



Silenus Consultoría

SOA Silenus

SOA/09009

Mayo de 2009



Análisis SOA Silenus



Índice

1	Introducción.....	4
2	Contexto del Proyecto.....	7
3	Casos de Uso.....	11
3.1	CU 1: Creación y Modificación de Cuentas.....	11
3.2	CU 2: Creación de Oportunidades de Negocio.....	11
3.3	CU 3: Oportunidades de Negocio Ganadas.....	11
3.4	CU 4: Cierre de Proyecto.....	11
3.5	CU 5: Facturación de Proyecto.....	12
3.6	CU 6: Cobro de Proyecto.....	12
4	Descripción Funcional	13
4.1	Enterprise Service Bus.....	15
4.1.1	Mule.....	15
4.1.2	ServiceMix / Fuse ESB.....	15
4.1.3	Jitterbit.....	16
4.1.4	OpenESB	16
4.1.5	Petals.....	16
4.1.6	WSO2 / Apache Synapse.....	16
4.1.7	Jboss ESB.....	17
4.2	Motores BPM / BPEL.....	17
4.2.1	Apache ODE.....	17
4.2.2	Intalio BPM.....	17
4.2.3	Active BPEL.....	18
4.2.4	Jboss JBPM.....	18
4.2.5	Orchestra y Bonita.....	18
4.3	Herramientas de Mapeo y Protocolos.....	18
4.3.1	Smooks.....	18
4.3.2	Apache CXF.....	19



4.3.3 Apache ActiveMQ.....	19
4.3.4 Stomps.....	19
5 Plan de Proyecto.....	20
5.1 Tareas.....	20



1 Introducción

Este documento sirve como punto de partida del proyecto SOA Silenus cuyos objetivos primordiales son:

- ✓ Empleo de Tecnologías para implementar Flujos de Trabajo verticales y horizontales entre aplicaciones de código abierto para facilitar su integración.
- ✓ Desarrollar una Prueba de Concepto que integre varios aplicativos de código abierto y defina flujos de trabajo.

Este proyecto es un caso atípico para Silenus Consultoría dado que el cliente es la propia empresa y el resultado supondrá las siguientes ventajas:

- ✓ Mejorar la calidad del servicio de Silenus Consultoría, gracias al empleo de tecnologías de integración apoyadas en estándares de mercado.
- ✓ Incrementar el portfolio de productos software y servicios de integración que ofrece la empresa.
- ✓ Proporcionar un incremento de valor del Capital Humano de la empresa que adquirirá formación en las tecnologías investigadas y productos escogidos para la demostración de la Prueba de Concepto.
- ✓ Disminuir los costes de integración de soluciones al contar con soluciones empaquetadas para problemas comunes de integración de componentes.
- ✓ Orientar los proyectos hacia la implementación de Procesos de Negocio como los visualizan los clientes.



Silenus Consultoría es una empresa cuya actividad se centra en cuatro áreas de actividad:

1. Prestación de Servicios de Consultoría en el ámbito de las Tecnologías de la Información y de las Comunicaciones.
2. Diseño e Implantación de Sistemas de Información en las organizaciones.
3. Soluciones de Inteligencia de Negocio.
4. Outsourcing de sistemas

Desde el comienzo de su actividad en Mayo de 2007 Silenus participa en diversos proyectos para sus clientes.

Satec

- ✓ Asistencia técnica para optimización de Red xEthernet de R Cable
- ✓ Configuración e instalación de Aceleradores de Aplicaciones de Juniper Networks para R Cable.

R Cable (Operador de Cable de Galicia)

- ✓ Ingeniería de Proyectos Especiales para el Área de Grandes Clientes
- ✓ Pruebas y Medidas de Calidad sobre Plataforma de Streaming de Vídeo
- ✓ Ingeniería e implantación de proyectos para Pymes
- ✓ Auditoría de Seguridad, Red y Sistemas de Grandes Clientes

APEC Lugo

- ✓ Diseño e implantación de la red multiservicio y de seguridad para la sede social de APEC Lugo.
- ✓ Implantación de CRM y Sistema de Gestión Documental



Fundación Laboral de la Construcción de Galicia

- ✓ Implantación de SugarCRM Profesional
- ✓ Auditoría de Red y Seguridad
- ✓ Outsourcing de Sistemas
- ✓ Herramienta de BI para explotación de datos de encuestas
- ✓ Sistema de gestión de Cursos basado en Intalio

