



**Universidad de las Ciencias Informáticas**  
**Facultad 10**



**Módulo de Servicios Web para dotProject.**  
**DOT-WS.**

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

**Autor**

*Yoani Santana Leiva.*

**Tutor**

*Ing. Roberto Rodríguez Montoya*

*Ing. Yoandy Pérez Villazón*

## **DECLARACIÓN DE AUTORÍA**

Declaro que soy el único autor de la presente tesis 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 \_\_\_\_ del año \_\_\_\_\_.

---

Yoani Santana Leiva

---

Ing. Yoandy Pérez Villazón

---

Ing. Karel Tito Catá

## **DATOS DE CONTACTO DE LOS TUTORES**

### **Datos generales**

**Nombre y Apellidos:** Yoandy Pérez Villazón

**Fecha de nacimiento:** 16/04/1984

**Carne de Identidad:** 84041609944

**Dirección:** Calle Luz Caballero #125 % Mariana Grajales y Martha Abreu. Colón. Matanzas. Cuba.

**Nacionalidad:** Cubana.

**Teléfono Centro de Trabajo:** (+53) (7) 837 3156

**Correo electrónico:** yvillazon@uci.cu

**Fecha de actualización del currículum:** Mayo 2009

**Institución:** Universidad de las Ciencias Informáticas.

### **Resumen**

Ingeniero en Ciencias Informáticas de la Universidad de las Ciencias Informáticas (UCI) desde el 30 de mayo del 2008. Una vez graduado se inicia como profesor en la UCI en la disciplina de Física. Actualmente se desempeña como líder del proyecto Mejoras al Sistema dotProject CNTI en la Facultad 10. Ha impartido las asignaturas de Física I y Física II. Referente a la investigación ha desarrollado trabajos de investigación para UCIENCIA e Informática 2009, en este último evento posee dos artículos, uno como autor en la modalidad de póster y otro como coautor en el evento virtual, ambos incluidos en las memorias del evento. Ha ejercido como miembro de diversos tribunales de tesis de grado y a tutorado diversos trabajos a estudiantes de pregrado. En el área productiva a trabajado en diversos proyectos relacionados con la migración a Software Libre.

### **Formación académica**

Ingeniería en Ciencias Informáticas, Universidad de las Ciencias Informáticas (UCI), 2008.

### **Categoría docente y/o categoría científica**

Instructor recién graduado / Ingeniero

## **Líneas de investigación**

Software Libre. Herramientas de Gestión de Proyectos.

Las investigaciones más importantes:

- “Metodología de Migración a Software Libre para el MES”, Mayo 2008.
- “Mini-Guía de Migración a Software Libre”, Octubre 2008.
- “Guía Cubana de Migración a Software Libre”, Marzo 2009.

## **Publicaciones y memorias de eventos**

- “Metodología para la Migración a Software Libre del MES”. Yoandy Pérez Villazón. IV Taller de Software Libre. Memorias de la Feria y Convención Internacional Informática 2009. La Habana, Cuba. 2009.
- “Guía Cubana de Migración a Software Libre”. Ramón Paumier Samón, Yoandy Pérez Villazón. IV Taller de Software Libre. Memorias de la Feria y Convención Internacional Informática 2009. La Habana, Cuba. 2009.

## **Datos generales**

**Nombre y Apellidos:** Karel Tito Catá

**Fecha de nacimiento:** 12/10/1985

**Carne de Identidad:** 85101227906

**Dirección:** Calle 7 Norte #706 % San Lino y Santa Rita. Guantánamo. Guantánamo. Cuba.

**Nacionalidad:** Cubana.

**Teléfono Centro de Trabajo:** (+53) (7) 837 2414

**Correo electrónico:** ktito@uci.cu

**Fecha de actualización del currículum:** Mayo 2010

**Institución:** Universidad de las Ciencias Informáticas.

## **Resumen**

Ingeniero en Ciencias Informáticas de la Universidad de las Ciencias Informáticas (UCI) desde el 29 de mayo del 2008. Una vez graduado se inicia como especialista en la Dirección de Producción #3 de la Infraestructura Productiva(IP) de la

UCI. En su trabajo como especialista de la IP se encarga del asesoramiento y guía de los procesos productivos, específicamente de los proyectos de exportación de la UCI. Actualmente se desempeña como especialista del Departamento de Gestión de Proyectos de la UCI. Referente a la investigación ha desarrollado trabajos de investigación para FORDES y la Vicerrectoría de Producción de la UCI, en primero posee un artículo como coautor en el evento presencial, incluido en las memorias del evento. Es asesor de tecnologías web del proyecto SIMAYS del Centro Productivo GEITEL de la UCI.

### **Formación académica**

Ingeniería en Ciencias Informáticas, Universidad de las Ciencias Informáticas (UCI), 2009.

### **Categoría docente y/o categoría científica**

Instructor recién graduado / Ingeniero

### **Líneas de investigación**

Software Libre. Gestión de Proyectos. Calidad de Software.

## **AGRADECIMIENTOS**

A mi familia, a mi mamá y a mi papá, a mis hermanos, a mis abuelos. Agradecimientos especiales a mi novia Sheilita y a su familia por estar siempre ahí conmigo. A mis tutores Karel y Yoandy por su ejemplo y amistad incondicional. A todos los que de una forma u otra me ayudaron en la formación de mi vida profesional y personal. A todos los que siempre me han brindado su apoyo en los malos y buenos momentos. Agradezco a mis compañeros del proyecto por su ayuda cuando la necesité.

## **DEDICATORIA**

*Dedico este trabajo especialmente a mi mamá, a mi papá, a mis abuelos, a mis hermanos, a mi novia Sheilita por su incansable apoyo, a toda la familia de mi novia por estar siempre conmigo.*

## RESUMEN

La Gestión de Proyecto ha tomado un auge vertiginoso en los últimos años. La convergencia entre esta ciencia y la informática ha dado lugar a los Sistemas de Gestión de Proyectos permitiendo automatizar los procesos de gestión de proyectos. Al Sistema de Gestión de Proyectos dotProject se le han incorporado funcionalidades que le brindan interoperabilidad para integrarse a la Plataforma de migración con el objetivo de que este pueda ser usado por la misma. El trabajo de diploma cuenta con un estudio de los servicios web y su utilización en los Sistemas de Gestión de Proyectos. Además se identificaron los estándares a utilizar para el desarrollo de servicios web y cuantos de estos se necesitan implementar. La utilización de la metodología de desarrollo de software SXP permitió agilizar el análisis, diseño e implementación de la solución propuesta. Se aplicó el estilo arquitectónico Modelo-Vista-Controlador (MVC), que separa la lógica del negocio de la presentación. Se utilizaron las facilidades que brinda el framework de dotProject. Finalmente los resultados se validaron utilizando las técnicas que propone SXP. Al concluir este trabajo se obtuvo un módulo de servicios web que le permiten a otras aplicaciones gestionar tareas, proyectos y usuarios del dotProject.

**Palabras claves:** Gestión de Proyectos, Sistemas de Gestión de Proyectos, dotProject, Servicios Web

## TABLA DE CONTENIDOS

CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA.....	5
1.1Definición de la Gestión de Proyectos(GP).....	5
1.2Evolución de la GP. ....	6
1.3Tendencias de la GP.....	8
1.4Los Sistemas de Gestión de Proyectos (SGP).....	8
1.4.1Ejemplos de SGP.....	9
1.4.2SGP interoperables. ....	11
1.5¿Qué son los Servicios Web?.....	11
1.6Importancia de los Servicios Web.....	11
1.7Tendencias de los Servicios Web.....	12
1.8Protocolos de comunicación entre aplicaciones.....	12
1.8.1XML-RPC.....	12
1.8.2REST.....	12
1.8.3SOAP.....	13
1.8.4Propuesta del protocolo de comunicación a utilizar.....	13
1.9Tecnologías para el desarrollo de Servicios Web.....	14
1.9.1NuSOAP.....	14
1.9.2Extensión de PHP5 para SOAP.....	14
1.9.3Propuesta de la tecnología de desarrollo a utilizar.....	14
1.10Herramientas, lenguajes y tecnologías a utilizar.....	14
1.10.1Lenguajes de programación.....	15
1.10.2Herramientas utilizadas.....	16
1.10.3Metodologías de desarrollo de software. ....	18
CAPÍTULO 2 ANÁLISIS Y DISEÑO DEL MÓDULO.....	20
2.1Concepción inicial.....	20
2.2Descripción del módulo de Servicios Web DOT-WS.....	20
2.3Levantamiento de requisitos.....	20
2.4Propuesta del sistema a implementar.....	22
2.5Descripción de la arquitectura.....	23
2.6Guía para la creación de un servicio web.....	26
2.6.1Uso del WSDL.....	26
2.7Guía para la creación de un módulo. ....	27
CAPÍTULO 3 IMPLEMENTACIÓN DEL MÓDULO.....	30
3.1Planificación del proyecto por roles.....	30
3.2Historias de usuarios.....	30
3.3Plan de versiones.....	41
3.4Tareas de Ingeniería.....	42
CAPÍTULO 4 VALIDACIÓN DEL MÓDULO.....	51
4.1Casos de Pruebas.....	51
4.2Resultados Obtenidos.....	54
4.3Funcionalidades obtenidas.....	54
CONCLUSIONES.....	55
RECOMENDACIONES.....	56
REFERENCIAS BIBLIOGRÁFICAS.....	57
BIBLIOGRAFÍA.....	60
GLOSARIO DE TÉRMINOS.....	63
ANEXOS.....	66

---

## Introducción

Durante los inicios del desarrollo de las aplicaciones uno de los problemas que se trató de solucionar era que los sistemas fueran independientes de la plataforma de hardware y sistema operativo (Web Services Journal 2009), lo que de alguna forma se logra con la aparición de lenguajes de programación que solucionan este problema, como por ejemplo Java (Java 2009). Internet abrió las puertas para generar aplicaciones muchos más independientes y accesibles desde cualquier parte del mundo, pero aún así existía un problema que se debía solucionar que era la heterogeneidad de los sistemas, los cuales estaban escritos en lenguajes distintos por lo que su interoperabilidad era difícil. Es en este punto donde nace una solución que se nutre de diversas tecnologías, que de alguna forma solucionan este problema de comunicación entre aplicaciones que se encuentran en ambientes distribuidos; y que se desarrollan absolutamente de forma diferente. Esta solución es conocida como Servicios Web (w3c 2008).

Por otro lado, en la actualidad es común el uso de sistemas informáticos que facilitan la gestión de los procesos que se llevan a cabo en las empresas productoras de software, tal es el caso de los Sistemas de Gestión de Proyectos (SGP). Diversas son las facilidades y ventajas que proporciona el empleo de los SGP, justificando así la gran importancia que han tomado dentro del mundo del desarrollo de software. Su repercusión más palpable es la de favorecer en gran medida el proceso de toma de decisiones con una mejor eficacia y rapidez. Actualmente existen diversas aplicaciones destinadas a estos fines que constan de muy buena calidad y aceptación a nivel mundial (Cynertia Consulting 2009). Entre ellas se encuentran GanttProject (Arleco 2008), OpenProj (Salva Castro 2008), ActiveCollab (activeCollab 2009), Planner (Planner team 2009), MSProject (Arlecob 2008) y dotProject (Ernesto Serrano 2009).

El SGP dotProject es una herramienta que permite gestionar los proyectos existentes en una empresa, además de las tareas, eventos y recursos asociados a cada uno de ellos. Para el líder de proyecto esta labor es compleja sin la ayuda de herramientas como los SGP.

El Grupo de proyecto UNICORNIOS de la Universidad de las Ciencias Informáticas (UCI) utilizó por alrededor de tres años este SGP, teniendo gran aceptación con el mismo. Luego de la reorganización en los procesos productivos de la

---

UCI se desintegra UNICORNIOS, surgiendo los centros de desarrollo y unido a ellos los departamentos productivos. El centro de Gestión de la Información y Tecnologías Libres (GEITEL) está compuesto por varios departamentos, siendo una de ellos el de Sistemas Operativos y Desarrollo de Tecnologías Libres (SODTL), en el cual se enmarca el proyecto de Servicios Integrales en Migración, Asesoría y Sistemas (SIMAYS). Una de las metas es la creación de la plataforma de migración, un sistema integral para la automatización de gran parte de las tareas de este proceso, siendo una de las actividades más importantes la automatización de las actividades de planificación de la migración. Como parte del desarrollo de la plataforma se necesita gestionar información del dotProject pero desde otras aplicaciones. En estos momentos existe la imposibilidad de gestionar información desde este SGP por otras aplicaciones que lo soliciten, debido a la ausencia en el mismo de funcionalidades que permitan la interoperabilidad.

Teniendo en cuenta lo descrito anteriormente se plantea como **problema científico**: ¿Cómo desarrollar un módulo para dotProject que permita la interoperabilidad del mismo con otras aplicaciones? Del problema anterior se puede definir que el **objeto de estudio** de este trabajo de diploma es el proceso de desarrollo de software en aplicaciones web. Acotando el **campo de acción** al Sistema de Gestión de Proyectos dotProject en el proyecto SIMAYS.

Para ello se plantea como **objetivo general de la investigación** desarrollar un módulo para el SGP dotProject que permita la interoperabilidad del mismo con otras aplicaciones.

Desglosando el mismo en los siguientes **objetivos específicos**:

- Sistematizar acerca de los servicios web y su utilización en los Sistemas de Gestión de Proyectos (SGP).
- Identificar los posibles estándares para utilizar durante la implementación.
- Identificar los distintos servicios que se brindarán.
- Analizar y diseñar un módulo para dotProject que permita brindar diferentes servicios web aplicando los principales estándares que existen.

- 
- Implementar el módulo de servicios web para dotProject (DOT-WS<sup>1</sup>).
  - Validar el módulo descrito anteriormente.

Teniendo como idea a defender que si se crea un módulo de servicios web para el SGP dotProject permitirá la interoperabilidad del mismo con otras aplicaciones.

Para darle cumplimiento al objetivo propuesto se plantean las siguientes **tareas**:

- Estudio acerca de los servicios web y su utilización en los SGP.
- Identificación de los principales estándares que se usan en los servicios web.
- Identificación los distintos servicios web que se brindarán.
- Estudio el proceso de integración al SGP dotProject.
- Modelación de las funcionalidades que brindará el módulo DOT-WS.
- Implementación de las funcionalidades que brindará el módulo DOT-WS.
- Validación de las funcionalidades que brindará el módulo DOT-WS.

## **Métodos Científicos**

En el desarrollo del cumplimiento de las tareas científicas se utilizaron métodos teóricos<sup>2</sup> como el análisis histórico-lógico, que permitió estudiar la evolución de la Gestión de Proyectos, de los Servicios Web y los logros que se han obtenido; el analítico-sintético facilitó el resultado del trabajo con el análisis de gran cantidad de bibliografía de la Gestión de Proyectos y los Servicios Web, permitiendo sintetizar las características generales.

## **Estructura de la investigación.**

Para una mejor comprensión de la investigación, el contenido ha sido desglosado en cuatro capítulos, conclusiones generales, recomendaciones, referencias bibliográficas y bibliografía utilizada, además de un glosario de términos en el cual se detallan los términos técnicos y poco claros utilizados en la elaboración del documento además de los anexos que complementan el trabajo realizado.

Los capítulos han sido estructurados de la siguiente manera:

---

<sup>1</sup> DOT-WS: Siglas del módulo de Servicios Web a implementar en la presente investigación.  
<sup>2</sup> Métodos teóricos: los métodos teóricos se utilizan para el desarrollo de una investigación.

---

**Capítulo 1 Fundamentación teórica:** En este se definen una serie de conceptos acerca de la Gestión de Proyectos y de los SGP incluyendo elementos históricos acerca del surgimiento de los mismos y ejemplos. Incluyendo además que son los Servicios Web, importancia y tendencia de los mismos. Se describen las tecnologías, lenguajes y herramientas a utilizar en la elaboración de la solución.

**Capítulo 2 Análisis y diseño del módulo:** A lo largo de este capítulo se recogen los requerimientos planteados por el cliente, se formula una propuesta de la solución a implementar y se diseña la arquitectura con que se implementará el sistema.

**Capítulo 3 Implementación del módulo:** En este capítulo se realiza la implementación de la solución propuesta en el capítulo anterior y se generan los artefactos correspondientes a la fase de desarrollo según la metodología seleccionada.

**Capítulo 4 Validación del módulo:** En este capítulo se realizan un conjunto de pruebas necesarias para validar el producto obtenido, además de explicarse en detalle las funcionalidades obtenidas en el mismo.

## **CAPÍTULO 1. FUNDAMENTACIÓN TEÓRICA**

### **1.1 Definición de la Gestión de Proyectos(GP).**

Un proyecto es un conjunto de actividades coordinadas e interrelacionadas que buscan cumplir con un cierto objetivo específico(proyecto 2008). Este generalmente debe ser alcanzado en un período de tiempo previamente definido y respetando un presupuesto. Es de gran importancia porque permite organizar el entorno de trabajo en tiempo y espacio. Estos siempre han existido desde tiempos remotos hasta nuestros días (Thompson 2009). La innovación siempre los ha acompañado, mejorando y perfeccionando día tras día las diferentes técnicas, teorías y competencias de la administración empresarial, así como el ámbito global de aplicación de todas ellas dentro de las organizaciones.

Actualmente la mayoría de los directores de proyectos de todo el mundo emplean numerosas técnicas, las cuales tienen ya muchos años de historia desde el momento en que se utilizaron por primera vez. Un ejemplo de ellos es el Diagrama de Gantt, técnica popular de planificación gráfica que fue desarrollada por Henry Lawrence Gantt en el año 1917 (Henry Lawrence 2009).

La gestión de proyectos se refiere a la organización y administración de recursos de manera tal que se pueda culminar todo el trabajo requerido en un proyecto dentro del alcance, el tiempo, y los costos definidos (Arq. Percy C. Acuña Vigil 2009).

Las características o atributos comunes a la mayoría de los proyectos son:

- Tienen un objetivo. La naturaleza de un proyecto debe ser real y debe tener sustento.
- Tienen un calendario de actividades, y deben tener un plan de trabajo.
- Son complejos, están compuestos por múltiples elementos.
- Demandan recursos. Requieren habilidades, conocimientos, capital y equipo de diversas áreas de una organización.
- Tienen una estructura organizacional, en la cual hay roles y responsabilidades.

Según el Project Management Institute (PMI):

- Son temporales. Tiene un principio y un fin.
- Son únicos. Se inician por un único propósito, y porque solo el proyecto usará ese único recurso, tiempo, espacio y demás recursos para ese único momento.
- Requieren de elaboración progresiva. Los procesos y ciclos de vida del proyecto se inician como una idea general y se detallan conforme transcurre el proyecto.

El primer desafío de la gestión de proyectos es asegurarse de que el proyecto sea realizado dentro de los parámetros definidos. El segundo es la asignación y la integración de las acciones necesarias para resolver esos objetivos predefinidos. El proyecto, por lo tanto, es un sistema cuidadosamente seleccionado de actividades para utilizar los recursos (tiempo, dinero, recursos humanos, materiales, energía, espacio, provisiones, comunicación, calidad, riesgos, etc.) con el objetivo de dar cumplimiento a las metas predefinidas (Arq. Percy C. Acuña Vigil 2009).

