

Fecha de confección:	08/08/2007
Título:	Arquitectura Tecnológica Simple
Temática:	Aplicativos .NET
Confeccionado por:	Marcelo Oviedo
Revisado por:	
Aprobado por	
Clasificación:	Público
Auditorio:	
Fecha de Impresión:	13/04/2011 05:41:26
Versión	1.0

Alcance

Esta normativa es aplicable a todo desarrollo de aplicaciones en .NET, ya sea internas de Grupo Action Line o Tercerizadas.

Elaboración / Revisión / Aprobación

El presente documento es elaborado por el Arquitecto Líder de la División QA, Revisado por el Responsable de División QA y aprobado por la Gerencia de Desarrollo.

Roles y Responsabilidades

Rol	Responsabilidad
Gerencia de desarrollo	Aprobar el estándar de Desarrollo.
Arquitecto Líder	Definir Estándares y auditar el estándar.
Responsable División QA	Revisión general del estándar.

Introducción

Las Soluciones de Negocio Corporativas deben, actualmente, enfrentar muchas complejidades tecnológicas y responder a altas expectativas de sus usuarios. No solo tienen que responder a variables relacionadas con Alta Disponibilidad, Escalabilidad y Alto Rendimiento, muchas veces en un contexto poco predecible, sino que también tienen que dar respuestas a un negocio altamente cambiante sin afectar la calidad de las soluciones y manteniendo el mejor nivel de productividad posible.

Las mejores soluciones son aquellas compuestas por un conjunto de simples mecanismos que resuelven escenarios recurrentes en forma confiable y eficiente. Durante el proceso de desarrollo de grandes soluciones, la combinación de estos simples mecanismos permite la creación de sistemas con requerimientos complejos.

Este documento define una *Arquitectura para Aplicaciones Corporativas* como la suma de patrones que resuelven problemas simples, aplicando a su vez un conjunto de buenas prácticas, probadas e implementadas en sistemas de gran envergadura.

Por otro lado, se ha elegido utilizar Plataforma .NET (Windows 2005, Framework .NET 2.0, VS.NET, Etc.) para soportar toda la problemática tecnológica que deriva de los tres puntos enunciados con anterioridad.

Propósito

El presente documento tiene como propósito definir el diseño de la arquitectura tecnológica sobre la cual se implementará toda solución de Action Line.

Dicho documento definirá un modelo prescriptivo que permita a Action Line desarrollar con una arquitectura basada en las siguientes variables:

- Alta disponibilidad
- Escalabilidad
- Productividad
- Performance

Es importante destacar que a efectos de contar con una adecuada transferencia de conocimientos, la elaboración de dicho documento es producto de un trabajo en conjunto de los distintos equipos de trabajo de Action Line y del área QA que centraliza la información de una forma unificada.

Audiencia

Dicho documento debe ser material de lectura para los siguientes equipos de trabajo:

- Equipo de arquitectura de Action Line SA, compuesto por recursos de la compañía y de socios de negocio.
- Equipo dedicado a llevar a cabo el diseño lógico del Sistema en cuestión.
- Equipo de desarrollo compuesto por recursos de la compañía y de socios de negocio.
- Program Manager de la solución que se está desarrollando.
- Todo recurso técnico que Action Line considere necesario.

Estructura del documento

El documento esta compuesto por una sección que define el diseño de arquitectura conceptual y tecnológica. La sección que define el diseño de la arquitectura de software se encuentra bajo el nombre ["Arquitectura para aplicaciones corporativas"](#).

Definiciones, acrónimos y abreviaturas

La siguiente tabla contiene la lista de definiciones, acrónimos y abreviaturas que se aplicarán en el documento.