EMESA

- ✓ Auditoría de red y sistemas
- ✓ Elaboración de Plan de Sistemas
- ✓ Outsourcing de Sistemas



2 Contexto del Proyecto

Silenus Consultoría es una joven empresa que apuesta por el empleo de tecnologías de código abierto para ofrecer soluciones de bajo coste y calidad contrastada a sus cliente. Dentro de su cartera de productos se encuentran los aplicativos:

- ✓ **Alfresco**: Gestor Documental que permite organizar, localizar, publicar, compartir y mantener documentación en diferentes formatos soportando estándares de mercado y proporcionando interoperabilidad con protocolos como HTTP, WebDAV, CIFS, FTP para el acceso a la documentación.
- ✓ **SugarCRM**: CRM o Gestor de Relaciones con Clientes más completo dentro de los aplicativos de código abierto, capaz de mejorar el grado de satisfacción del cliente y controlar el ciclo de venta en todas sus fases.
- ✓ **OpenBravo**: ERP o Planificador de Recursos Empresariales, permite organizar líneas de producción, coordinando los diferentes departamentos y operaciones de una empresa productiva tales como: compras, fabricación, almacenamiento, logística, facturación y contabilidad.

Dentro del ámbito de las aplicaciones de código abierto encontramos soluciones muy heterogéneas:

- x Cada producto emplea un lenguaje de programación distinto, lo que dificulta su integración en un mismo entorno.
- x Un aplicativo puede tanto ofrecer un mecanismo para ampliar su funcionalidad como carecer del mismo.
- x Los modelos de datos de aplicaciones distintas no son compatibles entre sí, por lo que resulta complicado hacer



compatibles los modelos de las aplicaciones que se desea integrar.

- x Cada aplicativo se construye con diferentes Framework. Un Framework es un esquema o estructura, con una serie de reglas que permite el desarrollo de nuevas funcionalidades dentro de un aplicativo, siguiendo una convención.
- x Los clientes cuentan con aplicaciones legadas o propietarias que también es necesario incluir en los proyectos de integración.
- x Los clientes necesitan soluciones globales y no aplicaciones dispersas, incapaces de comprender los procesos de negocio.

Para resolver este tipo de problemas existen aplicaciones y modelos de negocio definidos tales como:

- ✓ **EAI:** Integración de Aplicaciones Empresariales engloba las metodologías, procesos, herramientas y tecnologías usadas para conectar sistemas, datos y procesos de una entidad o de un conjunto de entidades. Los sistemas EAI que aparecieron a lo largo de los años 90 se basaban en una arquitectura de mensajería a través de un broker responsable de comunicar las aplicaciones entre sí y de transportar los mensajes a cada una de ellas. El principal problema de adopción de EAI fue el coste elevado de las soluciones del mercado que limitaron su implantación y continuidad, así como la falta de un estándar de mercado que derivaba en soluciones propietarias incompatibles entre los diferentes proveedores de aplicativos EAI.
- ✓ **BPM:** Gestores de Procesos de Negocio resultan de la evolución de los EAI para centrarse en la definición de los procesos que son los que determinan las necesidades de integración de las aplicaciones de una empresa. BPM ha intentado estandarizar el lenguaje de definición de procesos así como el intercambio de



información de los mismos empleando un lenguaje común. BPM contempla desde la automatización completa de un proceso hasta los procesos en los que se requiere la intervención humana para completar alguna de sus fases (a esto se le llaman Tareas Humanas o Human Tasks).

- ✓ **BPEL:** Lenguaje de Definición de Procesos Ejecutables es una ampliación de BPM aplicada a la orquestación de servicios web de diferentes aplicativos. Permite definir mediante un fichero XML con un esquema determinado cómo coordinar los diferentes servicios web de distintos aplicativos para implementar un proceso de negocio definido mediante reglas, condiciones y bucles.
- ✓ **SOA:** las Arquitecturas Orientadas a Servicios son un paradigma de la construcción de aplicaciones software en la que prima definir unos contratos (interfaces) para el uso de unos servicios que permitan su reutilización en la construcción de sistemas más complejos que integran diferentes componentes y aplicaciones. Los servicios así definidos, pueden ser orquestados o coordinados mediante otras herramientas y tecnologías como BPM o BPEL. La principal diferencia entre SOA y BPM es cómo se resuelven los problemas de integración entre sistemas. BPM soluciona los problemas de arriba hacia abajo, definiendo primero los procesos e identificando posteriormente a los sistemas y servicios de grano más fino que deben ser definidos para implementar las tareas o subproceso de un proceso. Por el contrario SOA establece que se deben definir primero unos servicios, que deberán ser interoperables (intercambiar datos en un formato que cualquiera pueda ser capaz de consumir y entender) y deberá existir un registro donde se publiquen dichos servicios para su posterior consumo y reutilización.