### **Las tres restricciones tradicionales.**

Los proyectos necesitan ser ejecutados y entregados bajo ciertas restricciones. Tradicionalmente, estas restricciones han sido alcance, tiempo y costo. Esto también se conoce como el Triángulo de la Gestión de Proyectos (M.C. Mario Garza González 2006), donde cada lado representa una restricción. Un lado del triángulo no puede ser modificado sin impactar a los otros. Un refinamiento posterior de las restricciones separa la calidad del producto del alcance, y hace de la calidad una cuarta restricción.

La disciplina de la gestión de proyectos consiste en proporcionar las herramientas y técnicas que permiten al equipo de proyecto (no solamente al gerente del proyecto) organizar su trabajo para cumplir con todas esas restricciones.

## **1.2 Evolución de la GP.**

Fue en los años 50 y 60 del siglo XX cuando la GP comenzó a profesionalizarse. Hasta entonces había sido meramente una disciplina. En aquella época, el desarrollo de grandes proyectos militares requería la coordinación del trabajo conjunto de equipos y áreas diferentes en la construcción de sistemas únicos (Angel Nieva López 2009).

Pero el perfeccionamiento en la GP no se ha limitado únicamente a la mejora y

madurez de sus técnicas y herramientas. El desvío en las planificaciones, costes y la falta, en ocasiones de calidad y utilidad de los entregables en los proyectos, provocó, que en los años 60 surgiera la necesidad de innovar y desarrollar nuevos métodos de organización y trabajo. Fue entonces cuando aparecieron diferentes organizaciones con el objetivo de desarrollar los conocimientos, metodologías, procesos (lo que se conoce como áreas del conocimiento) y las prácticas necesarias para gestionar esos trabajos, con las mejores garantías de previsibilidad y calidad de los resultados.

Entre estas organizaciones encontramos el Project Management Institute (PMI) y el International Project Management Association (IPMA), ambos fundados en 1965. El PMI es una Asociación Internacional de Profesionales de Gestión de Proyectos, que cuenta con más de 200.000 miembros vigentes con presencia en más de 125 países por todo el mundo y más de 85.000 certificados PMP's (Project Management Professional). Desde su fundación ha estado desarrollando y publicando estándares profesionales para la Gestión de Proyectos, los cuales se recogen actualmente en el PMBOK (Project Management Body of Knowledge).

La innovación en la Gestión de Proyectos siempre ha estado ligada a la innovación en Dirección Estratégica y Organización. Muestra de ello es la evolución que han experimentado tanto la propia Gestión de Proyectos como la figura del Director de Proyectos. Entre los años 1940 y 1970 podemos observar una Gestión de Proyectos tradicional (con su diagramas de Gantt y sus asignaciones de recursos). Se asume que las organizaciones viven en un mundo estático y predecible para desarrollar los proyectos y las decisiones se toman sin darle importancia a la satisfacción del cliente.

El presupuesto, cronograma y la asignación de recursos se manejan a través de una serie de herramientas fijas y el ciclo de vida de proyecto consta solo de cuatro fases: Concepto, Planificación, Ejecución y Cierre. En cambio, en la Gestión de Proyectos moderna (desde 1980 hasta nuestros días) las cosas cambian, ya que se asume que vivimos en un mundo caótico y no predecible para desarrollar los proyectos y, al revés que la Gestión de Proyectos tradicional, para la toma de decisiones se tiene en cuenta la opinión y satisfacción del cliente. Nacen nuevas técnicas y herramientas para gestionar los proyectos en los ámbitos de gestión de contratos, análisis de costes, análisis de riesgos, del entorno, negociación y gestión de cambios. Además, se incorporan nuevas fases al ciclo de vida de los proyectos.

### **1.3 Tendencias de la GP.**

En la actualidad el reto principal de la Gestión de Proyectos es la globalización, que provoca que la Gestión de Proyectos se expanda hacia entornos internacionales y por tanto, multiculturales. Por ello cada vez más se requieren conocimientos culturales específicos en idiomas y costumbres tanto de los países donde se desarrolla el proyecto como de los diferentes integrantes del equipo, para poder superar con éxito los retos con los cuales las empresas y los proyectos multiculturales se encuentran diariamente. Otros retos actuales a los que se enfrenta la Gestión de Proyectos moderna son las exigencias para conseguir resultados más rápidos y con alta calidad en proyectos, donde existe cada vez más un alto grado de incertidumbre y se multiplican los riesgos, el mantener un buen ambiente en el equipo de proyecto, adaptarse a las continuas reorganizaciones y la excesiva dependencia de la tecnologías de la información (Angel Nieva López 2009).

En un futuro cercano la Gestión de Proyectos será una competencia requerida en todas las organizaciones, todas requerirán que los Gestores de Proyectos estén certificados.

Existirá en pocos años mayor cantidad de oportunidades de trabajo en materia de gestión de de proyectos. No obstante, cada vez existirá mayor competencia y calidad entre los profesionales, que además tendrán que adaptarse a proyectos donde se trabaje a tiempo completo. Veremos como las organizaciones seguirán innovando en la Gestión de Proyectos y desarrollarán e incrementarán el uso de herramientas y evolucionarán nuevas teorías de gestión basadas en el liderazgo para poder afrontar con éxito los proyectos.

### **1.4 Los Sistemas de Gestión de Proyectos (SGP).**

En proyectos pertenecientes a las Tecnologías de la Información y las Comunicaciones (TIC) uno de los problemas más frecuentes es la falta de comunicación y de compartir conocimientos en los equipos de trabajo, un elemento que puede ayudar a establecer equipos de trabajo cohesionados de lo que está pasando con el proyecto es desarrollar modelos de interacción y comunicación efectivos, para ello la web 2.0 es un mecanismo muy potente, que permite un mayor grado interacción entre los usuarios (Alejandro Barros 2009).

En esta área se han desarrollado diferentes alternativas como las wiki o los blog

cuyo uso puede llegar a lograr discretos resultados si de gestión de proyectos se trata.

Poner a disposición de los miembros de un equipo de proyecto una herramienta tipo blog o wiki les permite interactuar de forma simple, manejar la historia y poder reducir la desigualdad de información del proyecto en sus diferentes áreas (técnica, económica, estratégica).

Los modelos colaborativos de administración presentan importantes ventajas, al establecer mecanismos efectivos de comunicación entre los diferentes miembros de un proyecto, cada uno desde su rol puede conocer lo que está pasando en el proyecto y aportar con comentarios y observaciones.

A diferencia de los modelos colaborativos expuestos anteriormente, los sistemas gestión de proyectos cuentan con una modalidad de uso estructurada y muy jerarquizada que facilita los procesos de gestión de proyectos.

Los SGP son herramientas que se orientan a la administración de forma automatizada de los recursos existentes para desarrollar un producto, cuya producción requiere de un conjunto de actividades o tareas que se desarrollen entre ellas en forma paralela o independiente y para las cuales una herramienta de este tipo cuenta con funcionalidades que automatizan y humanizan el trabajo.

#### **1.4.1 Ejemplos de SGP.**

##### **OpenProj**

Es multiplataforma, está desarrollado en Java y funciona bajo Windows, GNU/Linux, Unix y Mac OS. De hecho, una de las ventajas de OpenProj es que permite exportar e importar archivos .mpp y .mpx (extensiones de Microsoft Project). Además es completamente gratuito y está traducido a 8 idiomas incluyendo el español (Salva Castro 2008).

Como principal defecto podemos destacar la falta de documentación. Tiene una interfaz muy intuitiva y ligera, sin embargo, tiene algunas carencias, como errores en la traducción o que la estética de las ventanas es la de Java por defecto, no se integra con la del sistema y no usa servicios web.

##### **ActiveCollab**

Es una aplicación web, esto permite su administración y operación a través de un navegador y desde cualquier computadora con conexión. Fue desarrollado en el

lenguaje PHP y usa MySQL como sistema gestor de bases de datos.

La desventaja de esta herramienta es que existen traducciones para muchos idiomas excepto en español, y hasta el momento, no presenta soporte para LDAP y no utiliza servicios web (activeCollab 2009).

### **GanttProject**

Es una herramienta libre y fácil de usar. Está basada en la creación de diagramas de Gantt y de otros recursos gráficos. Puede organizar el proyecto en un árbol de tareas y asignar recursos humanos que tienen que trabajar en cada tarea, con la posibilidad de crear dependencias entre estas últimas. Es posible generar informes en HTML y PDF, el intercambio de datos con Microsoft Project y con hojas de cálculo, no presenta el uso de servicios web (Arleco 2008).

### **Microsoft Project**

Microsoft Project (MS Project) es un software para administración de proyectos desarrollado y vendido por Microsoft el cual está creado para asistir a los administradores de proyectos en el diseño de planes, asignación de recursos a tareas, rastreo de progresos y análisis de cargas de trabajo. Tiene como desventaja de que solo funciona sobre el sistema operativo Windows y no usa servicios web.

Aunque este software ha sido etiquetado como miembro de la familia Microsoft Office hasta el momento no ha sido incluido en ninguna de las ediciones de paquete de Office y se encuentra disponible en dos versiones, Standard y Professional (Arlecob 2008).

### **Planner**

Es una herramienta libre, una aplicación de escritorio para la gestión y seguimiento de proyectos de software. Permite la descomposición en tareas y subtareas, dependencias, identificación de la ruta crítica, diagramas de Gantt (Planner team 2009).

Entre sus funcionalidades permite definir un calendario, marcar eventos importantes, asignar recursos tanto humanos y materiales, establecer prioridades, entre otras más. Fue desarrollada para GNU/Linux.

### **dotProject**

Está construido haciendo uso de aplicaciones de código abierto. Es una aplicación basada en la web, multiusuario, soporta varios idiomas. Básicamente es un

programa al que se le van añadiendo proyectos, y dentro de estos proyectos las tareas que lo conforman (Ernesto Serrano 2009). Tiene todas las opciones básicas de cualquier gestor de proyectos:

- Empresas.
- Proyectos.
- Tareas.
- Calendario.
- Ficheros.
- Contactos.
- Usuarios.
- Sistema.

Es una herramienta que permite gestionar las distintas tareas que componen un proyecto. Además se perfila como una interesante herramienta para trabajar en entornos colaborativos, permitiendo a los integrantes del equipo trabajar compartiendo información relativa a los proyectos.

#### **1.4.2 SGP interoperables.**

Después de haber analizado las características y funcionalidades existentes en los SGP más importantes, se ha concluido que los analizados no presentan módulos o funcionalidades donde se publiquen servicios web con el fin de ofrecer información fuera del SGP y que sean utilizados por otras aplicaciones.

### **1.5 ¿Qué son los Servicios Web?**

Los Servicios Web son un conjunto de aplicaciones o de tecnologías con capacidad para interoperar en la web. Estas aplicaciones o tecnologías intercambian datos entre sí con el objetivo de ofrecer servicios. Los proveedores ofrecen sus servicios como procedimientos remotos y los usuarios solicitan un servicio llamando a estos procedimientos a través de la web (w3c 2008).

### **1.6 Importancia de los Servicios Web.**

- Mantener la coherencia y consistencia de la información en todos los sistemas que forman parte de la red distribuida de información.

- Posee un conjunto de funcionalidades que pueden ser accedidas desde cualquier lugar y plataforma.
- Son independientes de la plataforma de origen o destino.

## **1.7 Tendencias de los Servicios Web.**

Los Servicios Web comienzan a ser parte de las arquitecturas de software de los principales fabricantes. La red es cada vez más un entorno preparado para soportar el tráfico de servicios destinados a una gran variedad de dispositivos en el que interactúan tanto páginas web como usuarios. La tecnología de Microsoft, basada en Servicios Web sobre su estrategia .NET, es una de las más visionarias, según la consultora independiente Gartner Research (Gartner Research 2009). El uso de PHP unido al grupo de herramientas NUSOAP permitió el desarrollo de servicios web. Con la creación de la extensión de SOAP para PHP5 estas facilidades se incrementaron.

## **1.8 Protocolos de comunicación entre aplicaciones.**

Los protocolos son los que definen un conjunto de reglas para intercambiar información y cooperar (Revista redes 2009). Definen la manera como compartimos información a través del Internet o incluso la manera como chateamos.

### **1.8.1 XML-RPC.**

XML-RPC es un protocolo de llamada a procedimiento remoto que usa XML para codificar los datos y HTTP como protocolo de transmisión de mensajes. Es un protocolo muy simple ya que sólo define unos cuantos tipos de datos y comandos útiles, además de una descripción completa de corta extensión. La simplicidad del XML-RPC está en contraste con la mayoría de protocolos RPC que tiene una documentación extensa y requiere considerable soporte de software para su uso (XML-RPC 2008).

### **1.8.2 REST.**

La Transferencia de Estado Representacional (Representational State Transfer) o REST son un conjunto de principios de arquitectura, en la actualidad se usa en el sentido más amplio para describir cualquier interfaz web simple que utiliza XML y HTTP, sin las abstracciones adicionales de los protocolos basados en patrones de intercambio de mensajes como el protocolo de servicios web SOAP (Roy Thomas

Fielding 2000).

### 1.8.3 SOAP.

SOAP (Simple Object Access Protocol) es un protocolo estándar que define como dos objetos en diferentes procesos pueden comunicarse por medio de intercambio de datos XML. Este protocolo deriva de un protocolo creado por David Winer en 1998, llamado XML-RPC. SOAP fue creado por Microsoft , IBM y otros y está actualmente bajo el auspicio de la W3C. Es uno de los protocolos utilizados en los Servicios Web (w3c soap 2007).

### 1.8.4 Propuesta del protocolo de comunicación a utilizar.

En la siguiente tabla se evidencian las características de los distintos protocolos de comunicación que se están investigando con el objetivo de seleccionar el más adecuado a las necesidades de la solución a implementar.

**Tabla 1.** Comparación entre los protocolos de comunicación.

Protocolos	Ventajas
<b>REST</b>	<ul style="list-style-type: none"> <li>• Peso ligero no mucho margen suplementario xml.</li> <li>• Resultados humanos legibles.</li> <li>• Fácil construir, ningún juego de herramientas es requerido.</li> </ul>
<b>XML-RPC</b>	<ul style="list-style-type: none"> <li>• Simplicidad.</li> <li>• Las estructuras y series son siempre anónimas. Pasando múltiples estructuras o programadores de series tiene que confiar en la orden de los parámetros.</li> </ul>
<b>SOAP</b>	<ul style="list-style-type: none"> <li>• Fácil consumir.</li> <li>• Robusto.</li> <li>• Instrumentos de desarrollo.</li> </ul>

En el desarrollo de esta investigación se utilizará el protocolo de comunicación SOAP porque es fácil de consumir y robusto. Además es el protocolo definido por la Dirección de Informatización de la UCI en los lineamientos de la arquitectura para los sistemas que conforman la intranet universitaria<sup>3</sup>.

3 <http://uddi.uci.cu>: Sitio donde se encuentran publicados los lineamientos de arquitectura

## **1.9 Tecnologías para el desarrollo de Servicios Web.**

Los Servicios Web son calificados como una nueva etapa del desarrollo de los sistemas distribuidos, que permite aprovechar todas las ventajas de trabajar en el ambiente web, unidas a las tecnologías que pueden ser utilizadas para el desarrollo de los componentes finales.

### **1.9.1 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. Esta compuesto por una serie de clases que nos harán mucho más fácil el desarrollo de servicios web. Provee soporte para el desarrollo de clientes y de servidores. NuSOAP esta basado en SOAP 1.1, Web Services Description Language WSDL 1.1 (lenguaje de descripción de los servicios web) y HTTP 1.0/1.1 (Nusoap 2008).

### **1.9.2 Extensión de PHP5 para SOAP**

La extensión SOAP recientemente fue agregada dentro de PHP5. Como el soporte es nativo mejora 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.

### **1.9.3 Propuesta de la tecnología de desarrollo a utilizar.**

Nusoap tuvo mucha popularidad en su momento porque no existía una herramienta que implementara servicios web en PHP, con la aparición de la extensión de SOAP en PHP5 y la inactividad del proyecto Nusoap desde el 27 de julio del 2005, sería mejor el uso de dicha extensión, además viene con soporte para SOAP 1.1 y SOAP 1.2.

## **1.10 Herramientas, lenguajes y tecnologías a utilizar.**

A continuación se describen las herramientas, los lenguajes y las tecnologías a utilizar en el desarrollo del módulo de servicios web y en los artefactos que se generan en la metodología de desarrollo de software utilizada.

### **1.10.1 Lenguajes de programación.**

#### **JavaScript.**

Javascript es un lenguaje de programación utilizado para crear programas encargados de realizar acciones dentro del ámbito de una página web. Se trata de un lenguaje de programación del lado del cliente. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con Javascript podemos crear efectos especiales en las páginas y definir interactividades con el usuario (Angel Alvarez 2009).

#### **PHP (Hypertext Pre-processor).**

Es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. PHP soporta el trabajo con CGI, permite procesar la información de formularios, generar páginas con contenidos dinámicos, y el trabajo con cookies. PHP es un lenguaje de programación de estilo clásico, con variables, sentencias condicionales, bucles, 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 (php.net 2009).

#### **CSS (Cascading Style Sheets).**

Hojas de Estilo en Cascada (Cascading Style Sheets), es un mecanismo simple que describe cómo se va a mostrar un documento en la pantalla, o cómo se va a imprimir, o incluso cómo va a ser pronunciada la información presente en ese documento a través de un dispositivo de lectura. Esta forma de descripción de estilos ofrece a los desarrolladores el control total sobre el estilo y formato de sus documentos. (w3c CSS 2009).

#### **XHTML (eXtensible HyperText Markup Language).**

XHTML significa eXtensible HyperText Markup Language. Es el lenguaje en que se escriben los millones de documentos que hoy existen en el WWW. Cuando accedemos a uno de estos documentos, el cliente (Netscape, IE, Mosaic, Lynx, Mozilla) los interpreta y los muestra. Los documentos HTML son ficheros de texto plano (también conocidos como ASCII) que pueden ser editados con cualquier editor de textos (w3c html 2009).

### **XML (eXtensible Markup Language).**

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información (w3c XML 2009).

### **WSDL (Web Services Description Language).**

Es un protocolo basado en XML que describe los accesos al servicio web. Podríamos decir que es el manual de operación del servicio web, porque nos indica cuales son las interfaces que provee el Servicio Web y los tipos de datos necesarios para la utilización del mismo (w3c WSDL 2009).

### **UDDI (Universal Description, Discovery and Integration).**

Es un modelo de directorios para servicios web. Es una especificación para mantener directorios estandarizados de información acerca de los servicios web, sus capacidades, ubicación, y requerimientos en un formato reconocido universalmente. UDDI utiliza WSDL para describir las interfaces de los Servicios Web. Es un lugar en el cual podemos buscar cuales son los Servicios Web disponibles, una especie de directorio en el cual podemos encontrar los Servicios Web publicados y publicar los Servicios Web que desarrollemos (w3c UDDI 2009).

