



## APLICACIONES CON SOPORTE PARA COMUNICACIONES: UN ENFOQUE PRÁCTICO

---

Daniel Cárdenas Salas

### Resumen

*Aplicaciones con soporte para comunicaciones (CEA) es un concepto relativamente reciente que tiene como objetivo la implementación de diversos mecanismos de comunicación tanto entre aplicaciones como entre las aplicaciones y sus usuarios, con el fin de facilitar el flujo de los procesos de negocio. El presente artículo analiza la aplicación práctica de una plataforma genérica de comunicación llamada Coral CEA en un aplicativo responsable de la reserva de actividades recreativas en un resort canadiense. Para evitar que la interacción entre el aplicativo y los servicios proporcionados por la plataforma afecten negativamente al primero, se diseñó y desarrolló un servicio intermedio. Sobre esta base se analizaron los componentes de negocio involucrados en la solución, el marco teórico sobre el cual se basó la investigación, su influencia en la gestión del negocio y los detalles técnicos de su implementación.*

*Palabras clave: Aplicaciones con soporte para comunicaciones (CEA), reserva de actividades, arquitectura orientada a servicios, software como servicio / SaaS*

## Introducción

La convergencia entre sistemas de redes, telefonía y software es un proceso costoso que ha sido aplicado aisladamente, en general por grandes empresas de telecomunicaciones con el capital suficiente para implementar sus diversos componentes. Sin embargo, pequeñas y medianas empresas también pueden obtener los mismos beneficios a una fracción del costo mediante la utilización de plataformas genéricas de comunicación. Una de ellas es Coral CEA, entidad que ofrece servicios de Communications Enabled Applications (CEA) expuestos como componentes reutilizables. En este artículo se muestra cómo una pequeña empresa puede implementar de manera exitosa estos servicios de comunicación dentro de sus propias operaciones de negocio.

Por lo expuesto, el principal objetivo del trabajo de investigación consiste en implementar mecanismos automáticos de comunicación en una aplicación de reserva de actividades recreativas (Activity Box) utilizando los servicios proporcionados por una plataforma de comunicación (Coral CEA).

## Modelo de negocios

Rezact es una pequeña empresa que inició sus operaciones a principios del 2006 y está ubicada en la región de Mont-Tremblant, en la provincia de Quebec en Canadá. Esta región, convertida en Parque Nacional, es conocida por ser sede de uno de los principales *resorts* de la costa este de Norteamérica, y es especialmente favorable para la práctica del esquí. Desde el punto de vista del *marketing*, el *resort* orienta sus actividades a familias más que a deportistas expertos, y, por lo tanto, opera durante el transcurso del año, ofreciendo diversas actividades en una ubicación central llamada *The Activity Center*. Rezact fue creada para desarrollar e implementar un sistema que permita administrar la reserva de distintas actividades recreacionales en el *resort* por medio del Activity Center.

El modelo de negocios del Activity Center consta de tres entidades. La primera de ellas está formada por los turistas o clientes que acuden al *resort*. Este grupo está compuesto principalmente por familias con un alto poder adquisitivo, que permanecen en la región generalmente por una semana, convirtiéndose así en el mayor generador de ingresos del área. La segunda entidad la constituye pequeñas empresas locales, en su mayoría negocios familiares independientes, que se encargan de proporcionar actividades recreacionales complementarias en el *resort* a los turistas o clientes. Estas pueden ser tan diversas como servicios de *spa* y masajes, excursiones guiadas en canoas, en trineos o a caballo, clases de cocina, paseos en motonieve e incluso clases privadas de pilotaje. Estas empresas, denominadas los operadores, se encuentran distribuidas en un área geográfica relativamente

amplia, y tradicionalmente se han enfrentado a la dificultad de atraer clientes, los cuales están en su mayoría concentrados en el *resort*, con sus respectivos negocios. La tercera entidad es el *resort*, operado por una empresa transnacional llamada *Intrawest*, que tiene interés en ofrecer la mejor experiencia a sus clientes, y que esta vaya más allá de la práctica del esquí, proporcionándoles un punto centralizado donde aquellos puedan reservar diversas actividades.

El Activity Center se convierte así en un punto intermedio entre los operadores que ofrecen sus servicios de actividades y los clientes del *resort*. Bajo este modelo todos los involucrados se benefician: los clientes tienen un centro ubicado en el mismo *resort* dónde seleccionar sus actividades entre una vasta gama de opciones, los operadores tienen acceso a clientes que de otra forma estarían fuera de su alcance; y el *resort* puede proporcionar una experiencia mucho más completa durante todo el año a sus clientes. Este modelo de negocios se encuentra ilustrado en la Figura 1:

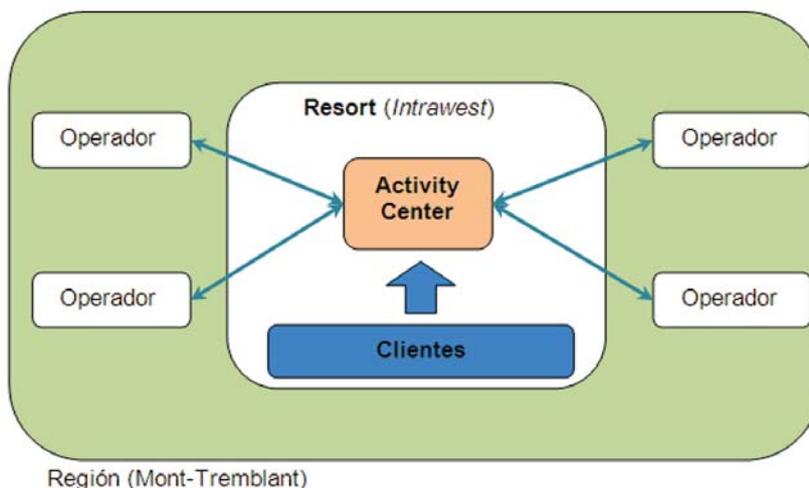


Figura 1. Modelo de negocios en el resort

Elaboración propia.