Definición / Acrónimo / Abreviatura	Descripción
MS	Microsoft
BC	Business Components / Componentes de negocio
SSC	Service Support Component
BSC	Business Support Component
BE	Business Entities / Entidades de negocio
SC	Servicio de consulta.
ST	Servicio transaccional.
DALC	Data Access Layer Components / Capa de componentes de acceso a los datos.
UI	User Interface / Interfase de usuario
SP	Stored Procedure / Procedimiento almacenados en la Base de Datos que permiten el acceso y mantenimiento de los datos.
DER	Diagrama de Entidad-Relación (Modelo de datos)
DB	Database / Base de datos
Roundtrip	Envío de requerimientos y espera de respuestas entre las diferentes capas físicas de la arquitectura. Ejemplos: Accesos a la DB, Ejecución de servicios de consulta o transaccionales.
DS	DataSet, estructura de datos que se utilizará para representar entidades.
DS Tipificados	Son DSs que contienen propiedades, métodos y eventos que están fuertemente ligados a una entidad
Request	Define al conjunto de datos que será enviado a la capa de servicios, desde la capa física de presentación y que corresponderá al parámetro de entrada de un servicio transaccional o de consulta.
Response	Define al conjunto de datos retornado por un servicio transaccional o de consulta.
CRUD	Acrónimo (inglés) que sintetiza las operaciones básicas sobre una entidad: C = Crear, R = Leer, U = Actualizar, D = Borrar.
ACID	Acrónimo (inglés) utilizado para definir el nivel de integridad y consistencia de datos que garantiza un sistema. A = Atomicidad, C = Consistencia, I = Aislamiento, D = Durabilidad.
MVC	Acrónimo (inglés) utilizado para definir el modelo de interacción entre la capa de presentación y el modelo del negocio. M = Modelo, V = Vista, C = Controlador.
OLTP	On-Line Transactional Processing, este concepto hace referencia al sistema transaccional de un sistema de información.
DAC	Data Access Component – Componentes de Acceso a Datos.
SOA	Service Oriented Applications – Aplicaciones orientadas a la ejecución de servicios de negocio.
SN	Servicio de Negocio – Se define como servicio de negocio a toda clase .NET que hereda de BusinessComponent , convirtiendo de esta manera a la clase en un servicio que expone lógica de negocio a los diferentes canales de acceso.
QA	Quality assurance (QA)

Arquitectura para aplicaciones corporativas

La gran mayoría de los sistemas se basan en la obtención de datos de un repositorio, presentarlo a través de una determinada tecnología al usuario y luego impactar los cambios realizados por el usuario al repositorio de datos. Dado que el comportamiento de los sistemas de información se acerca al supuesto mencionado, es posible que los arquitectos de soluciones se inclinen a acoplar la capa de datos a la de presentación. Si bien esto es posible trae algunos inconvenientes:

La lógica de presentación cambia con mayor frecuencia que la lógica de negocio.

Cada vez es mas necesario presentar la misma información en diferentes tipos de tecnologías / dispositivos.

Existen necesidades de conocimientos técnicos diferentes para desarrollar lógica de presentación respecto al desarrollo de lógica de negocio.

Complejidad para detectar errores dentro de la solución.

Dada la distribución de capas lógicas, encontramos que el patrón MVC (Model – View – Controller), es el mas adecuado para dar soporte a las necesidades y requerimientos del sistema transaccional. Bajo este modelo la presentación de datos al usuario representa a la *Vista*, el encargado de ejecutar una acción en función de un evento de entrada es el *Controlador*, y el conjunto de capas lógicas definidas bajo los Servicios de Negocio representan al *Modelo*.

DISEÑO CONCEPTUAL DEL MODELO MVC

En esta sección se proveen las definiciones básicas correspondientes al modelo MVC del sistema transaccional del Sistema.

Modelo

El modelo esta compuesto por una serie de servicios y componentes conceptualmente bien identificados y con objetivos y responsabilidades diferentes.

- División en 3 áreas
 - Modelo: Encapsula los datos y la información
 - Vista: Muestra la información al usuario
 - Controlador: Interpreta las acciones del usuario
- Ventajas
 - Separación total entre lógica de negocio y presentación.
 - A esto se le pueden aplicar opciones como el multilenguaje
 - Distintos diseños de presentación.

El siguiente gráfico visualiza las capas lógicas que conforman dicho modelo:

