

**TEXT2EMOTION: UN GESTOR DE EXPERIÈNCIES SOA.
EXTRACCIÓ D'EMOCIONS PER AL SUPORT A LA GESTIÓ DE RELACIÓ
AMB CLIENTS**

Memòria del Projecte Fi de Carrera
d'Enginyeria en Informàtica
realitzat per
Rodolfo Guichón Aguilar
i dirigit per
Joan Serrat Gual
Bellaterra, 22 de juny de 2011

A los que dan sentido a todo, Susi y Pol.

“Las personas compran emocionalmente y justifican lógicamente”

Buck Rodgers, IBM

Índice General

Resumen	viii
Resum	ix
Summary	x
I Introducción	1
1. Introducción y objetivos	2
1.1. Objetivo del proyecto	2
1.2. Estado del Arte	5
1.3. Estudio de Viabilidad..	7
1.4. Planificación	11
2. Fundamentos tecnológicos	13
2.1. ¿Qué es Cloud Computing?	13
2.2. ¿Qué es Service Oriented Architecture?	15
2.3. Plataformas de desarrollo	17
2.3.1. Google App Engine	17
2.3.2. Amazon EC2	18
2.3.3. Force.com	19
II Servicios de clasificación de experiencias	21
3. Modelado y clasificación de las emociones	22
3.1. Introducción	22
3.2. Parte emocional de las experiencias.	22
3.3. Clasificación de las experiencias	25
3.3.1. Preproceso de textos	25
3.3.2. Cambio de representación.	27
3.3.3. Clasificación.	29

4. Desarrollo de los servicios de clasificación	31
4.1. Introducción.	31
4.2. Modelado.	32
4.3. Ensamblado.	34
4.4. Despliegue.	44
III Casos de estudio	46
5. Caso 1: Departamento Comercial de una industria química	47
5.1. Introducción.	47
5.2. Acción y reacción: creando tareas a partir de valoraciones de experiencias.	48
5.3. Entorno de trabajo en Force.com.	49
5.3.1. Aplicación.	49
5.3.2. Objetos personalizables.	51
5.3.3. Interfaces de usuario.	52
5.3.4. Acceso a servicios externos.	55
5.4. Resultados.	58
6. Caso 2: Community Management	61
6.1. Introducción.	61
6.2. Acción y reacción: creando tareas a partir de valoraciones de experiencias.	62
6.3. Entorno de trabajo en Force.com.	63
6.3.1. Aplicación.	63
6.3.2. Objetos personalizables.	63
6.3.3. Interfaces de usuario.	64
6.3.4. Acceso a servicios externos.	66
6.4. Resultados.	66
Conclusiones	68
Bibliografía	69

Índice de figuras

Figura 1.1.1 Customer Experience Management	2
Figura 1.1.2 Esquema de la solución propuesta	3
Figura 1.2.1 Cloud Computing	6
Figura 2.1.1 Capas de servicios de la nube	14
Figura 2.2.1 Zona de afectación de los contratos	15
Figura 2.2.2 Zona de afectación del acoplamiento	15
Figura 2.2.3 Zona de Afectación de la abstracción	16
Figura 2.2.4 Localización de servicios	16
Figura 2.2.5 Composición de servicios	16
Figura 2.3.1.1 Logotipo Google App Engine	17
Figura 2.3.2.1 Logotipo Amazon Web Services	18
Figura 2.3.3.1 Logotipo Force.com	20
Figura 3.2.1 Factores que afectan al estado emocional	23
Figura 3.2.2 Cuadrantes emocionales	23
Figura 3.3.1.1 Raíces, morfemas, sufijos, prefijos e incrementos	26
Figura 3.3.2.1 Obtención de la matriz Documento-Palabra	27
Figura 3.3.2.2 Latent Semantic Analysis	28
Figura 3.3.3.1 k-nearest Neighbor	29
Figura 4.1.1 Metodología de desarrollo SOA	31
Figura 4.2..1 Modelo de Proceso de Entrenamiento	33
Figura 4.2. 2 Modelo del Proceso de Clasificación	34
Figura 4.3. 1 Contrato de Requerimiento de Servicios	35
Figura 4.3. 2 Contrato de Requerimiento de Servicios	36
Figura 4.3. 3 Diagrama de Unidades Funcionales	37
Figura 4.3. 4 Diagrama de Topología de Servicios	37
Figura 4.3. 5 Diagrama Componente Abstracto	38
Figura 4.3. 6 Especificación: Parsing	39
Figura 4.3. 7 Especificación: Analysing	39
Figura 4.3. 8 Especificación: Collecting	40
Figura 4.3. 9 Especificación: Training	40
Figura 4.3 10 Especificación: Classifying	40
Figura 4.3. 11 Service Data Model	41
Figura 4.3.12 Service Providers	42
Figura 4.3.13 Text2Emotion Service Provider	42
Figura 4.3 14 Subsistema Text2Emotion	43
Figura 4.4 1 Levantamiento de la instancia del servidor de aplicaciones	44
Figura 4.4 2 Consola del servidor de aplicaciones	45
Figura 4.4 3 Despliegue de los servicios Text2Emotion	45
Figura 4.4 4 Web del servicio desplegado	45
Figura 5.2 1 Recordatorio de tarea a realizar	49
Figura 5.3.1 1 Definición de aplicación	49

Figura 5.3.1 2 Definición de ficha	50
Figura 5.3.1 3 Aplicación final	50
Figura 5.3.2 1 Definición de llamada	51
Figura 5.3.2 2 Campos de Llamada	52
Figura 5.3.2 3 Desencadenador de llamada	52
Figura 5.3.3 1 Pantalla de Llamadas	53
Figura 5.3.3 2 Pantalla de Emails	53
Figura 5.3.3 3 Pantalla de Visita	54
Figura 5.3.3 4 Pantalla de Reuniones	54
Figura 5.3.3 5 Pantalla de Fax	55
Figura 5.3.3 6 Pantalla de Seguimiento	55
Figura 5.3.4 1 Acceso a sitio remoto	56
Figura 5.3.4 2 Paso 1: Seleccionar fichero WSDL	56
Figura 5.3.4 3 Paso 2: Análisis del fichero WSDL	57
Figura 5.3.4 4 Paso 3: Generación de las clases Apex	57
Figura 5.3.4 5 Seguimiento de Trabajos de Apex	57
Figura 5.3.4 6 Tarea generada automáticamente	57
Figura 5.4 1 Caso 1: Matriz de Confusión	59
Figura 5.4 2 Caso 1: Matriz de Error Relativo	59
Figura 5.4 3 Caso 1: Matriz de Desviación estándar	59
Figura 5.4 4 Efectividad de proceso comercial	60
Figura 5.4 5 Efectividad estimada con el uso de Text2Emotion	60
Figura 6.3.1 1 Aplicación Final	63
Figura 6.3.3 1 Pantalla de Hoteles	64
Figura 6.3.3 2 Pantalla de Opiniones	64
Figura 6.3.3 3 Pantalla de Respuestas	65
Figura 6.3.3 4 Pantalla de definición de respuestas	65
Figura 6.3.3 5 Pantalla de programación de descargas	66
Figura 6.3.3 6 Caso 2: Matriz de confusión	67
Figura 6.3.3 7 Caso 2: Matriz de Error Relativo	67
Figura 6.3.3 8 Caso 2: Matriz de Desviación Estándar	67

Resumen

Customer Experience Management (CEM) se ha convertido en un factor clave para el éxito de las empresas. CEM gestiona todas las experiencias que un cliente tiene con un proveedor de servicios o productos. Es muy importante saber como se siente un cliente en cada contacto y entonces poder sugerir automáticamente la próxima tarea a realizar, simplificando tareas realizadas por personas.

En este proyecto se desarrolla una solución para evaluar experiencias. Primero se crean servicios web que clasifican experiencias en estados emocionales dependiendo del nivel de satisfacción, interés, ... Esto es realizado a través de minería de textos. Se procesa y clasifica información no estructurada (documentos de texto) que representan o describen las experiencias. Se utilizan métodos de aprendizaje supervisado. Esta parte es desarrollada con una arquitectura orientada a servicios (SOA) para asegurar el uso de estándares y que los servicios sean accesibles por cualquier aplicación. Estos servicios son desplegados en un servidor de aplicaciones.

En la segunda parte se desarrolla dos aplicaciones basadas en casos reales. En esta fase *Cloud computing* es clave. Se utiliza una plataforma de desarrollo en línea para crear toda la aplicación incluyendo tablas, objetos, lógica de negocio e interfaces de usuario. Finalmente los servicios de clasificación son integrados a la plataforma asegurando que las experiencias son evaluadas y que las tareas de seguimiento son automáticamente creadas.

Resum

Customer Experience Management (CEM) s'ha convertit en un factor clau per l'èxit de les empreses. CEM gestiona totes les experiències que un client té amb un proveïdor de serveis o productes. És molt important saber com són els sentiments d'un client en cada contacte i així poder suggerir automàticament la pròxima tasca a realitzar, simplificant tasques realitzades per persones.

En aquest projecte es desenvolupa una solució per avaluar experiències. Primer es creen serveis web que classifiquen experiències en estats emocionals dependent del nivell de satisfacció, interès, ... Això és realitzat amb mineria de textos. Es processa i classifica informació no estructurada (documents de text) que representen o descriuen les experiències. S'utilitzen mètodes d'aprenentatge supervisat. Aquesta part és desenvolupada amb una arquitectura orientada a serveis (SOA) per assegurar l'ús d'estàndars i que els serveis siguin accessibles per qualsevol aplicació. Aquests serveis són desplegats en un servidor d'aplicacions.

En la segona part es desenvolupa dos aplicacions basades en casos reals. En aquesta fase *Cloud computing* es clau. S'utilitza una plataforma de desenvolupament en línia para crear tota l'aplicació incloent taules, objets, lògica de negoci i interfícies d'usuari. Finalment els serveis de classificació són integrats a la plataforma assegurant que les experiències són avaluades y que les tasques de seguiment són automàticament creades.

Summary

Customer Experience Management (CEM) is becoming a critical key success factor in today's business. It manages all experiences that a customer has with a supplier of goods or services. It's very important to know what the customer is feeling in every contact and then suggest automatically the next approach to make, simplifying the task for a human operator.

In this project, we develop a solution for evaluating experiences. First we create web services that classify the experiences into emotional states depending on satisfaction, interest,... This is done using Text Mining. We process and classify unstructured information (text documents) that represents or describes experiences. A supervised learning method is used. This part is developed using Service Oriented Architecture (SOA) to ensure that it's based on standards, and the services are accessible from any application. These services are deployed on an application server.

In the second part we develop two applications based on real cases. Cloud computing is key in this stage. An online development platform is used to create all the application including tables, objects, business logic and user interfaces. Finally the classification services are integrated with the platform ensuring that the experiences are evaluated and next tasks are automatically created.

Parte I

Introducción

Capítulo 1

Introducción y objetivos

CEM (*Customer Experience Management*) es la gestión de la experiencia "global" de un cliente con un producto, marca o proveedor. Tiene como objetivo gestionar la valoración, durante todo el ciclo de vida de la relación, de los puntos de contacto entre el cliente y el proveedor: el deseo o necesidad del cliente, el descubrimiento, la atracción, la compra, el uso, la respuesta postventa, la repetición, el fin de la relación o el encadenamiento con el descubrimiento de otros productos. También se encarga de convertir toda esta información en conocimiento necesario para que las empresas mejoren sus procesos de negocio.



Figura 1.1. 1 Customer Experience Management

1.1. Objetivo del proyecto

El objetivo de este proyecto es desarrollar una aplicación capaz de valorar experiencias y gestionar acciones de seguimiento a partir de éstas. Los proveedores podrán automatizar sistemas de seguimiento de las experiencias ocurridas en sus procesos de negocio enlazando sus sistemas informáticos actuales.

El primer objetivo proyecto es la búsqueda, configuración, prueba e integración de una aplicación CRM existente a una empresa real. Esta aplicación debe proveer interfaces potentes y seguras para el acceso a la información de su base de datos. Se necesitarán datos de contactos, empresas, tareas y eventos con los que interactuarán los servicios de valoración y seguimiento. Además, se desarrollarán nuevas tablas y vistas que serán necesarias para cubrir las necesidades funcionales de la empresa en cuestión. No obstante se mantendrá el nivel de abstracción suficiente de los servicios de valoración y seguimiento de experiencias para que se pueda integrar fácilmente con cualquier otra aplicación CRM o ERP.

El segundo objetivo es la creación de los servicios de valoración de experiencias. Se trata de un problema de clasificación. Los métodos de valoración se basarán en datos introducidos por personal de la empresa: transcripciones telefónicas, actas de reunión, faxes, correos electrónicos, ... Primero se obtendrá un conjunto de datos de aprendizaje(*Grand Truth*) con los que se entrenarán los algoritmos de *text mining* necesarios para la valoración de las experiencias. A partir de entonces, cada vez que se genere en la empresa una experiencia y ésta sea introducida en el sistema CRM, se disparará una llamada al servicio de valoración automáticamente.

El tercero y último objetivo es la creación de los servicios de gestión tareas. Estos servicios deben, de una manera flexible y automatizada, ayudar a la toma de decisiones sobre qué tareas se deben realizar ante una valoración emocional de una experiencia que puede ser tanto positiva (captación, recomendación o promoción) como negativa (destrucción).

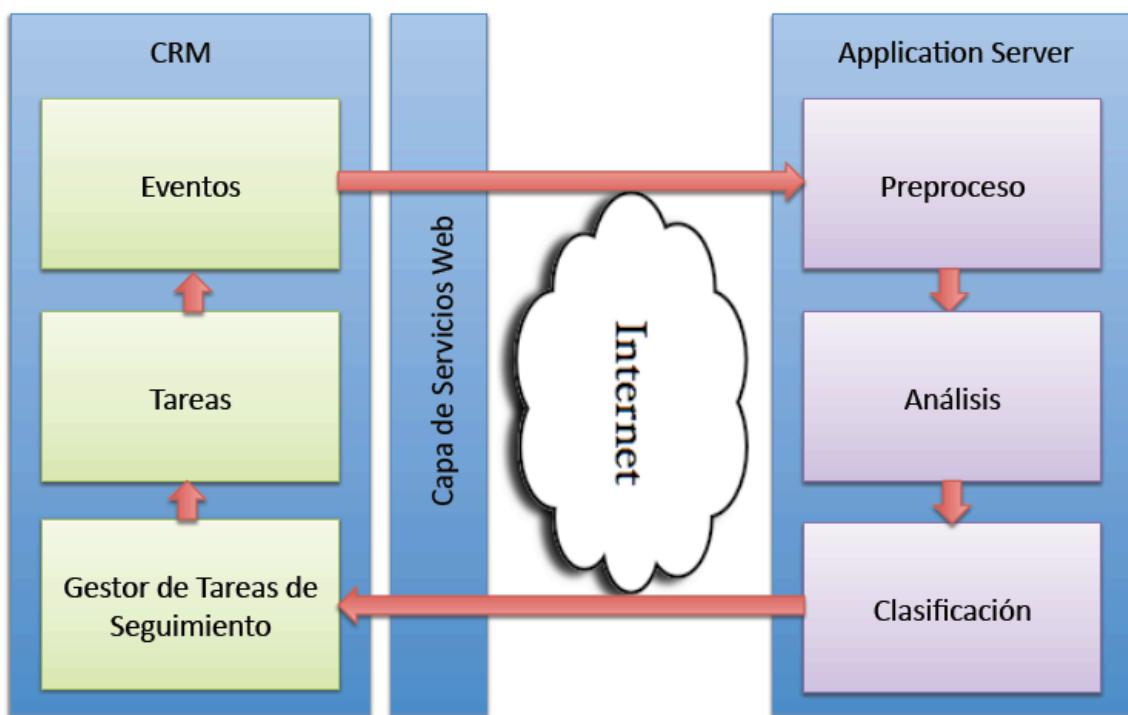


Figura 1.1. 2 Esquema de la solución propuesta

Tanto el segundo como tercero objetivo de este proyecto plantean ciertas dificultades. En ambos es necesario:

- Acceder a información de diferentes aplicaciones de empresa(CRM's, ERP's,...).
- Conectar con servicios disponibles en la Web (Web Services).
- Crear flujos de negocio de manera flexible.
- Permitir que otras aplicaciones accedan a sus servicios independientemente de la plataforma en que se ejecuten.

Para poder solventar estos requerimientos se implementará una arquitectura orientada a servicios. SOA(Service Oriented Architecture) es una arquitectura de sistemas que trata a los procesos de negocio como servicios atómicos, que realizan de manera independiente de los demás(sin estado) una acción muy concreta. Estos servicios pueden ser orquestados para

realizar reglas complejas de negocio. Se basa en lenguajes y protocolos estándares que garantizan la conectividad de diferentes plataformas.

Algunas de las ventajas de esta arquitectura son:

- Apertura a servicios externos.
- Reutilización de servicios
- Flexibilidad de los procesos de negocio
- Mejora del retorno de la inversión en un proyecto de desarrollo como éste.

Se seguirá el modelo de desarrollo Top-Down. Las fases para la creación de los servicios serán las siguientes:

1. **Modelado:** Descripción del servicio, sus métodos, la información que necesita (los argumentos de entrada) y el resultado (la salida). También en esta fase se modelará la interacción con otros servicios para la creación de funciones complejas de negocio. Obtendremos lo que llamaremos 'contratos' para cada servicio, lo que sería programar para la interface en el mundo de los objetos. Esta fase, en el mundo real, debería ser realizada en colaboración con un analista de negocio.
2. **Implementación:** Desarrollo del servicio previamente modelado. Se utilizará la tecnología disponible basada en estándares.
3. **Despliegue:** Se instalará un servidor de aplicaciones con capacidad para operar con servicios que puedan ser orquestados para desplegar los servicios implementados. También se introducirá estos servicios en un diccionario para potenciar la re-utilización de servicios ya implementados. Opcionalmente y en caso de necesidad se utilizará de un ESB(bus de mensajes) para facilitar el acceso a los servicios desde cualquier otro tipo de aplicación.
4. **Monitorización:** Tráfico y comportamiento de los servicios desplegados.

Se creará para cada servicio un SLA(Service Level Agreement) con el que estipularemos la calidad y disponibilidad del servicio. No existe ninguna normativa al respecto, aunque existen organizaciones que están intentando estandarizarlo.

Finalmente se realizarán pruebas integradas del sistema con las cuales valoraremos tanto la fiabilidad y eficiencia de la arquitectura como el acierto obtenido con los métodos de valoración de las experiencias.

1.2 Estado del Arte

El concepto de *Customer Experience* fue acuñado por Pine y Gilmore en 1998 con su artículo en *Business Harvard Review*.

El foco de las empresas actuales deben estar en los clientes, no en la propia organización. Este enfoque hace que la empresa tenga un factor diferenciador, aumentando sus beneficios y su valor.

Los clientes están cada día más preparados y capacitados y además disponen de toda la información que deseen a un clic de distancia. Son clientes muy sofisticados a los cuales, experiencias que se crean deben ser cada vez más memorables. La gestión de la experiencia se convierte en un factor de innovación clave en la empresa.

Todo ello es válido tanto en la relación con clientes particulares como empresas (B2B). Al final, los contactos con la empresa cliente se realizan con personas.

Empresas de éxito ya se benefician de ser las primeras que gestionan la experiencia de sus clientes. Cada vez más empresas se darán cuenta de la importancia de esta gestión y cada vez será más difícil desarrollar experiencias que las diferencie de las demás.

Por ello, es muy importante obtener información muy precisa de la experiencia global de nuestro clientes. El objetivo clave es identificar qué emociones estamos provocando en una experiencia y en qué grado. Se necesita un método para medir y monitorizar la reacción emocional de nuestros clientes.

Existen aplicaciones en el mercado llamadas CRM (*Customer Relation Management*) que gestionan la fuerza de ventas de una empresa. El objetivo de estas aplicaciones es la venta por medio de generación de oportunidades que se consiguen a través de conocer a nuestros clientes por las diferentes formas de contacto que tenemos con ellos, valoramos lo que sabemos de ellos. CEM, en cambio, se centra en lo que sienten.

Ejemplo de CRM es SalesForce. Más de ochenta mil usuarios avalan su éxito. Es una aplicación online que tiene como objetivo automatizar la fuerza de ventas. Ofrece a los representantes de ventas, directivos y ejecutivos herramientas para desarrollar conexiones más sólidas con los clientes, aumentar las tasas de ganancias y cerrar más contratos.

Además cuenta con una plataforma de desarrollo y despliegue de aplicaciones llamada *Force.com*.

Esta plataforma se basa en el modelo de *Cloud Computing*. Dispone de hardware, software y servicios que están disponibles para el uso inmediato ya que, se ejecutan en el *cloud* (Internet). Además el coste de estos servicios es exclusivamente calculado por los recursos que se utilizan.

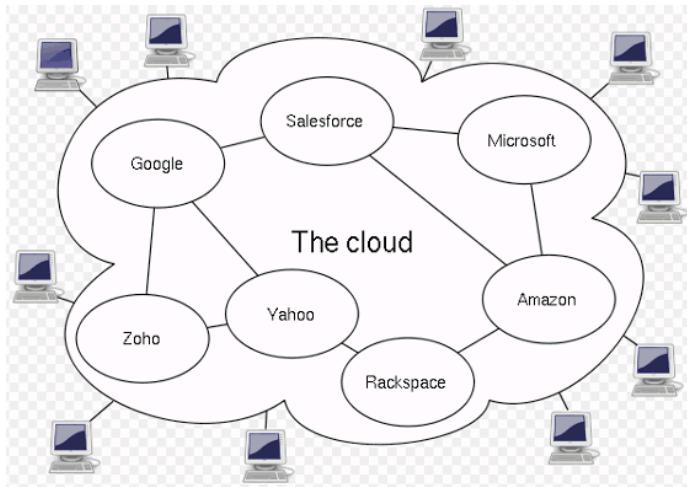


Figura 1.2. 1 Cloud Computing

Cloud Computing está, cada día más, siendo utilizado por empresas de todos los tamaños. Este modelo empieza a reemplazar métodos tradicionales de venta de licencias de aplicaciones. Esto es motivado por unos niveles de productividad, economía y éxito sin precedentes.

