 3

En esta iteración se implementarán las HU 5 y 6, con el fin de obtener una tercera versión de la capa de servicios web con nuevas funcionalidades.

Historia de usuarios	Tiempo de Implementación(Semanal)	
	Estimado	Real
Detalle de la Pieza	2	2
Buscar Pieza	1	1

Tabla 3.7 HU planificada para la 3ra Iteración.

Tareas de las historias de usuarios desarrolladas en la tercera iteración

Tarea:	
Número de la Tarea: 5	Número de la HU: 5
Nombre de la Tarea: Detalle de la Pieza	
Tipo de Tarea: Desarrollo	
Descripción: En esta tarea se crea un servicio web que permite mostrar información acerca de una pieza seleccionada.	

Tabla 3.8 Descripción de la tarea de desarrollo #5.

Tarea:	
Número de la Tarea: 6	Número de la HU: 6
Nombre de la Tarea: Buscar una Pieza	
Tipo de Tarea: Desarrollo	
Descripción: En esta tarea se crea un servicio web que permite buscar piezas y mostrar la información de ellas.	

Tabla 3.9 Descripción de la tarea de desarrollo #6.

3.3. Descripción de los servicios web

A continuación se hace una breve descripción de los servicios web implementados y la función que realizan cada uno de ellos. Se realizaron cuatro servicios web los cuales están acorde a los requisitos propuesto para la capa de servicios web.

3.3.1. Servicio Web CategorySoap

El servicio web **CategorySoap** permite listar todas las categorías que pertenecen al centro de contenido con la operación **categoryList**, y además permite listar todas las subcategorías pertenecientes a una categoría mediante la operación **subcategoryListByCategory**.

categoryList:

- Parámetros:
 1. *language*, tipo de dato *String*.
- Retorna un arreglo con los datos de las categorías, y en caso de no existir ninguna categoría se muestra un mensaje de error.

subcategoryListByCategory:

- Parámetros:
 1. *language*: tipo de dato *String*.
 2. *categoryId*: tipo de dato Entero.
- Retorna un arreglo con los datos de las subcategorías pertenecientes a la categoría pasada por parámetro, y en caso de no existir muestra un mensaje de error.

3.3.2. Servicio Web SubCategorySoap

El Servicio Web **SubCategorySoap** permite listar todas las subcategorías que pertenecen al centro de contenido con la operación **subCategoryList**, y además permite listar todas las familias de piezas pertenecientes a una subcategoría mediante la operación **familyListBySubCategory**.

subCategoryList:

- Parámetros:
 1. *language*, tipo de dato *String*.
- Retorna un arreglo con los datos de las subcategorías, y en caso de no existir ninguna subcategoría se muestra un mensaje de error.

familyListBySubCategory:

- Parámetros:
 1. *language*: tipo de dato *String*.
 2. *subCategoryId*: tipo de dato Entero.
- Retorna un arreglo con los datos de las familias de piezas pertenecientes a la subcategoría pasada por parámetro, y en caso de no existir muestra un mensaje de error.

3.3.3. Servicio Web FamilySoap

El Servicio Web **FamilySoap** permite listar todas las familias de piezas que pertenecen al centro de contenido con la operación **familyList**, y además permite listar todas las piezas pertenecientes a una familia de piezas mediante la operación **pieceListByFamily**.

familyList:

- Parámetros:
 1. *language*, tipo de dato *String*.
- Retorna un arreglo con los datos de las familias de piezas, y en caso de no existir ninguna familia de piezas se muestra un mensaje de error.

pieceListByFamily:

- Parámetros:
 1. *language*: tipo de dato *String*.
 2. *familyId*: tipo de dato Entero.
- Retorna un arreglo con los datos de las piezas pertenecientes a la familia de piezas pasada por parámetro, y en caso de no existir muestra un mensaje de error.

3.3.4. Servicio Web PieceSoap

El Servicio Web **PieceSoap** permite listar todas las piezas que pertenecen al centro de contenido con la operación **pieceList**, el mismo permite también buscar una pieza utilizando la operación **searchPiece**. Este servicio web permite además, mostrar la información acerca de una pieza en específico con la operación **getPieceFamilyDetail** y permite mostrar los detalles de una pieza seleccionada mediante la operación **getPieceDetail**.

pieceList:

- Parámetros:
 1. *language*, tipo de dato *String*.
- Retorna un arreglo con los datos de las piezas, y en caso de no existir ninguna pieza se muestra un mensaje de error.

searchPiece:

- Parámetros:
 1. *language*: tipo de dato *String*.
 2. *query*: tipo de dato *String*.

- Retorna un arreglo con los datos de las piezas que se encuentran en la búsqueda, y en caso de no existir muestra un mensaje de error.

getPieceFamilyDetail:

- Parámetros:
 1. *language*: tipo de dato *String*.
 2. *pieceId*: tipo de dato Entero.
- Retorna un arreglo con los datos de las piezas que coinciden con la búsqueda, y en caso de no existir muestra un mensaje de error.

getPieceDetail:

- Parámetros:
 1. *language*: tipo de dato *String*.
 2. *pieceId*: tipo de dato Entero.
- Retorna un arreglo con los detalles de las piezas que coinciden con la búsqueda, y en caso de no existir muestra un mensaje de error.

3.4. Aplicación web para el consumo de la capa de servicios web

Para la validación y el correcto funcionamiento de la capa de servicios web se desarrolló una aplicación web para el consumo de los servicios web implementados, la misma se creó utilizando el *framework* de desarrollo Yii. Esta aplicación web cuenta con una clase llamada *ClientController*, la cual se encarga de realizar las llamadas a los servicios web.

3.4.1. Consumo del servicio web CategorySoap

Para el cliente existen dos funciones las cuales consumen del servicio web CategorySoap:

- La función ***actionGetCategory***, encargada de consultar el servicio web para listar las categorías. (Anexo 1)
- La función ***actionGetSubcategoryByCategory***, encargada de consultar el servicio web para listar las subcategorías dada una categoría.

Operación actionGetCategory

- Espera como parámetro la url donde se encuentra publicado el servicio web que se va a consumir:
http://direccion_del_servidor/carpeta_del_servidor/index.php?r=categorySoap/ws
- Este *action* consulta la función *listCategory*, que se encarga de listar las categorías según el idioma pasado por parámetro.