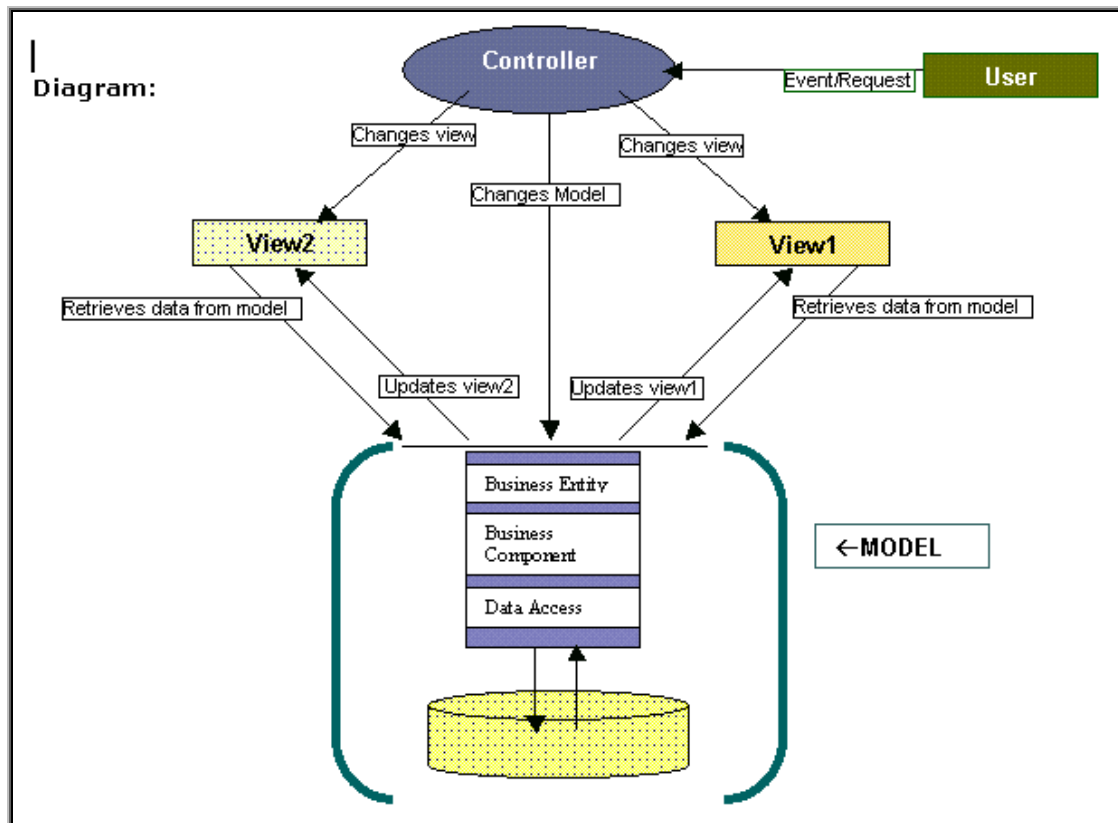


Figura 1: Capas Lógicas del Modelo

Tipos de componentes

El análisis de la mayoría de las soluciones empresariales basadas en modelos de componentes por capas muestra que existen varios tipos de componentes habituales. En la figura 2 se muestra una ilustración completa en la que se indican estos tipos de componentes.¹

Aunque la lista que se muestra en la figura 2 no es completa, representa los tipos de componentes de software más comunes encontrados en la mayoría de las soluciones distribuidas. Describiremos en profundidad cada uno de estos tipos.

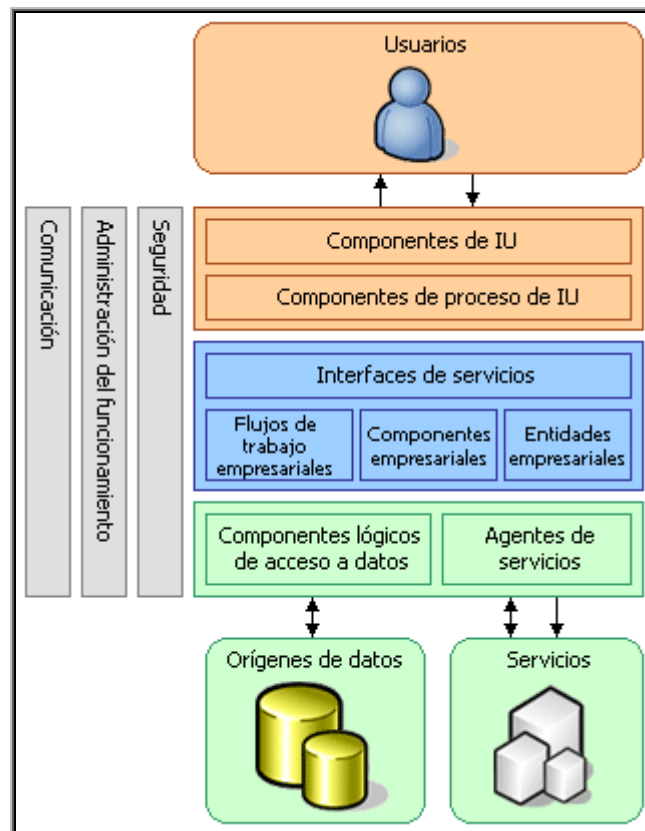


Figura 2. Tipos de componentes utilizados en el escenario comercial de ejemplo

Los tipos de componentes identificados en el escenario de diseño de ejemplo son:

