

**PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA**

**MODELO DE CAPTURA DE  
ESTÍMULOS Y RESPUESTA  
MULTICANAL PARA  
SISTEMAS DE ATENCIÓN AL CLIENTE**

**CAMILO FRANCISCO CUEVAS PÉREZ**

**INFORMA FINAL DE PROYECTO  
PARA OPTAR AL TÍTULO PROFESIONAL DE  
INGENIERO CIVIL EN INFORMÁTICA**

Agosto 2015

**PONTIFICIA UNIVERSIDAD CATÓLICA DE VALPARAÍSO  
FACULTAD DE INGENIERÍA  
ESCUELA DE INGENIERÍA INFORMÁTICA**

**MODELO DE CAPTURADE  
ESTÍMULOS Y RESPUESTA  
MULTICANALPARA  
SISTEMAS DE ATENCIÓN AL CLIENTE**

**CAMILO FRANCISCO CUEVAS PÉREZ**

Profesor Guía: **Rodrigo Alfaro Arancibia**

Profesor Correferente: **Claudio Cubillos**

Carrera: **Ingeniería Civil Informática**

Agosto 2015

## **Resumen**

En este documento se presenta un modelo de unificación de canales de comunicación y redes sociales, con el objetivo de identificar al cliente que acude a en ayuda de servicios entregados por la empresa, servicios que son gestionados por una estrategia y un sistema CRM. De esta manera se obtiene una mayor información del cliente que mejore el trato y la satisfacción. Esto se lleva a cabo gracias a las tecnologías de interconexión de aplicación basadas en servicios REST e interfaces de comunicación y autenticación como OAuth e IMAP. Se propone con esto una arquitectura de software que gestione los elementos interconectados.

**Keywords:** Servicio al cliente, CRM, Redes Sociales, Multicanal, API

## **Agradecimientos**

En este momento culmine de mi camino como estudiante, quiera agradecer a todos quienes han aportado a mi formación. En Primer lugar agradezco a mi familia, por su apoyo constante e incondicional, por enseñarme a buscar siempre logros concretos y cultivar habilidades que creen valor.

Agradezco a los profesores de la escuela de informática, por bríndame las herramientas necesarias para crear tecnología y por otorgar una visión amplia y adaptable del área profesional.

Agradezco la labor anónima de profesionales y autodidactas que crean tecnología y exponen su trabajo a libre disposición de otros a través del software de código abierto, su labor potencia e inspira.

Agradezco a mis compañeros de trabajos por su compañerismo y apoyo.

Finalmente agradezco la orientación de los profesores guía de este proyecto de titulación, por la oportunidad de presentar un tema de actualidad, por su paciencia e instrucción.

# Índice

Resumen .....	iii
Agradecimientos .....	iv
Índice .....	v
Lista de Ilustraciones .....	vii
Lista de Tabla .....	viii
1. Introducción.....	1
2. Objetivos.....	2
2.1. Objetivo General.....	2
2.2. Objetivos Específicos .....	2
3. Justificación .....	3
4. Marco Teórico .....	4
4.1. Servicio al Cliente .....	4
4.1.1 Satisfacción al cliente ISO 9001 .....	5
4.1.2. Modelo ACSI de Satisfacción del cliente.....	7
4.1.3. CRM: Customer Relationship Management .....	9
4.1.4. Procesos de Servicio al Cliente .....	10
4.2. Redes Sociales .....	12
4.2.1. API y Servicios REST .....	13
4.2.2. Social CRM .....	14
4.3. Necesidades de Mejora a Nivel Nacional.....	16
5. Definición del Problema.....	18
6. Metodología de Desarrollo .....	19
7. Análisis del Sistema .....	22
7.1. Alcance .....	22
7.2. Requerimientos Funcionales .....	22
7.3 Requerimientos No Funcionales.....	23
7.3.1 Performance.....	23
7.3.2. Términos de Servicio.....	23
7.3.3. Política de Privacidad .....	23
8. Diseño del sistema .....	24

8.1. Casos de Uso General.....	24
8.2. Modelo de Clases .....	26
8.3. Modelo de Integración de Canales .....	26
9. Arquitectura del modelo multicanal .....	29
9.1. Arquitectura de Software.....	30
9.2. Driver Canal .....	31
9.3. Integración de API.....	32
9.4. HybritAuth.....	33
9.5. Secuencias de Sincronización.....	35
10. Aplicación Modelo Multicanal.....	39
10.1 Canales .....	39
10.2. Contactos .....	41
10.2.1. Contactos Unificados.....	42
10.2.2. Perfiles de Contacto.....	43
10.2.3. Unir Contactos .....	43
10.3. Conversaciones .....	44
10.4. Chat.....	45
11. Experiencia de Desarrollo .....	47
12. Conclusiones.....	49
13. Referencias Bibliográficas.....	50
14. Referencias Web.....	51
Anexo 1: Procesos CRM .....	52
Anexo 2: Modelo de Clases.....	53
Anexo 3: Modelo de Clases Especificó.....	53
Anexo 4: Secuencia Sincronizar Contactos.....	66

## Lista de Ilustraciones

Ilustración 1: Modelo de Satisfacción Del Cliente.....	7
Ilustración 2: Procesos fundamentales de una estrategia CRM.....	11
Ilustración 3: Metodología de Prototipado Rápido.....	20
Ilustración 4: Diagrama de Casos de Uso.....	25
Ilustración 5: Modelo de Integración de Canales .....	27
Ilustración 6: Modelo de Clases .....	28
Ilustración 7: Arquitectura de Software.....	30
Ilustración 8: Configuración HybritAuth .....	34
Ilustración 9: Secuencia Sincronizar Contactos MVC .....	36
Ilustración 10: Secuencia SincronizarContactos CanalTwitter .....	38
Ilustración 11: Canales .....	39
Ilustración 12: Panel de Conexión.....	40
Ilustración 13: Noticias de Sincronización.....	41
Ilustración 14: Contactos Unificados .....	42
Ilustración 15: Perfiles de Contacto.....	43
Ilustración 16: Unir Contactos.....	44
Ilustración 17: Conversaciones.....	44
Ilustración 18: Timeline.....	45
Ilustración 19: Chat .....	46

## Lista de Tabla

Tabla 1: Diferencias entre CRM y CRM Social.....	15
Tabla 2: Licitaciones Públicas.....	18
Tabla 3: Ficha del Proyecto.....	29
Tabla 4: Driver Canal.....	32
Tabla 5: Integración Driver Canal.....	33

# 1. Introducción

La tecnología de redes sociales ha revolucionado el comportamiento de las personas en internet, creando una comunidad participativa y en constante conexión. La aparición de diversas redes según funciones, contenidos y público objetivo ha creado una red de comunicación multicanal entre las personas.

Como resultado adicional la aparición de las redes sociales, la identidad de las personas se ha fragmentado de acuerdo a cada perfil creado en estas redes. Esta condición impone un desafío a las empresas de servicios que dependen del contacto con el cliente, requiriendo la integración de diversos canales de comunicación y de la tecnología que permita identificar quien es quien en la red.

En el presente informe se muestra el desarrollo de un modelo de integración de canales de comunicación en la red, que permita el envío y recepción de mensajes entre múltiples canales y perfiles, que logre unificar la identidad de los usuarios y permita a una empresa comunicarse con sus clientes de manera oportuna..

## **2. Objetivos**

Se presentan a continuación los objetivos a cumplir en el proyecto de título.

### **2.1. Objetivo General**

- Definir un modelo de comunicación multicanal que permita la captura y respuesta de estímulos provenientes de diversos medios sociales y tecnológicos.

### **2.2. Objetivos Específicos**

- Identificar los canales de comunicación que permitan una integración con un sistema unificado.
- Definir un modelo genérico que permita acoplar diferentes tecnologías
- Presentar una arquitectura que soporte el modelo de comunicación multicanal.
- Implementar un software que muestre el funcionamiento básico que propone el modelo.

### **3. Justificación**

En la actualidad, las competencias entre las empresas las impulsan constantemente destacarse del resto a través de sus productos, procesos y tecnologías. Dentro de este marco, las mejoras en el ámbito del servicio al cliente son vitales para mantener a los clientes fieles a las marcas y aumentar su índice de satisfacción.

Los clientes actuales han cambiado, son más participativos, críticos y buscan la personalización de los servicios. Esto se debe en gran medida al auge de las redes sociales y su potencial de comunicación. Esto presenta un desafío a las empresas viéndose necesitada de medios para captar los mensajes de los clientes en los múltiples canales de comunicación que hoy existen y adaptar estos mecanismos a sus procesos.

Hoy existen productos de software que otorgan herramientas de análisis y operación de cuentas de usuario y clientes en varias redes sociales. Sin embargo, no existe gran información sobre el desarrollo de este tipo de herramientas a nivel académico, dificultando el estudio y alcance de mejoras y nuevas tecnologías.

Trabajar con múltiples canales de comunicación presenta un gran desafío de estandarización en la comunicación de aplicaciones y redes sociales, así como también una estandarización del perfil de usuario, autenticación e información relevante de las personas que se comunican por estos medios.

Los mercados digitales muestran esto como una demanda actual de varias empresas y su potencial abarca tanto como el nivel de satisfacción del cliente, la mejora en los procesos internos de la empresa así como también el cumplimiento de los marcos legales de los servicios de atención al cliente que debe controlar cada órgano regulador de los estados.

## **4. Marco Teórico**

El avance en las tecnologías de comunicación y medios sociales exigen una respuesta adaptativa a las áreas de servicio. Para abarcar en los conceptos fundamentales que requieren esta investigación y desarrollo, se mostrara información de las redes sociales actuales y las nuevas tendencias que posicionan al servicio al cliente en la vanguardia de los medios sociales.

### **4.1. Servicio al Cliente**

El servicio al cliente es el conjunto de estrategias que una compañía diseña para satisfacer, mejor que sus competidores, las necesidades y expectativas de sus clientes externos. Entre las características más comunes se encuentran las siguientes:

- a) Es Intangible, no se puede percibir con los sentidos.
- b) Es Perecedero, Se produce y consume instantáneamente.
- c) Es Continuo, quien lo produce es a su vez el proveedor del servicio.
- d) Es Integral, todos los colaboradores forman parte de ella.
- e) La Oferta del servicio es prometer y cumplir.
- f) El Foco del servicio es satisfacción plena del cliente.
- g) El Valor agregado es un plus al producto.

Actualmente es de gran importancia debido a la dirección que el mercado ha dado con la comercialización de los servicios y la trascendencia que dicho servicio tiene como elemento importante en la diferenciación de modelos de negocio.

En el mercado se encuentran en competencia todos los sectores económicos. Un factor que genera la diferencia es el servicio que se le brinda a los clientes, el cual, si es adecuadamente establecido y desarrollado se constituye en un elemento de competitividad, lo cual mejora el posicionamiento en el mercado de la organización. En los momentos de crisis, muy comunes hoy en día, es importante que las empresas garanticen la fidelidad y confianza de sus clientes, a través del ofrecimiento de mejores beneficios.

Debido a todo ello, es importante conocer los modelos y mecanismos que están empleando las empresas que buscan que el servicio al cliente sea su factor de diferencia, complementando con el conocimiento directo de la impresión que tiene el cliente respecto a las empresas.

### **4.1.1 Satisfacción al cliente ISO 9001**

La satisfacción del cliente es la percepción que el cliente tiene sobre el grado en que se han cumplido sus requisitos. La satisfacción es un estado psicológico, y por tanto subjetivo, cuya obtención asegura fidelidad. La satisfacción del cliente es a la organización, lo que la felicidad es a la persona.

ISO 9001:00 impulsa a las organizaciones a que alcancen la satisfacción del cliente. Sitúa este objetivo en su punto de mira, y propugna la mejora del sistema de gestión de la calidad como vehículo hacia su consecución. En ISO 9001 la meta no es la calidad, la meta es aumentar la satisfacción al cliente.

La diferencia de concepto es substancial. La organización no decide, deciden los clientes. Esta “democratización” del modelo se produjo con la transición de la Norma en el año 2000, cuando pasamos de un modelo de aseguramiento de la calidad, a otro de gestión de la calidad.

Los requisitos de la Norma con relación a la satisfacción del cliente abren las puertas a la realización de todo tipo de acciones, nos dice Qué, pero no Cómo. Pide literalmente lo siguiente: Como una de las medidas del desempeño del sistema de gestión de la calidad, la organización debe realizar el seguimiento de la información relativa a la percepción del cliente respecto al cumplimiento de sus requisitos por parte de la organización. Deben determinarse los métodos para obtener y utilizar dicha información. Obsérvese que no se habla de “calcular”, sino de “realizar el seguimiento”, concepto que desglosa a continuación en 2 etapas:

1ª Etapa: obtener información

2ª Etapa: utilizar la información

La organización debe determinar los métodos para realizar el seguimiento de la satisfacción del cliente, debe determinar Qué, Cómo, Quién y Cuándo se obtiene y se utiliza la información. La satisfacción del cliente se define en la Norma ISO9000:00 Fundamentos y vocabulario acompañada de 2 notas muy reveladoras

La autocomplacencia está reñida con el seguimiento de la satisfacción del cliente, lo que más importa es lo que opine el cliente. Aquí se rescata el célebre: el cliente siempre tiene razón, y si no la tiene, se aplica la primera regla. Más claves sobre la aplicación de este requisito las podemos encontrar en su hermana: ISO 9004:00. Allí se nos dice que el seguimiento y la medición de la satisfacción del cliente se basan en la revisión de información relacionada con el cliente, y que la recolección de esta información puede ser activa o pasiva. Indica que se deben reconocer las múltiples fuentes de información, y que

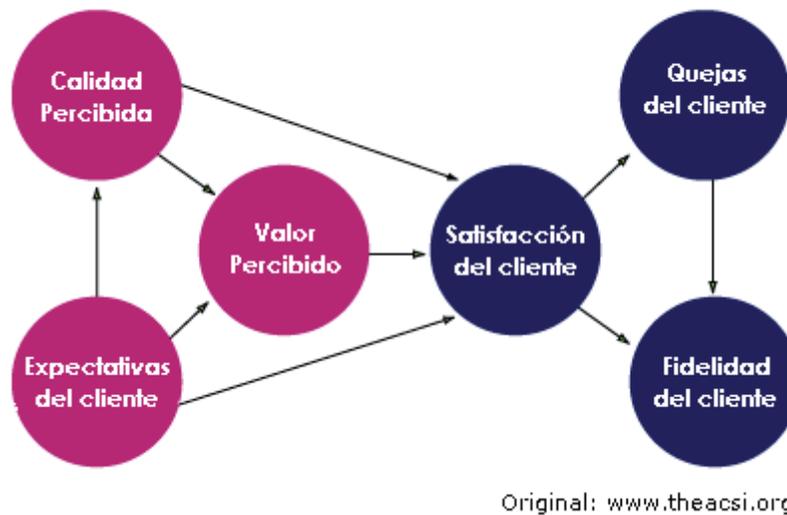
se deben establecer procesos eficaces para recolectarla. Ejemplos de información sobre la satisfacción del cliente:

- **Encuestas rellenas por el cliente:** la organización pregunta al cliente de forma activa.
- **Quejas del cliente:** fuente de información de incalculable valor a disposición de la organización sin mayor trabajo que tomar nota de ellas.
- **Opiniones del cliente sobre los productos:** el cliente expresa su opinión respecto al producto, la organización obtiene su opinión de forma pasiva.
- **Requisitos del cliente e información del contrato:** qué quería el cliente, qué le hemos dicho que le íbamos a dar, y qué le hemos entregado. Búsqueda activa de información indirecta sobre la satisfacción del cliente.
- **Necesidades del mercado:** qué está esperando el cliente y qué estamos ofreciendo nosotros. El cliente tiene expectativas, pero unas más importantes que otras. Conocer en qué medida nuestros puntos fuertes coinciden con lo que el cliente espera es obtener información indirecta sobre su satisfacción de forma activa.

ISO 9001:00 establece qué se debe hacer, pero no cómo. Las pistas más reveladoras y los ejemplos más vivos los podemos encontrar fuera de ella. No obstante, ISO 9001 es una “recién llegada” en casi todo, sus páginas son el fruto de un consenso sobre lo ya existe, y no suele tomar partido por ninguna técnica en particular, porque todas pueden ser útiles si se utilizan bien.

#### 4.1.2. Modelo ACSI de Satisfacción del cliente

The American Customer Satisfaction Index es un indicador que establece el nivel de satisfacción de los ciudadanos de los EEUU con los productos y servicios recibidos desde 1994. La representación gráfica del modelo la podemos ver a continuación:



**Ilustración 1: Modelo de Satisfacción Del Cliente**

El valor del indicador se obtiene del tratamiento de las respuestas de los estadounidenses a un cuestionario telefónico, y se presentan los resultados en 4 niveles:

- Valor del indicador a nivel nacional.
- Valor del indicador en 10 sectores económicos.
- Valor del indicador en 43 industrias diferentes.
- Valor del indicador en más de 200 empresas y agencias del gobierno.

Las fórmulas que cálculo ponderan la nota obtenida en cada una de los componentes del modelo:

**Expectativas del cliente:** las expectativas del cliente son una medida anticipada de la calidad que el cliente espera recibir por los productos y servicios que la organización

ofrece. Son resultado de la publicidad y de un conjunto de mensajes que el cliente asimila, de forma consciente e inconsciente, conformando una idea, “su idea”, sobre lo que le estamos ofreciendo.

**Calidad percibida:** tomando como entrada las expectativas del cliente, la Calidad percibida se considera asociada principalmente a 2 factores: la personalización y la fiabilidad. Las preguntas buscan determinar en qué medida el producto se adapta al cliente, y con qué frecuencia cree que el producto o servicio va a fallar.

**Valor percibido:** este parámetro expresa la relación entre la calidad obtenida y el precio pagado. Una vez decidida la compra, el cliente realiza un balance entre lo que esperaba obtener y lo que ha recibido. Si el balance es negativo, lo más probable es que el cliente no vuelva a repetir la experiencia. Y si lo hace, será porque no le queda más remedio, o porque ha bajado el precio.

