

Universidad de las Ciencias Informáticas

Facultad 5



**Implementación de una Arquitectura Orientada a
Servicios en los Procesos de Gestión de Proyectos de
ALBET, Ingeniería y Sistemas.**

**TRABAJO DE DIPLOMA PARA OPTAR POR EL TÍTULO DE
INGENIERO EN CIENCIAS INFORMÁTICAS**

Autores: Yusnier Blanco Ravelo.

Idelvis Escobar López.

Tutores: Ing. Manuel Alejandro Gil Martín.

Ing. Susej Beovides Luís.

Ciudad de La Habana, Junio 2009

“Año del 50 Aniversario del Triunfo de la Revolución”

AGRADECIMIENTOS

A la Revolución y a nuestro Comandante Fidel Castro, por habernos dado la oportunidad de estudiar en esta universidad.

A nuestros tutores Manuel y Susej por todo el apoyo, la confianza depositada y por contribuir a nuestra formación como profesionales.

A todos nuestros compañeros y amigos que siempre estuvieron presente en las buenas y malas, especialmente: Argelio, Israel, Prado, Leandro, Yuliet.

A nuestros padres de forma muy especial por guiarnos por el camino correcto y ayudarnos a alcanzar este sueño tan anhelado.

A todos aquellos que de una forma u otra nos ayudaron.

Muchas Gracias.

DEDICATORIA

Yusnier Blanco Ravelo:

De forma muy especial a las personas que más quiero en mi vida: a mis padres Martha y Emiliano, mi hermana Yoili, mi novia Kenia, mi sobrina Milie y mi abuela Julia por ser todo para mí en esta vida, por entregarme de forma incondicional su amor y dedicación, además de apoyarme y guiarme de forma correcta en la vida. A mis amigos Argelio, Israel e Idelvis por el apoyo brindado en el decursar por esta universidad. A mis tías Betty y Elena, mis tíos Enrique y Alejandro por ese calor familiar que solo ellos saben brindar.

Idelvis Escobar López:

Dedicárselo de forma especial a mi familia que sin su apoyo no hubiera sido posible este largo camino, en especial a mis abuelos María y Caridad a mi mamá Marisel, a mis tíos Cuqui y Felix en fin a todos. A mis amigos que siempre recordare Israel, Argelio, Yunier a Elizabeth y Yulie que regañan mas que mis padres, a todos ellos gracias de corazón por todo el apoyo que me dieron. Por ultimo a la gente de la comunidad, gracias y que siga así de unida.

RESUMEN

Albet, Ingeniería y Sistema es una empresa creada en el año 2005 como respuesta a los proyectos que se gestaron fundamentalmente en el marco del Convenio Integral de Cooperación con Venezuela. Desde septiembre del 2007 se ha venido desarrollando el reordenamiento de Albet para que funcionara realmente como una empresa, aunque es válido destacar que existen problemas en muchos de sus procesos de negocio como la Gestión de Proyectos.

Esta investigación está encaminada a contribuir al mejoramiento a través de una implementación de una Arquitectura Orientada a Servicios (SOA) de los Procesos de Gestión de Proyectos de Albet con el objetivo de elevar su efectividad en el seguimiento y control de los proyectos, la toma de decisiones y la productividad de los empleados.

Para satisfacer el objetivo de esta investigación se realizó un estudio de los conceptos básicos asociados al dominio del problema, se efectuó un modelado de negocio basado en procesos y un diseño basado en el framework de trabajo desarrollado por la universidad para proyectos implementados a partir de una SOA.

Todo el desarrollo de este trabajo está basado en tecnologías libres y multiplataforma, utilizando PHP 5 como lenguaje de programación, UML y BPMN como lenguajes de modelado, como Sistema Gestor de Base de Datos PostgreSQL 8.3 y SQL Manager 2005 for PostgreSQL para su administración, Apache 2.2 como servidor web, Visual Paradigm como herramienta UML CASE y SoapUI para realizarle las pruebas a los servicios.

Palabras claves: Arquitectura, Servicios, Procesos, Gestión.

TABLA DE CONTENIDOS

TABLA DE CONTENIDO

INTRODUCCIÓN	1
CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA	4
1.1 Introducción.....	4
1.2 Las tecnologías de la información y las comunicaciones.....	4
1.3 Arquitectura Orientada a Servicios (SOA).....	6
1.3.1 Ventajas de SOA.....	7
1.3.2 Segunda generación de SOA.....	8
1.3.3 Principales propiedades de SOA.....	9
1.3.4 Principios de la orientación a servicios.....	9
1.3.5 SOA define diferentes capas de software.....	10
1.4 Servicios Web.....	10
1.4.1 Antecedentes históricos de los Servicios WEB.....	12
1.4.2 Ventajas de los Servicios Web.....	14
1.4.3 Desventajas de los Servicios Web.....	14
1.5 Tecnologías para desarrollar una Arquitectura Orientada a Servicios en los procesos de gestión de proyectos en Albet.....	15
1.5.1 Herramientas de soporte a los procesos de negocios.....	16
1.5.2 Herramientas de soporte a Servicios Web.....	16
1.5.2.1 Servidores web.....	17
1.5.2.2 IDEs de Desarrollo.....	18
1.5.2.3 Sistemas de Gestión de Base Datos (SGBD).....	20
1.5.2.4 Lenguajes de programación.....	23
1.5.2.5 Herramienta de prueba.....	26
1.5.3 Lenguajes de modelado.....	27
1.5.4 Visual Paradigm.....	29

TABLA DE CONTENIDOS

1.6 Tecnologías a utilizar en la implementación de una SOA en los procesos de gestión de proyecto de ALBET.....	30
1.7 Conclusiones.....	30
CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO.....	32
2.1 Introducción.....	32
2.2 Modelo del negocio basado en procesos.....	32
2.3 Descripción de la empresa.....	33
2.4 Procesos de negocios.....	36
2.4.1 Procesos de aprobación de la solicitud de negocio.....	37
2.4.2 Procesos de elaboración de la ficha técnica.....	40
2.4.3 Proceso de creación de la oferta.....	42
2.4.4 Procesos de contratación.....	44
2.4.5 Procesos de registro de proyecto.....	46
2.4.6 Proceso de gestión de suministro con proveedores.....	48
2.5 Diagramas de procesos.....	50
2.6 Conclusiones.....	56
CAPÍTULO 3: DISEÑO DE LOS SERVICIOS.....	57
3.1 Introducción.....	57
3.2 Identificación de los dominios de información.....	57
3.2.1 Ratio de cohesión.....	59
3.2.2 Ratio de acoplamiento.....	60
3.3 Identificación de los servicios.....	60
3.3.1 Matriz CRU.....	60
3.3.2 Descripción de los servicios de negocios.....	61
3.3.3 Matriz procesos contra servicios.....	90
3.4 Conclusiones.....	91
CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.....	92

TABLA DE CONTENIDOS

4.1 Introducción.....	92
4.2 Modelo de implementación.	92
4.2.1 Diagramas de componentes.....	92
4.2.2 Diagrama de despliegue.....	96
4.3 Pruebas.	97
4.4 Conclusiones.....	140
CONCLUSIONES	141
RECOMENDACIONES	142
REFERENCIAS BIBLIOGRÁFICAS	143
BIBLIOGRAFÍA	146
GLOSARIO DE TÉRMINOS	149

INTRODUCCIÓN.

La UCI es una universidad productiva cuya misión es la producción de software y servicios informáticos a partir de la vinculación estudio trabajo como modelo de formación. La producción de software y servicios informáticos se basa en la integración de los procesos de formación, investigación y producción en torno a una temática para convertirla en una rama productiva. Este espacio de integración temática es denominado Polo Productivo y se promueve la formación de pregrado, postgrado, la colaboración nacional e internacional, el fomento de líneas de investigación, desarrollo y la ejecución de proyecto en el marco de acuerdos de trabajo. La idea es convertir la informática en una de las ramas más productivas y aportadoras de recursos para la nación, es el empleo a fondo de la inteligencia y del capital humano que tenemos y principalmente del que se puede crear casi como espina dorsal de la economía. (1)

Es el resultado legítimo de la cooperación entre varias organizaciones del país, por lo que aquí confluyen destacados académicos, profesores, científicos, especialistas de diversos sectores, obreros y directivos de los Ministerios de Educación, las Universidades, el sector empresarial y las organizaciones juveniles. (2)

Por las características antes expuestas y como respuesta a los proyectos que se gestaron fundamentalmente en el marco del Convenio Integral de Cooperación con Venezuela, a finales del año 2005 fue creada ALBET. Esta empresa comercializadora de todos los productos y servicios informáticos que se desarrollan en La Infraestructura Productiva (IP), es la “cara comercial” para todos los productos de exportación, coordinados a través de la IP y en el marco del Programa Nacional para la Producción de Software.

Desde septiembre del 2007 a diciembre del 2008 se desarrolló el Reordenamiento de ALBET, para que funcionara realmente como una empresa, para lo cual se modificó el Objeto Social, los Estatutos, la Estructura, la Plantilla de Cargos y Ocupaciones, además se realizó el traspaso de patrocinio, se actualizó la documentación legal y se inscribieron los registros mercantiles. Todo esto responde como se expresaba anteriormente a un perfeccionamiento para una mejor gestión, aunque es válido destacar que este proceso continúa, pues aún existen dificultades en algunos procesos de negocios como la Gestión de Proyectos.

En este sentido el **problema a resolver** es, ALBET, Ingeniería y Sistemas como empresa comercializadora de Soluciones Tecnológicas Integrales presenta problemas con la Gestión de Proyecto

induciendo poca eficiencia en el seguimiento de los proyectos, la toma de decisiones y la competitividad de la empresa.

Objeto de estudio: Procesos de negocios de ALBET, Ingeniería y Sistemas.

Campo de acción: Procesos de gestión de proyectos de ALBET, Ingeniería y Sistemas.

Objetivo General: Implementar una Arquitectura Orientada a Servicios que contribuya a la mejora de los procesos de gestión de proyecto de ALBET, Ingeniería y Sistemas.

Tareas a desarrollar:

- Fundamentar los elementos teóricos referidos al tema de investigación.
- Describir ALBET como empresa comercializadora de todos los productos y servicios informáticos que se desarrollan en la UCI.
- Definir los dominios de información.
- Refinar dominios y determinar servicios a implementar.
- Implementar servicios.
- Probar servicios.

Idea a defender: La implementación de la Arquitectura Orientada a Servicios teniendo en cuenta la reducción de costo y tiempo en el desarrollo de aplicaciones, así como la reutilización de módulos ya existentes, asistirá a la mejoría de los procesos de Gestión de Proyectos en ALBET de una manera rápida y flexible, facilitando elevar la efectividad en el seguimiento de proyectos, la toma de decisiones, la productividad de los empleados, el fortalecimiento de las relaciones con clientes y proveedores, así como la competitividad de la empresa.

Uno de los objetivos centrales de cualquier trabajo de investigación, es conocer la esencia del fenómeno analizado y sus formas de manifestación, por esta razón la aplicación consecuente de una metodología adecuada es condición fundamental a tales efectos.

Los **Métodos de Investigación** como procedimiento que se utiliza para estudiar la realidad, la naturaleza, la sociedad y el pensamiento con el propósito de descubrir su esencia y relaciones para dar cumplimiento al objetivo trazado y llegar a conclusiones, se muestran a continuación:

Para la siguiente investigación, el método fundamental utilizado lo constituye el **Materialismo Dialéctico** con sus principios: Vinculación e interacción de los objetos y fenómenos como un todo coherente,

movimiento perspectivo de cambio y renovación, cambios cualitativos a partir de los cuantitativos en forma de espiral ascendente y solución de las contradicciones internas a través de su agudización, lo cual da origen a lo nuevo.

Entre los métodos de investigación utilizados están los **Métodos Teóricos** con los que se crean las condiciones para ir más allá de las características fenomenológicas y superficiales de la realidad, posibilitando el conocimiento del estado de arte, su evolución en una etapa determinada y su relación con otros fenómenos.

Dentro de los Métodos Teóricos se utilizó el **Método Histórico- Lógico**, al observar el objeto de estudio, este se analizará de forma lógica en su desarrollo y movimiento de acuerdo a su cronología y las particularidades del desenvolvimiento actual de la empresa. Permitiendo estudiar de forma analítica la trayectoria histórica real de los fenómenos, su evolución y desarrollo. Su objetivo fundamental dentro de esta investigación es constatar teóricamente cómo ha evolucionado el fenómeno en un período de tiempo, en toda su trayectoria o en un fragmento temporal de la lógica de su desarrollo.

El **Análisis y la Síntesis** deben estar presentes en todo el desarrollo de la investigación al revelarse cada uno de los elementos que conforman el tema en cuestión, en sus detalles y partes componentes, vistos como un todo integral. Permite buscar la esencia de los fenómenos, los rasgos que lo caracterizan y lo distinguen, además de analizar las teorías, documentos, permitiendo la extracción de los elementos más importantes que se relacionan con el objeto de estudio.

El **Método Inductivo – Deductivo** como forma de razonamiento que permite llegar a un grupo de conocimientos generalizadores, tanto desde el análisis de lo particular a lo general, como desde el análisis de elementos generalizadores a uno de menor nivel de generalización.

Otros métodos utilizados fueron los **Métodos Empíricos** permitiendo extraer de los fenómenos analizados las informaciones que se necesitan sobre ellos a través de las técnicas como:

La **Observación** permitió investigar el fenómeno desde su manifestación externa. La misma se realizó con el objetivo de obtener un registro real de lo que ocurre realmente en los procesos de gestión de proyectos de ALBET.

Entrevista a Expertos, cuyo objetivo es la riqueza, profundidad y calidad de la información, por tanto se le aplicó a los especialistas de la Dirección de Proyecto, así como a los especialistas de la empresa en general con el propósito de obtener una información certera acerca del objeto de estudio, pues ellos son los más indicados y conocedores del tema en cuestión.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA.

1.1 Introducción.

En este capítulo se exponen los conceptos, tecnologías, tendencias y herramientas que pueden ser utilizados en la implementación de la Arquitectura Orientada a Servicio en los Procesos de Gestión de Proyecto de ALBET, Ingeniería y Sistemas.

1.2 Las tecnologías de la información y las comunicaciones.

Desde los años ochenta se comienza a hablar de las denominadas “Nuevas Tecnologías de la Información y las Comunicaciones”, en las que están presentes las redes globales de computadoras conectadas a través de sistemas de transmisión de datos a grandes velocidades, utilizando cables coaxiales, fibras ópticas, radio, satélites y telefonía.

Diversas son las definiciones de estas llamadas “nuevas tecnologías” basadas en el desarrollo de equipos y dispositivos para procesar, transmitir y almacenar grandes cantidades de datos. Merce Gisbert en 1992, las declara como: “conjunto de herramientas, soportes y canales para el tratamiento y acceso a la información” (3). Adell las definen como “el conjunto de procesos y productos derivados de las nuevas herramientas (hardware y software), soportes de la información y canales de comunicación relacionados con el almacenamiento, procesamiento y transmisión digitalizados de la información” (4). Orlando Bello Hernández las define “como todos aquellos medios puestos al servicio de la mejora de la comunicación y el tratamiento de la información, que han ido surgiendo de la sinergia de los avances tecnológicos, modificando los procesos técnicos básicos de la comunicación” (3).

En todas estas definiciones de Tecnologías de la Información y las Comunicaciones (TIC), existen puntos comunes que las hacen singulares y únicas; se entenderán como un conjunto diverso de herramientas y recursos tecnológicos (hardware y software) que se utilizan para procesar, almacenar y transmitir información digitalizada.

Se puede resumir entonces que las Tecnologías de la Información y las Comunicaciones son las computadoras, sus periféricos, los programas que estas soportan y las redes que las interconectan. Son

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

medios tangibles y no tangibles que permiten la recepción, procesamiento, almacenamiento y transmisión de la información digitalizada.

Las organizaciones intentan adherir dos visiones para realizar su negocio, la visión del negocio para especificar y mejorar sus procesos y la visión de las tecnologías de la información para informatizar dichos procesos evolucionando en la tecnología (desarrollo del software). En general esta unión ha sido compleja y problemática sin alcanzar una visión común del negocio ya que las organizaciones son cada vez más dependientes de sus sistemas informáticos, contando con diversidad de sistemas que tienen entre sí dependencias complejas y que han ido creciendo en forma separada y heterogénea.

Al comienzo, el desarrollo de aplicaciones se enfocaba en necesidades específicas: contabilidad, compras, nómina, pedidos. Cada aplicación era construida sin tener en cuenta otros sistemas de la empresa, ni cómo comunicarse con ellos. Dado que las aplicaciones eran "auto-suficientes", la información común a la empresa completa (la dirección de un cliente) y funciones de negocio comunes (encontrar un cliente) requerían de codificación compleja para todos o varios de los sistemas independientes. Como resultado, los diferentes sistemas de Tecnología de Información de la mayor parte de las empresas de hoy, no pueden tener acceso, ni procesar datos unos con otros. Un proceso único que se tomaría segundos si los sistemas se pudieran comunicar, se puede tomar semanas. Hoy los procesos de negocios necesitan de agilidad y rapidez para adaptarse a las necesidades y condiciones del mercado, esto puede incluir nuevos clientes, socios, o materiales para ellos, por lo que se puede decir, que los días en los que una empresa podía operar eficientemente con aplicaciones independientes, corriendo sobre sistemas separados y desconectados, terminaron. La inmediatez, exactitud y seguridad de los datos, de un extremo de un proceso de negocio hasta el otro, es ahora un mandato del negocio. Las organizaciones que pueden cumplirlo, tienen una ventaja competitiva diferente. El objetivo final es lograr un sistema distribuido donde las compañías sean capaces de desarrollar procesos de negocios que cubran toda la empresa basadas en el servicio al cliente y admitiéndole a este cambiar procesos como respuesta a las rápidas y cambiantes necesidades del negocio.

En la actualidad para lograr este objetivo todas empresas están llamadas a soportar una Arquitectura Orientada a Servicios (SOA), tanto en empresas pequeñas, en crecimiento, como en grandes corporaciones debido a que SOA no es una moda pasajera, sino que es una alternativa que permite enfrentarse a los retos de la organización, que necesitan una respuesta flexible y rápida en entornos competitivos y de alta complejidad.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.3 Arquitectura Orientada a Servicios (SOA)

Son muchas las definiciones de la Arquitectura Orientada a Servicios, algunas de estas son:

Una Arquitectura Orientada a Servicios es un conjunto de componentes que pueden ser invocados y cuyas descripciones de interfaz se pueden publicar y describir. (5)

SOA es un estilo arquitectónico cuyo objetivo es lograr el acoplamiento entre agentes de software que interactúan. Un servicio es una unidad de trabajo realizado por un proveedor de servicios para lograr los resultados finales de un servicio al consumidor. Tanto el consumidor, como el proveedor desempeñan funciones por agentes de software en nombre de sus propietarios. (6)

La Arquitectura SOA establece un marco de diseño para la integración de aplicaciones independientes, de manera que desde la red pueda accederse a sus funcionalidades, las cuales se ofrecen como servicios. La forma más habitual de implementarla es mediante Servicios Web, una tecnología basada en estándares e independiente de la plataforma, con la que SOA puede descomponer aplicaciones monolíticas en un conjunto de servicios e implementar esta funcionalidad en forma modular. (7)

La estrategia de orientación a servicios permite la creación de servicios y aplicaciones compuestas que pueden existir con independencia de las tecnologías subyacentes. En lugar de exigir que todos los datos y lógica de negocio residan en un mismo ordenador, el modelo de servicios facilita el acceso y consumo de los recursos de IT a través de la red. Puesto que los servicios están diseñados para ser independientes, autónomos y para interconectarse adecuadamente, pueden combinarse con suma facilidad en aplicaciones complejas que respondan a las necesidades de cada momento en el seno de una organización. Las aplicaciones compuestas (también llamadas “dinámicas”) son las que permite a las empresas mejorar y automatizar sus procesos manuales, disponer de una visión consistente de sus clientes y socios comerciales, orquestar sus procesos de negocio para que cumplan con las regulaciones legales y políticas internas. El resultado final es que las organizaciones que adoptan la orientación a servicios pueden crear y reutilizar servicios y aplicaciones, adaptarlos ante los cambios evolutivos que se producen dentro y fuera de ellas, y con ello adquirir la agilidad necesaria para ganar ventaja competitiva. (7)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.3.1 Ventajas de SOA.

Las soluciones SOA permiten:

Mejorar la toma de decisiones: Al integrar el acceso a los servicios e información de negocio dentro de un conjunto de aplicaciones dinámicas compuestas, los directivos disponen de más información y de mejor calidad (más exacta y actualizada). Las personas, procesos y sistemas que abarcan múltiples departamentos pueden introducirse de forma más directa en una panorámica unificada, lo que permite conocer mejor los balances de costes y beneficios que se producen en las operaciones de negocio que se realizan a diario. (7)

Mejorar la productividad de los empleados: Un acceso óptimo a los sistemas, la información y la posibilidad de mejorar los procesos, permiten a las empresas aumentar la productividad individual de los empleados. Estos pueden dedicar sus energías a los procesos importantes, los que generan valor añadido y a actividades de colaboración, semiestructuradas, en vez de aceptar las limitaciones y restricciones impuestas por los sistemas de IT rígidos y monolíticos. (7)

Potenciar las relaciones con clientes y proveedores: Los beneficios que ofrece SOA trascienden los límites de la propia organización. Los procesos de fusión y compra de empresas se hacen más rentables al ser más sencilla la integración de sistemas y aplicaciones diferentes. La integración comercial y la optimización de los procesos de la cadena de suministro son, bajo esta perspectiva, objetivos perfectamente asequibles. (7)

Aplicaciones más productivas y flexibles: La estrategia de orientación a servicios permite a IT conseguir una mayor productividad de los recursos de IT existentes –como pueden ser las aplicaciones y sistemas ya instalados e incluso los más antiguos- y obtener mayor valor de ellos de cara a la organización sin necesidad de aplicar soluciones de integración desarrolladas ex profeso para este fin. (7)

Desarrollo de aplicaciones más rápido y económico: El diseño de servicios basado en estándares facilita la creación de un repositorio de servicios reutilizables que se pueden combinar en servicios de mayor nivel y aplicaciones compuestas en respuesta a nuevas necesidades de la empresa. Con ello se reduce el coste del desarrollo de soluciones y de los ciclos de prueba, se eliminan redundancias y se consigue su puesta en valor en menos tiempo. (7)

Aplicaciones más seguras y manejables: Las soluciones orientadas a servicios proporcionan una infraestructura común (y una documentación común también) para desarrollar servicios seguros,

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

predecibles y gestionables. Conforme van evolucionando las necesidades de negocio, SOA facilita la posibilidad de añadir nuevos servicios y funcionalidades para gestionar los procesos de negocio críticos. Se accede a los servicios y no a las aplicaciones, y gracias a ello la arquitectura orientada a servicios optimiza las inversiones realizadas en IT potenciando la capacidad de introducir nuevas capacidades y mejoras. (7)

1.3.2 Segunda generación de SOA

Una SOA de segunda generación está formada por un conjunto de Funciones y por la Calidad del Servicio.

Las Funciones (8) están formadas por:

Transporte: Mecanismo utilizado para trasladar las peticiones desde el cliente, hasta el proveedor del servicio y viceversa.

Protocolo de comunicación: Es el sistema de comunicación entre el cliente y el proveedor de servicios.

Descripción del servicio: Es un esquema utilizado para describir qué servicio es, cómo se le puede invocar y cuáles son los datos necesarios para realizar su invocación.

Servicio: Es la implementación del servicio.

Proceso de negocio: Es una colección de servicios, invocados en una determinada secuencia, con un conjunto particular de reglas para satisfacer un requisito de negocio.

Registro de servicios: Es un repositorio de servicios y datos, usado por los proveedores de servicios para publicarlos y para los clientes, donde buscarlos.

La calidad del servicio por: (8)

Política: Conjunto de reglas bajo las cuales, un proveedor de servicio hace que el servicio esté disponible para los clientes (*WS-Policy*).

Seguridad: Con un conjunto de reglas que podrían ser aplicadas en la identificación, autorización y control de acceso a los servicios, por parte del cliente (*WS-Security*).

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Transacción: Conjunto de atributos que podrían ser aplicados sobre un grupo de servicios para devolver un conjunto de datos consistentes (*WS-Transaction*, *WS-Coordination*).

Gestión: Conjunto de atributos que podrían ser aplicados para gestionar los servicios proporcionados (*WS-Manageability*).

1.3.3 Principales propiedades de SOA.

Una Arquitectura Orientada a Servicios es una forma de arquitectura de sistemas distribuidos, caracterizada por las siguientes propiedades. (9)

Vista lógica: Es una vista que proporciona una imagen de los componentes del sistema, tales como bases de datos, procesos de negocios, programas, explicando que hace cada uno de ellos, normalmente llevándolos a la operación del nivel del negocio.

Orientación al mensaje: Se define el servicio en términos de los mensajes intercambiados por el agente solicitante y el agente proveedor. En SOA se ha abstraído algunas características de los agentes, tales como la estructura de la base de datos, sus lenguajes de implementación, estructuras de procesos, SOA se preocupa por los detalles que son expuestos en la descripción del servicio.

Orientación a la descripción: Un Servicio Web es descrito por metadatos procesables por máquina. La descripción debe soportar la naturaleza pública de SOA. La semántica del servicio debe ser definida completamente en su descripción.

Granularidad: Los servicios deben tener la tendencia a realizar un pequeño número de operaciones con una gran cantidad de mensajes.

Orientación a la red: Los Servicios Web deben conservar la tendencia de ser concebidos para ser usados sobre una red, sin embargo no es un requerimiento absoluto.

Plataforma Neutral: Los mensajes deben ser creados para una plataforma neutral, utilizando un lenguaje estándar a través de las interfaces. XML es el lenguaje que mejor cumple con esta restricción.

1.3.4 Principios (10) de la orientación a servicios

Acople bajo: Los servicios tienen relaciones que mantienen las dependencias bajas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Contratos de servicios: Los servicios se adhieren a acuerdos de comunicación.

Autonomía: Los servicios tienen control sobre la lógica que encapsulan.

Abstracción: Los servicios ocultan la lógica al mundo exterior, sólo se muestra el contrato.

Rehúso: La lógica se divide en servicios con la intención de promover el rehúso.

Composición: Colecciones de servicios pueden ser coordinadas y ensambladas en composiciones de servicios.

Ausencia de estado: Los servicios minimizan el mantenimiento de información sobre una actividad específica.

Descubrimiento: Los servicios se diseñan de modo que puedan ser descubiertos a través de mecanismos para ello.

1.3.5 SOA define diferentes capas de software (10)

Aplicación básica: Sistemas desarrollados bajo cualquier arquitectura o tecnología, geográficamente dispersos y bajo cualquier figura de propiedad.

Exposición de funciones: Son expuestas en forma de servicios.

Integración de servicios: Facilita el intercambio de datos entre elementos de la capa aplicativa orientada a procesos empresariales internos o en colaboración.

Composición de procesos: Los definen en términos de negocio y sus necesidades, y varía en función del negocio.

Entrega: Los servicios son desplegados a los usuarios finales.

Cuando esta arquitectura se implementa a través de la plataforma tecnológica webservices, SOA brinda el potencial para soportar y proveer estos principios a lo largo de los procesos y los dominios de automatización de una empresa.