- ✓ **ESB:** los Buses de Comunicación Empresariales constituyen una tecnología indispensable para la implementación de arquitecturas y servicios SOA. Permiten el registro y gestión de servicios de integración con diferentes aplicaciones, proporcionando una interfaz de comunicación para mediar entre el Bus de comunicaciones y el servicio real que implementa la interfaz. A diferencia de los EAI son distribuidos y se comunican entre sí, y además proporcionan un enrutamiento inteligente de los mensajes empleando reglas que se aplican sobre el propio contenido de los mensajes.

Silenus Consultoría pretende construir una solución de integración para varias aplicaciones de código abierto sobre una arquitectura SOA, que se apoyen sobre un ESB para implementar ejemplos o Pruebas de Concepto de Procesos de Negocio que mejoren la calidad de sus servicios y disminuyan el coste de implementación de los proyectos.



3 Casos de Uso

Con el fin de preparar una demostración de la Prueba de Concepto de la integración SOA de los diferentes aplicativos, se definirán unos Casos de Uso para poner a prueba la arquitectura desarrollada.

3.1 CU 1: Creación y Modificación de Cuentas

Cuando se crea una nueva cuenta en el CRM se desea crear en el Gestor Documental una nueva carpeta para almacenar la documentación relativa a la Cuenta. En caso de modificación del nombre de la cuenta, también debería modificarse la carpeta como consecuencia.

3.2 CU 2: Creación de Oportunidades de Negocio

Cuando se crea una oportunidad de negocio para una Cuenta determinada, es necesario crear dos carpetas dentro de la carpeta de la cuenta. La carpeta “Oferta Técnica” y la carpeta “Oferta Comercial”.

3.3 CU 3: Oportunidades de Negocio Ganadas

Cuando una oportunidad pasa a “ganarse” (es decir, el Cliente acepta la oferta), debe crearse un nuevo cliente en el ERP y crear un nuevo proyecto en el ERP con el nombre y referencia de la oferta. También es necesario crear una carpeta de proyecto en la carpeta de la Cuenta del Gestor Documental.

3.4 CU 4: Cierre de Proyecto

Cuando se cierre el proyecto en el ERP por haber concluido su desarrollo e implementación, es necesario cerrar la oportunidad correspondiente en el CRM y notificarlo al comercial correspondiente.



3.5 CU 5: Facturación de Proyecto

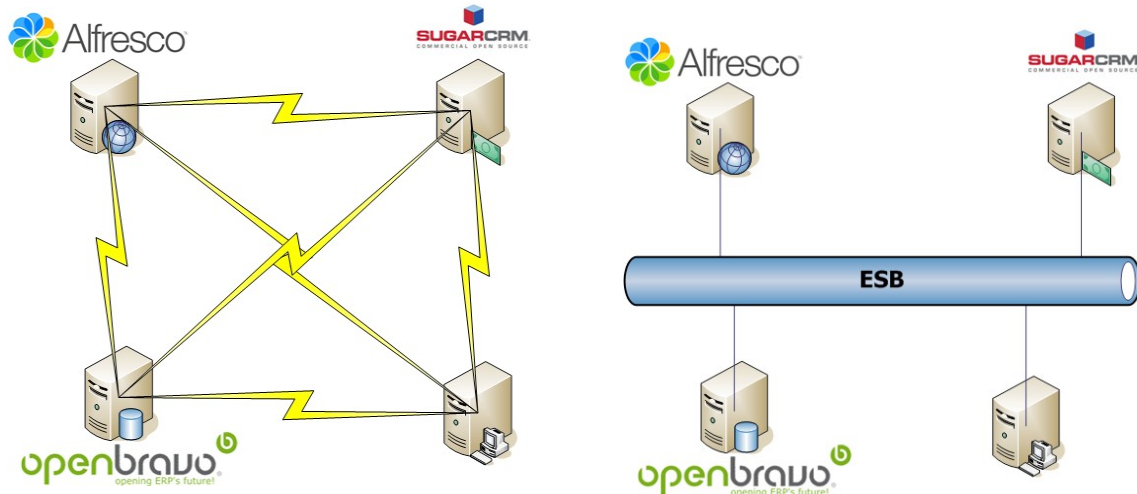
Cuando se facture el proyecto del ERP, deberá almacenarse la factura del mismo en el Gestor Documental.