Tanto empresas como desarrolladores de software pueden crear y ejecutar aplicaciones personalizadas en Internet sin la necesidad de enfrentarse a altas inversiones, de tiempo y dinero, en desarrollo de software, adquisición de hardware y mantenimiento de ambos.

Force.com pone a disposición del desarrollador la tecnología existente en Salesforce CRM para el desarrollo de aplicaciones empresariales altamente personalizables. Esta plataforma reúne todos los requerimientos del modelo Cloud Computing. Proveedores de software pueden fácilmente desplegar servicios de manera segura, fiable, configurable y personalizable. Además es fácilmente integrable con otras aplicaciones.

Aún así, las aplicaciones desplegadas están centradas, como la mayoría, en las acciones de la empresa y no en la experiencia del cliente. La innovación de este proyecto consiste en agrupar diferentes tecnologías y conocimiento para poder crear un asistente que ayude al seguimiento de acciones y mejore el nivel de satisfacción de las experiencias vividas por los clientes .

1.3. Estudio de viabilidad del proyecto

Introducción

El estudio de viabilidad pretende evaluar las garantías de éxito de la realización de este proyecto. Incluye un estudio que propone una solución conceptual al problema y posibles alternativas ante contingencias. Además se definen requisitos del proyecto, se evalúan los riesgos y finalmente se decide si el proyecto es viable o no.

Tipología y palabras clave

Tipología: Proyecto informático de desarrollo de un prototipo de sistema de software. No aplicable a distribución i basado en investigación.

Palabras clave: CEM, CRM, ERP, SOA, SAAS, cloud, experiencia, emociones, text mining, latent semantic indexing.

Descripción

Implementar una solución basada en arquitectura orientada a servicios y cloud computing que permita la valoración y seguimiento de experiencias de clientes.

Evaluación de riesgos

El objetivo del proyecto es integrar servicios disponibles en el *cloud*(internet) con servicios propios que sean capaces de valorar experiencias en forma de documentos escritos. Para ello puntualizamos los siguientes principales tareas a realizar:

- O1. Buscar una aplicación CRM que pueda proveer interfaces potentes y seguras para el acceso a la información de su base de datos a través de Internet.
- O2. Formarse en aspectos de programación, personalización e implantación de dicha aplicación.
- O3. Buscar empresa o datos reales con los cuales se realizó el proyecto.
- O4. Personalizar la aplicación CRM a los requerimientos de la empresa.
- O5. Importar datos relevantes para las pruebas del prototipo.
- O6. Analizar y entender los diferentes eventos que debemos considerar como experiencias y recoger los datos relevantes para la valoración.
- O7. Modelar los servicios que deben realizar las valoraciones.
- O7.1. Evaluar algoritmos de text mining para la clasificación de las experiencias a través de simulaciones.
- O8. Modelar los servicios que realizan el seguimiento de tareas.
- O9. Implementar los servicios.

O10. Desplegar los servicios.

O.10.1. Instalar y configurar servidor de aplicaciones.

O11. Probar la aplicación de forma integrada y valorar resultados

Situación Actual:

	CRÍTICO	PRIORITARIO	SECUNDARIO	RIESGO	ESTADO
O1	X			MEDIO	REALIZADO
O2	X			MEDIO	REALIZADO
O3		X		ALTO	REALIZADO
O4		X		MEDIO	REALIZADO
O5		X		MEDIO	REALIZADO
O6	X			BAJO	REALIZADO
O7	X			MEDIO	REALIZADO
O7.1.	X			ALTO	REALIZADO
O8	X			BAJO	REALIZADO
O9	X			MEDIO	REALIZADO
O10	X			MEDIO	REALIZADO
O10.1.	X			BAJO	REALIZADO
O11	X			BAJO	REALIZADO

Definiciones, acrònimos y abreviaciones

CRM: Customer Relation Management. Aplicaciones que gestionan fuerzas de venta.

ERP: Enterprise Resource Planning. Aplicaciones que gestionan los recursos de una empresa (gestión comercial, financiera-contable, producción).

CEM: Customer Experience Management. Gestión de las experiencias los clientes.

SOA: Service Oriented Architecture. Arquitectura de sistemas basada en la orquestación de servicios atómicos.

Partes interesadas

Usuario: Aunque no tengamos de antemano el cliente, en caso de no encontrarlo, se creará una figura que lo representará.

Equipo:

Nombre	Descripción	Responsabilidad
Joan Serrat	Director de proyecto	Supervisa, evalúa y orienta el trabajo del alumno.
Rodolfo Guichon	Jefe de proyecto	Gestiona y planifica el proyecto. Presenta los resultados parciales.
Rodolfo Guichon	Arquitecto de sistemas	Define la arquitectura de hardware y software necesarias para la realización del proyecto.
Rodolfo Guichon	Analista	Define el modelo a desarrollar.
Rodolfo Guichon	Programador	Desarrolla y despliega los servicios.
Rodolfo Guichon	Técnico de pruebas	Realiza las pruebas y analiza los resultados.

Producto y documentación del proyecto

Se entregará un prototipo de los servicios desarrollados.

Se elaborará una memoria del proyecto. En este se incluirá un tutorial tanto de la configuración del CRM como de la instalación del servidor de aplicaciones y servicios desplegados.

Contexto actual

Se ha buscado soluciones creadas parecidas y lo único que se ha encontrado son empresas que ofrecen consultoría en estos temas. Normalmente acaban en desarrollos a medida para clientes en concreto y no una solución generalizada.

Por otro lado se ha encontrado trabajos de investigación realizados en la materia. Destacan los que tratan el *text mining* para sistemas de recomendación.

Normativa y legislación

LOPD

Normativa de proyectos de final de carrera de la UAB

Ley de propiedad intelectual

US Patent 4,839,853 (<http://patft.uspto.gov/netacgi/nph-Parser?patentnumber=4839853>)

Requerimientos del sistema

No hay requerimientos específicos para la ejecución del prototipo ya que está desplegado en el *cloud*(Internet).

Alternativas ante contingencias

C1. No encontrar una aplicación CRM que pueda proveer interfaces potentes y seguras para el acceso a la información de su base de datos a través de Internet.

Solución: Desarrollar la parte necesaria de un CRM para poder intergrarse con los servicios.

Consecuencias: Retraso temporal grave en la fase afectada.

C2. No disponer de una empresa o datos reales con los cuales se realizó el proyecto.

Solución: Crear los datos manualmente o a partir de textos encontrados en páginas web de opiniones.

Consecuencias: Dispondríamos posiblemente de un conjunto de prueba más reducido y menos uniforme con lo que nos dificultaría la obtención de resultados concluyentes.

C3. No poder personalizar la aplicación CRM a los requerimientos de la empresa.

Solución: Adaptar los datos reales de la empresa en cuestión a las especificaciones del CRM.

Consecuencias: Retraso temporal medio en la fase afectada.

C4. No poder importar datos relevantes para las pruebas del prototipo.

Solución: Exportar la información a ficheros que texto o excel que podamos leer desde los métodos de entrenamiento de los servicios de text mining.

Consecuencias: Retraso temporal leve en la fase afectada.

C5. Los algoritmos de text mining para la clasificación de las experiencias a través de simulaciones no convergen según los resultados esperados.

Solución: Búsqueda exhaustiva de métodos de clasificación que puedan adaptarse al problema planteado. Se implementará el que mejor resultado consiga y se comentará, en resultados y conclusiones, la eficiencia de este método.

C6. Encontrar limitaciones técnicas del sistema CRM al Implementar los servicios de seguimiento.

Solución: Realizar el seguimiento desde una aplicación creada a tal efecto y que pueda ser consultada por el CRM.

Consecuencias: Retraso temporal medio en la fase afectada.

Conclusiones de viabilidad

Se considera que es posible cumplir con los objetivos en los tiempos previstos. Hay una componente de investigación, que es el *text mining*, que conlleva un cierto riesgo si se intenta conseguir resultados óptimos. Se tratará de implementar métodos ya estudiados y que hayan demostrado resultados aceptables para conseguir la mínima desviación sobre plan temporal creado. El cumplimiento de dicho plan pasa sobretodo por el esfuerzo del equipo que compone este proyecto.

Finalmente, en función de los objetivos anteriormente analizados, se considera este proyecto **viable**.

1.4. Planificación

El proyecto comienza el 13 de octubre del 2010 y tiene como fecha estimada de fin el 31 de abril del 2011. Se realizarán reuniones de seguimiento y aprobación de las diferentes fases semanalmente. El 14 de enero del 2011 se presentará la memoria previa, se solicitará la lectura entre 26 y el 31 de mayo, se entregará la memoria entre el 17 y 22 de junio y se realizará la lectura entre el 27 de junio y el 8 de julio de 2011. (Ver detalle en cap. 4)

Nº	Tareas a realizar	Duración	Previo
1	Asignación y matriculación del proyecto	2h	
2	Planificación	21h	
3	Estudio de viabilidad	20h	1
4	Aprobación del estudio de viabilidad	1h	3
5	Ánalisis de la aplicación	58h	
6	Estudio previo	40h	4
7	Arquitectura del sistema	5h	6
8	Ánalisis de requerimientos	10h	7
9	Ánalisis legal	2h	8
10	Aprobación del análisis	1h	9
11	Diseño de la aplicación	126h	
12	Diseño de las interfaces	20h	10
13	Diseño de las entidades de la aplicación	10h	12
14	Diseño de los servicios	80h	13
15	Diseño de los tests	10h	14
16	Restricciones de software y hardware	2h	15
17	Restricciones de seguridad	2h	16
18	Cumplimientos de normativa y legislación	1h	17
19	Aprobación del diseño	1h	18

20	Desarrollo de la aplicación	121h	
21	Personalización CRM	40h	19
22	Desarrollo Servicios de Valoración	60h	21
23	Desarrollo Servicios de Seguimiento	20h	22
24	Aprobación del desarrollo	1h	23
25	Despliegue de la aplicación	8h	
26	Configuración del servidor de aplicaciones	5h	24
27	Despliegue de los servicios	2h	26
28	Aprobación del despliegue	1h	27
29	Test y monitorización	41h	
30	Tests individuales	16h	28
31	Prueba Integrada	16h	30
32	Prueba de stress	8h	31
33	Aprobación de control de calidad	1	32
34	Elaboración de la memoria	40h	33
35	Presentación y defensa del proyecto	1h	34

Capítulo 2

Fundamentos tecnológicos

2.1. ¿Qué es Cloud Computing?

Introducción

Cloud computing es un modelo de pago por uso para poner a disposición, de manera compartida, recursos computacionales (redes, servidores, almacenamiento, aplicaciones, servicios) bajo demanda. Estos recursos pueden ser dados de alta y de baja con un mínimo esfuerzo o interacción por parte del proveedor. El modelo de nube promueve la disponibilidad y tiene cinco características fundamentales:

- **Autoservicio bajo demanda:** Un consumidor puede tener acceso unilateral a capacidad computacional sin interacción humana con el proveedor de servicios.
- **Heterogeneidad:** Los servicios están disponibles en la red y son accesibles a través de estándares que hacen posible la conexión desde diferentes plataformas de clientes (móviles, portátiles y tabletas).
- **Deslocalización de recursos:** Los recursos puestos a disposición por los proveedores de servicios utilizan el modelo de multipropiedad. Recursos físicos y virtuales son dinámicamente asignados y reasignados según las peticiones del cliente. La mayor parte de las veces el cliente no tiene control o conocimiento de la localización física de los recursos a los que accede.
- **Escalabilidad.** La capacidad computacional provista puede ser rápidamente aumentada o bajada según necesidad. Muchas veces, para el usuario, la capacidad parece infinita. Puede usar cuando y cuanto quiera.
- **Pago por uso:** El coste para los usuarios se rigen por modelos de pago por servicio o en base a publicidad.

Clases de *cloud computing*

Existen diferentes tipos de *cloud computing* según su propiedad y comunidad de usuarios:

- **Nubes privadas:** La infraestructura es propiedad de una organización y utilizada por usuarios pertenecientes a ésta.
- **Nubes comunitarias:** La infraestructura es compartida por muchas organizaciones y soporta una comunidad específica que comparte recursos.
- **Nubes públicas:** La infraestructura es propiedad de una organización y es vendida como servicio al público en general.
- **Nubes híbridas:** La infraestructura está formada por otras nubes, pero está presentada como una única.

Servicios en el Cloud

Los servicios ofrecidos por Cloud computing se puede segmentar en 11 categorías:

- **Storage-as-a-service:** Capacidad de almacenar información que físicamente existe en un servidor remoto pero es vista lógicamente como un recurso local para cualquier aplicación que necesite utilizarla.
- **Database-as-a-service:** Servicio de acceso a bases de datos remotas, compartiéndola con otros usuarios y puestas a disposición como si fuesen locales.
- **Information-as-a-service:** Habilidad de consumir cualquier tipo de información a través de interfaces bien definidas.
- **Process-as-a-service:** Recurso remoto capaz de conectar otros recursos, como servicios y datos, para crear procesos complejos de negocio.
- **Application-as-a-service:** Aplicaciones puestas a disposición del usuario a través de la Web, normalmente a través de navegadores. También es conocido como SaaS (*Software as a Service*).
- **Platform-as-a-service:** Plataforma remota que incluye servicios para el desarrollo de aplicaciones, interfaces y bases de datos. Esta categoría es la utilizada para la realización de parte de este proyecto.
- **Integration-as-a-service:** Servicios de integración capaces de integrar aplicaciones, mediar interfaces y controlar secuencias de procesos.
- **Security-as-a-service:** Habilidad de proveer seguridad en Internet a través de servicios de identificación y validación.
- **Management-as-a-service:** Servicio de gestión de nubes. Tratan tipologías, virtualización, utilización de recursos y disponibilidad.
- **Testing-as-a-service:** Sistemas de pruebas de software y arquitecturas disponibles en forma de servicios remotos.
- **Infrastructure-as-a-service:** Habilidad de acceder a recursos computacionales remotos. Se tiene acceso, normalmente, a una máquina virtual propia.

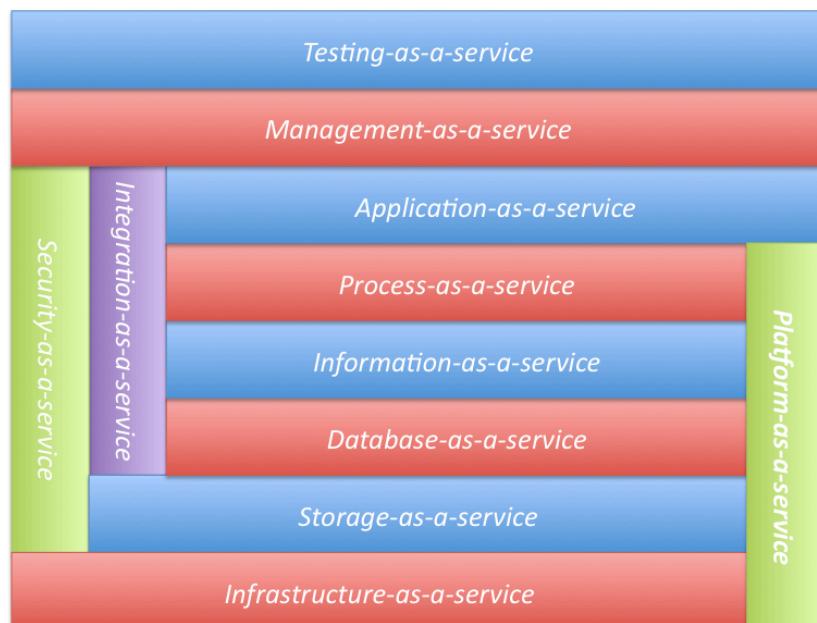


Figura 2.1. 1 Capas de servicios de la nube

2.2. ¿Qué es SOA?

Service Oriented Architecture es una arquitectura de sistemas con la cual se crean y utilizan procesos de negocio a través de la utilización de servicios.

Un servicio es un componente de software y tiene las siguientes propiedades:

1. Está definido por una interfaz que es independiente de la plataforma desde donde se utilice.
2. Es accesible a través de una red.
3. Las operaciones definidas en la interfaz llevan a cabo operaciones de negocio a través de objetos de negocio.
4. Las interfaces e implementaciones pueden ser extendidas en tiempo de ejecución.

Esta arquitectura requiere que los servicios cumplan los siguientes principios:

- **Contratos estandarizados:** Los servicios deben tener contratos que de una forma estándar describan la funcionalidad, la representación de los datos y la política de servicio.

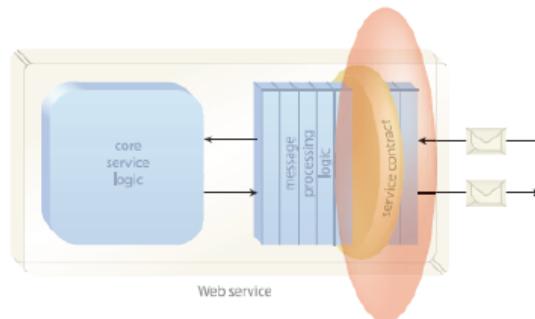


Figura 2.2. 1 Zona de afectación de los contratos

- **Acoplamiento débil:** Para poder maximizar la reutilización los servicios estos deben ser creados con el objetivo de poder interactuar con cualquier otro servicio y viceversa. Es necesario minimizar el nivel de dependencia entre componentes.

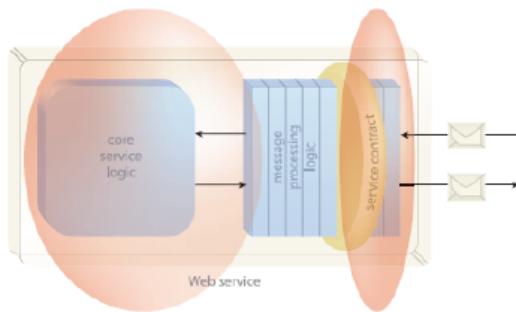


Figura 2.2. 2 Zona de afectación del acoplamiento

- **Abstracción de servicios no esenciales:** Es necesario ocultar la complejidad de la implementación y detalles específicos de los servicios como detalles de programación, sistemas operativos, funciones auxiliares y solo hacer visible las funciones imprescindibles para su uso.

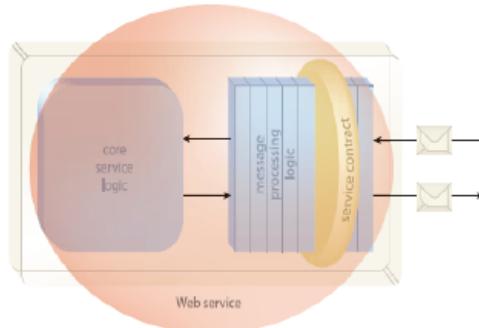


Figura 2.2. 3 Zona de Afectación de la abstracción

- **Reutilización:** La reutilización es un factor clave en la arquitectura. Se deben crear los servicios para que puedan ser utilizados por otros servicios.
- **Autonomía:** Representa el grado de independencia de la implementación de un servicio. Si otros servicios dependen de detalles de implementación y no de la interfaz se creará un alto nivel de dependencia.
- **Minimización de estado:** La escalabilidad depende, en gran medida, de que los servicios mantengan el mínimo estado posible, es decir, no mantener recursos ocupados más allá de su utilización.
- **Localizables:** Debe existir un diccionario para poder descubrir y localizar servicios existentes. Debe contener información suficiente para conseguir este objetivo.

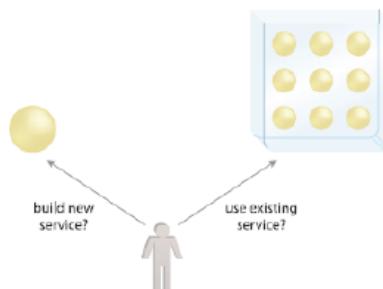


Figura 2.2. 4 Localización de servicios

- **Servicios compuesto:** La capacidad de resolver problemas complejos a partir de servicios que resuelven problemas más pequeños.

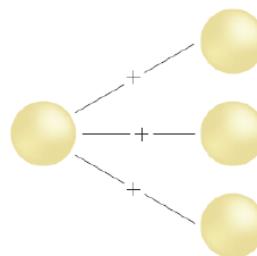


Figura 2.2. 5 Composición de servicios

2.3. Plataforma de desarrollo

Platform-as-a-service, por un lado, indica que se trata de una plataforma que provee la infraestructura necesaria para el desarrollo de aplicaciones de software y, por otro lado, la palabra *Service* indica que esta infraestructura es accesible a través de internet y que su modalidad de pago es por uso. Se han valorado tres posibles plataformas perteneciente a este tipo de Nube que se describen a continuación.

2.3.1. Google App Engine¹

Google App Engine permite crear aplicaciones web en los mismos sistemas escalables con los que funcionan las aplicaciones de Google. Se puede crear, mantener y actualizar estas aplicaciones y aumentar el tráfico y las necesidades de almacenamiento de datos bajo demanda.

Google App Engine admite aplicaciones escritas en varios lenguajes de programación. Gracias al entorno de tiempo de ejecución Java de App Engine se puede crear una aplicación a través de tecnologías Java estándar, que incluyen JVM, servlets Java y el lenguaje de programación Java o cualquier otro lenguaje que utilice un intérprete o compilador basado en JVM como, por ejemplo, JavaScript o Ruby. App Engine también ofrece un entorno de tiempo de ejecución Python dedicado, que incluye un rápido intérprete Python y la biblioteca estándar Python. Los entornos de tiempo de ejecución Java y Python se generan para garantizar que tu aplicación se ejecuta de forma rápida, segura y sin interferencias de otras aplicaciones en el sistema.