1.4 Servicios Web

La Web fue creada en sus inicios para compartir información entre los científicos. Hoy en día es consumida por gobiernos, empresas e individuos para que su información sea accesible vía “Web”. Sin embargo, gran cantidad de la información que hoy se comparte en el Web, es entendible sólo para los humanos o aplicaciones hechas a la medida para entenderla, por lo que, se transforma la Web en un

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

medio a través del cual la información pueda ser compartida, entendida y procesada por herramientas de forma automática.

Uno de los conceptos que están siendo desarrollados para poder lograr transformar el Web es el de Servicios Web. Los Servicios Web son la forma más habitual de implementar SOA. Los servicios Web son aplicaciones que utilizan estándares para el transporte, codificación y protocolo de intercambio de información. Permiten la intercomunicación entre sistemas de cualquier plataforma y se utilizan en una gran variedad de escenarios de integración, tanto dentro de las organizaciones como con clientes de negocios. Se basan en un conjunto de estándares de comunicación, como son: XML para la representación de datos, SOAP (Simple Object Access Protocol) para el intercambio de datos, el lenguaje WSDL (Web Services Description Language) para describir las funcionalidades de un servicio Web y UDDI (Universal Discovery Description and Integration) para la especificación de un registro distribuido de información (7). A continuación se detallan con mayor profundidad cada uno de estos estándares.

WSDL: Lenguaje de descripción de servicios web (WSDL, Web Service Description Language) es un estándar del sector reconocido para describir servicios web. WSDL se utiliza para definir los elementos clave de un servicio web, como por ejemplo el nombre, la dirección, las interfaces y las operaciones, así como los formatos de mensajes previstos. En los últimos años, se ha convertido en un estándar establecido y cuenta con el soporte de muchos proveedores de middleware. El estándar WSDL define un formato XML, pero no un estándar gráfico para el modelado. De modo similar, describe detalles técnicos, pero no la funcionalidad del servicio web. (11)

UDDI: (UDDI, Universal Discovery, Description and Integration) permite conocer con quién se van a comunicar y dónde encontrar otros objetos de negocio. Es una especificación para un registro distribuido de información acerca de los servicios Web. Define la forma en la cual se publica y descubre información acerca de éstos. (12)

SOAP: Protocolo de Acceso a Objetos (SOAP, Simple Object Access Protocol) es un protocolo liviano, basado en XML, para el intercambio de información estructurada en un ambiente descentralizado y distribuido. Es una especificación para la invocación de métodos en servidores, servicios, componentes y objetos, codifica la práctica existente de utilizar XML y HTTP como un mecanismo de invocación de métodos. Lo que SOAP permite es el paso de parámetros y comandos entre clientes y servidores de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

HTTP, independientemente de las plataformas y aplicaciones existentes en el cliente y en el servidor. Los parámetros y los comandos se codifican utilizando XML. (12)

1.4.1 Antecedentes históricos de los Servicios WEB

Internet por su diseño inicial exigía al usuario interactuar con el sistema. El usuario podía navegar, buscar, modificar, solicitar, realizar transacciones. Este sigue siendo el uso más común de Internet, y la red mundial sigue siendo el método principal de prestación de servicios basados en Internet. A medida que los protocolos basados en Internet se han ido generalizando en los sistemas utilizados en el entorno empresarial, ha ido aumentando la necesidad de integrar los sistemas informáticos de unas empresas con los de otras.

La “conversación” electrónica entre ordenadores ya existía desde hacía mucho tiempo: desde finales de los años setenta, cuando surgieron aplicaciones con capacidad de mensajería, que utilizaban formatos de comunicación normalizados, como ANSI X.12 y UN/EDIFACT. Pero la ausencia de normas universalmente aceptadas, el coste del desarrollo y mantenimiento de esas interfaces, el coste de los sistemas de telecomunicación patentados y el coste de integración de la información y las transacciones en las aplicaciones comerciales suponían que sólo unos pocos lograban triunfar en un camino sembrado de costosos fracasos. Ante una necesidad de estandarizar la comunicación entre distintas plataformas y lenguajes de programación se crearon diversos estándares los cuales fracasaron o no tuvieron el suficiente éxito, algunos de ellos son:

Sockets: Comunican dos procesos mediante datagramas o flujos de datos (streams). Requieren que las aplicaciones implementen sus propios protocolos para codificar y decodificar los mensajes que intercambian. Introducen complejidad en la solución, diferente a la del problema que se quiere resolver.

RPC (Llamada a Procedimiento Remoto): Es un protocolo que permite a un programa de ordenador ejecutar código en otra máquina remota sin tener que preocuparse por las comunicaciones entre ambos. Este protocolo es un gran avance sobre los sockets ya que de esta manera el programador no tenía que estar pendiente de las comunicaciones, estando éstas encapsuladas dentro de las RPC. Las RPC son utilizadas dentro del paradigma cliente-servidor, siendo el cliente el que inicia el proceso solicitando al

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

servidor que ejecute cierto procedimiento o función y enviando éste de vuelta el resultado de dicha operación al cliente.

RMI (Java Remote Method Invocation): Mecanismo ofrecido por Java para invocar métodos remotos. Al ser parte del entorno de ejecución de Java provee de un mecanismo simple para comunicar servidores en una aplicación distribuida que solamente necesite Java. Al estar específicamente diseñado para Java, RMI es muy amigable para los programadores, proveyendo el paso de objetos por referencia (no permitido por SOAP), "recolección de basura" distribuida y pasaje de tipos arbitrarios (funcionalidad no provista por CORBA). A través de RMI un programa Java puede exportar un objeto, a partir de esa operación este objeto está disponible en la red, esperando conexiones en un puerto TCP. Un cliente puede entonces conectarse e invocar métodos.

Net Remoting: Es una tecnología propietaria de Microsoft que permite crear aplicaciones distribuidas. Una de las principales características es la capacidad para poder trabajar desde una máquina con los objetos en memoria de la máquina Remota. Es una plataforma construida dentro del CLR que ofrece un conjunto único de características como canales, Objetos Proxy, Paso de Objetos, Tiempo de Vida, Hospedaje de Objetos en IIS, Modelos de Activación.

Tras estos intentos de estandarización en 1999, surge la idea de los Servicios Web, que se tradujo en el desarrollo de una serie de normas de la interfaz que permiten que sistemas informáticos distintos "dialoguen", precisamente cuando Microsoft publica su propuesta para un protocolo de comunicación XML, estándar llamado SOAP (Simple Object Access Protocol). Se buscaba encontrar un amplio consenso en cuanto a un protocolo de comunicación XML y un lenguaje para describir los Servicios Web (SOAP y WSDL respectivamente). Más adelante en mayo del 2000 diez compañías propusieron al w3c la versión 1.1 de SOAP, con esta base la w3c conformó el Grupo de Trabajo para Protocolos de XML el cual ayudó enormemente a la amplia aceptación y difusión de UDDI (Universal Description, Discovery and Integration), WSDL y SOAP, entre los años 2000 y 2001, lo cual sentó definitivamente las primeras bases sólidas de lo que conocemos hoy como Servicios Web. Aunque en esos tiempos se logró ponerse de acuerdo con los aspectos básicos de los Servicios Web, no siendo así en los aspectos tales como: seguridad, confiabilidad, transacciones o automatización de procesos de negocio lo cual alejan de momento la compleja autorización.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.4.2 Ventajas de los Servicios Web.

Son muchas las ventajas que actualmente ofrecen los servicios:

Interoperabilidad: La cual a su vez es la más beneficiosa debido a la filosofía misma de los Servicios Web, en donde no importa con qué lenguaje fue programado un Servicio Web, gracias a los métodos estándar de comunicación.

Usabilidad: La cual permite a los programadores de servicios web hacer uso de otros tantos para complementar su trabajo y evitar de este modo reinventar una y otra vez la rueda para satisfacer las necesidades particulares de cada cliente.

Facilidad de despliegue: En la medida en que los servicios web basan su presentación en tecnologías estándar de Internet que permiten por ejemplo, comunicación entre dos Servicios Web ubicados cada uno en lados opuestos del planeta.

1.4.3 Desventajas de los Servicios Web.

Los Servicios Web como la mayoría de las tecnologías nuevas, tienen algunas desventajas y/o aspectos por mejorar, los cuales son planteados a continuación:

Disponibilidad: Al tener que utilizar la misma infraestructura que utiliza un sitio web, no es prudente asegurar que la disponibilidad de un Servicio Web será del 100%. Si bien ya existen mecanismos tales como JMS (Java Messaging System) para reintentar una transacción varias veces, el problema sigue latente para la gran mayoría de Servicios Web basados exclusivamente en HTTP.

Satisfacción de requisitos: Debido a la filosofía misma de los Servicios Web en donde una sola talla le sirve a muchos clientes, la vieja costumbre del desarrollo de software de satisfacer esa pequeña característica extra que necesita un determinado cliente y que ningún otro cliente necesita, tendría que ser abolida.

Interfaces inmutables: Una vez que el Servicio Web se encuentre corriendo en producción, se deberá evitar, tanto como sea posible, modificar cualquiera de los métodos que provee dicho servicio web, así como los parámetros que espera el software del lado del cliente, para evitar de esta manera cualquier tipo de error de funcionamiento.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Falta de estándares: La seguridad y privacidad, la autenticación, la no repudiación y la forma en que se debe tarifar y establecer contratos, son temas delicados en los cuales no existe un estándar definido o existe, pero de forma incompleta.

Personal calificado: Como en los comienzos de cualquier otra tecnología, puede ser difícil e implícitamente costoso encontrar personal experimentado y calificado que implemente en una compañía el concepto de Servicios Web.

En general, SOA y Servicios Web son apropiados para aplicaciones empresariales:

- Que deben operar a través de Internet, donde la fiabilidad y la velocidad no se puede garantizar.
- Donde no existe habilidad de gestionar la instalación de forma que todos los solicitantes (clientes) y proveedores se actualicen a la vez.
- Donde los componentes de un sistema distribuido se ejecuten en distintas plataformas y distintos productos.
- Donde una aplicación existente necesite exponerse para ser usada a través de la red y pueda “decorarse” como un servicio Web.

En términos de modelos de interacción abstractos SOA requiere de las siguientes interacciones:

- Un servicio web publica su definición (WSDL) en un repositorio / directorio / registro distribuido (UDDI).
- El consumidor del servicio web busca la definición del servicio en el repositorio.
- El repositorio usa la información de la definición (WSDL) para enlazar con el servicio y enviar peticiones (SOAP) al servicio que ofrece el proveedor de servicios.

1.5 Tecnologías para desarrollar una Arquitectura Orientada a Servicios en los procesos de gestión de proyectos en Albet.

Para poner en marcha una Arquitectura Orientada a Servicios es necesario un conjunto de tecnologías que proporcionen el soporte necesario para la creación y mantenimiento de esta arquitectura en la empresa. Son varios los componentes que se utilizan para desarrollar y desplegar una SOA.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

1.5.1 Herramientas de soporte a los procesos de negocios.

Intalio BPM: Es un software Open Source basado en Java-J2EE, que implementa BPMS, y está basado en un conjunto de frameworks y arquitecturas muy conocidas en la industria del software y con una madurez aceptable. Intalio utiliza la notación para diseñar procesos de negocio establecida por el BPMN que puede adaptarse a los requisitos de SOA. Básicamente, Intalio proporciona un esquema de adopción sencillo, con bajos costos de propiedad, un soporte bastante amplio de estándares de la industria, una base de comunidades y desarrolladores que contribuyen continuamente con mejoras, corrección y detección de bugs, además cuenta con grandes facilidades para agregar nuevas características. (13)

Componentes:

- Es una herramienta para el diseño de los procesos de negocio, basada en Eclipse (ambientes gráficos para el desarrollo en java).
- Un engine que ejecuta los artefactos de software generados por el diseñador de procesos.
- Un servidor de aplicaciones donde residirán los servicios de procesos de negocios que se despliegue.

Modelo de Negocio

El modelo de negocio de Intalio, está basado en una licencia dual. IntalioBPMS se distribuye en 3 ediciones: La edición abierta de IntalioBPMS, bajo una licencia pública de Mozilla (MPL), una edición para la comunidad de IntalioBPMS, y la edición de IntalioBPMS Enterprise.

- La edición abierta incluye aproximadamente el 95% del código usado para la edición comunitaria y la de empresa. La edición abierta está desplegada sobre el servidor de Apache Geronimo J2EE, y la base de datos de MySQL.
- La edición comunitaria se distribuye con el servidor de IBM WebSphere, junto con MySQL.
- La edición empresarial puede desplegarse en otros servidores y bases de datos, su mayor características es el manejo transaccional.

1.5.2 Herramientas de soporte a Servicios Web.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

El modelo BPM/SOA no tiene por qué necesariamente estar basado en Servicios Web, pero hoy día generalmente todos los modelos BPM/SOA que son implementados y desplegados utilizan Servicios Web para de esta forma aprovechar los beneficios que brinda este modelo en el mayor grado posible.

1.5.2.1 Servidores web.

Los servidores web como soporte a estos Servicios Web son herramientas muy importantes para el desarrollo y despliegue de un escenario BPM/SOA. Un Servidor Web es una herramienta que sirve datos en forma de páginas Web, hipertextos o páginas HTML (del inglés HyperText Markup Language): textos complejos con enlaces, figuras, formularios, botones y objetos incrustados como animaciones o reproductores de sonidos. Un Servidor Web sirve contenido estático a un navegador, carga un archivo y lo sirve a través de la red al navegador de un usuario. Este intercambio es mediado por el navegador y el servidor que hablan el uno con el otro mediante el protocolo de transporte de Hipertexto (HTTP, HyperText Transport Protocol). Se pueden utilizar varias tecnologías en el servidor para aumentar su potencia más allá de su capacidad de entregar páginas HTML; estas incluyen scripts CGI, seguridad SSL y páginas activas del servidor (ASP). Existen Servidores Web propietarios muy conocidos como el Microsoft Internet Information Server (IIS) el cual funciona en el 35% de los servidores de todos los sitios web existentes, pero los Servidores Web Open Source son ampliamente utilizados.

Apache: Es el servidor web por excelencia, su configurabilidad, robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este programa. Presenta entre otras características: mensajes de error altamente configurables, bases de datos de autenticación y negociado de contenido. Su flexible sistema modular, permite cargar y descargar módulos sin necesidad de tocar el kernel. Apache permite personalizar la respuesta ante los posibles errores que se puedan dar en el servidor. Este Servidor corre en una multitud de Sistemas Operativos (Unix, Linux, MacOSX, Vms, Win32, OS2), lo que lo hace prácticamente universal. Es una tecnología gratuita de código fuente abierto. El hecho de ser gratuita es importante pero no tanto como que se trate de código fuente abierto. Esto le da una transparencia a este software de manera que si queremos ver que es lo que estamos instalando como servidor, lo podemos saber, sin ningún secreto, sin ninguna puerta trasera. (14)

Es un potente servidor Web que ofrece una infraestructura de gran fiabilidad, capacidad de manejo y escalabilidad para aplicaciones Web. IIS hace posible que las organizaciones aumenten la disponibilidad

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

de sus sitios y aplicaciones Web y a la vez reducir sus costes administrativos. Soporta la Iniciativa de Sistemas Dinámicos de Microsoft (DSI) con monitorización de estado de salud automático, aislamiento de procesos y capacidades de gestión mejoradas. Actualmente IIS es el segundo sistema de servidor web más popular (funciona en el 35% de los servidores de todos los sitios web). (14)

Apache Tomcat: También conocido como Jakarta Tomcat funciona como un contenedor de servlets desarrollado bajo el proyecto Jakarta en la Apache Software Foundation. Tomcat implementa las especificaciones de los servlets y de JavaServer Pages (JSP) de Sun Microsystems. No es un servidor de aplicaciones sino un servidor web con soporte de servlets y JSPs el cual incluye un compilador Jasper, que compila JSPs convirtiéndolas en servlets. Dado que el mismo fue escrito en Java, funciona en cualquier sistema operativo que disponga de la máquina virtual Java. Es mantenido y desarrollado por miembros de la Apache Software Foundation y voluntarios independientes. Los usuarios disponen de libre acceso a su código fuente y a su forma binaria en los términos establecidos en la Apache Software Licence. (15)

La jerarquía de directorios de instalación de Tomcat incluye: (15)

Bin: Arranque, cierre, otros scripts y ejecutables.

Common: Clases comunes que pueden utilizar Catalina y las aplicaciones web.

Conf: Ficheros XML y los correspondientes DTD para la configuración de Tomcat.

Logs: Logs de Catalina y de las aplicaciones.

Server: Clases utilizadas solamente por Catalina.

Shared: Clases compartidas por todas las aplicaciones web.

Webapps: Directorio que contiene las aplicaciones web.

Work: Almacenamiento temporal de ficheros y directorios

1.5.2.2 IDEs de Desarrollo

El entorno de desarrollo integrado (IDE, Integrated Development Environment) es un ambiente de programación que ha sido empaquetado como un programa de aplicación, consiste en un editor de código, un compilador, un depurador y un constructor de interfaz gráfica. Los IDEs pueden ser

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

aplicaciones por sí solas o pueden ser parte de aplicaciones existentes. Un buen entorno de desarrollo de aplicaciones no debería requerir un código personalizado pues el código personalizado es caro y normalmente requiere habilidades muy especializadas para su desarrollo y mantenimiento. Proporciona capacidades de gestión de procesos de negocio mediante un repositorio de procesos y ofrece un medio para probar y depurar el proceso.

Eclipse: Es una plataforma de código abierto que se puede utilizar para diseñar programas de primer nivel, componentes y sitios web. El marco de trabajo de Eclipse permite a los desarrolladores construir sus mejores herramientas combinando los complementos de este en un entorno de herramientas personalizado, además de facilitar la interoperabilidad y la facilidad de uso de una serie de herramientas heterogéneas que se ejecutan bajo un entorno Eclipse común. La fuerza y la frescura de la comunidad han conseguido convertirlo en el entorno de desarrollo Java líder en el mercado. Eclipse es una gran estructura formada por un núcleo y muchos plug-ins que van conformando la funcionalidad final. Además, da soporte a todo tipo de proyectos que abarcan desde el ciclo de vida del desarrollo de aplicaciones como automatizar, documentar, generación de código y pruebas de errores.

Netbeans IDE: Es una herramienta de desarrollo para escribir programas Java pero también, gracias a sus módulos, muchos otros lenguajes, desde C y C++ hasta XML, HTML y Java Server Pages. Es fácil de instalar y de uso instantáneo y se ejecuta en varias plataformas incluyendo Windows, Linux y Mac OS X y Solaris. Dentro de sus características se puede señalar que cuenta con un editor de código con sistema de coloración de la sintaxis y de auto-completado, anotaciones y macros del código. Las funciones de compilación, debug, y despliegue no están vinculadas a una máquina virtual como la mayoría de los IDE. Además, dispone de herramientas de concepción visual para crear y manipular componentes visuales, junto a un ejército de asistentes y utilidades de gestión y generación de código.

Zend Studio for eclipse: Es un completo entorno integrado de desarrollo (IDE) para el lenguaje de programación PHP que además de servir como editor de texto para páginas, este proporciona una serie de ayuda desde la creación y gestión de proyecto hasta la depuración de código. Está escrito en Java, lo que puede ser utilizado en cualquier Sistema Operativo donde esté instalada la máquina virtual de java. Consta de dos partes en las que se dividen las funcionalidades, la parte del cliente y la del servidor, las cuales se instalan por separado, la del cliente contiene la interfaz de edición y de ayuda. Permite hacer depuraciones simples de scripts, aunque para disponer de toda la potencia de la herramienta de depuración hay que instalar el servidor y el módulo PHP. Disponible para desarrolladores profesionales

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

pues abarca todas las capacidades de edición de código necesario para desarrollar modernas aplicaciones empresariales. Entre sus principales características están:

- Code Assist (elements, PHPDoc, parameter hints, include statements).
- Real time error detection.
- Bookmarks.
- Open resource (File / function).
- Search PHP element.
- Project Include Path.
- New PHP Elements Wizards (Class, Interface) o Getter and Setter Generation.
- CVS.
- Refactoring.
- Debug / Profile.
- Web Server Debugging.
- Web Servers Management.
- Web Server Profiler.
- Web Services Editor.
- Web Services support (Wizard and Inspection).

1.5.2.3 Sistemas de Gestión de Base Datos (SGBD)

Un Sistema de Gestión de Base Datos es un conjunto de programas que permiten crear y mantener una Base de Datos. Es un software de propósito general que facilita a un usuario procesos como la definición de una base de datos (especificación de tipo, estructuras y restricciones de datos), construcción (guarda los datos en algún medio controlado por el mismo SGBD) y manipulación (realizar consultas, actualizarla, generar informes) asegurando al mismo tiempo la integridad, confidencialidad y la seguridad de la Base de Datos. Existen diversos tipos de SGBD algunos ejemplos son: Oracle, Microsoft SQL Server, PostgreSQL, MySQL, en sentido general todos coinciden en un conjunto de objetivos que le permitirá manejar de

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

forma clara una colección de datos que posteriormente se convertirá en información los cuales se desglosan a continuación: (14)

Abstracción de la información: Los SGBD ahorran a los usuarios detalles acerca del almacenamiento físico de los datos. El hecho de que una base de datos ocupe uno o cientos de archivos se hace transparente al usuario. (14)

Redundancia mínima: Un buen diseño de la base de datos evitará la aparición de información redundante. (14)

Consistencia: En aquellos casos en que no se haya logrado una redundancia nula, será necesario vigilar que aquella información que aparece repetida se actualice de forma coherente. (14)

Seguridad: Los datos almacenados en una base de datos pueden llegar a tener un gran valor. Los SGBD deben garantizar que esta información se encuentre segura frente a usuarios malintencionados, que intentan leer información privilegiada, frente a ataques que desean manipular o destruir la información. (14)

Integridad: Se tratan de adoptar las medidas necesarias para garantizar la validez de los datos almacenados, de proteger los datos ante fallos de hardware, datos introducidos por usuarios descuidados o cualquier otra circunstancia capaz de corromper la información almacenada. (14)

Respaldo y recuperación: Los SGBD deben de proporcionar una forma eficiente de realizar copias de respaldo de la información almacenadas en ellos, y de respaldar a partir de estas copias los datos que se hayan podido perder. (14)

Control de la concurrencia: En la mayoría de los entornos es habitual que sean muchas las personas que acceden a una base de datos de forma simultánea, tanto para almacenar como para recuperar información. Los SGBD deben de ser capaz de controlar estos accesos concurrentes, que podría derivar en inconsistencia. (14)

Tiempo de respuesta: El tiempo de respuesta del SGBD antes diferentes solicitudes sea mínimo. (14)

Postgre SQL

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Es uno de los SGBD más populares desarrollados bajo la filosofía de código abierto liberado bajo la licencia BSD (Licencia de Software Distribuidos) que aproxima sus datos a un modelo objeto-relacional. Este sistema de base datos además de ser potente, soporta distintos tipos de datos: además de soportes para los tipos base, también soporta datos de tipo fecha, monetario, elementos gráficos, datos sobre redes y cadenas de bit, también permite la creación de tipos propios e incorpora una estructura de datos array que constituye un conjunto o agrupaciones de variables del mismo tipo cuyo acceso se realiza por índices. (14)

Entre sus principales ventajas están: (14)

Añade funciones de diversa índole: Manejo de fechas, geométricas y orientadas a operaciones con redes.

Permite las declaraciones de funciones propias: Así como la definición de disparadores.

Soporta el uso de índices, reglas y vistas: Incluye herencia entre tablas, aunque no entre objetos ya que no existen por lo que se incluye entre los SGBD objeto-relacional. Admite la gestión de diferentes usuarios: Como también los permisos asignados a cada uno de ellos.

Velocidad de respuesta: La velocidad de respuesta que ofrece este SGBD con bases de datos relativamente pequeñas pueden parecer un poco deficiente, pero esta misma velocidad la mantiene al gestionar bases de datos realmente grandes.

MySql

MySQL es un sistema de gestión de bases de datos SQL, el cual lo desarrolla, distribuye y soporta MySQL AB. MySQL AB es una compañía comercial Open Source de segunda generación que une los valores y metodología Open Source con un exitoso modelo de negocio, fundada por los desarrolladores de MySQL. (15)

Características. (16)

- Soporte a multiplataforma.
- Procedimientos almacenados.
- Triggers.
- Vistas actualizables.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Soporte a VARCHAR.
- Soporte para SSL.
- Soporte completo para Unicode.
- Uso de multihilos mediante hilos del kernel.
- Seguridad: ofrece un sistema de contraseñas y privilegios seguro mediante verificación basada en el host y el tráfico de contraseñas está cifrado al conectarse a un servidor.
- Se permiten hasta 64 índices por tabla (32 antes de MySQL 4.1.2).

1.5.2.4 Lenguajes de programación.

JAVA

Java es un lenguaje originalmente desarrollado por un grupo de ingenieros de Sun. Originalmente llamado OAK por los ingenieros de Sun Microsystems, Java fue diseñado para correr en computadoras incrustadas. Sin embargo, en 1995, dada la atención que estaba produciendo la Web, Sun Microsystems la distribuyó para sistemas operativos como Microsoft Windows. Permite escribir programas de interfaz gráfica o textual. También se pueden correr programas de manera incorporada o incrustada en los navegadores web de Internet en forma de Java applets, aunque no llegó a popularizarse como se esperaba. Los programas en Java generalmente son compilados a un lenguaje intermedio llamado bytecode, que luego son interpretados por una máquina virtual. Esta última sirve como una plataforma de abstracción entre la máquina y el lenguaje permitiendo que se pueda escribir el programa y correrlo en cualquier plataforma.

Java reduce en un 50% los errores más comunes de programación con lenguajes como C y C++ al eliminar muchas de las características de éstos, entre las que destacan:

- Aritmética de punteros
- No existen referencias
- Registros (struct)
- Definición de tipos (typedef)
- Macros (#define)

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Necesidad de liberar memoria (free)
- El intérprete completo de Java que hay en este momento es muy pequeño, solamente ocupa 215 Kb de RAM.

Ventajas:

Orientado a objetos: Java implementa la tecnología básica de C++ con algunas mejoras y elimina algunas cosas para mantener el objetivo de la simplicidad del lenguaje. Java trabaja con sus datos como objetos y con interfaces a esos objetos. Soporta las tres características propias del paradigma de la orientación a objetos: encapsulación, herencia y polimorfismo.

Distribuido: Java se ha construido con extensas capacidades de interconexión TCP/IP. Existen librerías de rutinas para acceder e interactuar con protocolos como http y ftp. Esto permite a los programadores acceder a la información a través de la red con tanta facilidad como a los ficheros locales

Robusto: Java realiza verificaciones en busca de problemas tanto en tiempo de compilación como en tiempo de ejecución. La comprobación de tipos en Java ayuda a detectar errores, lo antes posible, en el ciclo de desarrollo. Java obliga a la declaración explícita de métodos, reduciendo así las posibilidades de error. Maneja la memoria para eliminar las preocupaciones por parte del programador de la liberación o corrupción de memoria.

Multithreaded: Al ser multithreaded, Java permite muchas actividades simultáneas en un programa. Los threads (a veces llamados, procesos ligeros), son básicamente pequeños procesos o piezas independientes de un gran proceso. Al estar los threads construidos en el lenguaje, son más fáciles de usar y muy robustos.

Seguro: Fue pensado para ser un lenguaje de red, por lo que intrínsecamente cuenta con características de seguridad que evitan la infección por virus o trampas de programación, comunes en programas hechos con otros lenguajes.

Sencillo: Fácil de aprender, sin necesidad de comprender complejos algoritmos.