### Activity Box

El aplicativo Activity Box es un sistema de reservas en línea desarrollado por Rezact Inc. que le permite, tanto al Activity Center como a otros agentes de venta, la administración de diversos tipos de actividades recreativas, categorizadas de acuerdo con el tipo de recursos que dichas actividades necesitan:

- a) Por capacidad del local, en donde la disponibilidad se mide por el número de participantes que pueden reunirse en un determinado espacio. Este es el caso de actividades como conciertos o carreras de autos.
- b) Por el equipo utilizado, en donde la disponibilidad es calculada por el número de espacios vacantes en cada equipo y por la cantidad de equipos disponibles. Como ejemplos de estas actividades tenemos canotaje (dos personas por canoa) o paseos en helicóptero (tres personas por helicóptero).
- c) Por el número de guías requerido, cuando la actividad dicta que un guía puede atender un máximo número de participantes, como es el caso de excursiones guiadas al parque nacional ubicado en la región, o escalamiento de montaña.
- d) Por la combinación de equipos y guías, cuando se necesita controlar tanto la disponibilidad de espacios como de guías. Tal es el caso de actividades como masajes, en donde se requiere una sala de masaje y un masajista.

El sistema diseñado, Activity Box, permite controlar estos diferentes tipos de inventario en tiempo real. Fue diseñado bajo la premisa de software-como-servicio (SaaS, por sus siglas en inglés), es decir, Rezact Inc. ofrece la oportunidad de utilizar el sistema como si se tratase del alquiler de un servicio. No se requiere ninguna instalación y cada usuario del sistema (operadores y agentes de venta como el Activity Center) se convierte en *inquilino* de este, y solo tiene acceso a la información de la cual son propietarios. Es más, este modelo es extensible de forma que el mismo sistema, haciendo uso de los mismos servidores, puede ser implementado en distintas áreas geográficas o *resorts* como Whistler (Canadá) o White Horse (EE.UU.).

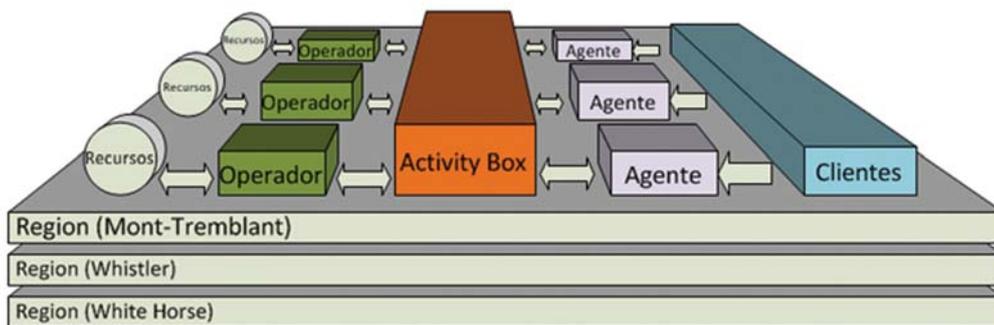


Figura 2. Componentes de Activity Box

Elaboración propia.

A la fecha de realización del estudio (febrero 2012), el sistema en su conjunto provee servicios a más de 39 operadores que administran más de 258 actividades en una red de 63 agentes de venta. Desde su lanzamiento en diciembre del 2008, Activity Box ha procesado más de 46.500 reservaciones con un monto que supera los 11 millones de dólares.

## La plataforma Coral CEA

Coral CEA es una organización sin fines de lucro que apoya a empresas con la implementación y comercialización de soluciones que utilicen mecanismos de comunicación integrados (Communications - Enabled Applications, CEA). Coral CEA es parte de un ecosistema de negocios (Moore 1993, Iansiti y Levien 2004a) basado en funcionalidades modulares de comunicación. Estos componentes enlazan las capacidades de las plataformas de redes de comunicación con la inteligencia de las aplicaciones actuales a fin de proporcionar un nuevo conjunto de servicios. Coral CEA se ha establecido como una entidad clave que provee la plataforma de servicios sobre la cual empresas complementarias como Rezact desarrollan sus actividades para el beneficio del ecosistema de negocios en su conjunto (Iansiti y Levien 2004b). Entre las entidades que colaboraron con la creación de Coral CEA se cuentan IBM, Nortel, Eclipse Foundation, la Universidad de Carleton y la Asociación de Tecnologías de Información de Canadá.

### 1. Definición del problema

A pesar de que la implementación exitosa del sistema Activity Box incrementó significativamente las ventas del Activity Center, aún existían procesos que podían ser mejorados:

- a) Muchos operadores, debido a que sus actividades se realizan en áreas dentro del Parque Nacional de Mont-Tremblant, carecían de acceso a Internet, por lo que se veían forzados a bloquear la generación de nuevas reservas hasta con un día de anticipación a fin de tener tiempo de acceder al sistema de reservas desde un punto de conexión cercano, pero con el costo de reducir así sus ventas potenciales.
- b) Una actividad puede ser cancelada por una serie de diversos motivos, generalmente por mal tiempo. Cuando esto sucede, el operador, en un proceso lento y tedioso, se veía obligado a comunicarse telefónicamente con todos sus clientes afectados por este evento para informarles que su actividad había sido cancelada.
- c) Muchas actividades requieren una preparación especial de los recursos a ser utilizados. Por ejemplo, paseos en trineos jalados por perros (*dogs*

*sledding*) requieren una preparación de por lo menos dos horas antes de la hora de inicio de la actividad. Si el cliente cancela la actividad dentro de dicho periodo, el operador puede haber desperdiciado hasta cuatro horas/hombre por evento, con el consiguiente perjuicio económico que esto representa.

- d) De forma similar, si una actividad es reservada con poco tiempo de anticipación, el operador necesita que sea notificado inmediatamente después de ocurrido el evento a fin de darle tiempo de preparar los recursos necesarios para la realización de dicha actividad.