**Quejas del cliente:** las quejas son la expresión más palpable de la insatisfacción. Cuanto más satisfecho está un cliente, menos ganas tiene de expresar una queja. Asumiendo esta máxima, calculan este indicador por expresión del porcentaje de personas que manifiestan haberse quejado de cierto producto en un determinado lapso temporal.

**Fidelidad del cliente:** la fidelidad del cliente es el componente crítico del modelo. Observemos que, si bien la satisfacción del cliente ocupa un lugar central en el diagrama, las flechas relacionales desembocan en este parámetro. La fidelidad del cliente es la plataforma de la rentabilidad del negocio.

Clientes fieles son a la organización, lo que sus fans a una estrella del Rock. Para obtener su valor numérico, se pregunta al cliente sobre la probabilidad de que vuelva a comprar el mismo producto y sobre su predisposición a comprar el mismo producto a un precio distinto (tolerancia de precio). El cuestionario se diseña a la medida del modelo, el objetivo es registrar la opinión del cliente sobre cada uno de los aspectos que considera esenciales: qué esperaba obtener, qué ha obtenido, qué valor le ha aportado, qué quejas ha tenido, y la probabilidad de que vuelva a repetir. Aquellos que estén interesados en profundizar más en el tema, les recomiendo que visiten la página de ACSI.

El trabajo que vienen realizando desde hace años es una demostración abrumadora de lo que se puede llegar a hacer con los resultados de una encuesta. Por ejemplo, se ha trabajado muy a fondo sobre los resultados de este indicador y su relación con los principales indicadores económicos del país. Sus resultados postulan el indicador de satisfacción como indicador anticipado de lo que va a suceder en el mundo financiero. La verdad es que los gráficos que aportan son bastante elocuentes. Dan ganas de correr a informarse sobre el estado actual del indicador e invertir en bolsa.

### 4.1.3. CRM: CustomerRelationship Management

CRM, en su traducción literal se entiende como la Gestión sobre la Relación con los Clientes o Administración de las relaciones con los clientes. Este término aplica el proceso completo mediante el cual se construye, monitorea, y mantienen las relaciones con los clientes. Su objetivo es alinear los procesos de negocio con las estrategias de clientes para desarrollar lealtad e incrementar las utilidades a través del tiempo.

Para poder aproximarnos al concepto en cuestión y llegar a su comprensión de la manera más simple posible, vale la pena comenzar aclarando que se trata de una estrategia que se apoya en sistemas de información. Vale la pena aclarar que únicamente el software no resolverá los problemas de la compañía ni incrementara la lealtad o satisfacción de los clientes. Lo que es verdaderamente importante es la correcta estrategia de CRM, y la redefinición de los procesos de la compañía.

Con el fin de comenzar a definir el alcance de esta herramienta, podemos ir definiendo que el verdadero objetivo del CRM, es el manejo adecuado de las relaciones con el cliente que permita a las organizaciones identificar, atraer e incrementar la lealtad de los consumidores más rentables. Bajo este concepto, sería bueno profundizar, ya que las tres palabras del CRM, implican mucho más que lo comentado. Según [Goldenberger], el CRM, incluye los siguientes 10 componentes

1. Funcionalidad de las ventas y su administración.
2. El Telemarketing.
3. El manejo del tiempo.
4. El servicio y soporte al cliente.
5. El Marketing.
6. El manejo de la información para ejecutivos.
7. La integración con el ERP (Enterprise ResourcePlanning).
8. La excelente sincronización de datos.
9. El e-commerce.
10. El servicio en el campo de ventas.

Sin embargo la palabra lealtad, sintetiza prácticamente su significado, ya que CRM se dedica a mantener la lealtad del cliente, específicamente de aquellas cuentas más valiosas.

David Sims, en su artículo “What is CRM?” dice: ”CRM es un término que realmente no es nuevo. Lo que es nuevo es toda la tecnología que permite hacer lo que anteriormente se hacía en las tiendas de barrio. El dueño tenía pocos clientes y suficiente memoria para saber que le gustaba a cada cliente. Lo que hace la tecnología es permitirnos

regresar a ese tipo de modelo”. El CRM no es nuevo, las nuevas tecnologías han permitido su potenciación.

Las empresas que cuenten con esta tecnología, presentaran una ventaja competitiva, respecto a sus competidores, anticipándose a las necesidades de los clientes, a la vez que aumentara el nivel de servicio y generara muchas más oportunidades de contacto entre cliente-proveedor.

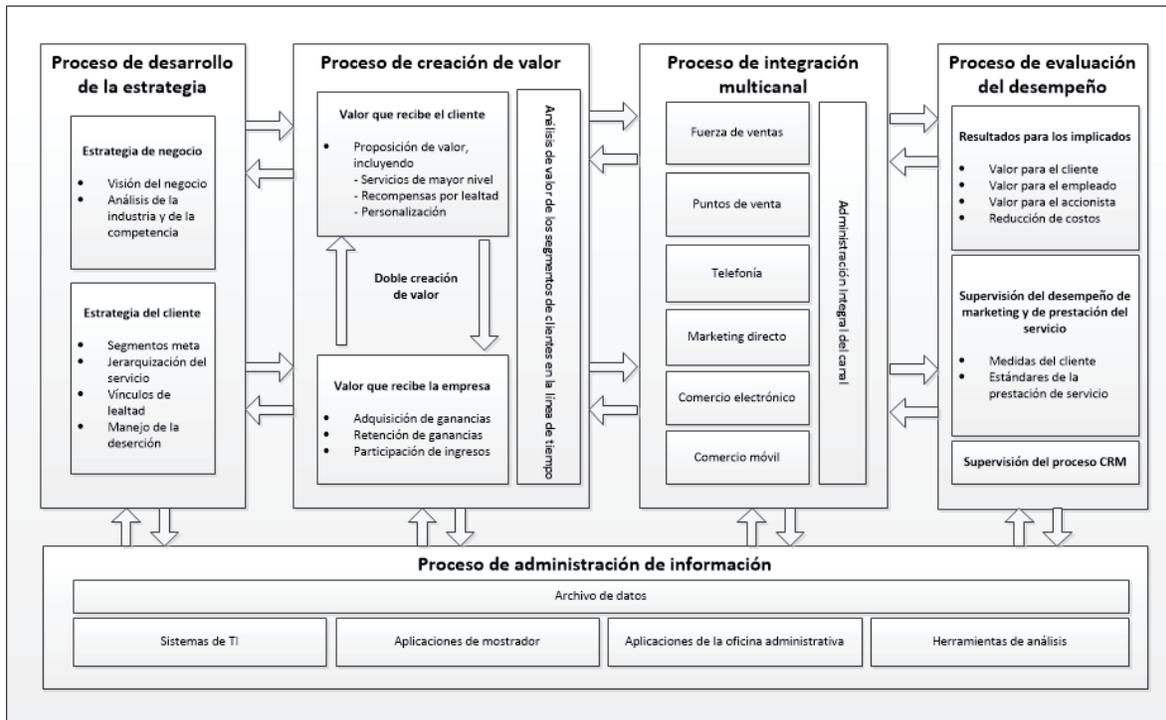
#### **4.1.4. Procesos de Servicio al Cliente**

La implementación de CRM en las empresas es un proceso desafiante. Por desgracia, la mayoría de las implementaciones de CRM fallaron en el pasado. Según GartnerGroup, la tasa de falla es de 55%, y Accenture asegura que es de alrededor de 60%. Una de las principales razones de esta alta tasa de fracasos es que las empresas a menudo consideran que la instalación de un sistema CRM equivale a una estrategia de relación con el cliente y olvidan que el sistema es solo una herramienta para mejorar las capacidades de atención y no la estrategia en si misma.

Implementar CRM en una empresa requiere de la combinación de su estrategia y proceso originales con una estrategia CRM que se aplique de forma transversal en sus procesos de negocio. [Gartner (2001)] aboga por un nuevo enfoque a los procesos de negocio en CRM que involucra tanto repensar cómo estos procesos son percibidos por el cliente y hacer reingeniería para centrarse más en el cliente. [Kale (2004)] apoya esta opinión y sostiene que un aspecto crítico de CRM implica la identificación de todos los procesos estratégicos que se realizan entre una empresa y sus clientes. Para hacer frente a este reto de adoptar un nuevo enfoque a los procesos de CRM, [Payne y Frow] proponen un esquema integrado de cinco procesos fundamentales de una estrategia CRM:

- 1) **El desarrollo de la estrategia:** implica una evaluación de la estrategia de negocio que incluye la misión y visión de la empresa, tendencias del mercado y los competidores. Una vez resuelta, la estrategia de negocios puede guiar el desarrollo de la estrategia del cliente, definiendo los segmentos de clientes, la jerarquización de la base de clientes, el diseño de vínculos de lealtad y el manejo de la deserción.
- 2) **La creación de valor:** traduce las estrategias de negocio y del cliente en proposiciones de valor específicas para los clientes y para la empresa. El valor creado para el cliente incluye todos los beneficios que se entregan a través de los servicios jerarquizados, las recompensas de lealtad, la individualización y personalización. El valor creado para la compañía necesita incluir menores costos de adquisición y retención de los clientes, y una mayor participación de los gastos de estos. Un aspecto principal de CRM es el concepto de doble creación de valor en

donde la participación de ambas partes crea una ganancia mutua. Un ejemplo de esto es la entrega de antecedentes por parte del cliente a la empresa que facilita futuros trámites, aquí el cliente ahorra tiempo y tramites al hacer nuevas solicitudes y a su vez la empresa obtiene información valiosa para el negocio.



**Ilustración 2: Procesos fundamentales de una estrategia CRM [Payne y Frow]**

- 3) **Integración multicanal:** la mayoría de las empresas de servicios interactúan con sus clientes a través de muchos canales, lo cual es un desafío de gestión al atender por todos ellos, así como ofrecer una interfaz unificada que entregue servicios personalizados. La integración de canales CRM aborda este problema.
- 4) **Administración de la información:** la entrega de servicios a través de muchos canales depende de la capacidad de la empresa reunir información del cliente en todos ellos, integrarla con la información relevante y ponerla a disposición del personal de contacto en los diversos puntos de venta. El proceso de administración de la información incluye el almacenamiento de todos los datos del cliente, los sistemas de TI, las herramientas de análisis de datos y los paquetes de aplicación más específicos como manejo de campaña, evaluación de crédito, descripción del cliente y sistemas de alerta de deserción.

- 5) **La evaluación del desempeño:** Debe abordar tres preguntas críticas. En primer lugar, ¿la empresa de CRM está creando valor para sus principales protagonistas, es decir, clientes, empleados, accionistas? En segundo lugar, ¿se están logrando los objetivos del marketing (que van desde la adquisición de clientes, la participación de los gastos de éstos y la retención, hasta la satisfacción de los mismos) y los objetivos del desempeño de la prestación del servicio (como estándares de tiempo de respuesta, tiempo de cierre de llamadas, tasa de resolución en la primera llamada)? En tercer lugar, ¿el proceso de CRM se está desempeñando de acuerdo a las expectativas? El proceso de evaluación de desempeño debe fomentar la mejora continua de la propia estrategia CRM.

## 4.2. Redes Sociales

Las redes son estructuras de sistemas informáticos organizados por módulos interconectados de manera no lineal con capacidad de almacenar y procesar datos.

Las redes sociales son servicios prestados a través de Internet que permiten a los usuarios generar un perfil desde el cual pueden hacer públicos datos e interactuar con otros usuarios y localizarlos en la red en función de las características publicadas en sus perfiles.

Entonces, las redes sociales son estructuras sociales compuestas de grupos de personas, las cuales están conectadas por uno o varios tipos de relaciones, tales como amistad, parentesco, intereses comunes o que comparten conocimientos. En su forma más simple, una red social es un mapa de todos los lazos relevantes entre todos los nodos estudiados. Se habla en este caso de redes sociocéntricas o completas. Otra opción es identificar la red que envuelve a una persona (en los diferentes contextos sociales en los que interactúa); en este caso se habla de red personal.

El modelo de crecimiento de estas plataformas se basa fundamentalmente en un proceso viral en el que un número inicial de participantes mediante el envío de invitaciones a través de correos a los conocidos ofrece la posibilidad de unirse al sitio web. Estas redes sociales se están difundiendo y desarrollando a un nivel exponencial, produciendo un cambio cultural tan vertiginoso que ha revolucionado el uso de Internet.

Redes sociales existen muchas, y pueden clasificarse en base a diferentes criterios (edad, temática, sector, etc). En general, tenemos 2 tipos de redes sociales:

- **Redes sociales horizontales:** Son aquellas que no giran en torno a una temática en concreto, sino que se tratan temas de todo tipo. Como ejemplo claro, podemos encontrar Facebook y Twitter.
- **Redes sociales verticales:** son aquellas centradas en un tema o área. Por ejemplo, en linkedin nos centramos en temas profesionales. MiNube es una red social orientada a viajeros, y en Galgomanía se dan cita virtual los amantes de los Galgos.

Desde la aparición de las redes sociales, la comunicación y los datos disponibles en internet han crecido enormemente. En un intento de afrontar esa situación, las empresas comienzan a utilizar herramientas y estrategias de negocio que permitan gestionar su presencia en internet, a esto se le llama social media. Este ambiente rico en información contiene diferentes fuentes de información que pueden ser tomadas para analizar el comportamientos de los clientes y tomar acciones que produzcan una mayor atracción de clientes a la empresa.

A pesar de ser varias las formas, estructuras, dinámicas y medios que ofrecen las redes sociales como medio de obtener información, se pueden encontrar 5 características principales

- a) **UserInteraction Data:** Datos del usuario que permiten identificarlos y establecer comunicación con él.

**Real-Time Content and Trends:** contenido publicado en tiempo real por los clientes y la empresa, que permiten analizar la tendencia de los clientes y el sentimiento que tienen hacia los productos de la empresa.

- b) **Friends and Relationships:** Datos de las redes de amistad y comunidades de clientes.
- c) **Location Data:** ubicación geográfica del individuo
- d) **Demographic Data:** información sobre el área geográfica a la cual pertenece el cliente, la cantidad de clientes en el mismo sector, en nivel de aceptación de estos clientes a la empresa.

#### 4.2.1. API y Servicios REST

Las redes sociales actuales cuentan con la tecnología necesaria para compartir su información y funciones con otros desarrolladores de aplicaciones, de esta manera, aplicaciones externas a la red social puede utilizar su información, actualizar y publicar contenido y agregar nuevas funciones.

La tecnología que permite la interacción entre aplicaciones se es llamada ApplicationProtocol Interface o API. Esto consiste en que la aplicación que elige compartir su información y funciones con otras, debe definir un conjunto de funciones públicas que abstraen un comportamiento dentro del programa y permiten recibir parámetros.

Existen diversas formas de construir y consumir una API, una de las formas más utilizadas por aplicaciones de nivel empresarial es a través de servicios web o WebServices, esta tecnología pública en la red un conjunto de funciones en formato abierto XML definido en un documento WSDL, de esta manera, otras aplicaciones pueden leer su definición e incorporar sus funciones y estructuras de datos.

Una tecnología más reciente y que se ha impuesto como estándar en las API de redes sociales son los Servicios REST, estos consisten en un conjunto de servicios accesibles mediante el protocolo HTTP que reciben parámetros de tipo GET, POST, PUT y DELETE, que son respondidos con un formato abierto de respuesta pudiendo ser XML o JSON.

Las ventajas de utilizar estas tecnologías por parte de las redes sociales han sido muchas, destacando el aumento de su popularidad al ser utilizadas a través de aplicaciones web, páginas web, otras redes sociales hasta aplicaciones móviles.

Las Desventajas de este método de interconexión de aplicaciones es que deja abierta la posibilidad de intrusos y ejecutar código mal intencionado (malware) que sabotee los sistemas o robe información en caso de no implementarse una metodología y técnica adecuada de permisos y autenticación. Para resolver este problema se ha elaborado una tecnología de autenticación abierta llamada OAuth, que requiere la inscripción de un desarrollador al portal de desarrollo de la red social, para otorgarle una identificación única y una contraseña secreta que permite cifrar la comunicación.

El auge de las redes sociales y el uso de las tecnologías API ha provocado una evolución en las estrategias de negocio y en las tecnologías que apoyan los procesos empresariales, está es una situación actual que está en constante evolución y cambio, planteando nuevos desafíos en tecnologías como CRM y otros.

#### **4.2.2. Social CRM**

Los clientes de hoy quieren ser jugadores activos en el mercado, que desean mantener un diálogo abierto con varias empresas, tener algún impacto en las estrategias de ventas y dar forma a la opinión pública.

Ahora, Social Media ofrece a los clientes estas oportunidades [Schaff y Harris, (2012)]. Los medios sociales son aquellas herramientas sociales y servicios online como: Facebook, Twitter, LinkedIn, YouTube, blogs, wikis, sitios tradicionales y vídeo. Permiten

diferentes personas comunicarse, crear y compartir diversos contenidos, así como para buscar y conectarse con diferentes personas de intereses y opiniones similares [Halligan y Shah, 2010; Greenberg, 2010]. La popularización de los medios sociales significa que los clientes pueden influir en la opinión pública, compartir su satisfacción (o insatisfacción) con los productos y servicios adquiridos. También pueden participar más eficazmente en el proceso de planificación y diseño de productos y servicios. Esta situación crea tareas completamente nuevas para los departamentos de marketing y relaciones públicas [(Kostojohn, Johnson & Paulen, 2011)].

