
Trabajo Final de Grado

Arquitectura
Cliente-Servidor/Cliente-Servidor
en Aplicaciones de Misión Crítica

Autor: Cristián Bircher
Tutor: Ing. Natalia Mira



Índice General

Índice.....	2
Dedicatoria.....	6
Agradecimientos	7
Resumen del Proyecto	9
Glosario.....	10
Objetivos	11
Objetivo General:	11
Objetivos específicos:	11
Alcances y Pertinencia	11

Capítulo 1: Introducción

Contexto.....	13
Descripción	13
Justificación del Trabajo	15
Marco Teórico	15
Metodología.....	16
Marco Metodológico:	16
Metodología Propuesta:	17

Capítulo 2: Antecedentes

El Problema Inicial	20
La idea original	21
Breve descripción de la solución	22

Capítulo 3: Arquitectura Empresarial

Arquitectura Empresarial.....	25
Marco Referencial / Framework.....	26



EABOK – EA Body of Knowledge	26
Framework de Zachman	28
The Open Group Architectural Framework (TOGAF)	30
The Federal Enterprise Architecture Framework (FEAF)	32
The Gartner Methodology	35
Conclusiones	37
Resumen	39

Capítulo 4: Arquitectura de TI

Arquitectura de TI.....	40
Aplicaciones de Misión Crítica	46
Conclusiones	47

Capítulo 5: ISO/IEEE 42010

IEEE 42010	48
Fundamentos	48
Modelo Conceptual de la Descripción de Arquitecturas.....	49
Modelo Conceptual del Framework para la Arquitectura	51
Lenguaje para la Descripción de Arquitecturas – ADL	51
Resumen	52

Capítulo 6: Arquitectura Propuesta

Introducción	53
Modelo de Capas	55
Modos de Operación	56
Perspectiva	58
Componentes, Servicios y Estados.....	58
Modelo de N Capas	60
Direccionalidad de la Información.....	61



Primitivas, Sentencias, Relaciones y Procesos.....	62
Arquitectura Cliente-Servidor / Cliente-Servidor.....	64
Resumen	66

Capítulo 7: Objetos Primitivos

Introducción	67
Modelos Primitivos.....	68
Primitivas	69
Conclusiones.....	71

Capítulo 8: Componentes y Servicios

Componentes y Servicios Primitivos	72
Identificación de Componentes.....	73
Estrategias.....	75
Conclusiones.....	76

Capítulo 9: Arquitectura CS/CS

Introducción	77
Principios	77
Evolución del Modelo de Referencia.....	78
Vistas de la Arquitectura Cliente-Servidor / Cliente-Servidor	80
Vistas de la Arquitectura de TI:	80
Modelo de dos Capas.....	80
Modelo de N Capas	81
Modelo de 2 Capas con Distribución de Nodos.....	81
Vistas de la Arquitectura de Aplicación:.....	82
Modelo de Distribución de Componentes y Servicios	82
Modelo de Distribución de Componentes y Servicios de Alto Nivel	82
Modelo de Operación en Modo Asíncrono	83



Modelo de Flujo de Datos.....	84
Relaciones entre componentes	84
Primitivas y Conectores	86
Identificación de Componentes Primitivos	87
Identificación de Relaciones entre Componentes	90
Construcción de Relaciones	91
Resumen	94

Capítulo 10: Aplicación Práctica

Breve Reseña – Ingenio Azucarero.....	95
Objetivos del Trabajo realizado.....	98
Alcances	98
Justificación	98
Problemas.....	98
Alternativas	99
Plan de Actualización Tecnológica:	99
Actualización de Hardware:.....	100
Alta disponibilidad	100
Arquitectura Cliente-Servidor / Cliente-Servidor vs. Otras Arquitecturas	101
Solución Propuesta.....	102
Otras herramientas de mercado	103
Mimix.....	104
Oracle Golden Gate	105
Conclusiones	106

Capítulo 11: Conclusiones

Bibliografía y Material de Soporte.....	109
---	-----



Dedicatoria

Para Paulita y Lucas, las zanahorias que hacen caminar al burro...

Para la señorita Berta, por permitirme ser parte de sus días...

Para mis Viejos, Hermanas y Suegros por el apoyo incondicional de siempre...

Para Doña Dominga, Don Titi, nuestro bebé y Fernando, por estar mirando desde alguna estrella...