Como se puede observar, el problema común en los tres escenarios mencionados consiste en la necesidad de comunicar a los usuarios del sistema, de una manera pronta, eficiente y eficaz, de la ocurrencia de determinados eventos previsible que afectan de una u otra forma el proceso de negocios.

## 2. Marco teórico

### 2.1. Aplicaciones con soporte para comunicaciones (CEA)

Aplicaciones con soporte para comunicaciones es el denominador común que reciben aquellos sistemas que integran componentes de tecnología de información y componentes de comunicaciones, de tal manera que ofrecen una arquitectura de servicios que busca optimizar procesos de negocio o mejorar la interacción con los usuarios de dicho sistema. CEA es el resultado natural de la evolución de las comunicaciones, en la cual la primera etapa la constituyó la convergencia de redes de datos y telefonía. La segunda etapa trajo como resultado comunicaciones unificadas (UC), y estuvo implementada por una red de telefonía integral, mensajes de texto (IM) y videoconferencias. Por último, la tercera etapa está representada por la convergencia de aplicaciones de negocio mediante la eliminación de latencias generadas por interacciones tanto entre personas como entre personas y aplicaciones (Carbone 2008). Mediante la utilización de una arquitectura orientada a servicios (SOA) es posible extender las funcionalidades de comunicaciones unificadas y convertirlas en métodos virtuales que pueden ser fácilmente utilizados por las aplicaciones de negocio.

#### 2.1.1. Arquitectura orientada a servicios (SOA)

Una arquitectura orientada a servicios (SOA) está definida como un conjunto de metodologías y patrones de diseño mediante los cuales aplicaciones e infraestructuras exponen sus funcionalidades como una serie de servicios relacionados

pero independientes entre sí. La forma más común de SOA es aquella utilizada por servicios web, en los cuales las interfaces de servicios y la forma de comunicarse con ellos son descritas utilizando el estándar Web Services Description Language (WSDL); los datos son transmitidos usando Simple Object Access Protocol (SOAP) normalmente sobre el protocolo HTTP, y opcionalmente utilizando Universal Description, Discovery and Integration (UDDI) como un servicio de directorio que permite localizar y registrar dichos servicios web.

La principal ventaja en la implementación de soluciones usando SOA es que permite interconectar aplicaciones que pueden estar alojadas en plataformas o sistemas operativos completamente disímiles entre sí. La independencia entre capas otorga una gran flexibilidad al momento de integrar aplicaciones, ya sean estas dentro de una misma organización como entre aplicaciones externas.

### 2.1.2. Software-como-servicio (SaaS)

Software-como-servicio (SaaS) es un modelo de distribución de software que ha visto un notable despegue en los últimos años. El concepto básico consiste en que los usuarios no pagan por la propiedad de una aplicación de software, y en su lugar, pagan por la utilización del mismo, convirtiéndose de esta manera en *arrendatarios* del proveedor de software que produce y mantiene la aplicación.

Si bien el concepto detrás de SaaS es bastante simple, su implementación ha ido evolucionando de acuerdo a las mejoras tecnológicas en las plataformas, especialmente de servidores, sistemas operativos y bases de datos. En la primera etapa, los proveedores de aplicaciones SaaS se veían forzados a crear una instancia de la aplicación por cada usuario del sistema, creando una barrera de mantenimiento bastante alta. SaaS avanzó a su segunda etapa cuando dichos proveedores permitieron a los usuarios de las aplicaciones que ellos mismos personalizaran su experiencia con la aplicación, a pesar de que aún existía una instancia por cliente. En la tercera etapa, solo existe una instancia de la aplicación que sirve a todos los clientes, pero la data de distintos clientes se encuentra separada de forma lógica (Chong y Carraro 2006).

La evolución natural de estas tecnologías ha llevado a la extensión del concepto para ser aplicado no solo a aplicaciones de software sino también a la infraestructura en sí misma, donde el recurso computacional es alquilado de acuerdo con el uso. Este concepto, inicialmente llamado Infraestructura como servicio (IaaS), ha sido recientemente llevado a la práctica bajo la denominación de *cloud computing* por empresas tales como Microsoft con su servicio Azure, Google con su servicio Google Apps, o Amazon con Elastic Compute Cloud (EC2).

## 2.2. Desarrollo e implementación

Para la implementación del proyecto, se optó por el desarrollo de una versión paralela del sistema Activity Box en producción a fin de reducir los riesgos al proceso de negocios que se encontraba en funcionamiento en ese momento. Para el desarrollo de dicha versión se tuvieron en cuenta las siguientes consideraciones:

- Esta versión de Activity Box debería tener un conjunto reducido de funcionalidades del sistema en producción, pero usando los mismos componentes con la finalidad de facilitar su eventual implementación en el sistema real.
- Al mismo tiempo, era necesario implementar las funcionalidades proporcionadas por Coral CEA, de tal manera que tanto usuarios potenciales como actuales del sistema pudieran apreciar los beneficios de la utilización de dicha plataforma.

A fin de balancear ambos requerimientos, de entre la amplia gama de servicios ofrecidos por Coral CEA, se decidió por la implementación del servicio de llamadas automáticas, cuya aplicación dentro del sistema fue dividida en dos componentes o funcionalidades que ayudaron directamente a resolver los problemas identificados anteriormente:

- a) Comunicar-a-cliente: Mediante esta funcionalidad, una reservación para una actividad es cancelada en el sistema por el operador, ya sea por mal tiempo u otras causas, el aplicativo debía automáticamente llamar al cliente afectado comunicándole la cancelación de su actividad.
- b) Comunicar-a-operador: De forma análoga, cuando un cliente cancela una actividad ya reservada (la cancelación en el sistema es realizada mediante un agente de ventas), el aplicativo debía llamar al operador de dicha actividad para notificarle de la cancelación. Adicionalmente, el operador sería notificado telefónicamente en el caso de una nueva reserva.