3.6 CU 6: Cobro de Proyecto

Cuando se cobre el proyecto deberá archivar toda su documentación en el Gestor Documental en una carpeta "Proyectos Cerrados" dentro de la carpeta de la Cuenta.



4 Descripción Funcional



En las figuras anteriores se ilustra la diferencia fundamental en la integración de sistemas. La figura de la izquierda muestra como se integran los sistemas de forma artesanal, comunicándolos unos con otros. En la figura de la derecha se muestra cómo se comunican los sistemas entre sí. Salta a la vista que el número de comunicaciones necesarias para cada aplicación que queremos comunicar con el resto es $2 \times \text{Número de Aplicaciones (3)} = 6$ mientras que en el caso del ESB es siempre 2 (entrante y saliente).

Conforme aumenta el número de aplicaciones que deben de integrarse, se incrementa la complejidad del código de integración que exige que se modifiquen las aplicaciones una y otra vez. Una arquitectura SOA sobre un ESB permite que definidos los servicios e interfaces para una aplicación, sólo sea necesario conectarlos desde el ESB que es ajeno a las aplicaciones y no exige modificación alguna de las mismas.



Para implementar la arquitectura SOA necesitaremos que cada aplicación susceptible de integrarse con el ESB disponga de:

- ✓ Generación de **eventos asíncronos** para enviar mensajes al ESB.
- ✓ **Servicios** web para la comunicación síncrona u otro tipo de servicio como por ejemplo un servicio activado mediante mensajería.

Los **eventos asíncronos** dispararán el envío de un mensaje con el contenido del evento al ESB. Dicho mensaje podrá:

- ✓ Ser transformado en otro tipo de mensaje mediante herramientas de mapeo o transformación para que la aplicación que lo vaya a recibir lo acepte.
- ✓ Enrutarse hacia una aplicación o grupo de aplicaciones concreto.
- ✓ Enriquecerse añadiéndole información adicional
- ✓ Disparar un Flujo de Trabajo.

La integración de sistemas mediante mensajería es objeto de estudio y desarrollo, fruto de lo cual han surgido **patrones de integración** que resuelven muchos de los problemas de una forma sistemática y probada y que se suelen encontrar implementados en los ESB tanto de código abierto como comerciales.

Los **servicios** web u otros servicios permitirán que desde el ESB o desde un motor BPM puedan orquestarse para ejecutar un Proceso de Negocio. Es muy importante que se diseñen pensando en las necesidades de integración futuras puesto que constituyen un contrato de las capacidades que ofrece cada aplicación para integrarse con el resto.



El ESB escogido deberá integrarse con un **motor BPM / BPEL** de manera que los Flujos de Trabajo complejos los lleve a cabo este componente y evite la necesidad de incorporar lógica de negocio a las aplicaciones existentes. Así se desacopla cualquier cambio futuro que desee aplicarse sobre los Flujos de Trabajo por cambios internos en la empresa que los define.

Existe un gran abanico de posibilidades a la hora de elegir los componentes para implementar una solución de integración y parte del proyecto consiste en evaluar los más relevantes y probar su compatibilidad.

4.1 Enterprise Service Bus

4.1.1 Mule

URL: <http://www.mulesource.org>

Se trata de uno de los primeros ESB de código abierto. Cuenta con soporte comercial si se desea y cerca de 20 transportes distintos para protocolos variados. Se integra con JMS, Web Services, BPM / BPEL... Aunque no cumple con el estándar JBI, posee un conector para interactuar con otros contenedores [JBI](#).

4.1.2 ServiceMix / Fuse ESB

URL: <http://servicemix.apache.org/home.html>

Es un ESB que cumple al 100% con JBI. Cuenta con soporte comercial si se desea y lo proporciona IONA a través de su proyecto **[Fuse ESB](#)**. Emplea Apache Camel, Apache CXF, Apache ODE y Apache ActiveMQ como componentes para proporcionar tantos transportes como Mule y servicios similares.