Agradecimientos

Ha pasado mucha agua bajo el puente y seguramente como en muchos otros aspectos de la vida, no soy el mismo que arrancó con esta carrera, apenas unos años atrás. En el medio me encontré con un montón de gente dispuesta a ayudarme sin interés alguno, a colaborar con su opinión, a aportar algún mate, o simplemente a no molestar cuando uno necesitaba concentrarse.

Defino a ese tiempo como una constante lucha contra mis limitaciones: de tiempo, de voluntad, obligaciones como Padre y Esposo, laborales... y principalmente mis años de edad, que no han venido solos.

Este humilde triunfo que espero como coronación de mi esfuerzo, no fue solo mío. Me encuentra afianzado laboral y profesionalmente, pero por sobre todas las cosas, me encuentra en medio de una familia de hierro. A mi esposa e hijos, no solamente les debo haberles robado valioso tiempo que podría haberles dedicado, sino también que hayan comprendido y aceptado los tiempos en los cuales no podía hacerlo. Ellos soportaron el gran peso de mis días de insomnio, mi acotada agenda y mis mil y una idas y vueltas, tratando de atajar todo lo que la vorágine de la vida nos fue metiendo en el camino.

Mi eterno agradecimiento a Doña Clemen y Don Raúl, quienes entre tantas, pero tantas enseñanzas, me dejaron marcado en la frente la importancia de ir para adelante, sin que merezca tanta atención que tanto se avanza, sino la forma en la cual se lo hace. Ellos son mi ejemplo de amor leal, incondicional y eterno, y forjadores de una familia perfecta, que no podría ser tal sin la presencia de mis hermanas, por quienes sufrí estoicamente durante tantos años las cargadas de mis amigos, tratando de que no se conviertan en cuñados.

A mis amigos del barrio, humilde origen al que vuelvo cada tanto, amigos del club por donde descargo mis pasiones fin de semana a fin de semana y amigos del colegio, compañeros incondicionales de noches interminables de filosofía de esquina, tardes de tablón o redonda, o noches de rock y blues. Cultores de la amistad incondicional y prestamistas de orejas para escuchar problemas ajenos, relatores de anécdotas inverosímiles, que rayan la ciencia ficción; pero siempre con el humor como bandera.

A mis amigos y compañeros de trabajo, quienes no solo comparten conmigo horas y horas de renegar con las responsabilidades diarias, sino que me permiten crecer día a día como persona y como profesional, con el humor y la seriedad de siempre, aunque parezca contradictorio. Esto incluye también a mis Jefes, actuales y pasados, porque no solamente respetaron mis tiempos de estudio sino que también me permitieron su amistad y abrieron sus conocimientos y experiencias, desde épocas en las que éramos todos soldados (yo lo sigo siendo).

A mis amigos de Tucumán y a los que hice en el camino (aunque allá haga tanto frío...), por permitirme no solamente su amistad y los cargamentos de empanadas tucumanas, sino también la posibilidad de hundir los garfios en sus problemas y necesidades y aportar mi grano de arena en sus éxitos laborales.

A los que ya no están y recuerdo con tanto cariño, protagonistas de tantas anécdotas maravillosas, compañeros de tantas andanzas. Todavía rezo por ellos y cada tanto, algunas historias de sus ocurrencias siguen alegrando alguna que otra reunión. Especialmente a Don Titi, por tantas tardes en el 15 de Abril, para alentar primero y alentarme después durante tantos, pero tantos años. A todos ellos los tengo presentes en cada oración, cada mañana.



A Dios, a quien solo pido que me dé la sabiduría necesaria para tomar las decisiones correctas, por permitirme la conciencia de salir a pelearla, darme la oportunidad trabajar en la empresa para la cual trabajo, y regalarme la maravillosa familia de la cual formo parte. Sin dudas, no puedo más que agradecer y esforzarme para ser digno de merecerlo, en una lucha constante en la que no puedo dar el brazo a torcer.

Nada de lo poco que logré hubiera sido posible si no hubiera contado con un poco de ayuda de mis amigos.



Resumen del Proyecto

Dentro del complejo mundo de las organizaciones empresariales existen innumerables oportunidades para reutilizar o adaptar soluciones existentes a nuevas necesidades del negocio. Incluso para algunos casos, las características de los procesos son tan particulares que invariablemente precisan la construcción de soluciones innovadoras, y la Arquitectura objetivo de este trabajo es uno de esos casos. Sin embargo, hay ciertas cuestiones del contexto que necesitan ser contempladas, ya que seguramente las nuevas necesidades se mueven en un ambiente dinámico, plagado de cambios, y donde seguramente ya existen ciertos elementos a los que se puede echar mano para reutilizarlos o adaptarlos e integrar los nuevos elementos a los existentes, adaptados o no. De esta forma es posible construir soluciones robustas y flexibles, reutilizando componentes y con el debido cuidado, mediante una correcta gestión de cambios encajarlo dentro de la maraña de procesos que configuran la operación de la organización y que contribuyen a la consecución de sus objetivos.