Con App Engine, sólo se paga lo que se utiliza. No existen costes de configuración ni tarifas recurrentes. Los recursos que utiliza una aplicación, como por ejemplo el almacenamiento y el ancho de banda, se miden por gigabytes y se facturan según tarifas. Se controla la cantidad máxima de recursos que consume cada aplicación, de modo que siempre permanezcan dentro de cada presupuesto.

App Engine incluye las siguientes funciones:

- Servidor web dinámico, totalmente compatible con las tecnologías web más comunes.
- Almacenamiento permanente con funciones de consulta, orden y transacciones.
- Escalado automático y balanceo de carga.
- API para autenticar usuarios y enviar correo electrónico a través de las cuentas de Google.
- un completo entorno de desarrollo local que simula Google App Engine en el equipo local.
- tareas programadas para activar eventos en momentos determinados y en intervalos regulares.



Figura 2.3.1. 1 Logotipo Google App Engine

¹ Adaptado desde <http://code.google.com/intl/es/appengine/docs/whatisgoogleappengine.html>

2.3.2. Amazon EC2²

Amazon Elastic Compute Cloud (Amazon EC2) es un servicio web que proporciona capacidad informática con tamaño modifiable en la nube. Se ha diseñado con el fin de que la informática web resulte más sencilla a los desarrolladores.

La sencilla interfaz de servicios web de Amazon EC2 permite obtener y configurar capacidad con una fricción mínima. Proporciona un control completo sobre sus recursos informáticos y permite ejecutarse en el entorno informático acreditado de Amazon. Amazon EC2 reduce el tiempo necesario para obtener e iniciar nuevas instancias de servidor a cuestión de minutos, lo que permite escalar con rapidez su capacidad (aumentarla o reducirla) cuando cambien los requisitos informáticos. Amazon EC2 permite pagar sólo por la capacidad que realmente utilice. Amazon EC2 proporciona a los desarrolladores las herramientas necesarias para crear aplicaciones resistentes a errores y para aislarse de los casos de error más comunes.

Amazon EC2 presenta un entorno informático virtual, que permite utilizar interfaces de servicio web para iniciar instancias con distintos sistemas operativos, cargarlas con su entorno de aplicaciones personalizadas, gestionar sus permisos de acceso a la red y ejecutar su imagen utilizando los sistemas que desee. Para utilizar Amazon EC2 se siguen los siguientes pasos:

- Seleccionar una imagen de plantilla preconfigurada para pasar a estar activo de inmediato. O bien crear una AMI (Amazon Machine Image) que contenga sus aplicaciones, bibliotecas, datos y valores de configuración asociados.
- Configurar la seguridad y el acceso a red en su instancia de Amazon EC2.
- Seleccionar los tipos de instancias y sistemas operativos que desee y, a continuación, iniciar, finalizar y supervisar tantas instancias de su AMI como sea necesario, a través de las API de servicio web o la variedad de herramientas de gestión proporcionadas.
- Determinar si desea una ejecución en varias localizaciones, utilizar puntos finales de IP estáticos o adjuntar almacenamiento de bloques continuo a sus instancias.
- Pagar sólo por los recursos que realmente consuma, como las horas de uso de instancias o la transferencia de datos.



Figura 2.3.2. 1 Logotipo Amazon Web Services

² Adaptado desde <http://aws.amazon.com/es/ec2/>

2.3.3. Force.com[8]

Force.com es diferente de las otras plataformas porque está enfocada a aplicaciones de negocios. Force.com es parte de Salesforce.com que empezó como un CRM (Customer Relation Management). Force.com, no obstante, se ha alejado del CRM para poder ofrecer una infraestructura completa para el desarrollo de cualquier aplicación de negocio que combina la capacidad de configuración y codificación. Esta infraestructura es ofrecida como servicio a través de Internet.

Se enumeran a continuación algunas de las características fundamentales de Force.com.

Multipropiedad

Multipropiedad es un concepto abstracto, pero representa un beneficio para los desarrolladores. Los clientes acceden a una infraestructura compartida con metadatos y datos almacenados en la misma base de datos lógica. La multipropiedad implica:

- **Infraestructura compartida:** Todos los clientes comparten la misma infraestructura.
- **Versión única:** Sólo existe una versión de Force.com. La misma plataforma es utilizada para cualquier aplicación y para cualquier número de usuarios.
- **Mejoras continuas a coste cero:** Cuando Force.com es actualizado para incluir mejoras o arreglos, esta acción es transparente para los usuarios y se aprovechan de ella instantáneamente.

Base de datos relacional

El corazón de Force.com es una base de datos relacional provista como servicio. A cada cliente o propietario Force.com provee una base de datos que está integrada con el resto de la plataforma. Se acaban las tareas de configuración, optimización y copias de seguridad necesarias en otras plataformas.

Metadatos declarativos

Casi todas las personalizaciones configuradas o codificadas en Force.com están disponibles a través de contratos de acceso en formato XML que permiten el acceso a través de servicios web. Esto permite la configuración de entornos idénticos o integración con sistemas de control.

Lenguaje de programación

Force.com provee su propio lenguaje de programación, llamada Apex. Este lenguaje permite al desarrollador interactuar con características de la plataforma, incluyendo las interfaces de usuario. Se parece al Java e incluye comandos de acceso a base de datos. Puede ser escrito vía web o a través de un plug-in de Eclipse.

Force.com ofrece cuatro grandes categorías de servicios.

Base de datos

Force.com está construido alrededor de una base de datos relacional. Permite la definición de tablas personalizadas con hasta 500 campos. Cada campo contiene datos fuertemente tipados, puede contener reglas de validación y se guarda el histórico de los cambios realizados.

Se pueden crear esquemas de datos complejos relacionando tablas. Los datos y metadatos están protegidos de borrados accidentales mediante el uso de una papelera de reciclaje. Los datos pueden ser importados vía utilidades de la plataforma o a través de acceso vía servicios web.

Lógica de Negocio

Apex es el lenguaje con el que se crea lógica de negocio en Force.com. Se pueden crear clases e interfaces. Soporta el modelo de programación orientado a objetos. La filosofía de desarrollo basado en pruebas está muy integrado a Force.com. Métodos de pruebas pueden ser automáticamente manejados por Force.com y se pueden obtener información de rendimiento.

Interfaces de usuario

Existen dos formas de crear interfaces de usuario: Plantillas y Visualforce. Las plantillas son creadas a partir de los modelos de datos, incluyendo las validaciones y, a través de un editor, se puede modificar la interfaz.

Visualforce permite al usuario construir interfaces de usuario personalizadas a través del uso de comandos que pueden ser incluidos en una página web propia.

Integración

Force.com soporta la integración con un amplio abanico de tecnologías, la mayoría basados en protocolos y mensajes estándares:

- Apex Web Services permite control remoto de datos, metadatos y procesos a través de servicios web.
- Toda la lógica de negocio programada en Apex puede ser expuesta como servicio web.
- Se puede acceder desde la plataforma a servicios externos.

La plataforma Force.com ha sido la escogida para el desarrollo de parte de este proyecto debido a su afinidad con los objetivos planteados.



Figura 2.3.3. 1 Logotipo Force.com

Parte II

Servicios de clasificación de experiencias

Capítulo 3

MODELADO Y CLASIFICACIÓN DE LAS EMOCIONES

3.1. Introducción

“Cuando un cliente vive una experiencia sublime, quiere repetir. Nuestros clientes no son sólo empresas o personas a las que vender, son a la vez, nuestros vendedores, nuestros consejeros, nuestros socios.”, Colin Shaw.

La mayoría de procesos que las organizaciones usan no están diseñados pensando en el cliente. El foco de las empresas actuales debe estar en los clientes, no en la propia organización. Este enfoque hace que la empresa tenga un factor diferenciador, aumentando sus beneficios y su valor.

Los clientes están cada día más preparados y capacitados y además disponen de toda la información que deseen a un clic de distancia. Son clientes muy sofisticados a los cuales, experiencias que creamos deben ser cada vez más memorables. Todo ello es válido tanto en la relación con clientes particulares como empresas (B2B). Al final, los contactos con la empresa cliente se realizan con personas.

La gestión de la experiencia es un factor de innovación clave en la empresa.

Empresas de éxito ya se benefician de poder gestionar las experiencias de sus clientes. Cada vez más empresas se darán cuenta de la importancia de esta gestión y cada vez será más difícil desarrollar experiencias que las diferencia de las demás. Por ello, es muy importante obtener información muy precisa de la experiencia global de nuestro clientes.

El objetivo clave es identificar qué emociones estamos provocando en una experiencia y en qué grado. Necesitamos un método para medir y monitorizar la reacción emocional de nuestros clientes.

3.2. Parte emocional de las experiencias

Se ha modelado un sistema de cálculo de la parte emocional de la experiencia de un cliente a partir de un método creado por Colin Shaw[1]. Este sistema provee una estimación financiera de la implicación de tener una experiencia emocional en particular. Ha sido diseñado para ser utilizado en procesos de negocio y su eficacia ha sido testada y probada.

Una experiencia es una interacción entre una organización y un cliente. Es la suma del rendimiento de los aspectos físicos de la organización (precio, producto, localización, ...), los sentidos estimulados y las emociones provocadas. Todo ello contrastado con la expectativa del cliente durante toda la interacción.

Se trata de emociones: Las emociones manejan el comportamiento humano.

Debemos empezar por entender el estado emocional de nuestros clientes y controlar tanto la experiencia en si misma como los hechos previos y posteriores que puedan afectar a la misma.

El estado emocional de un cliente viene dado por:

1. Rasgos inherentes: El propio carácter de cada persona.
2. Humor: Estados emocionales duraderos.
3. Emociones: Sentimientos que tienen lugar en un corto periodo de tiempo.

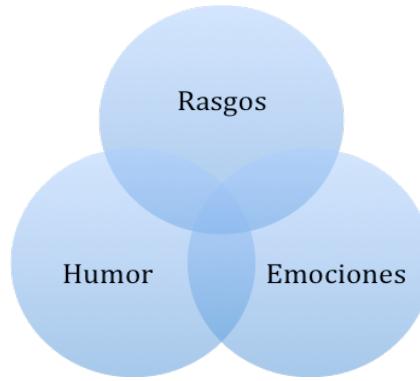


Figura 3.2. 1 Factores que afectan al estado emocional

Cada experiencia tiene una extensión que va más allá de ella misma.

Las fases que definen una experiencia en toda su extensión son:

1. Hechos externos que puedan afectar al humor
2. Experiencia previa: Influencias externas de nuestra experiencia que pueden afectar directamente al resultado de la misma.
3. La Experiencia
4. Experiencia posterior: Hechos no controlados que pueden llegar a hacer parte de la experiencia global.

Toda esta información es relevante en el momento de valorar una experiencia, aunque no siempre se disponen de ella en su globalidad.

Para facilitar la clasificación se han segmentado las emociones en 4 cuadrantes (Fig. 3.2.2):



Figura 3.2. 2 Cuadrantes emocionales

5. Cuadrante de Destrucción

Cuadrante formado por experiencias de clientes no satisfechos en general, es difícil eliminar pero sí se debe controlar. Es necesario tomar acciones para transformar estos clientes insatisfechos a través de nuevas experiencias que le transmitan interés, mejora y rentabilidad.

Ejemplos de experiencias destructivas:

- Entregas no realizadas a tiempo, o de forma incompleta.
- Procesos que llevan mucho más tiempo del que el cliente esperaba.
- Impotencia ante la falta de eficacia.
- Falta de personas de contacto apropiadas.
- Inexistencia de lo que busca el cliente.
- Sentirse ignorado, abandonado.
- No entender o satisfacer las expectativas físicas y emocionales del cliente.
- Traspaso constante de responsabilidades.
- Involucrar a clientes en procesos que sólo benefician la empresa.
- No mantener compromisos.
- Deshonestidad.
- Engaño.

6. Cuadrante de Captación

Experiencias que han provocado un estímulo positivo, provocaron interés. Las personas que vivieron esta experiencia son candidatos a ser clientes, si ya no lo son, y además convertirse en clientes fieles.

Para generar experiencias de captación se debe:

- Crear Novedades.
- Sorprender.
- Hacer participante al cliente en el desarrollo de nuevos productos.
- Entender el perfil de la audiencia.
- Provocar sentimientos de superación.
- Cambiar las experiencias regularmente.
- Control del tiempo.
- Personal que transmita energía.
- Transmitir sentido de valor añadido.

7. Cuadrante de Recomendación

Experiencias que transmiten calidad, seguridad y valor. Transforman clientes puntuales en clientes fieles y que repiten.

Ejemplos de acciones que provocan estas experiencias:

- Demostrar seguridad.
- Escuchar al cliente.
- Anticiparse a las necesidades del cliente.
- Centrar la atención en el cliente.
- Cumplir los compromisos pactados.
- Ser transparente.
- No abrumar al cliente con demasiadas experiencias a la vez.
- Enfocar las acciones hacia las expectativas del cliente.
- Captar toda la atención posible.
- Incentivar la fidelidad.

8. Cuadrante de Promoción

Experiencias que transforman clientes en promotores. Son aliados, fans. Recomiendan, defienden y ayudan a crecer.

Algunas acciones necesarias para lograr estas experiencias son:

- Crear relaciones duraderas con el cliente.
- Conocer emocionalmente al cliente.
- Ir más allá de las expectativas del cliente.

3.3. Clasificación de las experiencias

Las compañías generan un elevadísimo número de experiencias diarias con los clientes. Estas experiencias son generadas por llamadas, correos electrónicos, documentos, etc. Todas las experiencias pueden ser representadas por un documento, ya sea el mismo email, una transcripción de una llamada, una minuta de una reunión, etc. Por ello, se ha desarrollado una herramienta capaz de procesar estos documentos que contienen información no estructurada y hacer una clasificación automática encontrando significado a las palabras. Las ventajas de esta herramienta son el ahorro del tiempo necesarios para el análisis manual de las experiencias y la optimización de los procesos de reacción ante ellas. El objetivo es clasificar los textos de los documentos en los cuatro cuadrantes emocionales.

Los sistemas de clasificación de textos existentes se basan, en su mayoría, en espacios de vectores. Estos vectores representan a los documentos a través de las palabras que contienen. Dada la alta dimensionalidad de estos vectores, se utilizan métodos de reducción de dimensiones como *Latent Semantic Indexing* basado en *Single Value Decomposition* para ello. Este método es parte del sistema de clasificación desarrollado. Se ha planteado un clasificador genérico independiente del origen y del tipo del documento.

Para clasificar los textos ha sido necesario hacer un preproceso y realizar cambios de representación para facilitar la aplicación de métodos de entrenamiento y clasificación.

3.3.1. Preproceso de los documentos[23]

Text Normalization

El primer preproceso que se ha creado ha sido para normalizar los textos. Dado el origen incierto de los documentos, debemos unificar codificación de caracteres, acentos, mayúsculas o minúsculas, posibles faltas de ortografía y contracciones de las palabras utilizadas normalmente en correos electrónicos y mensajes sms. Este proceso se ha denominado *Text Normalization*.

Stop-words Removal

Después de normalizar el documento, el preproceso *Stop-words Removal* elimina del documento símbolos, letras sueltas, signos y palabras que no aportan información relevante al proceso de clasificación. Estos datos, que no aportan información, son enumerados previamente a la implementación y están separados por idioma. Para este proyecto se ha preparado el español y el inglés. Por ejemplo, se han creado enumeraciones de preposiciones candidatas a eliminarse de los documentos:

```
Prepositions_Es = {a, ante, bajo, cabe, con, contra, de, desde, en, entre, hacia, hasta, para, por, según, sin, so, sobre, tras, durante, mediante, pro, vía}
```

```
Prepositions_En = {a, about, above, across, after, again, against,
```

all, almost, alone, along, already, also, in, at, ago, before, between, by, during, from, to, till, until, past, since, within, among, around, behind, below, beside, close, down, front, of, inside, into, near, next, off, onto, opposite, out, outside, over, round, through, towards, under, up}

Stemmer

Se unifican palabras que puedan tener el mismo significado. Se ha creado un preproceso llamado *Stemmer*: unificación de palabras de una misma familia léxica o etimológica. Las palabras pertenecientes a una misma familia léxica comparten un mismo lexema o raíz, que es la parte que les da significado. Cada palabra puede contener prefijos o sufijos. En este caso lo que realmente interesa es quedarse con el lexema. Se ha utilizado para el inglés el *Porter Stemmer* para los documentos en inglés y una adaptación para el español. La tabla 6.2.1 muestra algunos ejemplos.

Palabra	Lexema	Morfemas		Prefijo	Sufijo	Incremento
		género	número			
literarios	liter	o	s		ari	
memorias	memori	a	s			
señorita	señor		a		it	
desesperanza	esper			des	anza	
periodismo	period	o			ism	
esclavitud	esclav				itud	
quemadura	quem	a			ur	ad

Figura 3.3.1. 1 Raíces, morfemas, sufijos, prefijos e incrementos

Semantic Relation

WordNet es una extensa base de datos léxica en inglés que agrupa substantivos, verbos, adjetivos y adverbios en conjuntos de sinónimos cognitivos llamados *synsets*. A través del uso de esta base de datos, se analizan las palabras para determinar su categoría morfológica (substantivo, verbo, adjetivo o adverbio) y además, es posible realizar un análisis semántico para tratar:

- **Homónimos:** palabras que tienen el mismo significado pero distinta etimología, por tanto, distinto significado.
- **Polisemias:** una misma palabra o signo lingüístico tiene varias acepciones.
- **Sinónimos:** palabras iguales que tienen un significado similar o idéntico entre sí, y pertenecen a la misma categoría gramatical.
- **Hipónimos:** palabras que poseen todos los rasgos semánticos, o semas, de otra más general, su hiperónimo, pero que añade en su definición otros rasgos semánticos que la diferencian de la segunda.

N-Grams

Se ha creado un preproceso que crea una representación de gramáticas N-Gram. Se crean conjuntos de N palabras consecutivas formadas a partir de deslizar una ventana sobre las palabras de los documentos. Ejemplo:

- **Secuencia de palabras:** N=2, texto = “es la parte de la palabra” → (“es”, “la”), (“la”, “parte”), (“parte”, “de”), (“de”, “la”), (“la”, “palabra”).

Estas secuencias pueden agregar información relevante para la clasificación.

Summarization

Es la tarea que tiene como objetivo crear una versión reducida del texto original. Trata con frases completas y su espectro es todo el documento. Se ha implementado teniendo en cuenta la semántica de las palabras y la posición de las frases. A través de una valoración por pesos se preservan las frases que superen un cierto umbral. Estos pesos valoran el contenido semántico de las frases, su localización en el texto, su reiteración, frases explicativas, ...

Segmentation (no realizado)

El objetivo de esta tarea es poder segmentar los documentos para obtener frases con contenidos similares. Los pasos que se siguen son:

1. División el texto en frases.
2. Análisis cada frase para obtener estructura sintáctica y morfológica.
3. Resolución de anáforas pronominales.
4. Cálculo de similitud entre frases.
5. Segmentación y etiquetado del texto.

3.3.2. Cambio de Representación

Bag-of-Words

Para poder clasificar los documentos ha sido necesario obtener una representación de la información que permitiera realizar operaciones matemáticas necesarias para la clasificación. En la representación *Bag-of-Words* cada palabra está representada por una variable separada que contiene un valor numérico. Se ha representado como una matriz donde las filas son palabras y las columnas documentos (ver fig. 3.3.2.1).

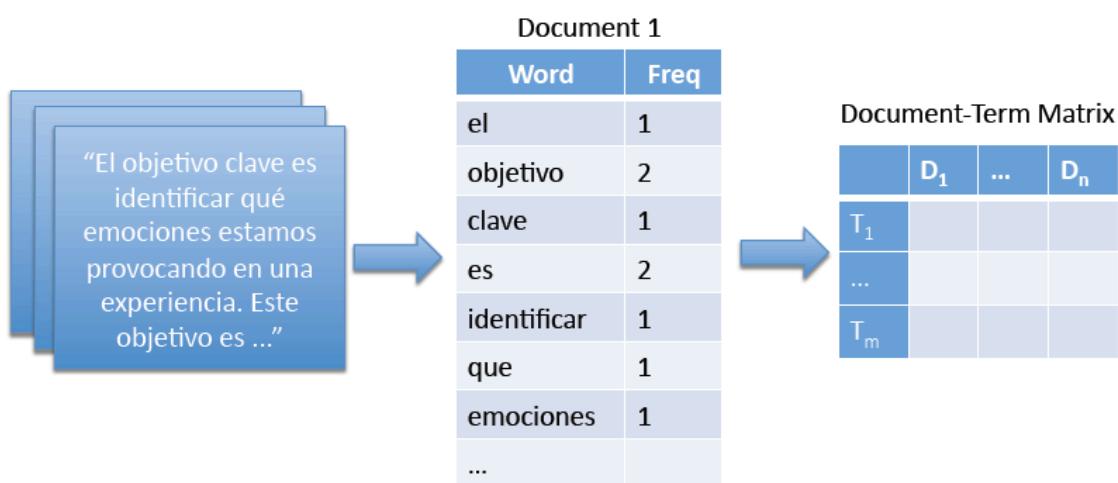


Tabla 3.3.1 1 Obtención de la matriz Documento-Palabra

Este valor representa la importancia relativa que tiene una palabra en un documento y puede ser la frecuencia de aparición de cada palabra en el texto o la frecuencia normalizada:

$$tfidf(w) = tf * \log\left(\frac{N}{df(w)}\right)$$

Donde:

$tf(w)$ es la frecuencia de cada palabra en un documento.

$df(w)$ es la frecuencia de una palabra en todos los documentos.

N es el número de documentos.

Feature Subset Selection

La misión de este proceso es seleccionar las mejores características, es decir, aquellas que realmente aporten información a la clasificación. Esta característica provee al sistema la capacidad de introducción de conocimiento previo. Se pueden cambiar la importancia de las palabras con esta función cambiando la matriz documentos-palabras.