## **1.10.2 Herramientas utilizadas.**

### **Zend Studio**

Zend Studio o Zend Development Environment es un completo entorno de desarrollo integrado para el lenguaje de programación 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.

## Apache.

Apache es un servidor de páginas web, un programa que permite acceder a páginas web alojadas en un ordenador. Es el más utilizado seguido de Microsoft Information Services.

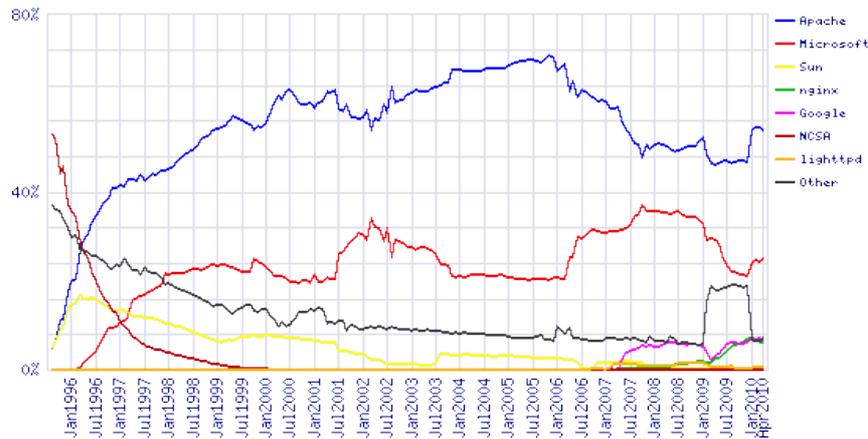


Figura 1. Uso de Apache en el mundo(netcraft 2010).

El servidor Apache es usado por múltiples razones como disponibilidad, facilidad de instalación, pocos recursos hardware necesarios, precio, disponibilidad del código fuente, entre otras. Existen otros aunque suelen estar especializados en nichos concretos de mercado. Corre en una multitud de sistemas operativos, lo que lo hace una herramienta multiplataforma (linux.ciberaula.com 2009).

## MySQL.

MySQL es un sistema de gestión de bases de datos relacional, licenciado bajo la licencia GNU/GPL, este gestor de bases de datos es, probablemente, el gestor más usado en el mundo del software libre, debido a su gran rapidez y facilidad de uso. Esta gran aceptación, es debida, en parte, a que existen infinidad de librerías y otras herramientas que permiten su uso a través de gran cantidad de lenguajes de programación, además de su fácil instalación y configuración (netpecos.org 2009).

## RapidSVN

RapidSVN es un cliente de interfaz gráfica para la comunicación con servidores Subversion. Facilita el versionado de ficheros, desde una interfaz sencilla e intuitiva. Está disponible para plataformas Windows, Linux, MAC OS y Solaris. Es rápido y eficiente, está escrito en C++. Distribuido bajo licencia GPL.

### **1.10.3 Metodologías de desarrollo de software.**

Las metodologías de desarrollo de software son un conjunto de procedimientos, técnicas y ayudas a la documentación para el desarrollo de productos software.

Van indicando paso a paso todas las actividades a realizar para lograr el producto informático deseado, indicando que personas deben participar en el desarrollo de las actividades y que papel deben de tener. Detallan la información que se debe producir como resultado de una actividad y la información necesaria para comenzarla (www.um.es 2009).

#### **Metodología SXP.**

SXP es una metodología de desarrollo de software compuesta por las metodologías SCRUM y XP que ofrece una estrategia tecnológica, a partir de la introducción de procedimientos ágiles que permitan actualizar los procesos de software para el mejoramiento de la actividad productiva fomentando el desarrollo de la creatividad, aumentando el nivel de preocupación y responsabilidad de los miembros del equipo, ayudando al líder del proyecto a tener un mejor control del mismo.

SCRUM es una forma de gestionar un equipo de manera que trabaje de forma eficiente y de tener siempre medidos los progresos, de forma que sepamos por dónde andamos. XP más bien es una metodología encaminada para el desarrollo; consiste en una programación rápida o extrema, cuya particularidad es tener como parte del equipo, al usuario final, pues es uno de los requisitos para llegar el éxito del proyecto.

Consta de 4 fases principales:

- Planificación-Definición donde se establece la visión, se fijan las expectativas y se realiza el aseguramiento del financiamiento del proyecto.
- Desarrollo, es donde se realiza la implementación del sistema hasta que este listo para ser entregado.
- Entrega, puesta en marcha.
- Mantenimiento, donde se realiza el soporte para el cliente.

De cada una de estas fases se realizan numerosas actividades tales como el levantamiento de requisitos, la priorización de la Lista de Reserva del Producto, definición de las Historias de Usuario, diseño, implementación, pruebas, entre otras; de donde se generan artefactos para documentar todo el proceso. Las entregas son frecuentes, lo que nos permite mejorar el diseño cada vez que se le añade una nueva funcionalidad.

SXP esta especialmente indicada para proyectos de pequeños equipos de trabajo, rápido cambio de requisitos o requisitos imprecisos, muy cambiantes, donde existe un alto riesgo técnico y se orienta a una entrega rápida de resultados y una alta flexibilidad. Ayuda a que trabajen todos juntos, en la misma dirección, con un objetivo claro, permitiendo además seguir de forma clara el avance de las tareas a realizar, de forma que los jefes pueden ver día a día cómo progresa el trabajo (Gladys Marsi Peñalver Romero 2008).

## CAPÍTULO 2 ANÁLISIS Y DISEÑO DEL MÓDULO

### 2.1 Concepción inicial.

Actualmente el SGP dotProject no tiene un módulo que ofrezca servicios web para posibilitar la interoperabilidad del mismo con otras aplicaciones. Este módulo se acoplará bajo la estructura modular que tiene el dotProject aumentando así las funcionalidades del mismo. El proyecto SIMAYS tiene como meta el desarrollo de la plataforma de migración de la cual este módulo es principal participe.

### 2.2 Descripción del módulo de Servicios Web DOT-WS.

El módulo DOT-WS se nutre principalmente de la información que existe en la base de datos del dotProject, está dividido por vistas: tarea, proyecto y usuario. Cada vista constituye un servicio y en cada una de ellos se muestra la descripción del mismo unido a las operaciones que contienen.

### 2.3 Levantamiento de requisitos.

Para recoger los requisitos se utiliza la Lista de Reserva del Producto según la metodología seleccionada; un artefacto que recoge en lenguaje natural las funcionalidades que debe cumplir el sistema para su desarrollo y está conformado por una lista priorizada que define el trabajo que se va a realizar en el proyecto.

**Tabla 2.** Levantamiento de Requisitos.

Prioridad	Ítem *	Descripción	Estimación	Estimado por
<b>Alta</b>				
	1	Crear la clase task_service que contenga el servicio tarea.	2	ANA. y PROG.
	2	Obtener la lista de las nuevas tareas de un usuario.	0.5	ANA. y PROG.
	3	Obtener la lista de todas las tareas de un usuario.	0.5	ANA. y PROG.
	4	Actualizar el estado de una tarea.	0.5	ANA. y PROG.
	5	Eliminar tarea.	0.5	ANA. y PROG.
	6	Crear tarea.	0.5	ANA. y PROG.
	7	Modificar tarea.	0.5	ANA. y PROG.

	8	Adicionar historial a una tarea.	0.5	ANA. y PROG.
	9	Modificar historial de una tarea.	0.5	ANA. y PROG.
	10	Eliminar historial de una tarea.	0.5	ANA. y PROG.
	11	Listar las tareas según su estado.	0.5	ANA. y PROG.
	12	Mostrar la descripción completa del servicio.	0.5	ANA. y PROG.
	13	Mostrar la descripción de las funciones del servicio.	0.5	ANA. y PROG.
	14	Crear la clase project_service que contenga el servicio proyecto.	2	ANA. y PROG.
	15	Listar todos los proyectos.	0.5	ANA. y PROG.
	16	Obtener datos de proyecto dado un identificador.	0.5	ANA. y PROG.
	17	Obtener las tareas de un proyecto.	0.5	ANA. y PROG.
	18	Crear un nuevo proyecto.	0.5	ANA. y PROG.
	19	Modificar un proyecto.	0.5	ANA. y PROG.
	20	Eliminar un proyecto.	0.5	ANA. y PROG.
	21	Crear la clase user_service que contenga el servicio usuario.	2	ANA. y PROG.
	22	Listar todos los usuarios.	0.5	ANA. y PROG.
	23	Obtener datos de un usuario.	0.5	ANA. y PROG.
	24	Insertar usuario.	0.5	ANA. y PROG.
	25	Eliminar usuario.	0.5	ANA. y PROG.
	26	Actualizar datos de usuario.	0.5	ANA. y PROG.
<b>Media</b>				
	27	Crear instalador del módulo.	2	ANA. y PROG.
	28	Crear vista de tarea.	1	ANA. y PROG.
	29	Crear vista de proyecto	1	ANA. y PROG.
	30	Crear vista de usuario	1	ANA. y PROG.
<b>RNF</b>				
	31	Manual de usuario del módulo DOT-WS	2	ANA. y PROG.
	32	Manual de identidad del módulo DOT-WS.	2	ANA. y PROG.
	33	Manual de desarrollo del módulo DOT-WS.	1	ANA. y PROG.
	34	Requiere de la extensión soap de php5 instalada en el servidor.	0.3	ANA. y PROG.

## 2.4 Propuesta del sistema a implementar.

De forma general el módulo de Servicios Web que se desea implementar debe mostrar al usuario previamente registrado en el sistema los servicios que se ofrecen en el mismo, se muestra una descripción del servicio con las funcionalidades de cada uno de ellos.

Deben mostrarse los estándares bajo los cuales están implementados estos servicios web, así como el transporte utilizado por los mismos, agregando los parámetros de entradas de las funciones y la respuestas de las mismas. Estos servicios serán muy fácil de utilizar y el nombre evidencia claramente lo que se quiere obtener.

La interfaz principal del módulo DOT-WS estará compuesto por un sistema de navegación por pestañas siendo consecuente con el diseño establecido por dotProject que dará acceso a cada uno de los diferentes servicios representados en las siguientes vistas:

### ➤ **Vista de tarea.**

Muestra al usuario una descripción detallada del servicio `task_service` junto a las diferentes funcionalidades que el mismo contiene, las cuales son descritas brevemente agregando los parámetros de entrada y lo que las mismas devuelven. Estas funciones realizarán acciones sobre las tareas.

### ➤ **Vista de proyecto.**

Muestra al usuario una descripción detallada del servicio `project_service` junto a las diferentes funcionalidades que el mismo contiene, las cuales son descritas brevemente agregando los parámetros de entrada y lo que las mismas devuelven. Estas funciones realizarán acciones sobre los proyectos.

### ➤ **Vista de usuario.**

Muestra al usuario una descripción detallada del servicio `user_service` junto a las diferentes funcionalidades que el mismo contiene, las cuales son descritas brevemente agregando los parámetros de entrada y lo que las mismas devuelven. Estas funciones realizarán acciones sobre los usuarios.

## 2.5 Descripción de la arquitectura.

La arquitectura Modelo-Vista-Controlador (MVC) separa la lógica de negocio (el modelo) y la presentación (la vista) por lo que se consigue un mantenimiento más sencillo de las aplicaciones. Si por ejemplo una misma aplicación debe ejecutarse tanto en un navegador estándar como un navegador de un dispositivo móvil, solamente es necesario crear una vista nueva para cada dispositivo; manteniendo el controlador y el modelo original. El controlador se encarga de aislar al modelo y a la vista de los detalles del protocolo utilizado para las peticiones (HTTP, consola de comandos, email, etc.). El modelo se encarga de la abstracción de la lógica relacionada con los datos, haciendo que la vista y las acciones sean independientes. En la siguiente figura se muestra el diagrama de componentes que describe la arquitectura del módulo.

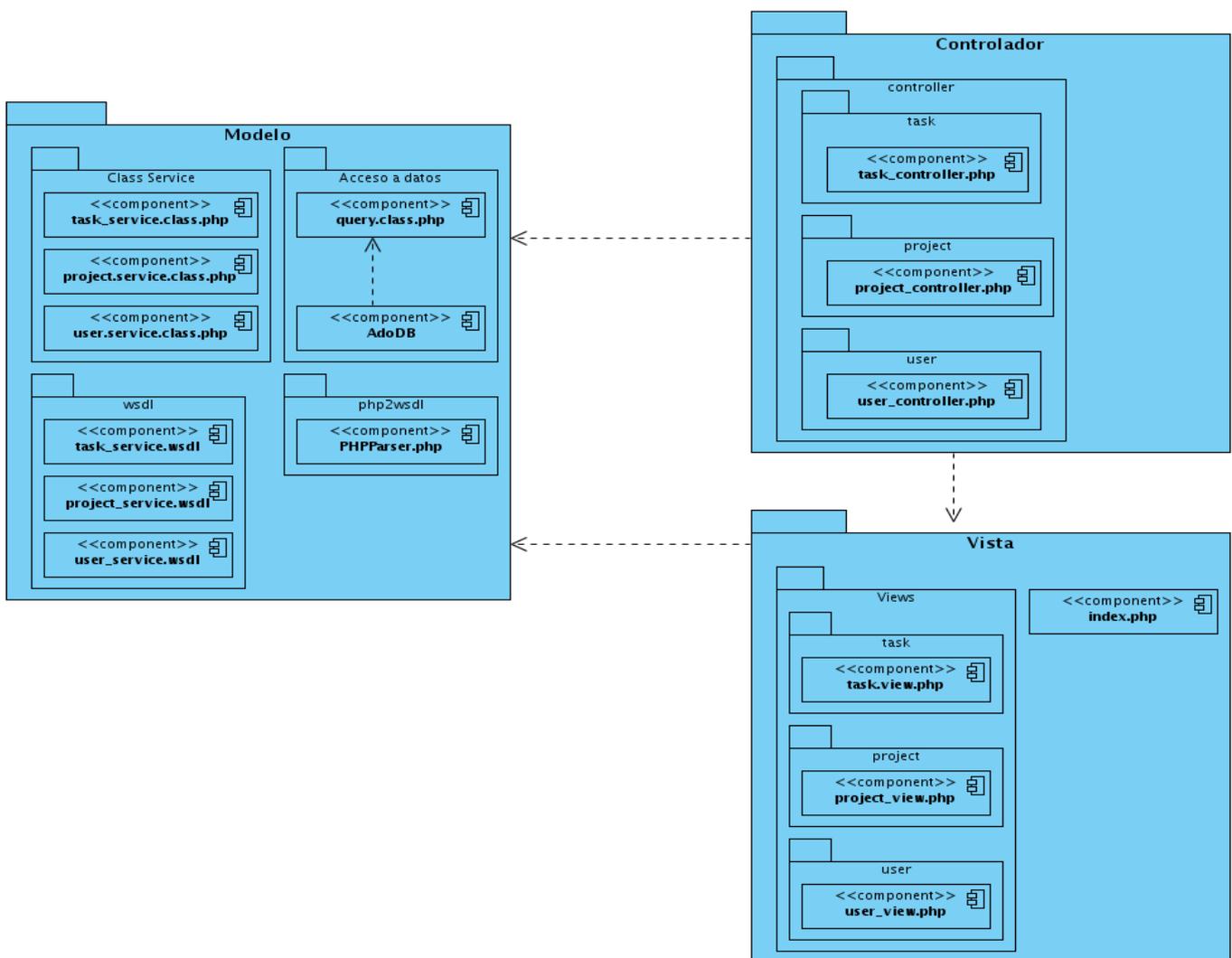


Figura 2. Diagrama de Componentes.

**Modelo:** Producto a que el dotProject constituye un framework, la mayoría de las clases de acceso a datos y lógica del negocio que utiliza el módulo DOT-WS son las mismas con las que cuenta el mismo. Logrando de esta manera una mayor reutilización del código.

## Clases y librerías utilizadas:

### 1. Librerías pertenecientes al dotProject.

- **DBQuery:** Clase encargada de realizar las conexiones a la base de datos haciendo uso de funcionalidades ya existentes en la librería de acceso a datos adodb.
- **CappUI:** Clase encargada de gestionar los elementos de interfaz de cada usuario que esté autenticado. Manipula la información personal del usuario, como son el nombre completo , compañía, departamento, correo electrónico, identificador de conexión, entre otros.
- **Adodb:** Adodb es un conjunto de librerías de bases de datos con soporte para varios lenguajes. Esta permite a los programadores desarrollar aplicaciones web de una manera portable, rápida y fácil. La ventaja reside en que la base de datos puede cambiar sin necesidad de reescribir cada consulta realizada por la aplicación. Son soportadas las siguientes bases de datos: MySQL , PostgreSQL, Interbase, Firebird, Informix, Oracle, MS SQL , Foxpro, Access, Sybase ASE, FrontBase, DB2, SQLite, Netezza y LDAP.

### 2. Librerías incluidas propias del módulo.

- **Clases:** En este caso las clases permiten organizar las funcionalidades que contiene un servicio. Dentro del directorio **class** se encuentran cada una de estas clases acotadas con el nombre correspondiente con la vista y el controlador que las relaciona por ejemplo la clase task maneja las funcionalidades del servicio tareas y se encuentra en el componente *task.service.class.php*.
- **Php2wsdl:** Esta librería se puede utilizar para generar un WSDL el documento a partir del código de una clase. Se analiza el código de los archivos dados en script PHP y genera una definición WSDL de la lista de las funciones de la clase.

Ciertas funciones se pueden omitir para no ser incluidos en la definición WSDL basado en el tipo de acceso a la función. El WSDL generado puede guardarse en un archivo determinado, devuelve como una cadena o sirve para descargar. Dentro del directorio **include** se encuentra esta librería. El componente que necesitamos utilizar se nombra *PHPParser.php*.

**Vista:** La vista es la responsable de recibir los datos del modelo y mostrarlos al usuario. Cada una de las vistas por lo general tiene asociado un controlador según la funcionalidad que realiza.

En el caso del módulo DOT-WS las vistas están agrupadas en varios directorios dentro del directorio **view** del módulo. Estos son:

- task
- project
- user

Todos los directorios correspondientes a las vistas cuentan con un componente que se encarga de gestionar todas las interfaces que muestra la vista en cuestión. Este componente es nombrado según la vista correspondiente si la vista es de tarea su nombre sería *task\_view.php* y en el caso de las vistas de proyecto y de usuario sería *project\_view.php* y *user\_view.php* respectivamente. En las vistas se gestionan las funciones de cada una de las mismas.

**Controlador:** Los controladores son los ficheros responsables de recibir desde la vista los eventos de entrada y hacer las peticiones al modelo. Los controladores están agrupados en varios directorios dentro del directorio **controller** del módulo. Estos son:

- task
- project
- user

En este directorio se encuentran todos los componentes controladores que se encuentran asociados a las vistas antes mencionadas. Para una mejor asociación entre los componentes *controladores* y los de las *vistas* se nombraron igual para lograr una

mejor localización. Ej: los controladores asociados al directorio de vista **task** se encuentran en el directorio **task** de los controladores y son acotados por el nombre seguido de *\_controller.php*.