Pero estas actividades no son actividades que se realizan en forma aislada del resto de la organización. Debe existir cierto alineamiento que debe bajar de los niveles superiores de la misma, en una especie de cascada que permite correlacionarlas. El diseño de los procesos es un proceso en sí mismo, que requiere una visión que llegue un poco más allá de lo que específicamente se esté desarrollando. Es por ello que necesitan ser altamente integrables, reutilizables y aplicables a cualquier nueva necesidad. Estos procesos son los que permitirán que la organización opere sus negocios en base a sus objetivos y estrategia.

La Arquitectura Empresarial es una disciplina base para lograr el alineamiento del cual se hablaba en el párrafo anterior, y por eso merece un capítulo especial. No solamente se pueden encontrar distintos sabores en el mercado sino que también se pueden combinar distintos elementos de las distintas metodologías para el objetivo específico que se persiga.

La Arquitectura de TI como disciplina no podría alinear sus actividades a los fines de la organización si no estuviera contemplada bajo el paraguas de la arquitectura empresarial. Sin embargo, y a pesar de estar explícitamente definidos los roles de los arquitectos empresariales y arquitectos de TI, en la práctica el diseño de la arquitectura que se trate, es llevada a cabo por equipos interdisciplinarios, cuyos miembros se especializan en distintos campos de acción. Tanto las tareas relacionadas con la Arquitectura Empresarial como las tareas de Arquitectura de TI, sea que se trate de aplicaciones, negocios, información, tecnología o una mezcla de ellas, son tareas que si bien en muchas ocasiones no están documentadas con la formalización que la magnitud e importancia de las organizaciones requieren, se realizan constantemente, a veces en forma inconsciente.

Las tareas básicas de un arquitecto del área que se trate, consisten en el diseño, desarrollo, cambio y administración de elementos primitivos. El trabajo en materia arquitectónica dentro del contexto de una organización es una tarea interdisciplinaria que no debe perder de vista aspectos técnicos, operativos, económicos y de infraestructura existente, y que requieren competencias, imaginación y visión estratégica para lograr soluciones innovadoras que cumplan con los objetivos para los cuales se realiza el esfuerzo y satisfagan las expectativas de los clientes internos.

Este trabajo pretende demostrar como una Arquitectura Cliente-Servidor / Cliente-Servidor puede constituirse en la mejor opción técnica con una relación costo / beneficio alta y cualidades operativas óptimas que aseguren la disponibilidad de las Aplicaciones de Misión Crítica que permitan la operación continua del negocio.



Glosario

- **Holística:** sistema de pensamiento por el cual las características de una entidad o de un conjunto solo pueden ser conocidas cuando se las considera y aprehende en conjunto, en su totalidad, y no cuando se estudia cada parte del mismo por separado.
- **Taxonomía:** ciencia que estudia los principios, métodos y fines de la clasificación. Aplicada a la disciplina de la Arquitectura, metodología para la organización y categorización de artefactos arquitectónicos.
- **Tautología:** fórmula bien formada de un sistema de lógica proposicional que resulta verdadera para cualquier interpretación; es decir, para cualquier asignación de valores de verdad que se haga a sus fórmulas atómica.
- **Ontología:** aplicado a distintas ramas de la ciencia la ontología analiza los diferentes elementos fundamentales que forman o componen un Universo.
- **Arquitectura:** estructura organizacional de un sistema que identifica sus componentes, interfaces, relaciones y comportamiento. Organización fundamental de un sistema incorporado en sus componentes, sus relaciones entre sí y con el medio ambiente, y los principios que guían su diseño y evolución.
- **Arquitectura de Software:** Estructura de componentes de un sistema y sus interrelaciones, principios y guías para su gobierno, diseño y evolución. Estructura de las Estructuras de un sistema.
- **Componentes de Software:** elementos que forman la Arquitectura de Software y desenvuelven la funcionalidad para la cual fueron creados.
- **Arquitectura de Sistemas:** Estructura que consiste en módulos activos, mecanismos que permiten la interacción entre ellos y un conjunto de reglas que las gobiernan y determinan su comportamiento.
- **Arquitecto:** responsable por el diseño, construcción, administración y cambios de la Arquitectura.
- **Descripción de la Arquitectura:** Colección de artefactos productos del trabajo del arquitecto a los fines de representarla, comunicarla y lograr entendimiento entre los stakeholders.
- **Modelo de Referencia / Framework:** Estructura o esqueleto que define artefactos sugeridos y que describe como esos artefactos se relacionan o deberían relacionarse para proveer definiciones genéricas aplicables a distintas situaciones o necesidades.
- **Arquitectura Empresarial:** Arquitectura en la cual el sistema en cuestión es la empresa, y especialmente sus procesos de negocios, tecnologías, y sistemas de información.
- **Arquitectura de Aplicación:** Arquitectura de una aplicación o conjunto de aplicaciones específicas.
- **Arquitectura de la Información:** Arquitectura que trata acerca de las metodologías de almacenamiento de la información y su tratamiento específico por fuera del uso que le dan las aplicaciones.