Latent Semantic Indexing (US Patent 4.839.853)

Se ha utilizado *Latent Semantic Indexing* (LSI)[22] como técnica estadística para reducir dimensiones y encontrar estructuras ocultas de contenido semántico de los documentos. LSI es una técnica de proceso de lenguaje natural. Con ella se ha analizado la relaciones entre documentos y palabras que éstos contienen. El proceso previo a éste, llamado *Bag-of-Words*, crea una matriz de forma que las filas son las palabras y las columnas los documentos. Los valores de esta matriz, como se ha comentado anteriormente, son las frecuencias de aparición (normalizadas o no) de las palabras en los documentos. Esta matriz crea el espacio semántico original.

Esta matriz, dada la cantidad de posibles palabras que pueden ser utilizadas en los documentos, tiene un tamaño demasiado grande que hace la inviable su utilización para el proceso de clasificación. LSI es capaz de encontrar una aproximación de menor dimensionalidad. Esto lo consigue mediante *Singular Value Decomposition* (SVD). SVD permite la modificación del espacio semántico de tal manera que refleja los principales patrones asociativos de la información y descarta aquella que es redundante o poco influyente en la clasificación.

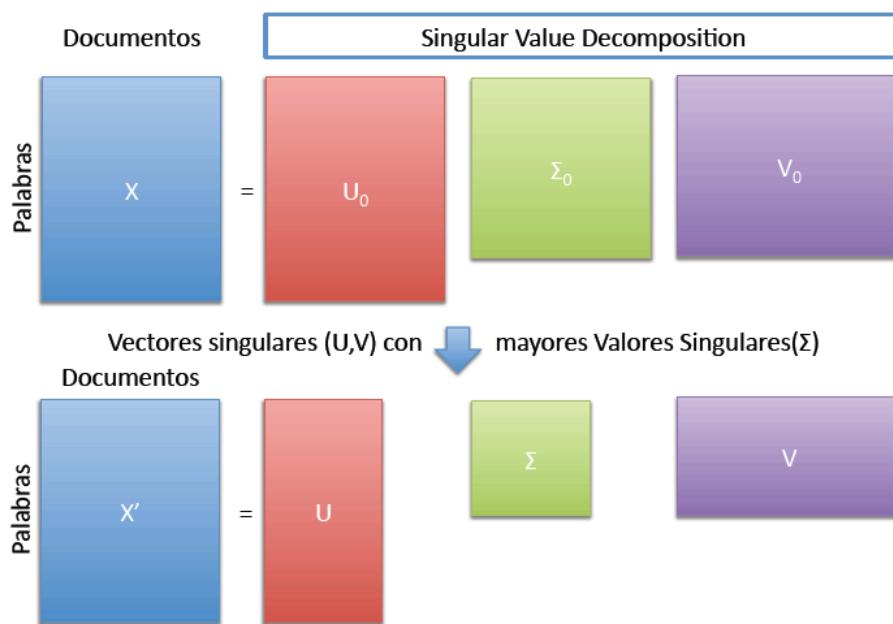


Fig. 3.3.1.2 Latent Semantic Analysis

Dada una matriz X donde cada elemento (i,j) representa la frecuencia de aparición de la palabra i en el documento j .

Con el uso de *Singular Value Decomposition* [22] se consigue una descomposición de X tal que U y V son matrices ortogonales y Σ es una matriz diagonal:

$$X = U \Sigma V^T$$

Las correlaciones de palabras XX^T y de documentos $X^T X$ derivan en:

$$XX^T = (U\Sigma V^T)(U\Sigma V^T)^T = U\Sigma\Sigma^T U$$

$$X^T X = (U\Sigma V^T)^T (U\Sigma V^T) = V\Sigma^T V$$

Como $\Sigma\Sigma^T$ y $\Sigma^T\Sigma$ son diagonales, U contiene los vectores propios de XX^T mientras V los de $X^T X$. $\Sigma\Sigma^T$ y $\Sigma^T\Sigma$ contienen los mismos valores propios, con lo que la descomposición finalmente resulta en:

$$\begin{bmatrix} x_1^1 & \dots & x_1^n \\ \vdots & \ddots & \vdots \\ x_m^1 & \dots & x_m^n \end{bmatrix} = \begin{bmatrix} u_1^1 & \dots & u_1^k \\ \vdots & \ddots & \vdots \\ u_m^1 & \dots & u_m^k \end{bmatrix} \cdot \begin{bmatrix} \sigma_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \sigma_k \end{bmatrix} \cdot \begin{bmatrix} v_1^1 & \dots & v_1^n \\ \vdots & \ddots & \vdots \\ v_k^1 & \dots & v_k^n \end{bmatrix}$$

donde $\sigma_1 \dots \sigma_k$ son valores singulares. Los vectores columna $[u_1^1 \dots u_m^1]^T \dots [u_1^k \dots u_m^k]^T$ y los vectores $[v_1^1 \dots v_1^n] \dots [v_k^1 \dots v_k^n]$ son, a su vez, vectores singulares.

Se selecciona los k mayores valores singulares y los correspondientes vectores singulares de U y V. Se consigue así, un aproximación a X de rango k con el mínimo error (Norma de Frobenius). Los vectores singulares seleccionados formarán un nuevo espacio semántico que aproxima X tal que

$$X_k = U_k \Sigma_k V_k^T$$

La matriz de vectores singulares U_k crea un espacio semántico con el cual se puede segmentar y clasificar las palabras. Por otro lado V_k permite la segmentación y clasificación de documentos. Para poder clasificar primero se debe proyectar el nuevo documento (o palabra) en el espacio semántico

$$d_j = U_k \Sigma_k \hat{d}_j \Rightarrow \hat{d}_j = \Sigma_k^{-1} U_k^T d_j$$

$$t_i^T = \hat{t}_i^T \Sigma_k V_k^T \Rightarrow \hat{t}_i = \Sigma_k^{-1} V_k^T t_i$$

donde \hat{d}_j es el nuevo documento proyectado y \hat{t}_i es la palabra proyectada.

3.3.3. Clasificación

La clasificación que se ha realizado se hizo a partir de un conjunto de entrenamiento supervisado. Por ello, se ha utilizado un método que calcula, a partir de las características conocidas, la clase más similar (más cercana) en el espacio de características.

Es decir, se trata de tener una forma rápida de calcular el caso más parecido, por esta razón se ha usado el algoritmo del vecino más próximo, en particular el k-NN (k-nearest neighbor algorithm). En este algoritmo cada elemento es clasificado según la clase de sus k vecinos más cercanos ponderada por el grado de similitud obtenido (ver Fig. 3.3.3.1).

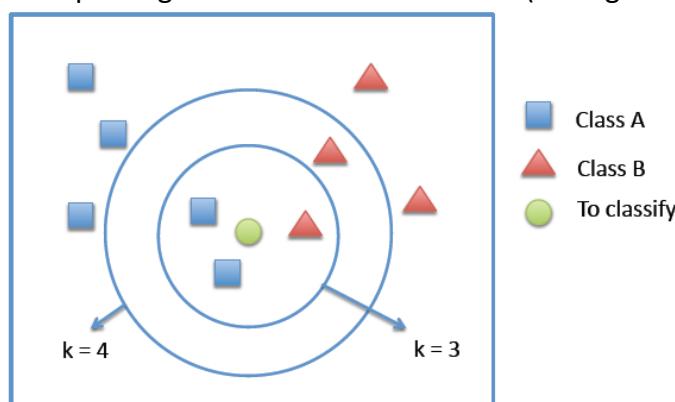


Fig. 3.3.2 1 k-nearest Neighbor

Se ha utilizado similitud del coseno para valorar el grado de similitud entre elementos del espacio de características. Dados dos vectores A y B, la similitud del coseno θ se define como:

$$\text{simcosinus} = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i \times B_i}{\sqrt{\sum_{i=1}^n (A_i)^2} \times \sqrt{\sum_{i=1}^n (B_i)^2}}$$

Los valores de similitud pueden variar de -1 (completamente opuesto) a 1 (exactamente el mismo). El valor cero indica independencia y los valores intermedios grados de similitud o diferencia.

Capítulo 4

Desarrollo de los servicios de clasificación

4.1. Introducción

Se ha desarrollado un sistema de clasificación de experiencias basado en una arquitectura orientada a servicios(SOA). Se ha utilizado UML para diseñar la solución SOA con el objetivo de garantizar la consecución de nuestros requerimientos.

La potencia de SOA reside en la habilidad de aumentar la agilidad empresarial mediante la integración y la reutilización de los procesos de negocio. Esto se consigue de dos maneras: por un lado estimulando la creación de soluciones basadas en servicios reutilizables que encapsulen capacidades funcionales ajenas a su implementación y por otro aportando herramientas que puedan gestionar relaciones entre capacidades funcionales[15].

El ciclo de vida de este proyecto se ha dividido en cuatro fases: Modelado, Ensamblado, Despliegue y Gestión. Se han modelado los procesos, se han ensamblado los componentes, se ha desplegado la aplicación compuesta y se ha gestionado su funcionamiento.

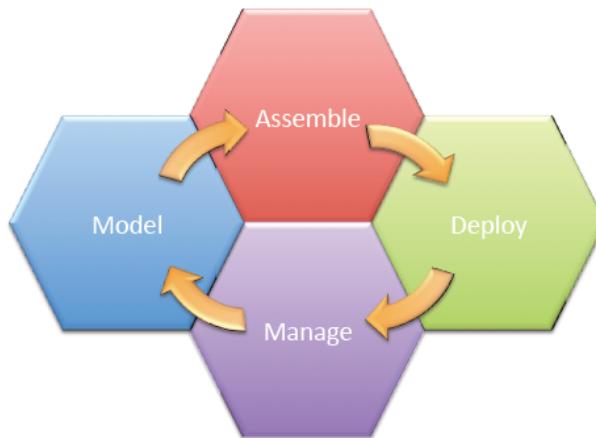


Figura 4.1.1 Metodología de desarrollo SOA

Una vez el modelo de negocio estaba implementado y desplegado, se ha podido cambiar la implementación de alguna de las tareas fácilmente. La flexibilidad es uno de los beneficios claves de SOA.

Para diseñar una solución SOA se necesita:

Establecer requerimientos de negocio

Identificar los servicios.

Diseñar las interfaces.

Decidir como serán implementados (Identificar Proveedores de servicios).

Componer los servicios que representan a los procesos complejos de negocio.

Los perfiles han participado en esta fase del proyecto son:

Business Executive: Determina objetivos y metas de negocio.

Business Analyst: Analiza los requerimientos de negocio. En nuestro caso tenemos la figura de un especialista en *Text Mining*.

Arquitecto de sistemas: Diseña la arquitectura de la solución

Programador: Implementa la solución.

4.2. Modelado

Modelar es necesario para unir los requerimientos iniciales de negocio a la solución final basada en servicios.

El grado de abstracción con el que se ha trabajado en esta fase es grande para que se pueda centrar toda la atención en los servicios de negocio. Se han formalizado los requerimientos de negocio para el correcto desarrollo de los servicios con la intención alcanzar los objetivos deseados.

El desarrollo basado en modelado ha ayudado a conseguir estos objetivos. Los modelos permitieron abstraer los detalles de la implementación y focalizar la atención en temas que atan a las decisiones arquitecturales.

El criterio que se ha seguido cumple con unos de los principios fundamentales de SOA para el desarrollo de soluciones SOA: separación de responsabilidades (*concerns*) y acoplamiento débil (*loose coupling*).

Los “analistas de negocio” han sido los responsables por definir los requerimientos organizacionales y operacionales necesarios para conseguir los objetivos y metas de negocio. Ha sido necesario tener una visión empresarial. No ha sido necesario, en cambio, tener conocimientos técnicos informáticos como reutilización, cohesión, acoplamiento de servicios, distribución, seguridad, persistencia, integridad o concurrencia.

Se ha empezado con la descripción de las motivaciones y objetivos que se deseaban lograr. Después se han capturado los procesos que expresan los requerimientos organizacionales y operacionales necesarios. Estas motivaciones y los procesos han establecido el contexto para identificar servicios que se necesitaban crear cumplir con los requerimientos de negocio.

Este proceso ha sido lo suficientemente completo y detallado como para ser utilizado como base del SLA (Service Level Agreement).

Requerimientos de Negocio:

Escenario: Desarrollo de un sistema capaz de clasificar experiencias a partir de textos que contengan servicios de negocio reutilizables.

Motivaciones:

- Establecer un medio común para que varias aplicaciones empresariales puedan valorar experiencias de clientes.
- Aumentar la satisfacción de los clientes con una reacción inmediata ante diferentes experiencias.
- Fidelizar los clientes.
- Convertir los clientes en nuestros promotores.

Identificación de los servicios:

Se han identificado dos procesos: el proceso de entrenamiento y el proceso de clasificación. Los dos están formados por servicios independientes que en algunos casos son reutilizados. La figura Fig. 4.2.1 se muestra el modelo creado para el proceso de entrenamiento. Se puede observar que los servicios han sido separados en cinco grupos: *Word Level*, *Sentence Level*, *Document Level*, *Linked Document Collection Level* y *Application Level*. Esta separación atiende a la lógica de los procesos de *text mining*. Además nos agrupa y simplifica el desarrollo de los servicios.

El proceso de entrenamiento empieza con un parámetro de entrada de tipo *Docs*. Se trata de un conjunto de documentos. Las tareas de *Text Normalization*, *Stop-words Removal*, *Stemmer*, *Lexical Relation*, *N-grams*, *Summarization* y *Segmentation* reciben este conjunto de documentos, los preprocesa a nivel de texto y retornan el mismo tipo de dato.

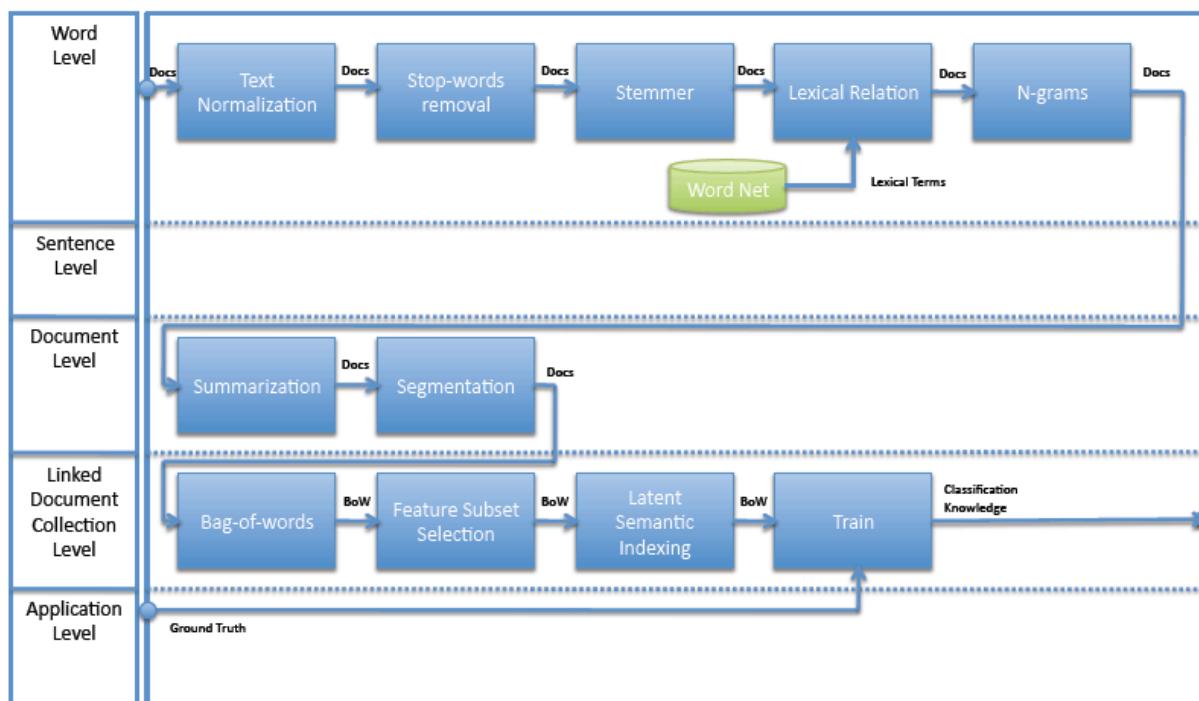


Fig. 4.2. 1 Modelo de Proceso de Entrenamiento

La tarea *Bag-of-words* en cambio, recibe el parámetro de tipo *Docs* y transforma el conjunto de documentos en un dato de tipo *BoW*. Un dato de tipo *BoW* contiene una matriz de características extraídas de los documentos y el diccionario de palabras utilizado. *Feature Subset Selection* y *Latent Semantic Indexing* son utilizados para reducir la dimensión de la matriz de características. También trabajan con el tipo de dato *BoW*.

Finalmente *Train* recibe el dato de tipo *BoW* (la matriz y el diccionario) y un dato de tipo *GroundTruth* (necesario para un aprendizaje supervisado). Los procesa y devuelve un dato de tipo *ClassificationKnowledge* que contiene toda la información de entrenamiento necesaria para el posterior proceso de clasificación.

La figura Fig. 4.2.2 nos muestra el modelo del proceso de clasificación. Este proceso es capaz de clasificar un conjunto de documentos, por lo cual aprovechamos el tipo de dato ya creado en el proceso de entrenamiento *Docs*.

El preprocess de los documentos, la creación de la matriz de características y la reducción de dimensiones son servicios reutilizados del proceso de entrenamiento.

La tarea *Classification* es la responsable de realizar la clasificación de los documentos. Utiliza la matriz de características y la información de entrenamiento necesaria (*ClassificationKnowledge*) y retorna un tipo de dato *ClassificationResult* que es una matriz con los resultados de la clasificación.

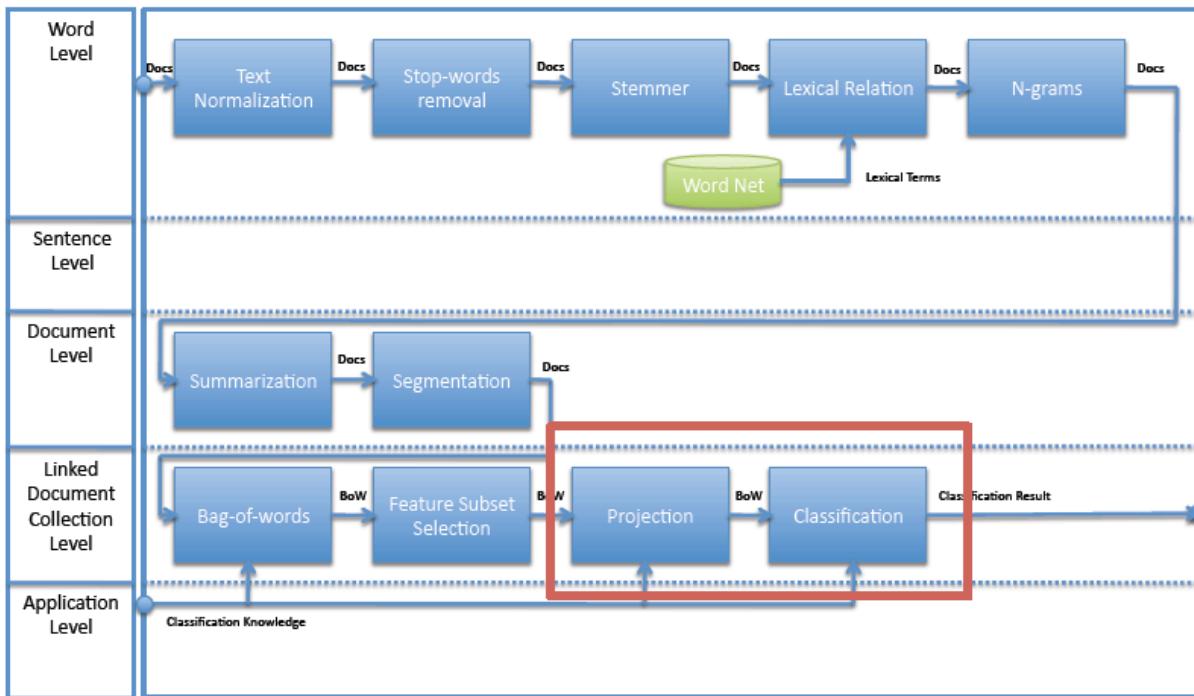


Fig. 4.2.2 Modelo del Proceso de Clasificación

4.3. Ensamblado

Después de haber entendido los requerimientos de negocio se ha procedido al diseño de las interfaces y desarrollo de los servicios que atienden a éstos.

Se han creado implementaciones de servicios que cumplieran con los objetivos deseados y además con una arquitectura que permitiera la reutilización y agilidad de los procesos de negocio.

En esta fase el arquitecto y el programador han tenido un papel fundamental. Han sido los responsables por desarrollar los diferentes servicios para componer el proceso de negocio al completo.

A partir del Modelo de Procesos que fue creado en la fase anterior, se crea el Diagrama de Ensamblado. Este diagrama contiene componentes que corresponden a los servicios del proceso. Cada componentes puede ser un servicio Web, un procesos BPEL, una tarea que requiera intervención humana, una regla de negocio, EJBs, POJOs, JMSs o otras aplicaciones a las que se accede por medio de un conector o un ESB. Se pueden integrar aplicaciones de diferentes plataformas, tecnología y lenguajes de programación. En nuestro caso todos los servicios que se han creado son servicios Web.

Diseño de las interfaces:

El Modelo de Procesos, creado en la fase anterior, ha sido considerado un Contrato de Requerimiento de Servicios. Este modelo indica los grupos de servicios, las tareas que deben realizar y las reglas de interacción entre ellos. Se ha desarrollados las interfaces a partir de este contrato. En la figura 4.3.1 se muestra el Contrato de Requerimiento de Servicios en UML obtenido a partir del modelo.