4.1.3 Jitterbit

URL: <http://www.jitterbit.com>

Es un ESB o algo parecido capaz de integrarse con bastantes aplicaciones comerciales conocidas y con otros transportes. Cuenta con un entorno de desarrollo gráfico y con un catálogo de plugins que se denominan jitterpacks.

4.1.4 OpenESB

URL: <https://open-esb.dev.java.net/>

Es un ESB desarrollado por Sun Microsystems y de código abierto. Cuenta con multitud de conectores y cumple al 100% con JBI. Está vinculado estrechamente a Glassfish como contenedor, mientras que otros ESB son agnósticos a ese respecto (ServiceMix o Mule, por ejemplo). Se integra con Netbeans como entorno de desarrollo.

4.1.5 Petals

URL: <http://petals.ow2.org/>

Es un ESB que cumple con JBI al 100% desde el 2008. Soporta múltiples transportes y se integra con BPEL o BPM. Está vinculado a otros proyectos de Object Web como Orchestra (BPEL) o Bonita (BPM).

4.1.6 WSO2 / Apache Synapse

URL: <http://wso2.org/>

Aunque no emplea el término ESB en la descripción del producto, proporciona una implementación de arquitectura SOA. Se trata de una implementación de SCA (Service Component Architecture) y posee el SCDL (Service Component Definition Language) que permite definir componentes que implementan y exponen servicios, con independencia del lenguaje de programación empleado para ello. Los componentes se comunican entre sí como los componentes JBI. Los



servicios tienen un “binding type” que permite publicarlos como servicios web o colas JMS por ejemplo.

4.1.7 Jboss ESB

URL: <http://www.jboss.org/jbossesb/>

Se trata de un ESB de Jboss que se integra con el resto de aplicaciones de Jboss como jBPM (BPM), Jboss MQ / Jboss Messaging (JMS), Drools (para las reglas de enrutado como es Camel para ServiceMix), Smooks para transformaciones.

Prácticamente la totalidad de los ESB anteriores se integran con **Spring Framework** también.

4.2 Motores BPM / BPEL

4.2.1 Apache ODE

URL: <http://ode.apache.org>

Cumple con WS-BPEL 2.0, por tanto es un servicio de orquestación de web services. Se integra con Apache Axis o JBI para las comunicaciones con los servicios web.

Es interesante apuntar que la herramienta de Intalio se apoya sobre Apache ODE.

4.2.2 Intalio BPM

URL: <http://www.intalio.com/products/bpm/>

Es un motor BPM y BPEL que se integra con Mule, ServiceMix o WSO2 / Apache Tuscany. Es 80% opensource, es decir se construye con componentes de código abierto pero no distribuyen dicho código sino que redirigen a la comunidad hacia los proyectos open source con que está construido como Apache ODE (BPEL) o Tempo (BPM) o



Orbeon (Human Tasks / Forms). Es una herramienta bastante completa y cuenta con un diseñador gráfico propio.

4.2.3 Active BPEL

URL: <http://www.activevos.com>

Motor BPEL 1.1 con soporte para transporte SOAP y HTTP (no soporta JMS ni otros transportes). No tiene herramienta para el diseño de Flujos de Trabajo y no soporta WS-Security (capa de seguridad sobre servicios web).

4.2.4 Jboss JBPM

URL: <http://www.jboss.org/jbossjbpm/>

Motor jPDL y BPEL (WS-BPEL 2.0) con un diseñador propio de Flujos de Trabajo. Hasta hace algún tiempo era poco compatible con estándares, pero ahora lo es. Es muy flexible y cómodo de utilizar para personas con experiencia en el desarrollo Java.

4.2.5 Orchestra y Bonita

URL: <http://bonita.objectweb.org/>

<http://orchestra.ow2.org/xwiki/bin/view/Main/WebHome>

BPEL 2.0 y XPDL 1.0. Con un diseñador propio de Flujos de Trabajo, una aplicación para monitorización y gestión de flujos Web 2.0.

4.3 Herramientas de Mapeo y Protocolos

4.3.1 Smooks

URL: <http://www.smooks.org>

Es un framework para procesar datos en formatos tales como XML, CSV, EDI, ... capaz de implementar transformaciones entre los mismos.



4.3.2 Apache CXF

URL: <http://cxf.apache.org/>

Es un framework para construir servicios web capaces de comunicarse a través de diferentes transportes (HTTP, JMS, JBI) y en diferentes protocolos (SOAP, XML/HTTP, RESTful HTTP, o CORBA).