Objetivos

Objetivo General:

- Desarrollar un modelo de arquitectura de referencia aplicable a soluciones de software de misión crítica.

Objetivos específicos:

- Estructurar un marco teórico de referencia para soportar bajo esta arquitectura aplicaciones de misión crítica en general;
- Profundizar el análisis de forma metodológica y, enfocando en componentes, obtener un modelo arquitectónico robusto que permita operar en forma continua los procesos de negocios críticos;
- Proporcionar fundamentos sólidos para el diseño e implementación de funcionalidades que cubran los procesos de negocio que se consideren críticos;
- Identificar componentes, objetos y funciones genéricas de la arquitectura propuesta.

Alcances y Pertinencia

Como continuidad del trabajo realizado anteriormente y a la luz de la experiencia obtenida en la realización del mismo, en principio el trabajo alcanzará a:

- Procesos de negocios de misión crítica en ambientes de TI centralizados;
- Combinación de componentes tecnológicos existentes que permitan moldear la arquitectura de acuerdo a los objetivos planteados.

En muchas situaciones y bajo distintas circunstancias se dan ciertas desconexiones entre el diseño empresarial y la arquitectura de TI que la soporta, entendiéndose a la arquitectura de TI como un todo, y al diseño empresarial como el conjunto de procesos mediante el cual la empresa establece relaciones con el mundo exterior. Esto hace que en la medida que se descienda en dicho diseño, se encuentren características, procesos y particularidades que no son tenidas en cuenta. Los procesos de misión crítica para una filial remota pueden ser no tan críticos para la operación de la empresa en su conjunto, pero conllevan implícitos costos de no operación que son necesarios identificar y mitigar.

Justifica la realización del trabajo la falta de material de referencia sobre arquitecturas que soporten este tipo de soluciones específicas. Adicionalmente, el resultado del desarrollo propone no solamente la estructuración de un modelo orientado a problemáticas similares, sino también a formalizar o consolidar un marco teórico sobre el cual se construyó la solución para el Ingenio azucarero, el cual podría ser altamente factible de aplicar a otros tipos de soluciones.



Con este proyecto se obtendrá un modelo de arquitectura mediante el cual mediante la combinación de los distintos componentes y elementos que se identificarán, permitirá construir sobre su base, soluciones de misión crítica robusta, flexibles, simples y adaptables a distintos entornos, permitiendo con su uso:

- Reconocer en la arquitectura actual los distintos elementos que pueden ser reutilizados, con su funciones actuales o funcionalidades que no se están explotando;
- Desarrollo de capacidades y estandarización del diseño para soluciones similares;
- Solución a problemas específicos de operación regional o de negocio en particular, que permitirán la reducción de costos de no operación;



Capítulo

1

Introducción

Este capítulo presenta el contexto actual y las motivaciones que desembocaron en un trabajo como este. Lo describe y justifica dando un breve repaso al marco teórico sobre el cual se apoya, posicionando al lector respecto de la obra, la metodología y los procedimientos utilizados en su desarrollo.

Contexto

Los cambios tecnológicos desafían a las compañías a mantener permanentemente cierto grado de flexibilidad en su arquitectura de TI, que les permita adaptarse a un entorno muchas veces extremadamente dinámico y con una larga lista de objetivos.

Esto incluye a la disponibilidad de las aplicaciones y por ende, a la disponibilidad de la infraestructura y sus diversos componentes, tanto de software como de hardware, y a la forma en la cual los mismos se disponen, conectan y utilizan.

Muchas veces, las estrategias de TI no contemplan ciertas particularidades de los negocios, necesidades regionales o simplemente costumbres, como por ejemplo los horarios de trabajo y descanso, que suelen variar en función de la distribución geográfica del negocio.