Activity Box es un aplicativo web altamente transaccional desarrollado en la plataforma Microsoft Asp.Net sobre una base de datos MS-SQL. El dominio principal del sistema es *Activity Box.ca* por lo que para la implementación del proyecto se creó un subdominio llamado *coraldemo.Activity Box.ca* y se realizaron los ajustes correspondientes en el servidor web (Windows Server 2003 utilizando Internet Information Manager IIS v.5.0). Finalmente, se utilizó una copia de la base de datos de producción a la cual se le eliminó información sensible como números de tarjetas de crédito.



### 2.3. Servicios de Coral CEA

Las funcionalidades de Coral CEA son proporcionadas como servicios web usando SOAP como protocolo de comunicaciones. La plataforma proporcionaba dos interfaces programables de programación (API) o puntos de entrada para la ejecución de servicios:

- a) Control de llamadas a terceros v2 (TPCv2): Permite la creación de llamadas en tiempo real entre dos entidades, la entidad *origen* que inicia la llamada y la entidad *destino* que responde a dicha llamada. Debido a que Activity Box no tenía el requerimiento de una comunicación en tiempo real entre operadores, agentes de venta y clientes, se decidió por la no implementación de esta interface.
- b) Control de llamadas a terceros v3 (TPCv3): Permite la creación de enlaces de comunicación no interactivos entre una o múltiples entidades. Uno de los servicios de esta interface en particular, llamada *Audio Call* permite a una aplicación la ejecución, durante una llamada, de un mensaje pregrabado entre uno o más participantes de la llamada, así como el monitoreo y control del estado de dicha llamada. Este servicio fue finalmente elegido para implementar durante la etapa de ejecución del proyecto.

Estos servicios estaban basados en Parlay X, un conjunto de estándares de servicios web para las redes de telefonía fija y móvil. Cabe mencionar que dicho estándar ha sido recientemente reemplazado por OneAPI bajo la tutela del grupo GSMA, el cual representa los intereses de operadores telefónicos de más de 220 países.

Para la fase de ejecución del proyecto, y en función de las funcionalidades disponibles en la plataforma, se analizaron los siguientes aspectos:

- a) *Texto-a-voz (TTS) multi-lenguaje*: Las funcionalidades requeridas para la implementación del proyecto, que fueron descritas anteriormente (Comunicar-a-cliente y Comunicar-a-operador), requerían que el mensaje transmitido al destinatario tuviera la mayor cantidad posible de información, tales como el nombre de la actividad, el código de reserva o la fecha y hora de inicio de dicha actividad. Activity Box, al momento del desarrollo del proyecto, manejaba más de 250 actividades entre verano e invierno, por lo que la posibilidad de contar con un servicio que convirtiera texto a voz (text-to-speech, TTS) se convirtió en un requerimiento crítico. Adicionalmente, había que considerar que cualquier comunicación tenía que realizarse, por ley, en alguno de los dos idiomas oficiales de Canadá, inglés o francés. Lamentablemente, esta funcionalidad no había sido aún implementada en la plataforma al momento de la puesta en ejecución de este proyecto. En su lugar, se optó por la opción de transmitir únicamente mensajes previamente grabados, los cuales estarían relacionados con las actividades registradas en el sistema.

- b) *Llamadas síncronas*: Los servicios disponibles en la interfaz programable (APIs) de Coral CEA son de naturaleza síncrona, lo que significaba que la aplicación necesitaba esperar para recibir una respuesta inicial de la plataforma una vez que la solicitud del servicio haya sido realizada. Este comportamiento, sin embargo, podría afectar negativamente a un sistema altamente transaccional como Activity Box, creando contención en el servidor web y reduciendo la escalabilidad general futura del sistema si es que se implementaba en el sistema real. Llamadas a la plataforma a través de API necesitaban entonces ser desvinculadas de la aplicación web de tal manera que la *performance* del sistema y la interacción con sus usuarios no se viera afectada.
- c) *Confirmación de llamadas*: Cada intento de comunicación generado por Activity Box necesitaba ser registrado para propósitos de auditoría, de tal forma que los interesados (agentes de venta u operadores) podían más tarde determinar si el afectado por un evento fue adecuadamente notificado. Para este efecto, la plataforma proporciona una serie de códigos de respuesta que indican el estado de una llamada (en proceso, terminada, etcétera), los cuales son mantenidos en el servidor de Coral CEA por solo diez minutos. Si bien es cierto que existían planes para crear un mecanismo que permitiera a aplicaciones existentes *escuchar* por actualizaciones de estado enviadas por la plataforma, al momento de la ejecución del proyecto no existían estimados de tiempo para su implementación.
- d) *Llamadas concurrentes*: El límite de llamadas concurrentes soportadas por la plataforma se calculó en ese momento en ocho. Aunque los detalles técnicos de la plataforma en términos de capacidades de comunicación se encontraban fuera del alcance del proyecto, se consideró necesario disponer de un mecanismo de amortiguamiento que permitiera soportar un número ilimitado de solicitudes de llamadas sobre un limitado número de líneas telefónicas. Coral CEA no proporcionaba este tipo de mecanismo.
- e) *Servicios adicionales*: Ejemplos de funcionalidades adicionales que sería interesante disponer podrían ser la capacidad de enviar faxes, la programación de una llamada para ser ejecutada a determinada fecha y hora, el envío de mensajes de texto, o la interacción entre el destinatario de la llamada y la plataforma (como solicitar que un mensaje sea repetido). Estos servicios adicionales no eran proporcionados por Coral CEA al momento de la ejecución del proyecto.

### 2.3.1. Agente Coral CEA

Para aplicaciones como Activity Box donde el tiempo de respuesta es un factor crucial, esperar que una entidad externa genere alguna respuesta puede tener efectos negativos en la aplicación. El método de respuesta síncrono ofrecido por Coral CEA

podría haber generado contención y problemas de escalabilidad en el aplicativo. Adicionalmente, el mecanismo ofrecido por la plataforma para obtener el estado de una llamada escapaba más allá de la arquitectura tradicional utilizada en aplicaciones web. Finalmente, las limitaciones inherentes en la plataforma con respecto al número máximo de llamadas concurrentes habrían generado la implementación de mecanismos de validación adicionales en el aplicativo. Por estas razones se decidió que la mejor alternativa era separar, en la medida de lo posible, la implementación de los servicios de Coral CEA del aplicativo en sí mismo a fin de aislar Activity Box de cualquier problema operacional.