Limitantes:

- La velocidad.
- Los programas hechos en Java no tienden a ser muy rápidos. Como los programas de Java son interpretados nunca alcanzan la velocidad de un verdadero ejecutable.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Java es un lenguaje de programación. Esta es otra gran limitante, por más que digan que es orientado a objetos y que es muy fácil de aprender sigue siendo un lenguaje y por lo tanto aprenderlo no es cosa fácil. Especialmente para los no programadores.

PHP

PHP Es un lenguaje de programación de páginas web del lado del servidor cuyas características principales son la independencia de plataforma y su gratuidad. Un lenguaje del lado del servidor es aquel que se ejecuta en el servidor web, justo antes de que se envíe la página a través de Internet al cliente. Las páginas que se ejecutan en el servidor pueden realizar accesos a bases de datos, conexiones en red, y otras tareas para crear la página final que verá el cliente. El cliente solamente recibe una página con el código HTML resultante de la ejecución de la PHP. (17)

Características: (17)

- Soporte para una gran cantidad de bases de datos: MySQL, PostgreSQL, Oracle, MS SQL Server, Sybase mSQL, Informix, entre otras.
- Integración con varias bibliotecas externas, permite generar documentos en PDF (documentos de Acrobat Reader) hasta analizar código XML.
- Ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación.
- Perceptiblemente más fácil de mantener y poner al día que el código desarrollado en otros lenguajes.
- Soportado por una gran comunidad de desarrolladores, como producto de código abierto, PHP goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y reparen rápidamente.
- El código se pone al día continuamente con mejoras y extensiones de lenguaje para ampliar las capacidades de PHP.
- Con PHP se puede hacer cualquier cosa que podemos realizar con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Para poder programar en PHP se requiere de un servidor preparado para ello. Como el lenguaje de programación es multiplataforma, cualquiera de los principales servidores web nos servirá para ello.
- PHP (Hypertext Preprocessor) es un lenguaje script (no se compila para conseguir códigos máquina si no que existe un intérprete que lee el código y se encarga de ejecutar las instrucciones que contiene éste código), para el desarrollo de páginas web dinámicas del lado del servidor, cuyos fragmentos de código se intercalan fácilmente en páginas HTML, debido a esto, y a que es de Open Source, es el más popular y extendido en la web.

Desventajas:

- Promueve creación de código desordenado y con un mantenimiento complejo.
- No posee adecuado manejo de unicode.
- Es muy difícil de optimizar.

1.5.2.5 Herramienta de prueba.

SoapUI

SoapUI es una herramienta para desarrollar, invocar, inspeccionar, simular y testear Web Services sobre HTTP (tanto los basados en WSDL como en REST). Es una completa herramienta de test que agiliza las labores de construcción y favorece la creación de Web Services robustos. Es capaz de generar, automáticamente, a partir de un descriptor de Web Service (WSDL o WADL), tests con todas las peticiones SOAP de las operaciones definidas en ese descriptor. Adicionalmente pueden crearse Web Services que simulen el comportamiento de un objeto verdadero. También es muy versátil en otros aspectos, pudiendo seleccionar el método HTTP, aceptando varios métodos de autenticación, ficheros adjuntos y permitiendo la incorporación de scripting que dinamice el comportamiento de peticiones y respuestas según nuestras necesidades.

Funcionalidades: (18)

- Soporte a SOAP 1.1 y 1.2.
- Incorpora un monitor SOAP para capturar y analizar el tráfico.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

- Inspecciona Web Services WSDL y REST (tanto WADL como WADLess) y los visualiza jerárquicamente.
- Genera automáticamente los tests y las peticiones SOAP de las operaciones definidas en el descriptor WSDL o WADL.
- Opcionalmente puede utilizarse el scripting de Groovy para que el comportamiento de los tests sea dinámico.
- Soporta varios métodos de autenticación: Basic, Digest, WS-Security y NTLM Web Service.
- Soporta diferentes tecnologías de ficheros adjuntos: MTOM, SOAP con Attachments, ficheros Inline para WSDL y MIME Attachments para REST.
- Verificación del contenido de mensajes con Xpath y Xquery.
- Versatilidad en la configuración de test de carga, pudiendo indicar límite (en tiempo o peticiones), el número de threads de ataque, el método HTTP de la petición (POST, GET...).
- Permite exponer Web Services de simulación con el contenido de respuesta personalizable. Opcionalmente puede utilizarse el scripting de Groovy para simular un comportamiento dinámico.

1.5.3 Lenguajes de modelado.

UML: es un lenguaje que proporciona un vocabulario y unas reglas para permitir una comunicación, centrándose en la representación gráfica de un sistema. Este lenguaje nos indica cómo crear y leer los modelos, pero no dice cómo crearlos, siendo este último el objetivo de las metodologías de desarrollo.

(19)

Objetivos: (19)

Visualizar: Permite expresar de una forma gráfica de forma que otro lo pueda entender.

Especificar: Permite especificar cuáles son las características de un sistema antes de su construcción.

Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.

Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que puede servir para su futura revisión.

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

Un modelo UML está compuesto por tres clases de bloques de construcción: (19)

Elementos: Los elementos son abstracciones de cosas reales o ficticias (objetos, acciones, etc.)

Relaciones: relacionan los elementos entre sí.

Diagramas: Son colecciones de elementos con sus relaciones.

Principales características: (20)

- Incluye estereotipos como mecanismo de extensibilidad.
- Aporta un lenguaje (Object Constraint Language) desarrollado por IBM para expresar restricciones mediante fórmulas bien formadas.
- Puede describir cualquier tipo de sistema en términos de diagramas orientados a objetos.

Ventajas de UML: (21)

- Es un lenguaje conocido
- Estándar
- Fácil de aprender
- Desventajas de UML
- No ha sido diseñado para modelar procesos de negocios.
- No está orientado al dominio del problema.
- Implica un enfoque orientado a objetos.
- Contradictorio con un enfoque “orientado al negocio”.
- Sólo lo conocen los expertos TI.

BPMN: Define un diagrama de procesos de negocio que está basado en la técnica de diagramas de flujo y adaptado para crear modelos gráficos de las operaciones de los procesos de la organización. Proporciona a los negocios la capacidad de entender sus procedimientos internos en una notación gráfica, facilitando a

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

las organizaciones la habilidad para comunicar esos procedimientos de una manera estándar. Es usado para comunicar una amplia variedad de información a una amplia variedad de audiencias. Está compuesto de un conjunto de elementos gráficos que facilitan el desarrollo de un solo diagrama entendible tanto por audiencias de negocios (analistas de negocios) como por audiencias técnicas (arquitectos de sistemas e ingenieros software). (22)

Objetivos: (23)

- Proveer una notación entendible para cualquiera desde el analista del negocio, el desarrollador técnico y hasta la gente propia del negocio.
- Crear un puente estandarizado entre el diseño de procesos de negocio y su implementación.
- Asegurar que los lenguajes para la ejecución de procesos de negocio puedan ser visualizados con una notación común.

Resultados: (23)

- Define la notación y semántica de un BPD (Business Process Diagram).
- Define un mapping de BPMN a BPEL4WS.
- Provee la capacidad de entender los procedimientos internos en una notación grafica y da a las organizaciones la habilidad de comunicarlos de una manera estándar.
- Mejora las capacidades de las notaciones de proceso de negocio tradicionales para manejar inherentemente los conceptos de procesos de negocio B2B.

1.5.4 Visual Paradigm.

Es una de las herramientas UML CASE del mercado, fue creada para el ciclo vital completo del desarrollo del software. Proporciona características como generación de código, ingeniería inversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a través del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Incorpora el soporte para el trabajo en equipo, que permite que varios desarrolladores

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.
(24)

1.6 Tecnologías a utilizar en la implementación de una SOA en los procesos de gestión de proyecto de ALBET.

Con los argumentos dados en los epígrafes anteriores y con un conocimiento del problema en cuestión se puede definir las herramientas y lenguajes para el desarrollo de la Arquitectura Orientada a Servicios en los procesos de gestión de proyectos de ALBET. En primer lugar se utilizará como servidor web el apache 2.2 por ser uno de los servidores Open Source más confiable, robusto y estable que existe en la actualidad instalado en millones de servidores por todo el mundo. Como lenguaje de programación se analizan los lenguajes PHP y Java, de los dos se decide utilizar PHP por las características del mismo y el potencial que nos brinda a la hora de crear servicios web y por considerar que la utilización del lenguaje Java requería de mucho recursos de hardware con los cuales no se cuenta en el procesos de desarrollo. Para la edición de código PHP, así como para la creación de los wsdl de los servicios web se utilizará Zend Studio for Eclipse por ser uno de los más completo entorno integrado de desarrollo para el lenguaje de programación PHP. Se hace un estudio de las potencialidades y debilidades de los dos SGBD descritos anteriormente se decide utilizar como servidor de base dato PostgreSQL y como cliente para su administración SQL Manager 2005 for PostgreSQL ya que es un ambiente gratuito de sencilla utilización que posee una interfaz muy intuitiva. Como herramientas CASE se utilizará Visual Paradigm para la creación de los diferentes diagramas, como los BPD (Diagrama de Proceso de Negocio) los cuales no son soportados por Rational Rose. Como lenguaje de modelado se utilizará UML para la creación de los diagramas de componentes y despliegue, así como BPMN para el desarrollo de los BPD, para las pruebas a los servicios web utilizaremos SoapUI 2.0.1.

1.7 Conclusiones.

Es este capítulo, se profundizó en los conceptos y definiciones necesarias para comprender el proceso de desarrollo de la Arquitectura Orientada a Servicios. Además se hizo un estudio de las tecnologías, lenguajes y herramientas que se tienen en cuenta para llevar a cabo el proceso de desarrollo, basado

CAPÍTULO 1: FUNDAMENTACIÓN TEÓRICA

completamente en software no propietario lo cual obedece a las políticas definidas por la empresa ALBET, Ingeniería y Sistema.

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO.

2.1 Introducción.

En este capítulo se realizará un modelado de negocio basado en procesos con el objetivo de llegar a describir todos los procesos de gestión de proyectos de la empresa ALBET, Ingeniería y Sistemas, con el objetivo de lograr un mayor entendimiento del funcionamiento de la misma, para facilitar de esta forma el desarrollo de una Arquitectura Orientado a Servicios en dichos procesos.

2.2 Modelo del negocio basado en procesos.

Existen dos puntos de vista distintos para el modelado del negocio, el primero como una representación de la manera en que una empresa hace negocios, orientado al valor/cliente (se orienta a explicar cómo la empresa crea valor para el cliente). El segundo como una representación de la estructura y funcionamiento de una empresa, orientado a la actividad/rol (hace énfasis en el modelado de procesos y actores de la empresa). En este caso por estar bien definidos los procesos de gestión de proyecto de ALBET utilizaremos el segundo caso.

Para conseguir sus objetivos, ALBET organiza sus actividades por medio de un conjunto de procesos de negocio. Cada uno de ellos se caracteriza por una colección de datos que son producidos y manipulados mediante un conjunto de tareas, en las que ciertos agentes, como trabajadores o departamentos, participan de acuerdo a un flujo de trabajo determinado. Estos procesos se hallan sujetos a un conjunto de reglas de negocio, que determinan la estructura de la información y las políticas de la empresa. La finalidad del modelado del negocio es describir cada proceso del negocio, especificando sus datos, actividades y reglas de negocio.


A través del modelado de las actividades y procesos se logra un mejor entendimiento del negocio y esto provee la oportunidad de mejorarlos, pues con la automatización se reducen errores, asegurando que los procesos se comporten siempre de la misma manera y brindando elementos que permitan visualizar los diferentes estados por los que pasan, de esta forma los administradores pueden asegurar una eficiente ejecución y recopilar la mayor cantidad de información posible. Es a través de la información que se obtiene de la ejecución diaria de los procesos, que se puede identificar posibles ineficiencias con el objetivo de actuar sobre ellas para erradicarlas, y de esta forma mejorar los procesos empresariales.

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

2.3 Descripción de la empresa.

El primer paso del modelado del negocio consiste en capturar los procesos de negocio de la organización. La definición del conjunto de procesos del negocio es una tarea crucial, ya que define los límites del proceso de modelado posterior, para esto es necesario conocer minuciosamente la empresa, su estructura y características.

ALBET Ingeniería y Sistemas posee los derechos comerciales de todos los productos y servicios que desarrolla la UCI y mediante la alianza con otras prestigiosas entidades ofrece soluciones integrales en la esfera de las tecnologías de la información y las comunicaciones. Sus resultados en la exportación la ubican como una de las principales empresas de su sector en el país, con una destacada participación en el Convenio de Cooperación Integral Cuba-Venezuela y con grandes perspectivas de crecimiento en su proyección internacional.

Albet, Ingeniería y Sistema		
 Equipo de desarrollo SOA GRUPO DE SOLUCIONES Y PROCESOS	Fecha:	Elaborado por:
	Responsable:	
	Cargo:	
Característica		
Como modelo novedoso de empresa socialista presenta características como las que se exponen a continuación, Principal Capital Humano: Profesores y Estudiantes. Alto Grado de Integración con Entidades Nacionales, Mínimos Activos Fijos, utilizando los ya existentes, Elevados Aportes al País de sus Utilidades, Gestión por Proyectos, Contribución a la Formación de los Estudiantes, Elevada Ética Revolucionaria, Empresa de la Batalla de Ideas.		
Valores		
ALBET promueve valores corporativos que le permitirá comprobar realmente quién es. Sus valores están en cada una de las soluciones, productos y servicios. Estos son: responsabilidad, con todo y para todos sus clientes. Profesionalidad, más que resultados, busca excelencia en los servicios. Trabajo en equipo, integración de conocimientos y desarrollo del potencial humano. Honestidad y transparencia en sus		

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

soluciones. Compromiso con el cliente con una atención al detalle. Innovación. Confianza como camino seguro para estimular el éxito. Excelencia y grandeza en sus proyecciones. Perfeccionamiento continuo.


Filosofía de trabajo

Potencia en todo momento un ambiente de Responsabilidad, Profesionalidad, Honestidad, Innovación, Confianza, Integridad y Compromiso con el cliente. Se distingue por agrupar en “Equipos Cuba”, a los mejores profesionales y especialistas de diferentes sectores y entidades del país, para brindar soluciones de alto valor agregado y elevado nivel científico. Sus profesionales trabajan integrados con un enfoque multidisciplinario que garantiza la excelencia en los servicios que ofrece. Es flexible y se ajusta a las necesidades del cliente. Desarrolla tanto en plataformas propietarias como libres. Potencia la soberanía e independencia tecnológica en sus soluciones y el uso de software libre si el cliente lo prefiere. Su meta es prestar servicios de excelencia, por lo que busca en todo momento satisfacer a sus clientes. Busca proporcionar mayor rentabilidad y resultados que contribuyan al bienestar institucional y social, trabaja sin descanso para que las soluciones, estén siempre al alcance del cliente.

Empresa con las que se relaciona

La acompañan a nivel nacional instituciones de alto prestigio. Algunas de ella son: La Universidad de las Ciencias Informáticas, Copextel, SEISA, Desoft, CEDAI, ICID, MIC, MES, MINED, MINSAP, CIMEX, ETECSA, ISDI. A nivel internacional la acompañan prestigiosas empresas como son: Gemalto, Teltronic, Tedial, Metaform, SOAIN Software Associates Corporation S.L

Para una mejor organización de la empresa esta se divide en áreas físicas y lógicas:

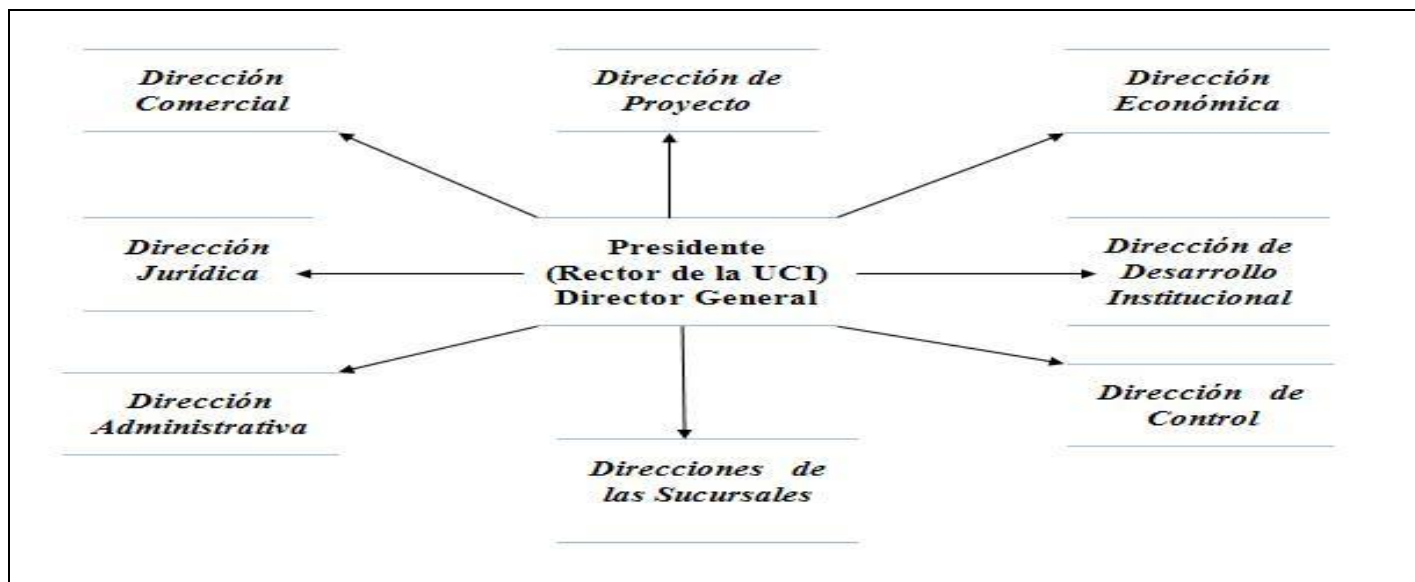
	Equipo de desarrollo	Fecha:	Elaborado por:
	SOA	Responsable:	
	GRUPO DE SOLUCIONES Y PROCESOS	Cargo:	
Departamentos o Áreas de la Empresa			
Nombre del Área	Descripción del Área		

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

Dirección de Proyecto	Se encarga de todo lo relacionado con la gestión de proyectos, la calidad la solución contratada, del cumplimiento del cronograma de ejecución, de la contratación de terceros y de la Plena satisfacción del cliente.
Dirección Comercial	Se encarga de todo lo relacionado con la gestión comercial, de lograr la continuidad con un nuevo proyecto, de lograr nuevos clientes, de convertir en productos los componentes del proyecto, del registro de la propiedad intelectual y de la gestión de compra - venta del proyecto.
Dirección Económica	Se encarga de la gestión económica-financiera, de lograr mayor utilidad que la planificada, de cumplir con los hitos de pagos contratados, de la efectiva gestión de cobros, del estricto control auditable y de la efectiva ingeniería financiera concertada.
Dirección Administrativa	Es la encargada de administrar todos los recursos con que cuenta la empresa con el fin de cumplir con los objetivos de la misma.
Dirección Jurídica	Se encarga de la Gestión Documental-Legal, de la gestión de todos los documentos legales, así como la firma de contratos y otros documentos.
Dirección de Desarrollo Institucional	Se encarga del perfeccionamiento institucional, del efectivo trabajo político ideológico, de la consagración y disciplina en el trabajo, de la ética y moral de la Revolución.
Dirección de Control	Esta es la encargada de llevar el control y seguimiento de todos los procesos que se están desarrollando en la empresa.

Diagrama de Estructuración de las Áreas de la Empresa

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO



2.4 Procesos de negocios.


Un proceso puede ser definido como un conjunto de actividades interrelacionadas entre sí que, a partir de una o varias entradas de materiales o información, dan lugar a una o varias salidas también de materiales o información con valor añadido, es decir, la manera en la que se hacen las cosas en la organización. (25)

Estos se caracterizan por ser: (26)

- Grandes y complejos.
- Muy dinámicos.
- Ampliamente distribuidos y particularizados.
- Larga duración.
- Automatizados.
- Dependientes de la inteligencia y el juicio humano.
- Difíciles de hacer visibles.

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

2.4.1 Procesos de aprobación de la solicitud de negocio.

 <p style="text-align: center;">Equipo de desarrollo SOA GRUPO DE SOLUCIONES Y PROCESOS</p>	Fecha:	Elaborado por:
	Área Organizativa:	
	Responsable:	
	Cargo:	
Número y nombre del proceso que lo antecede:	Número y nombre del proceso a describir: 1-. Proceso de aprobación de la Solicitud de Negocio.	
Código y nombre del proceso al que se subordina:	Código del proceso a describir: P1	
Objetivo: Analizar si la solicitud puede convertirse en un futuro negocio		
Clasificación: Proceso estratégico		
Relación de Entradas	Proveedores	
Solicitud de negocio	Cliente	
Relación de Salidas	Clientes	
Acta de aprobación de la solicitud	Dirección de Proyecto, Dirección Jurídica.	
Descripción textual del proceso		
<p>1. Solicitud de Negocio</p> <p>Las solicitudes de negocio pueden ser realizadas en Cuba, en alguna de las sucursales de la empresa en el extranjero o a través del sitio web de ALBET.</p> <p>Ejemplo:</p> <ul style="list-style-type: none"> ➤ Visitas a la Universidad de las Ciencias Informáticas ➤ Interés por algún producto de nuestra cartera ➤ Interés de una solución a la medida ➤ Identificación de nicho de negocio 		

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

➤ Visitas al sitio web de ALBET

1.1 Una vez que el solicitante muestra su interés de negocio con la empresa se le debe realizar una entrevista donde se obtengan como resultado los datos que se relacionan en el Anexo 1.

1.2 Se le presenta la solicitud de negocio al Director General de ALBET o a la máxima representación de ALBET en el sitio donde tenga lugar la solicitud de negocio y este designará la persona y/o Dirección que atenderá la solicitud. Esta respuesta no debe exceder las 24 h.

2. Atención de la solicitud

1 El responsable designado por la máxima Dirección se entrevistará nuevamente al solicitante con el objetivo de definir mejor sus intereses. Se obtendrá como resultado de esta entrevista los datos que se relacionan en el Anexo 2.

2 Esta planilla será analizada por la Dirección de Proyectos quien decidirá si se envía al Grupo Ejecutivo del Comité de Contratación o presentará sus criterios sobre la solicitud de negación directamente a la máxima Dirección de ALBET.

2.2.1 La dirección responsable enviará las planillas de solicitud al Grupo Ejecutivo del Comité de Contratación a la siguiente dirección de correo: ccontratacion@albet.cu.

2.2.2 La dirección responsable de atender la solicitud presenta la justificación de denegación a la solicitud y le presenta una respuesta al solicitante.

3. Estudio de factibilidad

3.1 El grupo operativo del CC se encargará de hacer el estudio de factibilidad del posible negocio.

- Recopilar información sobre cliente
- Realizar análisis costo/beneficio
- Analizar condiciones políticas
- Recopilar información sobre cuestión a negociar
- Analizar el contexto de la negociación
- Evaluar los recursos disponibles

El grupo operativo estará conformado por especialistas de la Dirección de Proyectos, Dirección Comercial, Dirección Económica, Dirección Jurídica y Dirección General de la IP; y será liderado por el Jefe del Grupo de Inteligencia Corporativa de ALBET.

3.2 Como resultado del estudio de factibilidad el grupo operativo generará un informe que enviará a

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

todos los miembros del CC a más tardar el viernes de cada semana (Anexo 3), al que se le adjuntarán las planillas 1 y 2.

4. Aprobación por el Comité de Contratación.

4.1 El CC teniendo en cuenta el informe de factibilidad, se reunirá y aprobará o rechazará las solicitudes recepcionadas.

4.1.1 En caso de no ser aprobada una solicitud, la dirección que la presentó le comunicará al cliente las causas por las que no fue aprobada, dejando abierta las puertas para futuras negociaciones.

4.1.2 En caso de ser aprobada una solicitud, la dirección que la presentó fijará una reunión con el cliente en vista a definir el cronograma de negociación.

Áreas organizativas implicadas

Dirección de Proyectos, Dirección Comercial, Dirección Económica, Dirección Jurídica y Dirección General de la IP

Políticas

Política	Documento	Fuente
La asignación por parte del Director General de ALBET del responsable de atender la solicitud no debe de exceder las 24 horas.	Aprobación de la solicitud de negocio.	Dirección de Proyectos.

Reglas de negocio asociadas al proceso

Regla de Negocio	Descripción de la regla

Indicadores:

Definiciones, anacrónicos, abreviaturas


Término	Descripción
CC	Comité de Contratación

Soporte del Proceso

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

Manual	Sistemas involucrados: Ninguno.
Comentario	

2.4.2 Procesos de elaboración de la ficha técnica.

 <p style="text-align: center;">Equipo de desarrollo SOA GRUPO DE SOLUCIONES Y PROCESOS</p>	Fecha:	Elaborado por:
	Área Organizativa:	
	Responsable:	
	Cargo:	
Número y nombre del proceso que lo antecede: 1-. Proceso de aprobación de la Solicitud de Negocio.	Número y nombre del proceso a describir: 2-. Proceso de elaboración de la ficha técnica	
Código y nombre del proceso al que se subordina:	Código del proceso a describir: P2	
Objetivo Definir alcance, recursos, presupuesto y precio del proyecto		
Clasificación: Proceso Estratégico		
Relación de Entradas	Proveedores	
Descripción del proyecto a realizar	Cliente	
Relación de Salidas	Clientes	
Ficha Técnica firmada	Dirección Jurídica	
Descripción textual del proceso		
1. Elaboración de la ficha del proyecto		
1.1 Para la elaboración de la ficha técnica del proyecto se acuerdan mesas de trabajo con el cliente. Se		

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

tendrán en cuenta los siguientes aspectos:

- Definición del alcance del proyecto.
- Definición de los recursos necesarios.
- Confección del presupuesto
- Gestión de oferta con los proveedores

La ficha se desarrollará en el formato establecido de acuerdo al marco de colaboración, ejemplo:

Marco del Convenio de Colaboración Cuba – Venezuela

1.2 Se realizarán encuentros sistemáticos con la contraparte hasta que se obtenga una versión acabada de la ficha.

1.3 La ficha técnica es presentada al CC para su aprobación.

Si la ficha no es aprobada por el CC, se ajusta según sugerencias y se presenta nuevamente.

Si la ficha es aprobada por el CC se presenta al cliente para su aprobación y firma.

Áreas organizativas implicadas

Dirección de Proyectos, Dirección Comercial, Dirección Legal, Dirección Económica.