La inclusión de los recursos sociales en la gestión de relaciones con clientes se llama Social CRM o CRM 2.0. Según [Greenberg (2010)] "Social CRM es una filosofía y una estrategia de negocio, con el apoyo de una plataforma tecnológica, reglas de negocio, procesos y características sociales, diseñado para involucrar al cliente en una conversación colaborativa con el fin de proporcionar beneficio de mutuo valor en un ambiente de negocios confiable y transparente". Además, [Deloitte (2011)] destaca que "paradigma de CRM social, se suma al enfoque tradicional de la construcción de la relación y el conocimiento del cliente mediante el análisis de las interacciones bidireccionales (hacia abajo y hacia arriba) y las horizontales, colaborativas, orientados a la comunidad, donde el componentes esenciales son las experiencias y las emociones de los usuarios y las comunidades". A continuación se presenta a modo de tabla las diferencias entre CRM tradicional y CRM social extraído de [Paul & Nilsson,( 2011)]

<b>Parámetro</b>	<b>CRM Tradicional</b>	<b>CRM Social</b>
Roles	Servicio al Cliente	Todos los actores del mercado involucrados
Función	Centrado en el proceso	Centrado en la conversación
Enfoque	Gestión de contactos	Gestión de Comunidades
Canal	Bien definido	Dinámico y en evolución
Valor	Conexión periódica con el cliente	Compromiso sostenido con el cliente
Modelo	Operaciones simples	Relaciones complejas

**Tabla 1: Diferencias entre CRM y CRM Social**

De acuerdo con [Paul y Nilsson (2011)], las organizaciones y los clientes, a través del uso de CRM sociales, pueden obtener muchos beneficios, tales como:

- Una organización puede centrarse en la comunidad y la construcción de relaciones a largo plazo.
- Los clientes son de los autores (dueños) de los contenidos (publicado en sitios de redes sociales) y mantienen el control de su contenido.
- La comunicación no es sólo del tipo de empresa a cliente, sino también de cliente a cliente y el cliente a empresa.
- Los clientes cooperan con la organización directa o indirectamente en el desarrollo y mejora de productos, servicios y la propia atención al cliente;

- El diálogo es menos formal y más "real", cambia el carácter de la conversación sobre la marca para hablar de y a la comunidad.

Es preciso señalar que el CRM social podría ser tratado como un nuevo enfoque de negocio que amplía las capacidades actuales del CRM tradicional, lo que permite a una organización tener en tiempo real, las conversaciones con sus clientes, controlar la expresión de los clientes y animarles a un diálogo constructivo. Sin embargo, la cantidad de información que necesita ser monitoreada, adquirida, recogida, tratada y analizada en el ámbito de los medios de comunicación social, a menudo excede la capacidad de percepción de los trabajadores. Proyectos de CRM social requiere de herramientas que de forma autónoma e inteligente exploren e investiguen los recursos sociales de manera integral.

### **4.3.Necesidades de Mejora a Nivel Nacional**

La entrega de un servicio al cliente que asegure las necesidades de los consumidores esta regularizado en Chile por el Servicio Nacional de Consumidor o SERNAC. Este instrumento del estado está a cargo de las leyes que defienden al consumidor en caso de reclamos con los productos y servicios adquiridos. Una de las formas de certificar el fiel cumplimiento de las leyes es mediante la entrega del sello SERNAC que acredita el cumplimiento de las normas estipuladas en la Ley 19496. Este sello se entre a bancos e instituciones financieras, establecimientos comerciales, compañías de seguros, cajas de compensación, cooperativas de ahorro y crédito y otros proveedores de servicios crediticios.

Una de las métricas que mide esta ley al momento de resolver controversias es el tiempo de respuesta en que tardan las empresas en responder a las necesidades del cliente. Esto queda especificado en la siguiente cita:

“Artículo 56.- El servicio de atención al cliente requerido para dar cumplimiento a la condición dispuesta en el número 2 del inciso primero del artículo 55 será organizado por los proveedores indicados en este Título, en forma exclusiva o conjunta, y será gratuito para el consumidor que haya suscrito un contrato de adhesión de los señalados en el inciso segundo del artículo 55, con un proveedor que cuente con el sello SERNAC.

El servicio de atención al cliente deberá responder fundadamente los reclamos de los consumidores, en el plazo de diez días hábiles contado desde su presentación. Esta respuesta se comunicará al consumidor por escrito o mediante cualquier medio físico o tecnológico y se enviará copia de ella al Servicio Nacional del Consumidor.

El proveedor deberá dar cumplimiento a lo señalado en la respuesta del servicio de atención al cliente en el plazo de cinco días hábiles, contado desde la comunicación al consumidor.

En caso de incumplimiento de las obligaciones indicadas en los dos incisos anteriores, el Servicio Nacional del Consumidor deberá denunciar al proveedor ante el juez de policía local competente, para que, si procediere, se le sancione con una multa de hasta cincuenta unidades tributarias mensuales, sin perjuicio del derecho del consumidor afectado para denunciar el incumplimiento de las obligaciones referidas.”

El cumplimiento a cabalidad de esta norma es una métrica difícil de medir dado a la variada cantidad de canales de comunicación de redes sociales que se encuentran hoy en día, complicando más la situación cuando una incidencia puede originarse en un canal informal como Facebook y ser resuelto en un medio formal como Correo electrónico o de forma presencial.

La necesidad de adaptarse a estas tecnologías y seguir cumpliendo con las normas y compromisos de calidad de servicio es una demanda hoy visible en los mercados digitales del país. Esto queda explícito en las diversas asesorías, servicios y herramientas que son solicitadas a en el portal de compras públicas ChileCompra. Dentro de las necesidades encontradas podemos destacar las siguientes:

- Asesoramiento en definición de estrategias de marketing digital
- Generación de contenidos para redes sociales y portales empresariales
- Medición de métricas de los comentarios de clientes en redes sociales
- Administración de perfiles en redes como Facebook, Twitter y Youtube
- Herramientas de administración que permitan centralizar los mensajes, administrar a múltiples usuarios y medir el desempeño y tiempo de respuesta.

A continuación se muestra un detalle de las licitaciones estudiadas en donde se encuentran presentes estas demandas.

<b>Licitación</b>	<b>Organismo</b>	<b>Proyecto</b>
591-42-LP14	FONASA	Servicio de Elaboración e Implementación de Estrategia de Redes Sociales de Fonasa
Fonasa requiere difundir la labor e información relativa a Fonasa y sus asegurados. Responder consultas, mejorar la experiencia del servicio y potenciar la marca Fonasa, a través de sus canales digitales actuales y evaluar la incorporación de nuevos canales digitales, como por ejemplo Youtube o Instagram u otros, cuando corresponda		
877-22-LE14	INTENDENCIA METROPOLITANA	MONITOREO DE REDES SOCIALES PARA INTENDENCIA RM.
La Intendencia Metropolitana requiere contratar los servicios monitoreo de las redes sociales y plataformas digitales de carácter público, con el propósito de insumo institucional. Se requieren herramientas de monitoreo de redes sociales definidas en contrato, generación de reportes y herramientas de búsqueda.		
681563-79-LE15	METRO S.A.	GESTIÓN DE REDES SOCIALES A TRAVÉS DE APLICACIÓN WEB
<p>Metro Requiere:</p> <p>Contestar mensajes recibidos en los perfiles de Metro de Santiago de Twitter y Facebook.</p> <p>Categorizar los mensajes recibidos de acuerdo a las categorías definidas por Metro.</p> <p>Categorización de los mensajes en función del sentimiento de las menciones.</p> <p>Manejar ficha de cliente para dar seguimiento al caso (con casillas para nombre del cliente, correo electrónico y teléfono).</p> <p>Derivar mensajes vía correo electrónico.</p>		

**Tabla 2: Licitaciones Públicas**

Con esto queda demostrado el interés actual del mercado en herramientas que mejoren sus procesos actuales y permitan estar aún más alerta a las necesidades y mensajes recibidos por los clientes. Para dar paso a una solución de estas necesidades, debemos plantear el problema a abarcar en este proyecto.

## 5. Definición del Problema

Considerando la internet como el mayor medio de comunicación actual, con ambiente rico en canales de comunicación a través de redes sociales, en el que cada individuo cuenta con múltiples cuentas o perfiles según las redes sociales que son de su

preferencia, se publica continuamente nuevo contenido, opiniones y preferencias, que definen la opinión de un grupo de individuos en una localidad determinada. Se presentan varios inconvenientes a la hora de desear unificar la comunicación en un medio que capte diferentes canal:

- a) ¿Cómo se vinculan diferentes medios de comunicación, con diferentes estructuras y datos, en un modelo que asocie sus características en común?
- b) ¿Cómo se identifica a una persona con diferentes perfiles en diferentes redes sociales y canales de comunicación?
- c) ¿Qué información es más relevante para definir las preferencias de una persona y poder entregar un trato personalizado?
- d) ¿De qué manera se pueden agregar con el tiempo nuevos canales de comunicación a una herramienta de unificación de canales?
- e) ¿Cómo se gestiona una conversación que puede iniciarse en un canal público y terminar en un canal privado?

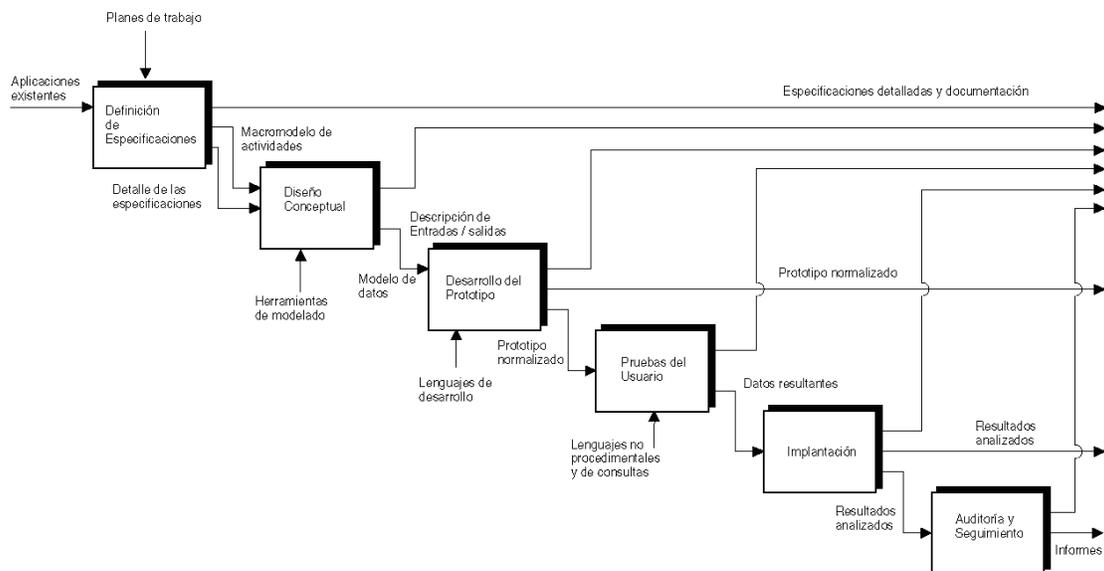
En resumen se necesita de un método que gestione la identidad de las personas en la red, que muestre información relevante y personalizada, que permita gestionar una comunicación en diferentes canales y que permita adaptar nuevos canales con el paso del tiempo.

## **6. Metodología de Desarrollo**

Para abordar un problema que requiere un estudio de múltiples tecnologías de comunicación y su integración en una única herramienta, se deben plantear las siguientes necesidades en el desarrollo:

- 1) Se debe trabajar en la elaboración de un componente general que interconecte y estandarice todas las tecnologías específicas bajo un mismo conjunto de funciones, es decir, se debe avanzar en anchura a lo largo de todos los canales de comunicación a integrar.
- 2) Es necesario trabajar cada canal de comunicación de forma específica, adaptando sus funciones y su estructura al componente general, es decir, se debe trabajar en profundidad.
- 3) Cada componente interconectado puede afectar el uso o avances en otros componentes.

Bajo estas necesidades se hace evidente que los avances deben ser capitalizados rápidamente a través de prototipos, que permitan cerrar requisitos, aprender y a contribuir a la construcción general del software. Es por esto que se decidió utilizar una Metodología de Prototipado Rápido (MPR) que permite desarrollar modelos de aplicaciones de software que permiten ver la funcionalidad básica de la misma, sin necesariamente incluir toda la lógica o características del modelo terminado.



**Ilustración 3: Metodología de Prototipado Rápido**

Como se ha expuesto, la idea fundamental de MPR es el desarrollo de prototipos. Un prototipo es un modelo inicial de lo que al final se corresponderá con la interfaz

definitiva junto con las funcionalidades asociadas. Este prototipo se someterá a pruebas para comprobar su funcionalidad, de las que surgirán modificaciones que darán origen a un segundo prototipo, versión mejorada y posiblemente ampliada del primero, el cual se volverá a probar, repitiéndose sucesivamente el proceso hasta alcanzar el prototipo definitivo. Las fases de este modelo de desarrollo se explican a continuación:

**Definición de Especificaciones:** Esta primera fase tiene por objetivo auditar la información relativa al problema con el fin de recabar todos los datos para su solución.

**Diseño Conceptual:** El objetivo de esta fase es construir un modelo de información que refleje el esquema conceptual del prototipo.

**Desarrollo del Prototipo:** Esta fase tiene por objetivo la construcción del primer prototipo operativo de la aplicación.

**Pruebas del Usuario:** En esta fase se realizarán todas las pruebas para validar el prototipo desarrollado en la fase anterior.

**Implantación:** En esta fase se ejecutará el plan de formación de los usuarios y se llevará a cabo el proceso de migración al entorno de ejecución real de la aplicación.

**Auditoria y Seguimiento:** Consiste en realizar una auditoría del rendimiento y la calidad de la aplicación, y de determinar y canalizar los mecanismos necesarios para realizar peticiones de modificación y para que estas sean llevadas a cabo por los equipos de mantenimiento.

## 7. Análisis del Sistema

El modelo de captura y estímulos multicanal de mensajería será redefinido por el nombre de aplicación Perfilink, los nombramientos de Perfilink en el documento harán referencia a los modelos, diseños y al sistema que este proyecto busca implementar.

El sistema se describe de forma general como un integrador de múltiples plataformas de comunicación e interacción social, por lo cual debe contar con las funciones de representar al usuario y a sus contactos de los canales de comunicación y soportar diversos formatos de mensajes, al existir numerosos medios de comunicación es vital delimitar el alcance del proyecto para su oportuno cierre u evaluación. También se ha descubierto la necesidad de definir políticas y términos de servicio para entregar las funcionalidades del sistema a los usuarios y utilizar responsablemente sus datos, estos requisitos se detallaran a continuación.

### 7.1. Alcance

El alcance del proyecto se delimita a crear un medio de comunicación 1 a 1 entre un usuario y sus contactos, en redes sociales y canales de comunicación que posean una interfaz de programación de aplicaciones API publica y bien mantenida, presentando las funciones de agrupar los mensajes provenientes de diversos medios y soportar diversos mensajes de comunicación, así como también permitir la extracción de información personal publica en medio sociales con un esquema que unifique la información de los diversos canales. Los canales objetivos a integrar son Twitter, Email y Google+, por disponer de medios de conexión fáciles de mantener y conectar. Los canales de comunicación de otras redes sociales que son más complicadas o mal mantenidas como Facebook y Whatsapp se consideraran en el diseño pero su implementación dependerán de que las condiciones para conectar a estos medios sean más favorables.

### 7.2. Requerimientos Funcionales

De forma breve se detallaran las funciones del sistema, estas pueden verse en más detalle a través del diagrama de casos de uso o bien el diagrama de clases:

- Manejo de Cuentas de Usuario: se registraran usuarios en el sistema con datos que permitan su identificación
- Manejo de Canales: el sistema registrara y conectara con canales de comunicación externos.
- Manejo de Perfiles: los perfiles son la representación de usuarios en un canal específico, el sistema deberá manejar y extraer la información específica y efectiva de cada canal integrado.

- **Manejo de Mensajes:** el sistema deberá manejar el envío y recepción de mensajes de los diversos canales soportando diferentes formatos disponibles.
- **Manejo de Contactos:** el sistema dispondrá de funciones que permitan identificar y agrupar los perfiles que pertenezcan a una misma persona.
- **Manejo de Conversaciones:** se agruparan los mensajes que tengan un tema y un contacto en común a través de conversaciones de mensajes multiplataforma.

## 7.3 Requerimientos No Funcionales

Se definen además requerimientos de calidad del sistema que son visibles de forma transversal o bien de forma añadida a los requerimientos antes citados.

### 7.3.1 Performance

El sistema al estar conectado con plataformas externas para extraer datos, sufre de variaciones y condiciones del medio a las cuales debe responder en forma oportuna y mitigar el riesgo, es por esto que se presentan los siguientes requerimientos del al sistema:

- **Gestión de Sesiones:** debe gestionar la conexión y estado de las sesiones externas a través del uso de cookies y otros medios de identificación.
- **Manejo de Errores y Excepciones de Conexión:** debe responder ante la pérdida de conexión o bien ante el cierre o actualización de los servicios de conexión.
- **Manejo de recursos externos:** las consultas a medios externos suelen estar limitadas en número a lo largo de un periodo de tiempo, el sistema debe avisar sobre esta causa al usuario.

### 7.3.2. Términos de Servicio

El sistema debe presentar un documento de términos de servicio que entregara al usuario de forma abierta, de tal modo que tanto los usuarios como los prestadores de canales externos puedan evaluar las condiciones a las que se compromete el usuario del programa.

### 7.3.3. Política de Privacidad

El sistema debe presentar un documento de política de privacidad de forma abierta al público explicando la gestión responsable y no maliciosa de los datos, así como también las consecuencias del uso y gestión de la información y los permisos entregados al sistema para su uso.