Para este efecto se diseñó un agente intermedio entre Coral CEA y Activity Box. El mecanismo propuesto tenía la siguiente estrategia: cada vez que se detectara un evento en Activity Box que requiriera la ejecución de una llamada a algún destinatario, el aplicativo almacenaría la solicitud en una tabla común llamada CEA Queue, incluyendo el nombre del archivo de audio asociado al evento y actividad. Este agente intermedio, implementado como un servicio del sistema operativo instalado en el servidor, se encargaría de consultar continuamente esta tabla en la base de datos a fin de detectar una nueva solicitud. Una vez que esto sucedía, el agente de software se encargaría de reenviar la solicitud a la interface de programación de Coral CEA (API), además de consultar continuamente acerca del estado de esta solicitud hasta que se obtenga un estado final. Cada cambio de estado es actualizado en la tabla CEA Queue, de tal forma que la aplicación cliente puede siempre estar informada del estado actual de la solicitud.

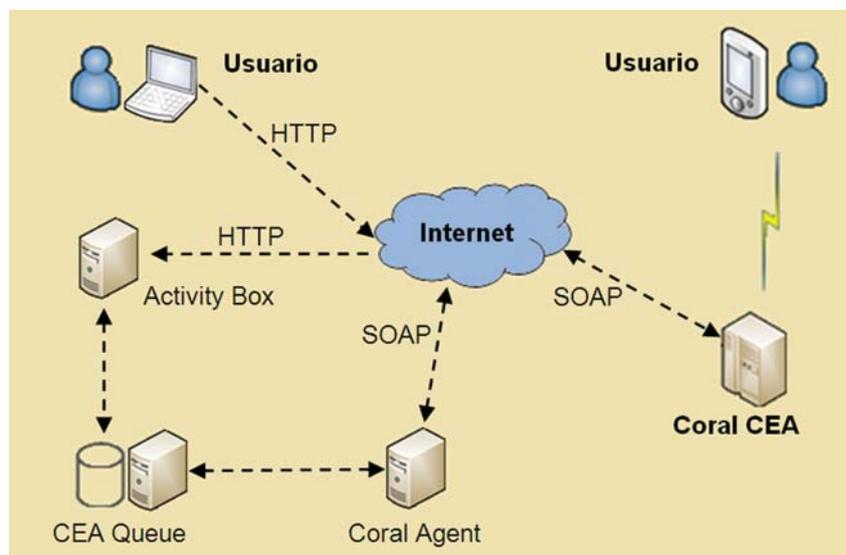


Figura 3. Diseño de la solución.

Elaboración propia.

El pseudocódigo que es ejecutado por el agente es mostrado a continuación:

```

List = GetList(CEA Queue)
For Each Call on List
  RequestCallToCoralCEA;
  Update(CEA Queue);
  If (CallStatus == Initializing) Then
    Update(CEA Queue);
    Do While ParticipantStatus <> Terminated
      If (ParticipantStatus = Connected)
        RequestAudioToCoralCEA;
        Update(CEA Queue);
      End If
      GetParticipantStatus;
    End Do
    Update(CEA Queue);
  End If
Next Call on List

```

Se añadieron otras funcionalidades complementarias al agente, como, por ejemplo, la posibilidad de reintentar una llamada un determinado número de veces si es que el primer intento no prosperó por razones técnicas (otras que no sean que el cliente colgó el teléfono); o que solo se permitan efectuar llamadas entre determinadas horas del día, para evitar llamadas a clientes a horas no apropiadas. El agente también encapsulaba los mecanismos necesarios para interactuar con Coral CEA, y los exponía como parámetros en el archivo de configuración. La lista de parámetros que podían usarse se muestra a continuación:

Parámetro	Valor por defecto	Descripción
DBPollInterval	30	Intervalo en segundos en el cual el agente debería consultar la tabla CEA Queue en busca de nuevas solicitudes.
ACEServerURL	Determinado por Coral CEA	Interface de servicio de Coral CEA.
ACEAudioServerURL	Determinado por Coral CEA	Interface de servicio de Coral CEA para la ejecución de mensajes pregrabados.
ACEUser	Determinado por Coral CEA	Usuario para comunicarse con Coral CEA.
ACEPassword	Determinado por Coral CEA	Clave del usuario.

(continúa)

(continuación)

Parámetro	Valor por defecto	Descripción
ACETimeout	20	Tiempo en segundos antes de que un intento de comunicación con Coral CEA expire.
CallAttempts	4	Máximo número de intentos de comunicación con Coral CEA.
LogEntries	1	Indica si se deben registrar eventos en el archivo de registros del servidor (0: NO, 1: SI).
StartHour	8	Hora de inicio en la cual el agente puede empezar a procesar llamadas en Coral CEA.
EndHour	20	Hora en la que el agente ya no puede registrar llamadas en Coral CEA.
ParticipantPrefix	Sip:9	Cualquier cadena de caracteres que necesite ser insertada antes de un número telefónico regular.
ParticipantSuffix	@134.117.254.226	Cualquier cadena de caracteres que necesite ser insertada después de un número telefónico regular.
AudioURLPrefix	play=	Cualquier cadena de caracteres que necesite ser insertada antes del nombre del archivo de audio.
AudioURLSuffix	;early=no;locale=en_us	Cualquier cadena de caracteres que necesite ser insertada después del nombre del archivo de audio

Tabla 1. Parámetros configurables del agente

Elaboración propia.