Política	Documento	Fuente
La ficha se desarrollará en el formato establecido de acuerdo al marco de colaboración	Marco del Convenio de Colaboración Cuba – Venezuela	Dirección de Proyectos

Reglas de negocio asociadas al proceso

Regla de Negocio	Descripción de la regla


Indicadores:

Definiciones, anacrónicos, abreviaturas

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

Término	Descripción
CC	Comité de Contratación
Soporte del Proceso	
Manual	Sistemas involucrados : Ninguno
Comentario	

2.4.3 Proceso de creación de la oferta.

 <p style="text-align: center;">Equipo de desarrollo SOA GRUPO DE SOLUCIONES Y PROCESOS</p>	Fecha:	Elaborado por:
	Área Organizativa:	
	Responsable:	
	Cargo:	
Número y nombre del proceso que lo antecede: 2-. Proceso de Elaboración de la Ficha Técnica	Número y nombre del proceso a describir: 3-. Proceso de creación de la oferta.	
Código y nombre del proceso al que se subordina:	Código del proceso a describir: P3	
Objetivo Ver si el producto a negociar es rentable o no.		
Clasificación Proceso clave.		
Relación de Entradas	Proveedores	
Ficha Técnica, Ficha de Costo	Facultad, Dirección Jurídica	
Relación de Salidas	Clientes	
Oferta Aprobada	ALBET	


CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

Descripción textual del proceso		
<p>Para la elaboración de la Oferta se siguen los siguientes pasos.</p> <ol style="list-style-type: none"> 1. La directora general de proyecto asigna responsable de la creación de la oferta. 2. El responsable asignado por la dirección de proyectos hace un estudio del producto solicitado en mercado para ver cuál es el precio en el mercado de dicho producto. Este le pide a la facultad que va a ejecutar el futuro producto que realice la ficha de costo. 3. La facultad realiza la ficha de costo y se la envía al especialista encargado de la realización de la oferta. 4. El especialista hace un estudio para ver si es rentable el negocio y de ahí presenta la oferta al cliente. 5. El cliente si está de acuerdo firma la oferta, sino el especialista deja las relaciones abiertas para futuras negociaciones. 		
Áreas organizativas implicadas		
Dirección de Proyectos, Dirección Jurídica		
Políticas		
Política	Documento	Fuente
Reglas de negocio asociadas al proceso		
Regla de Negocio	Descripción de la regla	
Indicadores:		
Definiciones, anacrónicos, abreviaturas		
Término	Descripción	
Soporte del Proceso		

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

Manual	Sistemas involucrados : Ninguno
Comentario	

2.4.4 Procesos de contratación

 <p style="text-align: center;">Equipo de desarrollo SOA GRUPO DE SOLUCIONES Y PROCESOS</p>	Fecha:	Elaborado por:
	Área Organizativa:	
	Responsable:	
	Cargo:	
Número y nombre del proceso que lo antecede: 3-. Proceso de creación de la Oferta.	Número y nombre del proceso a describir: 4-. Proceso de Contratación.	
Código y nombre del proceso al que se subordina:	Código del proceso a describir: P4	
Objetivo Establecer y oficializar mediante la firma del contrato, acuerdos con los clientes.		
Clasificación: Proceso Clave		
Relación de Entradas	Proveedores	
Ficha técnica del proyecto.	Dirección Jurídica	
Relación de Salidas	Clientes	
Contrato, Anexos, Suplementos.	ALBET, Cliente	
Descripción textual del proceso		
<p>1. Realización del cronograma para la contratación.</p> <p>1.1 El GG de acuerdo con el cliente y cada una de las áreas implicadas, elabora el cronograma de</p>		

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

contratación que garantice la elaboración y firma de cada uno de los documentos que se necesiten para formalizar las negociaciones.

En la medida en que se fijan los requisitos con el cliente se van acordando las obligaciones con los proveedores.

2. Elaboración del contrato, los anexos y los suplementos (Responsable: GG)

El contrato es elaborado por el equipo de proyecto bajo la supervisión metodológica de cada una de las direcciones que lo rigen (Dirección Económica, Dirección Jurídica).

Para la elaboración de los documentos legales se deben realizar reuniones de conciliación con los clientes en pos de ir negociando lo que se va elaborando.

3. Aprobación de los documentos legales.

3.1 Cada documento legal es enviado por el GG a las entidades proveedoras de acuerdo al procedimiento para su revisión y aprobación. Este proceso será iterativo hasta obtener una aprobación final.

Aprobación de documentos legales en la UCI.

3.2 Con la aprobación de las entidades proveedoras, los documentos son enviados al CC hasta el jueves.

3.3 El CC se reúne el martes para aprobar los documentos analizados. La aprobación del CC quedará reflejada en acta, detallando en caso de ser necesario los ajustes que debe hacer el equipo de proyecto al documento en cuestión.

El CC al recibir el documento tiene cierta cantidad de días para aprobarlo y en el transcurso del plazo establecido puede emitir criterios.

3.4 El equipo de proyecto realizará los ajustes definidos por el CC.

4. Firma del contrato, los anexos y los suplementos.

El GG presenta la versión aprobada por el CC al equipo de contratación de la contraparte, la cual se someterá a conciliación final para proceder a la firma.


Si ambas partes están de acuerdo con los deberes y derechos expuestos, así como con las especificaciones acotadas en el mismo, se procede a la firma.

5. Una vez firmado el documento se le entrega el original a la Dirección Jurídica de ALBET para ser archivado. El GG y la Dirección Jurídica de la sucursal se quedan con una copia y la Dirección de

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

Proyectos de ALBET, la Dirección Económica y la Dirección Comercial con una copia digital.		
Áreas organizativas implicadas		
Dirección Económica, Dirección Jurídica, Dirección Comercial		
Políticas		
Política	Documento	Fuente
Reglas de negocio asociadas al proceso		
Regla de Negocio	Descripción de la regla	
Indicadores:		
Definiciones, anacrónicos, abreviaturas		
Término	Descripción	
GG	Gerente General	
CC	Comité de Contratación	
Soporte del Proceso		
Manual	Sistemas involucrados : Ninguno	
Comentario		

2.4.5 Procesos de registro de proyecto.

 <p>Equipo de desarrollo SOA GRUPO DE SOLUCIONES Y PROCESOS</p>	Fecha:	Elaborado por:
	Área Organizativa:	
	Responsable:	


CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

	Cargo:
Número y nombre del proceso que lo antecede: 4-. Proceso de contratación.	Número y nombre del proceso a describir: 5-. Proceso de registro de proyecto.
Código y nombre del proceso al que se subordina: P4	Código del proceso a describir: P4.1
Objetivo: Registrar toda la información que la empresa necesita controlar de los proyectos.	
Clasificación : Procesos estratégicos:	
Relación de Entradas	Proveedores
Contrato firmado	Dirección Jurídica
Relación de Salidas	Clientes
Proyecto, hito de pago, entregable, subproyecto, recursos humanos del proyecto, Ente ejecutor, entidad.	ALBET
Descripción textual del proceso	
<p>Una vez que son aprobados por el cliente los documentos legales se procede al registro de toda la información referente al proyecto.</p> <p>1-. El DGP asigna un especialista para el registro del proyecto.</p> <p>2-. El especialista asignado registra:</p> <ol style="list-style-type: none"> a. Los subproyectos, las entidades que lo ejecutan. b. Registra recursos humanos y su respectivo plan de misión. c. Registra los hitos de pagos y sus respectivos entregables. d. Registra el ente ejecutor financiero. 	
Áreas organizativas implicadas	
Dirección de Proyecto	
Políticas	

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

Política	Documento	Fuente
Reglas de negocio asociadas al proceso		
Regla de Negocio	Descripción de la regla	
Indicadores:		
Definiciones, anacrónicos, abreviaturas		
Término	Descripción	
DGP	Director General de Proyecto	
Soporte del Proceso		
Manual	Sistemas involucrados : Ninguno	
Comentario		

2.4.6 Proceso de gestión de suministro con proveedores.

	Equipo de desarrollo SOA GRUPO DE SOLUCIONES Y PROCESOS	Fecha:	Elaborado por:
		Área Organizativa:	
		Responsable:	
		Cargo:	
Número y nombre del proceso que lo antecede: 5-. Proceso de registro de proyecto.	Número y nombre del proceso a describir: 6 Proceso de gestión de suministro con proveedores		
Código y nombre del proceso al que se	Código del proceso a describir: P5		

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

subordina:			
Objetivo Proveer los suministros necesarios para el desarrollo de un proyecto.			
Clasificación: Proceso de Soporte y Apoyo			
Relación de Entradas		Proveedores	
Solicitud de suministro		Equipo de proyecto	
Relación de Salidas		Clientes	
Oferta, Contrato Proveedores, anexos, suministro		Equipo de proyecto	
Descripción textual del proceso			
<p>Ante la necesidad de adquirir un suministro.</p> <ol style="list-style-type: none"> 1. El GGP envía una solicitud a la DC de ALBET. 2. La DC de ALBET revisa la solicitud y la envía a un proveedor. 3. El proveedor hace una oferta a la DC de ALBET. 4. Esta oferta es valorada por la DP, el cliente y la DC de ALBET y puede ser aprobada o rechazada. 5. Si la oferta es aprobada se firmará por el cliente y/o el proyecto. <ol style="list-style-type: none"> a. Se crea el contrato y sus anexos respectivamente. b. Se registra y se envía el suministro. 6. Si la oferta es rechazada <ol style="list-style-type: none"> a. Se le pide al proveedor que haga una nueva oferta. 			
Áreas organizativas implicadas			
Dirección de Proyecto, Dirección Comercial, Dirección Jurídica			
Políticas			
Política	Documento	Fuente	

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

Reglas de negocio asociadas al proceso	
Regla de Negocio	Descripción de la regla
Indicadores:	
Definiciones, anacrónicos, abreviaturas	
Término	Descripción
GGP	Gerente General del Proyecto
DC	Dirección Comercial
DP	Dirección del Proyecto
Soporte del Proceso	
Manual	Sistemas involucrados : Ninguno
Comentario	

2.5 Diagramas de procesos.

Es un diagrama diseñado para ser usado por las personas que diseñan y administran procesos de negocio. (23)

Las cuatro categorías básicas de elementos que se pueden encontrar en un BPD (Business Process Diagram) son: Objetos de flujo, objetos de conexión, swimlanes, artefactos (23).

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

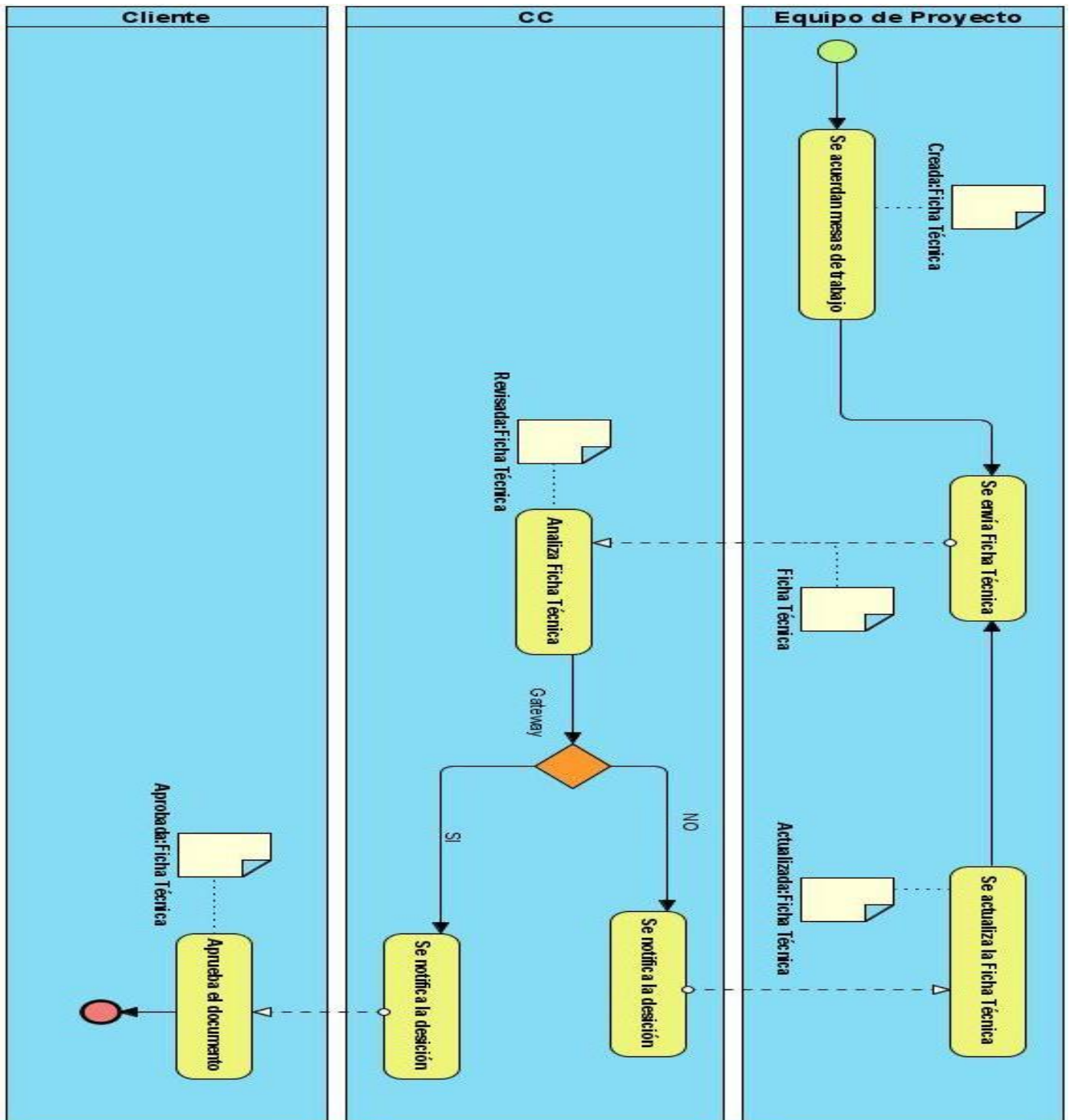


Figure 1 Diagrama del proceso elaboración de la ficha técnica.

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

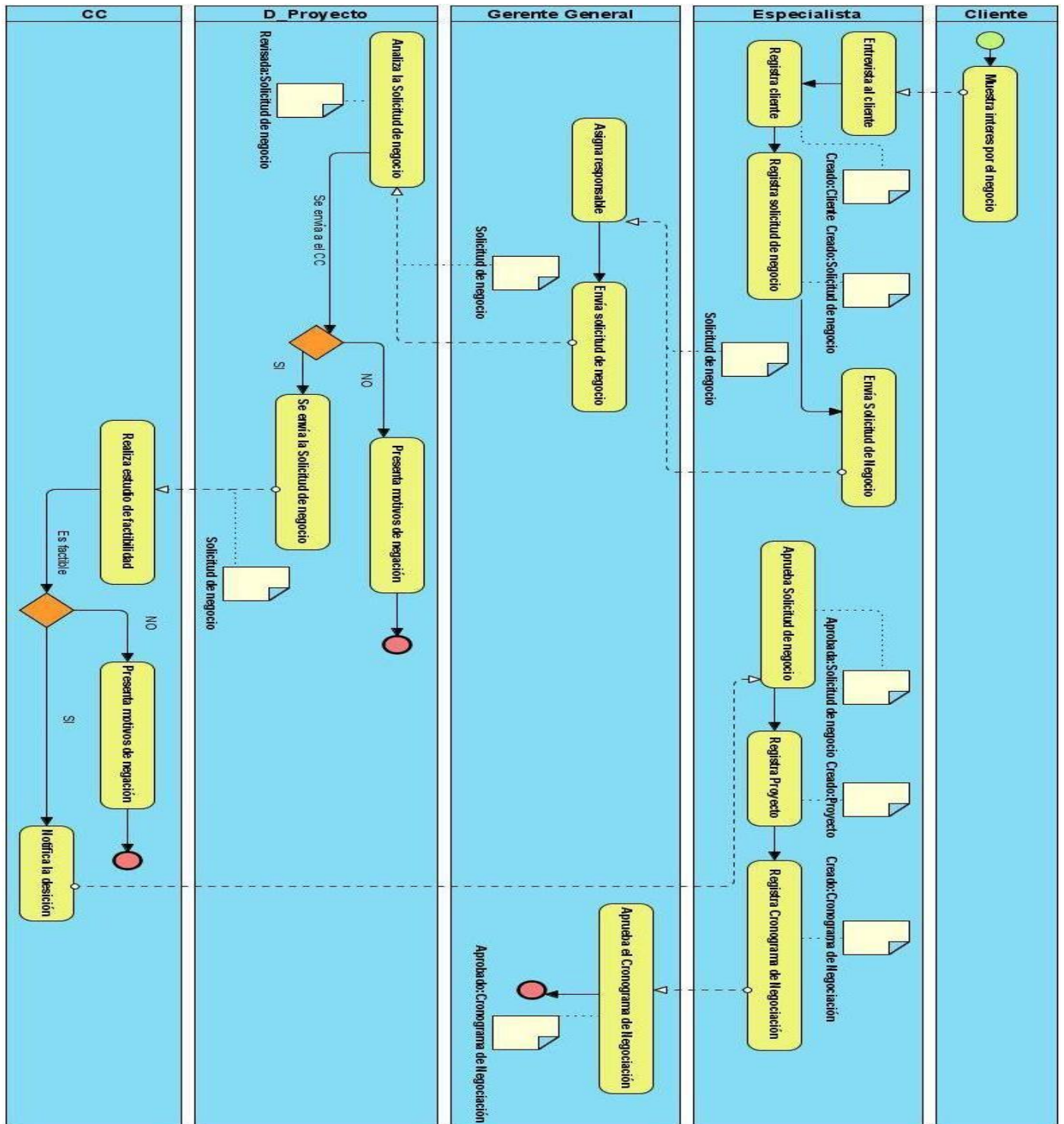


Figure 2 Diagrama del proceso aprobación de la solicitud de negocio.

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

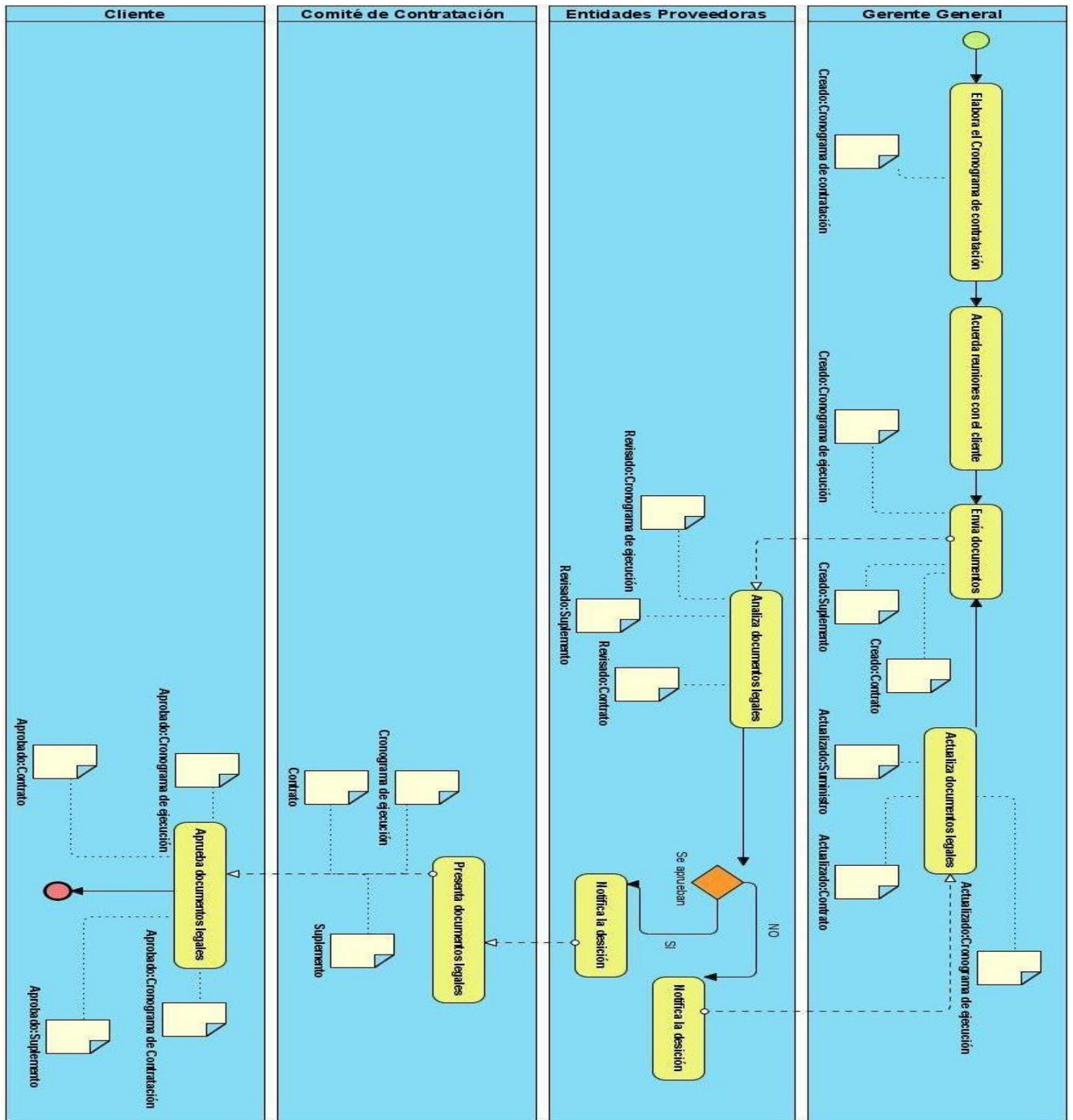


Figure 3 Diagrama del proceso de contratación.

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

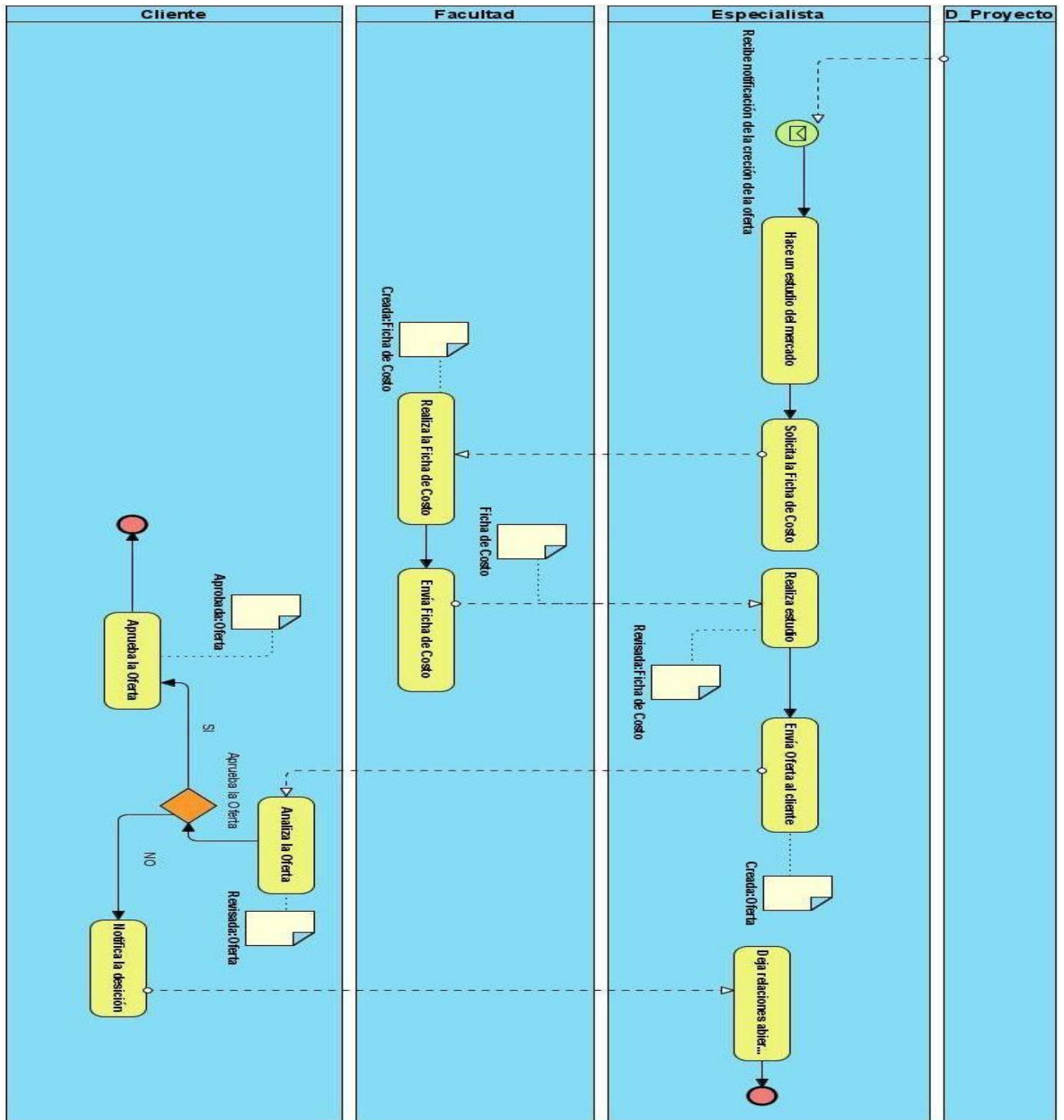


Figure 4 Diagrama del proceso creación de la oferta.

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

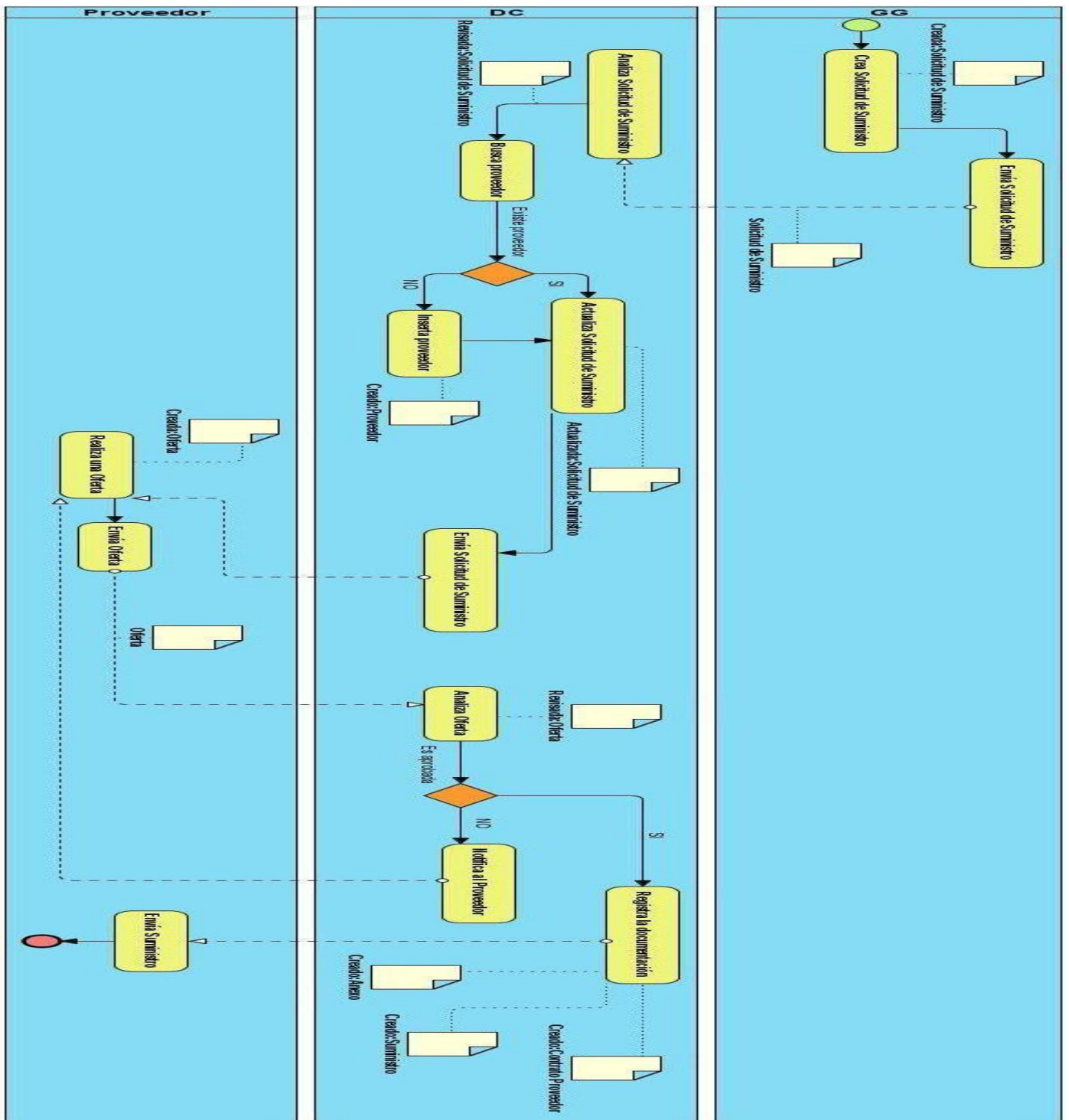


Figure 5 Diagrama del proceso gestión de suministro.