Operación **actionGetSubcategoryByCategory**

- Espera como parámetro la url donde se encuentra publicado el servicio web que se va a consumir:
http://direccion_del_servidor/carpeta_del_servidor/index.php?r=categorySoap/ws
- Este *acción* consulta la función *subcategoryListByCategory*, que se encarga de listar las subcategorías según el idioma y el id de la categoría pasado por parámetro.

3.4.2. Consumo del servicio web **SubCategorySoap**

Para el cliente existen dos funciones las cuales consumen del servicio web SubCategorySoap:

- La función ***actionGetListSubcategory***, encargada de consultar el servicio web para listar las subcategorías. (Anexo 2)
- La función ***actionGetFamilyBySubCategory***, encargada de consultar el servicio web para listar las familias de piezas, dada una subcategoría.

Operación **actionGetListSubcategory**

- Espera como parámetro la url donde se encuentra publicado el servicio web que se va a consumir:
http://direccion_del_servidor/carpeta_del_servidor/index.php?r=subcategorySoap/ws
- Este *acción* consulta la función *subCategoryList* que se encarga de listar las subcategorías según el idioma pasado por parámetro.

Operación **actionGetFamilyBySubCategory**

- Espera como parámetro la url donde se encuentra publicado el servicio web que se va a consumir:
http://direccion_del_servidor/carpeta_del_servidor/index.php?r=subcategorySoap/ws
- Este *acción* consulta la función *familyListBySubCategory*, que se encarga de listar las familias de piezas según el idioma y el id de la subcategoría pasado por parámetro.

3.4.3. Consumo del servicio web **FamilySoap**

Para el cliente existen dos funciones las cuales consumen del Servicio Web FamilySoap:

- La función ***actionGetListFamily***, encargada de consultar el servicio Web para listar las familias de piezas. (Anexo 3)
- La función ***actionGetPieceByFamily***, encargada de consultar el servicio Web para listar las piezas, dada una familia de piezas.

Operación **actionGetListFamily**

- Espera como parámetro la url donde se encuentra publicado el servicio web que se va a consumir:
http://direccion_del_servidor/carpeta_del_servidor/index.php?r=familySoap/ws
- Este action consulta la función *familyList* que se encarga de listar las familias de piezas según el idioma pasado por parámetro.

Operación **actionGetPieceByFamily**

- Espera como parámetro la url donde se encuentra publicado el servicio web que se va a consumir:
http://direccion_del_servidor/carpeta_del_servidor/index.php?r=familySoap/ws
- Este action consulta la función *pieceListByFamily*, que se encarga de listar las piezas según el idioma y el id de la familia de piezas pasado por parámetro.

3.4.4. Consumo del servicio web **PieceSoap**

Para el cliente existen cuatro funciones las cuales consumen del servicio web PieceSoap

- La función ***actionGetListPiece***, encargada de consultar el servicio web para listar las piezas. (Anexo 4):
- La función ***actionSearch***, encargada de consultar el servicio web para buscar una o varias piezas. (Anexo 5)
- La función ***actionDetailFamilyPiece***, encargada de consultar el servicio web para mostrar la información de la pieza. (Anexo 6)
- La función ***actionDetailPiece***, encarga de consultar el servicio web para mostrar los detalles de una pieza. (Anexo 7)

Operación **actionGetListPiece**

- Espera como parámetro la url donde se encuentra publicado el servicio web que se va a consumir:
http://direccion_del_servidor/carpeta_del_servidor/index.php?r=pieceSoap/ws
- Este action consulta la función *pieceList* que se encarga de listar las piezas según el idioma pasado por parámetro.

Operación **actionSearch**

- Espera como parámetro la url donde se encuentra publicado el servicio web que se va a consumir:
http://direccion_del_servidor/carpeta_del_servidor/index.php?r=pieceSoap/ws

- Este action consulta la función *searchPiece*, que se encarga de listar las piezas encontradas según el idioma pasado por parámetro.

Operación actionDetailFamilyPiece

- Espera como parámetro la url donde se encuentra publicado el servicio web que se va a consumir:
http://direccion_del_servidor/carpeta_del_servidor/index.php?r=pieceSoap/ws
- Este action consulta la función *getPieceFamilyDetail*, que se encarga de mostrar la información de las piezas encontradas según el idioma y el id de la pieza pasado por parámetro.

Operación actionDetailPiece

- Espera como parámetro la url donde se encuentra publicado el servicio web que se va a consumir:
http://direccion_del_servidor/carpeta_del_servidor/index.php?r=pieceSoap/ws
- Este action consulta la función *getPieceDetail*, que se encarga de mostrar los detalles de las piezas encontradas según el idioma y el id de la pieza pasado por parámetro.

Pasos para el consumo de los servicios usando cliente desarrollado en PHP.

1. Se debe tener instalado un servidor web y en la carpeta del servidor (*htdocs* o *www*) se debe copiar la carpeta del cliente.
2. Se debe modificar la variable *error_reporting* del *php.ini*, para evitar errores o alertas a causa de las versiones del PHP. Se debe cambiar el valor de esta variable como se muestra a continuación :

```
error_reporting=E_ALL & ~E_DEPRECATED & ~E_STRICT
```

Por el siguiente valor:

```
error_reporting=E_COMPILE_ERROR|E_RECOVERABLE_ERROR|E_ERROR|E_C  
ORE_ERROR
```

```
; error_reporting=E_ALL
```

3. Se debe cambiar la dirección donde se encuentra alojado la capa de servicios web en un documento del tipo *.txt*, llamado *host*, que se encuentra en la raíz del cliente.

3.5. Validación de la propuesta

Las pruebas son un proceso de ejecución de un programa, con la intención de descubrir un error, llegándose a ver en la ingeniería de software como un paso destructivo en lugar de constructivo. La metodología XP anima a probar tanto como sea posible, permitiendo

aumentar la calidad de los sistemas, ya que reduce el número de errores no detectados y disminuye el tiempo transcurrido entre la aparición de un error y su detección.

XP divide las pruebas del sistema en dos grupos:

- **Pruebas unitarias:** encargadas de verificar el código y diseñada por los programadores.
- **Pruebas de aceptación o pruebas funcionales:** destinadas a evaluar si al final de una iteración se consiguió la funcionalidad requerida, diseñada por el cliente final.