En la actualidad, nos encontramos día a día con nuevas propuestas tecnológicas que permiten mantener esquemas o arquitecturas centralizadas desde donde los servicios de TI se pueden brindar, pudiendo estar los clientes diseminados por cualquier lugar del planeta. Esto no solo permite optimizar recursos de procesamiento, sino mantener en forma centralizada la información, minimizando los recursos afectados a las mismas, y por ende, el costo económico de su operación, mantenimiento y eventualmente necesidades de escalabilidad, tanto de software como de hardware, y facilitando el acceso a la información prácticamente desde cualquier lugar.

El presente trabajo propone ahondar el análisis sobre un tipo de arquitectura bastante particular, enfocado en la problemática de las aplicaciones de misión crítica, necesarias para mantener la operación del negocio, aún bajo condiciones de no disponibilidad de los sistemas centrales.

Descripción

Desde la aparición misma de las primeras computadoras, su uso se fue adaptando a las distintas necesidades: de cálculo primero, de uso de la información después y obviamente, infiltrándose de a poco en diferentes aspectos de nuestras vidas cotidianas.



En muchos casos fue al revés: algunas necesidades se fueron adaptando a su uso, y aunque parezca un juego de palabras, aún hoy en día se sigue tras esas tendencias, y en muchos casos dejando a un costado cuestiones relacionadas con los procesos a los que sirven, sean estos de negocio o no.

Muchas compañías van apenas detrás de la última novedad tecnológica, casi sin invertir tiempo en la investigación, análisis, pruebas de concepto, integración con otras herramientas e inserción dentro del contexto de TI, tanto interno como externo.

En tal sentido, es común ver diariamente en muchas organizaciones un ajuste de las necesidades a lo que las nuevas tecnologías ofrecen. En el mejor de los casos y con algo de orden se logra eficiencia en el o los procesos. En muchos otros, aumentan las tareas, el trabajo se hace engorroso o burocrático, lento y traumático. En el peor, los procesos se interrumpen, generando demoras, acumulación de trabajo pendiente y, lo que es el terror de cualquier administrador, altos costos de no operación.

En el medio han ido quedando un largo listado de novedades, que han arrastrado a las organizaciones, obligándolas a subirse a las mismas o correr el riesgo de vivir en la oscuridad de la obsolescencia tecnológica. Otras han ido mutando lentamente a través de los años, para finalmente desaparecer algunas, o para converger otras. Y finalmente año a año, ese listado de novedades se ve alimentado con la aparición de nuevos términos, conceptos y novedades que tratan de generar impacto primero, captar seguidores después e infiltrarse lentamente en el intrincado y entramado laberinto de las soluciones tecnológicas. Las que lo logran, rápidamente se convierten en una necesidad para las compañías y el mercado, y paralelamente en un dolor de cabeza para los administradores y los clientes o usuarios de los servicios.

Es muy común también encontrar herramientas que reemplazan literalmente a otras, mejorando el rendimiento y facilitando la operación, pero siempre dependiendo del contexto en el cual se insertan. Aparecen otra vez nuevos términos, conceptos y novedades prometiendo la “felicidad eterna”, funcionalidades que se solapan con funcionalidades ya existentes en otras herramientas, interfaces que dejan de funcionar y deben reacomodarse o reescribirse, nuevas herramientas imposibles de integrar etc. Adicionalmente existen muchas otras cuestiones que deben mantenerse “mientras tanto” se transita hacia la arquitectura objetivo, la cual en el mejor de los casos está planificada y se conoce el camino a recorrer para alcanzarla.

La arquitectura de TI sobre la cual montar las aplicaciones de misión crítica no escapa a cualquiera de las aristas de esta realidad planteada. Pero, ¿qué se puede hacer para detener la enorme maquinaria que se esconde detrás de las soluciones informáticas, sin olvidar aquellos procesos que son críticos? O planteada la misma pregunta desde la vereda de enfrente: ¿qué se puede hacer para NO detener la maquinaria mencionada anteriormente, y al mismo tiempo apoyar a nuestros procesos críticos garantizando que seguirán operando más allá de lo que pase con el resto de la organización?

La respuesta no es fácil y depende de infinidad de factores, pero con un profundo análisis, el establecimiento de un nuevo orden para el ya existente, una clasificación basada en funciones de cada uno de los engranajes y su correcta categorización, sin dudas que allanará el camino. Flexibilidad, simplicidad, adaptación, reutilización son algunos de los términos y conceptos que, si bien no son nuevos, se proponen reflotar en este trabajo. El resultado esperado es un meta-modelo de arquitectura sobre la cual montar aplicaciones de misión crítica, que bajo los conceptos explicados anteriormente, sea aplicable a problemáticas similares bajo circunstancias diferentes, pero al mismo tiempo no escape a la realidad que afecta a las organizaciones y sus procesos.