CAPÍTULO 2: MODELADO DE PROCESOS DE NEGOCIO

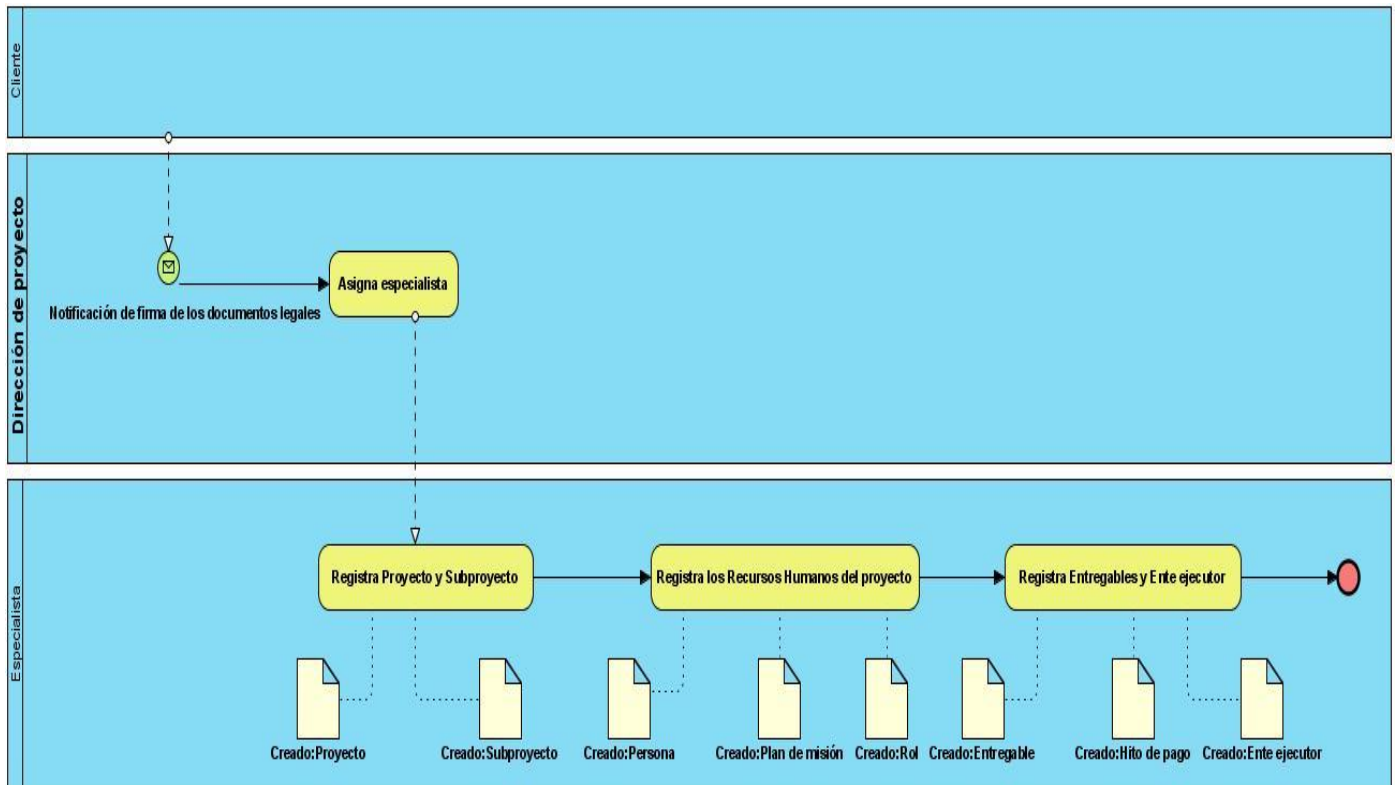


Figure 6 Diagrama del proceso ejecución de proyecto.

2.6 Conclusiones.

En este capítulo se describieron los diferentes procesos de gestión de proyecto de Albet, así como sus respectivos diagramas. Se descubrieron seis procesos de negocios que cubren todo el ciclo vida de un proyecto, los mismos son: Aprobación de la solicitud de negocio, Creación de la oferta, Contratación, Ejecución de proyecto, Aprobación de la ficha técnica y el proceso de Gestión de suministro con los proveedores.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS


CAPÍTULO 3: DISEÑO DE LOS SERVICIOS.

3.1 Introducción.

En este capítulo se llevará a cabo todo el proceso de diseño de los servicios. Hay que plantear que no existe en el ámbito libre ninguna metodología para el desarrollo de Arquitecturas Orientadas a Servicios. Estas metodologías o framework de trabajo como muchos la definen son de carácter propietario y no contamos con ellas en la Universidad. Entre ellas tenemos las metodologías SOAD (Service-oriented analysis and design), SOMA (Service-oriented modeling and architecture), CBDI (Component Based Development and Integration), teniendo todas ellas en común que no están lo suficientemente descritas para usarlas en este trabajo. La Universidad contrató los servicios de una consultora líder en este tema para la capacitación de personal. De este contrato quedaron las grabaciones de los cursos que los especialistas de dicha consultora impartieron, que unido a la evaluación de todas estas metodologías, fue utilizado por la Universidad a través del Grupo de Soluciones y Procesos para definir una guía de buenas prácticas y pasos para el diseño de los servicios en el desarrollo de los proyectos basados en SOA de la misma. Basándonos en esta guía se realizó el diseño de nuestro trabajo.

3.2 Identificación de los dominios de información.

Los dominios son la descomposición lógica del negocio por sus actividades, donde estas actividades van a tener vida propia y tienen relación con el resto de la organización. El dominio es más auténtico cuando tiene menor número de canales de comunicación con otros dominios. Es la asociación de entidades de negocios y procesos que tienen una misma proyección o una misma actividad.

 Equipo de desarrollo SOA GRUPO DE SOLUCIONES Y PROCESOS	Fecha:	Elaborado por:
DOMINIOS	PROYECTO	PROVEEDOR

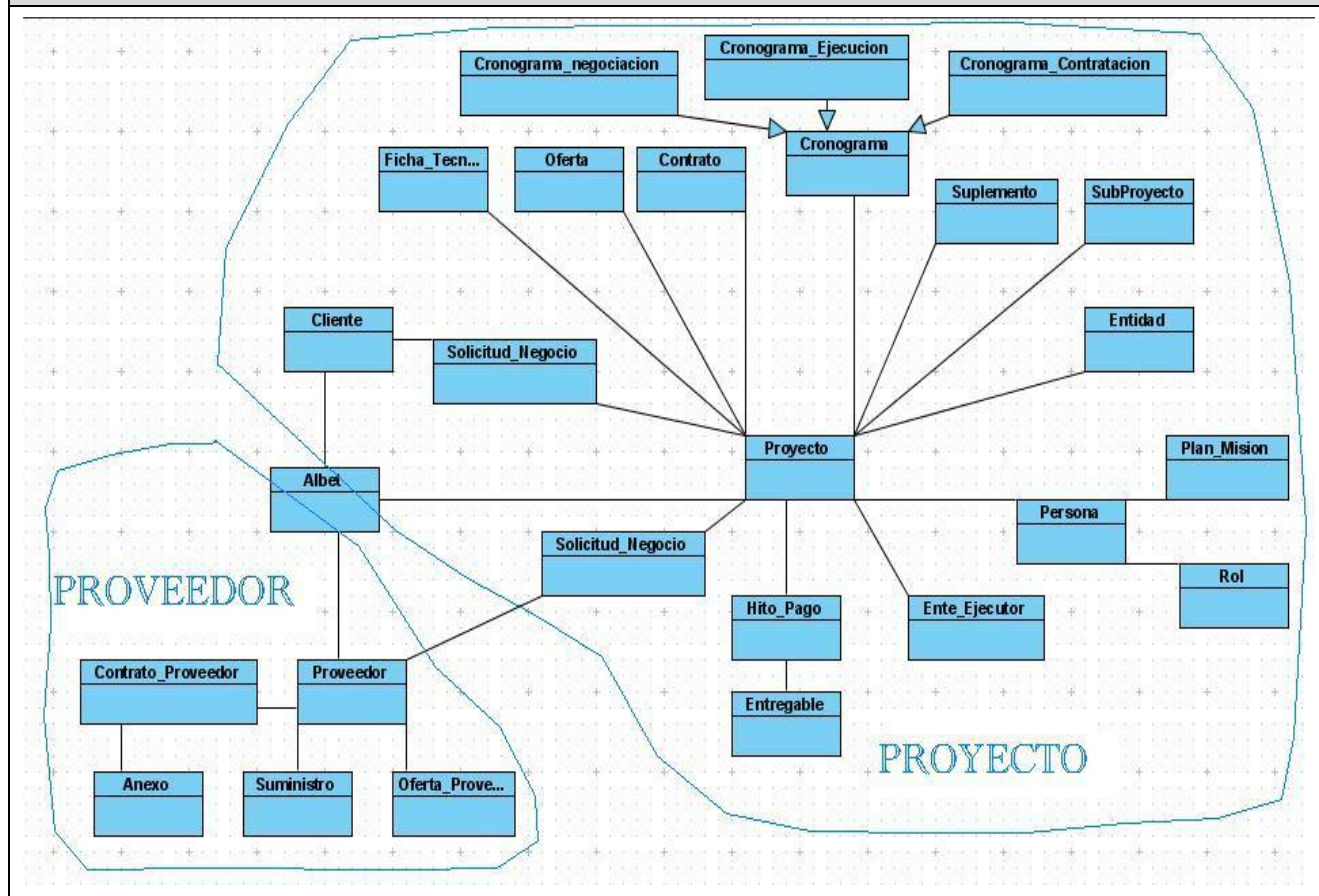
CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

ENTIDADES		
Solicitud de Proyecto	X	
Cliente	X	
Ficha Técnica	X	
Oferta	X	
Componente	X	
Contrato	X	
Anexo	X	
Suplemento	X	
Cronograma de ejecución	X	
Cronograma de contratación	X	
Cronograma de negociación	X	
Proyecto	X	
Subproyecto	X	
Entidad	X	
RH de proyecto	X	
Plan de Misión	X	
Hito de Pago	X	
Entregable	X	
Ente Ejecutor	X	
Solicitud de suministro		X
Proveedor		X
Oferta de Suministro		X

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Contrato		X
Anexo de Contrato		X
Suministro		X

Diagrama de los dominios de información



3.2.1 Ratio de cohesión.

RC= número de relaciones internas/número de asociaciones que cruzan

Dominio proveedores.

RC= 5/1

RC= 5

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Dominio proyecto

RC= 20/1

RC= 20

3.2.2 Ratio de acoplamiento.

RCM= número asociaciones que cruzan todos los dominios / número dominios

RA= 1/2

RA= 0.5

3.3 Identificación de los servicios.


3.3.1 Matriz CRU.

DOMINIO	PROYECTO					
PROCESOS	P1	P2	P3	P4	P4.1	P5
ENTIDADES						
Solicitud_Negocio	CRU					
Cliente	C					
Ficha Técnica		CRU				
Oferta			CRU			
Contrato				CRU		
Suplemento				CRU		
Cronograma ejecución				CRU		
Cronograma contratación				CRU		
Cronograma negociación	CU					

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Proyecto	C				U	
Subproyecto					C	
Entidad						
Persona					C	
Plan de Misión					C	
Rol					C	
Hito de pago					C	
Entregable					C	
Ente Ejecutor					C	
Solicitud de Suministro						C
DOMINIO	PROVEEDOR					
PROCESOS	P1	P2	P3	P4	P4.1	P5
ENTIDADES						
Proveedor						C
Oferta de Suministro						CRU
Contrato de Proveedor						CU
Suministro						C
Anexo de contrato						CU

3.3.2 Descripción de los servicios de negocios.

 <p style="text-align: center;">Equipo de desarrollo SOA GRUPO DE SOLUCIONES Y PROCESOS</p>	Fecha:	Elaborado por:
	Responsable:	
	Cargo:	

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Servicio	Cliente_Proyecto
Método	Bool Adicionar_Cliente_Proyecto(String nombre_e, String nombre_c, String apellidos, String pais_c, Integer telefono_c, String correo, String cargo)
Descripción	Adiciona un cliente del proyecto a la base de datos con los atributos nombre, apellidos, pais_c, telefono_c, correo y cargo que desempeña.
Método	Bool Eliminar_Cliente_Proyecto(Integer id)
Descripción	Elimina el cliente que corresponde al número de id que se le introduzca.
Método	Bool Actualizar_Cliente(Integer telefono_c, String correo, String cargo, Integer id)
Descripción	Actualiza el telefono_c, correo y cargo de un cliente en la base de datos según el número de id que se le introduzca.
Método	array Obtener_Todos_Cliente()
Descripción	Obtiene en un arreglo todos los clientes que tiene en ese momento Albet en su base de datos.
Método	array Obtene_Cliente_Dado_Pais(String pais)
Descripción	Obtiene en un arreglo todos los clientes que tiene en ese momento Albet en su base de datos que pertenezcan al país que se le introduzca.
Método	array Obtene_Cliente_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el cliente que tiene en ese momento Albet en su base de datos que se corresponda con el número de id que se le introduzca.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Servicio	Solicitud_Negocio
Método	Bool Insertar_Solicitud_Negocio(Integer cod_c, Date fecha_creacion, Date fecha_entrega)
Descripción	Inserta una solicitud de negocio en la base de datos con los atributos, codigo_Cliente (código que tiene el cliente que hace la solicitud en la base de datos), fecha_creacion y fecha_entrega de la misma.
Método	Bool Eliminar_Solicitud_Negocio(Integer id)
Descripción	Elimina la solicitud de negocio que se corresponda con el número de id que se le introduzca.
Método	Bool Actualizar_Solicitud(Date fecha, Integer id)
Descripción	Se actualiza la fecha de entrega de la solicitud de negocio que se corresponda con el número de id que se le introduzca.
Método	Bool Aprobar_Solicitud_Negocio(Integer id)
Descripción	El atributo aprobada pasa a verdadero en la solicitud de negocio que se corresponda con el número de id que se le introduzca.
Método	Bool Asignar_Responsable_Solicitud(String responsable, Integer id)
Descripción	El atributo responsable de la solicitud de negocio pasa a ser el que se le suministra en este método, en la solicitud de negocio que se le corresponda el número de id que se le introduzca.
Método	array Obtene_Solicitudes()

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Descripción	Obtiene en un arreglo todas las solicitudes de negocio que tiene en ese momento Albet en su base de datos.
Método	array Obtene_Solicitudes_Dado_Id(integer codigo_s)
Descripción	Obtiene en un arreglo la solicitud de negocio que se corresponda con el número de id que se le introduzca.
Método	array Obtene_Solicitudes_Dado_IdCliente(Integer codigo_s)
Descripción	Obtiene en un arreglo las solicitudes de negocio del cliente que se corresponda con el número de código que se le introduzca.
Método	array Obtene_Solicitudes_Dado_Responsable(String resp)
Descripción	Obtiene en un arreglo las solicitudes de negocio del responsable que se corresponda con el responsable que se le introduzca.
Servicio	Proyecto
Método	Bool Insertar_Proyecto(String nombre_p, Integer codigo_s , Float importe_s, String estado, String linea_t, Date fecha_i, Float monto_e_c, Float monto_transf, String alcance, String marco_a, Float fondo_pago, Date fecha_fin)
Descripción	Inserta un proyecto en la base de datos con los atributos, nombre del proyecto, código de la solicitud, importe, estado, línea de trabajo, fecha de inicio, monto de creación, monto de transferencia, alcance, marco de aprobación, fondo de pago y fecha de fin.
Método	Bool Eliminar_Proyecto(String nombre_p)

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Descripción	Elimina el proyecto que corresponde al nombre del proyecto que se le introduzca.
Método	Bool Actualizar_Proyecto(String nombre_p, Float importe_s, String estado, String facultad, Float monto_e_c, Float monto_transf, String alcance, String marco_a, Float fondo_pago, Date fecha_fin)
Descripción	Actualiza el nombre, importe, estado, facultad, monto de creación, monto de transferencia, alcance, marco de aprobación, fondo de pago y fecha de fin de el proyecto que le corresponda el número de id que se le introduzca.
Método	array Obtener_Proyectos()
Descripción	Obtiene en un arreglo todos los proyectos que tiene Albet en ese momento en su base de datos.
Método	array Obtener_Proyectos_Dado_IdSolicitud(Integer id)
Descripción	Obtiene en un arreglo todos los proyectos que tiene Albet en ese momento en su base de datos que se correspondan con el número de id de la solicitud que se le introduzca.
Método	array Obtener_Proyectos_Dado_Estado(String estado)
Descripción	Obtiene en un arreglo todos los proyectos que tiene Albet en ese momento en su base de datos que se correspondan con el estado que se le introduzca.
Método	array Obtener_Proyectos_Dado_linea_Trabajo(String linea_p)
Descripción	Obtiene en un arreglo todos los proyectos que tiene Albet en ese momento en su base de datos que se correspondan con la línea de trabajo que se le introduzca.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	array Obtener_Proyectos_Dado_facultad(String fac)
Descripción	Obtiene en un arreglo todos los proyectos que tiene Albet en ese momento en su base de datos que se correspondan con la facultad que se le introduzca.
Servicio	Ficha_Técnica
Método	Bool Insertar_Ficha_Tecnica(String nombre_p, Date fecha_c, Float version_f)
Descripción	Inserta una ficha técnica en la base de datos con los atributos, nombre del proyecto, fecha de creación y la versión de la ficha.
Método	Bool Eliminar_Ficha_Tecnica(Integer id)
Descripción	Elimina la ficha técnica que corresponde con el número de id que se le introduzca.
Método	Bool Actualizar_Version_Ficha_Tecnica(Float version_f, Integer id)
Descripción	Actualiza la versión de la ficha técnica que le corresponda el número de id que se le introduzca.
Método	Bool Aprobar_Ficha_Tecnica(Integer id)
Descripción	El atributo aprobada pasa a verdadero en la ficha técnica que se corresponda con el número de id que se le introduzca.
Método	array Obtener_Ficha_Tecnica()
Descripción	Obtiene en un arreglo todas las fichas técnicas que tiene Albet en ese momento en su base de datos.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	array Obtener_Ficha_Tecnica_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo la ficha técnica que se corresponda con el número de id que se le introduzca.
Método	array Obtener_Ficha_Tecnica_Dado_Nombre_Proyecto(String nombre)
Descripción	Obtiene en un arreglo la ficha técnica que se corresponda con el nombre del proyecto que se le introduzca.
Servicio	Contrato_Proyecto
Método	Bool Insertar_Contrato_Proyecto(String nombre_p, Date fecha_c, Float version_c)
Descripción	Inserta un contrato de un proyecto en la base de datos con los atributos, nombre del proyecto, fecha de creación y la version_c del contrato.
Método	Bool Eliminar_Contrato_Proyecto(Integer id)
Descripción	Elimina el contrato del proyecto que corresponde con el número de id que se le introduzca.
Método	Bool Actualizar_Version_Contrato_Proyecto(Float version, Integer id)
Descripción	Actualiza la versión del contrato del proyecto que le corresponda el número de id que se le introduzca.
Método	Bool Aprobar_Contrato_Proyecto(Integer id)
Descripción	El atributo aprobada pasa a verdadero en el contrato de proyecto que se corresponda con el número de id que se le introduzca.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	array Obtener_Contrato_Proyecto()
Descripción	Obtiene en un arreglo todos los contratos de proyectos que tiene Albet en ese momento en su base de datos.
Método	array Obtener_Contrato_Proyecto_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el contrato de proyecto que se corresponda con el número de id que se le introduzca.
Método	array Obtener_Contrato_Proyecto_Dado_Nombre_Proyecto(String nombre)
Descripción	Obtiene en un arreglo el contrato de proyecto que se corresponda con el nombre del proyecto que se le introduzca.
Servicio	Oferta_Proyecto
Método	Bool Insertar_Oferta(String nombre_p, Date fecha_c, Float version, String descripcion_o)
Descripción	Inserta una oferta en la base de datos con los atributos, nombre del proyecto, fecha de creación, la version de la oferta y la descripción.
Método	Bool Eliminar_Oferta(Integer codigo_o)
Descripción	Elimina la oferta que corresponde con el número del código que se le introduzca.
Método	Bool Actualizar_Version_Oferta(Float version, String descripción_o, Integer id)
Descripción	Actualiza la versión y la descripción de la oferta que le corresponda el número de id que se le introduzca.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Bool Aprobar_Oferta(integer id)
Descripción	El atributo aprobada pasa a verdadero en la oferta que se corresponda con el número de id que se le introduzca.
Método	Array Obtener_Oferta_Proyecto_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo la oferta que se corresponda con el número de id que se le introduzca.
Método	Array Obtener_Oferta_Proyecto_Dado_Nombre_Proyecto(String nombre)
Descripción	Obtiene en un arreglo la oferta que se corresponda con el nombre del proyecto que se le introduzca.
Servicio	Suplemento
Método	Bool Insertar_Suplemento_Proyecto(String nombre_p, Float version, Date fecha_c)
Descripción	Inserta un suplemento de proyecto en la base de datos con los atributos, nombre del proyecto, fecha de creación y la version del suplemento.
Método	Bool Eliminar_Suplemento_Proyecto(Integer id)
Descripción	Elimina el suplemento de proyecto que corresponde con el número de id que se le introduzca.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Bool Actualizar_Suplemento_Proyecto(Float version, Integer id)
Descripción	Actualiza la versión del suplemento de proyecto que le corresponda el número de id que se le introduzca.
Método	Bool Aprobar_Suplemento_Proyecto(Integer id)
Descripción	El atributo aprobada pasa a verdadero en el suplemento de proyecto que se corresponda con el número de id que se le introduzca.
Método	Array Obtener_Suplemento_Proyecto()
Descripción	Obtiene en un arreglo todos los suplementos de proyectos que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Suplemento_Proyecto_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el suplemento de proyecto que se corresponda con el número de id que se le introduzca.
Método	Array Obtener_Suplemento_Proyecto_Dado_Nombre_Proyecto(String nombre)
Descripción	Obtiene en un arreglo el suplemento de proyecto que se corresponda con el nombre del proyecto que se le introduzca.
Servicio	Cronograma_Negociacion
Método	Bool Insertar_Cronograma_Negociacion(Integer codigo_n, String nombre_p, Date fecha_c, String negociacion_c, Date fecha_neg, String responsable)
Descripción	Inserta un cronograma de negociación en la base de datos con los atributos, nombre del proyecto, código del cronograma, fecha de creación, negociación,

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

	fecha de negociación y responsable.
Método	Bool Aprobar_Cronograma_Negociacion(Integer codigo)
Descripción	El atributo aprobada pasa a verdadero en el cronograma de negociación que se corresponda con el número del codigo que se le introduzca.
Método	Bool Actualizar_Cronograma_Negociacion(Integer codigo, Date fecha_c, String negociacion_c, Date fecha_neg, String responsable)
Descripción	Actualiza la fecha de creación, la negociación, la fecha de negociación y el responsable del cronograma de negociación que le corresponda el número del codigo que se le introduzca.
Método	Bool Eliminar_Cronograma_Negociacion(Integer codigo)
Descripción	Elimina el cronograma de negociación que le corresponda el número del codigo que se le introduzca.
Método	Array Obtener_Cronograma_Negociacion()
Descripción	Obtiene en un arreglo todos los cronogramas de negociación que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Cronograma_Negociacion_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el cronograma de negociación que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Cronograma_Negociacion_Dado_Nombre_Proyecto(String nombre)

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Descripción	Obtiene en un arreglo el cronograma de negociación que le corresponda el nombre del proyecto que se le introduzca.
Servicio	Cronograma_Ejecucion
Método	Bool Insertar_Cronograma_Ejecucion(Integer codigo, String nombre_p, Date fecha_c, String tarea, Date fecha_i, Date fecha_f, String responsable)
Descripción	Inserta un cronograma de ejecución en la base de datos con los atributos, nombre del proyecto, fecha de creación, tarea, fecha de inicio, fecha de fin y responsable.
Método	Bool Aprobar_Cronograma_Ejecucion(Integer codigo)
Descripción	El atributo aprobada pasa a verdadero en el cronograma de ejecución que se corresponda con el número del código que se le introduzca.
Método	Bool Actualizar_Cronograma_Ejecucion(Integer codigo, Date fecha_c, String tarea, Date fecha_i, Date fecha_f, String responsable)
Descripción	Actualiza la fecha de creación, la tarea, fecha de inicio, fecha de fin y el responsable del cronograma de negociación que le corresponda el número del código que se le introduzca.
Método	Bool Eliminar_Cronograma_Ejecucion(Integer codigo)
Descripción	Elimina el cronograma de ejecución que le corresponda el número del código que se le introduzca.
Método	Array Obtener_Cronograma_Ejecucion()
Descripción	Obtiene en un arreglo todos los cronogramas de ejecución que tiene en ese momento Albet en su base de datos.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Array Obtener_Cronograma_Ejecucion_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el cronograma de ejecución que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Cronograma_Ejecucion_Dado_Nombre_Proyecto(String nombre)
Descripción	Obtiene en un arreglo el cronograma de ejecución que le corresponda el nombre del proyecto que se le introduzca.
Servicio	Cronograma_Contratacion
Método	Bool Insertar_Cronograma_Contratacion(Integer codigo, String nombre_p, Date fecha_c, String contratación_c, Date fecha_cont, String responsable)
Descripción	Inserta un cronograma de contratación en la base de datos con los atributos, código, nombre del proyecto, fecha de creación, contratación, fecha de contratación y responsable.
Método	Bool Aprobar_Cronograma_Contratacion(Integer codigo)
Descripción	El atributo aprobado pasa a verdadero en el cronograma de contratación que se corresponda con el número del código que se le introduzca.
Método	Bool Actualizar_Cronograma_Contratacion(Integer codigo, Date fecha_c, String contratación_c, Date fecha_cont, String responsable)
Descripción	Actualiza la fecha de creación, fecha de contratación, contratación y el responsable del cronograma de negociación que le corresponda el número del código que se le introduzca.
Método	Bool Eliminar_Cronograma_Contratacion(Integer codigo)

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Descripción	Elimina el cronograma de contratación que le corresponda el número del código que se le introduzca.
Método	Array Obtener_Cronograma_Contratacion()
Descripción	Obtiene en un arreglo todos los cronogramas de contratación que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Cronograma_Contratacion_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el cronograma de contratación que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Cronograma_Contratacion_Dado_Nombre_Proyecto(String nombre)
Descripción	Obtiene en un arreglo el cronograma de contratación que le corresponda el nombre del proyecto que se le introduzca.
Servicio	SubProyecto
Método	Bool Insertar_Subproyecto(String nombre_p, String nombre_s, String facultad, String alcance)
Descripción	Inserta un subproyecto en la base de datos con los atributos, nombre del proyecto, nombre del subproyecto, facultad y alcance.
Método	Bool Eliminar_Subproyecto(Integer id)
Descripción	Elimina el subproyecto que le corresponda el número de id que se le introduzca.
Método	Bool Actualizar_Subproyecto(String nombre, String facultad, String alcance, Integer id)