3.5.1. Pruebas unitarias

Las pruebas unitarias son pruebas dirigidas a probar clases aisladamente y están relacionadas con el código y la responsabilidad de cada clase y sus fragmentos de códigos más críticos [27]. Las pruebas unitarias ayudan a garantizar la corrección del programa comprobando que el código de la aplicación realiza la acción esperada.

Características de las pruebas unitarias:

- Asegura calidad del código entregado. Es la mejor forma de detectar errores tempranamente en el desarrollo.
- Ayuda a definir los requerimientos y responsabilidades de cada método en cada clase probada.
- Constituye una buena forma de ejecutar pruebas de concepto.
- Permite hacer *refactoring* tempranamente en el código.
- Permite hacer pruebas de estrés tempranamente en el código.
- Permite encontrar errores o bugs tempranamente en el desarrollo.

3.5.1.1. Framework de Prueba PHPUnit

PHPUnit es una herramienta para la ejecución de pruebas al desarrollo de programas o sistemas efectuados específicamente en PHP. PHPUnit nos permite realizar las pruebas pertinentes a nuestro código, verificando que el funcionamiento de las aplicaciones PHP es el deseado, y en su caso encontrando bugs y errores que una vez solucionados mejorarán la calidad de nuestros desarrollos web [28].

Características del Framework de Prueba PHPUnit:

- Puerto completo de JUnit para PHP5.
- Soporte para Mock Objects (jMock).
- Soporte para testeo de la base de datos.
- Forma parte del grupo de *framework* de xUnit.
- Almacena los resultados en una Test DataBase.

El *framework* Yii proporciona soporte para crear y ejecutar pruebas unitarias y funcionales, basadas en PHPUnit. Por ser un *framework* basado en la arquitectura Modelo-Vista-Controlador, provee *hooks* dentro del *framework*, para la ayuda de escrituras de las pruebas. Existen diferentes aserciones que se utilizan para probar los resultados de todos los tipos de llamadas que se pueden realizar en una aplicación, para las pruebas desarrolladas en la capa de servicios web, se utilizó el *AssertEquals*, el cual permite comprobar el resultado frente a otra entrada en busca de coincidencias.

El Centro de Contenido se encuentra estructurado por categorías, donde se encuentran almacenadas cinco categorías en idioma español y cinco categorías en idioma inglés, por lo que se realiza una prueba como se muestra en la Figura 10, para comprobar que la función *categoryList*, lista las diez categorías que se encuentran en el Centro de Contenido.

```
<?php
|/** Created by PhpStorm. ...*/

require_once(dirname(__FILE__).'/../../controllers/ClientController.php');
class CategoryTest extends PHPUnit_Framework_TestCase {

    public function testOK(){
        $client = new ClientController();
        $client->init();
        $countCategory = $client->getCategoryList();
        $this->assertEquals(count($countCategory),10);
    }
}
}
```

Figura 10 Prueba Unitaria para la función *categoryList*.

Se realizaron las pruebas unitarias utilizando el *AssertEquals*, a todos los servicios desarrollados en la capa de servicios web, para comprobar el correcto funcionamiento de los mismos. En la siguiente tabla se muestran los resultados de las pruebas unitarias realizadas.

Nombre de la Prueba	Aserciones	Fallos	Errores
<i>CategoryTest</i>	1	0	0
<i>SubCategoryTest</i>	1	0	0
<i>FamilyTest</i>	1	0	0
<i>PieceTest</i>	1	0	0

Tabla 3.10 Pruebas unitarias realizadas.

3.5.2. Pruebas de aceptación

Después de implementados los servicios web, se hace necesario verificar el funcionamiento y el cumplimiento de los requisitos que satisfacen las necesidades del cliente. La metodología XP propone además de las pruebas unitarias, las pruebas de aceptación, con el objetivo de evaluar al final de cada iteración si se alcanzaron las metas propuestas por el cliente para cada funcionalidad. Las mismas son diseñadas por el equipo de desarrollo conjuntamente con el cliente, basándose en las HU.

Durante una iteración la HU seleccionada en la planificación de iteraciones se convertirá en una prueba de aceptación. Una HU puede tener una o tantas pruebas de aceptación como sean necesarias para garantizar su correcto funcionamiento, y no se considera completa hasta que no supera sus pruebas de aceptación. Esto significa que debe desarrollarse un nuevo test de aceptación para cada iteración. Una prueba de aceptación es como una prueba de caja negra. Cada una de ellas representa una salida esperada del sistema. Es responsabilidad del cliente verificar la corrección de las pruebas de aceptación y tomar decisiones acerca de las mismas.

Caso de prueba de aceptación		Número: 1	Historia de Usuario: 1
Nombre: Listar Categorías.			
Descripción: Prueba para la funcionalidad Listar Categorías, donde se debe mostrar un listado con el nombre y la descripción de las categorías.			
Condiciones de ejecución: Ninguna			
Entradas: Ninguna			
Resultados Esperados: Se retorna un listado con el nombre y la descripción de las categorías.			
Evaluación:	1ra Iteración de la prueba Prueba Satisfactoria.		

Tabla 3.11 Caso de prueba: Listar Categorías.

Caso de prueba de aceptación		Número: 2	Historia de Usuario: 2
Nombre: Listar Subcategorías.			
Descripción: Prueba para la funcionalidad Listar Subcategorías, la cual debe mostrar un listado con el nombre y la descripción de las subcategorías.			
Condiciones de ejecución: Ninguna			
Entradas: Ninguna			
Resultados Esperados: Se retorna un listado con el nombre y la descripción de las subcategorías.			
Evaluación:	1ra Iteración de la prueba Prueba Satisfactoria.		

Tabla 3.12 Caso de prueba: Listar Subcategorías.

Caso de prueba de aceptación		Número: 3	Historia de Usuario: 1
Nombre: Listar Subcategorías dada una Categoría.			
Descripción: Prueba para la funcionalidad Listar Subcategorías dada una Categoría, la cual debe mostrar un listado con el nombre y la descripción de las subcategorías que pertenecen a una categoría dada.			
Condiciones de ejecución: Ninguna			
Entradas: Identificador de la categoría que es un valor entero.			
Resultados Esperados: Se retorna un listado con el nombre y la descripción de las subcategorías que pertenecen a una categoría dada.			
Evaluación:	1ra Iteración de la prueba El sistema no devuelve ninguna subcategoría.		
	2da Iteración de la prueba Prueba Satisfactoria.		