Mediante este mecanismo que aislaba Activity Box de Coral CEA, no solo se proporcionó una red de seguridad para el aplicativo en caso de fallas o problemas en la plataforma de Coral CEA, sino que también se mitigaron las modificaciones necesarias en el aplicativo para incluir los servicios de Coral CEA. Dichas modificaciones se redujeron a simples cambios en los procedimientos almacenados en la base de datos que se ejecutaban durante los eventos que se deseaban controlar. De forma análoga, para proporcionar el estado actualizado de estas llamadas, Activity Box solo necesitaba consultar esta nueva tabla sin necesidad de realizar continuas llamadas remotas al servicio de Coral CEA.

Más aún, con este diseño era fácilmente posible ofrecer funcionalidades de Coral CEA a otras aplicaciones distintas de Activity Box siempre y cuando estas actualizaran la tabla CEA Queue, debido a que el agente era completamente independiente del aplicativo que generaba la solicitud. La limitación existente por la cual Coral CEA solo almacenaba información acerca de la llamada por un lapso reducido (entre 30 segundos y 10 minutos) también quedaba eliminada al guardarse el estado de la solicitud de forma local. Finalmente, este mecanismo podía ser ofrecido a otros miembros del ecosistema que enfrentaran problemas similares, o incluso podía ser implementado con ciertas modificaciones por la misma plataforma.

## 2.4. Tabla CEA Queue

Al centro de la solución se encuentra la tabla CEA Queue cuyo único propósito es servir como punto de contacto entre cualquier aplicación cliente y el agente o servicio encargado de comunicarse directamente con la plataforma Coral CEA. El diseño de esta tabla permitía al aplicativo cliente, en este caso Activity Box, requerir llamadas al agente, quien a su vez se encargaba de realizar la solicitud a la plataforma Coral CEA. Aunque en la versión implementada solo se tenía soporte para un participante a la vez, el diseño podía extenderse para soportar varios participantes por llamada.

Entre las funcionalidades incluidas mediante la utilización de esta tabla se puede mencionar la posibilidad de indicar la prioridad con la cual las llamadas deben ser solicitadas a Coral CEA. Esta opción es útil en circunstancias donde existan pocas líneas telefónicas disponibles y se necesite asegurar que las llamadas más importantes sean ejecutadas primero. En el caso de Activity Box, por ejemplo, se asignó mayor prioridad a las llamadas relacionadas con cancelaciones de actividades, seguidas por las llamadas generadas por nuevas reservaciones.

La tabla también es utilizada para indicar al agente cuál archivo de audio debería transmitirse una vez que la llamada tomaba lugar. Para Activity Box, los archivos de audio están asociados con cada actividad, así como con cada uno de los eventos controlados. Es en esta tabla donde también se registran el número de intentos de comunicación realizados por el agente. Finalmente, se proporcionaban una serie de Campos-Definidos-por-el-Usuario (UDF) que permitían a las aplicaciones cliente incluir información útil en cada llamada. En Activity Box, por ejemplo, se utilizaron dos de estos campos para indicar el número de reservación y el nombre del destinatario de la llamada.

La estructura general de la tabla utilizaba terminología propia de Coral CEA, pero se simplificó su diseño y se añadieron campos adicionales para soportar la funcionalidad descrita líneas arriba. Los campos de la tabla son descritos a continuación:



<b>Campo</b>	<b>Descripción</b>
CallSessionIdentifier varchar(50) NOT NULL	Identificador único (retornado por la operación make-CallSession) que identifica la sesión de llamada.
CallParticipantStatus int NOT NULL DEFAULT 99	Estado del participante de la llamada. Posibles valores: 99: Llamada pendiente. 0: callParticipantInitial (llamada al participante en progreso). 1: callParticipantConnected (participante activo en llamada). 2: callParticipantTerminated (llamada al participante terminada).
CallParticipantStartTime varchar(50) NULL	Fecha y hora en la que el participante fue añadido a la llamada. Este parámetro solo es registrado si la llamada al participante fue retornada exitosamente (si CallParticipantStatus es diferente de callParticipantInitial).
CallParticipantDuration int NULL DEFAULT 0	La duración en segundos de la llamada si el participante estuvo activo en ella. Este parámetro solo es relevante si el participante ya no se encuentra activo en la llamada (si CallParticipantStatus es callParticipantTerminated).
CallParticipantTerminationCause int NULL	La cause de terminación de la llamada. Este parámetro solo es relevante si callStatus = callTerminated. Posibles valores: 0: CallingPartyNoAnswer 1: CalledPartyNoAnswer 2: CallingPartyBusy 3: CalledPartyBusy 4: CallingPartyNotReachable 5: CalledPartyNotReachable 6: CallHangUp 7: CallAborted
LastCallFaultCode varchar(500) NULL	Último código de error tal como fue reportado por Coral CEA.
CallParticipant varchar(50) NULL	Dirección del participante incluido en la llamada.
AudioURL varchar(500) NULL	Nombre del archivo de audio a ser transmitido durante la llamada.

(continúa)

(continuación)

<b>Campo</b>	<b>Descripción</b>
Audioidentifier varchar(50)	Identificador único de la solicitud para transmitir un mensaje grabado (retornado por la operación playAudioMessage).
AudioStatus int NULL	Estado del mensaje de audio. Valores posibles: 0: Played 1: Playing 2: Pending 3: Error
Attempts int 0	Número de intentos realizados por el agente para comunicarse con Coral CEA.
NextAttemptUTC datetime GETUTCDATE()	Fecha y hora del siguiente intento de comunicación con Coral CEA, indicado en formato UTC ( <i>Coordinated Universal Time</i> ).
CallPriority int NOT NULL DEFAULT 1	Prioridad de la llamada. Cuanto mayor sea el valor, mayor será la prioridad.

Tabla 2. Estructura de la tabla CEA Queue

Elaboración propia.