## 2.6 Guía para la creación de un servicio web.

### 2.6.1 Uso del WSDL.

Los servicios web utilizan el XML como un lenguaje estándar de comunicación entre aplicaciones, por la fácil descripción y lectura de un servicio. El WSDL está escrito en lenguaje XML y tiene una estructura como muestra en el **Anexo 1**.

A continuación veremos los pasos a seguir para la creación de un servicio dentro del módulo DOT-WS.

Creación de un servicio:

- Crear un nuevo fichero dentro del directorio **class** con el nombre *name\_service.class.php* donde name sería el nombre del servicio seguido de *\_service*, dentro del mismo escribir una clase con el mismo nombre ejemplo *name\_service*.
- Agregar las funciones que se deseen con los comentarios correspondientes (**Anexo 2**).
- Generar el WSDL con el ZendStudio 5.5.0:
  1. Ir al menú herramientas/Generador WSDL.
  2. Poner el nombre de la configuración. Por defecto el nombre del fichero wsdl es el mismo.
  3. Ubicar la dirección del fichero en el directorio **wsdl** del módulo. Dar clic en Siguiente (**Anexo 3**).
  4. Añadir el fichero donde se encuentra la clase.
  5. Poner la dirección del url del controlador correspondiente donde se crea el servicio. Dar clic en Siguiente (**Anexo 4**).
  6. Aparece el Resumen de la configuración del WSDL a crear. Dar clic en Terminar (**Anexo 5**).

## 2.7 Guía para la creación de un módulo.

El SGP dotProject es distribuido bajo licencia GPL, lo cual nos permite reutilizar el código y reimplementar funcionalidades ya existentes o implementar nuevas de acuerdo a las necesidades del usuario. El mismo tiene la ventaja de estar implementado con una estructura modular. Esto facilita en gran medida los procesos de mejoras de este software en cuanto al aumento de las funcionalidades que posee. Para poder integrar el módulo DOT-WS al dotProject primeramente hay tener conocimientos relacionados con el proceso de creación e instalación de un módulo en dotProject.

A continuación veremos los pasos a seguir para la creación de módulos, esto nos dará la posibilidad de tener mayor organización y una homogeneidad a la hora de crear el módulo DOT-WS.

### Creación de un módulo para dotProject en seis pasos.

- Crear un directorio con el nombre del módulo.
- Las imágenes e iconos se deben guardar en un directorio /images dentro de ese mismo directorio.
- Desarrollar un script de instalación nombrado setup.php para instalar, actualizar, eliminar y configurar el módulo. A continuación se muestra el código fuente de un instalador de ejemplo:

```
<?php
$config = array();
$config['mod_name'] = 'Prueba'; // nombre del módulo
$config['mod_version'] = '0.1'; // especificar la versión del módulo
$config['mod_directory'] = 'prueba'; //define donde esta el directorio del módulo
$config['mod_setup_class'] = 'DP_Prueba';//el nombre de la clase instaladora que se define
más adelante en este fichero
$config['mod_type'] = 'user'; //tipo de módulo, como estándar se pone 'user' para los módulos
adicionales
$config['mod_ui_icon'] = 'prueba.jpg'; // nombre del icono o logo del módulo
$config['mod_description'] = 'Esto es un módulo de prueba'; // Pequeña descripción del
módulo
$config['mod_config'] = false; // determina si se muestra o no el link de configuración en la
vista de administración del módulo.

if (@$a == 'setup'){
    echo dPshowModuleConfig( $config );
}

class DP_Autos {
    function install()
    {
```

```
/*En esta función se especifican los requerimientos del nuevo módulo,
desde las tablas que necesite hasta la creación de directorios auxiliares*/
//ejemplo de la creación de una tabla para el módulo
```

```
$sql = '(
    prueba_id VARCHAR(45) NOT NULL,
    descripción TEXT NOT NULL,
    PRIMARY KEY(prueba_id)
)TYPE=MyISAM';
```

```
$q = new DBQuery;
$q->createTable('prueba');
$q->createDefinition($sql);
$q->exec();
```

```
}
```

```
function remove() {
```

```
/*forma de eliminar los recursos creados por el módulos y que eran necesarios solo para su
funcionamiento*/
```

```
//como eliminar la tabla del módulo
```

```
$q = new DBQuery;
$q->dropTable('prueba');
$q->exec();
```

```
}
```

```
}
```

```
?>
```

- Crear el fichero *index.php* donde se encuentra la interfaz general que brindará el módulo.
- Según las observaciones realizadas a todos los módulos del dotProject, se determinó los elementos comunes para los mismos en cuanto a la implementación de su interfaz general (*index.php*). Se concluyó que la manera utilizada por dotProject para mostrar los elementos de cabecera de los módulos es como se muestra en el siguiente fragmento de código fuente:

```
<?php
$titleBlock = new CTitleBlock($AppUI->_('Módulo de prueba'),'prueba.png', $m, "$m.
$a" );
$titleBlock->show();
?>
```

- Por último realizamos los pasos necesarios para la instalación de un módulo en dotProject. (**Anexo 6**).

El diseño propuesto a partir de los requerimientos planteados por el cliente cumple con los estándares que exige dotProject para la creación de módulos y la navegación en los mismos, se cumple además con los estándares de creación del fichero WSDL para descripción de los servicios web. El estilo arquitectónico MVC que

se propone para el módulo proporcionará robustez al mismo por garantizar que ante posibles cambios se afecten la menor cantidad de componente posibles. En otros aspectos se alcanzará una integración bastante eficiente al dotProject si seguimos las pautas que al final de este capítulo se plantean.

## CAPÍTULO 3 IMPLEMENTACIÓN DEL MÓDULO

En este capítulo se realiza la implementación de la solución propuesta en el capítulo anterior y se generan los artefactos correspondientes a la fase de desarrollo según la metodología seleccionada.

### 3.1 Planificación del proyecto por roles.

SXP define diferentes roles para lograr un exitoso resultado en el proceso de desarrollo de software, estos son:

**Tabla 3.**Planificación del proyecto por roles.

Rol	Nombre
Gerente	Yoani Santana Leiva
Cliente	SIMAYS. UCI Dirección de Producción #3. UCI
Programadores	Yoani Santana Leiva
Analista	Yoani Santana Leiva Sheila Castro Hernandez
Diseñadores	Yoani Santana Leiva
Encargado de Pruebas	Yoani Santana Leiva
Arquitecto	Yoani Santana Leiva

### 3.2 Historias de usuarios.

Las historias de usuario son la técnica utilizada por SXP para especificar los requerimientos de software. Las mismas son escritas por los clientes como las tareas que el sistema debe hacer y su construcción es en lenguaje natural, no excediendo su tamaño de unas pocas líneas de texto.

### Historia de usuario HHU-DOT-WS-01

Historia de Usuario	
<b>Número:</b> HHU-DOT-WS-01	<b>Nombre Historia de Usuario:</b> <i>Crear la clase task_service.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Contiene las funcionalidades del servicio tarea.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-02

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-02	<b>Nombre Historia de Usuario:</b> <i>Obtener la lista de las nuevas tareas de un usuario.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Muestra la lista de las nuevas tareas agregadas al dotproject dado un usuario determinado.	
<b>Observaciones:</b> Muestra todos los datos esenciales de una tarea.	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-03

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-03	<b>Nombre Historia de Usuario:</b> <i>Obtener la lista de todas las tareas de un usuario.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite obtener la lista de todas las tareas de dado un usuario.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-04

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-04	<b>Nombre Historia de Usuario:</b> <i>Actualizar el estado de una tarea.</i>
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite actualizar el estado de una tarea.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-05

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-05	<b>Nombre Historia de Usuario:</b> <i>Eliminar tarea.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite eliminar una tarea.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-06

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-06	<b>Nombre Historia de Usuario:</b> <i>Crear tarea.</i>
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> Ninguna
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite al usuario crear una tarea.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

**Historia de usuario HU-DOT-WS-07**

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-07	<b>Nombre Historia de Usuario:</b> <i>Modificar tarea.</i>
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> Ninguna
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 2
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 2.5
<b>Descripción:</b> Permite modificar una tarea.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

**Historia de usuario HU-DOT-WS-08**

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-08	<b>Nombre Historia de Usuario:</b> <i>Adicionar historial a una tarea.</i>
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite adicionar historial a una tarea.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

**Historia de usuario HU-DOT-WS-09**

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-09	<b>Nombre Historia de Usuario:</b> <i>Modificar historial de una tarea.</i>
<b>Modificación de Historia de Usuario Número:</b> 1	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite modificar el historial de una tarea.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-10

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-10	<b>Nombre Historia de Usuario:</b> <i>Eliminar historial de una tarea.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite eliminar el historial de una tarea.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-11

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-11	<b>Nombre Historia de Usuario:</b> <i>Listar las tareas según su estado.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite listar las tareas según el estado.	
<b>Observaciones:</b> El estado de una tarea puede ser Futura, En Proceso, Atrasado, Hecho, Debió haber comenzado.	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-12

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-12	<b>Nombre Historia de Usuario:</b> <i>Mostrar la descripción completa del servicio.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite mostrar la descripción de un servicio determinado	
<b>Observaciones:</b> En este caso existen tres servicios principales como son: tarea, proyecto y usuario.	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-13

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-13	<b>Nombre Historia de Usuario:</b> <i>Mostrar la descripción de las funciones del servicio.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite mostrar la descripción de las funciones del servicio	
<b>Observaciones:</b> Muestra todas las funciones que contiene la clase task_service junto con lo que reciben y devuelven sus funcionalidades.	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-14

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-14	<b>Nombre Historia de Usuario:</b> <i>Crear la clase project_service</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 0.5
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 0.5
<b>Descripción:</b> Permite crear una clase llamada project_service que contiene las funcionalidades que brinda el servicio proyecto.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-15

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-15	<b>Nombre Historia de Usuario:</b> <i>Listar todos los proyectos.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 0.5
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 0.5
<b>Descripción:</b> Permite obtener todos los proyectos.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-16

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-16	<b>Nombre Historia de Usuario:</b> <i>Obtener datos de proyecto dado un identificador.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 1
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite obtener datos de un proyecto dado un identificador.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-17

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-17	<b>Nombre Historia de Usuario:</b> <i>Obtener las tareas de un proyecto.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 0.5
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 0.5
<b>Descripción:</b> Permite obtener las tareas de un proyecto.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-18

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-18	<b>Nombre Historia de Usuario:</b> <i>Crear un nuevo proyecto.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 0.5
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 0.5
<b>Descripción:</b> Permite crear un nuevo proyecto.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-19

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-19	<b>Nombre Historia de Usuario:</b> <i>Modificar un proyecto.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 0.5
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 0.5
<b>Descripción:</b> Permite modificar un proyecto.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-20

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-20	<b>Nombre Historia de Usuario:</b> <i>Eliminar un proyecto.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 2
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 0.5
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 0.5
<b>Descripción:</b> Permite eliminar un proyecto.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-21

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-21	<b>Nombre Historia de Usuario:</b> <i>Crear la clase user_service</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite crear una clase llamada user_service que contiene las funcionalidades del servicio usuario.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-22

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-22	<b>Nombre Historia de Usuario:</b> <i>Listar todos los usuarios.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite obtener todos los usuarios presentes en la BD del dotproject.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-23

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-23	<b>Nombre Historia de Usuario:</b> <i>Obtener datos de un usuario.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite obtener los datos de un usuario.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-24

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-24	<b>Nombre Historia de Usuario:</b> <i>Insertar usuario.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite adicionar un usuario.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-25

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-25	<b>Nombre Historia de Usuario:</b> <i>Eliminar usuario.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite eliminar un usuario.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-26

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-26	<b>Nombre Historia de Usuario:</b> <i>Actualizar datos de usuario.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 3
<b>Prioridad en Negocio:</b> Alta	<b>Puntos Estimados:</b> 1
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 1
<b>Descripción:</b> Permite actualizar los datos de un usuario.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b>	

### Historia de usuario HU-DOT-WS-27

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-27	<b>Nombre Historia de Usuario:</b> <i>Crear instalador del módulo.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 4
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 2
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 2
<b>Descripción:</b> Permite la creación del instalador base del módulo DOT-WS. Se encarga de crear un módulo solo con las configuraciones necesarias para su instalación y sin interfaz alguna.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b> Anexo 7	

### Historia de usuario HU-DOT-WS-28

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-28	<b>Nombre Historia de Usuario:</b> <i>Crear vista de tarea.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 4
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 2
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 2
<b>Descripción:</b> Permite crear la vista de tarea. Esta vista contiene la descripción del servicio tarea y muestra las funcionalidades que el mismo brinda así como los parámetros de entrada y de salida de dichas funciones.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b> Anexo 8	

### Historia de usuario HU-DOT-WS-29

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-29	<b>Nombre Historia de Usuario:</b> <i>Crear vista de proyecto.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 4
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 2
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 2
<b>Descripción:</b> Permite crear la vista de proyecto. Esta vista contiene la descripción del servicio proyecto y muestra las funcionalidades que el mismo brinda así como los parámetros de entrada y de salida de dichas funciones.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b> Anexo 9	

### Historia de usuario HU-DOT-WS-30

Historia de Usuario	
<b>Número:</b> HU-DOT-WS-30	<b>Nombre Historia de Usuario:</b> <i>Crear vista de usuario.</i>
<b>Modificación de Historia de Usuario Número:</b> Ninguna	
<b>Usuario:</b> Yoani Santana Leiva	<b>Iteración Asignada:</b> 4
<b>Prioridad en Negocio:</b> Media	<b>Puntos Estimados:</b> 2
<b>Riesgo en Desarrollo:</b> Bajo	<b>Puntos Reales:</b> 2
<b>Descripción:</b> Permite crear la vista de usuario. Esta vista contiene la descripción del servicio usuario y muestra las funcionalidades que el mismo brinda así como los parámetros de entrada y de salida de dichas funciones.	
<b>Observaciones:</b>	
<b>Prototipo de interfaz:</b> Anexo 10	

### 3.3 Plan de versiones

La plantilla de Plan de versiones, es un artefacto generado por la metodología empleada durante el proceso de desarrollo. En este el cliente define el valor que posee el negocio según las características deseadas.

Esta plantilla proporciona ventajas, tales como:

- Define cuales son las historias de usuario más significativas, y las ubican en las iteraciones según esta prioridad.
- Divide el proceso de desarrollo de software en iteraciones, planificando el trabajo a realizar en cada una de ellas.

**Tabla 4.** Plan de releases.

Release	Descripción de la iteración	Orden de la HU a implementar	Duración total
Iteración 1	Se implementa la clase task_service junto con sus funcionalidades.	HU-DOT-WS-01 HU-DOT-WS-02 HU-DOT-WS-03 HU-DOT-WS-04 HU-DOT-WS-05 HU-DOT-WS-06 HU-DOT-WS-07 HU-DOT-WS-08 HU-DOT-WS-09 HU-DOT-WS-10 HU-DOT-WS-11 HU-DOT-WS-12 HU-DOT-WS-13	15-02-2010 al 23-03-2010
Iteración 2	Se implementa la clase project_service junto con sus funcionalidades.	HU-DOT-WS-14 HU-DOT-WS-15 HU-DOT-WS-16 HU-DOT-WS-17 HU-DOT-WS-18 HU-DOT-WS-19 HU-DOT-WS-20	23-03-2010 al 26-03-2010
Iteración 3	Se implementa la clase user_service junto con sus funcionalidades.	HU-DOT-WS-21 HU-DOT-WS-22 HU-DOT-WS-23 HU-DOT-WS-24 HU-DOT-WS-25 HU-DOT-WS-26	26-03-2010 al 31-03-2010
Iteración 4	Se crea el módulo DOT-WS. Se diseñan los prototipos de interfaz: tarea, proyecto y usuario.	HU-DOT-WS-27 HU-DOT-WS-28 HU-DOT-WS-29 HU-DOT-WS-30	01-03-2010 al 15-03-2010

### 3.4 Tareas de Ingeniería.

En la fase de desarrollo de la metodología SXP se propone la entrega de las tareas realizadas para dar cumplimiento a cada historia de usuario. Estas tareas son llamadas Tareas de Ingeniería.

#### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-01

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-01
<b>Nombre Tarea:</b> <i>Crear la clase task_service.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 16-03-2010	<b>Fecha Fin:</b> 16-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Se crea la clase task_service que contiene las funcionalidades del servicio tarea.	
<b>Observación:</b>	

#### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-02

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-02
<b>Nombre Tarea:</b> <i>Obtener la lista de las nuevas tareas de un usuario.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 17-03-2010	<b>Fecha Fin:</b> 17-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Se obtiene la lista de las tareas dado un usuario.	
<b>Observación:</b>	

#### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-03

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-03
<b>Nombre Tarea:</b> <i>Obtener la lista de todas las tareas de un usuario.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 17-03-2010	<b>Fecha Fin:</b> 17-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Dado un usuario se listan todas las tareas del mismo.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-04**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-04
<b>Nombre Tarea:</b> <i>Actualizar el estado de una tarea.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 18-03-2010	<b>Fecha Fin:</b> 18-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Actualiza el estado de una tarea.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-05**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-05
<b>Nombre Tarea:</b> <i>Eliminar tarea.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 18-03-2010	<b>Fecha Fin:</b> 18-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Elimina una tarea.	
<b>Observación:</b> Se eliminarán todos los ficheros existentes.	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-06**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-06
<b>Nombre Tarea:</b> <i>Crear tarea.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 19-03-2010	<b>Fecha Fin:</b> 19-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Diseño de la base de datos del módulo DOT-WS .	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-07**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-07
<b>Nombre Tarea:</b> <i>Modificar tarea.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 19-03-2010	<b>Fecha Fin:</b> 19-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Modifica los datos de una tarea.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-08**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-08
<b>Nombre Tarea:</b> <i>Adicionar historial a una tarea.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 20-03-2010	<b>Fecha Fin:</b> 20-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Adiciona historial a una tarea.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-09**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-09
<b>Nombre Tarea:</b> <i>Modificar historial de una tarea.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 20-03-2010	<b>Fecha Fin:</b> 20-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Modifica los datos del historial de una tarea.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-10**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-10
<b>Nombre Tarea:</b> <i>Eliminar historial de una tarea.</i>	
<b>Tipo de Tarea:</b> Estudio	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 21-03-2010	<b>Fecha Fin:</b> 21-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Elimina el historial de una tarea.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-11**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-11
<b>Nombre Tarea:</b> <i>Listar las tareas según su estado.</i>	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 21-03-2010	<b>Fecha Fin:</b> 21-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Lista las tareas según su estado.	
<b>Observación:</b>	

### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-12

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-12
<b>Nombre Tarea:</b> <i>Mostrar la descripción completa del servicio.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 22-03-2010	<b>Fecha Fin:</b> 22-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> muestra la descripción del servicio correspondiente.	
<b>Observación:</b>	

### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-13

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-13
<b>Nombre Tarea:</b> <i>Mostrar la descripción de las funciones del servicio.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 22-03-2010	<b>Fecha Fin:</b> 23-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Muestra la descripción de las funciones del servicio.	
<b>Observación:</b>	

### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-14

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-14
<b>Nombre Tarea:</b> <i>Crear la clase project_service</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 24-03-2010	<b>Fecha Fin:</b> 24-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Creación de la clase project_service que contiene todas las funciones del servicio proyecto.	
<b>Observación:</b>	

### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-15

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-15
<b>Nombre Tarea:</b> <i>Listar todos los proyectos.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 24-03-2010	<b>Fecha Fin:</b> 24-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Lista todos los proyectos.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-16**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-16
<b>Nombre Tarea:</b> <i>Obtener datos de proyecto dado un identificador.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
Fecha Inicio: 24-03-2010	Fecha Fin: 24-03-2010
Programador Responsable: Yoani Santana Leiva.	
Descripción: Obtiene los datos de un proyecto dado un identificador.	
Observación:	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-17**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-17
<b>Nombre Tarea:</b> <i>Obtener las tareas de un proyecto.</i>	
Tipo de Tarea : Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 25-03-2010	<b>Fecha Fin:</b> 25-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Obtiene las tareas de un proyecto.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-18**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-18
<b>Nombre Tarea:</b> <i>Crear un nuevo proyecto.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 25-03-2010	<b>Fecha Fin:</b> 25-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> crea un nuevo proyecto.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-19**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-19
<b>Nombre Tarea:</b> <i>Modificar un proyecto.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 26-03-2010	<b>Fecha Fin:</b> 26-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Modifica los datos de un proyecto.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-20**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-20
<b>Nombre Tarea:</b> <i>Eliminar un proyecto.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 26-03-2010	<b>Fecha Fin:</b> 26-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Elimina un proyecto.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-21**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-21
<b>Nombre Tarea:</b> <i>Crear la clase user_service</i>	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 26-03-2010	<b>Fecha Fin:</b> 26-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Creación de la clase user_service que contiene las funciones del servicio usuario.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-22**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-22
<b>Nombre Tarea:</b> <i>Listar todos los usuarios.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 27-03-2010	<b>Fecha Fin:</b> 27-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Lista todos los usuarios.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-23**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-23
<b>Nombre Tarea:</b> <i>Obtener datos de un usuario.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 28-03-2010	<b>Fecha Fin:</b> 28-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Muestra los datos de un usuario.	
<b>Observación:</b>	

### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-24

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-24
<b>Nombre Tarea:</b> <i>Insertar usuario.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 29-03-2010	<b>Fecha Fin:</b> 28-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Adiciona los datos de un usuario.	
<b>Observación:</b>	

### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-25

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-25
<b>Nombre Tarea:</b> <i>Eliminar usuario.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 30-03-2010	<b>Fecha Fin:</b> 30-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Elimina un usuario.	
<b>Observación:</b>	

### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-26

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-26
<b>Nombre Tarea:</b> <i>Actualizar datos de usuario.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.1
<b>Fecha Inicio:</b> 31-03-2010	<b>Fecha Fin:</b> 31-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Actualiza los datos de un usuario.	
<b>Observación:</b>	

### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-27

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-27
<b>Nombre Tarea:</b> <i>Implementar el instalador del módulo.</i>	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 01-03-2010	<b>Fecha Fin:</b> 03-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Implementar el instalador del módulo.	
<b>Observación:</b>	

### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-28

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-28
<b>Nombre Tarea:</b> Diseño e implementación de la vista de tarea.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 03-03-2010	<b>Fecha Fin:</b> 04-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Diseño e implementación de la vista de tarea. Se muestra la descripción del servicio y sus funcionalidades con su descripción, además de los parámetros de entrada y lo que las mismas devuelven.	
<b>Observación:</b>	

### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-28

Tarea de Ingeniería	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> HU-DOT-WS-28
<b>Nombre Tarea:</b> Implementación del controlador de la vista de tarea.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 05-02-2009	<b>Fecha Fin:</b> 06-02-2009
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Creación de un fichero controlador que permita crear el servicio tarea así como gestionar todos sus datos y funcionalidades.	
<b>Observación:</b>	

### Tareas de Ingeniería para la historia de usuario HU-DOT-WS-29

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-29
<b>Nombre Tarea:</b> Implementación del controlador de la vista de proyecto.	
<b>Tipo de Tarea:</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 07-03-2010	<b>Fecha Fin:</b> 08-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Creación de un fichero controlador que permita crear el servicio proyecto así como gestionar todos sus datos y funcionalidades.	
<b>Observación:</b>	

Tarea de Ingeniería	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> HU-DOT-WS-29
<b>Nombre Tarea:</b> Diseño e implementación de la vista de proyecto.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 09-03-2010	<b>Fecha Fin:</b> 10-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Diseño e implementación de la vista de proyecto. Se muestra la descripción del servicio y sus funcionalidades con su descripción, además de los parámetros de entrada y lo que las mismas devuelven.	
<b>Observación:</b>	

**Tareas de Ingeniería para la historia de usuario HU-DOT-WS-30**

Tarea de Ingeniería	
<b>Número Tarea:</b> 1	<b>Número Historia de Usuario:</b> HU-DOT-WS-30
<b>Nombre Tarea:</b> Diseño e implementación de la vista de usuario.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 11-03-2010	<b>Fecha Fin:</b> 13-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Diseño e implementación de la vista de usuario. Se muestra la descripción del servicio y sus funcionalidades con su descripción, además de los parámetros de entrada y lo que las mismas devuelven.	
<b>Observación:</b>	

Tarea de Ingeniería	
<b>Número Tarea:</b> 2	<b>Número Historia de Usuario:</b> HU-DOT-WS-30
<b>Nombre Tarea:</b> Implementación del controlador de la vista de usuario.	
<b>Tipo de Tarea :</b> Desarrollo	<b>Puntos Estimados:</b> 0.5
<b>Fecha Inicio:</b> 14-03-2010	<b>Fecha Fin:</b> 15-03-2010
<b>Programador Responsable:</b> Yoani Santana Leiva.	
<b>Descripción:</b> Creación de un fichero controlador que permita crear el servicio usuario así como gestionar todos sus datos y funcionalidades.	
<b>Observación:</b>	

La implementación del módulo DOT-WS se desarrolló satisfactoriamente a partir del diseño planteado en el capítulo anterior. Se cumplieron con las exigencias de los estándares exigidos por el framework dotProject. Solo basta realizarle las pruebas de aceptación pertinentes para validar su correcta implementación.

## CAPÍTULO 4 VALIDACIÓN DEL MÓDULO

Los casos de prueba de aceptación tienen como objetivo validar que el sistema cumpla con el funcionamiento esperado y permitir al cliente determinar su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

### 4.1 Casos de Pruebas

Las pruebas de aceptación son definidas por el cliente y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al cliente. La utilización de estas, proporcionan grandes ventajas, permitiendo a los programadores principalmente medir la calidad de su trabajo y garantizar la entrega de un producto con calidad y en correspondencia con las necesidades del cliente.

#### Caso de Prueba de la historia de usuario HU-DOT-WS-02

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU-DOT-WS-02-01	<b>Nombre Historia de Usuario:</b> HU-DOT-WS-02
<b>Nombre de la persona que realiza la prueba:</b> Yoani Santana Leiva	
<b>Descripción de la Prueba:</b> Dado un usuario se debe obtener la lista de las tareas nuevas existentes en el dotProject.	
<b>Condiciones de Ejecución:</b> Estar conectado al WSDL del servicio Tarea	
<b>Entrada / Pasos de ejecución:</b> Se especifica el usuario.	
<b>Resultado Esperado:</b> En caso de existir nuevas tareas deben mostrarse en forma de arreglo y si no el arreglo que se obtiene es vacío.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

#### Caso de Prueba de la historia de usuario HU-DOT-WS-03

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU-DOT-WS-03-01	<b>Nombre Historia de Usuario:</b> HU-DOT-WS-03
<b>Nombre de la persona que realiza la prueba:</b> Yoani Santana Leiva	
<b>Descripción de la Prueba:</b> Dado un usuario se debe obtener la lista de todas las tareas existentes en el dotProject.	
<b>Condiciones de Ejecución:</b> Estar conectado al WSDL del servicio Tarea	
<b>Entrada / Pasos de ejecución:</b> Se especifica el usuario.	
<b>Resultado Esperado:</b> En caso de existir tareas debe mostrarse un arreglo y si no el arreglo que se obtiene es vacío.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

### Caso de Prueba de la historia de usuario HU-DOT-WS-11

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU-DOT-WS-11-01	<b>Nombre Historia de Usuario:</b> HU-DOT-WS-11
<b>Nombre de la persona que realiza la prueba:</b> Yoani Santana Leiva	
<b>Descripción de la Prueba:</b> Debe mostrar las tareas existentes según el estado dado por parámetro.	
<b>Condiciones de Ejecución:</b> Estar conectado al WSDL del servicio Tarea	
<b>Entrada / Pasos de ejecución:</b> Se especifica el estado de las tareas que se quiere obtener.	
<b>Resultado Esperado:</b> En caso de existir tareas con el estado dado por parámetro debe obtenerse un arreglo y si no el arreglo que se obtiene es vacío.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

### Caso de Prueba de la historia de usuario HU-DOT-WS-12

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU-DOT-WS-12-01	<b>Nombre Historia de Usuario:</b> HU-DOT-WS-12
<b>Nombre de la persona que realiza la prueba:</b> Yoani Santana Leiva	
<b>Descripción de la Prueba:</b> Debe mostrar la descripción WSDL de un servicio.	
<b>Condiciones de Ejecución:</b>	
<b>Entrada / Pasos de ejecución:</b> Dar clic en el vínculo Ver WSDL	
<b>Resultado Esperado:</b> Muestra un popup con la descripción WSDL del servicio correspondiente a esa vista.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

### Caso de Prueba de la historia de usuario HU-DOT-WS-15

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU-DOT-WS-15-01	<b>Nombre Historia de Usuario:</b> HU-DOT-WS-15
<b>Nombre de la persona que realiza la prueba:</b> Yoani Santana Leiva	
<b>Descripción de la Prueba:</b> Debe listar todos los proyectos existentes en el dotProject.	
<b>Condiciones de Ejecución:</b> Estar conectado al WSDL del servicio Proyecto.	
<b>Entrada / Pasos de ejecución:</b>	
<b>Resultado Esperado:</b> Muestra un arreglo con la lista de los proyectos existentes en el dotProject.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Caso de Prueba de la historia de usuario HU-DOT-WS-16**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU-DOT-WS-16-01	<b>Nombre Historia de Usuario:</b> HU-DOT-WS-16
<b>Nombre de la persona que realiza la prueba:</b> Yoani Santana Leiva	
<b>Descripción de la Prueba:</b> Debe mostrar los datos de un proyecto dado su identificador.	
<b>Condiciones de Ejecución:</b> Estar conectado el WSDL del servicio Proyecto.	
<b>Entrada / Pasos de ejecución:</b>	
<b>Resultado Esperado:</b> Muestra un arreglo con la lista de los proyectos existentes en el dotProject.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Caso de Prueba de la historia de usuario HU-DOT-WS-22**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU-DOT-WS-22-01	<b>Nombre Historia de Usuario:</b> HU-DOT-WS-22
<b>Nombre de la persona que realiza la prueba:</b> Yoani Santana Leiva	
<b>Descripción de la Prueba:</b> Debe listar todos los usuarios existentes en el dotProject.	
<b>Condiciones de Ejecución:</b>	
<b>Entrada / Pasos de ejecución:</b>	
<b>Resultado Esperado:</b> Muestra un arreglo con la lista de los usuarios existentes en el dotProject.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

**Caso de Prueba de la historia de usuario HU-DOT-WS-23**

<b>Caso de Prueba de Aceptación</b>	
<b>Código Caso de Prueba:</b> HU-DOT-WS-23-01	<b>Nombre Historia de Usuario:</b> HU-DOT-WS-23
<b>Nombre de la persona que realiza la prueba:</b> Yoani Santana Leiva	
<b>Descripción de la Prueba:</b> Debe listar los datos de un usuario dado el identificador en el dotProject.	
<b>Condiciones de Ejecución:</b> Estar conectado al WSDL del servicio usuario.	
<b>Entrada / Pasos de ejecución:</b> Se especifica el identificador del usuario.	
<b>Resultado Esperado:</b> Muestra un arreglo con los datos del usuario si existe.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

### Caso de Prueba de la historia de usuario HU-DOT-WS-28

Caso de Prueba de Aceptación	
<b>Código Caso de Prueba:</b> HU-DOT-WS-28-01	<b>Nombre Historia de Usuario:</b> HU-DOT-WS-28
<b>Nombre de la persona que realiza la prueba:</b> Yoani Santana Leiva	
<b>Descripción de la Prueba:</b> Debe visualizar la vista de tarea en la que se describe el servicio <code>task_service</code> junto con sus métodos.	
<b>Condiciones de Ejecución:</b> Estar autenticado en el dotProject.	
<b>Entrada / Pasos de ejecución:</b>	
<b>Resultado Esperado:</b> Se muestra la interfaz de la vista de tarea.	
<b>Evaluación de la Prueba:</b> Prueba satisfactoria.	

### 4.2 Resultados Obtenidos.

Como resultado de este trabajo DOT-WS queda disponible en su versión 1.0. Se obtuvo un módulo que cumple con todas las especificaciones de dotProject desde el punto de vista de diseño. Se logró un alto grado de optimización utilizándose los componentes propios del framework dotProject logrando una reutilización del código a gran escala.

### 4.3 Funcionalidades obtenidas.

A continuación se muestra una tabla con la descripción de las funcionalidades obtenidas clasificadas por las vistas que contiene el módulo.

**Tabla 5.** Descripción de las funcionalidades obtenidas

Vista	Descripción
Tarea	Muestra la descripción el servicio <code>task_service</code> , un vínculo al fichero WSDL, y la lista de las funcionalidades del servicio implementada junto con los parámetros de entrada y de salida.
Proyecto	Muestra la descripción el servicio <code>project_service</code> , un vínculo al fichero WSDL, y la lista de las funcionalidades del servicio implementada junto con los parámetros de entrada y de salida.
Usuario	Muestra la descripción el servicio <code>user_service</code> , un vínculo al fichero WSDL, y la lista de las funcionalidades del servicio implementada junto con los parámetros de entrada y de salida.

## **CONCLUSIONES**

Podemos observar que la Gestión de Proyectos siempre ha estado en continua evolución y actualmente está más activa que nunca, teniendo por delante un futuro prometedor donde la innovación jugará un papel decisivo. En el presente trabajo se expusieron elementos de gran importancia relacionados con la gestión de proyectos y los SGP como herramientas web capaces de automatizar los diversos procesos que la forman y los servicios web que le brindan al dotProject interoperabilidad con otras aplicaciones. Se desarrolló el módulo DOT-WS aplicando el estilo arquitectónico Modelo-Vista-Controlador, usando los estándares adecuados para el desarrollo de servicios web, brindándole funcionalidades al dotProject que permiten la interoperabilidad con otras aplicaciones.

## **RECOMENDACIONES**

- Integrar el dotProject a la plataforma de migración.
- Agregar nuevos servicios web al módulo DOT-WS.

**REFERENCIAS BIBLIOGRÁFICAS**

1. activeCollab. activeCollab - Project Management and Collaboration Tool. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.activecollab.com/>>.
2. Alejandro Barros. Sistemas Colaborativos de gestión de proyectos. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.alejandrobarrros.com/content/view/134919/Sistemas-Colaborativos-de-gestion-de-proyectos.html>>.
3. Angel Alvarez, Miguel . Concepto de JavaScript - MDC. 2009. [cited 2 December 2009]. Available from world wide web: <[https://developer.mozilla.org/index.php?title=Es/Gu%C3%ADa\\_JavaScript\\_1.5/Concepto\\_de\\_JavaScript](https://developer.mozilla.org/index.php?title=Es/Gu%C3%ADa_JavaScript_1.5/Concepto_de_JavaScript)>.
4. Angel Nieva López. Innovación en la Gestión de Proyectos. Una historia en evolución - Noticias.com. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.noticias.com/opinion/innovacion-gestion-proyectos-historia-evolucion-9h8.html>>.
5. Arleco. GanttProject: una buena alternativa de programas de proyectos - AplicacionesEmpresariales.com. 2008. [cited 2 December 2009]. Available from world wide web: <<http://www.aplicacionesempresariales.com/ganttproject-una-buena-alternativa-de-programas-de-proyectos.html>>.
6. Arlecob. MSProject.com - Expertise in Microsoft Project Training Consultancy and Solutions including Microsoft Project Office 2003 and Microsoft Project EPM solution. 2008. [cited 2 December 2009]. Available from world wide web: <<http://www.msproject.com/>>.
7. Arq. Percy C. Acuña Vigil. Gestión de proyectos de desarrollo urbano | Urbano Perú: Perullaqta, E. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.urbanoperu.com/node/360>>.
8. w3c CSS. Cascading Style Sheets. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.w3.org/Style/CSS/>>.
9. Cynertia Consulting. CMMI. PMBOK. PRINCE2. Project Management, oficina de proyectos, gestión de proyectos Barcelona, Cataluña, España. 2009. [cited 10 December 2009]. Available from world wide web: <[http://www.cynertiaconsulting.com/1-220\\_sistemas\\_gestion\\_proyectos.html](http://www.cynertiaconsulting.com/1-220_sistemas_gestion_proyectos.html)>.
10. Ernesto Serrano. Propuesta de utilización de DotProject como herramienta de software libre para la Gestión de Proyectos Socio - Técnicos - Comunitarios 2009. Available from world wide web: <[http://www.laccei.org/LACCEI2009-Venezuela/Papers/ExtendedAbstract/EA073\\_Pariata.pdf](http://www.laccei.org/LACCEI2009-Venezuela/Papers/ExtendedAbstract/EA073_Pariata.pdf)>.
11. Gartner Research. Gartner Technology Business Research Insight. 2009. [cited 2 April 2010]. Available from world wide web: <<http://www.gartner.com/technology/home.jsp>>.

12. Gladys Marsi Peñalver Romero. MA-GMPR-UR2 Metodología ágil para proyectos de software libre. 2008, 11-50.
13. Henry Lawrence. Gantt Chart Tutorial: Timelines and Project Schedules - How To Create. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.gantt-chart.biz/gantt-charting-made-easy/>>.
14. Java. Java: un lenguaje multiplataforma. 2009. [cited 2 April 2010]. Available from world wide web: <<http://www.dtic.upf.edu/~vlopez/docencia/castellano/todoshtml/Plt3cas/node16.html>>.
15. linux.ciberaula.com. Una Introducción a Apache. 2009. [cited 2 April 2010]. Available from world wide web: <[http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/)>.
16. netpecos.org. MySQL. 2009. [cited 2 April 2010]. Available from world wide web: <[http://www.netpecos.org/docs/mysql\\_postgres/x57.html](http://www.netpecos.org/docs/mysql_postgres/x57.html)>.
17. Nusoap 2008. Servicios Web con PHP (NuSOAP). 2008. [cited 9 March 2010]. Available from world wide web: <<http://www.nociondigital.com/webmasters/php-tutorial-servicios-web-con-php-nusoap-detalle-168.html>>.
18. php.net. PHP: Hypertext Preprocessor. 2009. [cited 2 December 2009]. Available from world wide web: <<http://php.net/>>.
19. Planner team. Planner - GNOME Live! 2009. [cited 2 December 2009]. Available from world wide web: <<http://live.gnome.org/Planner>>.
20. Roy Thomas Fielding. Architectural Styles and the Design of Network-based Software Architectures. 2000. [cited 2 December 2009]. Available from world wide web: <<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>.
21. Salva Castro. OpenProj 1.01, un MS Project open source que promete. 2008. [cited 2 December 2009]. Available from world wide web: <<http://www.genbeta.com/herramientas/openproj-101-un-ms-project-open-source-que-promete>>.
22. Thompson. CONCEPTO DE PROYECTO. 2009. [cited 10 December 2009]. Available from world wide web: <<http://www.promonegocios.net/proyecto/concepto-proyecto.html>>.
23. w3c UDDI. WSDL and UDDI. 2009. [cited 2 December 2009]. Available from world wide web: <[http://www.w3schools.com/WSDL/wsd\\_uddi.asp](http://www.w3schools.com/WSDL/wsd_uddi.asp)>.
24. w3c. Guía Breve de Servicios Web. 2008. [cited 2 December 2009]. Available from world wide web: <<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>>.
25. w3c html. W3C XHTML2 Working Group Home Page. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.w3.org/Markup/>>.
26. w3c soap. SOAP Specifications. 2007. [cited 2 December 2009]. Available from world wide web: <<http://www.w3.org/TR/soap/>>.