Tabla 3.13 Caso de prueba: Listar Subcategorías dada una Categoría.

Caso de prueba de aceptación		Número: 4	Historia de Usuario: 3
Nombre: Listar Familias de Piezas			
Descripción: Prueba para la funcionalidad Listar Familias de Piezas, la cual debe mostrar un listado con el nombre y la descripción de las familias de piezas.			
Condiciones de ejecución: Ninguna			
Entradas: Ninguna			
Resultados Esperados: Se retorna un listado con el nombre y la descripción de las familias de piezas.			
Evaluación:	1ra Iteración de la prueba		
	Prueba Satisfactoria.		

Tabla 3.14 Caso de prueba: Listar Familias de Piezas.

Caso de prueba de aceptación		Número: 5	Historia de Usuario: 2
Nombre: Listar Familias de Piezas dada una Subcategoría.			
Descripción: Prueba para la funcionalidad Listar Familias de Piezas dada una Subcategoría, la cual debe mostrar un listado con el nombre y la descripción de las familias de piezas que pertenecen a una subcategoría dada.			
Condiciones de ejecución: Ninguna			
Entradas: Identificador de la Subcategoría que es un valor entero.			
Resultados Esperados: Se retorna un listado con el nombre y la descripción de las familias de piezas que pertenecen a una subcategoría dada.			
Evaluación:	1ra Iteración de la prueba		
	Prueba Satisfactoria.		

Tabla 3.15 Caso de prueba: Listar Familias de Piezas dada una Subcategoría.

Caso de prueba de aceptación		Número: 6	Historia de Usuario: 4
Nombre: Listar Piezas en Vista de Lista.			
Descripción: Prueba para la funcionalidad Listar Piezas, la cual debe mostrar un listado con el nombre y la descripción de las piezas.			
Condiciones de ejecución: Ninguna			
Entradas: Ninguna			
Resultados Esperados: Se retorna un listado con el nombre y la descripción de las piezas.			
Evaluación:	1ra Iteración de la prueba		
	Prueba Satisfactoria.		

Tabla 3.16 Caso de prueba: Listar Piezas en Vista de Lista.

Caso de prueba de aceptación		Número: 7	Historia de Usuario: 4
Nombre: Listar Piezas en Iconos.			

Descripción: Prueba para la funcionalidad Listar Piezas, la cual debe mostrar el nombre de las piezas con el icono de las mismas.	
Condiciones de ejecución: Ninguna	
Entradas: Ninguna	
Resultados Esperados: Se retorna el nombre de las piezas con el icono de la pieza.	
Evaluación:	1ra Iteración de la prueba Prueba Satisfactoria.

Tabla 3.17 Caso de prueba: Listar Piezas en Iconos.

Caso de prueba de aceptación		Número: 8	Historia de Usuario: 3
Nombre: Listar Piezas dada una Familia de Piezas.			
Descripción: Prueba para la funcionalidad Listar Piezas dada una Familia de Piezas, la cual debe mostrar un listado con el nombre de las piezas a partir de una familia de piezas dada.			
Condiciones de ejecución: Ninguna			
Entradas: Identificador de las Familias de Piezas que es un valor entero.			
Resultados Esperados: Se retorna el nombre de las piezas pertenecientes a una familia de piezas seleccionada.			
Evaluación:	1ra Iteración de la prueba		
	No se muestra el mensaje al no existir ninguna pieza.		
	2da Iteración de la prueba		
	Prueba Satisfactoria.		

Tabla 3.18 Caso de prueba: Listar Piezas dada una Familia de Piezas.

Caso de prueba de aceptación		Número: 9	Historia de Usuario: 5
Nombre: Información de la Pieza			
Descripción: Prueba para la funcionalidad Información de la Pieza, la cual debe mostrar la información de una pieza seleccionada.			
Condiciones de ejecución: Ninguna			
Entradas: Identificador de la pieza que es un valor entero			
Resultados Esperados: Se muestra la información como nombre, descripción, organización y estándar a la que pertenece la pieza.			
Evaluación:	1ra Iteración de la prueba		
	No muestra toda la información El mensaje mostrado tiene faltas de ortografía.		
	2da Iteración de la prueba		
	El mensaje mostrado tiene faltas de ortografía.		
	3ra Iteración de la prueba		
	Prueba Satisfactoria.		

Tabla 3.19 Caso de prueba: Información de la Pieza.

Caso de prueba de aceptación		Número: 10	Historia de Usuario: 5
Nombre: Detalle de la Pieza			
Descripción: Prueba para la funcionalidad Detalle de la Pieza, la cual debe mostrar una información más detallada acerca de una pieza.			
Condiciones de ejecución: Ninguna			
Entradas: Ninguna			
Resultados Esperados: Se muestran los datos de las piezas, los mismos son los atributos de las piezas con su correspondiente valor.			
Evaluación:	1ra Iteración de la prueba		
	No muestra toda la información		
	2da Iteración de la prueba		
	Prueba Satisfactoria.		

Tabla 3.20 Caso de prueba: Detalle de la Pieza.

Caso de prueba de aceptación		Número: 11	Historia de Usuario: 6
Nombre: Buscar Pieza			
Descripción: Prueba para la funcionalidad Buscar Pieza, la cual debe buscar piezas.			

Condiciones de ejecución: Ninguna	
Entradas: Identificador de la pieza, que es un valor entero.	
Resultados Esperados: Se muestran las piezas buscada tanto en vista de lista como en vista de iconos.	
Evaluación:	1ra Iteración de la prueba Prueba Satisfactoria.

Tabla 3.21 Caso de prueba: Buscar Pieza.

3.5.3. Resultados de las pruebas

Se realizaron 11 pruebas de aceptación para validar que el sistema funcionara correctamente, mostrando las salidas correspondientes a cada escenario. De estas pruebas realizadas como se muestra en la Figura 11, en una primera iteración fueron clasificadas de satisfactorio 3 pruebas de 3 que fueron realizadas. En una segunda iteración se evaluaron satisfactoriamente 5, de un total de 5 pruebas realizadas. En la tercera y última iteración se realizaron 3 pruebas y las 3 resultaron satisfactorias, constituyendo un 100% de pruebas funcionales exitosas.