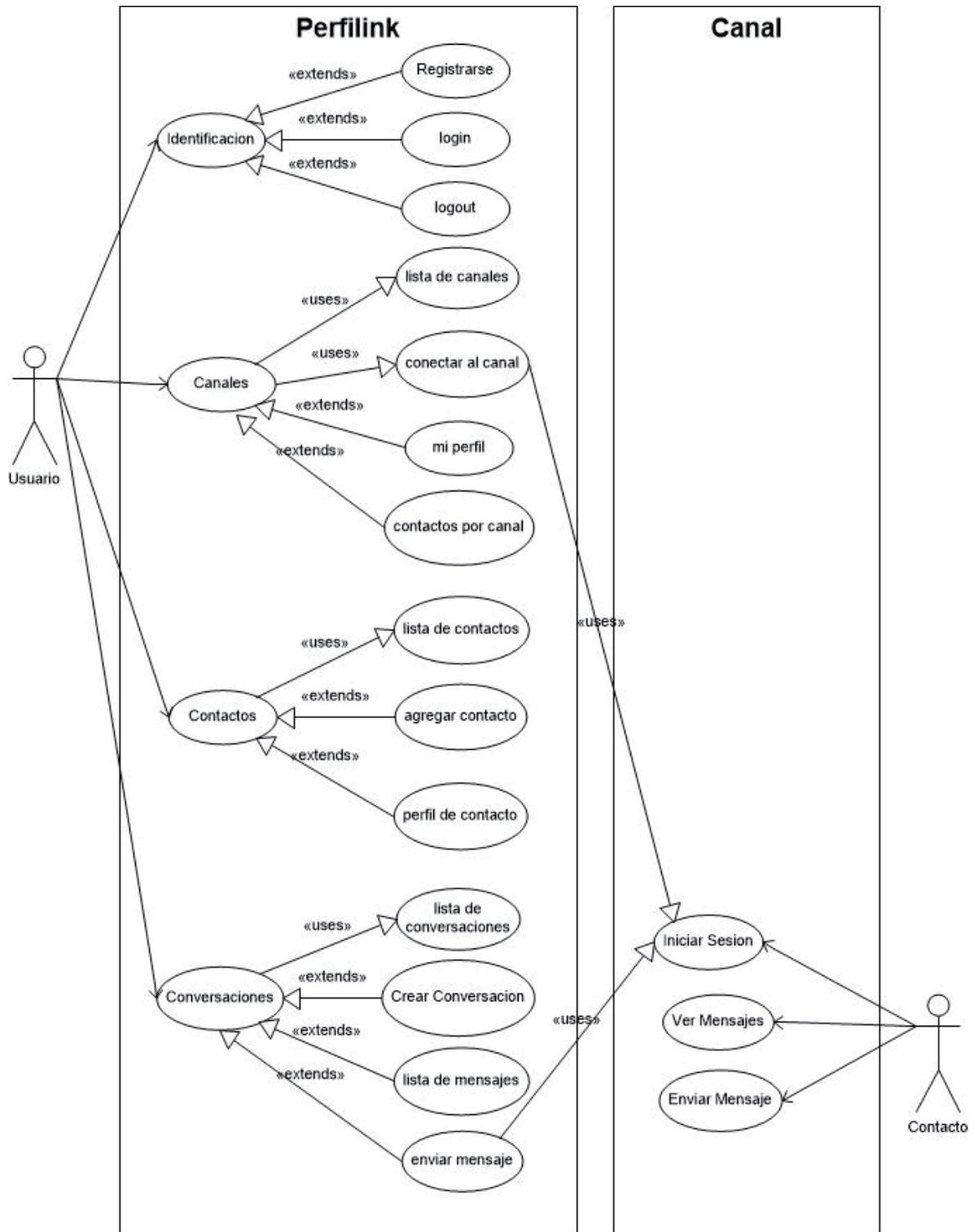
## 8. Diseño del sistema

Para representar los componentes del sistema de una manera clara y fácil de implementar se ha decidido utilizar notación UML 2 con diagramas de casos de uso y diagramas de clases.

### 8.1. Casos de Uso General

Se muestra el caso de uso general del sistema que muestra dos sistemas que interactúan internamente entre sí, por un lado el sistema perfilink utilizado por la empresa o el usuario, y por otro, el canal de comunicación que usa el cliente, pudiendo ser cualquier canal conectado al sistema. Las funciones son bien intuitivas por lo cual se detallara solo los principales casos de uso:

- Identificación: el usuario registrara sus datos en el sistema para iniciar o cerrar sesión a través de un login o logout respectivamente.
- Canales: se mostrara el listado de canales disponibles, dado la posibilidad de conectar o desconectar con él. Desde estos canales se puede extraer la información del perfil del usuario y los contactos asociados a este perfil.
- Contactos: se listaran los contactos registrados, se mostrara su perfil por canal y se contara con las funciones de añadir nuevos contactos a partir de los perfiles disponibles en cada canal.
- Conversaciones: se mostraran las conversaciones realizadas con los contactos, mostrando los mensajes que componen estas conversaciones, permitiendo responder a través de un nuevo mensaje.



**Ilustración 4: Diagrama de Casos de Uso**

## 8.2. Modelo de Clases

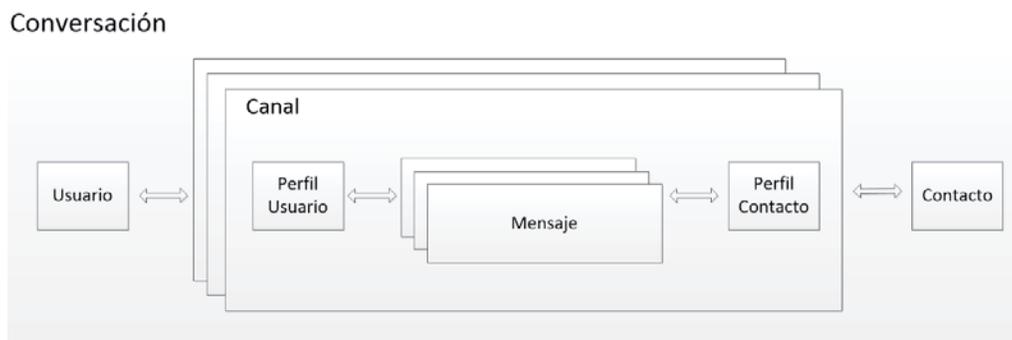
Modelo de clases describe los objetos, atributos, operadores, interfaces y sus relaciones. Para poder comprender este conjunto de modelos se presentara en primer lugar un diagrama de clases general que muestra las clases y sus relaciones entre ellas, luego se mostraran las clases en detalle y sus relaciones cercanas.

Antes de ver el modelo, es vital repasar los siguientes conceptos [Bell] que han ayudado a simplificar la tarea de modelado:

- **Clases abstractas, clases e interfaces:** las clases abstractas poseen atributos y métodos que deben ser heredados e implementados por una clase hija, esta clase por sí sola no se puede instanciar pero se puede utilizar para agrupar elementos que tiene la misma herencia. Las clases normales si permiten instancia y sus atributos y métodos pueden ser públicos o privados, estas clases pueden extender su funcionalidad y compartirla en común con otras clases a través de un contrato llamado interfaz. Las interfaces son conjuntos de métodos abstractos y públicos, es decir, la clase que los implemente debe definir sus operaciones internas y cumplir.
- **Clases con Parámetros:** las clases con parámetros reciben parámetros antes de su declaración y posterior instancia, estos parámetros pueden definir su nombre, tipos de datos y más.
- **Funciones Getter y Setter:** estas funciones definen la visibilidad y modificación de atributos respectivamente. Se instancian de la forma: `get_nombreatributo()` y `set_nombreatributo()`.

## 8.3. Modelo de Integración de Canales

En función del problema explicado, se desarrolló un modelo abstracto de comunicación que identifica a los dos involucrados, el usuario y su contacto, los cuales utilizan múltiples canales de comunicación en los que cada uno tiene uno o muchos perfiles dentro del canal, en donde se envía y recibe mensajes.



**Ilustración 5: Modelo de Integración de Canales**

El desafío que en este modelo viene dado por los medios y estructuras que disponga cada canal de comunicación para ser integrado y extendido en otros sistemas, requiriendo un mapeo de las estructuras de datos a un modelo de datos estándar.

Afortunadamente, los avances tecnológicos y la necesidad de interconexión ha promovido la aparición de API (Application Programming Interface) que definen un conjunto de funciones a modo de contrato que otras aplicaciones externas pueden utilizar para obtener y enviar datos. Una API ampliamente utilizada para conectar a los servidores de correo electrónico es IMAP, esta permite obtener la bandeja de entrada del usuario autenticado, explorar las carpetas de correo y enviar mensajes de correo electrónico.

Uno de los estándares actuales para implementar API son los servicios REST, estos servicios utilizan las rutas del protocolo HTTP y sus diferentes cabeceras (GET, POST, PUT, DEL) para enviar y recibir información en formato XML, en la actualidad se pueden encontrar API REST en las principales redes sociales como Graph API de Facebook, Social API de Google+, Twitter API, LinkedIn API, Spotify API, entre muchos otros. Si bien REST es un estándar efectivo y ampliamente usado para la comunicación entre aplicaciones, los datos enviados a través de él no están normalizados, es aquí donde reside la necesidad de analizar cada estructura y crear un medio de mapeo de los datos a un modelo de datos de perfiles, mensajes y funciones estándar.

Para responder a esta necesidad se creó en un trabajo anterior un modelo de datos que define cada componente en sus atributos y funciones junto con sus relaciones a través de un diagrama de clases, estas clases permiten ser heredadas unas a otras, definiendo un nivel de jerarquía que va de las clases más abstractas a las más específicas, esta característica permite trabajar con la implementación en detalle que requiere cada canal de comunicación.



## 9. Arquitectura del modelo multicanal

Para probar el modelo se decidió crear un prototipo funcional que disponga tanto de las características de programación orientada a objeto (POO), así como también disponga de las librerías a API que permitan conectar con los demás canales de comunicación. Es por esto que se eligió el lenguaje PHP como base por ser un lenguaje flexible, que incluye orientación a objeto y que dado a su gran popularidad y buen desempeño en crear aplicaciones web, cuenta con amplias librerías de API que son mantenidas por la comunidad de programación y cuentan con documentación para su integración y uso libre. Aquí se muestra una ficha con las características generales del proyecto:

Ficha de Proyecto	
Nombre:	Perfilink
Objetivo:	Crear un prototipo de plataforma web que implementa un modelo de integración de canales
Lenguaje:	PHP 5.4.27
Framework:	Laravel 4.2 / HybritAuth 2.0
Patrón de diseño:	MVC + Routes
Base de datos:	Mysql
Complementos:	-Jquery -Bootstrap -Eloquent ORM
API:	-php_imap -twitterAPIExchange -facebooksdk v4
Link:	<a href="http://perfilink.cl">http://perfilink.cl</a>

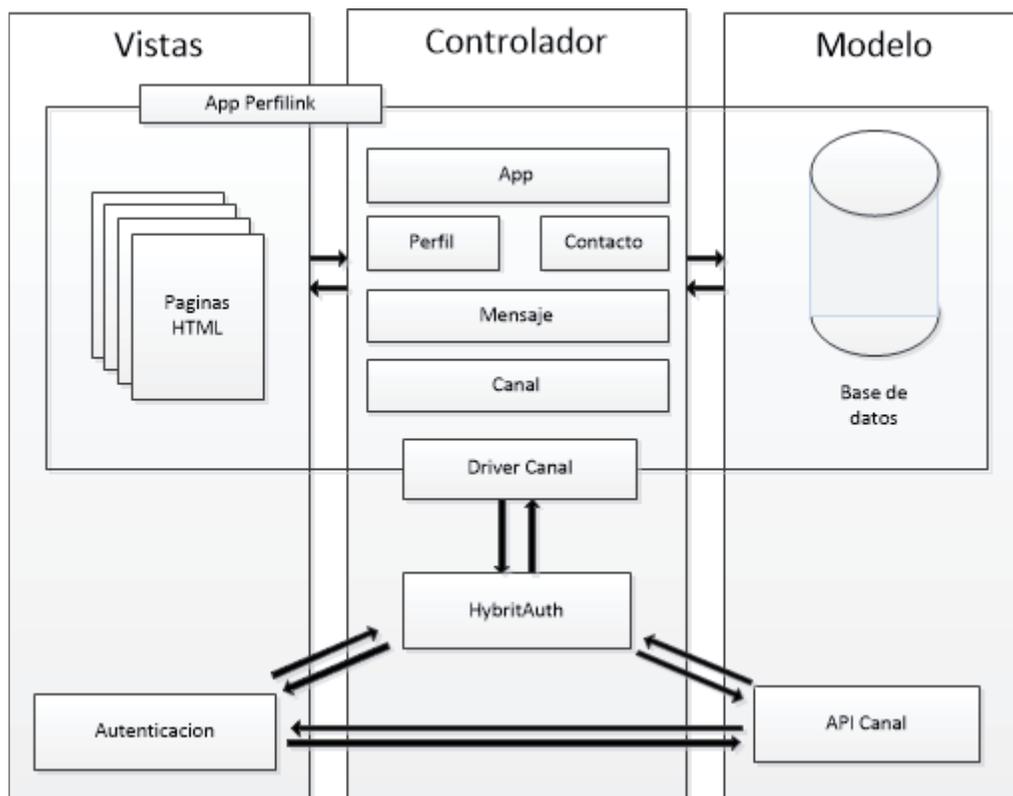
**Tabla 3: Ficha del Proyecto**

Con este conjunto de tecnologías se pretende crear una aplicación web que aplique el modelo de integración multicanal y permita hacer pruebas de su funcionamiento y una evaluación de su valor y aporte al área de servicio al cliente.

A continuación se presentara la arquitectura que representa la aplicación de manera abstracta y permite entender la combinación de sus componentes. Además se entrara en detalle en aquellos elementos que son el núcleo de la integración. Luego, se presentara un catálogo de los elementos a implementar, su descripción y su nivel de avance a la fecha.

## 9.1. Arquitectura de Software

Para implementar el modelo se escogió el patrón de arquitectura MVC (Modelo – Vista - Controlador) que separa los componentes de la aplicación en las vistas que manipula el usuario, el controlador que responde a los eventos del usuario y gestiona el modelo de datos y las vistas, y por último el modelo que representa los datos relacionados entre sí en una base de datos o algún otro almacén de datos. Ésta arquitectura se ha convertido en un estándar para aplicaciones empresariales y esta soportado por la mayoría de framework disponibles.



**Ilustración 7: Arquitectura de Software**

Como se observa en la imagen, la aplicación cuenta con diferentes vistas html que muestran la información al usuario, esta información es manejada por los componentes de la capa controlador. Cada componente controlador maneja las acciones sobre el modelo de datos que administra y la relación con otros componentes, por ejemplo, el controlador Contacto permite mostrar, crear, editar y eliminar un contacto, así como también mostrar un listado filtrado de los contactos. El funcionamiento de cada controlador es similar en naturaleza con los otros controladores y siguen el modelo de clases presentado anteriormente.

El componente que separa a esta aplicación del resto de sistemas transaccional reside en el componente Driver Canal, que se encarga de conectar los componentes dentro del límite de la aplicación con las interfaces de las aplicaciones externas.

## 9.2. Driver Canal

El componente Driver Canal es el elemento esencial para la integración de los canales de comunicación. Su definición es vital para entender el funcionamiento del modelo ya que este funciona como un intermediario entre los controladores que manejan las vistas y modelos de datos; y las API externas al sistema correspondiente a cada canal de comunicación integrado. Es por esto que debe implementarse un componente Driver Canal para cada canal integrado en el sistema, teniendo como resultado las clases CanalTwitter, CanalEmail, CanalFaceook. Las 3 funcione esenciales de este componente son:

1. **Autenticación:** Identifica la identidad del usuario en el canal y se encarga de mantener la sección activa mientras el usuario este presente.
2. **Mapeo:** Permite tomar las funciones y estructuras de datos de la API y traducirlas a funciones y modelos de datos ya definidos por los componentes de software.
3. **Sincronización:** Compara los elementos almacenados en el modelo local de la aplicación, con los datos obtenidos por el API para actualizar los registros con nuevos datos y ajustar las modificaciones realizadas a los elementos ya guardados. Para esta tarea es vital que en mapeo del api exista un medio de identificación de cada perfil y mensaje por cada canal de comunicación.

Entendiendo estas funciones esenciales se puede construir un conjunto de funciones específicas para que los componentes de software utilicen las funciones de cada API sin necesidad de implementar cambios drásticos en los componentes del sistema MVC, abstrayendo el mapeo a una simple llamada de función. Acá se muestra una tabla con las funciones específicas a implementar en el Driver Canal:

1	Estado	Devuelve el estado del canal, verificando la disponibilidad y la autenticación.
2	Conexión	Devuelve un esquema de datos o formulario definiendo los datos requeridos para conectar con el canal a través de un perfil de usuario.
3	Conectar	Recibe el esquema de perfil de usuario enviado en Conexión, con los datos de autenticación, valida los datos, si estos son válidos, se asegura mantener en variables de sesión los datos para realizar consultas al canal.
4	Desconectar	Recibe el identificador de perfil de usuario y desconecta su sesión de la aplicación, eliminando los datos de sesión.
5	Sincronización de Perfil de Usuario	Una vez conectado, se verifican los datos personales del usuario y se actualizan sus registros.
6	Sincronización de Perfiles de Contacto	Se realiza una búsqueda de los contactos que posee el usuario en este canal, se identifica a cada uno y se compara con los registros almacenados en el sistema, de no existir se agregan al modelo, si poseen cambios se actualiza.
7	Sincronización de Mensajes	Se realiza una búsqueda de los mensajes enviados y recibidos en el canal. Se crean los registros nuevos de mensajes no leídos, mensajes sincronizados enviados y recibidos.
8	Agregar Perfil de Usuario	Permite guardar un perfil de usuario al modelo para conectar con el canal. Requiere de un identificador único del perfil en el canal, la contraseña no es requerida pero será solicitada al pedir una conexión.
9	Agregar Perfil de Contacto	Permite agregar un nuevo contacto del canal a los registros de contacto, se valida su existencia en el canal y se mapea su identificación y datos personales en el modelo.
10	Ver Perfil de Contacto	Recibe los datos de identificación del contacto y busca la información de su perfil
11	Crear Mensaje	Devuelve un esquema con los datos necesarios para definir un mensaje en el canal.
12	Enviar Mensaje	Recibe un esquema enviado por Crear Mensaje y valida su estructura y destinatario, si todo es válido se envía el mensaje por el canal.
13	Recibir Mensajes	Recibe los datos un contacto y obtiene los mensajes entrantes para así mantener una conversación activa con el contacto.

**Tabla 4: Driver Canal**

A partir de esta tabla se puede definir claramente el trabajo a realizar con cada API.