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Descripción	Actualiza el nombre del subproyecto, facultad y del subproyecto que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Subproyecto()
Descripción	Obtiene en un arreglo todos los subproyectos que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Subproyecto_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el subproyecto que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Subproyecto_Dado_IdProyecto(Integer proy_s)
Descripción	Obtiene en un arreglo los subproyectos que le corresponda el número de id del proyecto que se le introduzca.
Método	Array Obtener_Subproyecto_Dado_faculta(String fac)
Descripción	Obtiene en un arreglo los subproyectos que le corresponda la facultad que se le introduzca.
Servicio	Entidad
Método	Bool Insertar_Entidad(String nombre_e, String nombre_p)
Descripción	Inserta una entidad en la base de datos con los atributos, nombre de la entidad y nombre del proyecto.
Método	Bool Eliminar_Entidad(Integer id)

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Descripción	Elimina la entidad que le corresponda el número de id que se le introduzca.
Método	Bool Actualizar_Entidad(String nombre, Integer id)
Descripción	Actualiza el nombre de la entidad que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Entidad()
Descripción	Obtiene en un arreglo todos las entidades que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Entidad_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo la entidad que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Entidad_Dado_IdProyecto(String proy_e)
Descripción	Obtiene en un arreglo las entidades que le corresponda el número de id del proyecto que se le introduzca.
Método	Array Obtener_Entidad_Dado_Nombre(String nombre)
Descripción	Obtiene en un arreglo la entidad que le corresponda el nombre que se le introduzca.
Servicio	Persona
Método	Bool Insertar_Persona(Integer n_solapin, String nombre_p, String nombre_pers, String titulo, Date fecha_i, Date fecha_f, String evaluacion, Integer carne, String nivel_e, String nivel_calif, String cargo, String

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

	conocimientos, String direccion, String roll, String area, String habilidades)
Descripción	Inserta una persona en la base de datos con los atributos, número de solapín, nombre del proyecto, nombre de la persona, título, fecha de inicio, fecha de fin, evaluación, carné de identidad, nivel de evaluación, nivel de calificación, cargo, conocimientos, dirección, rol, área y habilidades.
Método	Bool Eliminar_Persona(Integer n_solapin)
Descripción	Elimina la persona que le corresponda el número de solapín que se le introduzca.
Método	Actualizar_Persona(Date fecha_f, String evaluacion, String cargo, String rol_a, Integer n_solapin)
Descripción	Actualiza la fecha de fin, la evaluación, el cargo y el rol que desempeña de la persona que le corresponda el número de solapín que se le introduzca.
Método	Array Obtener_Persona()
Descripción	Obtiene en un arreglo todas las personas que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Persona_Dado_Solapin(Integer solapin_p)
Descripción	Obtiene en un arreglo la persona que le corresponda el número de solapín que se le introduzca.
Método	Array Obtener_Persona_Dado_IdProyecto(String proy)
Descripción	Obtiene en un arreglo las personas que le corresponda el nombre del proyecto que se le introduzca.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Array Obtener_Persona_Dado_Carne(Integer carne)
Descripción	Obtiene en un arreglo la persona que le corresponda el número de carnet de identidad que se le introduzca.
Método	Array Obtener_Persona_Dado_Rol(String rol)
Descripción	Obtiene en un arreglo las personas que le corresponda el rol que se le introduzca.
Servicio	Plan_De_Mision
Método	Bool Insertar_Plan_Mision(Integer n_solapin, Date fecha_s, String tarea, Date fecha_e)
Descripción	Inserta un plan de misión en la base de datos con los atributos, número de solapín de la persona, fecha de salida, tarea y fecha de entrada.
Método	Bool Eliminar_Plan_Mision(Integer Id)
Descripción	Elimina el plan de misión que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Plan_Mision()
Descripción	Obtiene en un arreglo todos los planes de misión que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Plan_Mision_Dado_SolapinPersona(Integer solapin)
Descripción	Obtiene en un arreglo el plan de misión que le corresponda el número de solapín que se le introduzca.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Servicio	Rol
Método	Bool Insertar_Rol(Integer n_solapin, String rol)
Descripción	Inserta un rol en la base de datos con los atributos, número de solapín de la persona y rol.
Método	Bool Eliminar_Rol(Integer numero)
Descripción	Elimina el rol que le corresponda el número que se le introduzca.
Método	Bool Obtener_Rol_Dado_SolapinPersona(Integer solapin)
Descripción	Obtiene en un arreglo el rol que le corresponda el número de solapín que se le introduzca.
Servicio	Ente_Ejecutor
Método	Bool Insertar_Ente_Ejecutor(Integer carne, String nombre_p, String nombre, String direccion, String cargo, String ministerio)
Descripción	Inserta un ente ejecutor en la base de datos con los atributos, carnet de identidad, nombre del proyecto, nombre de la persona, dirección, cargo y ministerio.
Método	Bool Actualizar_Ente_Ejecutor(Integer carne, String nombre, String direccion, String cargo, String ministerio)
Descripción	Actualiza el nombre de la persona, dirección, cargo y ministerio del ente ejecutor que le corresponda el número de carnet de identidad que se le introduzca.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Bool Eliminar_Ente_Ejecutor(Integer carne)
Descripción	Elimina el ente ejecutor que le corresponda el número de carnet de identidad que se le introduzca.
Método	Array Obtener_Ente_Ejecutor()
Descripción	Obtiene en un arreglo todos los entes ejecutores que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Ente_Ejecutor_Dado_Carne(Integer carne)
Descripción	Obtiene en un arreglo el ente ejecutor que le corresponda el número de carnet de identidad que se le introduzca.
Método	Array Obtener_Ente_Ejecutor_Dado_IdProyecto(String proy)
Descripción	Obtiene en un arreglo el ente ejecutor que le corresponda el proyecto que se le introduzca.
Servicio	Hito_De_Pago
Método	Bool Insertar_Hito_Pago(Integer numero, String nombre_p, Date fecha_e, String estado)
Descripción	Inserta un hito de pago en la base de datos con los atributos, número, nombre del proyecto, fecha de entrada y estado.
Método	Bool Actualizar_Hito_Pago(Integer numero, Date fecha_e, String estado)
Descripción	Actualiza la fecha de entrada y estado del hito de pago que le corresponda el número que se le introduzca.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Bool Eliminar_Hito_Pago(Integer numero)
Descripción	Elimina el hito de pago que le corresponda el número que se le introduzca.
Método	Array Obtener_Hito_Pago()
Descripción	Obtiene en un arreglo todos los hitos de pago que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Hito_Pago_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el hito de pago que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Hito_Pago_Dado_IdProyecto(String proy)
Descripción	Obtiene en un arreglo el hito de pago que le corresponda el proyecto que se le introduzca.
Método	Obtener_Hito_Pago_Dado_Estado(String estado)
Descripción	Obtiene en un arreglo el hito de pago que le corresponda el estado que se le introduzca.
Servicio	Entregable
Método	Bool Insertar_Entregable(Integer n_hito, Float monto, Date fecha_e, String estado, Date fecha_a)
Descripción	Inserta un entregable en la base de datos con los atributos, número del hito, monto, fecha de entrada, estado y -----.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Bool Actualizar_Entregable(Integer numero, Float monto, Date fecha_e, String estado, Date fecha_a)
Descripción	Actualiza el monto, fecha de entrada, el estado y ----- del entregable que le corresponda el número que se le introduzca.
Método	Bool Eliminar_Entregable(Integer numero)
Descripción	Elimina el entregable que le corresponda el número que se le introduzca.
Método	Array Obtener_Entregable()
Descripción	Obtiene en un arreglo todos los entregables que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Entregable_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el entregable que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Entregable_Dado_IdHito(Integer idhito)
Descripción	Obtiene en un arreglo el entregable que le corresponda el número de id del hito que se le introduzca.
Método	Array Obtener_Entregable_Dado_Estado(String estado)
Descripción	Obtiene en un arreglo el entregable que le corresponda el estado se le introduzca.
Servicio	Solicitud_Suministro

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Bool Insertar_Solicitud_Suministro(String nombre_p, String quien_e, String quien_r, Date fecha_r, Date fecha_e)
Descripción	Inserta una solicitud de suministro en la base de datos con los atributos, nombre del proyecto, nombre de quien la envía, nombre de quien la recibe, fecha de envío y fecha de recibo.
Método	Bool Actualizar_Solicitud_Suministro(Integer codigo_s, String quien_e, String quien_r, Date fecha_r, Date fecha_e)
Descripción	Actualiza el nombre de quien la envía, nombre de quien la recibe, fecha de envío y fecha de quien la recibe de la solicitud de suministro que le corresponda el número de código que se le introduzca.
Método	Bool Actualizar_Proveedor_Solicitud_Suministro(Integer codigo_s, Integer prov)
Descripción	Actualiza el nombre del proveedor de la solicitud de suministro que le corresponda el número de código que se le introduzca.
Método	Bool Eliminar_Solicitud_Suministro(Integer codigo_s)
Descripción	Elimina la solicitud de suministro que le corresponda el número de código que se le introduzca.
Método	Array Obtener_Solicitud_Suministro()
Descripción	Obtiene en un arreglo todas las solicitudes de suministro que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Solicitud_Suministro_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo la solicitud de suministro que le corresponda el número de id que se le introduzca.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Array Obtener_Solicitud_Suministro_Dado_Nombre_Proyecto(String nombre)
Descripción	Obtiene en un arreglo la solicitud de suministro que le corresponda el nombre que se le introduzca.
Método	Array Obtener_Solicitud_Suministro_Dado_IdProveedor(Integer id)
Descripción	Obtiene en un arreglo las solicitudes de suministro que le corresponda el número de id del proveedor que se le introduzca.
Servicio	Proveedor
Método	Bool Insertar_Proveedor(Integer codigo, String nombre_e, String nombre_prov, String direccion, String nombre_cont, String correo, Integer telef, String nacionalidad, String t_sumin, Date tiempo_r)
Descripción	Inserta un proveedor en la base de datos con los atributos, código, nombre de la empresa, nombre del proveedor, dirección, correo, teléfono, nacionalidad.
Método	Bool Eliminar_Proveedor(Integer id)
Descripción	Elimina un proveedor que le corresponda el número de id que se le introduzca.
Método	Bool Actualizar_Proveedor(Integer codigo, String nombre_prov, String direccion, String nombre_cont, String correo, Integer telef, String nacionalidad, String t_sumin, Date tiempo_r)
Descripción	Actualiza el nombre del proveedor, dirección, correo, teléfono, nacionalidad de el proveedor que le corresponda el número de código que se le introduzca.

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Array Obtener_Proveedores()
Descripción	Obtiene en un arreglo todos los proveedores que tiene en ese momento Albet en su base de datos.
Método	Array Obtene_Proveedores_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el proveedor que le corresponda el número de id que se le introduzca.
Método	Array Obtene_Proveedores_Dado_Nombre(String nombre)
Descripción	Obtiene en un arreglo el proveedor que le corresponda el nombre que se le introduzca.
Método	Array Obtener_Proveedores_Dado_Tipo_Suministro(String sum)
Descripción	Obtiene en un arreglo el proveedor que le corresponda el suministro que se le introduzca.
Servicio	Oferta_Proveedor
Método	Bool Insertar_Oferta_Proveedor(Integer codigo_prov, String descrip, Integer cantidad, Date tiempo_g, Date fecha_r, Date plazo_e, Date t_validez, Float importe_total)
Descripción	Inserta una oferta del proveedor en la base de datos con los atributos, codigo del proveedor, descripción, cantidad, tiempo de garantía, fecha de recibo, plazo de envío, tiempo de validez e importe total.
Método	Bool Eliminar_Oferta_Proveedor(Integer codigo_p)

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Descripción	Elimina una oferta de proveedor, que le corresponda el número de código que se le introduzca.
Método	Bool Actualizar_Oferta_Proveedor(Integer codigo, String descrip, Integer cantidad, Date tiempo_g, Date fecha_r, Date plazo_e, Date t_validez, Float importe_total)
Descripción	Actualiza la descripción, cantidad, tiempo de garantía, fecha de recibo, plazo de entrega, tiempo de validez e importe total de la oferta que le corresponda el número de código que se le introduzca.
Método	Bool Aprobar_Oferta_Proveedor(Integer codigo)
Descripción	El atributo aprobado pasa a verdadero en oferta del proveedor que se corresponda con el número del código que se le introduzca.
Método	Array Obtener_Oferta_Proveedor()
Descripción	Obtiene en un arreglo todas las ofertas de proveedores que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Oferta_Proveedor_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo las ofertas que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Oferta_Proveedor_Dado_IdProveedor(Integer id)
Descripción	Obtiene en un arreglo las ofertas que le corresponda el número de id del proveedor que se le introduzca.
Servicio	Contrato_Proveedor

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Bool Insertar_Contrato_Proveedor(Integer codigo_prov, Date fecha_r, String obligaciones, String ficha)
Descripción	Inserta un contrato con el proveedor en la base de datos con los atributos, codigo del proveedor, fecha de recibo, obligaciones y ficha.
Método	Bool Actualizar_Contrato_Proveedor(Integer codigo_cont, Date fecha_r, String obligaciones, Date ficha)
Descripción	Actualiza la fecha de recibo, las obligaciones y la ficha del contrato con el proveedor que le corresponda el número de código que se le introduzca.
Método	Bool Eliminar_Contrato_Proveedor(Integer codigo_cont)
Descripción	Elimina un contrato con el proveedor, que le corresponda el número de código que se le introduzca.
Método	Array Obtener_Contrato_Proveedor()
Descripción	Obtiene en un arreglo todos los contratos de los proveedores que tiene en ese momento Albet en su base de datos.
Método	Array Obtene_Contrato_Proveedores_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el contrato de proveedores que le corresponda el número de id que se le introduzca.
Método	Array Obtene_Contrato_Proveedores_Dado_IdProveedor(Integer id)
Descripción	Obtiene en un arreglo los contratos de proveedores que le corresponda el número de id del proveedor que se le introduzca.
Servicio	Anexo_Contrato

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Bool Insertar_Anexo_Contrato(Integer codigo_cont, Float monto, String alcance, Date fecha_f)
Descripción	Inserta un anexo del contrato en la base de datos con los atributos, código del contrato, monto, alcance y fecha de fin.
Método	Bool Actualizar_Anexo_Contrato(Integer codigo, Float monto, String alcance, Date fecha_f)
Descripción	Actualiza el monto, el alcance y la fecha de fin del anexo del contrato que le corresponda el número de código que se le introduzca.
Método	Bool Eliminar_Anexo_Contrato(Integer codigo)
Descripción	Elimina un anexo del contrato, que le corresponda el número de código que se le introduzca.
Método	Array Obtener_Anexo_Contrato()
Descripción	Obtiene en un arreglo todos los anexos de contratos que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Anexo_Contrato_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el anexo del contrato que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Anexo_Contrato_Dado_IdContrato(Integer id)
Descripción	Obtiene en un arreglo los anexos del contratos que le corresponda el número de id del contrato que se le introduzca.
Servicio	Suministro


CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Método	Bool Insertar_Suministro(Integer cod_prov, String estado)
Descripción	Inserta un anexo del contrato en la base de datos con los atributos, código del proveedor y estado.
Método	Bool Eliminar_Suministro(Integer cod)
Descripción	Elimina un suministro que le corresponda el número de código que se le introduzca.
Método	Bool Actualizar_Suministro(String estado, Integer cod)
Descripción	Actualiza el estado del suministro que le corresponda el número de código que se le introduzca.
Método	Array Obtener_Suministro()
Descripción	Obtiene en un arreglo todos los suministros que tiene en ese momento Albet en su base de datos.
Método	Array Obtener_Suministro_Dado_Id(Integer id)
Descripción	Obtiene en un arreglo el suministro que le corresponda el número de id que se le introduzca.
Método	Array Obtener_Suministro_Dado_Idproveedor(Integer id)
Descripción	Obtiene en un arreglo los suministros que le corresponda el número de id del proveedor que se le introduzca.
Método	Array Obtener_Suministro_Dado_Estado(String estado)

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Descripción	Obtiene en un arreglo los suministros que le corresponda el estado que se le introduzca.
-------------	--

3.3.3 Matriz procesos contra servicios.

 Equipo de desarrollo SOA GRUPO DE SOLUCIONES Y PROCESOS		Fecha:		Elaborado por:		
Procesos	P1	P2	P3	P4	P4.1	P5
Servicios						
Cliente_Proyecto	X					
Solicitud_Negocio	X					
Proyecto	X					
Ficha_Técnica		X				
Contrato_Proyecto				X		
Oferta_Proyecto			X			
Suplemento				X		
Cronograma_Negociacion	X					
Cronograma_Ejecucion				X		
Cronograma_Contratacion				X		
SubProyecto					X	
Entidad					X	
Persona					X	

CAPÍTULO 3: DISEÑO DE LOS SERVICIOS

Plan_De_Mision					X	
Rol					X	
Ente_Ejecutor					X	
Hito_De_Pago					X	
Entregable					X	
Solicitud_Suministro						X
Proveedor						X
Oferta_Proveedor						X
Contrato_Proveedor						X
Anexo_Contrato						X
Suministro						X

3.4 Conclusiones.

En este capítulo tras un estudio exhaustivo de la empresa se identificaron los diferentes dominios de información como primer paso para el diseño de los diferentes servicios web, estos servirán de soporte en el desarrollo de la Arquitectura Orientada a Servicios en los procesos de gestión de proyecto de ALBET. Se descubrieron un total de 26 servicios web; se hizo una descripción de cada unos de estos, así como de sus funcionalidades, los cuales cubren todos los procesos de negocios.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS.

4.1 Introducción

El presente capítulo constituye la secuencia lógica del diseño, a través del mismo se implementan las clases y subsistema en términos de componentes. Se obtendrán los diagramas de componentes y de despliegue como parte del modelo de implementación, en este último se indicará cómo la solución implementada estará distribuida físicamente. En él se recogerán los resultados de las pruebas de los principales servicios que van a ser utilizados en el despliegue de la SOA en los Procesos de Gestión de Proyectos de ALBET.

4.2 Modelo de implementación.

El modelo de implementación describe cómo los elementos serán implementados en términos de componentes describiendo además su organización de acuerdo con los mecanismos de estructuración y modularización disponibles en el entorno de implementación y el lenguaje utilizado así como las dependencias entre estos componentes.

4.2.1 Diagramas de componentes.

Un componente es el empaquetamiento físico de los elementos de un modelo como son las clases. A continuación serán expuestos los diagramas de componentes asociados a varios de los subsistemas de implementación que han sido reconocidos para la implementación de los servicios identificados en el capítulo anterior.

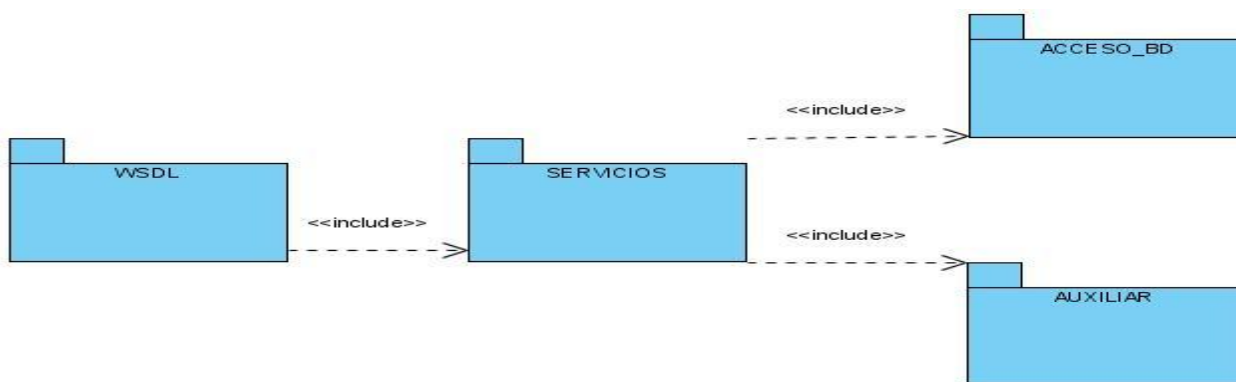


Figure 7 Subsistema de implementación.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

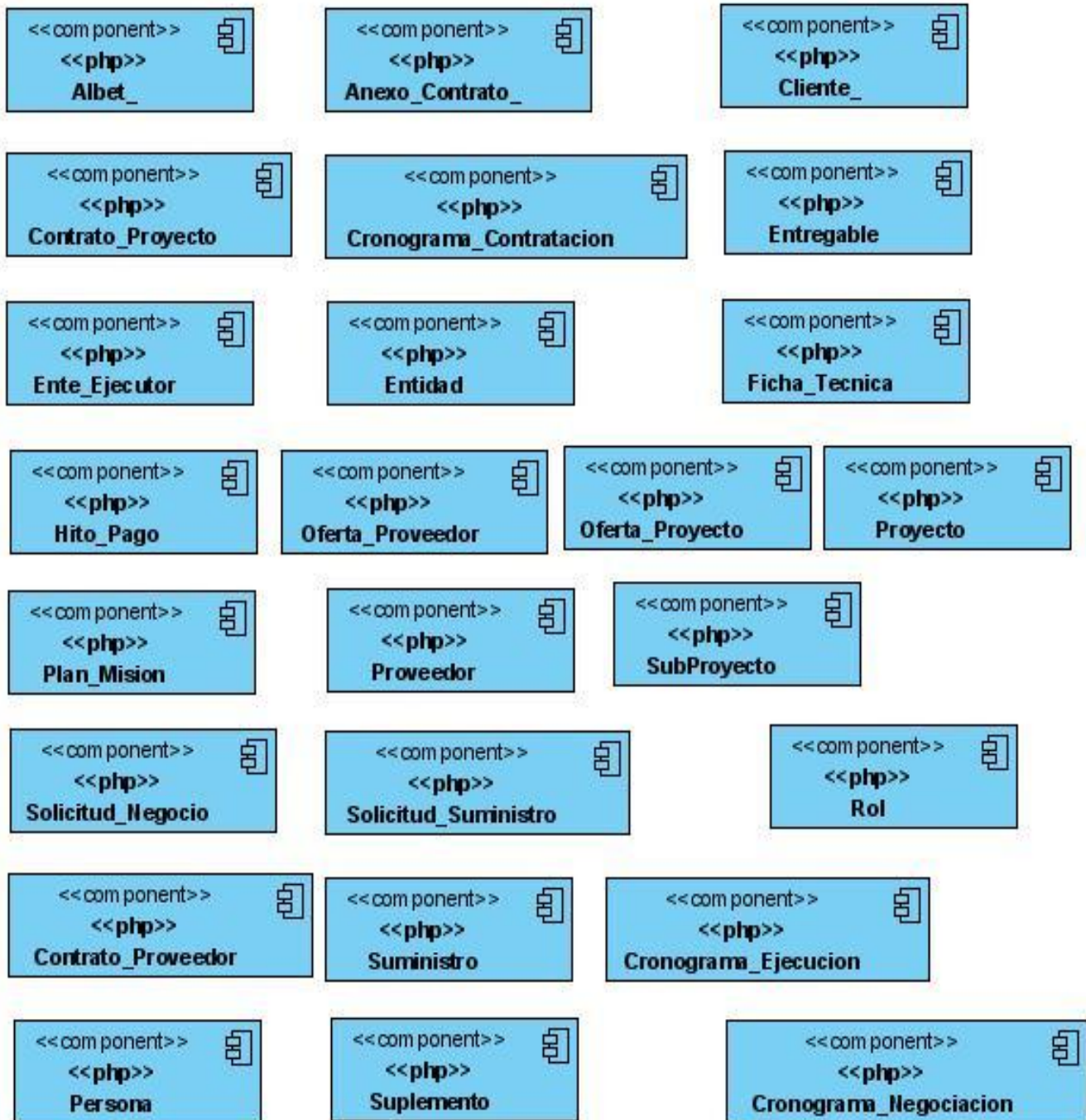


Figure 8 Subsistema de implementación. Servicios.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