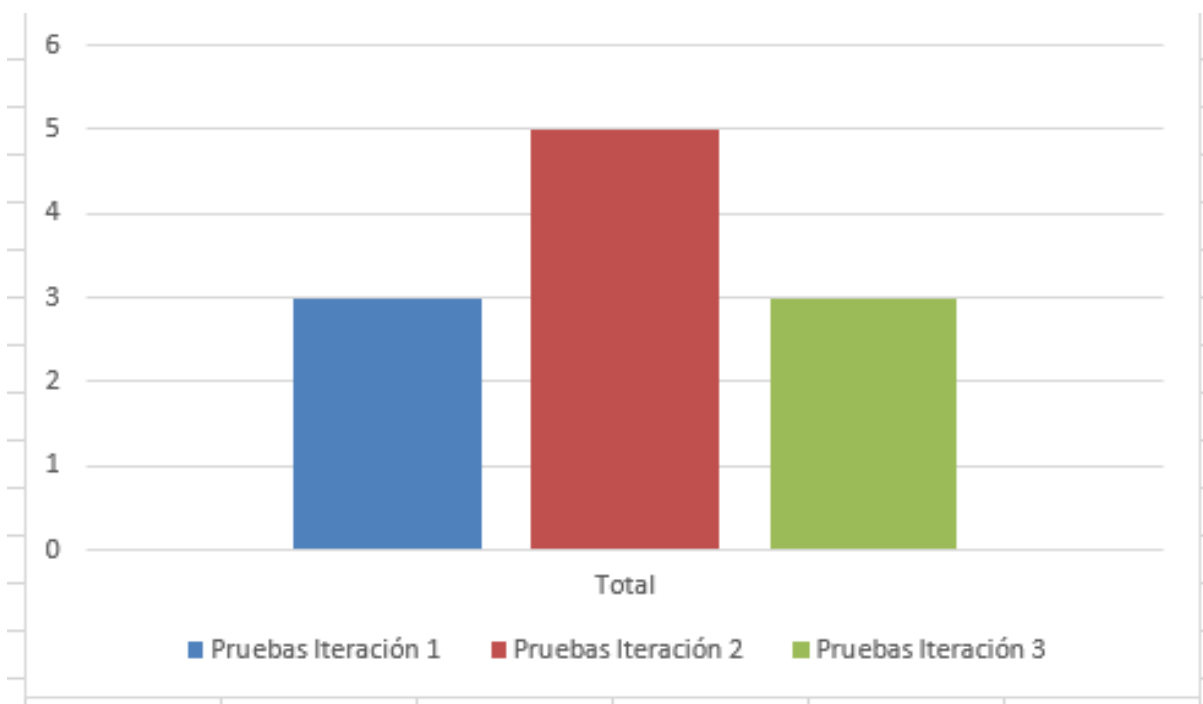


Figura 11 Pruebas de Aceptación realizadas en cada iteración.

3.6. Conclusiones parciales del capítulo

La construcción de la capa de servicios web por iteraciones, permitió obtener versiones del producto al culminar cada iteración. Se realizó la implementación de la aplicación según los

estándares de códigos especificados. Mediante las pruebas de aceptación aplicadas se pudo demostrar que el sistema se encuentra listo y cumple con los requisitos propuestos en las Historias de Usuarios. Al finalizar la última iteración se obtuvo una capa de servicios web completamente funcional que cumple con las especificaciones establecidas por el cliente.

CONCLUSIONES GENERALES

El desarrollo de la presente investigación permitió resolver el problema de la integración de las aplicaciones del proyecto DISEM y el Centro de Contenido. La implementación de la capa de servicios web, permitió que diferentes aplicaciones puedan visualizar los modelos almacenados en la base de datos, sin tener que interactuar directamente con el gestor de bases de datos.

RECOMENDACIONES

A partir de los resultados de este trabajo, se proponen las siguientes recomendaciones:

Utilizar otros estándares de SW-*, en busca de mejorar aspectos relacionados con la seguridad e interoperabilidad de los servicios.

Realizar pruebas de interoperabilidad, para comprobar el consumo de los servicios desde otras plataformas.