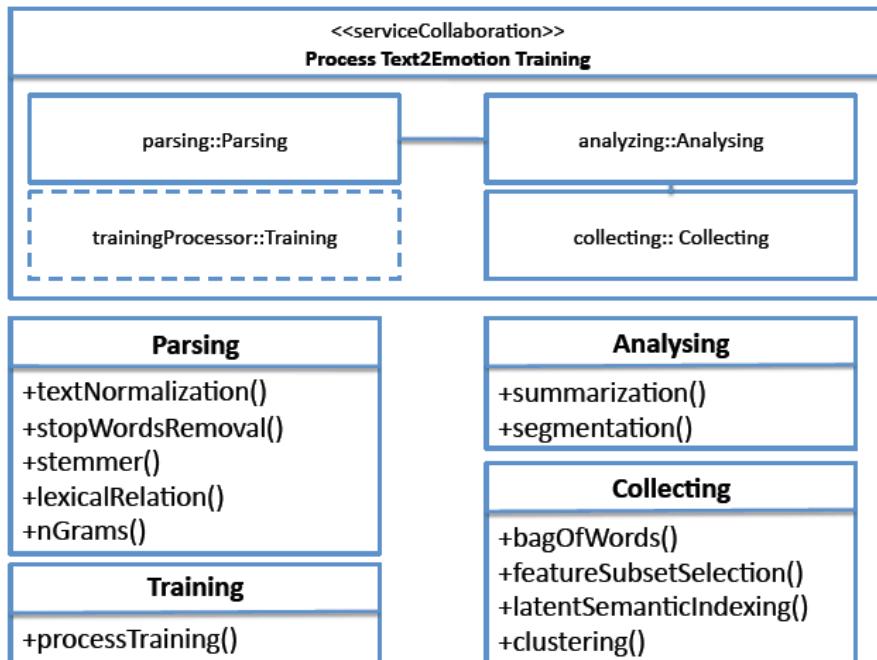


Fig. 4.3. 1 Contrato de Requerimiento de Servicios

El término <<serviceCollaboration>> indica una colaboración entre servicios que corresponden al proceso de entrenamiento. Esta colaboración enmarca un contrato con las siguientes “cláusulas”:

Motivo de la colaboración: procesar el entrenamiento.

Participantes: Los cuatro servicios que participan (*Word-Level* → *Parsing*, *Document-Level* → *Analysing*, *Linked-Document-Collection-Level* → *Collecting* y el propio proceso *Training*).

Responsabilidades: Cada grupo es responsable de una serie de tareas.

Parsing: textNormalization, stopWordsRemoval, stemmer, lexicalRelation, nGrams.

Analysing: summarization, segmentation.

Collecting: bagOfWords, featureSubsetSelection, latentSemanticIndexing, clustering.

Interacción entre grupos: Establece la dependencia entre los grupos (*Parsing* → *Analysing* → *Collecting*).

A partir del modelo del proceso hemos extraído la implementación de la composición de los servicios, es decir la versión BPEL del proceso cómo de puede ver en la Fig. 4.3.2

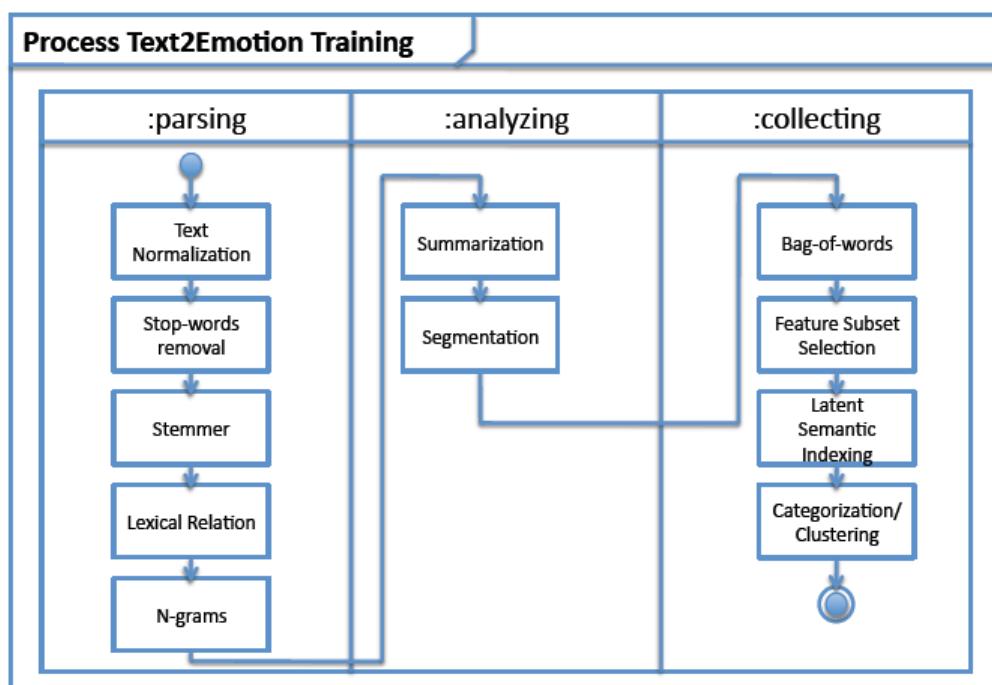


Fig. 4.3. 2 Contrato de Requerimiento de Servicios

Se ha dividido el problema en diferentes unidades funcionales. Se han utilizado *packages* para ayudar a reducir la complejidad, establecer espacios de nombres, facilitar la reutilización y separar las funciones. En la Fig. 4.3.3 se muestra el diagrama de unidades funcionales.

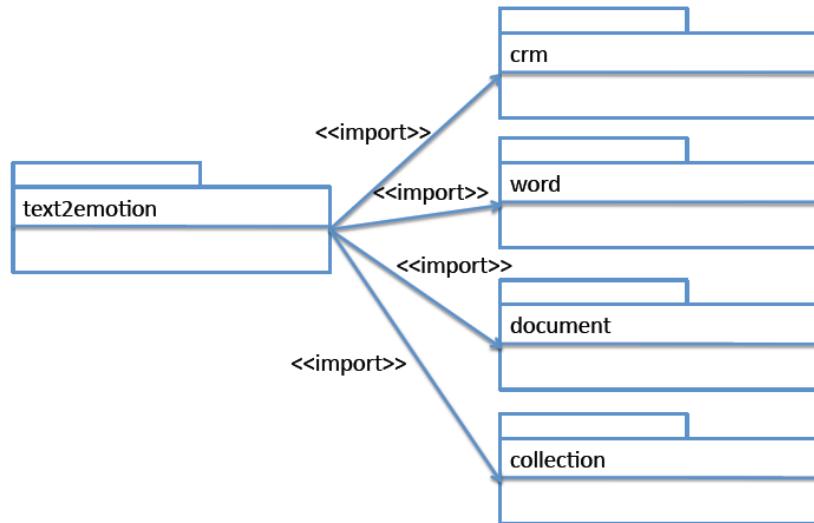


Fig. 4.3. 3 Diagrama de Unidades Funcionales

Cuando se identificaron los servicios en la fase anterior no se han detallado los servicios ni tampoco cómo interactúan. Sólo se ha hecho un esquema de cómo deberían relacionar.

La figura Fig.4.3.4 muestra el Diagrama de Topología de Servicios. Representa las unidades funcionales y los servicios que fueron asignados a estos *packages*. El uso de dependencias en este diagrama tiene como objetivo enseñar cómo se relacionan unos servicios con otros y es de gran importancia para empezar a identificar el uso y el acoplamiento de diferentes sistemas que deben trabajar conjuntamente.

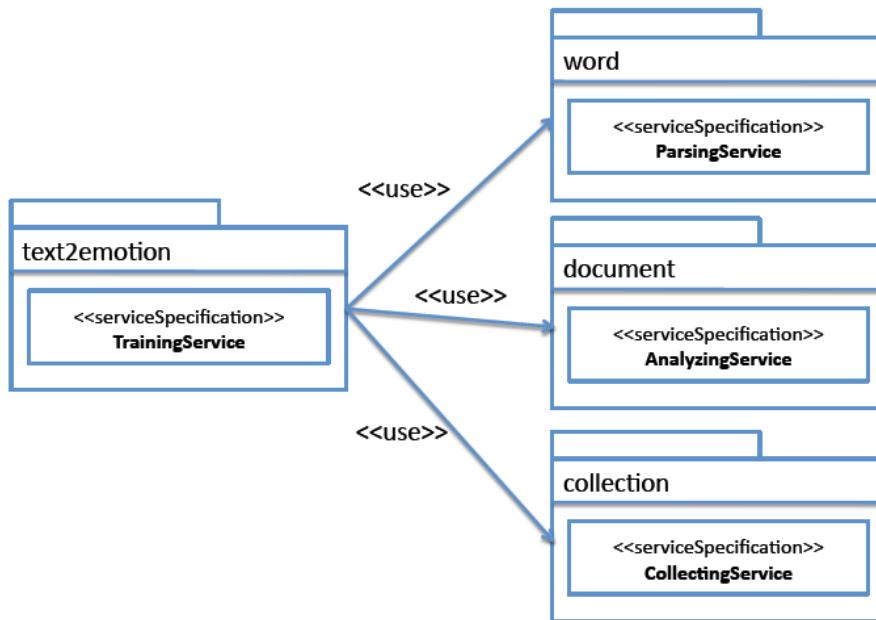


Fig. 4.3. 4 Diagrama de Topología de Servicios

Para completar la identificación de los servicios, la figura 4.3.5 muestra un componente abstracto que modela el contexto para el proceso de entrenamiento. Este diagrama muestra el uso potencial de cada especificación de servicio en un contexto en particular. Las partes son de tipo abstracto ya que son instancias de las especificaciones de servicios y no instancias de los proveedores de servicios que los implementan.

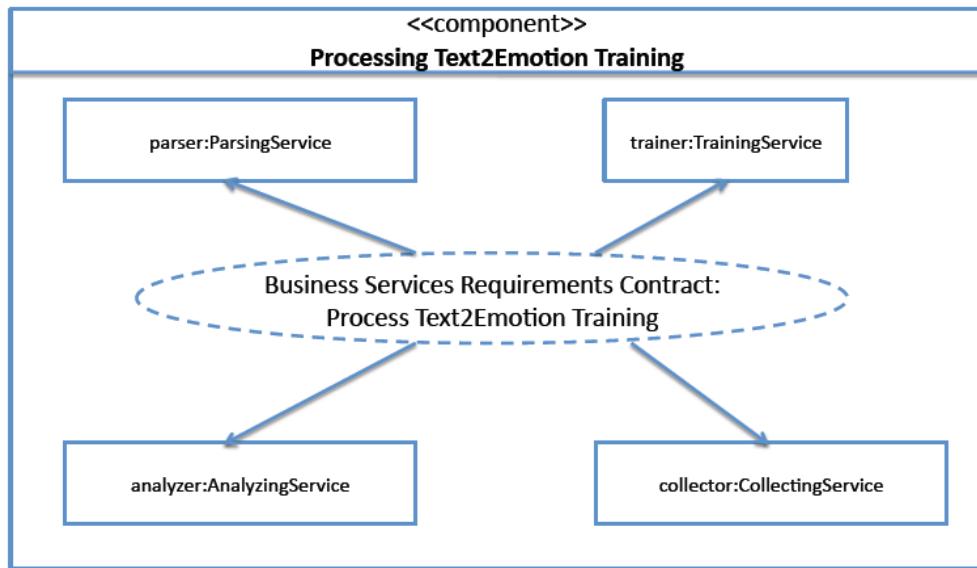


Fig. 4.3. 5 Diagrama Componente Abstracto

Especificación de los servicios:

La especificación de un servicio debe contener toda la información que un potencial consumidor debe conocer para decidir si le interesa utilizarlo y cómo debe hacerlo. También debe incluir todo lo necesario para la implementación del servicio. Es un contrato entre el consumidor y el proveedor del servicio.

Se han elaborado las especificaciones a partir de las interfaces del Diagrama de Topología de Servicios (Fig.4.3.4). Estas especificaciones proveen la siguiente información:

- El nombre del servicio, indicando que hace.
- Las interfaces requeridas y expuestas, describiendo las capacidades funcionales del servicio. Cada capacidad funcional debe incluir:
 - Un nombre: normalmente un verbo que indica lo que hace.
 - Datos de entrada y salida, obligatorios o opcionales.
 - Acciones previas y posteriores al uso del servicio.
 - Excepciones y condiciones de error que puedan ser lanzadas.
- Protocolos de comunicación necesarios.
- Capacidad computacional necesaria para consumir el servicio
- Requerimientos exigidos en la implementación
- Calidad del servicio: coste, disponibilidad, rendimiento, ...
- Políticas de uso: seguridad e integridad.

Las especificaciones de los servicios de este proyecto, al tratarse de un prototipo, se simplificaron a las exigencias definidas en el plan de viabilidad. Todos los servicios involucrados contienen una serie de funcionalidades que pueden ejecutarse opcionalmente y de forma independiente.

Servicio Parsing:

El servicio *Parsing* ha sido definido en el paquete *word*. Como se puede observar en la figura Fig 4.3.6 provee cinco operaciones:

1. Normalizar los documentos.
2. Eliminar símbolos o palabras que no tengan sentido para el entrenamiento.
3. Eliminar raíces de las palabras.
4. Extraer información léxica de las palabras.
5. Crear conjuntos de palabras.

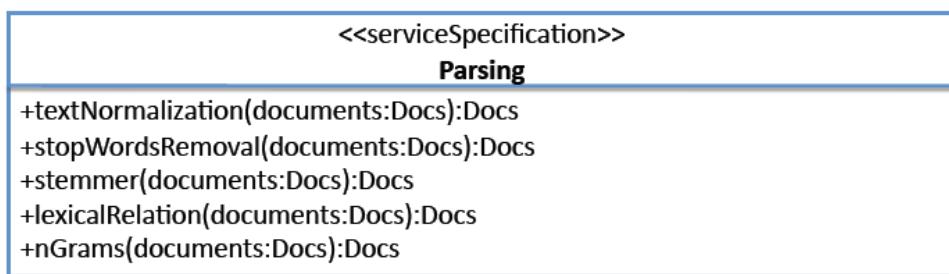


Fig. 4.3. 6 Especificación: Parsing

Servicio Analysing:

El servicio *Analysing*, definido en el paquete *document* está compuesto por dos operaciones: una funcionalidad capaz de resumir el texto y otra capaz de segmentar el texto en frases y eliminar redundancia. Su diagrama se muestra en la figura Fig. 4.3.7

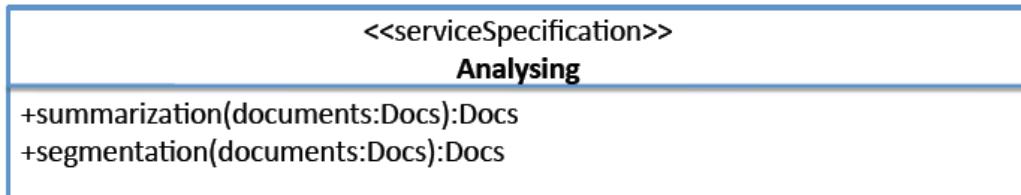


Fig. 4.3. 7 Especificación: Analysing

Servicio Collecting:

El servicio *Collecting* es el responsable de transformar los documentos en conocimiento para la clasificación. Está formado por las siguientes operaciones:

1. Creación de *Bag-of-Words*
2. Selección de subespacio de características.

3. Reducción de la matriz palabra-documento.
4. Segmentación del espacio de características.

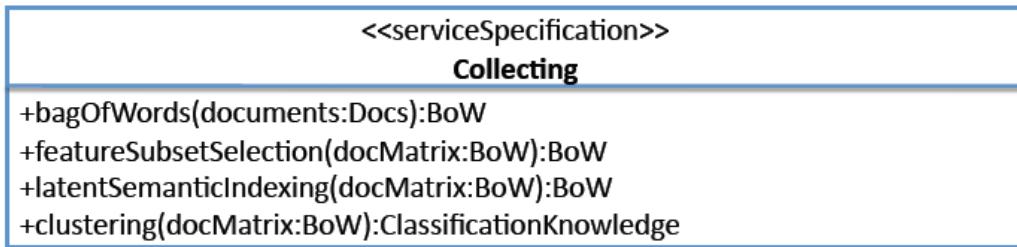


Fig. 4.3. 8 Especificación: Collecting

Servicio *Training*:

Esta es la especificación del servicio que da entrada al proceso de entrenamiento. Sólo tiene una operación.

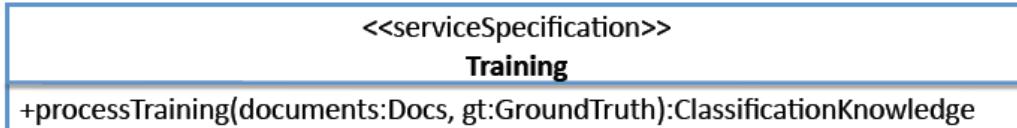


Fig. 4.3. 9 Especificación: Training

Servicio *Classifying*:

Esta es la especificación del servicio que da entrada al proceso de clasificación. Sólo tiene una operación.



Figura 4.3 10 Especificación: Classifying

Modelo de Datos de Servicios:

Los Datos de Servicios es la información que intercambian los consumidores y proveedores de servicios. Cómo se observa en la figura Fig. 4.3.11 se han utilizado en este sistema cinco clases diferentes, todas etiquetadas como mensajes (<<message>> indica la utilización de este tipo de dato para la comunicación con y entre los servicios).

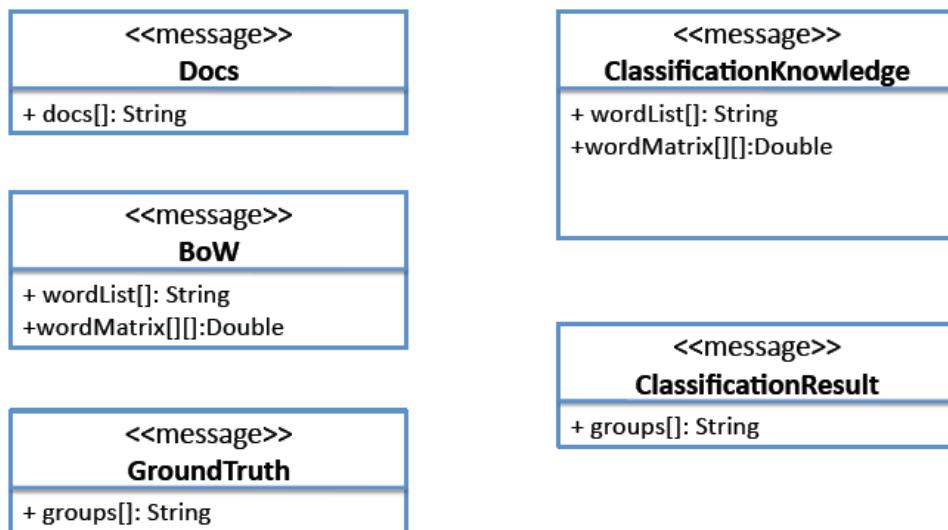


Fig. 4.3. 11 Service Data Model

- Docs: Contiene el conjunto de documentos de texto a analizar.
- BoW: Contiene una lista de palabras y una matriz numérica asociada a estas palabras y los documentos.
- ClassificationKnowledge: Contiene una lista de palabras y una matriz numérica asociada a estas palabras y los documentos para la posterior clasificación.
- ClassificationResult: Matriz con los resultados de la clasificación.
- GroundTruth: Lista de la clasificación real del conjunto de entrenamiento.

Proveedores de Servicios:

Un servicio define un conjunto de funcionalidades puestos a disposición por el proveedor de servicios. La siguiente tareas que se ha realizado ha sido el determinar cuales eran estos proveedores de servicios. Podríamos haber implementado todo el sistema en un solo servicio. Pero de esta manera todos los clientes dependerían de un solo servicio y cualquier cambio afectaría a todos. También esta decisión podría tener efectos sobre el rendimiento, la disponibilidad y la seguridad de la aplicación.

Se han creado cuatro proveedores de servicios para el sistema de valoración de experiencias. Tres correspondientes a los servicios de *Parsing*, *Analysing* y *Collecting*. Cómo podemos observar en la figura Fig. 4.3.12 los tres proveedores proveen la implementación de las interfaces que representan. No requieren, según el diseño, ninguna otra implementación externa para poderse ejecutar.

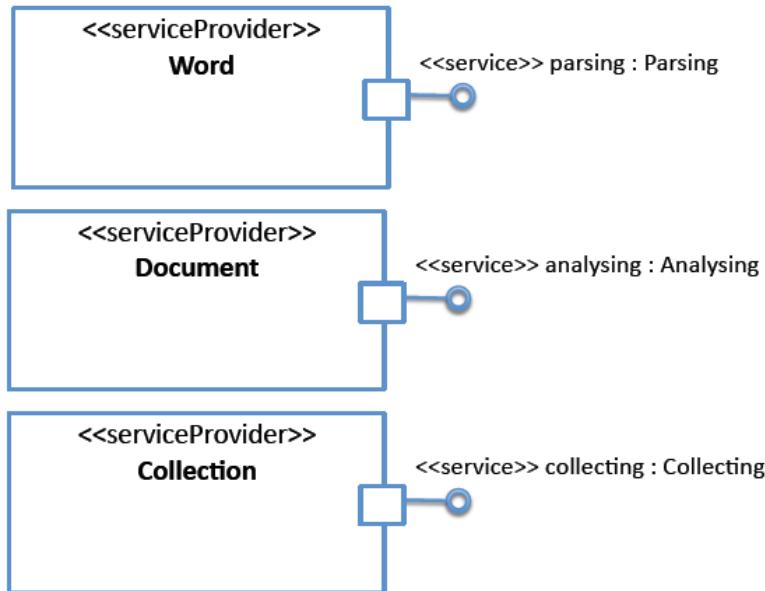


Fig. 4.3.12 Service Providers

El otro proveedor de servicios provee los servicios de *Training* y *Classifying* (ver Fig 4.3.13). Este proveedor contiene la implementación de los servicios de entrenamiento y clasificación y requiere para su funcionamiento de implementaciones de los servicios de *Parsing*, *Analysing* y *Collecting* vistos anteriormente.