Entre otros estándares de servicios web soporta: WSI Basic Profile, WSDL, WS-Addressing, WS-Policy, WS-ReliableMessaging, WS-Security, WS-SecurityPolicy, y WS-SecureConversation.

4.3.3 Apache ActiveMQ

URL: <http://activemq.apache.org/>

Plataforma de mensajería con soporte para JMS 1.1, empotrable y capaz de comunicarse con aplicaciones escritas en Java, C, C++, C#, Ruby, Perl, Python o PHP.

4.3.4 Stomps

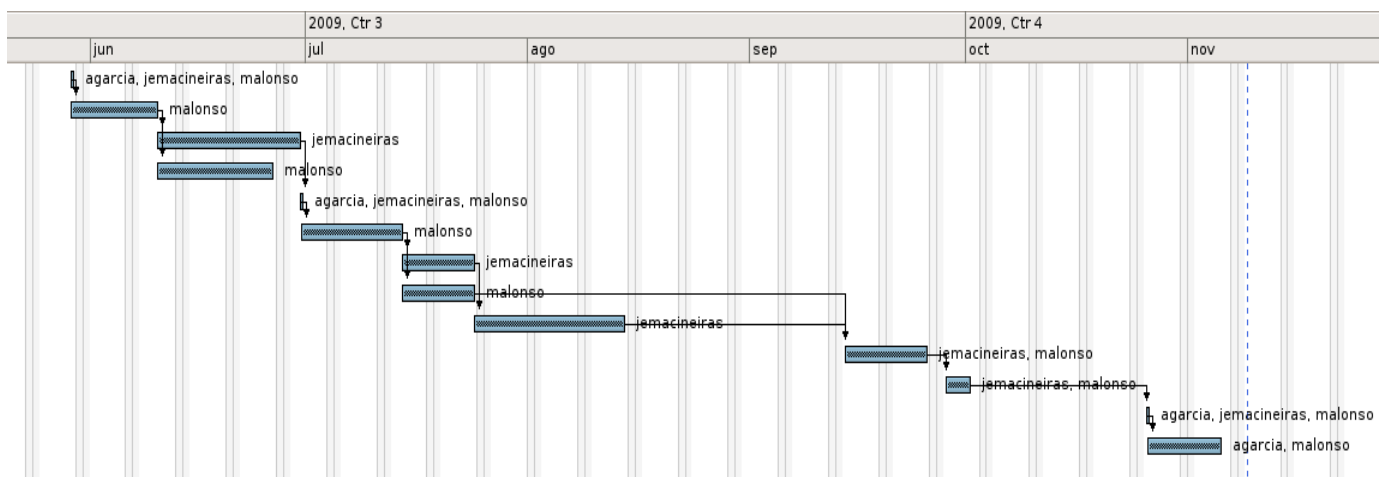
URL: <http://stomp.codehaus.org/Home>

Protocolo simple para el envío de mensajes basados en texto. Tiene un conector para JMS y librerías para PHP que permiten comunicar aplicaciones en PHP con plataformas de mensajería como ActiveMQ.



5 Plan de Proyecto

El proyecto tendrá un periodo inicial de análisis y evaluación de herramientas para implementar la solución y con posterioridad se diseñará la Prueba de Concepto y se implementará convenientemente.



5.1 Tareas

WBS	Nombre	Inicio	Fin	Trabajo
1	Reunión Inicial	May 29	May 29	1d
2	Documentación de Análisis	May 18	Jun 10	8d
3	Pruebas de componentes BPM	Jun 10	Jun 30	14d
4	Pruebas de componentes ESB	Jun 10	Jun 26	12d
5	Reunión de Diseño	Jun 30	Jun 30	1d
6	Documentación de Diseño	Jun 30	Jul 14	10d
7	Desarrollo de Servicios para SugarCRM	Jul 14	Jul 24	8d
8	Desarrollo de Servicios para Alfresco	Jul 14	Jul 24	8d
9	Desarrollo de Flujos de Trabajo	Jul 24	Ago 14	15d
10	Pruebas de Integración	Sep 14	Sep 25	20d
11	Corrección de Errores	Sep 28	Oct 1	8d
12	Reunión de Cierre	Oct 26	Oct 26	1d
13	Documentación Final	Oct 26	Nov 2	10d