BIBLIOGRAFÍA

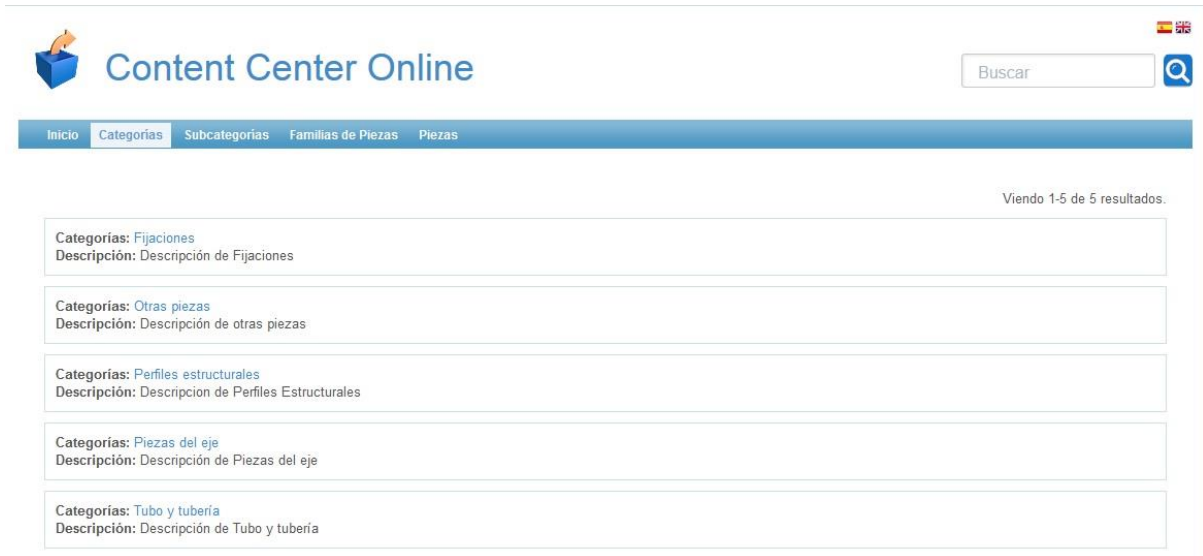
1. **Díaz Marcheco, Yerandi.** Base de Datos de Partes Estandarizadas Para Sistemas CAD Desarrollado Sobre Tecnologías Libres. Universidad de las Ciencias Informáticas, Cuba, 2014.
2. **Solórzano Ríos, José Antonio.** Desarrollo de Practicas Avanzadas para el Laboratorio de Diseño y Manufactura Asistido por Computadora. Tesis Pregrado. MÉXICO, D.F: Universidad Nacional Autónoma de México, Facultad de Ingeniería, 2012. Disponible desde: <http://132.248.52.100:8080/xmlui/bitstream/handle/132.248.52.100/1887/TESIS.pdf>
3. **Valcárcel García, Ignacio y Munilla Calvo, Eduardo.** *E-business Colaborativo.* España, Fundación Confemetal, 2003. ISBN: 84-95428-98-9.
4. **Dewit, Oliver.** *Asp.net Programación Web con Visual Studio y Web Matrix.* s.l. Ediciones ENI, 2003.
5. **Saffirio, Mario.** Tecnologías de Información y Gestión de Procesos de Negocios. [En línea] 5 de Febrero de 2006. <http://msaffirio.wordpress.com/2006/02/05/%C2%BFque-son-los-web-services/>.
6. Aner Sistemas informáticos [En línea 2005] <http://www.aner.com/software-de-gestion-empresarial/que-es-un-erp.html>
7. **de Soto, Adolfo R y Cuervo Fernández, Eva.** Nuevas Tendencias en Sistemas de Información: *Procesos y Servicios.* s.l.: Pecvnia, 2006.
8. Universidad de Toronto, Canadá, Facultad de Hans-Arno Jacobsen. [En línea 2001] <http://www.msrg.org/>
9. **Arboleda, Liliana M.** Servicios Web: Distribución e Integración. Calis, Colombia, Universidad Icesi 2004.
10. **Schmidt, J.** (2000). Enabling Next-Generation Enterprises. EaiJournal.
11. **Fernández Lino, Carlos Roberto.** Desarrollo de aplicaciones basadas en servicios aplicado a incorporar nuevas tecnologías. Universidad Mayor San Andrés. La Paz, Bolivia: Universidad Mayor San Andrés, 2009. Tesis de Grado.
12. **Galviz Portilla, Jorge Armando.** Arquitecturas Orientadas a Servicios como soporte a modelos de educación virtual módulos: académico, investigación, entorno y los servicios a estudiantes), Universidad de Pamplona, 2006.
13. **W3C.** W3C.org. <http://www.w3c.org/TR/ws-gloss/>.

14. **Mustelier Blanco, Alberto.** Solución de Software para la integración funcional del Sistema de Gestión Bibliotecaria basadas en Servicios Web. Habana, Cuba, Universidad de las Ciencias Informáticas, 2012.
15. **UDDI, Introduction to UDDI: Important Features and Functional Concepts** October, 2004. Organization for the Advancement of Structured Information Standards, <http://www.uddi.org/pubs/uddi-tech-wp.pdf>.
16. **Márquez, Santiago.** La Web de Joaquín. Introducción a la creación de servicios web con .NET. [En Línea] 23 de octubre de 2007. <http://jms32.eresmas.net/2007/textos/CreacionDeServiciosWeb.html>.
17. **Manzaneque Alberca, Alejandro y Tendero Díaz, Jesús Gálvez.** Modelo avanzado de base de datos. Bases de datos Orientadas a Objetos y Objeto-Relacionales. 2008. S.l.:s.n.
18. **Pressman, Roger.** Ingeniería del Software: Un enfoque Práctico. s.l.: McGraw-Hill, 2005. ISBN: 9701054766.
19. **PHP.** PHP: Hypertext Processor. [En línea] [Citado el: 14 de Enero de 2012] www.php.net.
20. **Krajewski, Lee J.** Administración de operaciones: estrategia y análisis. Pearson Educación, 2000
21. **Jiménez, Fernando y Hernández García, Luis Enrique.** Yii: Un excelente framework para PHP. <http://softpei.blogspot.com/2014/04/yii-un-excelente-framework-para-php.html> [Online 29 de abril del 2014]
22. **Batalla, Liber; Fontanarossa, Diego; Garderes, Javier y Gatto, Andres.** Gestión de software, Extreme Programing(XP). 2006.
23. **Fulvio Corno, Dario Bonino.** PHP web services with the NuSOAP library. Dipartimento di Automatica e Informatica, Politecnico di Torino. Italia. Abril, 2009.
24. **Cossio Taboada, Miguel Humberto y Floren, Huanca A.** Diagrama de Despliegue. Universidad Salesiana de Bolivia. Bolivia. 2009.
25. **Salazar, Christian.** La Arquitectura de Yii para los servicios web. <http://www.yiiframeworkenespanol.org/doc/webservices-con-yii-plano-UML-christiansalazar.gif>
26. **Fernández Romero, Yenisleydis y González Díaz, Yanette.** Patrón Modelo-Vista-Controlador, Revista Digital de las Tecnologías de la Información y las Comunicaciones, TELEMATICA. Cuba. Enero-Abril 2012.
27. **Rodríguez, Jorge.** Pruebas Unitarias. <http://blog.continuum.cl/wp-content/uploads/2008/08/pruebas-unitarias.pdf>. Marzo 2006.

28. **Guaita Martínez, Álvaro.** PHPUnit: herramienta de testeo PHP.
http://www.desarrolloweb.com/de_interes/phpunit-herramienta-testeo-php-4436.html.
29. **W3C.** W3C.org. <http://www.w3.org/TR/wsd>