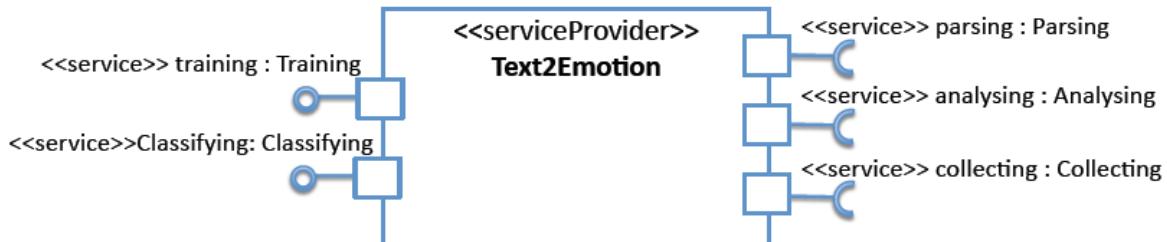


Fig. 4.3.13 Text2Emotion Service Provider

Implementar los servicios de esta manera nos permite flexibilizar la implementación y el despliegue de éstos, ya que cada proveedor de servicios puede estar implementado en diferentes plataformas y en diferentes lugares.

Composición de servicios:

Una colaboración de servicios se define como un conjunto de servicios actuando conjuntamente de una manera acordada según una especificación de proceso.

Para ello se han ensamblado componentes a partir de la composición de servicios con el objetivo de alcanzar los requerimientos originales. Con la “orquestación” de instancias de los servicios de *Parsing*, *Analysing* y *Collecting* se han creado componentes con los servicios de *Training* y *Classifying*.

En las especificaciones de los servicios los servicios de entrenamiento y clasificación contienen las siguientes operaciones:

Training:

+processTraining(documents:Docs, gt:GroundTruth):ClassificationKnowledge

Classifying:

```
+processClassifying(documents:Docs, ck:ClassificationKnowledge):ClassificationResult
```

Estas funcionalidades son ofrecidas por el proveedor de servicios Text2Emotion. Text2Emotion es un componente que provee una interface por cada servicio. Cualquier interacción con estos servicios se realiza a través de sus puertos correspondientes.

El último paso ha sido crear es Subsistema de Entrenamiento y el de Clasificación. Este Subsistema utiliza los Proveedores de Servicios que se han implementado y ensambla las partes creando una solución lista para el despliegue y ejecución.

Estos Subsistemas consisten en instancias de componentes de los proveedores de servicios Text2Emotion, Word, Document y Collection. En la figura 4.3.14 se muestra el subsistema al completo.

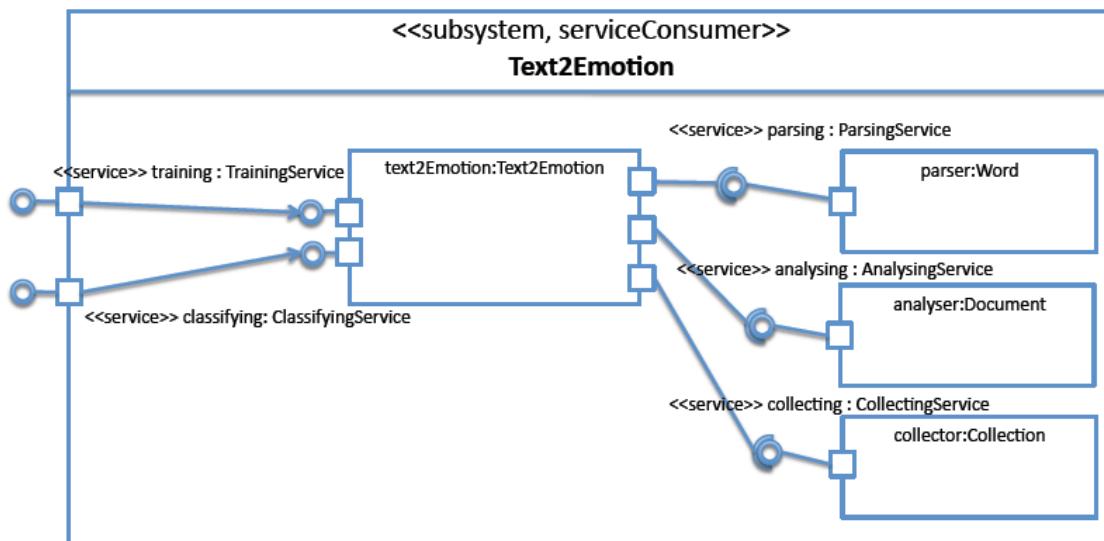


Figura 4.3 14 Subsistema Text2Emotion

4.4. Despliegue

Una vez modelados y ensamblados los servicios, el paso final ha sido el despliegue. Esta fase pone los servicios desarrollados a disposición de sus posibles clientes a través de un servidor de aplicaciones. En el momento de seleccionar el servidor de aplicaciones es importante considerar tres características muy importante que influirán el funcionamiento de los servicios:

- Rendimiento
- Seguridad
- Escalabilidad
- Basado en estándares

También se han considerado otras capacidades:

- Capacidad para tratar con datos y procesos.
- Posibilidad de trabajar con recursos externos.
- Capacidad de administración.
- Capacidad de controlar la seguridad.
- Control de versiones.

Se ha desplegado los servicios con Glassfish. Glassfish es un servidor de aplicaciones compatible con Java EE, de código abierto y está preparado para trabajar en entornos de producción. Este servidor de aplicaciones incluye toda la tecnología necesaria para el desarrollo de esta parte de este proyecto.

Como se muestra en la figura 4.4.1 se ha instanciado un dominio en donde están desplegados los servicios. Este dominio responde a una dirección IP y un puerto, es este caso el 8080.

```
rodolfo-guichons-macbook:bin rodolfovichon$ ./asadmin start-domain domain1
Waiting for DAS to start .....
Started domain: domain1
Domain location: /workbench/glassfishv3/glassfish/domains/domain1
Log file: /workbench/glassfishv3/glassfish/domains/domain1/logs/server.log
Admin port for the domain: 4848
Command start-domain executed successfully.
```

Figura 4.4.1 Levantamiento de la instancia del servidor de aplicaciones

La consola de administración se accede a través de cualquier navegador a través del puerto 4848. Con esta consola podemos administrar usuarios, recursos y aplicaciones desplegadas.

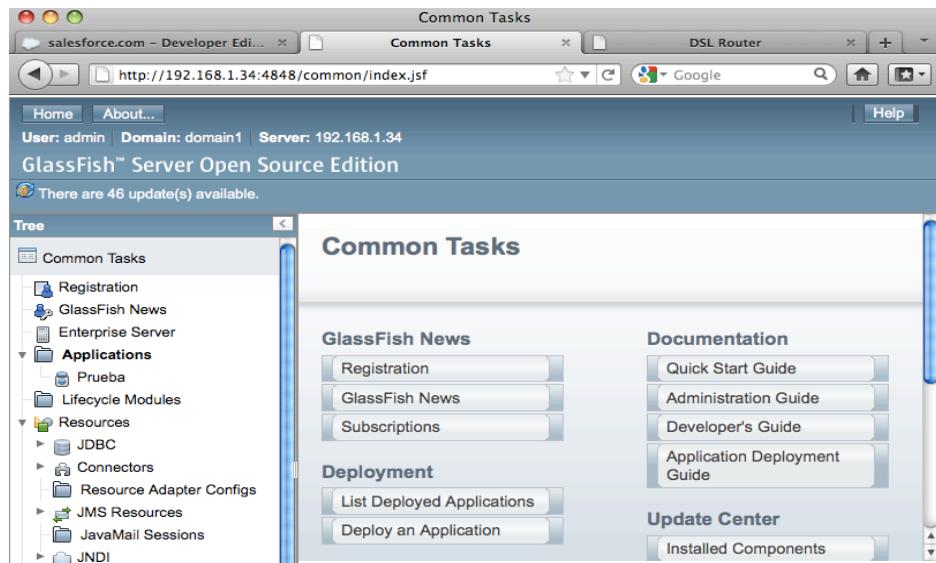


Figura 4.4.2 Consola del servidor de aplicaciones

La figura 4.4.3 muestran el despliegue con éxito de los servicios. En este momento es servidor ya está escuchando posibles peticiones de acceso a este servicio.

```
INFO: WS00018: Webservice Endpoint deployed
Text2EmotionService listening at address at http://localhost:8080/Text2EmotionProject/Text2EmotionServiceService
INFO: Loading application Text2EmotionProject at /Text2EmotionProject
INFO: Loading application Text2EmotionProject at /Text2EmotionProject
INFO: Loading application Text2EmotionProject at /Text2EmotionProject
INFO: Text2EmotionProject was successfully deployed in 268 milliseconds.
```

Figura 4.4.3 Despliegue de los servicios Text2Emotion

Accediendo a página <http://localhost:8080/Text2EmotionProject> se pueden visualizar el servicio desplegado con su contrato (fichero wsdl). Este contrato será necesario para que las aplicaciones creadas posteriormente puedan conocer y acceder a este servicio.

Endpoint	Information
Service Name: {http://service.text2emotion.com}Text2EmotionService Port Name: {http://service.text2emotion.com}Text2EmotionServicePort	Address: http://localhost:8080/Text2EmotionProject/Text2EmotionServiceService WSDL: http://localhost:8080/Text2EmotionProject/Text2EmotionServiceService?wsdl Implementation class: com.text2emotion.service.Text2EmotionService

Figura 4.4.4 Web del servicio desplegado

Parte III

Casos de estudio

Capítulo 5

Caso 1: Departamento comercial de una industria química

En la última fase de este proyecto se ha desarrollado dos prototipos de soluciones adaptadas a dos entornos reales: el departamento comercial de una empresa química y un *Community Management* basado en un portal de opiniones sobre hoteles. Ambas se han desarrollado con el uso de plataformas disponibles en Internet (*cloud computing*) e integradas con aplicaciones orientadas a servicio (Text2Emotion).

5.1. Introducción

Se ha creado un nuevo entorno de trabajo para el departamento comercial de una empresa química. Esta empresa cuenta con la siguiente estructura departamental:

- **Operadores de telemarketing:** realizan constantemente llamadas a posibles nuevos clientes. Su argumento de venta es la historia de la empresa, su variedad y calidad de productos y la buena relación calidad-precio. El objetivo de la llamada es despertar el interés de los clientes y así poder enviarles información detallada, marcar visitas con un comercial, enviar muestras o cotizaciones de productos, etc.
- **Comerciales:** están repartidos por zonas geográficas. Su misión es visitar a clientes y provocar nuevas ventas.
- **Director Comercial:** es el responsable de la organización y de los objetivos del departamento comercial. Reporta directamente a gerencia.

Actualmente los operadores de telemarketing son los responsables de valorar sus propias tareas y hacer el seguimiento de éstas, debido a la gran cantidad de llamadas que se realizan al día. Todas las llamadas son registradas en un sistema CRM.

La propuesta ha sido optimizar la toma de decisiones mejorando por un lado el entorno de trabajo y por el otro, con la utilización de la inteligencia artificial y el procesamiento del lenguaje natural, la automatización de la clasificación de las experiencias creadas a través de las acciones comerciales de esta empresa.

Se ha creado un prototipo de aplicación funcional en Force.com teniendo en cuenta cuales eran las necesidades de esta empresa, qué tipos de datos tratan, qué procedimientos son necesarios, etc. Se ha aprovechado toda la infraestructura de la plataforma y se ha desarrollado sobre ella. Se ha integrado la solución *text2emotion* con la plataforma. El objetivo ha sido valorar cada experiencia y tomar decisiones y acciones automáticamente. Con este prototipo se pretende:

- Organizar de mejor forma la información.

- Facilitar el acceso al sistema a operadores, comerciales y responsables.
- Rebajar costes de mantenimiento de equipos y programas.
- Mejorar la eficiencia de los procesos comerciales.
- Aumentar las ventas.

5.2. Acción y reacción: creando tareas a partir de valoraciones de experiencias

Durante el levantamiento de datos que se ha realizado en la empresa, se han podido distinguir cinco eventos básicos que se realizan en el departamento comercial:

- Llamadas: entrantes o salientes.
- Visitas: concertada o puerta fría
- Reuniones: presentación o trabajo.
- Correos electrónicos.
- Faxes.

Cada una de estos cinco eventos representa un punto de contacto con el cliente, y por lo tanto, una experiencia. Se ha analizado la información contenida en cada uno de ellos para seleccionar cual era relevante para la clasificación, la posterior toma de decisiones.

La toma de decisión, provocada por una experiencia, es una tarea creada y asignada a una persona automáticamente para asegurar su seguimiento. Esto se realiza teniendo en cuenta la siguiente información:

- Usuario que originó la experiencia.
- Cliente que vivió la experiencia.
- Tipo de experiencia (llamada, email, fax, visita o reunión).
- Resultado de clasificación de la experiencia (Destrucción, Captación).

A partir de esta información se puede automatizar el seguimiento conociendo:

- Persona que debe realizar el seguimiento.
- Tipo de tarea a realizar.
- El intervalo de tiempo recomendado para su realización.

La creación de estas tareas no necesitan interacción del usuario. Se realizan automáticamente a través de disparadores. Cuando un usuario genera un nuevo evento, se ejecuta un disparador asociado a este tipo de registro que llama al servicio de clasificación de emociones *text2emotion*. Éste retorna el resultado de la clasificación y una rutina desarrollada en la plataforma es la encargada de crear la nueva tarea de seguimiento. Todo esto se hace de forma asíncrona y transparente para el usuario. En la fig. 5.2.1 se muestra un recordatorio para la realización de una tarea.

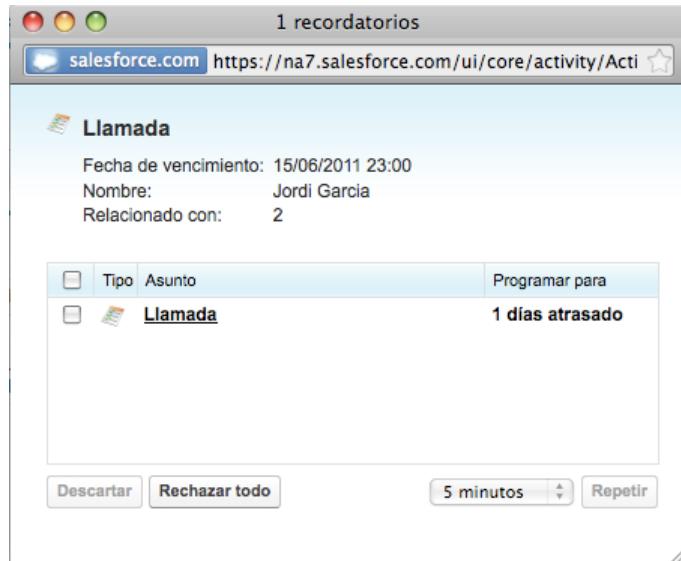


Figura 5.2 1 Recordatorio de tarea a realizar

5.3. Creación del entorno de trabajo en Force.com

5.3.1. Aplicación

Se ha creado una nueva aplicación en la plataforma Force.com, con una serie de fichas para aportar la funcionalidad necesaria a la aplicación. Aunque la plataforma ya provee algunas aplicaciones esto ha sido necesario para adaptarse a la forma de trabajo de la empresa en cuestión. En la Fig.5.3.1.1 de detalle de la definición de la aplicación.

Detalle de Aplicación personalizada	
Etiqueta de aplicación	MICRM
Nombre de aplicación	MICRM
Logotipo	
Descripción	Customización CRM
creada por	RODOLFO GUICHON, 17/11/2010 9:23
modificada por	RODOLFO GUICHON, 17/11/2010 11:55
Fichas incluidas	Inicio Empresas Contactos Llamadas Email Visitas Reuniones Fax Asuntos Paises
Ficha de inicio predeterminada	Inicio

Figura 5.3.1 1 Definición de aplicación

Las fichas corresponden a los diferentes eventos que se deben tener en cuenta para la gestión de la información de experiencias con los clientes. También se han incluido algunas fichas para el mantenimiento de información auxiliar, pero necesaria para la operativa. A cada ficha se vincula un objeto personalizado que se explica a continuación. La fig. 5.3.1.2 muestra la definición de la ficha de Llamadas.

Llamadas

Complete los campos siguientes para definir la ficha personalizada.

Modificar Definición de la ficha personalizada

Información de la ficha de objeto personalizado |= Información obligatoria

Etiqueta de ficha	Llamadas
Objeto	<u>Llamada</u>
Estilo de la ficha	 

(Opcional) Seleccione un vínculo personalizado a la página de inicio para mostrarla como página de bienvenida la primera vez que los usuarios hagan clic en esta ficha.

Vínculo personalizado a la página de bienvenida	--Ninguno-- 
---	---

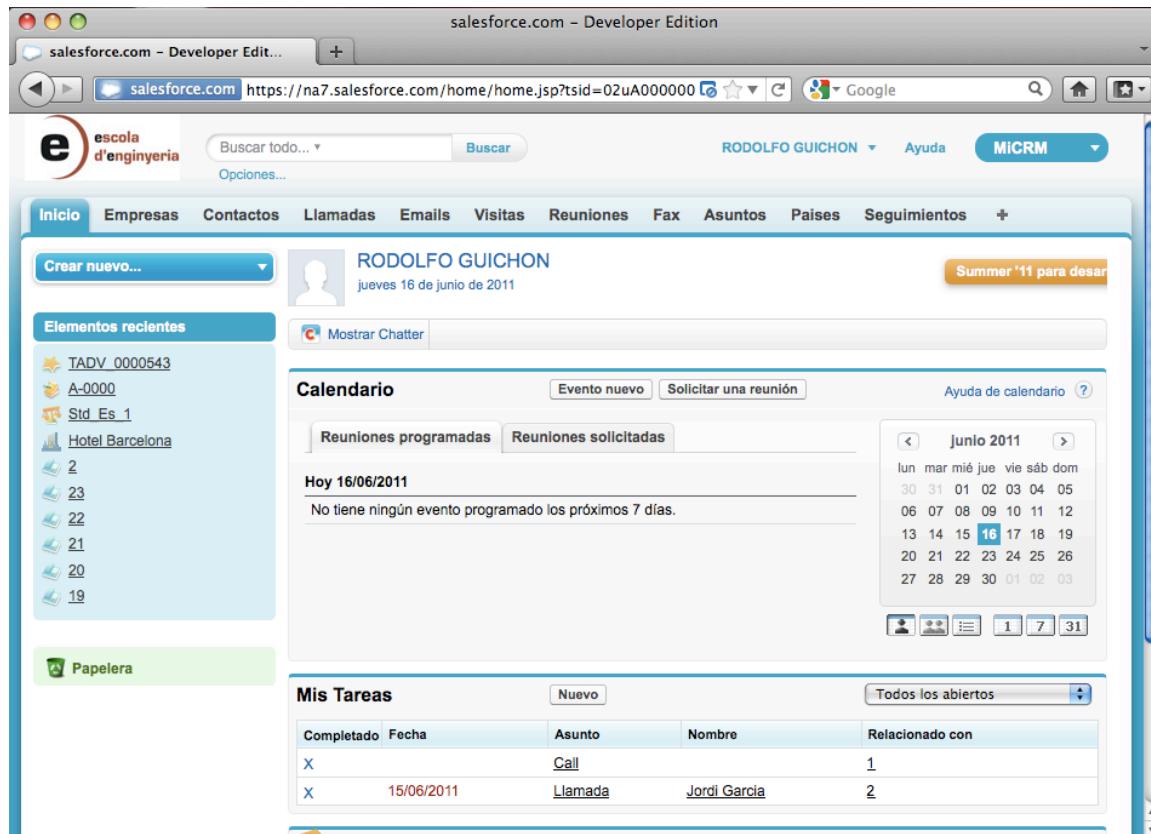
Escriba una descripción breve.

Descripción

Guardar **Cancelar**

Figura 5.3.1.2 Definición de ficha

Una vez generada la aplicación y sus fichas, se puede visualizar el aspecto del nuevo entorno de trabajo en la fig. 5.3.1.3.



Completo	Fecha	Asunto	Nombre	Relacionado con
X	Call		1	
X	15/06/2011	Llamada	Jordi Garcia	2

Figura 5.3.1.3 Aplicación final

5.3.2. Objetos Personalizables

El siguiente paso ha sido la creación de objetos personalizados. Los objetos personalizados son tablas de base de datos que le permiten almacenar datos específicos en Force.com. Se han utilizado objetos personalizados para ampliar las funciones existentes y para añadir nueva funcionalidad en la aplicación. Los objetos que han sido creados son los siguientes:

- Asunto:
Tabla de posibles asuntos a tratar.
- Email
Recolección de correos electrónicos recibidos.
- Empresa
Base de datos con datos fiscales de los clientes.
- Fax
Tabla de Faxes entrantes.
- Llamada
Tabla con las llamadas realizadas y sus transcripciones.
- País
Tabla de países
- Reunión
Reuniones realizadas con sus minutas correspondientes.
- Visita
Visitas realizadas a clientes y resumen de las mismas.

Para crear los objetos primero se ha definido las características generales de cada uno de ellos (Fig 5.3.2.1).

Etiqueta singular	Llamada	Descripción	Cualquier llamada entrante o saliente de una empresa/contacto
Etiqueta plural	Llamadas	Activar informes	✓
Nombre de objeto	Llamada	Realizar un seguimiento de actividades	✓
Nombre de la API	Llamada__c	Seguimiento de historial de campos	✓
		Estado de implementación	Implementado
		Configuración de la Ayuda	Ventana estándar Ayuda de Salesforce
Creada por	RODOLFO GUICHON, 17/11/2010 9:31	Modificado por	RODOLFO GUICHON, 17/11/2010 15:36

Figura 5.3.2.1 Definición de llamada

El siguiente paso ha sido indicar la información que contiene este objeto. Esto campos pueden ser de tipo primitivo o hacer referencia a otros objetos (Fig 5.3.2.2).