### 3. Cambios en el aplicativo

Debido a que el manejo de la comunicación con Coral CEA es realizado por el agente, las modificaciones requeridas en Activity Box fueron reducidas a un mínimo. En primer lugar, se modificaron las rutinas de base de datos que controlaban la creación y cancelación de reservas a fin de que insertaran la información requerida en la tabla CEA Queue descrita anteriormente. La información añadida incluía el tipo de evento (nueva reserva o cancelación), el número de reservación asociada con la llamada y su prioridad. La segunda modificación estuvo relacionada con proporcionar a los usuarios del sistema, mecanismos de verificación de las llamadas requeridas. Para este efecto, se añadieron en determinados puntos del sistema, enlaces que mostraban una información más detallada de la solicitud, tales como el evento que ocasionó la llamada, el número telefónico utilizado, el estado actual, la duración de la llamada y la causa de su terminación. Estos cambios pueden apreciarse en la figura siguiente:

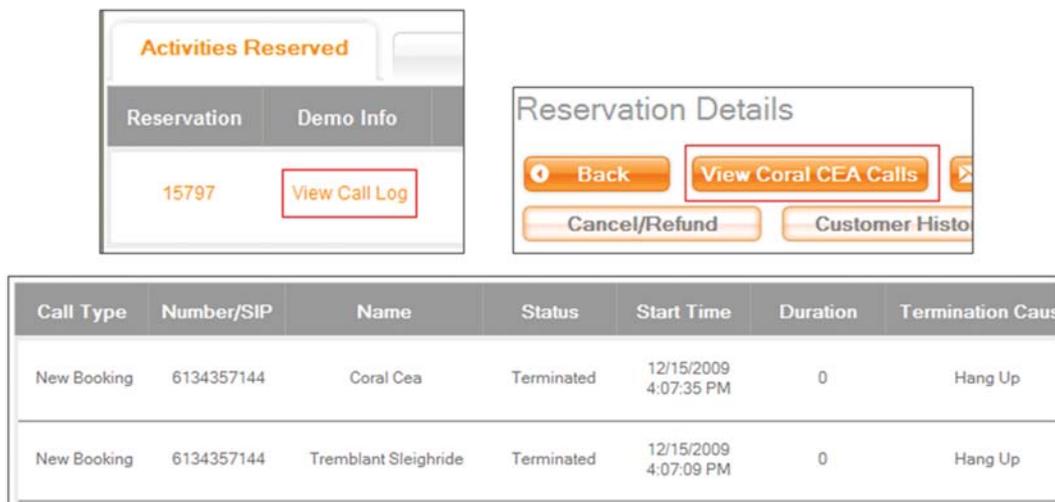


Figura 4. Cambios en la interface de Activity Box

Elaboración propia.

### 3.1. Plataformas similares

Coral CEA no es la única plataforma que provee servicios de comunicación unificada. En años recientes han aparecido algunas plataformas con un catálogo de servicios similar al de Coral CEA. Entre ellas se puede mencionar Twilio ([www.twilio.com](http://www.twilio.com)), una empresa ubicada en San Francisco, California, orientada a desarrolladores, y con la particularidad de que sus servicios son ofrecidos como *Infrastructure as a Service* (IaaS) y son accedidos usando el protocolo HTTP. Twilio posee cerca de 75.000 desarrolladores.

Voxeo ([www.voxeo.com](http://www.voxeo.com)) es una empresa ubicada en Orlando, Florida, que ofrece soluciones utilizando *Interactive Voice Response* (IVR) o *Voiceover IP* (VoIP). Voxeo se presenta como una plataforma SaaS, *Platform as a Service* (PaaS) o combinaciones híbridas de ambas. La empresa reportó alrededor de 200.000 desarrolladores en su plataforma a fines del 2011.

## 4. Resultados

- a) La implementación de una prueba de concepto de la utilización de servicios de Coral CEA basado en una aplicación existente y en pleno funcionamiento. La documentación detallada del desarrollo de dos servicios (Comunicar-A-Cliente y Comunicar-A-Operador) permitió a los directivos de Rezact evaluar la

factibilidad de una implementación real en el futuro, así como de los beneficios esperados.

- b) Los servicios proporcionados por Coral CEA al momento de la ejecución del proyecto presentaban ciertas limitaciones. Con la finalidad de mitigar su efecto se desarrolló un agente de software (Agente Coral CEA) que se comporta como un mecanismo de amortiguamiento entre cualquier aplicación cliente y Coral CEA. El código fuente de dicho agente fue entregado a los responsables de Coral CEA para que sea utilizado como punto de partida para su propia implementación.
- c) Coral CEA reconoció la necesidad de contar con servicios esenciales como texto-speech y mensajes de texto.
- d) Se incrementó la visibilidad y exposición al mercado en general, y al ecosistema de negocios en particular, de Rezact y su producto Activity Box.

## 5. Conclusiones y recomendaciones

### 5.1 Conclusiones

- a) La plataforma de Coral CEA posee los elementos tecnológicos clave, necesarios para la creación de un ecosistema de negocios exitoso. El valor ofrecido por Coral CEA puede ser considerablemente incrementado si se implementan servicios complementarios, tales como TTS o envío de mensajes de texto.
- b) Es relativamente sencillo, desde el punto de vista de aplicaciones comerciales existentes, de implementar características de comunicación usando Coral CEA. Ciertas limitaciones encontradas durante la implementación del proyecto pueden ser resueltas con un adecuado esfuerzo de desarrollo, por lo menos hasta que dichas limitaciones sean resueltas por la plataforma misma.
- c) Un factor clave para una implementación exitosa de los servicios de Coral CEA es el aislamiento necesario entre dicha plataforma y las aplicaciones que hacen uso de dichos servicios. Esto puede lograrse usando el agente descrito en el proyecto, o con otros mecanismos similares ajustados a las necesidades y escenarios particulares de cada aplicación.

### 5.2. Recomendaciones

#### 5.2.1 Para Coral CEA

- a) Un servicio de texto-a-voz (TTS) es fundamental y debería ser implementado como parte de los servicios de Coral CEA. También se recomendó que cualquier implementación de servicios TTS debiera considerar por lo menos los dos idiomas oficiales de Canadá (inglés y francés).