1. **Componentes de interfaz de usuario (IU).** La mayor parte de las soluciones necesitan ofrecer al usuario un modo de interactuar con la aplicación. En el ejemplo de aplicación comercial, un sitio Web permite al cliente ver productos y realizar pedidos, y una aplicación basada en el entorno operativo Microsoft Windows® permite a los representantes de ventas escribir los datos de los pedidos de los clientes que han telefonado a la empresa. Las interfaces de usuario se implementan utilizando formularios de Windows Forms, páginas Microsoft ASP.NET, controles u otro tipo de tecnología que permita procesar y dar formato a los datos de los usuarios, así como adquirir y validar los datos entrantes procedentes de éstos.
2. . De este modo, el flujo del proceso y la lógica de administración de estado no se incluye en el código de los elementos de la interfaz de usuario, por lo que varias interfaces podrán utilizar el mismo "motor" de interacción básica.
3. **Flujos de trabajo empresariales.** Una vez que el proceso de usuario ha recopilado los datos necesarios, éstos se pueden utilizar para realizar un proceso empresarial. Por ejemplo, tras enviar los detalles del producto, el pago y el envío a la aplicación comercial, puede comenzar el proceso de cobro del pago y preparación del envío. Gran parte de los procesos empresariales conllevan la realización de varios pasos, los cuales se deben organizar y llevar a cabo en un orden determinado. Por ejemplo, el sistema empresarial necesita calcular el valor total del pedido, validar la información de la tarjeta de crédito, procesar el pago de la misma y preparar el envío del producto. El tiempo que este proceso puede tardar en completarse es indeterminado, por lo que sería preciso administrar las tareas necesarias, así como los datos requeridos para llevarlas a cabo. Los flujos de trabajo empresariales definen y coordinan los procesos empresariales de varios pasos de ejecución larga y se pueden implementar utilizando herramientas de administración de procesos empresariales, como BizTalk Server Orchestration.

4. **Componentes empresariales.** Independientemente de si el proceso empresarial consta de un único paso o de un flujo de trabajo organizado, la aplicación requerirá probablemente el uso de componentes que implementen reglas empresariales y realicen tareas empresariales. Por ejemplo, en la aplicación comercial, deberá implementar una funcionalidad que calcule el precio total del pedido y agregue el costo adicional correspondiente por el envío del mismo. Los componentes empresariales implementan la lógica empresarial de la aplicación.
5. **Agentes de servicios.** Cuando un componente empresarial requiere el uso de la funcionalidad proporcionada por un servicio externo, tal vez sea necesario hacer uso de código para administrar la semántica de la comunicación con dicho servicio. Por ejemplo, los componentes empresariales de la aplicación comercial descrita anteriormente podría utilizar un agente de servicios para administrar la comunicación con el servicio de autorización de tarjetas de crédito y utilizar un segundo agente de servicios para controlar las conversaciones con el servicio de mensajería. Los agentes de servicios permiten aislar las idiosincrasias de las llamadas a varios servicios desde la aplicación y pueden proporcionar servicios adicionales, como la asignación básica del formato de los datos que expone el servicio al formato que requiere la aplicación.
6. **Interfaces de servicios.** Para exponer lógica empresarial como un servicio, es necesario crear interfaces de servicios que admitan los contratos de comunicación (comunicación basada en mensajes, formatos, protocolos, seguridad y excepciones, entre otros) que requieren los clientes. Por ejemplo, el servicio de autorización de tarjetas de crédito debe exponer una interfaz de servicios que describa la funcionalidad que ofrece el servicio, así como la semántica de comunicación requerida para llamar al mismo. Las interfaces de servicios también se denominan *fachadas empresariales*.
7. **Componentes lógicos de acceso a datos.** La mayoría de las aplicaciones y servicios necesitan obtener acceso a un almacén de datos en un momento determinado del proceso empresarial. Por ejemplo, la aplicación empresarial necesita recuperar los datos de los productos de una base de datos para mostrar al usuario los detalles de los mismos, así como insertar dicha información en la base de datos cuando un usuario realiza un pedido. Por tanto, es razonable abstraer la lógica necesaria para obtener acceso a los datos en un capa independiente de componentes lógicos de acceso a datos, ya que de este modo se centraliza la funcionalidad de acceso a datos y se facilita la configuración y el mantenimiento de la misma.
8. **Componentes de entidad empresarial.** La mayoría de las aplicaciones requieren el paso de datos entre distintos componentes. Por ejemplo, en la aplicación comercial es necesario pasar una lista de productos de los componentes lógicos de acceso a datos a los componentes de la interfaz de usuario para que éste pueda visualizar dicha lista. Los datos se utilizan para representar entidades empresariales del mundo real, como productos o pedidos.
9. **Componentes de seguridad, administración operativa y comunicación.** La aplicación probablemente utilice también componentes para realizar la administración de excepciones, autorizar a los usuarios a que realicen tareas determinadas y comunicarse con otros servicios y aplicaciones.