Justificación del Trabajo

La idea nace como una necesidad particular del Negocio Azucarero en el norte del país. Articulando distintos componentes de la infraestructura existente y, moviendo cierta lógica de negocios desde los Sistemas Centrales hacia el punto remoto de operación, se logró desarrollar una arquitectura lo suficientemente robusta, simple, adaptable y por ello también flexible, como para soportar la operación continua del Ingenio.

Si bien en ese trabajo hubo ideas prestadas de otras soluciones existentes, ideas que surgieron inicialmente como respuestas a las necesidades, y otras que nacieron como combinación de distintos factores, la iniciativa de este trabajo nace ante la no disponibilidad de un marco referencial y metodológico formal que apoye el diseño de soluciones de similares características para problemáticas similares, y que al mismo tiempo sea aplicable a otros negocios, independientemente de la tecnología utilizada, fundamentando las bases teóricas sobre las cuales se realizó el proyecto

Al mismo tiempo este trabajo se propone como una continuidad al anteriormente realizado, limando algunas inconsistencias que en la práctica pudieran haber existido, y sumando otras necesidades encontradas en el camino; pero tomándolo sólo como una muestra más de la aplicabilidad de lo que se pretende formalizar.

El análisis se realizó a la luz de las experiencias adquiridas en la formulación de la solución para el Negocio Azucarero mencionado anteriormente, pero abordando temáticas propias de Arquitectura de TI, combinando elementos de distintas disciplinas relacionadas.

La innovación reside en la estructuración de un nuevo modelo de arquitectura para aplicaciones de misión crítica que surgen como la necesidad de aislar los sistemas centrales de las necesidades regionales en casos que sean conveniente, como por ejemplo una falla del mismo, de tal forma de mantener la operación continua de un sitio remoto. Esto también supone la posibilidad de trabajar en línea con los sistemas centrales (modo sincrónico) o fuera de línea de los mismos (modo asincrónico); siempre hablando dentro del alcance de los procesos que componen los circuitos críticos de la operatoria local.

Marco Teórico

Los procesos de misión crítica deben estar soportados por soluciones (sean de software, hardware, ambas o ninguna) que consideren sus necesidades, características y particularidades de manera tal de garantizar su correcto funcionamiento en el momento oportuno. Cuando estos procesos están apoyados en soluciones de software y/o hardware, la infraestructura de TI sobre la que se ejecutan debe estar correctamente diseñada, pensada y configurada a tales fines. Estas cuestiones serían difíciles de mantener y administrar en los casos donde no exista una estrategia clara y precisa, acerca de cómo estas soluciones se construyen y que componentes se utilizan para ello, dada la gran cantidad de elementos aplicables a cada necesidad. Esta estrategia es la que baja (o debería bajar) desde la dirección de TI de la compañía y que delinea la arquitectura de TI.

Sin embargo, para que la arquitectura de TI sea eficiente, flexible, adaptable y precisa, debe existir cierta alineación entre los procesos de negocios y la/s tecnología/s que los soportan, lo cual se logra con un diseño de la arquitectura empresarial adecuado a cada necesidad.



Si bien hay cierta relación en tal sentido, esto no significa que cada una de las capas mencionadas en los párrafos anteriores necesariamente tenga que estar contenida dentro de la siguiente, sino que en la práctica la interacción entre cada una de ellas permite diseñar procesos de negocios soportados por tecnologías adecuadas, ajustadas a las necesidades de cada caso. Esto hace a los procesos de negocios reaccionar de forma más ágil a los cambios, ser más eficientes, gestionar la información de modo estratégico, contribuyendo a la generación de conocimiento, componente básico de la innovación.

Si bien en su mayor parte el trabajo es de desarrollo propio condimentado en una proporción menor por actividades de investigación y otras experiencias de proyectos anteriores, se realizó sobre la base de modelos conceptuales existentes, combinando los distintos elementos que los componen.