- b) Se sugirió incrementar el tiempo durante el cual el estado de una llamada es mantenido en los servidores de Coral CEA por lo menos a un día. Un lapso mayor permitiría que procedimientos automáticos en lote (*batch*) en el cliente puedan ser ejecutados fuera de las horas pico (por ejemplo, a medianoche) para consultar y actualizar el estado de llamadas requeridas durante el día.
- c) La capacidad de programar el día y la hora de la ejecución de una llamada es un servicio que Coral CEA podría implementar entre sus servicios, de forma relativamente fácil, si dispusiera de un mecanismo de amortiguamiento similar al descrito para el agente utilizado en este proyecto. Además, esto eliminaría la necesidad de los clientes de consultar continuamente los servidores de Coral CEA para averiguar el estado de una solicitud, liberando así capacidad de procesamiento en los servidores que puede ser utilizada para otras tareas.
- d) Se consideró que no todas las aplicaciones cliente necesitarían utilizar servicios TTS. Algunas pueden depender en gran medida del uso de mensajes pre-grabados. Al momento de la ejecución del proyecto, Coral CEA no contaba con ningún mecanismo automatizado que permitiera *subir* al servidor dichos archivos de audio. Adicionalmente, los archivos de audio requerían estar en un formato especial,<sup>1</sup> lo cual añadía un nivel más de complejidad a potenciales usuarios de la plataforma. Se recomendó la creación de un servicio web que permitiera a aplicaciones cliente enviar sus archivos de audio, y que encapsulara cualquier conversión de formato a los requeridos por la plataforma.
- e) Desarrollar métodos de llamada asíncronos a Coral CEA que permitan a aplicaciones cliente que sean sensibles al tiempo, ejecutar solicitudes de servicio sin necesidad de esperar por una respuesta de retorno, permitiéndoles continuar con el proceso de negocio que generó la solicitud.
- f) Algunas aplicaciones cliente podrían necesitar una forma de comunicación menos intrusiva, como lo es el envío de mensajes de texto a teléfonos celulares. Este servicio no se encontraba disponible al momento de la ejecución del proyecto.
- g) Se recomendó añadir funcionalidades básicas que permitieran a los destinatarios de una llamada proporcionar información de regreso útil a la plataforma. Un ejemplo de esto podría ser la posibilidad de que el usuario, presionando una tecla de su teléfono, pudiera solicitar que el mensaje sea repetido. Un mecanismo similar podría permitir al destinatario rechazar una llamada, lo cual indicaría a la plataforma que el número telefónico utilizado en la aplicación cliente fue erróneo.

---

1 Linear PCM, 8 bit little-endian unsigned integer, 1 channels, 8000 Hz.

### 5.2.2 Para clientes de la plataforma

- a) Cualquier interacción con Coral CEA debería ser tratada de forma independiente y fuera del proceso regular o del evento que generó la llamada, a fin de prevenir escenarios en los cuales el factor limitante es un restringido número de líneas telefónicas disponibles, o cuando el aplicativo necesita esperar una respuesta. Una forma de obtener este resultado consiste en la implementación de un servicio o agente como el descrito en el proyecto.
- b) Basado en la experiencia durante la implementación del proyecto, se estimó que miembros del ecosistema que planeen implementar servicios de Coral CEA deberían considerar una duración del proyecto de por lo menos un mes, lo cual, obviamente, puede variar dependiendo del tamaño de la aplicación cliente y del número de servicios a implementar.
- c) Otro factor que refuerza la recomendación del uso de un agente o servicio para manejar las comunicaciones con Coral CEA es el hecho de que, una vez que dicho mecanismo está implementado, los cambios en el aplicativo cliente se reducen a las interacciones con la tabla CEA Queue o equivalente, reduciendo así el riesgo en el aplicativo.
- d) Aplicaciones cliente que planeen implementar servicios CEA, deberían estimar cuidadosamente el número de eventos, durante determinado espacio de tiempo, que pueden generar solicitudes de llamadas, especialmente si es que no se utiliza un mecanismo de amortiguamiento como el agente descrito en el reporte, debido a que se corre el riesgo de sobresaturar los servidores de Coral CEA.

### Bibliografía

- Banerjee, Nilanjan y Dasgupta, Koustuv (2008). *Telecom mashups: enabling web 2.0 for telecom services*. ICUIMC 2008. Nueva York: ACM.
- Carbone, Peter (2008). *Communications enabled applications – an evolution to the next value plane?* Massachusetts: OASIS Symposium, Open Standards.
- Cárdenas, Daniel (2009). "A CEA practical implementation: activity box". Tesis para optar el grado de magíster en gestión de innovación tecnológica. Ottawa: Carleton University.
- Chong, Frederick y Gianpaolo Carraro (2006). *Architecture strategies for catching the long tail*. Washington: Microsoft Corporation. <<http://msdn.microsoft.com/en-us/library/aa479069.aspx>>. [Consulta: julio del 2012].

- Cusumano, Michael y Annabelle Gawer (2002). "The elements of platform leadership". MIT Sloan Management Review, Spring, pp. 51-58. Massachusetts.
- Jansiti, Marco y Roy Levien (2004). "Strategy as ecology". *Harvard Business Review*, March, pp. 68-78. Massachusetts.
- . (2004). *The keystone advantage*. Massachusetts: Harvard Business School Press.
- Lüthje, Christian y Cornelius Herstatt (2004). "The lead user method: An outline of empirical findings and issues for future research". *R&D Management* 34 (5), pp. 553-568. Manchester.
- Manchester, Moore, James (1993). *Predators and prey: a new ecology of competition*. *Harvard Business Review* 71 (3), pp. 75-86. Massachusetts.
- Von Hippel, Eric (2005). *Democratizing innovation*. Cambridge: The MIT Press. <<http://mitpress.mit.edu/books/0262002744/0262002744.pdf>>. [Consulta: julio del 2012].