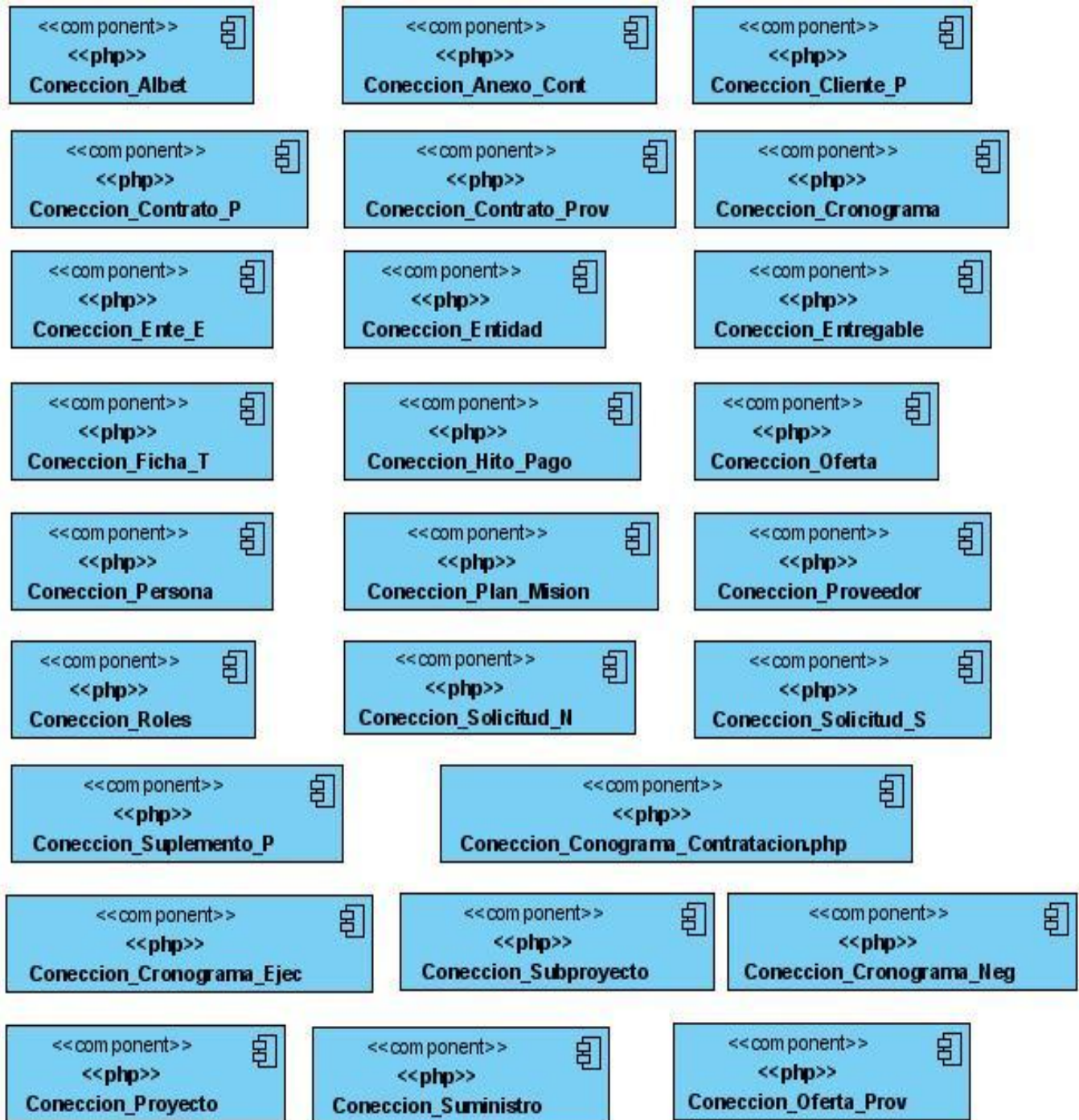


Figure 9 Subsistema de implementación. Acceso_bd

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

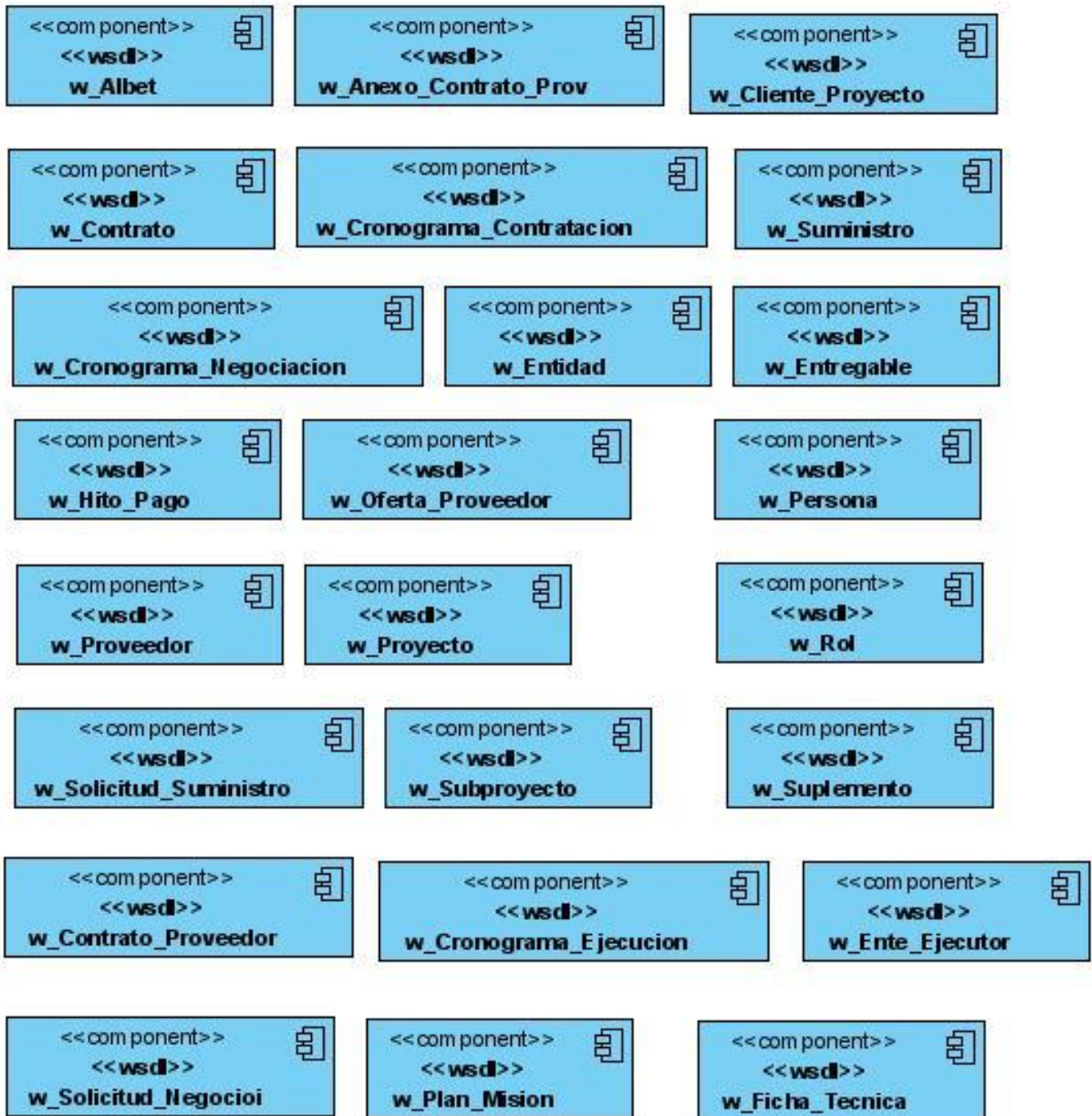


Figure 10 Subsistema de implementación. WSDL.

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

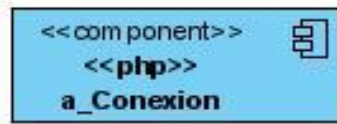


Figure 11 Subsistema de implementación. Auxiliar.

4.2.2 Diagrama de despliegue.

Un diagrama de despliegue es un modelo de objeto que describe la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo. Por esta razón tiene una gran influencia para las actividades de implementación. Cada nodo representa un recurso de cómputo, en el despliegue de la Arquitectura Orientada a Servicios en los procesos de gestión de proyectos de ALBET este será distribuido en tres servidores el primero que va a contener los procesos de negocios conectados a una PC cliente, el segundo contendrá toda la lógica del negocio el cual va a estar conectado al tercero, es decir, al servidor de datos.

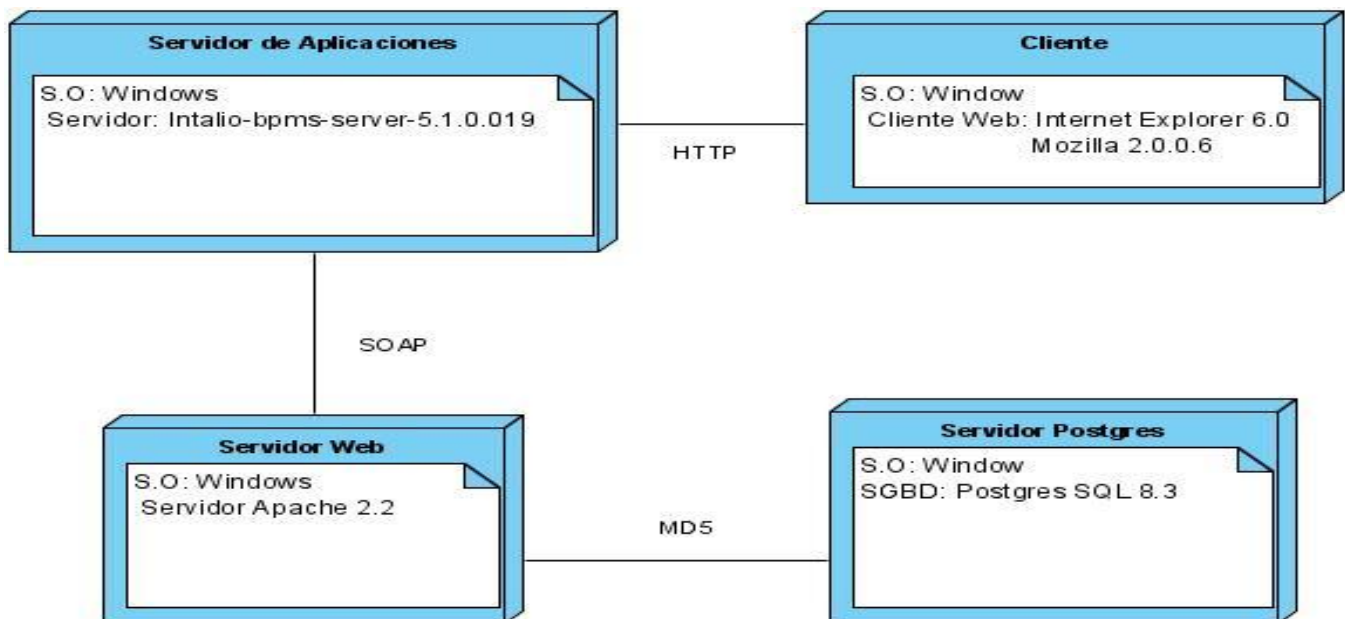


Figure 12 Diagrama de despliegue

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

4.3 Pruebas.

Servicio	SW_Albet
Función	Insertar_Datos_Empresa
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_albet"> <soapenv:Header/> <soapenv:Body> <urn:Insertar_Datos_Empresa soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <nombre xsi:type="xsd:string"> nombre </nombre> <mision xsi:type="xsd:string"> misión </mision> <descrip xsi:type="xsd:string"> descripción </descrip> <filosofia xsi:type="xsd:string"> Empresa1 </filosofia> </urn:Insertar_Datos_Empresa> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_albet" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Insertar_Datos_EmpresaResponse> <Insertar_Datos_EmpresaReturn xsi:type="xsd:integer">1</Insertar_Datos_EmpresaReturn> </ns1:Insertar_Datos_EmpresaResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	139 ms
Función	Eliminar_Empresa
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_albet"> <soapenv:Header/> <soapenv:Body> <urn:Eliminar_Empresa soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <nombre xsi:type="xsd:string">Albet</nombre> </urn:Eliminar_Empresa> </soapenv:Body> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	</soapenv:Envelope>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_albet" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Eliminar_EmpresaResponse> <Eliminar_EmpresaReturn xsi:type="xsd:integer">0</Eliminar_EmpresaReturn> </ns1:Eliminar_EmpresaResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	62 ms
Función	Actualizar_Albet
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_albet"> <soapenv:Header/> <soapenv:Body> <urn:Actualizar_Albet </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <mision xsi:type="xsd:string"> misión </mision> <descrip xsi:type="xsd:string"> descripción </descrip> <filos xsi:type="xsd:string"> filosofía </filos> <nombre_e xsi:type="xsd:string">Albet</nombre_e> </urn:Actualizar_Albet> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_albet" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Actualizar_AlbetResponse> <Actualizar_AlbetReturn xsi:type="xsd:integer">1</Actualizar_AlbetReturn> </ns1:Actualizar_AlbetResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	108 ms

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

Servicio	SW_Cliente_Proyecto
Función	Actualizar_Cliente
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_cliente_proyecto"> <soapenv:Header/> <soapenv:Body> <urn:Actualizar_Cliente soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <telefono xsi:type="xsd:integer">8375852</telefono> <correo xsi:type="xsd:string">yarde@estudiantes.uci.cu</correo> <cargo xsi:type="xsd:string">Administrador</cargo> <id xsi:type="xsd:integer">15</id> </urn:Actualizar_Cliente> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_cliente_proyecto" </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Actualizar_ClienteResponse> <Actualizar_ClienteReturn xsi:type="xsd:integer">1</Actualizar_ClienteReturn> </ns1:Actualizar_ClienteResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	1447 ms
Función	Adicionar_Cliente_Proyecto
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_cliente_proyecto"> <soapenv:Header/> <soapenv:Body> <urn:Adicionar_Cliente_Proyecto soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <nombre_e xsi:type="xsd:string">Albet</nombre_e> <nombre_c xsi:type="xsd:string">Israel </nombre_c> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <apellidos xsi:type="xsd:string">Siles Siles</apellidos> <pais xsi:type="xsd:string">Cuba</pais> <telefono xsi:type="xsd:integer">3126</telefono> <correo xsi:type="xsd:string">isiles@estudiantes.uci.cu</correo> <cargo xsi:type="xsd:string">Jefe Piso</cargo> </urn:Adicionar_Cliente_Proyecto> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_cliente_proyecto" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Adicionar_Cliente_ProyectoResponse> <Adicionar_Cliente_ProyectoReturn xsi:type="xsd:integer">1</Adicionar_Cliente_ProyectoReturn> </ns1:Adicionar_Cliente_ProyectoResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

TR	122 ms
Función	Eliminar_Cliente_Proyecto
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_cliente_proyecto"> <soapenv:Header/> <soapenv:Body> <urn:Eliminar_Cliente_Proyecto soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <id xsi:type="xsd:integer">22</id> </urn:Eliminar_Cliente_Proyecto> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_cliente_proyecto" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Eliminar_Cliente_ProyectoResponse> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <Eliminar_Cliente_ProyectoReturn xsi:type="xsd:integer">1</Eliminar_Cliente_ProyectoReturn> </ns1:Eliminar_Cliente_ProyectoResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	146 ms
Función	Obtene_Cliente_Dado_Id
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_cliente_proyecto"> <soapenv:Header/> <soapenv:Body> <urn:Obtene_Cliente_Dado_Id soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <id xsi:type="xsd:integer">1</id> </urn:Obtene_Cliente_Dado_Id> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre>xmlns:ns1="urn:wSDL_cliente_proyecto" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <SOAP-ENV:Body> <ns1:Obtene_Cliente_Dado_IdResponse> <Obtene_Cliente_Dado_IdReturn SOAP-ENC:arrayType="ns1:Cliente[1]" xsi:type="ns1:ClienteArray"> <item xsi:type="ns1:Cliente"> <apellidos xsi:type="xsd:string">de la Rosa Rodriguez</apellidos> <cargo xsi:type="xsd:string">Jefe Piso</cargo> <correo xsi:type="xsd:string">adellarosar@estudiantes.uci.cu</correo> <id xsi:type="xsd:integer">1</id> <nombre_c xsi:type="xsd:string">Argelio</nombre_c> <nombre_e xsi:type="xsd:string">Albet</nombre_e> <pais xsi:type="xsd:string">Cuba</pais> <telefono xsi:type="xsd:integer">3120</telefono> </item> </Obtene_Cliente_Dado_IdReturn> </ns1:Obtene_Cliente_Dado_IdResponse> </SOAP-ENV:Body></pre>
--	--

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	</SOAP-ENV:Envelope>
TR	100 ms
Función	Obtene_Cliente_Dado_Pais
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_cliente_proyecto"> <soapenv:Header/> <soapenv:Body> <urn:Obtene_Cliente_Dado_Pais soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <pais xsi:type="xsd:string">Cuba</pais> </urn:Obtene_Cliente_Dado_Pais> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_cliente_proyecto" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <SOAP-ENV:Body> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <ns1:Obtene_Cliente_Dado_PaisResponse> <Obtene_Cliente_Dado_PaisReturn SOAP-ENC:arrayType="ns1:Cliente[1]" xsi:type="ns1:ClienteArray"> <item xsi:type="ns1:Cliente"> <apellidos xsi:type="xsd:string">de la Rosa Rodriguez</apellidos> <cargo xsi:type="xsd:string">Jefe Piso</cargo> <correo xsi:type="xsd:string">adelarosar@estudiantes.uci.cu</correo> <id xsi:type="xsd:integer">1</id> <nombre_c xsi:type="xsd:string">Argelio</nombre_c> <nombre_e xsi:type="xsd:string">Albet</nombre_e> <pais xsi:type="xsd:string">Cuba</pais> <telefono xsi:type="xsd:integer">3120</telefono> </item> </Obtene_Cliente_Dado_PaisReturn> </ns1:Obtene_Cliente_Dado_PaisResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	70 ms
Función	Obtener_Todos_Cliente
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_cliente_proyecto"> <soapenv:Header/> <soapenv:Body> <urn:Obtener_Todos_Cliente soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding"/> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_cliente_proyecto" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <SOAP-ENV:Body> <ns1:Obtener_Todos_ClienteResponse> <Obtener_Todos_ClienteReturn SOAP-ENC:arrayType="ns1:Cliente[1]" xsi:type="ns1:ClienteArray"> <item xsi:type="ns1:Cliente"> <apellidos xsi:type="xsd:string">de la Rosa Rodríguez</apellidos> <cargo xsi:type="xsd:string">Jefe Piso</cargo> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <correo xsi:type="xsd:string">adelarosar@estudiantes.uci.cu</correo> <id xsi:type="xsd:integer">1</id> <nombre_c xsi:type="xsd:string">Argelio</nombre_c> <nombre_e xsi:type="xsd:string">Albet</nombre_e> <pais xsi:type="xsd:string">Cuba</pais> <telefono xsi:type="xsd:integer">3120</telefono> </item> </Obtener_Todos_ClienteReturn> </ns1:Obtener_Todos_ClienteResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	73 ms
Servicio	SW_Solicitud_Negocio
Función	Insertar_Solicitud_Negocio
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_solicitud_negocio"> <soapenv:Header/> <soapenv:Body> <urn:Insertar_Solicitud_Negocio </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <cod_c xsi:type="xsd:integer">1</cod_c> <fecha_creacion xsi:type="xsd:anyType">6/25/2009</fecha_creacion> <fecha_entrega xsi:type="xsd:anyType">8/25/2009</fecha_entrega> </urn:Insertar_Solicitud_Negocio> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_solicitud_negocio" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Insertar_Solicitud_NegocioResponse> <Insertar_Solicitud_NegocioReturn xsi:type="xsd:integer">1</Insertar_Solicitud_NegocioReturn> </ns1:Insertar_Solicitud_NegocioResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	163 ms

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

Función	Actualizar_Solicitud
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_solicitud_negocio"> <soapenv:Header/> <soapenv:Body> <urn:Actualizar_Solicitud soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <fecha xsi:type="xsd:anyType">9/25/2009</fecha> <id xsi:type="xsd:integer">15</id> </urn:Actualizar_Solicitud> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_solicitud_negocio" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Actualizar_SolicitudResponse> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <Actualizar_SolicitudReturn xsi:type="xsd:integer">1</Actualizar_SolicitudReturn> </ns1:Actualizar_SolicitudResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	133 ms
Función	Aprobar_Solicitud_Negocio
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_solicitud_negocio"> <soapenv:Header/> <soapenv:Body> <urn:Aprobar_Solicitud_Negocio soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <id xsi:type="xsd:integer">15</id> </urn:Aprobar_Solicitud_Negocio> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> xmlns:ns1="urn:wSDL_solicitud_negocio" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Aprobar_Solicitud_NegocioResponse> <Aprobar_Solicitud_NegocioReturn xsi:type="xsd:integer">1</Aprobar_Solicitud_NegocioReturn> </ns1:Aprobar_Solicitud_NegocioResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	115 ms
Función	Asignar_Responsable_Solicitud
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_solicitud_negocio"> <soapenv:Header/> <soapenv:Body> <urn:Asignar_Responsable_Solicitud soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <responsable xsi:type="xsd:string">Emiliano Blanco Medel</responsable> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <id xsi:type="xsd:integer">15</id> </urn:Asignar_Responsable_Solicitud> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_solicitud_negocio" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Asignar_Responsable_SolicitudResponse> <Asignar_Responsable_SolicitudReturn xsi:type="xsd:integer">1</Asignar_Responsable_SolicitudReturn> </ns1:Asignar_Responsable_SolicitudResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	117 ms
Función	Eliminar_Solicitud_Negocio
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> xmlns:urn="urn:wSDL_solicitud_negocio"> <soapenv:Header/> <soapenv:Body> <urn:Eliminar_Solicitud_Negocio soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <id xsi:type="xsd:integer">17</id> </urn:Eliminar_Solicitud_Negocio> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_solicitud_negocio" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Eliminar_Solicitud_NegocioResponse> <Eliminar_Solicitud_NegocioReturn xsi:type="xsd:integer">1</Eliminar_Solicitud_NegocioReturn> </ns1:Eliminar_Solicitud_NegocioResponse> </SOAP-ENV:Body> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	</SOAP-ENV:Envelope>	
TR	158 ms	
Función	Obtener_Solicitudes	
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_solicitud_negocio"> <soapenv:Header/> <soapenv:Body> <urn:Obtene_Solicitudes soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"/> </soapenv:Body> </soapenv:Envelope> </pre>	
Respuesta	<pre> <SOAP-ENV:Envelope ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_solicitud_negocio" ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <SOAP-ENV:Body> <ns1:Obtene_SolicitudesResponse> <Obtene_SolicitudesReturn SOAP-ENC:arrayType="ns1:SNegocio[2]" </pre>	<pre> SOAP- xmlns:SOAP- xmlns:SOAP- </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre>xsi:type="ns1:SNegocioArray"> <item xsi:type="ns1:SNegocio"> <aprob xsi:type="xsd:string">SI</aprob> <fech_c xsi:type="xsd:string">2009-06-25</fech_c> <fech_e xsi:type="xsd:string">2009-09-25</fech_e> <id_c xsi:type="xsd:integer">1</id_c> <id_s xsi:type="xsd:integer">15</id_s> <resp xsi:type="xsd:string">Emiliano Blanco Medel</resp> </item> <item xsi:type="ns1:SNegocio"> <aprob xsi:type="xsd:string">NO</aprob> <fech_c xsi:type="xsd:string">2009-01-25</fech_c> <fech_e xsi:type="xsd:string">2009-08-25</fech_e> <id_c xsi:type="xsd:integer">1</id_c> <id_s xsi:type="xsd:integer">16</id_s> <resp xsi:nil="true"/> </item> </Obtene_SolicitudesReturn> </ns1:Obtene_SolicitudesResponse></pre>
--	---

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	72 ms
Función	Obtener_Solicitudes_Dado_Id
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_solicitud_negocio"> <soapenv:Header/> <soapenv:Body> <urn:Obtene_Solicitudes_Dado_Id soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <codigo xsi:type="xsd:integer">15</codigo> </urn:Obtene_Solicitudes_Dado_Id> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_solicitud_negocio" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <SOAP-ENV:Body> <ns1:Obtene_Solicitudes_Dado_IdResponse> <Obtene_Solicitudes_Dado_IdReturn SOAP- ENC:arrayType="ns1:SNegocio[1]" xsi:type="ns1:SNegocioArray"> <item xsi:type="ns1:SNegocio"> <aprob xsi:type="xsd:string">SI</aprob> <fech_c xsi:type="xsd:string">2009-06-25</fech_c> <fech_e xsi:type="xsd:string">2009-09-25</fech_e> <id_c xsi:type="xsd:integer">1</id_c> <id_s xsi:type="xsd:integer">15</id_s> <resp xsi:type="xsd:string">Emiliano Blanco Medel</resp> </item> </Obtene_Solicitudes_Dado_IdReturn> </ns1:Obtene_Solicitudes_Dado_IdResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	75 ms
Función	Obtener_Solicitudes_Dado_IdCliente
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> xmlns:urn="urn:wSDL_solicitud_negocio"> <soapenv:Header/> <soapenv:Body> <urn:Obtene_Solicitudes_Dado_IdCliente soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <codigo xsi:type="xsd:integer">1</codigo> </urn:Obtene_Solicitudes_Dado_IdCliente> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_solicitud_negocio" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <SOAP-ENV:Body> <ns1:Obtene_Solicitudes_Dado_IdClienteResponse> <Obtene_Solicitudes_Dado_IdClienteReturn SOAP- ENC:arrayType="ns1:SNegocio[2]" xsi:type="ns1:SNegocioArray"> <item xsi:type="ns1:SNegocio"> <aprob xsi:type="xsd:string">SI</aprob> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre><feh_c xsi:type="xsd:string">2009-06-25</feh_c> <feh_e xsi:type="xsd:string">2009-09-25</feh_e> <id_c xsi:type="xsd:integer">1</id_c> <id_s xsi:type="xsd:integer">15</id_s> <resp xsi:type="xsd:string">Emiliano Blanco Medel</resp> </item> <item xsi:type="ns1:SNegocio"> <aprob xsi:type="xsd:string">NO</aprob> <feh_c xsi:type="xsd:string">2009-01-25</feh_c> <feh_e xsi:type="xsd:string">2009-08-25</feh_e> <id_c xsi:type="xsd:integer">1</id_c> <id_s xsi:type="xsd:integer">16</id_s> <resp xsi:nil="true"/> </item> </Obtene_Solicitudes_Dado_IdClienteReturn> </ns1:Obtene_Solicitudes_Dado_IdClienteResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>
TR	113 ms

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

Función	Obtene_Solicitudes_Dado_Responsable
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_solicitud_negocio"> <soapenv:Header/> <soapenv:Body> <urn:Obtene_Solicitudes_Dado_Responsable soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <resp xsi:type="xsd:string">Emiliano Blanco Medel</resp> </urn:Obtene_Solicitudes_Dado_Responsable> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_solicitud_negocio" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <SOAP-ENV:Body> <ns1:Obtene_Solicitudes_Dado_ResponsableResponse> <Obtene_Solicitudes_Dado_ResponsableReturn SOAP- </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> ENC:arrayType="ns1:SNegocio[1]" xsi:type="ns1:SNegocioArray"> <item xsi:type="ns1:SNegocio"> <aprob xsi:type="xsd:string">SI</aprob> <fech_c xsi:type="xsd:string">2009-06-25</fech_c> <fech_e xsi:type="xsd:string">2009-09-25</fech_e> <id_c xsi:type="xsd:integer">1</id_c> <id_s xsi:type="xsd:integer">15</id_s> <resp xsi:type="xsd:string">Emiliano Blanco Medel</resp> </item> </Obtene_Solicitudes_Dado_ResponsableReturn> </ns1:Obtene_Solicitudes_Dado_ResponsableResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	75 ms
Servicio	SW_Proyecto
Función	Insertar_Proyecto
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_proyecto"> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <soapenv:Header/> <soapenv:Body> <urn:Insertar_Proyecto soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <nombre_p xsi:type="xsd:string">Simulador</nombre_p> <codigo_s xsi:type="xsd:integer">15</codigo_s> <importe_s xsi:type="xsd:float">123456</importe_s> <estado xsi:type="xsd:string">negocio</estado> <linea_t xsi:type="xsd:string">RV</linea_t> <fecha_i xsi:type="xsd:anyType">6/25/2009</fecha_i> <monto_e_c xsi:type="xsd:float">123456</monto_e_c> <monto_transf xsi:type="xsd:float">123456</monto_transf> <alcance xsi:type="xsd:string">Validacion Cliente</alcance> <marco_a xsi:type="xsd:string">Marco</marco_a> <fondo_pago xsi:type="xsd:float">123456</fondo_pago> <fecha_fin xsi:type="xsd:anyType">11/25/2009</fecha_fin> </urn:Insertar_Proyecto> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wsdl_proyecto" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Insertar_ProyectoResponse> <Insertar_ProyectoReturn xsi:type="xsd:integer">1</Insertar_ProyectoReturn> </ns1:Insertar_ProyectoResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	575 ms
Función	Actualizar_Proyecto
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wsdl_proyecto"> <soapenv:Header/> <soapenv:Body> <urn:Actualizar_Proyecto soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <nombre_p xsi:type="xsd:string">Simulador</nombre_p> <importe_s xsi:type="xsd:float">123456789</importe_s> <estado xsi:type="xsd:string">Ejecución</estado> <facultad xsi:type="xsd:string">fac6</facultad> <monto_e_c xsi:type="xsd:float">123456</monto_e_c> <monto_transf xsi:type="xsd:float">123456</monto_transf> <alcance xsi:type="xsd:string">Pruebas</alcance> <marco_a xsi:type="xsd:string">marco</marco_a> <fondo_pago xsi:type="xsd:float">123258</fondo_pago> <fecha_fin xsi:type="xsd:anyType">1/25/2009</fecha_fin> </urn:Actualizar_Proyecto> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wsdl_proyecto" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Actualizar_ProyectoResponse> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <Actualizar_ProyectoReturn xsi:type="xsd:integer">1</Actualizar_ProyectoReturn> </ns1:Actualizar_ProyectoResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	98 ms
Función	Eliminar_Proyecto
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_proyecto"> <soapenv:Header/> <soapenv:Body> <urn:Eliminar_Proyecto soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <nombre_p xsi:type="xsd:string">Sistema de Gestión</nombre_p> </urn:Eliminar_Proyecto> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> xmlns:ns1="urn:wSDL_proyecto" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/"> <SOAP-ENV:Body> <ns1:Eliminar_ProyectoResponse> <Eliminar_ProyectoReturn xsi:type="xsd:integer">1</Eliminar_ProyectoReturn> </ns1:Eliminar_ProyectoResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	181 ms
Función	Obtener_Proyectos
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_proyecto"> <soapenv:Header/> <soapenv:Body> <urn:Obtener_Proyectos soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> </soapenv:Body> </soapenv:Envelope> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