27. Web Services Journal. Computer Database Document. 2009. [cited 10 December 2009]. Available from world wide web: <[http://find.galegroup.com/gtx/retrieve.do?contentSet=IAC-Documents&resultListType=RESULT\\_LIST&qrySerId=Locale%28es%2C%2C%29%3AFQE%3D%28KE%2CNone%2C12%29web+services%3AAnd%3AFQE%3D%28TX%2CNone%2C3%29w3c%3AAnd%3AFQE%3D%28TX%2CNone%2C23%29concept+of+web+services%24&sgHitCountType=None&inPS=true&sort=DateDescend&searchType=BasicSearchForm&tabID=T003&prodId=CDB&searchId=R7&currentPosition=21&userGroupName=ucinf&docId=A99326393&docType=IAC](http://find.galegroup.com/gtx/retrieve.do?contentSet=IAC-Documents&resultListType=RESULT_LIST&qrySerId=Locale%28es%2C%2C%29%3AFQE%3D%28KE%2CNone%2C12%29web+services%3AAnd%3AFQE%3D%28TX%2CNone%2C3%29w3c%3AAnd%3AFQE%3D%28TX%2CNone%2C23%29concept+of+web+services%24&sgHitCountType=None&inPS=true&sort=DateDescend&searchType=BasicSearchForm&tabID=T003&prodId=CDB&searchId=R7&currentPosition=21&userGroupName=ucinf&docId=A99326393&docType=IAC)>.
28. w3c WSDL. Web Service Definition Language (WSDL). 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.w3.org/TR/wsdl>>.
29. www.um.es. Universidad de Murcia. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.um.es/>>.
30. w3c XML. Extensible Markup Language (XML). 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.w3.org/XML/>>.
31. XML-RPC. XML-RPC Home Page. 2008. [cited 2 December 2009]. Available from world wide web: <<http://www.xmlrpc.com/>>.
32. Zend. Creando un Web Service con PHP | Programando por diversión. 2009. [cited 2 April 2010]. Available from world wide web: <<http://copstone.com/2010/03/creando-un-web-service-con-php/>>.

**BIBLIOGRAFÍA**

1. NuSOAP - SOAP Toolkit for PHP | Get NuSOAP - SOAP Toolkit for PHP at SourceForge.net. [cited 10 November 2009]. Available from world wide web: <<http://sourceforge.net/projects/nusoap/>>.
2. Protocolos de comunicaciones. [cited 1 December 2009]. Available from world wide web: <<http://html.rincondelvago.com/protocolos-de-comunicaciones.html>>.
3. Web Services. [cited 1 December 2009]. Available from world wide web: <[http://74.125.93.132/search?q=cache:2B8L\\_Sm5riwJ:www.uvmsf.cl/~ssanchez/images/Aportes/Monografia\\_de\\_Web\\_Services.pdf+%C2%BFque+importancia+tienen+los+web+services%3F&cd=1&hl=es&ct=clnk&gl=cu](http://74.125.93.132/search?q=cache:2B8L_Sm5riwJ:www.uvmsf.cl/~ssanchez/images/Aportes/Monografia_de_Web_Services.pdf+%C2%BFque+importancia+tienen+los+web+services%3F&cd=1&hl=es&ct=clnk&gl=cu)>.
4. Web Services. [cited 24 November 2009]. Available from world wide web: <[http://74.125.93.132/search?q=cache:2B8L\\_Sm5riwJ:www.uvmsf.cl/~ssanchez/images/Aportes/Monografia\\_de\\_Web\\_Services.pdf+definicion+de+web+services+%2B+universidad+madrid&cd=7&hl=es&ct=clnk&gl=cu](http://74.125.93.132/search?q=cache:2B8L_Sm5riwJ:www.uvmsf.cl/~ssanchez/images/Aportes/Monografia_de_Web_Services.pdf+definicion+de+web+services+%2B+universidad+madrid&cd=7&hl=es&ct=clnk&gl=cu)>.
5. activeCollab. activeCollab - Project Management and Collaboration Tool. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.activecollab.com/>>.
6. Alejandro Barros. Sistemas Colaborativos de gestión de proyectos. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.alejandrobarrros.com/content/view/134919/Sistemas-Colaborativos-de-gestion-de-proyectos.html>>.
7. Angel Alvarez, Miguel . Concepto de JavaScript - MDC. 2009. [cited 2 December 2009]. Available from world wide web: <[https://developer.mozilla.org/index.php?title=Es/Gu%C3%ADa\\_JavaScript\\_1.5/Concepto\\_de\\_JavaScript](https://developer.mozilla.org/index.php?title=Es/Gu%C3%ADa_JavaScript_1.5/Concepto_de_JavaScript)>.
8. Angel Nieva López. Innovación en la Gestión de Proyectos. Una historia en evolución - Noticias.com. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.noticias.com/opinion/innovacion-gestion-proyectos-historia-evolucion-9h8.html>>.
9. Arleco. GanttProject: una buena alternativa de programas de proyectos - AplicacionesEmpresariales.com. 2008. [cited 2 December 2009]. Available from world wide web: <<http://www.aplicacionesempresariales.com/ganttproject-una-buena-alternativa-de-programas-de-proyectos.html>>.
10. Arlecob. MSProject.com - Expertise in Microsoft Project Training Consultancy and Solutions including Microsoft Project Office 2003 and Microsoft Project EPM solution. 2008. [cited 2 December 2009]. Available from world wide web: <<http://www.msproject.com/>>.
11. Arq. Percy C. Acuña Vigil. Gestión de proyectos de desarrollo urbano | Urbano Perú: Perullaqta, E. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.urbanoperu.com/node/360>>.
12. w3c CSS. Cascading Style Sheets. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.w3.org/Style/CSS/>>.

13. Cynertia Consulting. CMMi. PMBOK. PRINCE2. Project management, oficina de proyectos, gestión de proyectos Barcelona, Cataluña, España. 2009. [cited 10 December 2009]. Available from world wide web: <[http://www.cynertiaconsulting.com/1-220\\_sistemas\\_gestion\\_proyectos.html](http://www.cynertiaconsulting.com/1-220_sistemas_gestion_proyectos.html)>.
14. Ernesto Serrano. Propuesta de utilización de DotProject como herramienta de software libre para la Gestión de Proyectos Socio - Técnicos - Comunitarios . 2009. Available from world wide web: <[http://www.laccei.org/LACCEI2009-Venezuela/Papers/ExtendedAbstract/EA073\\_Pariata.pdf](http://www.laccei.org/LACCEI2009-Venezuela/Papers/ExtendedAbstract/EA073_Pariata.pdf)>.
15. Gartner Research. Gartner Technology Business Research Insight. 2009. [cited 2 April 2010]. Available from world wide web: <<http://www.gartner.com/technology/home.jsp>>.
16. Gladys Marsi Peñalver Romero. MA-GMPR-UR2 Metodología ágil para proyectos de software libre. . 2008, 11-50.
17. glosarium. Hipermedios - Diccionario Informático. diccionarios online. glosarios de terminos. Hipermedios - Diccionario Informático. 2010. [cited 2 April 2010]. Available from world wide web: <<http://www.glosarium.com/term/727,14,xhtml>>.
18. Henry Lawrence. Gantt Chart Tutorial: Timelines and Project Schedules - How To Create. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.gantt-chart.biz/gantt-charting-made-easy/>>.
19. Java. Java: un lenguaje multiplataforma. 2009. [cited 2 April 2010]. Available from world wide web: <<http://www.dtic.upf.edu/~vlopez/docencia/castellano/todoshtml/Plt3cas/node16.html>>.
20. linux.ciberaula.com. Una Introducción a Apache. 2009. [cited 2 April 2010]. Available from world wide web: <[http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/)>.
21. Microsoft. Web services .NET, el 'esperanto' de la red. 2008. [cited 26 November 2009]. Available from world wide web: <[http://www.microsoft.com/spain/enterprise/perspectivas/numero\\_2/tendencias.msp](http://www.microsoft.com/spain/enterprise/perspectivas/numero_2/tendencias.msp)>.
22. netbeans.org. Welcome to NetBeans. 2010a. [cited 6 April 2010]. Available from world wide web: <<http://netbeans.org/>>.
23. netbeans.org. Welcome to NetBeans. 2010b. [cited 5 April 2010]. Available from world wide web: <<http://netbeans.org/>>.
24. netpecos.org. MySQL. 2009. [cited 2 April 2010]. Available from world wide web: <[http://www.netpecos.org/docs/mysql\\_postgres/x57.html](http://www.netpecos.org/docs/mysql_postgres/x57.html)>.
25. Nusoap 2008. Servicios Web con PHP (NuSOAP). 2008. [cited 9 March 2010]. Available from world wide web: <<http://www.nociondigital.com/webmasters/php-tutorial-servicios-web-con-php-nusoap-detalle-168.html>>.
26. php.net. PHP: Hypertext Preprocessor. 2009. [cited 2 December 2009]. Available from world wide web: <<http://php.net/>>.
27. Planner team. Planner - GNOME Live! 2009. [cited 2 December 2009]. Available from world wide web: <<http://live.gnome.org/Planner>>.

28. proyecto. Definición de proyecto - Qué es, Significado y Concepto. 2010. [cited 2 April 2010]. Available from world wide web: <<http://definicion.de/proyecto/>>.
29. Robert Richards. *Pro PHP-XML and Web Service*. 2006.
30. Rolando Alfredo Hernández León. *El Paradigma Cuantitativo de la Investigación Científica*. [UCI]: Universitaria, 2002.
31. Roy Thomas Fielding. Architectural Styles and the Design of Network-based Software Architectures. 2000. [cited 2 December 2009]. Available from world wide web: <<http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm>>.
32. Salva Castro. OpenProj 1.01, un MS Project open source que promete. 2008. [cited 2 December 2009]. Available from world wide web: <<http://www.genbeta.com/herramientas/openproj-101-un-ms-project-open-source-que-promete>>.
33. Thompson. CONCEPTO DE PROYECTO. 2009. [cited 10 December 2009]. Available from world wide web: <<http://www.promonegocios.net/proyecto/concepto-proyecto.html>>.
34. w3c UDDI. WSDL and UDDI. 2009. [cited 2 December 2009]. Available from world wide web: <[http://www.w3schools.com/WSDL/wsdl\\_uddi.asp](http://www.w3schools.com/WSDL/wsdl_uddi.asp)>.
35. w3c. Guía Breve de Servicios Web. 2008. [cited 2 December 2009]. Available from world wide web: <<http://www.w3c.es/Divulgacion/GuiasBreves/ServiciosWeb>>.
36. w3c html. W3C XHTML2 Working Group Home Page. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.w3.org/MarkUp/>>.
37. w3c soap. SOAP Specifications. 2007. [cited 2 December 2009]. Available from world wide web: <<http://www.w3.org/TR/soap/>>.
38. Web Services Journal. Computer Database Document. 2009. [cited 10 December 2009]. Available from world wide web:
39. w3c WSDL. Web Service Definition Language (WSDL). 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.w3.org/TR/wsdl>>.
40. www.um.es. Universidad de Murcia. 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.um.es/>>.
41. w3c XML. Extensible Markup Language (XML). 2009. [cited 2 December 2009]. Available from world wide web: <<http://www.w3.org/XML/>>.
42. XML-RPC. XML-RPC Home Page. 2008. [cited 2 December 2009]. Available from world wide web: <<http://www.xmlrpc.com/>>.
43. Zend. Creando un Web Service con PHP | Programando por diversion. 2009. [cited 2 April 2010]. Available from world wide web: <<http://copstone.com/2010/03/creando-un-web-service-con-php/>>.

## GLOSARIO DE TÉRMINOS

**Artefactos:** Un artefacto es un producto tangible resultante del proceso de desarrollo de software. Algunos artefactos como los casos de uso, diagrama de clases u otros modelos UML ayudan a la descripción de la función, la arquitectura o el diseño del software.

**ASCII:** El código ASCII (acrónimo inglés de American Standard Code for Information Interchange) pronunciado generalmente [áski], es un código de caracteres basado en el alfabeto latino tal como se usa en inglés moderno y en otras lenguas occidentales.

**Blog:** Un blog, o en español también una bitácora, es un sitio web periódicamente actualizado que recopila cronológicamente textos o artículos de uno o varios autores, apareciendo primero el más reciente, donde el autor conserva siempre la libertad de dejar publicado lo que crea pertinente.

**GNU/Linux:** es el término empleado para referirse al sistema operativo similar a Unix que utiliza como base las herramientas de sistema de GNU y el núcleo Linux.

**GPL:** La Licencia Pública General de GNU o más conocida por su nombre en inglés GNU General Public License o simplemente su acrónimo del inglés GNU GPL, es una licencia creada por la Free Software Foundation a mediados de los 80, y está orientada principalmente a proteger la libre distribución, modificación y uso de software. Su propósito es declarar que el software cubierto por esta licencia es software libre y protegerlo de intentos de apropiación que restrinjan esas libertades a los usuarios.

**Interoperabilidad:** Es la capacidad de los sistemas de intercambiar procesos y datos, comúnmente a través de servicios.

**Librería:** En ciencias de la computación, una biblioteca (o librería) es un conjunto de subprogramas utilizados para desarrollar software. Las bibliotecas contienen código y datos, que proporcionan servicios a programas independientes, es decir, pasan a formar parte de éstos.

**Licencia de software:** Una licencia de software es un contrato entre el titular del derecho de autor (propietario) y el usuario del programa informático (usuario final), para utilizar éste en una forma determinada y de conformidad con unas condiciones convenidas.

**Paquete:** Un paquete de software es una serie de programas que se distribuyen conjuntamente. Algunas de las razones suelen ser que el funcionamiento de cada uno complementa a o requiere de otros, además de que sus objetivos están relacionados como estrategia de mercadotecnia.

**Release:** Se refiere a una versión funcional de un producto software.

**SCRUM:** Define un marco de trabajo ágil para la gestión de proyectos, basado en iteraciones y entregas incrementales de desarrollo de un producto o servicio. Especialmente indicado para proyectos con requisitos cambiantes.

**Servicio Web:** Es un sistema software diseñado para soportar la interacción máquina-máquina a través de una red, utilizando protocolos estándares de Internet y XML como formato de mensajes.

**Servidor Web:** Software que suministra páginas Web en respuesta a las peticiones de los navegadores Web.

**SOAP:** Protocolo de comunicación entre aplicaciones, fácil de consumir y es robusto.

**TIC:** Tecnologías de la información y las Comunicaciones.

**UML(Unified Modeling Language):**Lenguaje Unificado de Modelado, es un lenguaje de modelado de sistemas de software. Permite la especificación, visualización, construcción y documentación de elementos de la Ingeniería del Software.

**WEB 2.0:** El término, Web 2.0 fue acuñado por Tim O'Reilly en 2004 para referirse a una segunda generación en la historia de la Web basada en comunidades de usuarios y una gama especial de servicios, como las redes sociales, los blogs y los wikis.

**Web Semántica:** Es la descripción de los datos a través de metadatos (datos que

describen y ayudan a ubicar otros datos) llevado al marco de la web. La información queda reflejada a través de otros datos que cobran sentido y significado, de manera que las máquinas pueden procesarla.

**W3C (World Wide Web Consortium):** Organismo internacional que tiene por fin establecer normas para el desarrollo y uso de la web.

**Wiki :** Un wiki, o una wiki, es un sitio web cuyas páginas web pueden ser editadas por múltiples voluntarios a través del navegador web.

**WWW :** World Wide Web o Red Global Mundial es un sistema de documentos de hipertexto y/o hipermedios enlazados y accesibles a través de Internet.

**XP(eXtremme Programing):** Es una metodología ágil centrada en potenciar las relaciones interpersonales como clave para el éxito en desarrollo de software, promoviendo el trabajo en equipo, preocupándose por el aprendizaje de los desarrolladores, y propiciando un buen clima de trabajo.