### 9.3. Integración de API

Acá podemos comparar la concordancia de cada api disponible con el módulo de integración de canales, de esta manera se puede comprobar cuales elementos están disponibles a ser utilizados y cuales debes ser mitigados o trabajados de manera indirecta. Un ejemplo de este análisis se presenta a continuación para la el canal IMAP:

N°	Función Driver Canal	FunciónImap
1	Estado	Imap_open
2	Conexión	Se aplica indirectamente a partir de Imap_open
3	Conectar	Imap_open
4	Desconectar	Imap_close
5	Sincronización de Perfil de Usuario	Se aplica indirectamente a partir de imap_header
6	Sincronización de Perfiles de Contacto	Se aplica indirectamente a partir de imap_header
7	Sincronización de Mensajes	Imap_search imap_header imap_body imap_feachstructure
8	Agregar Perfil de Usuario	Se aplica indirectamente a partir de Imap_open
9	Agregar Perfil de Contacto	-
10	Ver Perfil de Contacto	-
11	Crear Mensaje	imap_mail_compose
12	Enviar Mensaje	Imap_mail
13	Recibir Mensajes	Imap_search

**Tabla 5: Integración Driver Canal**

Este formulario se puede aplicar a las diferentes API disponibles en las redes sociales, se pretende utilizar en el análisis de Twitter, Facebook, LinkedIn, WhatsAPI, Skype, y Spotify.

Para llevar a cabo la tarea de implementar cada Driver Canal, se utilizara un API que unifica el comportamiento en común de otras API para autenticar usuarios y obtener acceso a su API, en esta etapa entra en acción HybritAuth.

## 9.4. HybritAuth

HybritAuth es componente de software que facilita la autenticación, la obtención del perfil, la conexión con una API y el envío y recibo de mensajes de está. Se encuentra

disponible en PHP y cuenta con soporte para un gran número de redes sociales y aplicaciones. Entre sus principales proveedores de autenticación están OpenId, Facebook, Twitter, Google, LinkedIn, Yahoo y Windows Live. Es mantenida por una comunidad OpenSource en sourceforge.net y está en constante actualización.

Esta API funciona se integra a otro software a partir de la interacción con 3 clases Principales:

## 1) Hybrit\_Auth

Clase principal que inicia la configuración de las aplicaciones registradas en la red y ejecuta la autenticación por cada canal.

Métodos:

- 1.1) **Hybrit\_Auth(\$config):** Constructor de clase, recibe una arreglo o string con la configuración de las aplicaciones definiendo sus claves publicas y claves secretas, un ejemplo se muestra en la siguiente ilustración:

```
1. $config_array = array(  
2.     "base_url" => "http://mywebsite.com/path/to/hybridauth/",  
3.     "providers" => array (  
4.         "Twitter" => array (  
5.             "enabled" => true, "keys" => array ( "key" => "", "secret" => "" )  
6.         ),  
7.         "Facebook" => array (  
8.             "enabled" => true, "keys" => array ( "id" => "", "secret" => "" )  
9.         ),  
10.        "OpenID" => array (  
11.            "enabled" => true  
12.        )  
13.    )  
14. );  
15.  
16. $hybridauth = new Hybrid_Auth( $config_array );
```

**Ilustración 8: ConfiguraciónHybritAuth**

- 1.2) **Authenticate(\$provider,\$params):** Método que gestiona la autenticación OAuth entre el usuario y el proveedor señalado en \$provider, está a cargo de redireccionar a la página de autenticación externa, recibir la respuesta en la página endpoint y continuar con el flujo del programa devolviendo un objeto Hybrid\_Provider\_Adapter que representa el canal de conexión.
- 1.3) **getAdapter(\$provider):** Devuelve una instancia de la clase Hybrid\_Provider\_Adapter correspondiente al canal señalado en \$provider.

- 1.4) **isConnectedWith(\$provider):** Devuelve true o false si se ha iniciado sesión con el proveedor señalado.
- 1.5) **getConnectedProviders():** Devuelve un arreglo con todos los canales que se encuentren conectados.

## 2) Hybrid\_Provider\_Adapter

Clase de envoltorio que gestiona las operaciones de autenticación particulares para cada proveedor o canal que se encuentre soportado.

Métodos:

- 2.1) **isUserConnected():** Devuelve true o false si se ha iniciado sesión con el proveedor.
- 2.2) **getUserProfile():** Devuelve la clase Hybrit\_User\_Profile que encapsula los datos del usuario,
- 2.3) **api():** Devuelve un objeto que encapsula la API para así enviar mensajes URL y hacer peticiones JSON.
- 2.4) **logout():** Cierra la sesión y elimina los datos temporales de autenticación.

## 3) Hybrit\_User\_Profile

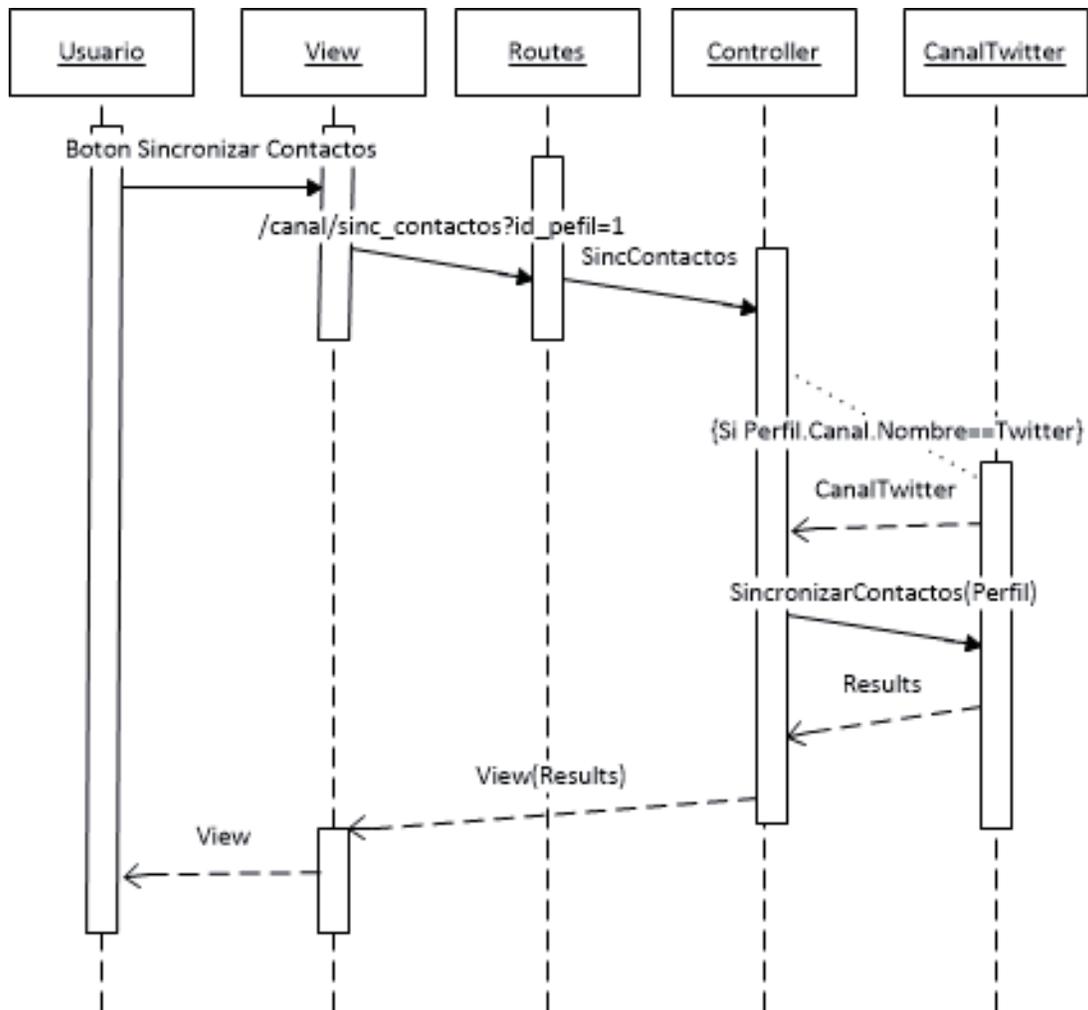
Clase que encapsula los datos del usuario, tiene un conjunto de atributos que pueden estar o no estar soportados por el proveedor.

A través de estos componentes, HybritAuth provee de funciones estándar para que los usuarios puedan identificarse en los sistemas por medio de 32 medios distintos de autenticación. Facilitando la incorporación de datos ya existentes en la red con la identificación propia de cada sistema.

## 9.5. Secuencias de Sincronización

A modo de ejemplificar como se integran todos los componentes de la arquitectura señalada, se muestra a continuación la secuencia de mensajes entre los objetos que

permiten sincronizar los mensajes de uno de los canales, con el modelo de datos del sistema:



**Ilustración 9: Secuencia Sincronizar Contactos MVC**

La secuencia de la Ilustración 5 muestra como los componentes del modelo MVC pueden cooperar entre sí para organizar las tareas, separa los componentes visuales genéricos que interactúan directamente con el usuario de los componentes que desarrollan el control y el trabajo específico de la sincronización por cada canal. Los objetos involucrados en esta secuencia se definen como:

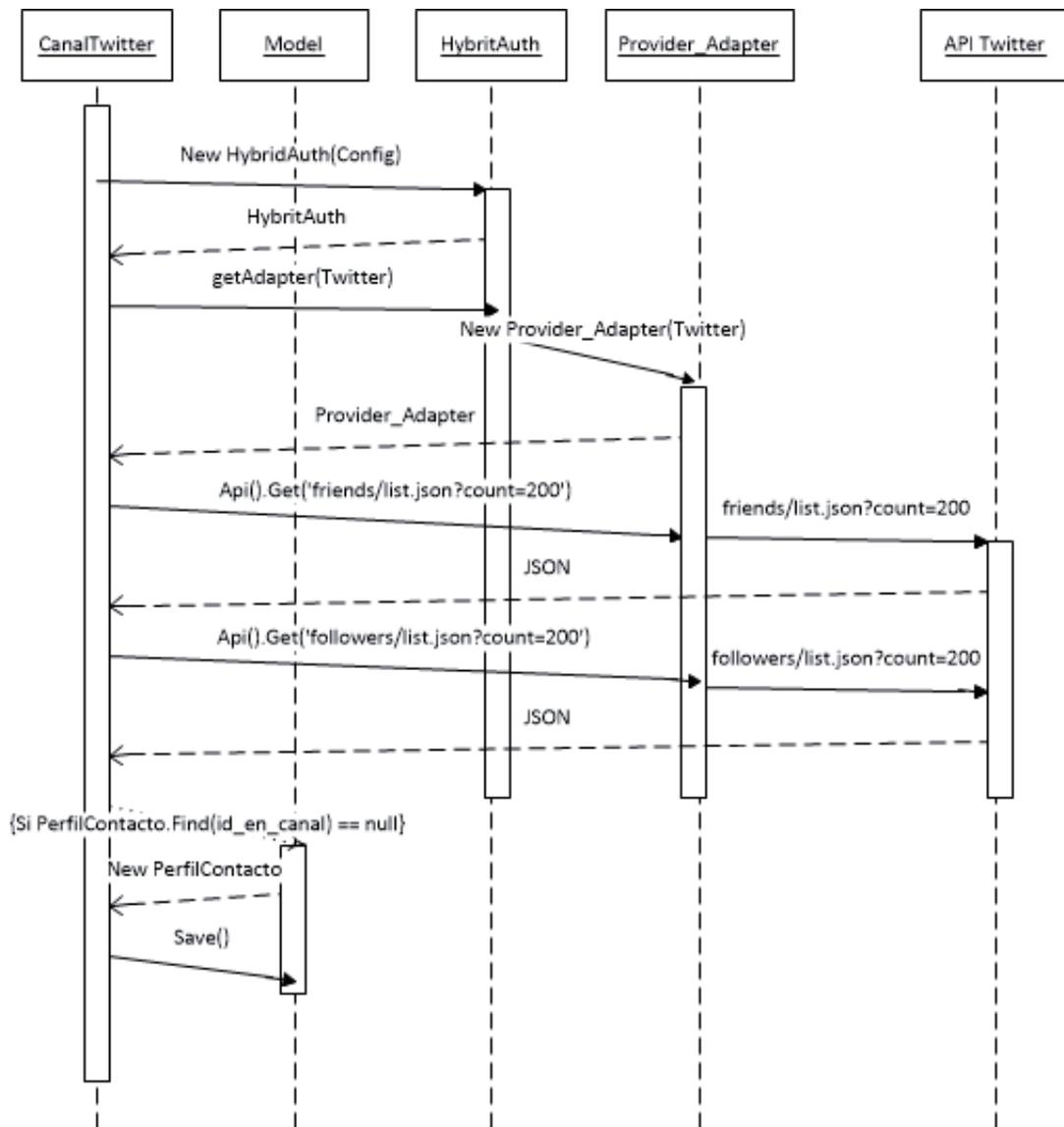
- **Usuario:** Representa al usuario y sus acciones sobre el programa.

- **View:** Es la representación visual del sistema, conformado por tecnologías HTML, CSS, Javascript dando al usuario un conjunto de componentes que permite al usuario interactuar con tablas, formularios botones y enlaces. Cada acción del usuario se traducirá en un enlace URL que puede estar acompañado por un formulario HTTP con parámetros y un método específico como GET,POST,PUT,DELETE.
- **Routes:** Componente que traduce una ruta URL recibida desde View a una acción o función particular en un archivo Controller.
- **Controller:** Clase que centraliza las acciones sobre un componente o modelo del sistema. Tiene la facultad de conectarse con elementos de la base de datos, llamar a clases, módulos y repositorios, ejecutar condiciones de control, redireccionar a otras acciones o controladores y responder al usuario con un nuevo componente View.
- **CanalTwitter:** Es el Driver del canal Twitter que contiene las llamadas a la API del canal y encargarse de la acciones de conectar, sincronizar, y consultar los datos de Twitter.

La secuencia muestra que la interacción del usuario al presionar un botón en una vista del sistema se traduce en un mensaje de URL al sistema MVC, este sistema enlaza este mensaje con una función en un archivo controlador encargado de conectar los componentes de las vistas, modelos y canales para efectuar la sincronización.

El archivo controlador está encargado de recibir los parámetros que identifican el perfil a sincronizar, rescatando desde la base de datos los datos del perfil para luego tomar acciones según su canal. En este caso el canal del perfil es Twitter, para esto el archivo controlador llamara a un nuevo archivo llamado CanalTwitter encargado de administrar todas las acciones sobre este canal, para continuar con la sincronización, el controlador delegara esta tarea a CanalTwitter a través del método SincronizarContactos que recibe como parámetro el perfil de usuario. Finalmente los resultados de la operación son recibidos por el controlador que mostrara al usuario una respuesta creando una nueva vista con los datos.

A continuación podemos ver un ejemplo del trabajo conjunto entre el Driver de canal y HybritAuth al profundizar sobre el método SincronizarContactos del archivo CanalTwitter:



**Ilustración 10: Secuencia SincronizarContactosCanalTwitter**

La secuencia de sincronización de Twitter muestra el objeto CanalTwitter gestiona los componentes de HybritAuth para enviar mensajes a la API externa al sistema, de esta manera HybritAuth delega las acciones a la clase Provider\_Adaptar que esta instanciada para envolver el proveedor de datos de Twitter. Asi es como CanalTwitter envía mensajes a la API de Ttwitter pidiendo el listado de Friends y Followers en formato JSON que permitirá a la aplicación estandarizar los datos y compara la colección de contactos de API Twitter con la colección de contactos del sistema para agregar los nuevos contactos detectados y actualizar los datos de los contactos ya existentes.

## 10. Aplicación Modelo Multicanal

Para utilizar el modelo se logró construir una aplicación web que maneja los datos a través de 3 menús principales: Canales, Contactos y Conversaciones. Se explicará a continuación cada uno de los módulos y componentes, así también sus funciones en la lógica de la comunicación multicanal.

### 10.1 Canales

En el menú de canales tiene como objetivo iniciar las conexiones con los canales externos a la aplicación y sincronizar los datos enviados y recibidos de cada perfil de usuario asociado al canal.

Para cumplir su objetivo, se puede utilizar un panel de conexión para cada uno de los canales habilitados, desde este panel se ingresa y conecta al canal a través de los perfiles de usuario, estos perfiles representan la identidad de la persona o empresa que utiliza esta aplicación para contactar a otras personas o clientes.

The screenshot shows the 'Perfilink' application interface with a dark blue header containing navigation links: 'Perfilink', 'Canales', 'Contactos', 'Conversaciones', and 'Cerrar Sesión'. Below the header are three channel panels, each with a title bar and a table of profiles.

**Twitter Panel:** Header 'Twitter' with 'agregar perfil' button. Table columns: 'img', 'username', 'datos', 'sincronizar', 'operaciones'. Profiles: 'camilo\_cp' (42 followers, 17 messages), 'Icarum\_cl' (42 followers, 0 messages).

**Facebook Panel:** Header 'Facebook' with 'agregar perfil' button. Table columns: 'img', 'username', 'datos', 'sincronizar', 'operaciones'. Profile: 'Camillo Francisco Cuevas Pérez' (0 followers, 0 messages).

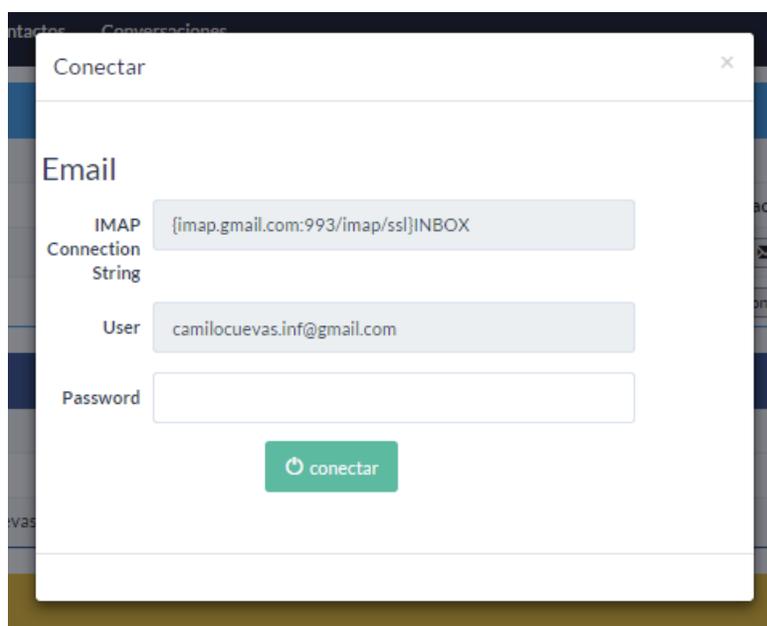
**Email Panel:** Header 'Email' with 'agregar perfil' button. Table columns: 'img', 'username', 'datos', 'sincronizar', 'operaciones'. Profiles: 'camilocuevas.inf@gmail.com' (66 followers, 96 messages), 'camilo.cuevas@ideas-ti.cl' (66 followers, 19 messages), 'camilo.cuevas@informaticapucv.cl' (66 followers, 92 messages).