Visión conceptual de la arquitectura del Sistema transaccional

Antes de comenzar con las definiciones técnicas de la arquitectura propuesta, es necesario entender cuales son los conceptos básicos planteados en dicha arquitectura. A efecto de comprender los bloques conceptuales que definen la *Arquitectura Tecnológica*, a continuación se visualiza un gráfico exponiendo los diferentes componentes que integran la Visión Conceptual de la Arquitectura Simple.

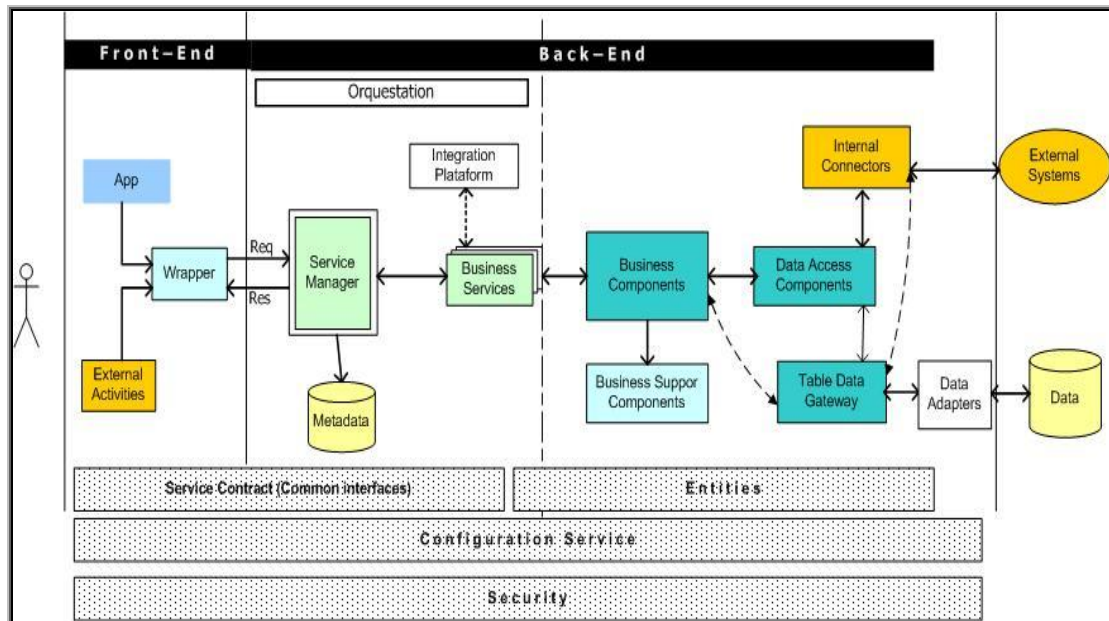


Figura 3. Tipos de componentes utilizados en el escenario comercial de ejemplo

A efectos de cumplimentar esta premisa básica, hemos decidido diseñar un sistema basado en capas (Layers). Bajo este patrón, denominado Múltiples Capas (Multilayer), se definen dos capas bien diferenciadas:

Front End System

Back End System.

A continuación iremos describiendo cada uno de los componentes visualizados en el gráfico anterior. Los componentes se describirán comenzando de izquierda a derecha, en función de las capa físicas que conforman la arquitectura.

FRONT-END O CAPA DE PRESENTACIÓN

La capa de presentación es la encargada de brindar al usuario un entorno de trabajo, en el cual se listan sus actividades y tareas. Por otro lado, es quien interactúa con la capa de Middleware para ejecutar servicios de consulta o transaccionales.

Es la capa que contiene las siguientes funcionalidades de los aplicativos:

Obtener datos del Sistema de Información, a través de servicios de consulta.