## ANEXOS

## Anexo 1. WSDL del servicio project\_service.

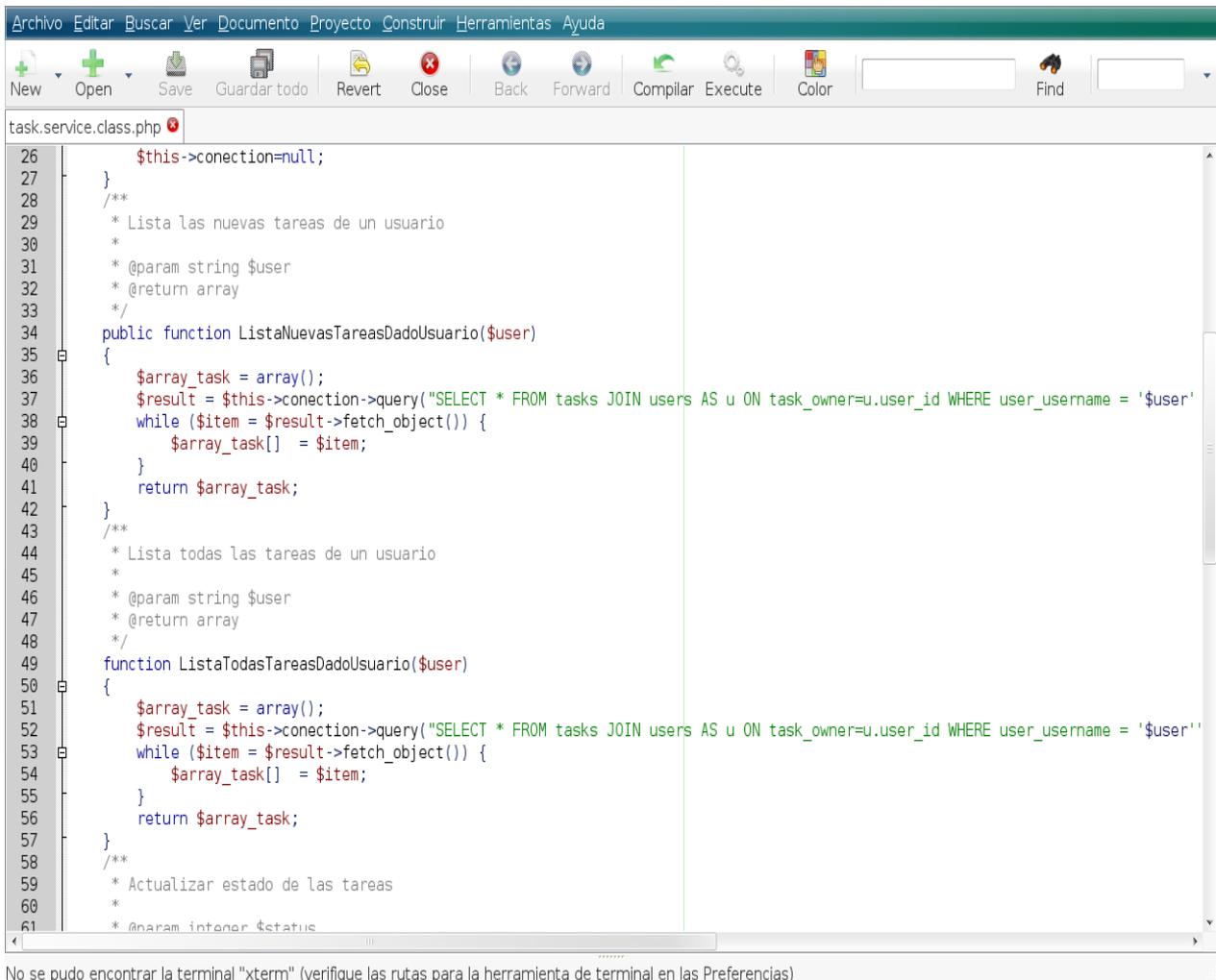
This XML file does not appear to have any style information associated with it. The document tree is shown below.

```

<!-- WSDL file generated by Zend Studio. -->
- <definitions name="project_service" targetNamespace="urn:project_service">
  <message name="_construct"/>
  <message name="_constructResponse"/>
  <message name="_destruct"/>
  <message name="_destructResponse"/>
  <message name="ActualizarEstadoTareaporUsuario">
    <part name="status" type="xsd:integer"/>
    <part name="user" type="xsd:string"/>
    <part name="task_id" type="xsd:integer"/>
  </message>
  <message name="ActualizarEstadoTareaporUsuarioResponse">
    <part name="ActualizarEstadoTareaporUsuarioReturn" type="xsd:boolean"/>
  </message>
  <message name="ListaNuevasTareasDadoUsuario">
    <part name="user" type="xsd:string"/>
  </message>
  <message name="ListaNuevasTareasDadoUsuarioResponse">
    <part name="ListaNuevasTareasDadoUsuarioReturn" type="xsd:anyType"/>
  </message>
  <message name="ListaTodasTareasDadoUsuario">
    <part name="user" type="xsd:string"/>
  </message>
  <message name="ListaTodasTareasDadoUsuarioResponse">
    <part name="ListaTodasTareasDadoUsuarioReturn" type="xsd:anyType"/>
  </message>
  <portType name="project_servicePortType">
    <operation name="_construct">
      <documentation> Constructor de la Clase Tasks </documentation>
      <input message="typens:_construct"/>
      <output message="typens:_constructResponse"/>
    </operation>
    <operation name="_destruct">
      <documentation> Destructor de la Clase Tasks </documentation>
      <input message="typens:_destruct"/>
    </operation>
  </portType>
</definitions>

```

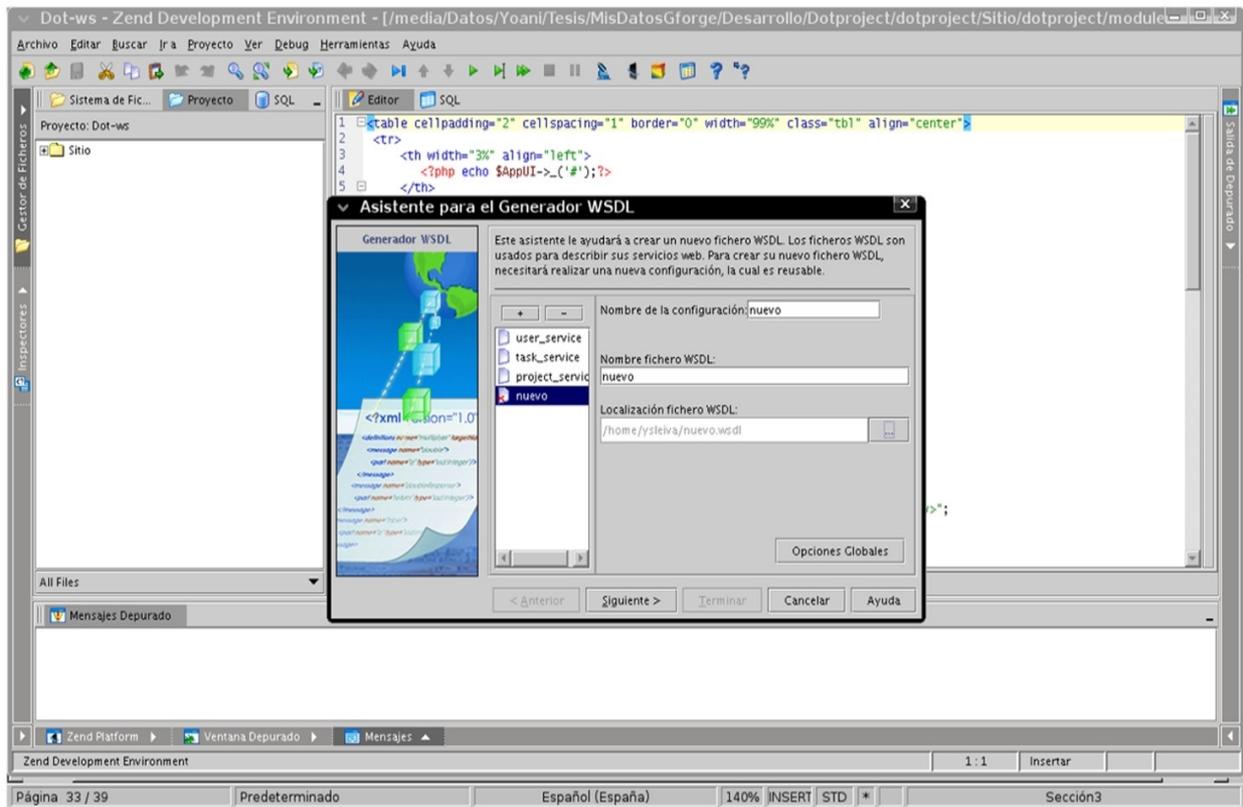
## Anexo 2. Comentarios de una función.



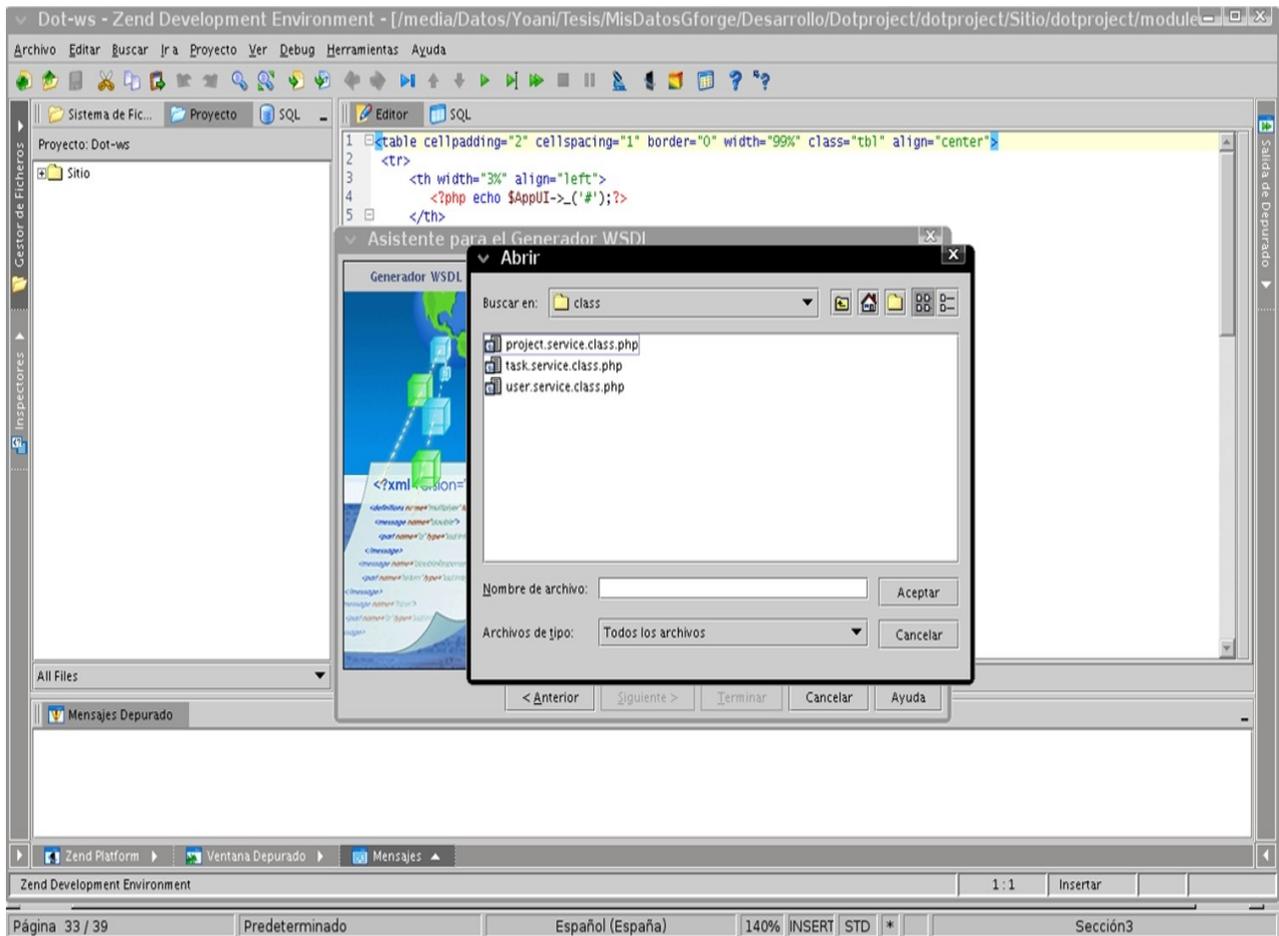
```
task.service.class.php
26     $this->conexion=null;
27 }
28 /**
29  * Lista las nuevas tareas de un usuario
30  *
31  * @param string $user
32  * @return array
33  */
34 public function ListaNuevasTareasDadoUsuario($user)
35 {
36     $array_task = array();
37     $result = $this->conexion->query("SELECT * FROM tasks JOIN users AS u ON task_owner=u.user_id WHERE user_username = '$user'");
38     while ($item = $result->fetch_object()) {
39         $array_task[] = $item;
40     }
41     return $array_task;
42 }
43 /**
44  * Lista todas las tareas de un usuario
45  *
46  * @param string $user
47  * @return array
48  */
49 function ListaTodasTareasDadoUsuario($user)
50 {
51     $array_task = array();
52     $result = $this->conexion->query("SELECT * FROM tasks JOIN users AS u ON task_owner=u.user_id WHERE user_username = '$user'");
53     while ($item = $result->fetch_object()) {
54         $array_task[] = $item;
55     }
56     return $array_task;
57 }
58 /**
59  * Actualizar estado de las tareas
60  *
61  * @param integer $status
```

No se pudo encontrar la terminal "xterm" (verifique las rutas para la herramienta de terminal en las Preferencias)

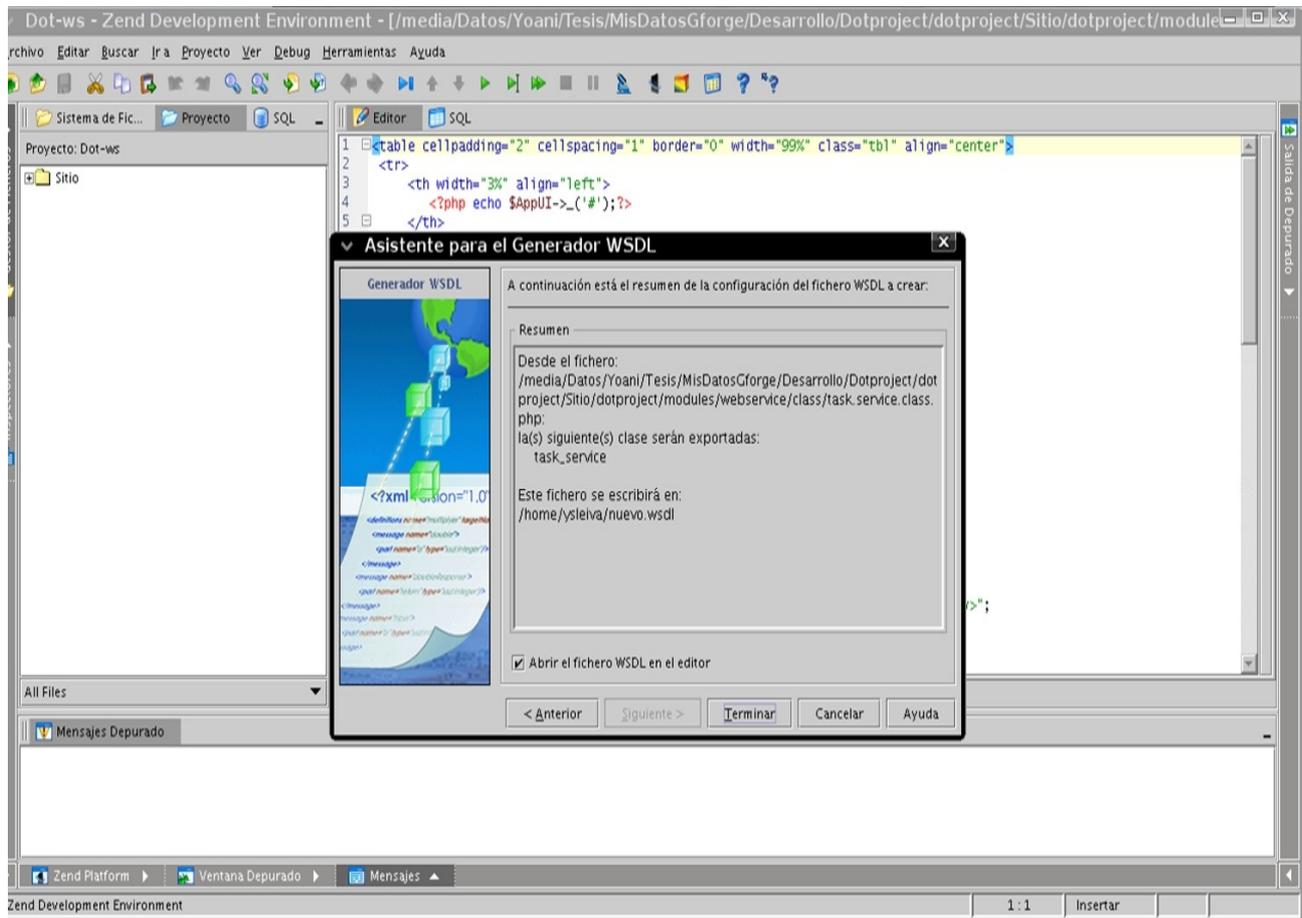
### Anexo 3 Configurar la descripción del WSDL



## Anexo 4 Agregar el fichero de la clase.



## Anexo 5 Resumen de la generación del fichero WSDL.



**Anexo 6** Pasos para la instalación de módulos al dotProject.

El SGP dotProject dispone de diferentes módulos que aumentan sus funcionalidades básicas.

1. Existe un grupo de módulos para dotProject que se pueden descargar desde la dirección <http://sourceforge.net/projects/dotmods> en caso de que no sea un módulo creado por nosotros mismos.
2. Copiar el directorio del módulo en el directorio *modules* del dotproject.  
Ej. Directorio: dotproject/modules/módulo\_de\_prueba
3. Activar el módulo.

Para activar los módulos dotProject brinda una interfaz bastante amigable con el usuario. La ruta a seguir para activar los módulos es la siguiente:

- Sistema
- Ver módulos
- Seleccionar módulo a instalar
- Activar el módulo.
- Darle visibilidad módulo (para que aparezca en el menú general).