Las siguientes disciplinas constituyen un marco referencial base sobre la cual se apoyó el trabajo, y por ello durante el desarrollo se realizaron continuas referencias a los elementos que se consideraron necesarios:

- **Arquitectura Empresarial:**

Sin entrar en la discusión acerca de si la Arquitectura Empresarial constituye una disciplina por sí misma o no, la perspectiva desde la cual se utilizó este marco de referencia define a la actividad como una práctica estratégica que permite según Gartner Group, “trasladar una visión y estrategia de negocio en un cambio efectivo, comunicando las capacidades actuales y repensando los principios y los modelos que describen el estado futuro de la empresa y facilitan su evolución”, lo cual constituyen una base fundamental para el diseño de los procesos que lo harán efectivo.

Si bien en muchas referencias se establece una fuerte relación con la tecnología, el concepto que interesa a este trabajo es el de moldear los procesos necesarios para sustentar el negocio de una forma integrada y coherente, al tiempo de ofrecer un medio para alcanzar el entendimiento y la conceptualización para la decisión en cuestiones estratégicas.

Metodología

Marco Metodológico:

De acuerdo a su definición, la metodología hace referencia al conjunto de procedimientos utilizados para alcanzar los objetivos de una investigación científica, una exposición doctrinal o descripción que requiera habilidades, conocimientos o cuidados específicos.

La Metodología Científica, la Metodología del Conocimiento y la Metodología Empírica, aportan un conjunto de procedimientos que indistintamente se utilizaron durante el desarrollo del trabajo, sin que necesariamente se haya hecho referencia a la metodología y/o procedimiento asociado utilizado.

Las metodologías nombradas en principio aportaron la formalización necesaria para:

- la conexión entre el lenguaje ordinario y el lenguaje teórico que supone la abstracción de contenidos en diferentes niveles de complejidad, desde los más simples y cotidianos a los más elaborados y abstractos;



- la utilización de meta-lenguajes en los casos que correspondieron y fueron aplicables;
- la conceptualización de fenómenos que comparten características teóricas y la aprehensión de características generales del/los objeto/s en estudio (abstracción);
- adicionalmente, la conceptualización permitió abstraer la realidad para describir y explicar lo relevante a la disciplina del conocimiento;
- la generalización para definir el objeto de estudio de acuerdo al conjunto de características que se comparten o se aplican a contextos similares;
- la formalización para explicar el objeto de estudio desde el punto de vista de los elementos que la componen y las relaciones que se pueden inferir entre ellos;
- la argumentación, transformando el objeto de estudio y/o sus componentes en observables, es decir aspectos concretos de la realidad que deben ser entendidos dentro de un marco teórico de referencia. Un argumento es la exposición fundamentada de un número de premisas que nos permiten sostener una afirmación;
- la operacionalización, es decir el proceso de “aterrizar” el marco teórico. La correcta realización del mismo es condición para definir el tipo de datos que se requiere obtener, como así también la técnica mediante la cual se obtienen;
- el sistema de referencias documentales con el cual se trabajará, y que no significa que no se modifique durante el transcurso del trabajo.

Adicionalmente, la investigación que aportará los contenidos necesarios para fundamentar técnicamente el desarrollo, se apoyará en:

- investigación documental;
- investigación de campo;
- investigación descriptiva;
- investigación aplicada;
- investigación experimental.

Metodología Propuesta:

A continuación se describe el proceso utilizado para obtener los objetivos prefijados para el trabajo, de acuerdo a los lineamientos establecidos en el Marco Metodológico de referencia del apartado anterior. Este proceso describe a grandes rasgos las distintas etapas por las que atravesó el trabajo tanto en etapas de investigación como desarrollo propio de los temas, transformando lo que inicialmente fue un problema y luego una idea, en un producto definido y aplicable.

Se hace referencia a dos tipos de Fuentes de Información:

- Experiencia: debido a que el trabajo se plantea como continuidad de la experiencia adquirida en el trabajo realizado para el Ingenio Azucarero, son obvias las referencias al mismo; como así también otros trabajos relacionados que aplican en parte, componentes o elementos tratados;



- Otras Fuentes de Información: usados para apoyar con material anexo y para enriquecer con fundamentos principalmente técnicos o teóricos el desarrollo.

Adicionalmente, es frecuente encontrar a lo largo del trabajo, menciones a antecedentes y conocimientos previamente adquiridos.

Este tipo de información en estado inicial suele venir en un formato fácilmente entendible, utilizando un lenguaje natural, común u ordinario. Al provenir de distintas fuentes, son escasas sino nulas las posibilidades de que se encuentren en un orden correcto, siempre a los fines del trabajo. Es por ello que a modo de filtros, la información fue debidamente tratada, evaluando su pertinencia, aplicabilidad, relevancia y factibilidad técnica, en su correcto orden, de tal forma de facilitar el entendimiento.