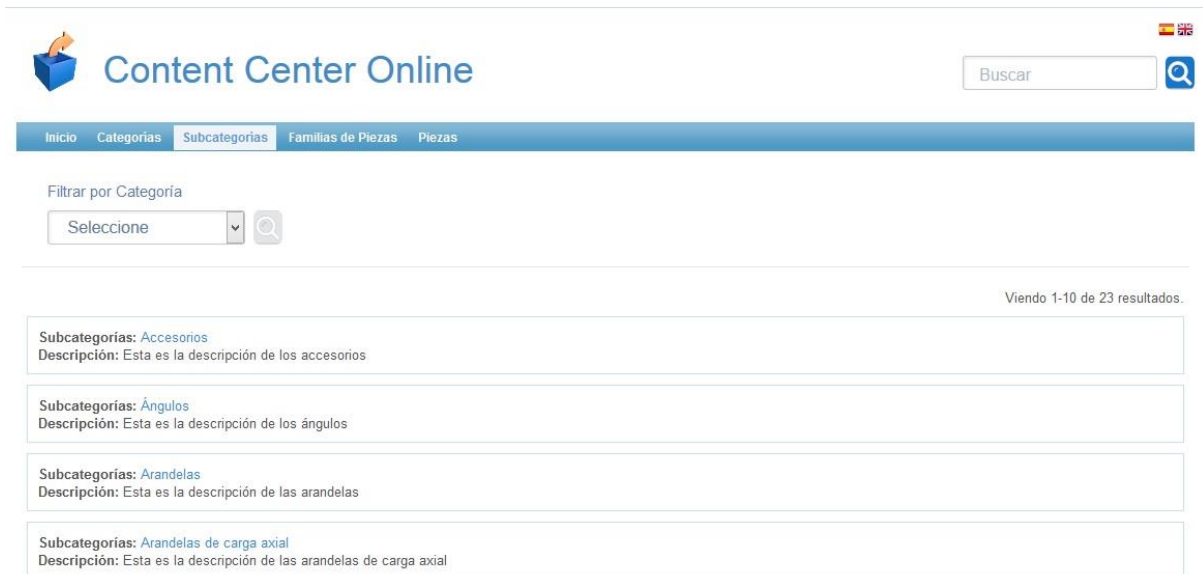
ANEXOS

Anexo 1 Vista del cliente en el consumo del servicio web CategorySoap de la función ListCategory.



The screenshot shows the 'Content Center Online' interface. At the top left is the logo, and at the top right is a search bar with the text 'Buscar' and a magnifying glass icon. Below the search bar is a navigation menu with tabs: 'Inicio', 'Categorías', 'Subcategorías', 'Familias de Piezas', and 'Piezas'. The 'Categorías' tab is selected. Below the navigation menu, there are five search results, each in a separate box. Each box contains the category name and a description. The results are: 1. 'Categorías: Fijaciones' with description 'Descripción de Fijaciones'. 2. 'Categorías: Otras piezas' with description 'Descripción de otras piezas'. 3. 'Categorías: Perfiles estructurales' with description 'Descripción de Perfiles Estructurales'. 4. 'Categorías: Piezas del eje' with description 'Descripción de Piezas del eje'. 5. 'Categorías: Tubo y tubería' with description 'Descripción de Tubo y tubería'. In the top right corner of the results area, it says 'Viendo 1-5 de 5 resultados.'

Anexo 2 Vista del cliente en el consumo del servicio web SubCategorySoap de la función ListSubCategory.



The screenshot shows the 'Content Center Online' interface. At the top left is the logo, and at the top right is a search bar with the text 'Buscar' and a magnifying glass icon. Below the search bar is a navigation menu with tabs: 'Inicio', 'Categorías', 'Subcategorías', 'Familias de Piezas', and 'Piezas'. The 'Subcategorías' tab is selected. Below the navigation menu, there is a filter section titled 'Filtrar por Categoría' with a dropdown menu showing 'Seleccione' and a magnifying glass icon. Below the filter section, there are four search results, each in a separate box. Each box contains the subcategory name and a description. The results are: 1. 'Subcategorías: Accesorios' with description 'Esta es la descripción de los accesorios'. 2. 'Subcategorías: Ángulos' with description 'Esta es la descripción de los ángulos'. 3. 'Subcategorías: Arandelas' with description 'Esta es la descripción de las arandelas'. 4. 'Subcategorías: Arandelas de carga axial' with description 'Esta es la descripción de las arandelas de carga axial'. In the top right corner of the results area, it says 'Viendo 1-10 de 23 resultados.'

Anexo 3 Vista del cliente en el consumo del servicio web FamilySoap de la función ListFamily.

The screenshot shows the 'Content Center Online' web application. At the top left is the logo and the text 'Content Center Online'. To the right is a search bar with the text 'Buscar' and a magnifying glass icon. Below the logo is a navigation menu with items: 'Inicio', 'Categorías', 'Subcategorías', 'Familias de Piezas', and 'Piezas'. The 'Familias de Piezas' item is highlighted. Below the navigation menu is a filter section titled 'Filtrar por Subcategorías' with a dropdown menu showing 'Seleccione' and a search icon. Below the filter is a list of results, showing the first four items: 'Familia: Acanalado', 'Familia: Acoplamientos', 'Familia: Avellanado', and 'Familia: Boquillas'. Each item has a description. The text 'Viendo 1-10 de 68 resultados.' is visible at the top right of the results area.

Anexo 4 Vista del cliente en el consumo del servicio web PieceSoap de la función ListPiece.

The screenshot shows the 'Content Center Online' web application. At the top left is the logo and the text 'Content Center Online'. To the right is a search bar with the text 'Buscar' and a magnifying glass icon. Below the logo is a navigation menu with items: 'Inicio', 'Categorías', 'Subcategorías', 'Familias de Piezas', and 'Piezas'. The 'Piezas' item is highlighted. Below the navigation menu is a filter section titled 'Filtrar por Familia de Piezas' with a dropdown menu showing 'Seleccione' and a search icon. Below the filter is a view selector with two options: 'Vista de Iconos' (selected) and 'Vista de Lista'. Below the view selector is a list of results, showing the first ten items, each represented by a large 'P' icon and a label: 'ISO 1035/1 - 1980', 'ISO 1035/2 - 1980', 'ISO 1035/3 - 1980', 'ISO 104 (Dirección única)', 'ISO 104 (Doble dirección)', 'ISO 1051', 'ISO 10642', 'ISO 1127 Tubo', 'ISO 1127 Tubo con finales roscados', and 'ISO 1206 (I)'. The text 'Viendo 1-10 de 562 resultados.' is visible at the top right of the results area.

Anexo 5 Vista del cliente en el consumo del servicio web PieceSoap para la función Search.

The screenshot shows the 'Content Center Online' website interface. At the top right, there is a search bar containing 'ISO 70' and a magnifying glass icon. Below the search bar is a navigation menu with 'Inicio', 'Categorías', 'Subcategorías', 'Familias de Piezas', and 'Piezas'. Underneath the menu, there are two tabs: 'Vista de Iconos' (selected) and 'Vista de Lista'. The main content area displays the search results for 'ISO 70', showing 'Resultado de la Búsqueda ISO 70' and 'Viendo 1-10 de 199 resultados.' There are ten search results, each represented by a large letter 'P' icon and a text description. The results are arranged in two rows of five. The first row contains results for ISO PN 10, 16, 25, 40, and 6. The second row contains results for ISO PN 150(B2), 20(B2), 260(B2), and 420(B2).