## Anexo 7 Código fuente del instalador de DOT-WS

```

<?php
/**
 * Name: Web Service
 * Directory: webservice
 * Version 0.1
 * Type: user
 * UI Name: Web Service
 * UI Icon: ?
 */
//require_once(DP_BASE_DIR.'/modules/report/includes/functions/file.php');
$config = array();
$config['mod_name'] = 'Web Service'; // name the module
$config['mod_version'] = '0.1'; // add a version number
$config['mod_directory'] = 'webservice'; // tell dotProject where to find this module
$config['mod_setup_class'] = 'DP_WebService'; // the name of the PHP setup class (used
below)
$config['mod_type'] = 'user'; // 'core' for modules distributed with dP by standard, 'user' for
additional modules from dotmods
$config['mod_ui_name'] = 'Web Service'; // the name that is shown in the main menu of the
User Interface
$config['mod_ui_icon'] = 'charts.png'; // name of a related icon
$config['mod_description'] = 'Servicios Web'; // some description of the module
$config['mod_config'] = false; // show 'configure' link in viewmods

if (@$a == 'setup') {
    echo dPshowModuleConfig( $config );
}
class DP_WebService{
function install(){
    $sql = '(
        id VARCHAR(45) NOT NULL,
        name DATE NOT NULL,
        description VARCHAR(45) NOT NULL,
        PRIMARY KEY(id,name)
    )TYPE=MyISAM';

    $q = new DBQuery;
    $q->createTable('webservice');
    $q->createDefinition($sql);
    $q->exec();
    $q->clear();
    return db_error();
} //end install()

function remove() {
    $q = new DBQuery;
    $q->dropTable('webservice');
    $q->exec();
    $q->clear();
    return db_error();
} //end remove()
}
?>

```

## Anexo 8 Prototipo de interfaz de la historia de usuario crear vista de tarea.

Sistema de planificación del Proyecto Unicornios dotProject.net  
FREE SOFTWARE

Empresas | Proyectos | Tareas | Calendario | Contactos | Ficheros | Foros | Tickets | Departamentos | Usuarios | Sistema | Reportes | Risks | Servicios Web - New Item

Bienvenido Yoani Santana Leiva Ayuda | Mis datos | **A realizar** | Hoy | Salir

**Servicios Web**

con pestañas : plano

Vista de tarea | Vista de proyecto | Vista de usuario

**Servicio: task\_service**

**WSDL del servicio:** [http://localhost/dot-ws/dotproject/modules/webservice/wsd/tasks\\_service.wsdl](http://localhost/dot-ws/dotproject/modules/webservice/wsd/tasks_service.wsdl) Ver WSDL

**Namespace:** urn:task\_service

**URL:** [http://localhost/dot-ws/dotproject/modules/webservice/controller/task/task\\_controller.php](http://localhost/dot-ws/dotproject/modules/webservice/controller/task/task_controller.php)

**Protocolo:** SOAP

**Estilo por defecto:** rpc

**Transporte:** SOAP sobre HTTP

**Lista de métodos**

#	Nombre	Descripción	Entrada	Salida
1	Autenticar	Autenticar SQL	username: <b>string</b> , password: <b>string</b> ,	boolean
2	ListaNuevasTareasDadoUsuario	Lista las nuevas tareas de un usuario	user: <b>string</b> ,	array
3	ListaTodasTareasDadoUsuario	Lista todas las tareas de un usuario	user: <b>string</b> ,	array
4	ActualizarEstadoTarea	Actualizar estado de una tarea	status: <b>integer</b> , task_id: <b>integer</b> ,	boolean
5	EliminarTareaDadoId	Eliminar una tarea	task_id: <b>integer</b> ,	boolean
6	CrearTareas	Crear una nueva tarea	task_name: <b>string</b> , task_project: <b>integer</b> , task_owner: <b>integer</b> , task_start_date: <b>string</b> , task_duration: <b>integer</b> , task_duration_type: <b>integer</b> , task_hours_worked: <b>integer</b> , task_end_date: <b>string</b> , task_status:	boolean

## Anexo 9 Prototipo de interfaz de la historia de usuario crear vista de proyecto.

Sistema de planificación del Proyecto Unicornios dotProject.net  
FREE SOFTWARE

Empresas | Proyectos | Tareas | Calendario | Contactos | Ficheros | Foros | Tickets | Departamentos | Usuarios | Sistema | Reportes | Risks | Servicios Web - New Item -

Bienvenido Yoani Santana Leiva Ayuda | Mis datos | **A realizar** | Hoy | Salir

### Servicios Web

con pestañas : plano

Vista de tarea | **Vista de proyecto** | Vista de usuario

**Servicio: project\_service**

**WSDL del servicio:** [http://localhost/dot-ws/dotproject/modules/webservice/wsd/project\\_service.wsdl](http://localhost/dot-ws/dotproject/modules/webservice/wsd/project_service.wsdl) Ver WSDL

**Namespace:** urn:project\_service

**URL:** [http://localhost/dot-ws/dotproject/modules/webservice/controller/project/project\\_controller.php](http://localhost/dot-ws/dotproject/modules/webservice/controller/project/project_controller.php)

**Protocolo:** SOAP

**Estilo por defecto:** rpc

**Transporte:** SOAP sobre HTTP

**Lista de métodos**

#	Nombre del método	Descripción	Entrada	Salida
1	ListaProyectos	Listado de proyectos		array
2	ObtenerTareasdeProyecto	Obtener tareas de un proyecto.	project_id: <b>integer</b> ,	array
3	CrearNuevoProyecto	Crear un proyecto	project_company: <b>integer</b> , project_department: <b>integer</b> , project_name: <b>string</b> , project_short_name: <b>string</b> , project_owner: <b>integer</b> , project_url: <b>string</b> , project_demo_url: <b>string</b> , project_start_date: <b>string</b> , project_end_date: <b>string</b> , project_actual_end_date: <b>string</b> , project_status: <b>integer</b> , project_percent_complete: <b>integer</b> ,	boolean

## Anexo 10 Prototipo de interfaz de la historia de usuario crear vista de usuario.

Sistema de planificación del Proyecto Unicornios dotProject.net  
FREE SOFTWARE

Empresas | Proyectos | Tareas | Calendario | Contactos | Ficheros | Foros | Tickets | Departamentos | Usuarios | Sistema | Reportes | Risks | Servicios Web - New Item +

Bienvenido Yoani Santana Leiva Ayuda | Mis datos | **A realizar** | Hoy | Salir

### Servicios Web

con pestañas : plano

Vista de tarea | **Vista de proyecto** | Vista de usuario

 **Servicio: user\_service**

**WSDL del servicio:** [http://localhost/dot-ws/dotproject/modules/webservice/wsd/user\\_service.wsdl](http://localhost/dot-ws/dotproject/modules/webservice/wsd/user_service.wsdl) Ver WSDL

**Namespace:** urn:user\_service

**URL:** [http://localhost/dot-ws/dotproject/modules/webservice/controller/user/user\\_controller.php](http://localhost/dot-ws/dotproject/modules/webservice/controller/user/user_controller.php)

**Protocolo:** SOAP

**Estilo por defecto:** rpc

**Transporte:** SOAP sobre HTTP

 **Lista de métodos**

#	Nombre	Descripción	Entrada	Salida
1	ListadeUsuarios	Lista de usuarios		array
2	ObtenerDatosdeUsuarioDadoId	Obtener datos de un usuario	user_id: <b>integer</b> ,	array
3	ActualizarDatosdeUsuario	Actualizar datos de usuario	user_id: <b>integer</b> , contact_first_name: <b>string</b> , contact_phone: <b>integer</b> , contact_group: <b>integer</b> , contact_sex: <b>string</b> , contact_apartment: <b>integer</b> , contact_d: <b>string</b> , contact_city: <b>string</b> , contact_lastname: <b>string</b> ,	boolean
4	AdicionarDatosUsuario	Adiciona los datos de un usuario	contact_first_name: <b>string</b> , contact_phone: <b>integer</b> , contact_group: <b>integer</b> , contact_sex: <b>string</b> , contact_apartment: <b>integer</b> ,	boolean

## Anexo 11 Estándar de codificación utilizado.

El código debe ser legible y estar escrito una sola vez, pero revisado y modificado varias veces. Lo cual permite que el código sea comprensible, y por tanto, que pueda ser mantenible. Aunque el rendimiento es importante, si hay una elección entre el rendimiento y la facilidad de lectura, la lectura debe tener preferencia.

### Sangrado y longitud de la línea

La forma más elegante para marcar indentación (al principio del renglón) es la utilización de Tabs. Solamente se puede utilizar Tabs para la indentación o sangrado. Cada nivel de indentación puede ser solamente un Tab.

### Modo de configuración de las líneas o las entradas de otros editores.

Dado que los editores modernos ya no se limitan a 80 caracteres de ancho de línea, las longitudes de línea son menos importantes que la lectura. Ya las líneas de 75 caracteres debe ser consideradas obsoletas y las líneas de más de 100 caracteres de longitud tienen que romperse.

### Estructuras de control

Estas incluyen **if**, **for**, **while**, **switch**, etc. A continuación se muestra un ejemplo de declaración, ya que es el más complicado de ellos:

```
<?php
  if ((condicion1) || (condicion2)){
    accion1;
  } elseif ((condicion3) && (condicion4)){
    accion2;
  } else {
    accion_por_defecto;
  }
?>
```

Las declaraciones de control deben tener un espacio entre las palabras clave de control y la apertura de paréntesis, para distinguirlos de las llamadas a funciones.

Es recomendable utilizar siempre llaves, incluso en situaciones en las que son técnicamente opcional ya que estas aumentan la legibilidad y disminuye la probabilidad de introducir errores lógicos cuando se añaden nuevas líneas.

Para declaraciones utilizando switch:

```
<?php
switch (condicion) {
    case 1:
        accion1;
        break;
    case 2:
        accion2;
        break;
    default:
        accion_por_defecto;
        break;
}
?>
```

## Funciones

Las funciones deben ser, sin espacios entre el nombre de la función, la apertura de paréntesis, y el primer parámetro, comas y espacios entre cada parámetro, y ningún espacio entre el último parámetro, el paréntesis de cierre, y el punto y coma. He aquí un ejemplo:

```
<?php
$var = foo($bar, $variable, $pvar);
?>
```

Como se muestra en el ejemplo anterior, debe haber un espacio a ambos lados de un signo de igualdad utilizado para asignar el valor de retorno de una función a una variable. En el caso de un bloque de asignaciones, se puede introducir más espacio para facilitar la lectura e interpretación del código, a continuación veremos un ejemplo:

```
<?php
$short = foo($bar);
$long_variable = foo($variable);
?>
```

## Definición de funciones

La declaración de funciones sigue el “estilo K & R”, el cual permite ahorrar espacio vertical de lectura. Ejemplo:

```
<?php
function fooFunction($arg1,$arg2){
    if (condicion) {
        definiciones;
    }
}
```

```

return $val;
}
?>

```

Los argumentos de las funciones que tengan valores por defecto se colocan al final de la lista de argumentos que posee la función. Ejemplo:

```

<?php
function connect(&$dsn, $persistent = false) {
    if (is_array($dsn)) {
        $dsninfo = &$dsn;
    } else {
        $dsninfo = DB::parseDSN($dsn);
    }

    if (!$dsninfo || !$dsninfo['phptype']) {
        return $this->raiseError();
    }
    return true;
}
?>

```

## Comentarios

Completar la documentación en línea con bloques de comentarios proporcionados. Estos bloques de comentarios deben seguir el estilo C (`/**` `*/`) ó el estándar de C++ (`//`).

Una regla general es que si nos fijamos en una sección de código es necesario comentar antes que se olvide cómo funciona.

## Etiquetas de Código PHP

Siempre se debe usar `<?php` para abrir y `?>` para cerrar, y de esta forma delimitar el código PHP, nunca utilice `<? >`.

Etiquetas necesarias que tienen contenido variable

- Descripciones breves

Todos los bloques de comentarios deben ser breves descripciones, preferentemente en una oración pequeña.

- Versiones de PHP

Uno de los siguientes bloques de comentarios debe ir en el nivel de página:

- PHP versión 4
- PHP versión 5
- PHP versión 4 y 5

➤ Autor

No existe ninguna norma para determinar cuándo un nuevo contribuyente de código debe añadirse a la lista de los autores de un determinado archivo fuente. En general, los cambios deben caer en la categoría sustancial, es decir, alrededor de 10% al 20% de los cambios en el código. Las excepciones pueden ser hechas para la reescritura de funciones o en su contribución.

➤ Derechos de autor.

Los desarrolladores de módulos son libres de aplicar los derechos de autor independientemente de que usted desee. El año debe estar en formato de cuatro dígitos y, si es un período de años se trata de usar un guión entre la primera y el año más reciente. El titular del derecho de autor puede ser usted, una lista de personas, una empresa, el Grupo de PHP, etc. Ejemplos:

- \* @copyright 2003 John Doe and Jennifer Buck
- \* @copyright 2001-2004 John Doe
- \* @copyright 1997-2004 The PHP Group
- \* @copyright 2001-2004 XYZ Corporation

Para el código básico se recomienda que se aplique la siguiente declaración sobre derecho de autor:

- \* @copyright 2001-2007 The dotProject Development Team.

➤ Ver

Añadir una etiqueta **@see** cuando quiera referirse a los usuarios o a otras secciones de la documentación del paquete.

## Nombres convencionales

➤ Clases

Las clases deben tener nombres descriptivos. Evite el uso de abreviaturas cuando sea posible. Los nombres de clase siempre debe comenzar con una letra mayúscula. Las clases fundamentales son con el prefijo C. El módulo principal de nombres de clase deben comenzar con el nombre del módulo. Ejemplos de buenos nombres de clase son los siguientes:

- CAppUI
- MyModuleClass

- Métodos y funciones

Las funciones tienen además del nombre del paquete un prefijo para evitar las colisiones entre los nombres de paquetes. La letra inicial del nombre (tras el prefijo) es minúscula, y cada letra que comienza una nueva "palabra" es mayúscula. Algunos ejemplos son:

- Connect()
- GetData()
- BuildSomeWidget()
- XML\_RPC\_serializeData()

Los miembros privados de la clase (es decir, los miembros de la clase que van a utilizarse sólo dentro de la misma clase en que se declaró; realmente PHP no soporta todavía ejecutar variables privadas) están precedidos por un solo subrayado. Por ejemplo:

- `_sort()`
- `_initTree()`
- `$this->_status`

En PHP5 los miembros protegidos de la clase (es decir, los miembros de la clase que van a utilizarse sólo dentro de la misma clase en la que se declaran o en subclases que se extienden desde la misma) no están precedidos por ningún subrayado. Por ejemplo:

- `protected $somevar`
- `protected function initTree()`
  
- Constantes

Todas las constantes deben ser siempre en mayúsculas y con subrayados para separar palabras. Estas poseen un prefijo de nombres constantes con el nombre de la clase o módulo en el que se utiliza. Las únicas constantes que se exceptúan de la norma de todas las mayúsculas y siempre deben estar en minúsculas son las constantes **true**, **false** y **null**.

➤ Variables Globales

Si el módulo tiene que definir las variables globales, debe empezar su nombre en mayúsculas y con el nombre de la clase o del módulo.

**Anexo 12** Ejemplos de Códigos de Pruebas utilizados para las Pruebas de Aceptación.

### Código del Caso de Prueba HU-DOT-WS-02-01

```
<?php
//Host donde esta público el wsdl
$wsdl= "http://localhost/dot-ws/dotproject/modules/webservice/wsdl/task_service.wsdl";
//Crear cliente soap
$soap = new SoapClient($wsdl,array("connection_timeout"=>50));
//Llamar a la función en el servicio
$resonse = $soap->ListaNuevasTareasDadoUsuario('yvillazon');
echo "<pre>";
//Imprimir respuesta
print_r($resonse);
echo "</pre>";
?>
```

### Código del Caso de Prueba HU-DOT-WS-03-01

```
<?php
//Host donde esta público el wsdl
$wsdl= "http://localhost/dot-ws/dotproject/modules/webservice/wsdl/task_service.wsdl";
//Crear cliente soap
$soap = new SoapClient($wsdl,array("connection_timeout"=>50));
//Llamar a la función en el servicio
$resonse = $soap->ListaTodasTareasDadoUsuario('yvillazon');
echo "<pre>";
//Imprimir respuesta
print_r($resonse);
echo "</pre>";
?>
```

### Código del Caso de Prueba HU-DOT-WS-11-01

```
<?php
//Host donde esta público el wsdl
$wsdl= "http://localhost/dot-ws/dotproject/modules/webservice/wsdl/task_service.wsdl";
//Crear cliente soap
$soap = new SoapClient($wsdl,array("connection_timeout"=>50));
//Llamar a la función en el servicio se le pasa el estado Atrasado = 3
$resonse = $soap->ListaTodasTareasSegunEstado(3);
echo "<pre>";
//Imprimir respuesta
print_r($resonse);
echo "</pre>";
?>
```

### Código del Caso de Prueba HU-DOT-WS-15-01

```
<?php
//Host donde esta público el wsdl
$wsdl= "http://localhost/dot-ws/dotproject/modules/webservice/wsdl/projects_service.wsdl";
//Crear cliente soap
$soap = new SoapClient($wsdl,array("connection_timeout"=>50));
```

```
//Llamar a la función en el servicio
$resonse = $soap->ListaProyectos();
echo "<pre>";
//Imprimir respuesta
print_r($resonse);
echo "</pre>";
?>
```

### Código del Caso de Prueba HU-DOT-WS-16-01

```
<?php
//Host donde esta público el wsdl
$wsdl= "http://localhost/dot-ws/dotproject/modules/webservice/wsdl/projects_service.wsdl";
//Crear cliente soap
$soap = new SoapClient($wsdl,array("connection_timeout"=>50));
//Llamar a la función en el servicio se le pasa el id del proyecto.
$resonse = $soap->ObtenerDatosProyectoDadold(10);
echo "<pre>";
//Imprimir respuesta
print_r($resonse);
echo "</pre>";
?>
```

### Código del Caso de Prueba HU-DOT-WS-22-01

```
<?php
//Host donde esta público el wsdl
$wsdl= "http://localhost/dot-ws/dotproject/modules/webservice/wsdl/user_service.wsdl";
//Crear cliente soap
$soap = new SoapClient($wsdl,array("connection_timeout"=>50));
//Llamar a la función en el servicio
$resonse = $soap->ListadeUsuarios();
echo "<pre>";
//Imprimir respuesta
print_r($resonse);
echo "</pre>";
?>
```

## Anexo 13 Interfaz del caso de prueba HU-DOT-WS-15-01

### Bienvenido al Sistema de Prueba de los Servicios Web

Usuario Conectado: ysleiva [Cerrar sesion](#)

Proyectos [Mis Tareas](#)

## Proyectos

nuevo proyecto : [editar proyecto](#)

Nombre	Fecha de Inicio	Fecha de Fin	Dueño
Observatorio de Tecnologías Libres	2007-03-30 00:00:00	2007-07-15 23:59:59	Yenner Joaquin Diaz Nuñez
UXI v2.0 (Serv)	2007-09-03 00:00:00	2008-07-18 23:59:59	Abel Garcia Vitier
Gtk-Octave (EXP)	2007-03-15 00:00:00	2007-07-20 23:59:59	Eiger Mora Moredo
UXI (Revista de SWL de la UCI)	2007-01-01 00:00:00	2007-06-30 23:59:59	Abel Garcia Vitier
Concurso CHC62	2007-03-01 00:00:00	2007-06-15 23:59:59	
Portal de Ajedrez	2007-02-01 00:00:00	2007-09-01 23:59:59	Raciel Garrido Torres
Servicio de Repositorio Debian	2006-09-01 00:00:00	2007-07-15 23:59:59	
Serv. DOC y UPLOAD Cliente (EXP)	2007-02-27 00:00:00	2007-07-15 23:59:59	Jose Ernesto Torres Sanchez
Software Libre para Ciegos	2007-02-01 00:00:00	2007-07-15 23:59:59	
Sistema de Clonacion de Imagenes	2006-09-01 00:00:00	2008-06-19 23:59:59	Dayron Perez Roldan
ServiceDesk UCI	2006-12-06 00:00:00	2007-07-15 23:59:59	Yusdel Juan Concepcion Peña
Gestión del Proyecto UNICORNIOS	2007-04-13 00:00:00	2007-07-15 23:59:59	Abel Meneses Abad
Portal de SWL 2	2006-10-01 00:00:00	2007-07-15 23:59:59	Keiver Hernandez Fernandez
NDE(Tocororo Desktop)	2007-04-24 00:00:00	2009-06-30 23:59:59	Eiger Mora Moredo
Grupo de Invest. de Migracion	2007-04-24 00:00:00	2007-07-15 23:59:59	
Multimedia de GNU/Linux Básico	2007-04-24 00:00:00	2007-05-24 23:59:59	
Imagenes para la Docencia (Serv)	2007-04-24 00:00:00	2007-07-15 23:59:59	Jorge Mijail Vazquez Paredes
Dotproject	2007-05-07 00:00:00	2007-05-31 23:59:59	
FreeStudio	2007-09-03 00:00:00	2008-01-28 23:59:59	Alexander Martinez Fajardo
Gestion del Proyecto 2007 - 08	2007-08-27 00:00:00	2008-07-19 23:59:59	Abel Meneses Abad
Práctica Profesional	2007-06-04 00:00:00	2007-07-14 23:59:59	Abel Meneses Abad
Migracion de ETECSA	2007-12-14 00:00:00	2010-04-30 23:59:59	Keiver Hernandez Fernandez
FreeVIUX (serv)	2008-02-29 00:00:00	2009-02-27 23:59:59	David Padron Alvarez
Miración Centro Nacional Música Popular	2007-09-03 00:00:00	2007-12-03 23:59:59	Ivisan Condero Sadradin