Ilustración 11: Canales

En la ilustración se puede ver que cada panel posee el nombre del canal que representa y un botón que permite agregar un nuevo perfil de usuario. En la parte interior del panel se muestra una tabla con la información de cada perfil de usuario en el canal, la información desplegada se explica a continuación:

- **img**: imagen del perfil de usuario
- **username**: nombre de usuario
- **datos**: muestra información de la cuenta, a la izquierda el número de contactos que posee, a la derecha el número de mensajes.
- **sincronizar**: menú habilitado cuando el usuario está conectado, dispone 3 botones que permiten sincronizar los datos de guardados en la aplicación con los datos disponibles en el canal. Los botones sincronizar respectivamente: el perfil de usuario, los contactos vinculados, los mensajes enviados y recibidos.
- **operaciones**: permite iniciar la conexión de la cuenta. Una vez conectado dispone los listados de mensajes y contactos vinculados al perfil. También tiene el botón de logout de la sesión.

Al momento de iniciar una conexión, el sistema dispondrá de 2 alternativas de acción: 1) desplegar un panel para el ingreso de los datos de conexión como usuario, password y otros; 2) redireccionar a la página que maneja la autenticación desde el canal externo. Esto dependerá de la naturaleza del canal, aquellos sistemas que implementan el protocolo OAuth 2.0 piden expresamente a las aplicaciones que los utilizan, redirecciones al usuario a su página para autenticar y así proteger los datos privados de aplicaciones mal intencionadas.



The image shows a 'Conectar' (Connect) dialog box. It has a title bar with the text 'Conectar' and a close button (X). The main content area is titled 'Email' and contains three input fields: 'IMAP Connection String' with the value '{imap.gmail.com:993/imap/ssl}INBOX', 'User' with the value 'camilocuevas.inf@gmail.com', and 'Password' which is empty. Below the input fields is a green button with a power icon and the text 'conectar'.

**Ilustración 12: Panel de Conexión**

Una vez iniciada la conexión, el sistema dispondrá de los botones que permiten sincronizar la información que se encuentre en la nube, con la ya almacenada en la aplicación, una vez terminada esta operación de sincronización el sistema mostrara un panel con las noticias de la sincronización.



### Ilustración 13: Noticias de Sincronización

Una vez se encuentra la información sincronizada y disponible en el sistema, se puede navegar a través de ella en los siguientes menús del programa.

## 10.2. Contactos

El menú de contactos maneja la identidad de los contactos o clientes vinculados a los perfiles que el usuario conectados anteriormente desde el menú de canales. El objetivo principal de esta sección es desplegar la información de cada uno de los perfiles de contacto separados por su canal y agruparla de tal manera que la identidad de la persona quede identificada en un único elemento llamado **Contacto Unificado**. Con un contacto unificado se puede proceder posteriormente realizar una comunicación multicanal.

Para cumplir con su tarea esta sección se divide 3 componentes: Contactos Unificados, Perfiles de Contacto y Unir Contactos.

En Contactos Unificados se encuentra el listado de agrupación de perfiles que representa a una persona o cliente en todas sus facetas multicanal. Por su lado, Perfiles de Contacto permite ver cada perfil por separado y sus mensajes de manera independiente. Finalmente el componente Unir Contactos toma los perfiles que se encuentren libres y permite agruparlos para crear más Contactos unificados.

## 10.2.1. Contactos Unificados

En este menú se visualizan todos los contactos unificados del sistema en una grilla que permite la navegación de los registros.

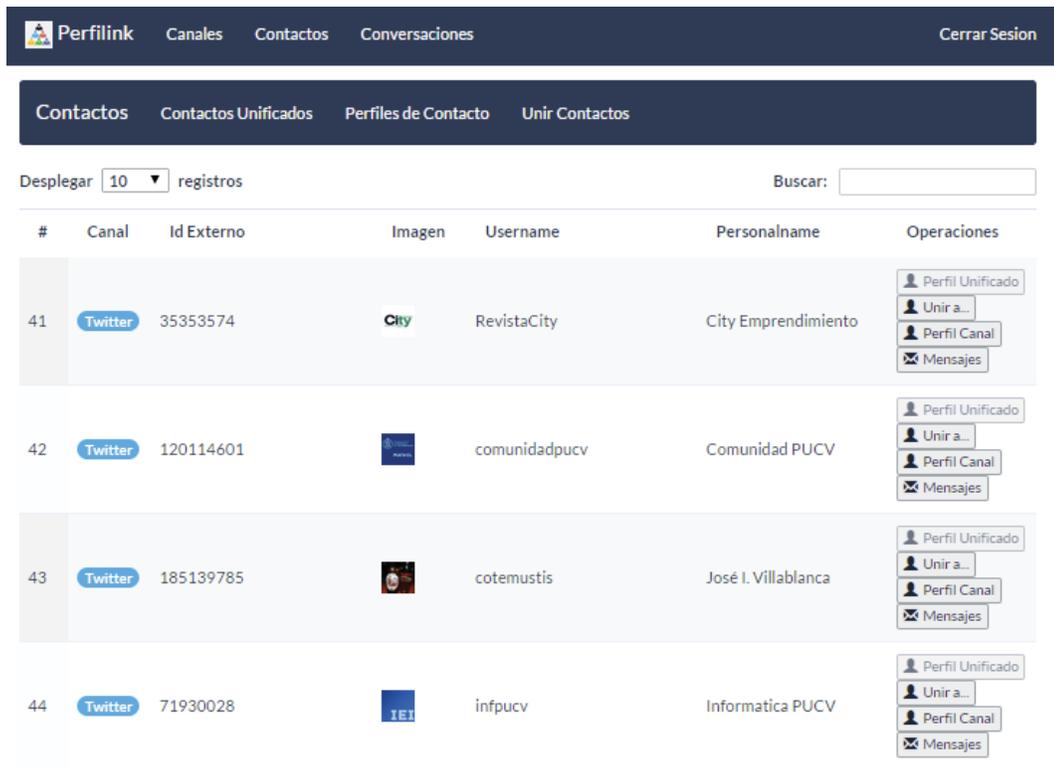
#	Imagen	Personalname	Perfiles	Operaciones
1		Rafael Mellado	<a href="mailto:rafaelmellado@me.com">rafaelmellado@me.com</a> <a href="mailto:felipe.diaz@ideas-ti.cl">felipe.diaz@ideas-ti.cl</a> <a href="mailto:eduardo.avello.b@gmail.com">eduardo.avello.b@gmail.com</a>	<a href="#">Perfil Unificado</a> <a href="#">Agregar perfil</a> <a href="#">Mensajes</a>
2		Alu	<a href="#">alubster</a> <a href="#">Aluu_Aluu</a>	<a href="#">Perfil Unificado</a> <a href="#">Agregar perfil</a> <a href="#">Mensajes</a>
3		Carito Zec	<a href="#">Caritobzs</a>	<a href="#">Perfil Unificado</a> <a href="#">Agregar perfil</a> <a href="#">Mensajes</a>
4		Alexis Abarca	<a href="mailto:alexis.abarca@ideas-ti.cl">alexis.abarca@ideas-ti.cl</a> <a href="mailto:alexis.abarca@informaticapucv.cl">alexis.abarca@informaticapucv.cl</a>	<a href="#">Perfil Unificado</a> <a href="#">Agregar perfil</a> <a href="#">Mensajes</a>

**Ilustración 14: Contactos Unificados**

En la grilla muestran los datos básicos de los Contactos, su imagen y su nombre, los perfiles vinculados entre sí y los botones que permiten entrar en más detalle y añadir más perfiles de contacto al contacto unificado.

## 10.2.2. Perfiles de Contacto

En este menú se disponen cada uno de los perfiles de contactos por separado, se puede navegar y filtrar sus datos de cada contacto.



#	Canal	Id Externo	Imagen	Username	Personalname	Operaciones
41	Twitter	35353574		RevistaCity	City Emprendimiento	<a href="#">Perfil Unificado</a> <a href="#">Unir a...</a> <a href="#">Perfil Canal</a> <a href="#">Mensajes</a>
42	Twitter	120114601		comunidadpucv	Comunidad PUCV	<a href="#">Perfil Unificado</a> <a href="#">Unir a...</a> <a href="#">Perfil Canal</a> <a href="#">Mensajes</a>
43	Twitter	185139785		cotemustis	José I. Villablanca	<a href="#">Perfil Unificado</a> <a href="#">Unir a...</a> <a href="#">Perfil Canal</a> <a href="#">Mensajes</a>
44	Twitter	71930028		infpuv	Informatica PUCV	<a href="#">Perfil Unificado</a> <a href="#">Unir a...</a> <a href="#">Perfil Canal</a> <a href="#">Mensajes</a>

Ilustración 15: Perfiles de Contacto

## 9.2.3. Unir Contactos

Desde esta función se pueden unir los contactos que no estén vinculados anteriormente, para estos se dispone de un input de búsqueda que lista los resultados que coincidan. Una vez seleccionados los perfiles, se unirán y se asignarán los datos del primer perfil, su imagen y nombre, como los datos del contacto.



**Ilustración 16: Unir Contactos**

### 10.3. Conversaciones

En el menú de conversaciones se listan los contactos unificados con los que se tenga algún mensaje en común. Esta sección muestra la siguiente información:

- Contacto: muestra la imagen y nombre del contacto.
- Fecha: es la fecha del último mensaje.
- Ultimo mensaje: contenido del último mensaje enviado o recibido.
- Chat: redirección a chat que visualiza los mensajes enviados y recibidos del contacto y dispone de un input para la comunicación multicanal.



**Ilustración 17: Conversaciones**

## 10.4. Chat

Luego de elegir un contacto desde el menú de conversaciones, la aplicación despliega el chat de conversación multicanal entre el usuario y sus perfiles de usuario, con el contacto unificado y sus perfiles de contacto. En esta sección se destaca el panel de Timeline que muestra todos los mensajes entre el usuario y el contacto a través del tiempo y los canales.



**Ilustración 18: Timeline**

En el timeline se destaca la imagen del emisor del mensaje en la franja central del panel, a un costado de la imagen se puede ver la fecha de emisión del mensaje y al costado opuesto se ve el canal y el nombre del perfil emisor junto con el cuerpo del mensaje. La columna izquierda de mensajes muestra únicamente los mensajes emitidos por el usuario y sus perfiles, en cambio, la columna derecho se encuentran los del contacto.

En el panel de chat se encuentran los input necesarios para enviar un mensaje a los diferentes perfiles de contacto del contacto unificado.

The image shows a web interface for sending a message. It has a dark blue header with the word 'Chat'. Below the header is a white form area. On the left side of the form, there are three dropdown menus. The first is labeled 'Desde:' and has a small profile picture and the text 'camilo\_cp'. The second is labeled 'Tipo Mensaje:' and has the text 'TWEET'. The third is labeled 'Hacia:' and has a small profile picture and the text 'Caritobzs'. To the right of these dropdowns is a large, empty text input field with a light blue background and a vertical cursor. Below the text input field is a small, light blue button with the text 'Enviar'.

### Ilustración 19: Chat

En el Chat se puede seleccionar el perfil desde el cual se enviara el mensaje, el tipo de mensaje según el canal de los perfiles y el perfil del contacto al que llegara el mensaje enviado. Al momento de seleccionar un perfil, se reconocerá automáticamente el canal al que pertenece y se asignara un tipo de mensaje adecuado, a su vez se buscara un perfil de contacto que concuerde con el canal, si no lo encuentra, el sistema avisara que no existe un perfil adecuado y no permitirá enviar el mensaje.

Una vez definidos los perfiles, el tipo y el contenido, se enviara el mensaje. Internamente el sistema revisará el estado de la conexión del perfil, en caso de no estarlo, la pedirá. Una vez conectado enviara el contenido a la API del canal. Cuando el API responda al mensaje, el sistema avisara al usuario si el envió fue exitoso o fallido. En caso de acertar, el sistema actualizara el Timeline y postrara el nuevo mensaje en la pantalla.

## 11. Experiencia de Desarrollo

Luego del Trabajo realizado para construir una aplicación multicanal, se puede describir el proceso como un desafío lleno de cambios y variables que influyen en el resultado final del proyecto. Los principales factores que han influido en el desarrollo son:

**Medidas de control de API:** Para conectar y extraer información de los canales externos, es necesario inscribirse en los sistemas de autenticación de aplicaciones de las API de las redes sociales. A través de esta inscripción, se tiene acceso al portal de desarrolladores de aplicaciones oficiales en donde se registran los datos de la empresa, la información del equipo de desarrollo, la descripción de la aplicación y las configuraciones de aplicación y permisos. Este método de inscripción es usado para la gran mayoría de redes sociales que extienden sus servicios y aplicaciones, pudiendo encontrar esto en Facebook, Twitter, LinkedIn, Google+ y Spotify.

Uno de los factores que pueden limitar el desarrollo de aplicaciones en este ámbito es la restricción de permisos de acceso a las consultas de la API. En el caso de Facebook, por ejemplo, tenemos acceso a consultar todas las funciones de su API desde el portal de desarrolladores, pero, al momento de probar estas funciones en la aplicación desarrollada, solo se tiene acceso a información básica del usuario que está usando la aplicación, al listado de sus amigos y al correo electrónico del usuario.

Para poder habilitar acceso a más funciones de las API es necesario cumplir con las políticas de privacidad y términos de servicio, designados por el portal de desarrollo. Dentro de estos términos se puede encontrar la necesidad de implementar un protocolo seguro de encriptación a través de certificado SSL y protocolo HTTPS.

Para llevar el control del cumplimiento de las políticas, la aplicación debe pasar un periodo de prueba por parte del portal de desarrollo, en donde se analiza el contenido de la aplicación, las funciones, el uso de la información solicitada a la API, la seguridad y el posible uso mal intencionado de información. Una vez pase este periodo de prueba, la aplicación puede usar los permisos solicitados en un ambiente de producción.

### **Documentación superficial e incompleta:**

Uno de los inconvenientes más limitantes al momento de trabajar con API es la documentación básica que se encuentra en los portales oficiales de desarrollo. En estos portales se encuentra la documentación suficiente como para inscribir la aplicación, implementar un login de usuario a través de la autenticación de la API, tutoriales básicos de consulta y descripción general de los servicios.

Esta documentación se presenta segmentada de acuerdo a la plataforma que consumirá los servicios, en donde se pueden encontrar Plataformas Web, App Móvil para Android e IOS, Aplicaciones de Escritorio y Scripts Javascript. Sin mencionar que cada una de estas plataformas puede utilizar diversas tecnologías como: php, javascript, java, c#, phyton, perl y otros. La documentación es amplia en cuanto a tecnologías abarcables, pero en el interior de esta documentación, las descripciones son básicas y repetitivas.

La principal falencia de esta documentación es la falta de información, en donde esta ausenta la descripción de los procesos de autenticación, el proceso de envío y recibo de mensajes, la descripción en detalle de cada servicio y la descripción y solución de los errores más comunes. En vista a la ausencia de esta información, la forma de llevar adelante el desarrollo se hace a partir de práctica constante, ensayo y error, y recurrir a la información de foros de desarrollo.

**Cambios en las API:** desde el inicio del proyecto hasta su concepción final, las API de cada red social han presentado cambios en sus funciones tanto para arreglar errores como también dar de baja los servicios. Esto limita el control del programa sobre la disponibilidad de funciones que puede prometer a dar fiel cumplimiento. El motivo de estos cambios es variado: políticas internas de la empresa, protección de datos privados, mejoras o redistribución de sus funciones, respuesta a ataques maliciosos, etc.

Para aplicar estos cambios, las redes sociales dan aviso anticipado de las nuevas medidas y una posterior marcha blanca de los servicios renovados, en donde las viejas versiones de las API siguen en funcionamiento hasta una fecha límite. Para poder responder de forma efectiva, es necesario contar con personal especializado en el área de desarrollo de social media, que esté atento a los avisos y novedades de este tipo de aplicaciones, además de contar con el tiempo y los recursos necesarios para realizar los cambios de adaptación.

**Configuración y Sistemas de Seguridad Web:** Para utilizar los métodos de autenticación de las API, se hacen uso de tecnologías de servidor para mantener los datos del usuario, tales como las Sesiones y las Cookies. Adicionalmente, algunas API pueden solicitar más acciones como permisos de lectura, escritura y ejecución de archivos, o bien el uso de certificados SSL para protocolo HTTPS. Estos parámetros de configuración del servidor deben ser revisado y habilitados la comunicación de las API. También es necesario habilitar los puertos desde el Firewall del sistema y validar que las versiones de las plataformas utilizadas soporten la versión instalada de la API.

Todos estos factores mencionados son vitales de controlar para que un proyecto con API de social media pueda ser acabado y puesto en marcha. Por estas mismas condiciones se hace necesario contar con personal que tenga la experiencia de desarrollo en estas API y la cultura de desarrollado que este informado de los cambios más recientes en estas tecnologías.

## 12. Conclusiones

El servicio al cliente es un elemento clave para aumentar la satisfacción y fidelidad de los clientes de las empresas, entregando un valor adicional a las ventas y mejorando el nivel de confianza. Para llevar a cabo un sólido servicio al cliente, se utiliza un marco de trabajo de proceso CRM, que define una estrategia, otorga valor al cliente y la empresa, requiere una integración de canales y debe medir el desempeño final del servicio.