Icono	Descripción
P	ISO 7005-1 Brida ciega tipo 05 - Cara plana - ISO PN 10
P	ISO 7005-1 Brida ciega tipo 05 - Cara plana - ISO PN 16
P	ISO 7005-1 Brida ciega tipo 05 - Cara plana - ISO PN 25
P	ISO 7005-1 Brida ciega tipo 05 - Cara plana - ISO PN 40
P	ISO 7005-1 Brida ciega tipo 05 - Cara plana - ISO PN 6
P	ISO 7005-1 Brida ciega tipo 05 - Cara saliente - ISO PN 110(B2)
P	ISO 7005-1 Brida ciega tipo 05 - Cara saliente - ISO PN 150(B2)
P	ISO 7005-1 Brida ciega tipo 05 - Cara saliente - ISO PN 20(B2)
P	ISO 7005-1 Brida ciega tipo 05 - Cara saliente - ISO PN 260(B2)
P	ISO 7005-1 Brida ciega tipo 05 - Cara saliente - ISO PN 420(B2)

Anexo 6 Vista del cliente en el consumo del servicio web PieceSoap para la función DetailFamilyPiece.

The screenshot shows a detailed view of a search result for 'ISO 1035/1 - 1980'. The window title is 'ISO 1035/1 - 1980'. There are two tabs: 'Información de familias' (selected) and 'Detalle de la Pieza'. The main content area displays a large letter 'P' icon and a table with the following information:

Nombre de la Pieza:	Descripción de la Pieza:	Organización Estándar:	Estándar:
ISO 1035/1 - 1980	Barras de acero laminadas en caliente - Parte 1: barras redondas	ISO 1035/1	ISO

Below the table, there is a 'Cerrar' button. At the bottom right of the window, it says 'Viendo 1-10 de 562 resultados.' Below the window, there are three search results listed:

- Piezas: ISO 1035/1 - 1980
Descripción: Barras de acero laminadas en caliente - Parte 1: barras redondas
- Piezas: ISO 1035/2 - 1980
Descripción: Barras de acero laminadas en caliente - Parte 2: barras cuadradas
- Piezas: ISO 1035/3 - 1980
Descripción: Barras de acero laminadas en caliente - Parte 3: barras planas

Anexo 7 Vista del cliente en el consumo del servicio web PieceSoap para la función DetailPiece.

ISO 1035/1 - 1980														
Información de familias												Detalle de la Pieza		
id	Material	Nombre de archivo de pieza	Número de pieza	Referencia de almacen	Longitud de la base	Módulo de rigidez de torsion (2)	Coefficiente de desplazamiento cortante	Momento de inercia (x)	Masa especifica	Módulo de rigidez de torsion (1)	Módulo de seccion (x)	Designación	Área de sección	Diámetro de barra
1	Steel Mild	ISO 1035/1 - 8 - 0.001	ISO 1035/1 - 8 - 0.001	8	0.0001	0.100530937	1.128000021	0.020106189	0.395086618566	0.040212378	0.050265472	8 - 0.001	0.502654731	8
2	Steel Mild	ISO 1035/1 - 10 - 0.001	ISO 1035/1 - 10 - 0.001	10	0.0001	0.196349502	1.128000021	0.049087375	0.617322832716	0.098174751	0.098174751	10 - 0.001	0.785398006	10
3	Steel Mild	ISO 1035/1 - 12 - 0.001	ISO 1035/1 - 12 - 0.001	12	0.0001	0.33929193	1.128000021	0.101787582	0.888944857386	0.203575164	0.169645965	12 - 0.001	1.130973101	12
4	Steel Mild	ISO 1035/1 - 14 - 0.001	ISO 1035/1 - 14 - 0.001	14	0.0001	0.538783014	1.128000021	0.188574061	1.209952738164	0.377148122	0.269391507	14 - 0.001	1.539380074	14
5	Steel Mild	ISO 1035/1 - 15 - 0.001	ISO 1035/1 - 15 - 0.001	15	0.0001	0.662679553	1.128000021	0.248504847	1.388976374004	0.497009695	0.331339777	15 - 0.001	1.767145514	15
6	Steel Mild	ISO 1035/1 - 16 - 0.001	ISO 1035/1 - 16 - 0.001	16	0.0001	0.804247558	1.128000021	0.321699023	1.58034647505	0.643398046	0.402123779	16 - 0.001	2.010618925	16
7	Steel Mild	ISO 1035/1 - 17 - 0.001	ISO 1035/1 - 17 - 0.001	17	0.0001	0.964665055	1.128000021	0.409982651	1.784062946196	0.819965303	0.482332528	17 - 0.001	2.269800186	17
8	Steel Mild	ISO 1035/1 - 18 - 0.001	ISO 1035/1 - 18 - 0.001	18	0.0001	1.14511025	1.128000021	0.515299618	2.000125881762	1.030599356	0.572555125	18 - 0.001	2.544689417	18
9	Steel Mild	ISO 1035/1 - 19 - 0.001	ISO 1035/1 - 19 - 0.001	19	0.0001	1.346761227	1.128000021	0.639711618	2.228535468816	1.279423237	0.673380613	19 - 0.001	2.835286856	19
10	Steel Mild	ISO 1035/1 - 20 - 0.001	ISO 1035/1 - 20 - 0.001	20	0.0001	1.570796013	1.128000021	0.785398006	2.469291332436	1.570796013	0.785398006	20 - 0.001	3.141592026	20
11	Steel Mild	ISO 1035/1 - 21 - 0.001	ISO 1035/1 - 21 - 0.001	21	0.0001	1.818392754	1.128000021	0.954656184	2.72239365969	1.909312367	0.909196377	21 - 0.001	3.463605165	21
12	Steel Mild	ISO 1035/1 - 22 - 0.001	ISO 1035/1 - 22 - 0.001	22	0.0001	2.051441401	1.128000021	1.244511025	3.000125881762	2.051441401	1.244511025	22 - 0.001	3.926990817	22