Respuesta	<pre><SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_proyecto" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <SOAP-ENV:Body> <ns1:Obtener_ProyectosResponse> <Obtener_ProyectosReturn SOAP-ENC:arrayType="ns1:CProyecto[1]" xsi:type="ns1:CProyectoArray"> <item xsi:type="ns1:CProyecto"> <alcance xsi:type="xsd:string">Validacion Cliente</alcance> <estado xsi:type="xsd:string">negocio</estado> <fac xsi:nil="true"/> <fecha_f xsi:type="xsd:string">2009-11-25</fecha_f> <fecha_i xsi:type="xsd:string">2009-06-25</fecha_i> <fondo_p xsi:type="xsd:float">123456</fondo_p> <id_p xsi:type="xsd:integer">13</id_p> <id_s xsi:type="xsd:integer">15</id_s> <importe_s xsi:nil="true"/> <linea xsi:type="xsd:string">RV</linea></pre>
------------------	---

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <m_aprob xsi:type="xsd:string">Marco</m_aprob> <mont_c xsi:type="xsd:float">123456</mont_c> <monto_t xsi:type="xsd:float">123456</monto_t> <nombre_e xsi:type="xsd:string">Albet</nombre_e> <nombre_p xsi:type="xsd:string">Simulado</nombre_p> </item> </Obtener_ProyectosReturn> </ns1:Obtener_ProyectosResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	74 ms
Función	Obtener_Proyectos_Dado_Estado
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_proyecto"> <soapenv:Header/> <soapenv:Body> <urn:Obtener_Proyectos_Dado_Estado soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <estado xsi:type="xsd:string">negocio</estado> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> </urn:Obtener_Proyectos_Dado_Estado> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_proyecto" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <SOAP-ENV:Body> <ns1:Obtener_Proyectos_Dado_EstadoResponse> <Obtener_Proyectos_Dado_EstadoReturn SOAP- ENC:arrayType="ns1:CProyecto[1]" xsi:type="ns1:CProyectoArray"> <item xsi:type="ns1:CProyecto"> <alcance xsi:type="xsd:string">Validacion Cliente</alcance> <estado xsi:type="xsd:string">negocio</estado> <fac xsi:nil="true"/> <fecha_f xsi:type="xsd:string">2009-11-25</fecha_f> <fecha_i xsi:type="xsd:string">2009-06-25</fecha_i> <fondo_p xsi:type="xsd:float">123456</fondo_p> <id_p xsi:type="xsd:integer">13</id_p> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <id_s xsi:type="xsd:integer">15</id_s> <importe_s xsi:nil="true"/> <linea xsi:type="xsd:string">RV</linea> <m_aprob xsi:type="xsd:string">Marco</m_aprob> <mont_c xsi:type="xsd:float">123456</mont_c> <monto_t xsi:type="xsd:float">123456</monto_t> <nombre_e xsi:type="xsd:string">Albet</nombre_e> <nombre_p xsi:type="xsd:string">Simulado</nombre_p> </item> </Obtener_Proyectos_Dado_EstadoReturn> </ns1:Obtener_Proyectos_Dado_EstadoResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope> </pre>
TR	74 ms
Función	Obtener_Proyectos_Dado_facultad
Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_proyecto"> <soapenv:Header/> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre> <soapenv:Body> <urn:Obtener_Proyectos_Dado_facultad soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <fac xsi:type="xsd:string">fac5</fac> </urn:Obtener_Proyectos_Dado_facultad> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_proyecto" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <SOAP-ENV:Body> <ns1:Obtener_Proyectos_Dado_facultadResponse> <Obtener_Proyectos_Dado_facultadReturn SOAP- ENC:arrayType="ns1:CProyecto[1]" xsi:type="ns1:CProyectoArray"> <item xsi:type="ns1:CProyecto"> <alcance xsi:type="xsd:string">Validacion Cliente</alcance> <estado xsi:type="xsd:string">negocio</estado> <fac xsi:type="xsd:string">fac5</fac> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre><fecha_f xsi:type="xsd:string">2009-11-25</fecha_f> <fecha_i xsi:type="xsd:string">2009-06-25</fecha_i> <fondo_p xsi:type="xsd:float">123456</fondo_p> <id_p xsi:type="xsd:integer">13</id_p> <id_s xsi:type="xsd:integer">15</id_s> <importe_s xsi:nil="true"/> <linea xsi:type="xsd:string">RV</linea> <m_aprob xsi:type="xsd:string">Marco</m_aprob> <mont_c xsi:type="xsd:float">123456</mont_c> <monto_t xsi:type="xsd:float">123456</monto_t> <nombre_e xsi:type="xsd:string">Albet</nombre_e> <nombre_p xsi:type="xsd:string">Simulador</nombre_p> </item> </Obtener_Proyectos_Dado_facultadReturn> </ns1:Obtener_Proyectos_Dado_facultadResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>
TR	72 ms
Función	Obtener_Proyectos_Dado_IdSolicitud

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

Petición	<pre> <soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_proyecto"> <soapenv:Header/> <soapenv:Body> <urn:Obtener_Proyectos_Dado_IdSolicitud soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <id xsi:type="xsd:integer">15</id> </urn:Obtener_Proyectos_Dado_IdSolicitud> </soapenv:Body> </soapenv:Envelope> </pre>
Respuesta	<pre> <SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_proyecto" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"> <SOAP-ENV:Body> <ns1:Obtener_Proyectos_Dado_IdSolicitudResponse> <Obtener_Proyectos_Dado_IdSolicitudReturn SOAP- ENC:arrayType="ns1:CProyecto[1]" xsi:type="ns1:CProyectoArray"> </pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

```
<item xsi:type="ns1:CProyecto">
  <alcance xsi:type="xsd:string">Validacion Cliente</alcance>
  <estado xsi:type="xsd:string">negocio</estado>
  <fac xsi:type="xsd:string">fac5</fac>
  <fecha_f xsi:type="xsd:string">2009-11-25</fecha_f>
  <fecha_i xsi:type="xsd:string">2009-06-25</fecha_i>
  <fondo_p xsi:type="xsd:float">123456</fondo_p>
  <id_p xsi:type="xsd:integer">13</id_p>
  <id_s xsi:type="xsd:integer">15</id_s>
  <importe_s xsi:nil="true"/>
  <linea xsi:type="xsd:string">RV</linea>
  <m_aprob xsi:type="xsd:string">Marco</m_aprob>
  <mont_c xsi:type="xsd:float">123456</mont_c>
  <monto_t xsi:type="xsd:float">123456</monto_t>
  <nombre_e xsi:type="xsd:string">Albet</nombre_e>
  <nombre_p xsi:type="xsd:string">Simulador</nombre_p>
</item>
</Obtener_Proyectos_Dado_IdSolicitudReturn>
</ns1:Obtener_Proyectos_Dado_IdSolicitudResponse>
```


CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre></SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>
TR	80 ms
Función	Obtener_Proyectos_Dado_linea_Trabajo
Petición	<pre><soapenv:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:urn="urn:wSDL_proyecto"> <soapenv:Header/> <soapenv:Body> <urn:Obtener_Proyectos_Dado_linea_Trabajo soapenv:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/"> <linea xsi:type="xsd:string">RV</linea> </urn:Obtener_Proyectos_Dado_linea_Trabajo> </soapenv:Body> </soapenv:Envelope></pre>
Respuesta	<pre><SOAP-ENV:Envelope SOAP- ENV:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/" xmlns:SOAP- ENV="http://schemas.xmlsoap.org/soap/envelope/" xmlns:ns1="urn:wSDL_proyecto" xmlns:SOAP- ENC="http://schemas.xmlsoap.org/soap/encoding/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"></pre>

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre><SOAP-ENV:Body> <ns1:Obtener_Proyectos_Dado_linea_TrabajoResponse> <Obtener_Proyectos_Dado_linea_TrabajoReturn SOAP- ENC:arrayType="ns1:CProyecto[1]" xsi:type="ns1:CProyectoArray"> <item xsi:type="ns1:CProyecto"> <alcance xsi:type="xsd:string">Validacion Cliente</alcance> <estado xsi:type="xsd:string">negocio</estado> <fac xsi:type="xsd:string">fac5</fac> <fecha_f xsi:type="xsd:string">2009-11-25</fecha_f> <fecha_i xsi:type="xsd:string">2009-06-25</fecha_i> <fondo_p xsi:type="xsd:float">123456</fondo_p> <id_p xsi:type="xsd:integer">13</id_p> <id_s xsi:type="xsd:integer">15</id_s> <importe_s xsi:nil="true"/> <linea xsi:type="xsd:string">RV</linea> <m_aprob xsi:type="xsd:string">Marco</m_aprob> <mont_c xsi:type="xsd:float">123456</mont_c> <monto_t xsi:type="xsd:float">123456</monto_t> <nombre_e xsi:type="xsd:string">Albet</nombre_e> <nombre_p xsi:type="xsd:string">Simulador</nombre_p></pre>
--	---

CAPÍTULO 4: IMPLEMENTACIÓN Y PRUEBAS

	<pre></item> </Obtener_Proyectos_Dado_linea_TrabajoReturn> </ns1:Obtener_Proyectos_Dado_linea_TrabajoResponse> </SOAP-ENV:Body> </SOAP-ENV:Envelope></pre>
TR	93 ms

4.4 Conclusiones

Una vez concluido el proceso de implementación, se han obtenido servicios web con la totalidad de las funcionalidades previstas para el correcto funcionamiento en la ejecución de los diferentes procesos de negocios definidos en capítulos anteriores. En este capítulo también se comprobó a través de las pruebas a los servicios el correcto funcionamiento de los mismos plasmado así en los resultados que arrojaron.

CONCLUSIONES

- Ha sido realizado un estudio de los conceptos y definiciones necesarias para comprender el proceso de desarrollo de la Arquitectura Orientada a Servicios en los Procesos de Gestión de Proyectos de ALBET.
- Se realizó un estudio de las tecnologías, lenguajes y herramientas que se tuvieron en cuenta para llevar a cabo el proceso de desarrollo, basado completamente en software no propietario lo cual obedece a las políticas definidas por la empresa ALBET.
- Se realizó un modelo de negocio basado en procesos donde se describieron los diferentes procesos de gestión de proyectos de ALBET los cuales cubren todo el ciclo de vida de un proyecto. Estos son: aprobación de la solicitud de negocio, creación de la oferta, contratación, ejecución de proyecto, aprobación de la ficha técnica y el proceso de gestión con suministro con proveedores.
- Concluido el proceso de diseño e implementación, se han obtenido servicios web con la totalidad de las funcionalidades previstas para el correcto funcionamiento en la ejecución de los diferentes procesos de negocios.
- Se realizaron de forma satisfactoria las pruebas a los servicios web.

RECOMENDACIONES

- Que la universidad continúe con el proceso de adquisición de la metodología CBDI para dar lugar al desarrollo de proyectos con más calidad.
- Que se continúe con la orquestación de los servicios que aquí se describen.
- Que la universidad de mayor preparación a los estudiantes y profesores en el tema de SOA por la importancia que está revistiendo en la actualidad.
- Que las empresas del país hagan un estudio y si es factible introduzcan cuanto antes SOA para que obtengan los beneficios que la misma brinda.

REFERENCIAS BIBLIOGRÁFICAS

1. **Universidad de las Ciencias Informáticas.** *La producción en la UCI.* [En línea] [Citado el: 15 de diciembre de 2008.] <http://www.uci.cu/?q=node/46>.
2. **Gil, Melchor.** *Carta del rector.* [En línea] [Citado el: 15 de diciembre de 2008.] <http://www.uci.cu/?q=node/47>.
3. **López, Salvio, y otros.** *Por un enfoque social en el concepto de “Nuevas tecnologías de la informática y la comunicación”.* [En línea] 2006. [Citado el: 15 de diciembre de 2008.] <http://revistas.mes.edu.cu/Pedagogia-Universitaria/articulos/2006/4/189406408.pdf>.
4. **Jordi, Adell.** *Tendencias en educación en la sociedad de las tecnologías de la información.* [En línea] noviembre de 1997. [Citado el: 15 de diciembre de 2008.] <http://www.uib.es/depart/gte/revelec7.html>.
5. **W3C.** W3C. *Web Services Glossary.* [En línea] W3C Working Group, 11 de febrero de 2004. [Citado el: 6 de enero de 2009.] <http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/>.
6. **He, Hao.** XML.com. *What Is Service-Oriented Architecture.* [En línea] 30 de septiembre de 2003. [Citado el: 9 de enero de 2009.] <http://webservices.xml.com/pub/a/ws/2003/09/30/soa.html>.
7. **Microsoft Corporation.** *La Arquitectura Orientada a Servicios (SOA) de Microsoft.* [En línea] diciembre de 2006. [Citado el: 15 de enero de 2009.] http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf.
8. **Barco, Antonio.** *Arquitectura Orientada a Servicios (SOA). Principios de la orientación a servicios.* [En línea] 22 de junio de 2006. [Citado el: 6 de marzo de 2009.] http://arquitecturaorientadaaservicios.blogspot.com/2006_06_01_archive.html.
9. **Morales, Carlos Andrés.** *Estado del Arte: Servicios Web.* Bogotá : s.n., 2006.
10. **Erl, Thomas.** *Service-Oriented Architecture: Concepts, Technology, and Design.* s.l. : Prentice Hall PTR, 2005. 0-13-185858-0.

REFERENCIAS BIBLIOGRÁFICAS

11. **Microsoft Visual Studio 2005.** MSDN. *Formatos de mensajes SOAP estándar especificados utilizando WSDL.* [En línea] [Citado el: 15 de febrero de 2009.] [http://msdn.microsoft.com/es-es/library/x2ccke44\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/x2ccke44(VS.80).aspx).
12. **Vega Lebún, Carlos Arturo.** eumed.net. *INTEGRACIÓN DE HERRAMIENTAS DE TECNOLOGÍAS DE INFORMACIÓN "PORTALES COLABORATIVOS DE TRABAJO" COMO SOPORTE EN LA ADMINISTRACIÓN DEL CONOCIMIENTO.* [En línea] 2007. [Citado el: 22 de febrero de 2009.] <http://www.eumed.net/tesis/2007/cavl/Estandares%20de%20los%20servicios%20Web.htm>. ISBN-13: 978-84-691-1002-7.
13. **Cejas, Julio.** Intalio BPMN + BPEL + Open Source. *Primeros Pasos con Intalio BPM.* [En línea] 2007. [Citado el: 17 de febrero de 2009.] <http://intaliobpm.blogspot.com/2007/03/primeros-pasos-con-intalio-bpm.html>.
14. **Centeno, Karina, Rojas, Danisbel y Solís, Hector Manuel.** *Componente de Seguridad para aplicaciones del Área Temática Sistemas de Apoyo a la Salud.* Ciudad de La Habana : UCI, 2008.
15. **Sun Microsystems, Inc.** MySQL. *Panorámica del sistema de gestión de base de datos MySQL.* [En línea] [Citado el: 10 de diciembre de 2008.] <http://dev.mysql.com/doc/refman/5.0/es/what-is.html>.
16. **Pecos, Daniel.** PostGreSQL vs. MySQL. *PostGreSQL vs. MySQL.* [En línea] [Citado el: 26 de marzo de 2009.] http://www.netpecos.org/docs/mysql_postgres/index.html.
17. **Rodas, Raul.** linuxCentro.net. *Características de PHP.* [En línea] LinuxCentro.Net, 22 de febrero de 2007. [Citado el: 15 de marzo de 2009.] <http://www.linuxcentro.net/linux/staticpages/index.php?page=CaracteristicasPHP>.
18. **eviware.com.** eviware SoapUI. *Feature Overview.* [En línea] [Citado el: 20 de mayo de 2009.] <http://www.soapui.org/features.html>.
19. **Hernández, Enrique.** Teleformacion. *El Lenguaje Unificado de Modelado (UML).* [En línea] [Citado el: 22 de enero de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11361>.

REFERENCIAS BIBLIOGRÁFICAS

20. **GARZ, JAVIER.** Teleformacion. *SISTEMAS DE INFORMACIÓN.* [En línea] [Citado el: 23 de febrero de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11361>.
21. **Hernández, Violena.** Teleformacion. *Modelado de procesos de negocio.* [En línea] [Citado el: 23 de febrero de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11880>.
22. **Ruiz, Francisco.** Teleformacion. *Tecnología para la Gestión de Procesos de Negocio.* [En línea] noviembre de 2006. [Citado el: 7 de marzo de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11880>.
23. **Teleformacion.** Teleformacion. *BPMN Notación de modelado de procesos de negocio.* [En línea] [Citado el: 24 de febrero de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11880> .
24. **Visual Paradigm.** Visual Paradigm. *Why Visual Paradigm for UML?* [En línea] [Citado el: 26 de febrero de 2009.] <http://www.visual-paradigm.com/product/vpuml/>.
25. **IMPROVEN consultores.** Teleformacion. *GESTIÓN Y REINGENIERÍA DE PROCESOS.* [En línea] [Citado el: 26 de febrero de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11880>.
26. **Ruiz, Francisco.** Teleformacion. *Proceso Software y Gestión del Conocimiento.* [En línea] [Citado el: 26 de marzo de 2009.]

BIBLIOGRAFÍA

1. **Barco, Antonio.** Arquitectura Orientada a Servicios (SOA). *Principios de la orientación a servicios*. [En línea] 22 de junio de 2006. [Citado el: 6 de marzo de 2009.] http://arquitecturaorientadaaservicios.blogspot.com/2006_06_01_archive.html.
2. **Becerril, Francisco.** *JAVA a su alcance*. México: McGRAW-HILL INTERAMERICANA EDITORES, S.A, 1998. ISBN 970-10-1774-9.
3. **Carter, Sandy.** *The New Language*. s.l. : IBM Press, 2007. ISBN-13: 978-0-13-195654-4.
4. **Cejas, Julio.** Intalio BPMN + BPEL + Open Source. *Primeros Pasos con Intalio BPM*. [En línea] 2007. [Citado el: 17 de febrero de 2009.] <http://intaliobpm.blogspot.com/2007/03/primeros-pasos-con-intalio-bpm.html>.
5. **Erl, Thomas.** *Service-Oriented Architecture: Concepts, Technology, and Design*. s.l. : Prentice Hall PTR, 2005. 0-13-185858-0.
6. **Ethan, Cerami.** *Web Services*. First. s.l. : O'Reilly, 2002.
7. **Ganci, John, y otros.** *Best Practices for SOA Management*. s.l. : Redpaper, 2007. REDP-4233-00.
8. **GARZ, JAVIER.** Teleformacion. *SISTEMAS DE INFORMACIÓN*. [En línea] [Citado el: 23 de febrero de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11361>.
9. **Hernández, Enrique.** Teleformacion. *El Lenguaje Unificado de Modelado (UML)*. [En línea] [Citado el: 22 de enero de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11361>.
10. **Hernández, Roberto, Fernández, Carlos y Baptista, Pilar.** *Metodología de la investigación*. México : Programas Educativos, S A, 1998. ISBN 970-10-1899-0.
11. **Hernández, Violena.** Teleformacion. *Modelado de procesos de negocio*. [En línea] [Citado el: 23 de febrero de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11880>.
12. **Hurwitz, Judith, y otros.** *Service Oriented Architecture for Dummies*. Indianapolis : Wiley Publishing, Inc, 2007. NJ 07030-5774.

13. **IMPROVEN consultores.** Teleformacion. *GESTIÓN Y REINGENIERÍA DE PROCESOS*. [En línea] [Citado el: 26 de febrero de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11880>.
14. **Krafzig, Dirk, Banke, Karl y Slama, Dirk.** *Enterprise SOA: Service-Oriented Architecture Best Practices*. s.l. : Prentice Hall PTR, 2004. ISBN: 0-13-146575-9.
15. **Microsoft Corporation.** *La Arquitectura Orientada a Servicios (SOA) de Microsoft*. [En línea] diciembre de 2006. [Citado el: 15 de enero de 2009.] http://download.microsoft.com/download/c/2/c/c2ce8a3a-b4df-4a12-ba18-7e050aef3364/070717-Real_World_SOA.pdf.
16. **Microsoft Visual Studio 2005.** MSDN. *Formatos de mensajes SOAP estándar especificados utilizando WSDL*. [En línea] [Citado el: 15 de febrero de 2009.] [http://msdn.microsoft.com/es-es/library/x2ccke44\(VS.80\).aspx](http://msdn.microsoft.com/es-es/library/x2ccke44(VS.80).aspx).
17. PHP Ya. *POO (Programación Orientada a Objetos)*. [En línea] [Citado el: 4 de marzo de 2009.] <http://phpya.com.ar/poo/>.
18. PHP Ya. *Curso PHP*. [En línea] [Citado el: 4 de marzo de 2009.] <http://phpya.com.ar>.
19. **Petritsch, Helmut.** *Service-Oriented Architecture (SOA)*.
20. **Ratschiller, Tobias y Gerken, Till.** *Web Application Development with PHP 4.0*. s.l. : New Riders Publishing, 2000. ISN: 0-7357-0997-1.
21. **Ruiz, Francisco.** Teleformacion. *Tecnología para la Gestión de Procesos de Negocio*. [En línea] noviembre de 2006. [Citado el: 7 de marzo de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11880>.
22. **Ruiz, Francisco.** Teleformacion. *Proceso Software y Gestión del Conocimiento*. [En línea] [Citado el: 26 de marzo de 2009.]
23. **Sæther, Stig, y otros.** *Manual de PHP*. s.l. : Grupo de documentación de PHP, 2001.

24. **Salvador, Pozo.** MySQL con Clase. *Gestión de Base de Datos*. [En línea] marzo de 2004. [Citado el: 10 de diciembre de 2008.] <http://mysql.conclase.net/curso/>.
25. **Teleformacion.** Teleformacion. *BPMN Notación de modelado de procesos de negocio*. [En línea] [Citado el: 24 de febrero de 2009.] <http://eva.uci.cu/mod/resource/view.php?id=11880> .

GLOSARIO DE TÉRMINOS

CGI: Common Gateway Interface es una importante tecnología de la World Wide Web que permite a un cliente solicitar datos de un programa ejecutado en un servidor web. CGI especifica un estándar para transferir datos entre el cliente y el programa. Es un mecanismo de comunicación entre el servidor web y una aplicación externa.

DTD: Siglas en inglés de Document Type Definition. La definición de tipo de documento DTD es una descripción de estructura y sintaxis de un documento XML o SGML. Su función básica es la descripción del formato de datos, para usar un formato común y mantener la consistencia entre todos los documentos que utilicen la misma DTD. De esta forma, dichos documentos, pueden ser validados, conocen la estructura de los elementos y la descripción de los datos que trae consigo cada documento, y pueden además compartir la misma descripción y forma de validación dentro de un grupo de trabajo que usa el mismo tipo de información.

Framework: Es una estructura de soporte definida, mediante la cual otro proyecto de software puede ser organizado y desarrollado. Típicamente, puede incluir soporte de programas, bibliotecas y un lenguaje interpretado entre otros software para ayudar a desarrollar y unir los diferentes componentes de un proyecto. Representa una arquitectura de software que modela las relaciones generales de las entidades del dominio. Provee una estructura y una metodología de trabajo la cual extiende o utiliza las aplicaciones del dominio.

HTTP: Es el protocolo usado en cada transacción de la Web (WWW). HTTP fue desarrollado por el consorcio W3C y la IETF. HTTP define la sintaxis y la semántica que utilizan los elementos software de la arquitectura web (clientes, servidores, proxies) para comunicarse. Es un protocolo orientado a transacciones y sigue el esquema petición-respuesta entre un cliente y un servidor. HTTP es un protocolo sin estado, es decir, que no guarda ninguna información sobre conexiones anteriores.

JMS: Servicios de Mensajería de Java es la solución creada por SUN para el uso de colas de mensajes. Este es un estándar de mensajería que permite a los componentes de aplicaciones basados en la plataforma de Java 2 crear, enviar, recibir y leer mensajes. También hace posible la comunicación confiable de manera síncrona y asíncrona. El servicio de mensajería también es conocido como

Middleware Orientado a Mensajes y es una herramienta universalmente reconocida para la construcción de aplicaciones empresariales. Dicha API es parte integral de la versión 2 de Java.

JSP: JavaServer Pages es una tecnología Java que permite generar contenido dinámico para web, en forma de documentos HTML, XML o de otro tipo. Las JSP's permiten la utilización de código Java mediante scripts. Además, es posible utilizar algunas acciones JSP predefinidas mediante etiquetas. Estas etiquetas pueden ser enriquecidas mediante la utilización de Librerías de Etiquetas externas e incluso personalizadas.

Repositorio: Depósito o archivo es un sitio centralizado donde se almacena y mantiene información digital. Los depósitos están preparados para distribuirse habitualmente sirviéndose de una red informática como Internet o en un medio físico como un disco compacto. Y pueden ser de acceso público, o pueden estar protegidos y necesitar de una autenticación previa.

Servlets: Son objetos que corren dentro del contexto de un contenedor de servlets y extienden su funcionalidad. También podrían correr dentro de un servidor de aplicaciones, que, además de contenedor para servlet, tendrá contenedor para objetos más avanzados, como son los EJB. La palabra servlet deriva de otra anterior, applet, que se refería a pequeños programas escritos en Java que se ejecutan en el contexto de un navegador web. Por contraposición, un servlet es un programa que se ejecuta en un servidor. El uso más común de los servlets es generar páginas web de forma dinámica a partir de los parámetros de la petición que envíe el navegador web.

SSL: Secure Sockets Layer -Protocolo de Capa de Conexión Segura- (SSL) y Transport Layer Security -Seguridad de la Capa de Transporte- (TLS), su sucesor, son protocolos criptográficos que proporcionan comunicaciones seguras por una red, comúnmente Internet.

TIC: Las tecnologías de la información y la comunicación son un conjunto de servicios, redes, software y dispositivos que tienen como fin la mejora de la calidad de vida de las personas dentro de un entorno, y que se integran a un sistema de información interconectado y complementario. La ciencia informática se encarga del estudio, desarrollo, implementación, almacenamiento y distribución de la información mediante la utilización de hardware y software como recursos de los sistemas informáticos.

WS-Policy: Es una especificación que forma parte de la familia de especificaciones de tecnologías basadas en servicios web del W3C. Esta especificación permite a los programadores de servicios web anunciar sus políticas relativas a seguridad, calidad de servicio, etc. y a los clientes de servicios web

especificar sus requisitos de calidad de servicio, seguridad, latencia. La especificación WS-Policy está formada por un conjunto de especificaciones que describen las capacidades y restricciones asociadas a determinados servicios web.

WS-Security: Es un protocolo de comunicaciones que suministra un medio para aplicar seguridad a los Servicios Web. WS-Security incorpora características de seguridad en el encabezado de un mensaje SOAP, trabajando en la capa aplicación. Así asegura seguridad extremo a extremo.

XML: Siglas en inglés de Extensible Markup Language, es un metalenguaje extensible de etiquetas desarrollado por el World Wide Web Consortium (W3C). Permite definir la gramática de lenguajes específicos. Por lo tanto XML no es realmente un lenguaje en particular, sino una manera de definir lenguajes para diferentes necesidades. Algunos de estos lenguajes que usan XML para su definición son XHTML, SVG, MathML.