Las redes sociales y los medios de comunicación digitales constituyen el principal medio de comunicación actual, dando participación activa al cliente a través de su opinión e intereses. Afortunadamente, Las tecnologías actuales permiten la interconectividad entre aplicaciones web mediante API y Servicios REST.

A través de un modelo general, se logró estandarizar las funcionalidades de las API de comunicación de diferentes canales, tales como Twitter, Facebook y Correo Electrónico. Utilizando una plataforma web, es posible implementar un sistema de recepción y envío de mensaje multicanal entre la empresa y sus clientes.

Como resultado de la integración, se cuenta con un sistema que captura de manera centralizada los mensajes de los clientes y permite comunicarse con el cliente por múltiples canales, apoyado de un perfil e historial de mensajes unificado.

La integración de los canales requiere mantención y control constante, que pueda responder a los cambios en las tecnologías aplicadas y a las políticas de privacidad de información de los usuarios y las empresas. Este trabajo requiere personal especializado en el área de desarrollo de aplicaciones de Social Media.

## 13. Referencias Bibliográficas

[Brown] Andrew Brown, Gestión de la Atención al Cliente, 1992.

[Solis et al, 2008] Brian Solis, Custom[Bell] Donald Bell, UML basics: An introduction to the Unified Modeling Language © 2003 Rational Software Service, The Art of Listening and Engagement Through Social Media, Marzo 2008

[Russell et al, 2011] Matthew A. Russell, Mining the Social Web, Febrero 2011

[Payne] Adrian Payne y Pennie Frow, A Strategic Framework for Customer Relationship Management

[Gartner Group (2001)] "CRM Economics: Figuring Out the ROI on Customer Initiatives," whitepaper, Stamford, CT.

[Halligan y Shah, 2010] Halligan, B., & Shah, D. (2010). Inbound marketing get found using google, social media, and blogs. New Jersey: John Wiley & Sons Ltd.

[Greenberg (2010)] Greenberg, P. (2010). CRM at the speed of light. Social CRM strategies, tools, and techniques for engaging your customer. New York: McGraw-Hill.

[Paul, A., & Nilsson, J. (2011)]. *Social CRM*. Chichester: John Wiley & Sons Ltd.

[Goldenberg] Goldenberg, Barton. "¿Qué es CRM y cuál es el verdadero significado?". Noviembre 2002.

[Deloitte (2011)] Deloitte. (2011). CRM 2.0 or social CRM for financial industry. Septiembre 2011

[Schaff y Harris, (2012)] Schaff, C., & Harris, G. (2012). 7 secrets to social media business success

[Bell] Donald Bell, UML basics: An introduction to the Unified Modeling Language © 2003 Rational Software

## 14. Referencias Web

<https://developers.facebook.com/docs/graph-api/reference>

<https://dev.twitter.com/docs/api/1.1>

<https://github.com/shirioko/WhatsAPI/wiki/WhatsApp-Registration-Flow>

<https://developers.google.com/+/quickstart/php>

<http://oauth.net/>

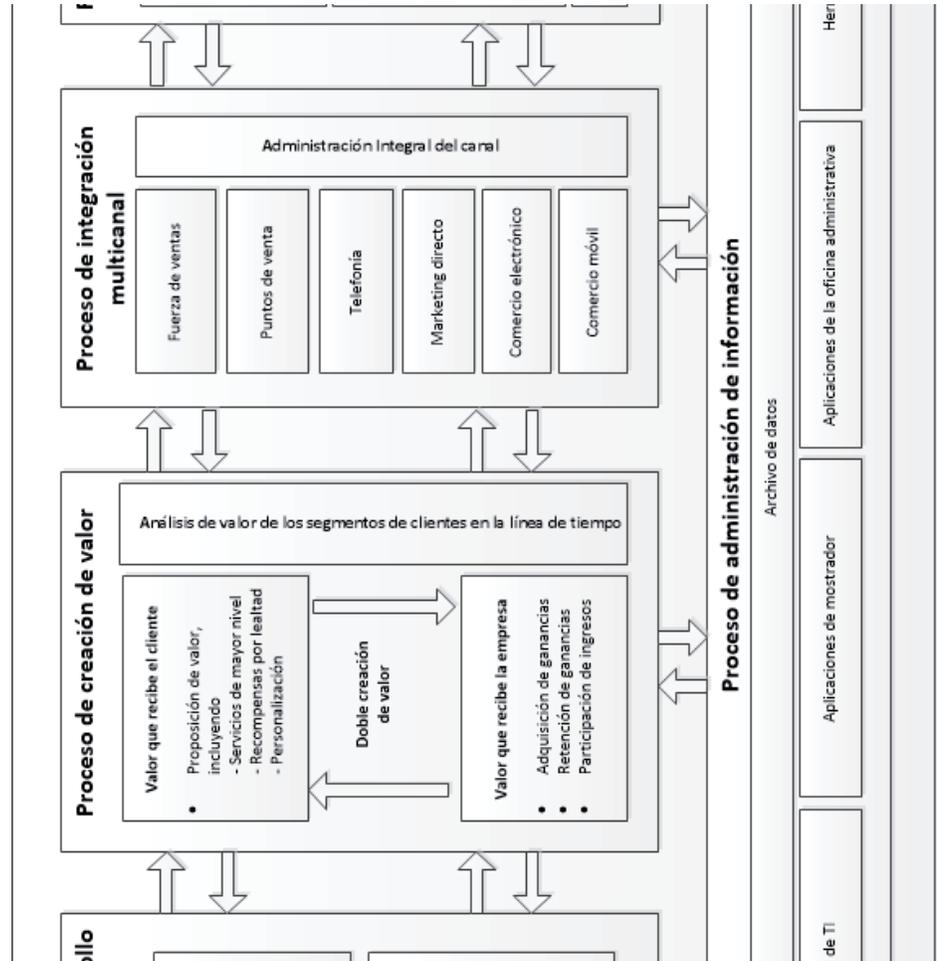
<http://php.net/manual/es/book.imap.php>

<https://perfilink.cl>

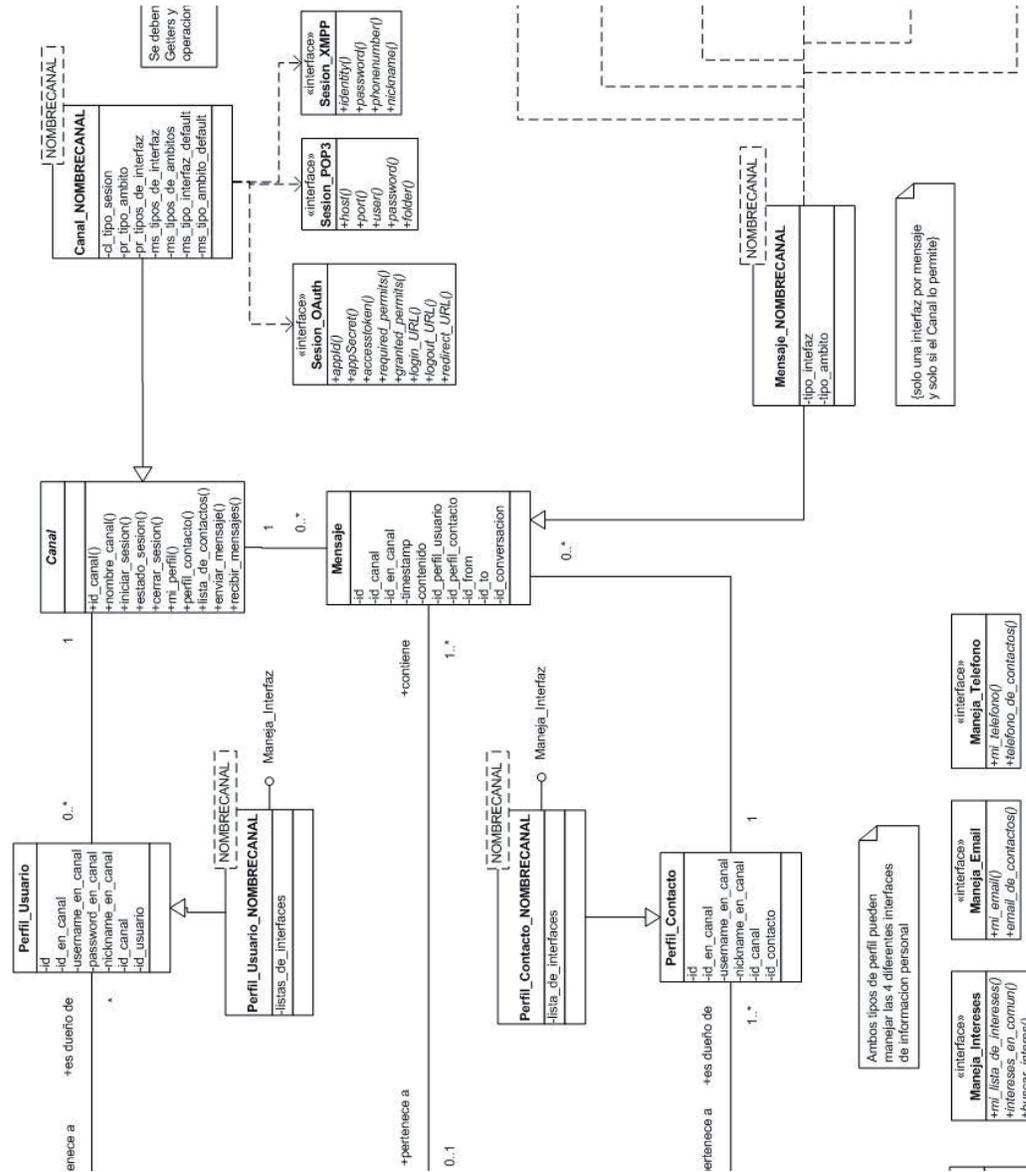
<http://laravel.com/>

<http://hybridauth.sourceforge.net/>

# Anexo 1: Procesos CRM



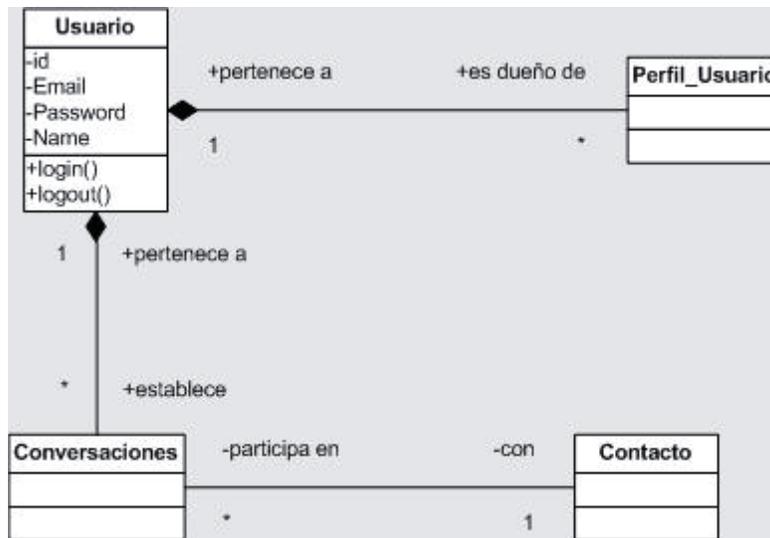
## Anexo 2: Modelo de Clases



## Anexo 3: Modelo de Clases Específico

### Clase Usuario

Esta clase representa al usuario o actor dentro del sistema, requiere datos de identificación y muestra las relaciones de composición que posee con las clases Perfil\_Usuario y Conversacion, las cuales comparten el mismo tiempo de vida.



Clase Usuario					
atributos	ámbito	tipo	setter	getter	Propósito
id	privado	int	no	si	identificador numerico en el sistema
Email	privado	string	si	si	dato de contacto e identificador en el sistema
Password	privado	string	si	si	contraseña para acceder al sistema
Name	privado	string	si	si	nombre del usuario
metodos	ambito	tipo retorno	parametros		Propósito
login	publico	boolean	email,password		iniciar sesion en el sistema perfilink, no inicia sesiones en canales
Logout	publico	boolean	null		cerrarsesion en el sistema perfilink y en cuentas asociadas.

### Clase Conversación

Esta clase representa un conjunto de mensajes unidos por un tema en común que pueden pertenecer a diversos canales de comunicación y están delimitados dentro de un periodo de tiempo.

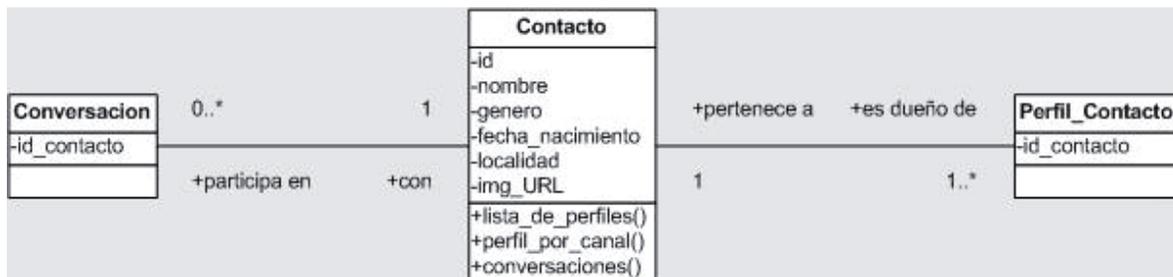


Clase Conversacion					
atributos	ambito	tipo	setter	getter	proposito
Id	privado	int	no	si	identificador numérico en el sistema
id_usuario	privado	int	no	si	identifica al usuario dueño de la conversación
id_contacto	privado	int	no	si	identifica al contacto participante
Asunto	publico	string	si	si	tema central de la conversación
timestamp_start	privado	datetime	no	si	instante de partida
timestamp_end	privado	datetime	no	si	instante del cierre de la conversación
estado	privado	string	no	si	estados: abierto (se puede modificar y agregar nuevos mensajes) y cerrado (tema cerrado y no requiere mas modificaciones)
metodos	ambito	tipo retorno	parámetros		Propósito
ultimo_mensaje	publico	Clase:Mensaje	null		devuelve el ultimo mensaje recibido o enviado
listar_mensajes	publico	lista<Mensaje>	fecha_inicio, fecha_termino		listado de los mensajes asociados ordenados en el tiempo proveniente de diversos canales
estado	publico	string	null		devuelve el estado (abierto,cerrado)
cerrar_conversacion	publico	boolean	null		deshabilita la modificación

abrir_conversacion	publico	boolean	null	habilita la modificacion
responder	publico	boolean	Clase:Mensaje	envia un mensaje por medio de un canal al perfil del usuario del mismo canal
incluir_mensaje	publico	boolean	Clase:Mensaje	se agrega un mensaje desde un canal a esta conversacion.

## Clase Contacto

Esta clase contiene los datos que representan a una persona con la que se tiene contacto, esta clase obtiene información de los diversos perfiles asignados a esta identidad con ayuda del usuario.



Clase Contacto					
atributos	ambito	tipo	setter	getter	proposito
Id	privado	int	no	si	identificador unico
Nombre	publico	string	si	si	nombre del contacto
Genero	publico	string	si	si	género: masculino, femenino
fecha_nacimiento	publico	datetime	si	si	fecha de nacimiento
localidad	publico	string	si	si	ubicación geografica
img_url	privado	string	si	si	direccion de la imagen del contacto
metodos	ambito	tipo retorno	parametros		proposito
lista_de_perfiles	publico	boolean	null		devuelve cada uno de los perfiles del contacto asociados a los canales

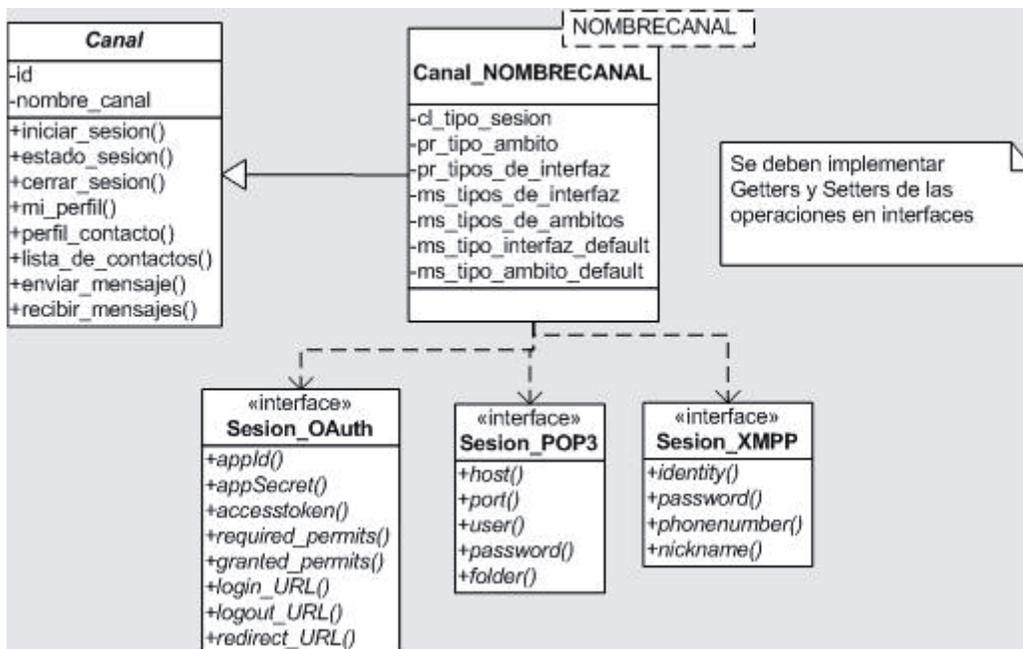
perfil_por_canal	publico	boolean	nombre_canal	devuelve el perfil del contacto asociado al canal
conversaciones	publico	boolean	null	listas de conversaciones con el contacto

## Clase Canal

Acá se presenta una clase abstracta llamada canal, esta clase deberá heredarse en cada uno de los canales añadidos al sistema y sus operaciones serán definidas para utilizar las API de estos canales.