Actualizar el Sistema de Información a través de servicios transaccionales

Presentación de datos al usuario a través de formularios, utilizando diferentes tipos de tecnologías.

Piezas de software que permiten la interacción entre los formularios y los servicios de negocio.

Actividades propias del sistema

Las actividades representan a una o mas acciones que interactúan con usuarios del sistema de información. Un usuario del sistema de información puede estar representado por una persona física, una aplicación, un proceso, una máquina de estado, etc.

Las actividades forman parte del sistema de información y comúnmente corresponden a una actividad de un proceso de negocio.

Las actividades pueden ser ejecutadas desde diferentes tecnologías / dispositivos: Web Browsers, Celulares, Web Services, Etc..

Actividades Externas

Las actividades externas son aquellas que no forman parte directa del sistema de información, sino que son extensiones al mismo. Un ejemplo de una actividad externa podría ser un requerimiento de datos, desde el aplicativo de un proveedor.

Lo importante a destacar en este punto es que las actividades externas acceden al sistema de información de la misma forma que las actividades de los procesos de negocio del sistema de información.

Wrapper de aplicación

El Wrapper es el único que puede interactuar con la capa de servicios de negocio para resolver requerimientos de negocio. Dicha interacción se realiza a través de un adaptador de un canal, enviando un Request y esperando la respuesta en un Response.

Los adaptadores de canal son piezas de código que permiten desacoplar el envío de un requerimiento a un servicio de negocio, con su correspondiente respuesta, de la tecnología de transporte necesaria para acceder a dicho servicio de negocio (de consulta o transaccional).

El sistema de adaptadores de canal es extensible, de esta forma se podrían ampliar los tipos de canales de acceso a los servicios. A través de ellos se podrían ejecutar servicios desde Web Services, Dispositivos Real-Time, Páginas WEB, Aplicaciones Winform, Etc.

BACKEND

El Back-End es la capa física que cumple tres funciones básicas:

Brindar soporte al sistema transaccional (OLTP).
Brindar soporte a los servicios de negocio.
Brindar soporte a las problemáticas de integración.

Por lo tanto expone servicios a los aplicativos del Front-End. Es la intermediaria entre los requerimientos de los aplicativos y los componentes del Back-End.

El Back-End se encuentra conformado por las siguientes subsistemas y capas lógicas:

Dispatcher (SOA – Service Oriented Applications)
Business Services
Servicios sincrónicos / asincrónicos
Servicios transaccionales
Servicios de consulta
Componentes de negocio y de soporte al negocio (BC)
Componentes de acceso a datos (DAC)
Plataforma de integración

Dispatcher de servicios

Es el encargado de recibir los requerimientos realizados por los diferentes clientes (*Request*), ubicar el servicio requerido dentro del catálogo de acciones, ejecutarlo bajo un contexto controlado y condicionado por la configuración establecida en el catálogo, y retornar al cliente el resultado de esta ejecución (a través de un objeto de tipo *Response*). A través del *Dispatcher* se exponen todas los servicios de negocio (de consulta y transaccionales) del Sistema.

El Dispatcher es el único punto de entrada a los servicios de negocio (transaccionales o de consulta) del Sistema.

El Dispatcher brinda un contexto de ejecución para los servicios y provee funcionalidades como:

- 1- Entorno transaccional
- 2- Control de ejecución de servicios
- 3- Sistema de mensajes y configuración
- 4- Manejo de excepciones
- 5- Sistema de conectores para acceder a plataformas no integradas nativamente al sistema de información. (Ej. SAP, BizTalk, Otros Servicios nuevos o preexistentes del cliente)

Business Services

Los Business Services representan a cualquier tipo de servicio, administrados por el Dispatcher ya sean transaccionales o de consulta, sincronos o asíncronos. Los servicios se pueden clasificar de la siguiente manera:

Servicios sincrónicos: se deben utilizar, en los casos en los cuales exista interacción con el sistema de base de datos del Sistema o con sistema transaccionales cuyos de tiempos de respuestas sean óptimos.

Servicios asíncrónicos: se deben utilizar en los casos en los cuales se requiera interactuar con sistemas no transaccionales, o en los cuales los tiempos de procesamiento no son óptimos o todo el procesamiento de un determinado paso de la transacción se realiza en sistemas externos. Un ejemplo que muestra la necesidad de utilizar servicios asíncrónicos podría ser una transacción que interactúa con un mainframe bajo un patrón de diseño de Request / Response es decir el servicio envía un mensaje al mainframe y queda a la espera de una respuesta.