Campos estándar						Ayuda de Campos estándar ?
Acción	Etiqueta de campo	Nombre de campo	Tipo de datos	Campo de control	Seguimiento de historial	
Modificar	<u>Creada por</u>	CreatedBy	Buscar(Usuario)		<input type="checkbox"/>	
Modificar	<u>Id de Llamada</u>	Name	Numeración automática		<input type="checkbox"/>	
Modificar	<u>Propietario</u>	Owner	Buscar(Usuario, Cola)		<input type="checkbox"/>	
	<u>Última modificación por</u>	LastModifiedBy	Buscar(Usuario)		<input type="checkbox"/>	

Relaciones y campos personalizados						Ayuda de Relaciones y campos personalizados ?
Acción	Etiqueta de campo	Nombre de la API	Tipo de datos	Campo de control	Modificado por	Seguimiento de historial
Modificar Eliminar	<u>Asunto</u>	Asunto_c	Buscar(Asunto)		<u>RODOLFO GUICHON</u> , 17/11/2010 10:53	<input type="checkbox"/>
Modificar Eliminar	<u>Contacto</u>	Contacto_c	Buscar(Contacto)		<u>RODOLFO GUICHON</u> , 17/11/2010 10:29	<input type="checkbox"/>
Modificar Eliminar	<u>Descripción</u>	Descripcion_c	Área de texto(255)		<u>RODOLFO GUICHON</u> , 17/11/2010 10:28	<input type="checkbox"/>
Modificar Eliminar	<u>Fecha/Hora</u>	Fecha_Hora_c	Fecha/hora		<u>RODOLFO GUICHON</u> , 17/11/2010 10:32	<input type="checkbox"/>
Modificar Eliminar	<u>Teléfono</u>	Telefono_c	Área de texto(255)		<u>RODOLFO GUICHON</u> , 17/11/2010 15:55	<input type="checkbox"/>
Modificar Eliminar Reemplazar	<u>Tipo</u>	Tipo_c	Lista de selección		<u>RODOLFO GUICHON</u> , 17/11/2010 10:32	<input type="checkbox"/>
Modificar Eliminar	<u>Transcripción</u>	Transcripcion_c	Área de texto largo(32000)		<u>RODOLFO GUICHON</u> , 17/11/2010 16:03	<input type="checkbox"/>

Figura 5.3.2 2 Campos de Llamada

Por último, la vinculación de un desencadenador ha sido fundamental para lograr los objetivos de este proyecto. Los desencadenadores han sido desarrollados en *Apex Code* que es un lenguaje de programación orientado a objetos propio de la plataforma Force.com. A través de este código se accede al servicio externo de clasificación *Text2Emotion* y se crean las tareas de seguimiento.

Desencadenadores							Ayuda de Desencadenadores ?
Acción	Nombre	Versión de API	Es válido	Estado	Tamaño sin comentarios	Última modificación por	
Modificar Eliminar	<u>text2EmotionLlamadaTrigger</u>	20,0	✓	Inactive	252	<u>RODOLFO GUICHON</u> , 6/06/2011 21:35	

Figura 5.3.2 3 Desencadenador de llamada

5.3.3. Interfaces de Usuario

A partir de los objetos personalizados, se han diseñado las interfaces de usuario, con herramientas propias de la plataforma y éste ha sido el resultado:

1. Gestión de Llamadas

Edición de Llamada
Nueva Llamada

Modificar Llamada

Información

Descripción	<input type="text"/>	Propietario RODOLFO GUICHON
Contacto	<input type="text"/> <input type="button" value=""/>	
Transcripción	<input type="text"/>	
Teléfono	<input type="text"/>	
Fecha/Hora	<input type="text"/> [6/06/2011 17:38]	
Tipo	<input type="button" value="--Ninguno--"/>	
Asunto	<input type="text"/> <input type="button" value=""/>	

Figura 5.3.3 1 Pantalla de Llamadas

2. Gestión de Emails

Edición de Email
Nuevo Email

Modificar Email

Información

De	<input type="text"/>
Para	<input type="text"/>
Contacto	<input type="text"/> <input type="button" value=""/>
Asunto	<input type="text"/> <input type="button" value=""/>
Mensaje	<input type="text"/>

Figura 5.3.3 2 Pantalla de Emails

3. Gestión de Visitas

Edición de Visita

Nueva Visita

Modificar Visita Guardar Guardar y nuevo Cancelar

Información

Fecha/Hora	<input type="text" value="8/06/2011 8:17"/>
Duración	<input type="text"/>
Motivo	<input type="text"/>
Resultado	<input type="text"/>
Contacto	<input type="text"/> 
Tipo	<input type="text" value="--Ninguno--"/> 
Asunto	<input type="text"/> 

Guardar Guardar y nuevo Cancelar

Figura 5.3.3 3 Pantalla de Visita

4. Gestión de Reuniones

Edición de Reunión

Nueva Reunión

Modificar Reunión Guardar Guardar y nuevo Cancelar

Información

Asunto	<input type="text"/> 
Descripción	<input type="text"/>
Fecha/Hora	<input type="text" value="8/06/2011 8:41"/>
Memorandum	

Figura 5.3.3 4 Pantalla de Reuniones

5. Gestión de Faxes

Edición de Fax

Nuevo Fax

Modificar Fax Guardar Guardar y nuevo Cancelar

Información

Contacto	<input type="text"/>	Propietario	RODOLFO GUICHON
Descripción	<input type="text"/>		
Fecha/Hora	[8/06/2011 10:40]		
Asunto	<input type="text"/>		

Guardar Guardar y nuevo Cancelar

Figura 5.3.3 5 Pantalla de Fax

6. Seguimiento

Edición de Seguimiento

Nuevo Seguimiento

Modificar Seguimiento Guardar Guardar y nuevo Cancelar

Información

Contacto	<input type="text"/> Jordi Garcia
Usuario	<input type="text"/> RODOLFO GUICHON
Tipo Evento	<input type="text"/> <Todos>
Clasificación	<input type="text"/> No Satisfactoria
Usuario Destino	<input type="text"/>
Tarea a realizar	<input type="text"/> Llamada
Intervalo	<input type="text" value="--Ninguno--"/> <div style="border: 1px solid #ccc; padding: 2px; display: inline-block;"> --Ninguno-- Inmediatamente En un día En una semana En quince días En un mes En tres meses En seis meses En un año </div>

Guardar Guardar y nuevo Cancelar

Copyright © 2000-2011 salesforce.com. Todos los derechos reservados. | Declaración de privacidad | De

Figura 5.3.3 6 Pantalla de Seguimiento

5.3.4. Acceso a servicios externos

Por último, se ha configurado el entorno para el acceso a los servicios externos de clasificación de experiencias. Antes de que cualquier página de Force.com, disparadores de Apex o código de JavaScript pueda acceder a un sitio externo, dicho sitio debe registrarse como sitio remoto o el acceso no se realizaría correctamente. Esta configuración incluye el nombre y la URL del sitio remoto (ver Fig 5.3.4.1).

Por razones de seguridad, se restringe los puertos de salida que puede especificar a uno de los siguientes:

- 80: este puerto sólo acepta conexiones HTTP.
- 443: este puerto sólo acepta conexiones HTTPS.
- 1024–66535 (incluido): estos puertos aceptan conexiones HTTP o HTTPS.

Modificar Sitio remoto

Guardar Guardar y nuevo Cancelar

Nombre del sitio remoto	text2Emotion
URL del sitio remoto	http://80.31.109.186:8080
Desactivar protocolo de seguridad	<input checked="" type="checkbox"/> i
Descripción	<input type="text"/>
Activo	<input checked="" type="checkbox"/>

Guardar Guardar y nuevo Cancelar

Figura 5.3.4 1 Acceso a sitio remoto

La dirección URL corresponde a la IP (80.31.109.186) y el puerto (8080) por donde se puede acceder al servicio *text2emotion*.

Una vez configurado el acceso remoto, se ha importado el documento WSDL que contiene la interfaz abstracta a través de la cual se puede conocer y acceder a los servicios de clasificación de *text2Emotion*. Es un fichero XML y describe la forma de comunicación, es decir, los requisitos del protocolo y los formatos de los mensajes necesarios para interactuar con los servicios disponibles. Las operaciones y mensajes que soporta se describen en abstracto y se ligan después al protocolo concreto de red y al formato del mensaje.

Son tres, apenas, los pasos necesarios para integrar los servicios de clasificación:

1. Seleccionar el fichero WSDL

Nuevo código Apex de WSDL [Ayuda para esta página](#) 

Paso 1: Seleccione el documento WSDL

Analizar WSDL Cancelar

Seleccione un documento WSDL de su sistema de archivos de red. El documento WSDL puede tener como máximo 1 MB. Si el documento se analiza, cada espacio de nombre se convierte en una clase de Apex.

[Examinar...](#)

Analizar WSDL Cancelar

Figura 5.3.4 2 Paso 1: Seleccionar fichero WSDL

2. Analizar y validar el fichero.

Nuevo código Apex de WSDL

Ayuda para esta página 

Paso 2: Especifique los nombres de clase

Anterior Generar código Apex Cancelar

 Parse successful: Warnings 0, Errors 0

Las siguientes clases de Apex se han derivado de los espacios de nombre del documento WSDL. Modifique el nombre(s) de clase según proceda. El mismo nombre de clase de Apex se puede utilizar para múltiples espacios de nombres de WSDL, creando una clase única. Tenga en cuenta que una clase simple de Apex puede tener como máximo 100.000 caracteres. La clase(s) de Apex generada incluye métodos para activar los servicios Web de terceras partes representados por el documento WSDL.

Nombre de espacio WSDL	Nombre de clase de Apex
http://service.text2emotion.com/	serviceText2emotionCom

Anterior Generar código Apex Cancelar

Figura 5.3.4 3 Paso 2: Análisis del fichero WSDL

3. Generar la clase Apex.

Nuevo código Apex de WSDL

Ayuda para esta página 

Paso 3: Código generado

Listo

La siguiente clase(s) generada se ha compilado satisfactoriamente sin errores:
serviceText2emotionCom

Listo

Figura 5.3.4 4 Paso 3: Generación de las clases Apex

El resultado es la creación de clases Apex a través de las cuales se puede acceder de forma transparente a los servicios externos de *text2emotion*. Estas clases son utilizadas por los disparadores. Cuando se genera un nuevo registro, ya sea una llamada, un fax, un email, una reunión o una visita, se ejecuta el disparador correspondiente que llama a la función disponible en la interfaz del servicio externo, de forma asíncrona. Cuando el servicio retorna el resultado de la clasificación, se genera entonces la tarea de seguimiento.

Existe un monitor de los procesos generados donde se puede visualizar el estado de las tareas asíncronas (Fig. 5.3.4.5). También, en caso de error, se envía automáticamente un correo electrónico al administrador del sistema.

Trabajos de Apex

Ayuda para esta página 

Controlar el estado de todos los trabajos de Apex y, opcionalmente, cancelar los trabajos que estén en curso.

Ver: **Todos**  Crear nueva vista

Acción	Fecha de envío	Tipo de trabajo	Estado	Detalle del estado	Lotes totales	Lotes procesados	Fallos	Enviado por	Fecha de finalización	Clase de Apex	Método Apex	Id. de trabajo de Apex
	11/06/2011 10:50	Futuro	Completado		0	0	0	GUICHON, RODOLFO	11/06/2011 10:50	Text2EmotionService	train	707A000000DHUO6

Figura 5.3.4 5 Seguimiento de Trabajos de Apex

Finalmente, en la Fig. 5.3.4.6, se puede observar el resultado final: la generación automática de la tarea de seguimiento.

Tarea
Llamada

Modificar formato | Ayuda para esta página ?

1 7 31

« Volver a la lista: Trabajos de Apex

Detalle de Tarea		Modificar	Eliminar	Crear tarea de seguimiento	Crear evento de seguimiento
Asignado a	 <u>RODOLFO GUICHON</u>	Estado	Not Started		
Asunto	Llamada	Nombre	<u>Jordi Garcia</u>		
Fecha de vencimiento	15/06/2011	Relacionado con			
Teléfono		Correo electrónico			
Prioridad	Normal				
Creada por	<u>RODOLFO GUICHON</u> , 8/06/2011 16:07	Última modificación por	<u>RODOLFO GUICHON</u> , 8/06/2011 16:07		
Comentarios	Tarea generada automáticamente por text2emotion. Descripción llamada generada por Lidia el 15/06/2011 15:23: Al cliente le ha parecido bien las muestras. Solicita cotizaciones.				
Recordatorio					
Recordatorio	<input checked="" type="checkbox"/> 15/06/2011 8:00				

Figura 5.3.4 6 Tarea generada automáticamente

5.4. Resultados

Para poder evaluar los resultados obtenidos, primero se ha realizado una valoración de la efectividad del clasificador y luego la repercusión económica que la solución completa puede ocasionar en la empresa.

Efectividad del clasificador:

Para realizar las pruebas del clasificador se ha recogido un total de 860 eventos registrados en el sistema CRM de la empresa. Los dos eventos que más muestras tienen son llamadas y visitas. Se han seleccionado eventos que pudiesen representar las experiencias generadas y se han descartados errores y repeticiones. Con la ayuda del director comercial se ha creado el *Ground Truth*. Al tratarse de acciones comerciales de captación de nuevos clientes los dos resultados posibles de la clasificación fueron: destrucción o captación.

Las pruebas se han realizado mediante la implementación, como un servicio, del modelo estadístico de validación cruzada *k-fold Cross Validation*. La implementación divide aleatoriamente las muestras en k subconjuntos con una proporción parecida de elementos de cada clase. De los k conjuntos, uno es escogido como conjunto de test y los k-1 restantes son las muestras que serán utilizadas para el entrenamiento. El proceso se repite k veces, una por cada conjunto de test. Para obtener el resultado final se ha realizado la media del error relativo y la desviación estándar. La ventaja de este método es que se utiliza todo el conjunto de muestras y que cada muestra se utiliza una sola vez. En nuestro caso hemos utilizado conjuntos de test de 10 elementos (k=10).

Los grados de libertad que ofrecen con los servicios, y por lo tanto, la posibles variaciones del método de clasificación que se dispone para obtener un resultado aceptable son:

1. Lista de palabras y símbolos que se deben eliminar.
2. Uso o descarte de palabras según su categoría morfológica.
3. Uso de conocimiento previo para manipular la importancia de ciertas palabras.
4. Utilización de secuencias de n palabras.

5. Variación de pesos según la posición en el documento.
6. Potencia espectral en la reducción de dimensiones.
7. Número de vecinos a ser considerados para la clasificación.

Después de haber realizado pruebas exhaustivas con variaciones de todas estas variables el resultado obtenido ha sido:

Matriz de Confusión			
Clasificación	Real		
	Destrucción	Destrucción	Captación
	Destrucción	247	125
Captación	183	305	

Figura 5.4 1 Caso 1: Matriz de Confusión

Error Relativo		
Destrucción	Captación	Total
0,43	0,29	0,36

Figura 5.4 2 Caso 1: Matriz de Error Relativo

Desviación Estándar		
Destrucción	Captación	Total
0,095	0,085	0,090

Figura 5.4 3 Caso 1: Matriz de Desviación estándar

Se puede observar que la clasificación es mejor en los eventos del cuadrante de captación. Eso es debido a que los textos son más parecidos entre sí. En cambio los textos de eventos que pertenecen al cuadrante de destrucción son menos homogéneos y se confunden más con los de captación. La desviación estándar, nos indica dispersión de los resultados de la clasificación, la cual cosa nos indica que los resultados de cada iteración en el proceso de validación ha variado. De todas formas, en general, los resultados son buenos debido al grado de similitud entre los documentos. Además muchas veces los textos se repiten, lo que haría aumentar, aún más, la efectividad de la clasificación.

Para analizar la repercusión de la utilización de la solución planteada se ha realizado un análisis de las acciones comerciales de dicha empresa y se ha contrastado los siguientes datos:

1. Los operadores de telemarketing realizan, en media, de 1000 llamadas al trimestre.
2. Un 30% de los receptores de estas llamadas demuestran algún tipo de interés.
3. Un 5% de los interesados acaban realizando algún pedido.
4. Un 40% de los clientes que demostraron interés tienen un seguimiento comercial no adecuado.

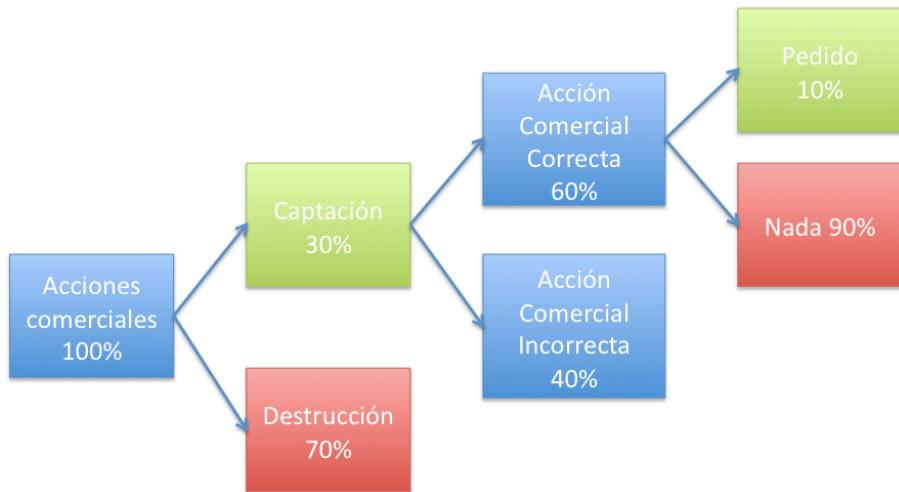


Figura 5.4 4 Efectividad de proceso comercial

Con la ayuda de la solución planteada y ateniéndose a los resultados obtenidos:

1. El 71% de los eventos pertenecientes al cuadrante de captación son bien clasificados y se generan acciones de seguimiento según la política comercial establecida.
2. Calculado en base a los datos anteriores, se incrementa el numero de clientes interesados con un seguimiento adecuado en un 18%. Con lo que aumentamos en la misma proporción la posibilidad de generar pedidos.

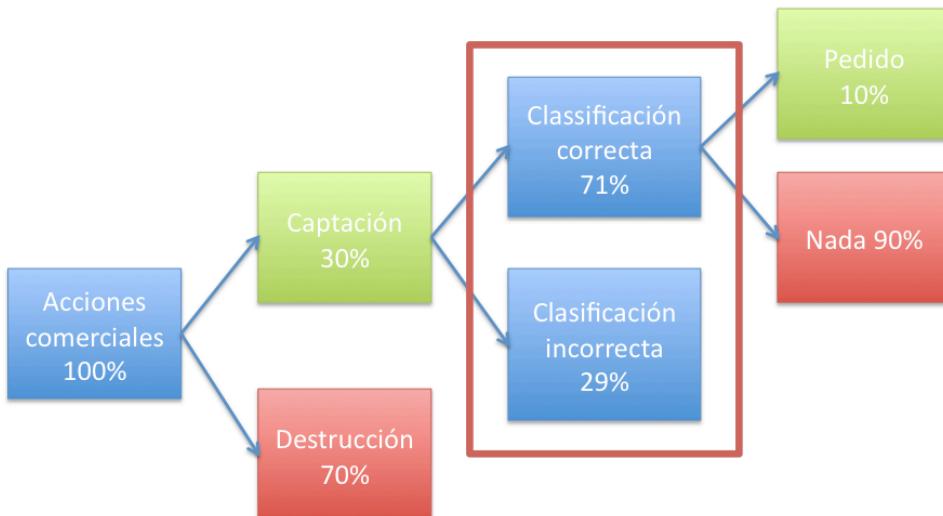


Figura 5.4 5 Efectividad estimada con el uso de Text2Emotion

Los resultados se consideran aceptables y pueden aportar ventajas la utilización de la solución para la empresa.

Capítulo 6

Caso 2: Community Management

6.1. Introducción

TripAdvisor® representa la comunidad de viajes más grande del mundo. Proporciona a sus miembros recomendaciones para hoteles, complejos turísticos, hostales, vacaciones, paquetes de viajes, paquetes de vacaciones y guías de viajes. Los números que describen esta comunidad son:

1. Recibe más de 40 millones de visitas al mes.
2. Tiene más de 20 millones de miembros registrados
3. Posee una base de datos de más de 45 millones de críticas y opiniones.

En este caso, se plantea una solución para un grupo hotelero que quiera controlar la satisfacción de sus clientes en internet. Para ello es necesario acceder a la información desplegada en las redes sociales, procesarla, analizarla y clasificarla para la posterior toma de decisión.

Se ha creado un nuevo entorno de trabajo para el control de los hoteles y sus clientes. En este caso entra en juego un nuevo perfil: el *community manager*. Esta persona es la responsable de la reputación del hotel en las redes sociales. Debe agradecer y fortalecer vínculos con aquellos clientes que les recomiendan y promocionan, y también, debe rescatar aquellos clientes que están en el cuadrante de destrucción.

Dado el volumen de información generada en las redes sociales, es imposible controlar, valorar y reaccionar ante cada opinión o comentario que se crea en Internet.

La propuesta ha sido optimizar la reacción utilizando, por un lado, el entorno de trabajo creado en la plataforma y por el otro, la automatización de la clasificación de las opiniones y comentarios para dar respuestas de la forma más automatizada posible.