Junto a esto se presenta la situación de que cada canal tiene una tecnología de sesión que lo caracteriza, dentro de las cuales contamos con OAuth para redes sociales y servicios REST, POP3 para los emails y XMPP para sistemas de mensajería como whatsapp y la antigua plataforma de mensajes de google, googletalk. Cada tipo de canal deberá implementar su interfaz de sesión y asociar sus métodos a las funciones de iniciar y cerrar sesión.

Por ultimo se menciona que las clases de canal contendrán información que describen los perfiles y mensajes, estos atributos se identifican a través del prefijo cl\_, pr\_,ms\_ para canal, perfil y mensaje respectivamente.



## Clase Canal

<b>atributos</b>	<b>ambito</b>	<b>tipo</b>	<b>setter</b>	<b>getter</b>	<b>Propósito</b>
id	privado	int	no	si	identificador único
nombre_canal	privado	string	no	si	nombre identificador
<b>metodos</b>	<b>ambito</b>	<b>tipo retorno</b>	<b>parametros</b>		<b>Propósito</b>
iniciar_sesion	publico	boolean	Perfil_Usuario		conectar con el canal
estado_sesion	publico	string	Perfil_Usuario		estado de conexión
cerrar_sesion	publico	boolean	Perfil_Usuario		Desconectar
mi_perfil	publico	Perfil_Usuario	Usuario		perfil de usuario asociado al canal
perfil_contacto	publico	Perfil_Contacto	Contacto		devuelve el perfil del contacto asociado a este canal
lista_de_contatos	publico	lista<Contacto>	Perfil_Usuario		devuelve la lista de contactos asociados al perfil
enviar_mensaje	publico	boolean	Mensaje		envia el mensaje a través del canal al contacto identificado en el mensaje
recibir_mensajes	publico	lista<Mensajes>	fecha_inicio, fecha_termino		devuelve una lista de mensajes asociados al perfil de este canal

<b>Clase Canal_NOMBRECANAL</b>					
<b>parametro de clase</b>			<b>proposito</b>		
NOMBRECANAL			Es el nombre del canal externo a la cual esta clase dara soporte, requiere para ello implementar de forma especifica los metodos heredados de la clase Canal y ademas incluir las interfaces necesarias para iniciar sesion.		
<b>atributos</b>	<b>ambito</b>	<b>tipo</b>	<b>setter</b>	<b>getter</b>	<b>proposito</b>
cl_tipo_sesion	privado	string	no	si	tipo de interfaz requerida para iniciar sesion.
pr_tipo_ambito	privado	string	no	si	define que ambito social tiene el perfil: publico, privado, solo contactos.

pr_tipos_de_interfaz	privado	lista<string>	no	si	tipos de interfaz de informacion personal que manejan los perfiles de este canal.
ms_tipos_de_interfaz	privado	lista<string>	no	si	tipos de interfaz de comunicaci3n de mensajes que permite este canal.
ms_tipos_de_ambitos	privado	lista<string>	no	si	tipos de ambito social que permiten los mensajes: publico, privado, solo contactos.
ms_tipo_interfaz_default	privado	string	si	si	tipo de interfaz default que usara el sistema, se asume que es texto.
ms_tipo_ambito_default	privado	string	si	si	tipo de ambito default que usara el sistema en los mensajes

<b>Interface Sesion_Oauth</b>			
<b>metodos</b>	<b>tipo retorno</b>	<b>parametros</b>	<b>Prop3sito</b>
appId	string	(string) =>funcion set, (null) =>funcionget	id de la aplicaci3n registrada en el canal externo.
appSecret	string	(string) =>funcion set, (null) =>funcionget	clave privada de la aplicaci3n.
accesstoken	string	(string) =>funcion set, (null) =>funcionget	token de identificaci3n de sesi3n valido por un limite de tiempo.
required_permits	lista<string>	(string) =>funcion set, (null) =>funcionget	lista de permisos requeridos en el login para usar el canal.
granted_permits	lista<string>	null	lista de permisos finalmente otorgados por el usuario.
login_URL	string	null	url que direcciona al login del canal.
logout_URL	string	null	url que redirecciona al cierre de sesi3n del canal.

redirect_URL	string	(string) =>funcion set, (null) =>funcionget	url en la aplicación a la cual retornaran las funciones login y logout.
--------------	--------	--	---

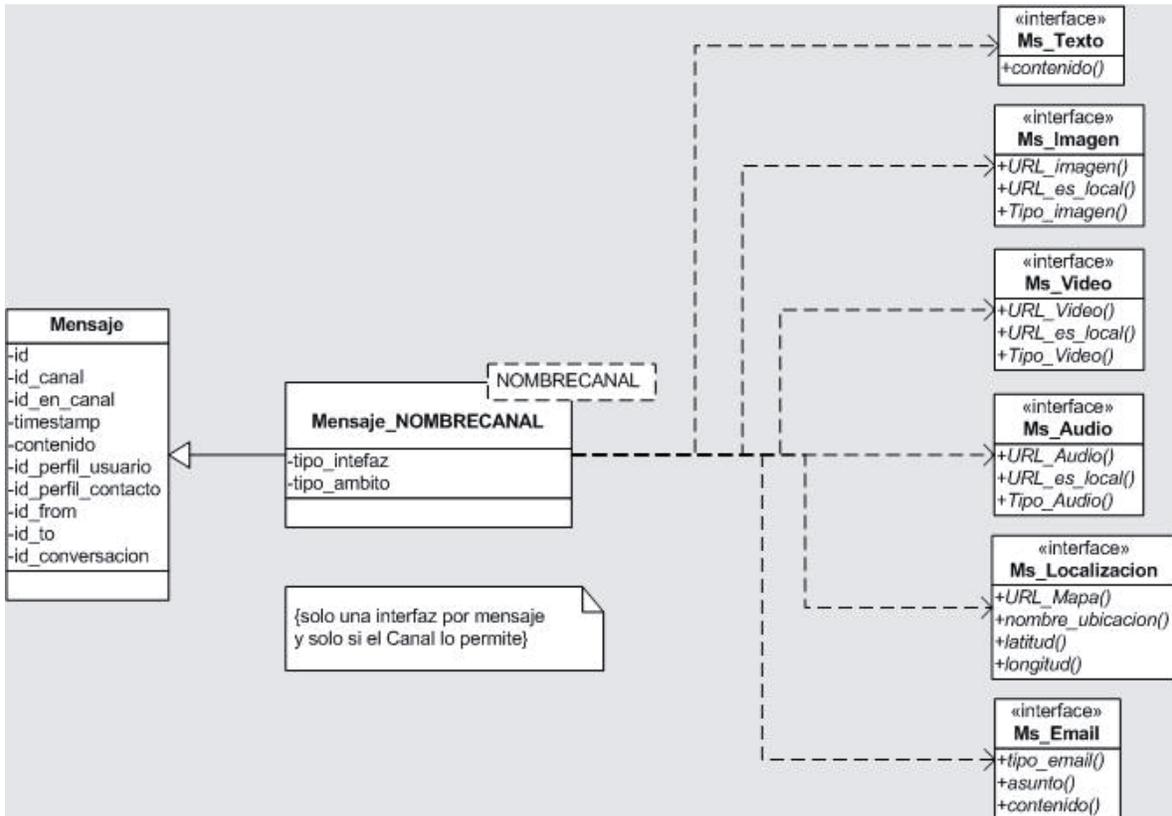
<b>Interface Sesion_POP3</b>			
<b>metodo</b>	<b>tipo retorno</b>	<b>parametros</b>	<b>Propósito</b>
host	string	string = funcion set, null = funcionget	direccion URL del servidor de correo
port	int	int = funcion set, null = funcionget	puerto asignado al servidor de correo
user	string	string = funcion set, null = funcionget	Email
password	string	string = funcion set, null = funcionget	Contraseña
folder	string	string = funcion set, null = funcionget	carpeta de correo, ejemplo: INBOX

<b>Interface Sesion_XMPP</b>			
<b>metodo</b>	<b>tipo retorno</b>	<b>parametros</b>	<b>proposito</b>
identity	string	string = funcion set, null = funcionget	identificar la sesion XMPP
password	string	string = funcion set, null = funcionget	contraseña de la sesion
phonenummer	string	string = funcion set, null = funcionget	numero de telefono
nickname	string	string = funcion set, null = funcionget	nombre visible a los contactos

## Clase Mensaje

La Clase canal contiene la información transversal para lleva a cabo cualquier comunicación a través de los canales, por defectos se asume que el formato del mensaje será texto, para realizar un formato distinto soportado por el canal se debe instanciar una clase Mensaje\_NOMBRECANAL, que posee las mismas particularidades del la clase

Canal\_NOMBRECANAL, y así implementar las interfaces que definen el formato del mensaje.



Clase Mensaje					
atributos	ambito	tipo	setter	getter	Propósito
id	privado	int	no	si	identificador en el sistema.
id_canal	privado	int	si	si	identificador del canal al que pertenece.
id_en_canal	privado	int	no	si	identificador externo que proviene del canal.
timestamp	privado	datetime	no	si	instante de creacion del mensaje en el canal.
contenido	privado	string	si	si	texto del mensaje, para otro tipo de mensaje se requieren interfaces.
id_perfil_usuario	privado	int	si	si	id del perfil del usuario en el canal.

id_perfil_contacto	privado	int	si	si	id del perfil del contacto en el canal.
id_from	privado	int	si	si	id del perfil que envia (usuario o contacto)
id_to	privado	int	si	si	id del perfil que recibe
id_conversacion	privado	int	si	si	id de la conversacion a la que pertenece.

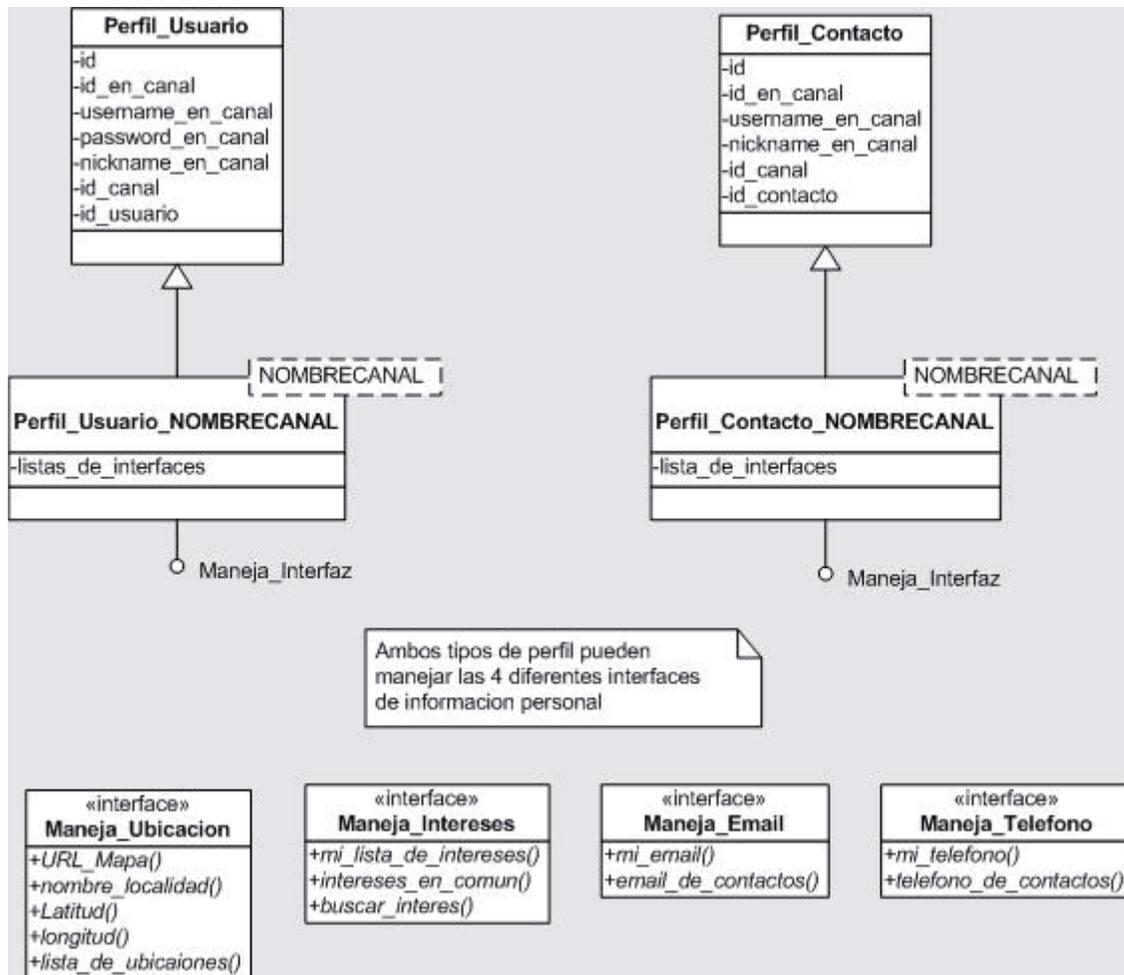
Clase Mensaje_NOMBRECANAL					
parametro de clase		Propósito			
NOMBRECANAL		Es el nombre del canal externo del cual este mensaje proviene, por tanto, esta clase requiere implementar las interfaces especificas permitidas por este canal			
atributo	ambito	tipo	setter	getter	Propósito
tipo_interfaz	privado	string	si	si	Define la interfaz del mensaje, o bien, el tipo de contenido que el mensaje posee.
tipo_ambito	privado	string	si	si	Define el ámbito social del contenido del mensaje: publico, privado, solo contactos.

Interfaces de Mensaje			
metodo	tipo retorno	parametros	Propósito
<b>Ms_Texto</b>			
contenido	string	string = funcion set / null = funcionget	representa el mensaje de texto
<b>Ms_Imagen</b>			
URL_imagen	string	string = funcion set / null = funcionget	representa la URL de la ubicación del archivo de imagen
URL_es_local	boolean	null	verifica si el archivo esta guardada dentro del sistema o pertenece a un enlace externo
tipo_imagen	string	null	devuelve el tipo de imagen del archivo, ej: JPEG, GIF
<b>Ms_Video</b>			
URL_Video	string	string = funcion set / null = funcionget	representa la URL de la ubicación del archivo de video

URL_es_local	boolean	null	verifica si el archivo esta guardada dentro del sistema o pertenece a un enlace externo
Tipo_Video	string	null	devuelve el tipo de video, ej: flv
<b>Ms_Localizacion</b>			
URL_Mapa	string	string = funcion set / null = funcionget	devuelve una url donde visualizar la ubicación en un mapa.
nombre_ubicacion	string	string = funcion set / null = funcionget	nombre del lugar
latitud	int	int = funcion set / null = funcionget	numero de latitud
longitud	int	int = funcion set / null = funcionget	numero de longitud
<b>Ms_Email</b>			
tipo_email	string	string = funcion set / null = funcionget	tipo de email: new, repley, forward
asunto	string	string = funcion set / null = funcionget	asunto en comun con varios email
contenido	string	string = funcion set / null = funcionget	cuerpo del email
<b>Ms_Audio</b>			
URL_Audio	string	string = funcion set / null = funcionget	representa la URL de la ubicación del archivo de audio.
URL_es_local	boolean	null	verifica si el archivo esta guardada dentro del sistema o pertenece a un enlace externo.
Tipo_Audio	string	null	devuelve el tipo de audio, ej: mp3, 3gp, midi

## Clases de Perfiles

En el caso de los perfiles estos pueden manejar distintos contenidos a travez de las interfaces según lo permita el canal, ambos tipos de permisos pueden implementar las interfaces, con esto se debe tener en cuenta que las restricciones que pongan los contactos al compartir sus datos delimitaran el uso de estas interfaces en ellos.



Clase Perfil_Usuario					
atributos	ambito	tipo	setter	getter	proposito
Id	privado	int	no	si	identificador en el sistema.
id_en_canal	privado	int	no	si	identificador externo del perfil en el canal
username_en_canal	privado	string	si	si	nombre de usuario en el canal
password_en_canal	privado	string	si	si	password de acceso
nickname_en_canal	privado	string	no	si	alias en el canal
id_canal	privado	int	no	si	identificador del canal
id_usuario	privado	int	no	si	identificador del usuario

Clase Perfil_Contacto					
atributos	ambito	tipo	setter	getter	proposito
Id	privado	int	no	si	identificador en el sistema.
id_en_canal	privado	int	no	si	identificador externo del perfil en el canal
username_en_canal	privado	string	si	si	nombre de usuario en el canal
nickname_en_canal	privado	string	no	si	alias en el canal
id_canal	privado	int	no	si	identificador del canal
id_contacto	privado	int	no	si	identificador del contacto

Interfaces de Perfil			
metodo	tipo retorno	parametros	proposito
<b>Maneja_Intereses</b>			
mi_lista_de_intereses	lista<string>	null	muestra una lista de los intereses entregados por en canal
intereses_en_comun	lista<string>	Perfil_Usuario, Perfil_Contacto	representa la URL de la ubicación del archivo de imagen
buscar_intereses	lista<string>	Perfil_Contacto	Devuelve una lista de intereses del contacto.
<b>Maneja_Email</b>			
mi_email	string	string = funcion set / null = funcionget	obtiene el email asociado al perfil
email_de_contactos	lista<(string,string)>	null	devuelve una lista con el nombre del contacto y su email
Tipo_Video	string	null	devuelve el tipo de video, ej: flv
<b>Maneja_Ubicacion</b>			
URL_Mapa	string	string = funcion set / null = funcionget	devuelve una url donde visualizar la ubicación en un mapa.
nombre_ubicacion	string	string = funcion set / null = funcionget	nombre del lugar
latitud	int	int = funcion set / null = funcionget	numero de latitud
longitud	int	int = funcion set / null = funcionget	numero de longitud

lista de ubicaciones	lista<string>	fecha_inicio, fecha_termino	devuelve el listado de ubicaciones en un periodo de tiempo
<b>Maneja_Telefono</b>			
mi_telefono	string	string = funcion set / null = funcionget	numero de telefono con codigo de localidad incluido
telefono_de_contactos	lista<(string,string)>	null	devuelve una lista con el nombre del contacto y su telefono.

## **Anexo 4: Secuencia Sincronizar Contactos**