Servicios transaccionales: son aquellos que actualizan las bases de datos del sistema. La interacción con dichos subsistemas se realizará a través de los Componentes de Negocio o Componentes de Soporte a Servicios.

Servicios de consulta: son aquellos que solo obtienen datos del sistema de información. En este caso, el acceso al sistema de información también es realizado por los mismos mecanismos.

Plataforma de integración

La plataforma de integración se compone por un conjunto de componentes y servicios que permiten la interacción entre el sistema de información y sistemas externos .

La plataforma de integración puede recibir eventos provenientes del Dispatcher y es responsable de entregarlos en forma confiable a los sistemas externos. Por otro lado, tiene la capacidad de recibir eventos provenientes de sistemas externos que deban impactar en una base de datos del sistema o ejecutar un servicio transaccional o de consulta.

A continuación se describen algunos de los servicios provistos por la plataforma de integración:

- Envío confiable de mensajes
- Firma / Validación de certificados digitales

- Encriptación / Desencriptación de mensajes
- Auditoria de los mensajes intercambiados
- Validación del formato de los mensajes
- Transformación de un tipo de mensaje entrante a un tipo de mensaje saliente
- Sistema de adaptadores a sistemas externos extensibles, a través de la exposición de APIs.
- Herramientas para la administración de la plataforma y el diseño de las interfases.

Nota: En este documento es todo lo que se detallara sobre plataforma de integración ya que es un amplio tema que escapa a los fines prácticos del objetivo principal de una propuesta de arquitectura simple. Además se explicaran solo a nivel conceptual los servicios asíncronos y los conectores que utiliza la plataforma de integración, la implementación técnica de los mismos serán detallados en un documento anexo a la propuesta de arquitectura simple.

Business Components (BC)

Las BC (Business components) contiene la lógica del negocio de la Solución. A través de los componentes de negocio se realizan validaciones, se obtienen datos del sistema de persistencia, se actualiza el sistema de información, etc.

Los componentes de negocio no acceden a la base de datos en forma directa, lo hacen a través de los componentes de acceso a datos (DAC).

Data Access Components (DAC)

DALC es el acrónimo de Data Access Logic Components (componentes de lógica de acceso a datos). Dichos componentes son los que tienen el conocimiento de cómo interactuar con el sistema de persistencia (Ej. Base de Datos SQL Server).

Data Adapters

Los adaptadores para el acceso a datos son piezas de software que permiten interactuar a los componentes de acceso a datos (DAC) con la Base de Datos. ADO.NET es un ejemplo de un adaptador para el acceso a datos.

Database

La base de datos es el repositorio de datos para el Sistema de Información. Es una base de datos relacional que contiene el modelo de entidad-relación correspondiente a dicho sistema de información.

External Systems

Los sistemas externos son aquellos que no constituyen un componente nativo del sistema de información pero que son requeridos para la operación diaria del negocio.

La interacción con los sistemas externos puede ser Punto a punto asíncrona sin espera de respuesta, Punto a punto asíncrona con espera de respuesta, Punto a punto sincrónica.

No se debe confundir una interfase asíncrona con un proceso batch. Por ejemplo, una interfase de tipo *Punto a punto asíncrona sin espera de respuesta* podría agregar registros a un archivo de texto plano que será procesado a una determinada hora del día.

Especificaciones técnicas

En esta sección se especifican con mayor nivel de detalle los aspectos técnicos de los componentes definidos en el modelo anterior.

Como se puede visualizar en el documento el único punto de entrada a los servicios de negocio es el Dispatcher. Así como también se puede apreciar en el diagrama que el único punto de acceso al sistema de persistencia del Sistema son los Componentes de Acceso a Datos (DAC).

El resto de las clases del diagrama intentan representar de forma genérica como serán implementados los diferentes escenarios de uso definidos para la capa de Servicio & Dominio.

Nota: Todos los nombres de componentes, clases y servicios, así como también los Namespaces visualizados en los ejemplos de código, son ficticios. El sistema de nomenclatura de Namespaces formará parte de este documento, las normas con respecto a componentes, clases, servicios se definirán en un documento separado.