Se ha creado un prototipo de aplicación funcional en Force.com utilizando, nuevamente, toda la infraestructura de la plataforma. Se ha aprovechado la anterior integración de la solución *text2emotion* con la plataforma. Con este prototipo se pretende:

- Organizar de mejor forma la información.
- Análisis de la información disponible en redes sociales en tiempo pseudo-real.
- Mejorar la eficiencia de los procesos de marketing.
- Aumentar la satisfacción de los clientes.
- Mejorar la **reputación digital**

6.2. Acción y reacción: creando tareas a partir de valoraciones de experiencias

Tal y como se ha comentado anteriormente, el *community manager* debe controlar las opiniones, comentarios y valoraciones que se realizan en las redes sociales para asegurar la buena reputación de su empresa en las comunidades online.

En la comunidad con la que se ha trabajado, las personas que opinan son miembros registrados. También pueden estar vinculados con otras redes sociales como Facebook, por lo tanto, pueden ser identificadas y registradas como contactos en la plataforma. Las opiniones son introducidas en formato de texto. Primero se introduce una frase representativa y luego el usuario puede describir con detalle la experiencia. También valoran el hotel, tanto a nivel general, como en aspectos concretos con una puntuación del 1 al 5.

Se ha utilizado, para la clasificación, el título y el texto detallado. Se han utilizado las puntuaciones para crear el Ground Truth. Analizando el conjunto de entrenamiento se ha hecho corresponder las puntuaciones con los cuadrantes emocionales:

- 1-2 puntos: Destrucción.
- 3 puntos: Captación.
- 4 puntos: Recomendación.
- 5 puntos: Promoción.

La respuesta propuesta, provocada por una inserción de una opinión en la red, es una tarea creada y asignada a un usuario automáticamente para asegurar su seguimiento. Esto se realiza teniendo en cuenta la siguiente información:

- Hotel.
- País
- Categoría
- Cliente que vivió la experiencia.
- Origen (TripAdvisor, Facebook, ...).
- Resultado de clasificación de la experiencia (Destrucción, Captación, Recomendación o Promoción).

A partir de esta información se puede automatizar el seguimiento conociendo:

- Persona que debe realizar el seguimiento (*community manager*).
- Tipo de respuesta.
- Respuesta recomendada.

La creación de estas tareas no necesitan interacción del usuario. Se realizan automáticamente. En ciertos intervalos de tiempo se ejecuta

Con tareas programadas y a través de servicios Web, se extrae de la red social nuevas opiniones, se utiliza el servicio de clasificación de emociones *text2emotion*. Éste retorna el resultado de la clasificación y una rutina desarrollada en la plataforma es la encargada de crear la nueva tarea de seguimiento. Todo esto se hace de forma asíncrona y transparente para el usuario.

6.3. Creación del entorno de trabajo en Force.com

6.3.1. Aplicación

Se ha creado otra aplicación en la plataforma Force.com llamada *Community Management*. Las fichas corresponden a objetos personalizados de la problemática que se trata. El entorno de trabajo se muestra en la Fig. 6.3.1.1.

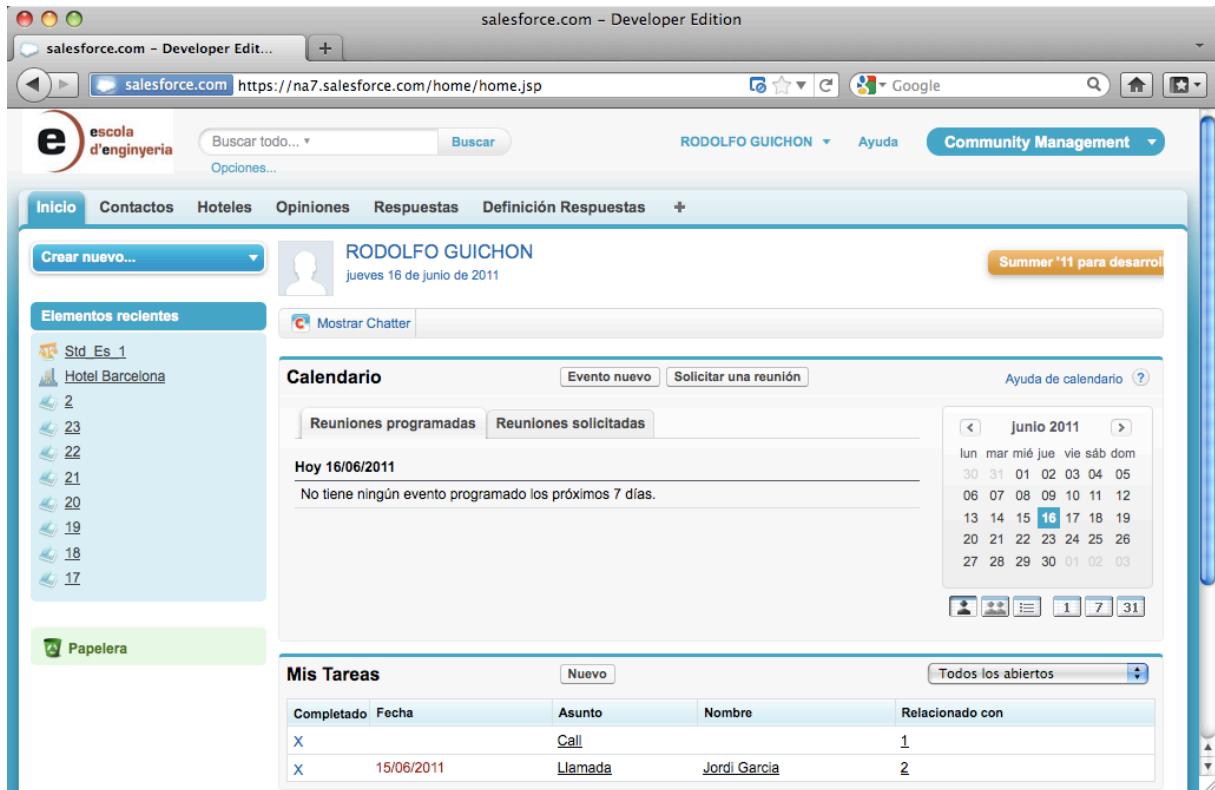


Figura 6.3.1.1 Aplicación Final

6.3.2. Objetos Personalizables

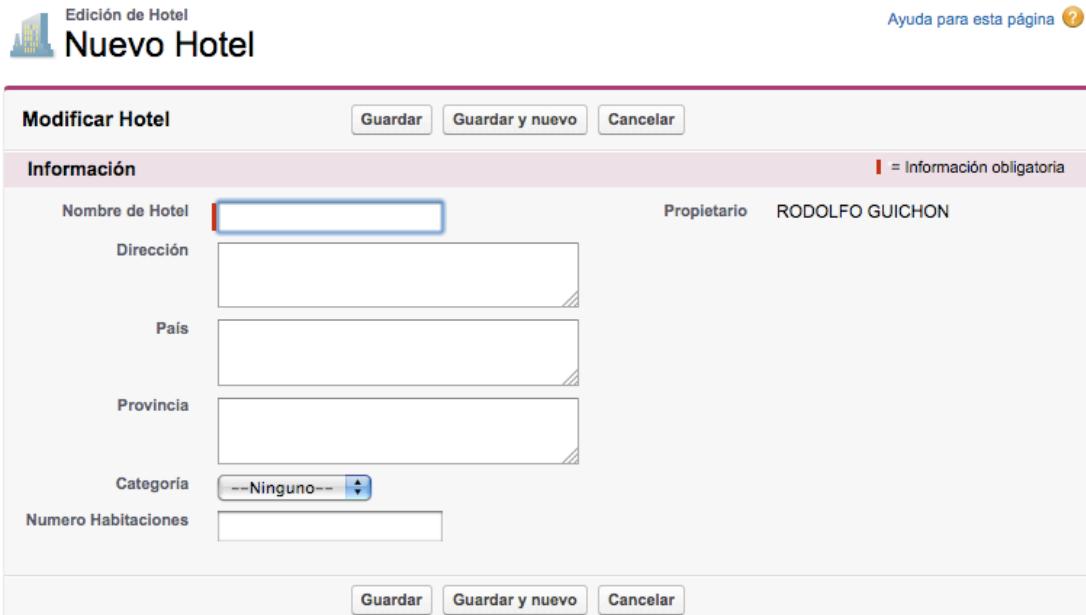
Los objetos creados son los siguientes:

- **Hoteles:**
Datos de los hoteles a los cuales se vinculan las opiniones.
- **Opiniones:**
Tabla que contiene las opiniones descargadas de las redes sociales. Cuando se inserta el registro se clasifica la opinión y se crea una tarea para responder a ésta.
- **Respuestas:**
Base de respuestas efectivamente realizadas vinculadas a una opinión.
- **Definición de respuestas:**
Reglas de creación de tareas para dar respuestas.

6.3.3. Interfaces de Usuario

A partir de los objetos personalizados, se han diseñado nuevas interfaces de usuario, con herramientas propias de la plataforma:

1. Gestión de Hoteles



Edición de Hotel

Nuevo Hotel

Modificar Hotel

Guardar Guardar y nuevo Cancelar

Información

■ = Información obligatoria

Nombre de Hotel

Dirección

País

Provincia

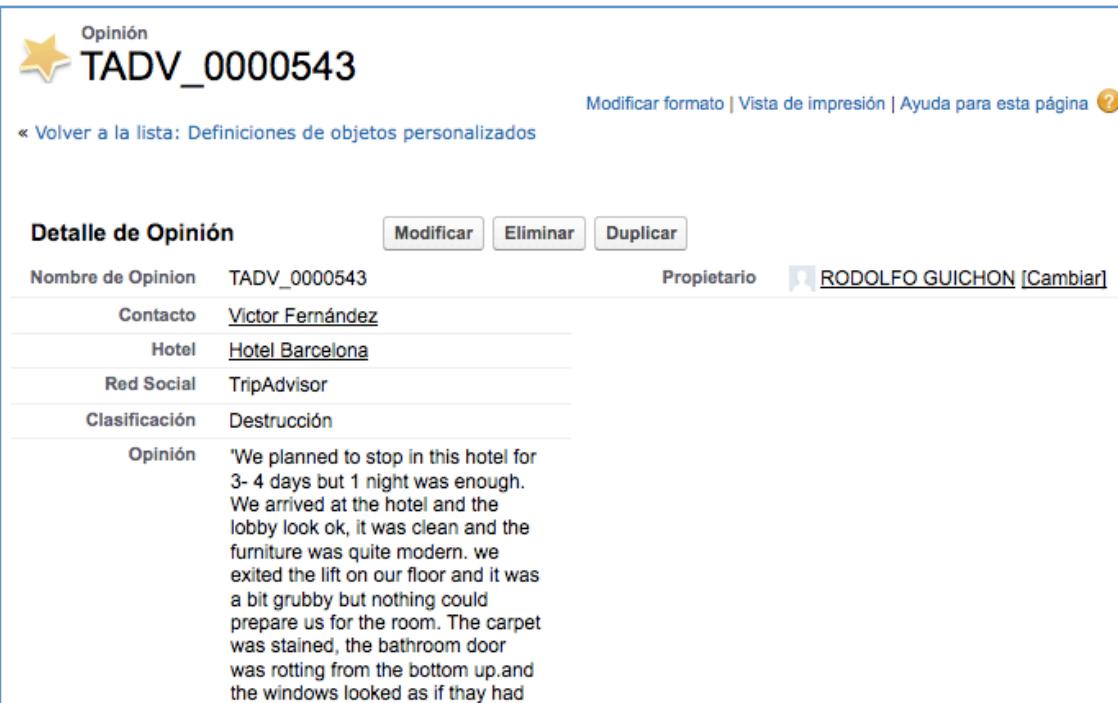
Categoría

Número de Habitaciones

Guardar Guardar y nuevo Cancelar

Figura 6.3.3 1 Pantalla de Hoteles

2. Gestión de Opiniones



Opinión

TADV_0000543

Modificar formato | Vista de impresión | Ayuda para esta página

« Volver a la lista: Definiciones de objetos personalizados

Detalle de Opinión

Modificar Eliminar Duplicar

Nombre de Opinión TADV_0000543 Propietario RODOLFO GUICHON [Cambiar]

Contacto Victor Fernández

Hotel Hotel Barcelona

Red Social TripAdvisor

Clasificación Destrucción

Opinión

"We planned to stop in this hotel for 3- 4 days but 1 night was enough. We arrived at the hotel and the lobby look ok, it was clean and the furniture was quite modern. we exited the lift on our floor and it was a bit grubby but nothing could prepare us for the room. The carpet was stained, the bathroom door was rotting from the bottom up, and the windows looked as if they had

Figura 6.3.3 2 Pantalla de Opiniones

3. Gestión de Respuestas

Figura 6.3.3 3 Pantalla de Respuestas

4. Gestión de Definición de Respuestas (reglas)

Figura 6.3.3 4 Pantalla de definición de respuestas

5. Programación de Recolección de Opiniones

Edición de Programación [Ayuda para esta página](#) 

Nueva Programación

Modificar Programación Guardar Guardar y nuevo Cancelar

Información

Red Social	<input type="text" value="TripAdvisor"/>	Propietario	RODOLFO GUICHON
Día Semana	<input style="background-color: #f0f0f0; border: 1px solid #ccc; padding: 2px; width: 100px; height: 100px;" type="text" value="Disponible"/> <div style="border: 1px solid #ccc; padding: 5px; width: 150px; height: 150px; margin-top: 5px;"> Elegido Lunes Martes Miércoles </div>		
Mes	<input style="background-color: #f0f0f0; border: 1px solid #ccc; padding: 2px; width: 100px; height: 100px;" type="text" value="Disponible"/> <div style="border: 1px solid #ccc; padding: 5px; width: 150px; height: 150px; margin-top: 5px;"> Elegido Enero Febrero </div>		
Hora	<input type="text" value="16/06/2011 19:46"/>		
Intervalo	<input type="text" value="Cada 30 minutos"/>		
Inicio	<input type="text" value="16/06/2011 19:46"/>		
Fin	<input type="text" value="16/06/2011 19:46"/>		

Figura 6.3.3.5 Pantalla de programación de descargas

6.3.4. Acceso a servicios externos

De la misma forma que en el caso anterior, la plataforma Force.com accede a los servicios de Text2Emotion para poder clasificar las opiniones. Existe un desencadenador asociado a la creación de los registros de Opiniones que se encarga de ello.

También hay un nuevo acceso remoto: los servicios API de TripAdvisor. Estos servicios están disponibles solamente para clientes de TripAdvisor y no se ha podido tener acceso a ellos durante el desarrollo de este proyecto. Aún así, se han configurado las interfaces para el acceso a estos servicios cuando estuviesen disponibles. Se ha creado además otro objeto que controla la programación de las tareas de recolección de opiniones. Se puede programar, para cada red social, el intervalo en el que se desea recolectar los datos. Una vez que la tarea es ejecutada automáticamente, se generan las opiniones y se invoca al servicio de clasificación.

6.4. Resultados

Las pruebas del clasificador se han realizado con 400 opiniones, con valoraciones balanceadas, extraídas del portal TripAdvisor. Como se ha comentado anteriormente se ha relacionado las valoraciones con los cuadrantes emocionales. Estas opiniones están escritas en inglés.

Las pruebas se han realizado, nuevamente, mediante la utilización del modelo estadístico de validación cruzada *k-fold Cross Validation* con conjuntos de test de 10 elementos ($k=10$).

Los grados de libertad que ofrecen con los servicios y con los cuales se han realizado pruebas para obtener un buen resultado siguen siendo:

- ### 1. Lista de palabras y símbolos que se deben eliminar.

2. Uso o descarte de palabras según su categoría morfológica.
3. Uso de conocimiento previo para manipular la importancia de ciertas palabras.
4. Utilización de secuencias de n palabras.
5. Variación de pesos según la posición en el documento.
6. Potencia espectral en la reducción de dimensiones.
7. Número de vecinos a ser considerados para la clasificación.

Después de haber realizado pruebas exhaustivas con variaciones de todas estas variables el resultado obtenido ha sido:

Matriz de Confusión					
Clasificación	Real				
		Destrucción	Captación	Recomendación	Promoción
	Destrucción	33	12	3	6
	Captación	28	36	1	2
	Recomendación	26	32	54	40
	Promoción	13	20	42	52

Figura 6.3.3 6 Caso 2: Matriz de confusión

Error Relativo				
Destrucción	Captación	Recomendación	Promoción	Total
0,67	0,64	0,46	0,48	0,56

Figura 6.3.3 7 Caso 2: Matriz de Error Relativo

Desviación Estándar				
Destrucción	Captación	Recomendación	Promoción	Total
0,13	0,15	0,06	0,07	0,10

Figura 6.3.3 8 Caso 2: Matriz de Desviación Estándar

Los resultados no son tan buenos como en el caso anterior. Se puede observar que las opiniones pertenecientes a los cuadrantes Recomendación y Promoción se confunden entre sí pero no entre los otros. Esto es aceptable ya que tanto uno como otro está formado por buenas opiniones. En cambio, los cuadrantes de Destrucción y Captación están más distribuidos debido a la diversidad de textos que se ha tratado. Se plantea para futuras mejoras la utilización de un Support Vector Machine para mejorar los resultados de clasificación.

Conclusiones

El hilo conductor de este proyecto ha sido la creación de una aplicación capaz de valorar experiencias. Para ello, primero se han creado los servicios que realizan la clasificación y después las aplicaciones de negocio que utilizan estos servicios.

Para la creación de los servicios de clasificación se ha utilizado la arquitectura orientada a servicios. De esta forma se ha podido desarrollar servicios que pueden ser utilizados por cualquier tipo de aplicación desplegada en cualquier plataforma. Además esto ha aportado otras ventajas como la reutilización de servicios: se han utilizado todos los servicios de preproceso y análisis de textos tanto para el proceso de entrenamiento como el de clasificación. Por otro lado el uso de esta arquitectura ha facilitado también la variación de la composición de los servicios para poder evaluar los resultados utilizando diferentes parámetros y funciones.

Todavía en esta primera parte, cabe valorar los algoritmos de clasificación. Clasificar las experiencias emocionalmente es muy difícil. La creación del Ground Truth ya suponía todo un reto ya que muchas transcripciones de llamadas hacían dudar acerca del sentimiento que realmente estaba sintiendo la persona al otro lado del teléfono. Para poder obtener mejores resultados se debe ir mucho más allá de los procesos de text mining convencionales. Se ha de entender sintácticamente y semánticamente los documentos para poder extraer resultados más fiables. Por último, los resultados obtenidos con el método *k-nearestneighbor* podrían, posiblemente, ser mejorados con un *Support Vector Machine*. Estaba fuera del alcance de este proyecto el desarrollo de este clasificador pero podría ser una buena opción.

La segunda parte fue desarrollada íntegramente en la plataforma de desarrollo Force.com. Se ha podido comprobar la facilidad de uso, la fiabilidad y escalabilidad que esta plataforma ofrece en modo de *Cloud Computing*. Se ha podido integrar fácilmente los servicios de clasificación de experiencias y las dos aplicaciones creadas son capaces de gestionar los eventos, clasificarlos emocionalmente y crear las tareas de seguimiento que fuesen necesarias.

Ha sido una gran experiencia personal el poder haber integrado en una solución *Cloud Computing, Service Oriented Architecture* y métodos de clasificación de inteligencia artificial para crear una herramienta de soporte que puede aportar mucho valor añadido a empresas que la puedan utilizar.

Bibliografía

- [1] Shaw, Colin. The DNA of Customer Experience. PALGRAVE MACMILLAN, 2007.
- [2] Coussement, Kristof; Van den Poel, Disk . Improving customer complaint management by automatic email classification using linguistic features as predictors. ScienceDirect, 2007.
- [3] Severt, Denver Eugene. The Customer's Path to Loyalty: A parcial Test of the Relationships of Prior Experience, Justice and Customer Satisfaction. UMI, 2002.
- [4] Levine, Locke, Searls, Weinberger. The Cluetrain Manifesto: The end of Business as usual. 1999,2001.
- [5] B. Joseph Pine II, James H. Gilmore. Welcome to the Experience Economy. Harvard Business Review, 1998.
- [6] Tate, Rick, Stroup Josh. The Service Pro: Creating Better, Faster and Different Customer Experiences. HDR Press, 2003.
- [7] http://en.wikipedia.org/wiki/Customer_experience. Customer Experience.
- [8] media.techtarget.com/searchSOA/.../Force_com_CH01.pdf. Introducing Force.com.
- [9] <http://aye.comp.nus.edu.sg/~qiu/NLPTools/JavaRAP.html>. Resolution of Anaphora Procedure.
- [10] <http://incubator.apache.org/opennlp/index.html>. Apache Open NLP.
- [11] <http://www.rightnow.com/cx-suite.php>. Right Now CX.
- [12] <http://www.responsetek.com>. Customer Experience Management Software.
- [13] <http://www.crmag.it/News/36/CEM-Customer-Experience-Management.aspx>. Il CEM (Customer Experience Management).
- [14] <http://www.data-miners.com/companion/Chapter14-Example2.ppt>. Data Mining Techniques for CRM.
- [15] Martin Keen, Greg Ackerman, Islam Azaz, Manfred Haas, Richad Johnson, JeeWook Kim, Paul Robertson. Patterns: SOA Foundation-Business Process Management Scenario. RedBooks, 2006.
- [16] Linthicum, David. Cloud Computing and SOA Convergence in Your Enterprise. Addison-Wesley, 2010.
- [17] Force.com Apex Code Developer's Guide. Salesforce, 2010.
- [18] Web Services Api Developer Guide. Salesforce, 2010.
- [19] <http://www.anderson.ucla.edu/faculty/jason.frand/teacher/technologies/palace/datamining.htm>. What is Data Mining?
- [20] <http://www.clarabridge.com>. Customer Experience Management Software

- [21] http://en.wikipedia.org/wiki/Latent_semantic_analysis. Latent Semantic Analysis.
- [22] Scott Deerwester, Susan T. Dumais, Richard Harshman. Indexing by Latent Semantic Analysis. 1998
- [23] Marko Grobelnik, Dunja Mladenic. Text Mining Tutorial. J. Stefan Institute.