Documento	
Entidades	Arquitectura Tecnológica Entidades V2.0.doc
Servicios	Arquitectura Tecnológica Servicios SOA V.2.0.doc
Componentes BackEnd (BC y DAC)	Arquitectura Tecnológica BC-DAC V2.0.doc
FrontEnd-Clientes	Arquitectura Tecnológica FrontEnd V.2.0.doc

Tabla de Contenidos

<i>Alcance</i>	<i>1</i>
<i>Elaboración / Revisión / Aprobación</i>	<i>1</i>
<i>Roles y Responsabilidades</i>	<i>1</i>
<i>Introducción</i>	<i>1</i>
<i>Propósito</i>	<i>2</i>
<i>Audiencia</i>	<i>2</i>
<i>Estructura del documento</i>	<i>2</i>
<i>Definiciones, acrónimos y abreviaturas</i>	<i>3</i>
<i>Arquitectura para aplicaciones corporativas</i>	<i>3</i>
<i>Diseño conceptual del modelo MVC</i>	<i>4</i>

<i>Modelo</i>	<i>4</i>
<i>Tipos de componentes</i>	<i>5</i>
<i>Visión conceptual de la arquitectura del Sistema transaccional.....</i>	<i>7</i>
Front-End o capa de Presentación	8
<i>Actividades propias del sistema.....</i>	<i>8</i>
<i>Actividades Externas</i>	<i>9</i>
<i>Wrapper de aplicación.....</i>	<i>9</i>
Backend.....	9
<i>Dispatcher de servicios</i>	<i>9</i>
<i>Business Services</i>	<i>10</i>
<i>Plataforma de integración.....</i>	<i>10</i>
<i>Business components (BC).....</i>	<i>11</i>
<i>Connectores</i>	<i>¡Error! Marcador no definido.</i>
<i>Data Access Logic Components (DALC) o DAC.....</i>	<i>11</i>
<i>Data Adapters.....</i>	<i>11</i>
<i>Database</i>	<i>11</i>
<i>External Systems.....</i>	<i>11</i>
<i>Especificaciones técnicas.....</i>	<i>12</i>
Cientes.....	¡Error! Marcador no definido.
<i>Requests y Responses</i>	<i>¡Error! Marcador no definido.</i>
<i>Context Information.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Wrapper</i>	<i>¡Error! Marcador no definido.</i>
<i>Ejemplo de implementacion</i>	<i>¡Error! Marcador no definido.</i>
<i>Construcción de las interfases o contratos del servicio.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Interfaz de respuesta</i>	<i>¡Error! Marcador no definido.</i>
<i>Tipo genérico funcional recibido por un Request o Response (IEntity)</i>	<i>¡Error! Marcador no definido.</i>
<i>Utilización de las interfaces del lado del Cliente:.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Nomenclaturas.....</i>	<i>¡Error! Marcador no definido.</i>
Servicios	¡Error! Marcador no definido.
<i>Servicios Sincrónicos</i>	<i>¡Error! Marcador no definido.</i>
<i>Servicios de consulta (SC).....</i>	<i>¡Error! Marcador no definido.</i>
<i>Conclusiones:.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Servicios transaccionales.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Conclusiones:.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Servicios Asíncronos</i>	<i>¡Error! Marcador no definido.</i>
<i>Conclusiones:.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Conectors</i>	<i>¡Error! Marcador no definido.</i>
<i>Conclusiones:.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Componentes de Negocio (BC).....</i>	<i>¡Error! Marcador no definido.</i>
<i>Conclusiones:.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Componentes de Soporte al Negocio (BSC).....</i>	<i>¡Error! Marcador no definido.</i>
<i>Conclusiones:.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Componentes de Acceso a Datos (DAC) y Table Data Gateay (TDG)</i>	<i>¡Error! Marcador no definido.</i>
<i>Conclusiones:.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Entidades</i>	<i>¡Error! Marcador no definido.</i>
<i>Entidades Simples</i>	<i>¡Error! Marcador no definido.</i>
<i>Colecciones.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Serializacion</i>	<i>¡Error! Marcador no definido.</i>
<i>Entidades de relación (ER).....</i>	<i>¡Error! Marcador no definido.</i>
<i>Clases Base Entity/Entities</i>	<i>¡Error! Marcador no definido.</i>
<i>Conclusiones acerca de las entidades simples:.....</i>	<i>¡Error! Marcador no definido.</i>
<i>Conclusiones acerca de las entidades tipo colecciones:</i>	<i>¡Error! Marcador no definido.</i>
<i>Sistema de espacios de nombres (Namespaces)</i>	<i>¡Error! Marcador no definido.</i>

<i>Tabla de Contenidos</i>	<i>12</i>
---	------------------