Como se mencionara párrafos arriba, la idea general es que este tipo de arquitectura contribuya al diseño de soluciones de misión crítica robusta, escalable, flexible y adaptable, como marco de referencia. Esto obliga a extremar los cuidados como para no extender los alcances del trabajo pero al mismo tiempo no dejar afuera componentes o elementos que puedan significar su aplicabilidad o no bajo condiciones generales a otras problemáticas.

La utilización de un lenguaje teórico y/o técnico, o aún más elaborado meta-lenguaje, aporta las bases necesarias como para avanzar en la conceptualización de los distintos elementos que componen el objeto de estudio y permite estructurar sus relaciones para conformar un todo, el cual no podría ser aplicable a cualquier problemática de similares características a la experimentada, sin cierta generalización, de tal forma de que lo expuesto no se convierta en una solución específica y puntual, y se acerque lo más posible a la formalización de como estructurar soluciones bajo esta arquitectura.

Se logra esto sobre la base del ejemplo, la experimentación y sus conclusiones por el lado menos formal del proceso de investigación; y el análisis, la argumentación y la operacionalización, por el lado más formal del mismo. La correlación de los distintos elementos expuestos más las inferencias que a partir de ellos se determinen, proporcionaron los recursos necesarios para modelar formalmente el objeto que se pretende estudiar.

La Metodología general resultante como integración de los procedimientos que la componen y adaptados a los objetivos, se puede resumir en el siguiente esquema de bloques:

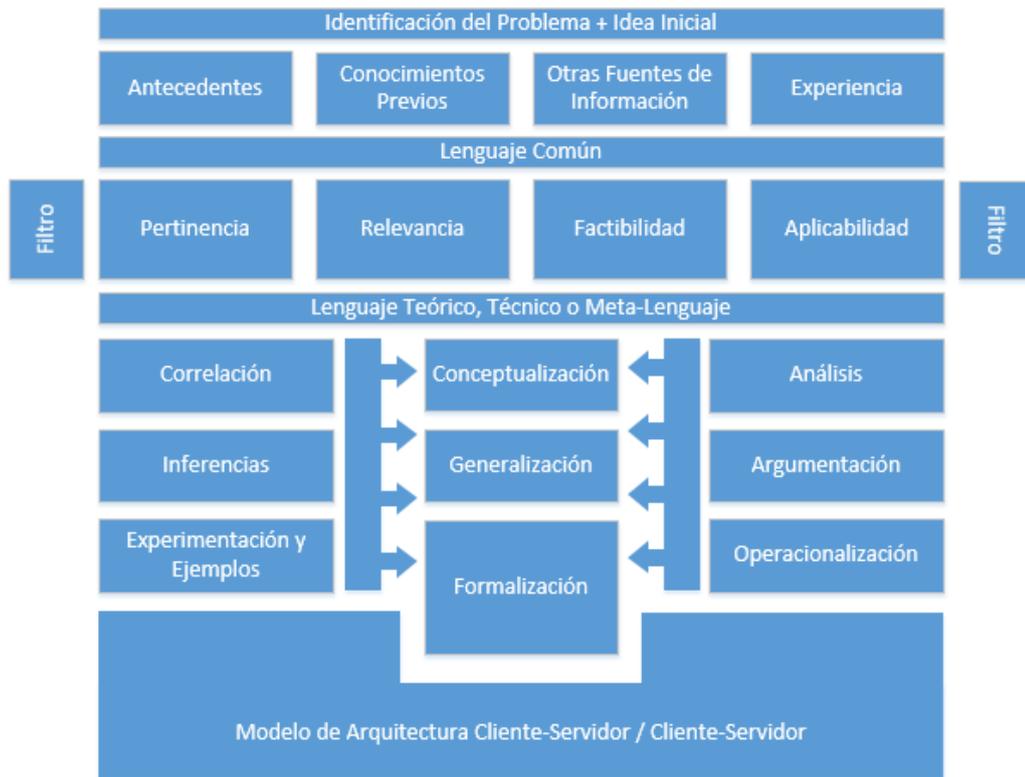


Fig. 1 – Metodología



Capítulo

2

Antecedentes

Esta unidad provee al lector de una breve introducción acerca del origen de la necesidad y como fue diseñada la solución. En este capítulo se dará una rápida revisión al problema original, a la solución aplicada y a la forma en la cual trabaja la misma, incluyendo sus dos formas de operación, a modo de introducción específica de lo que en próximos capítulos se generalizará.

El Problema Inicial

En el norte del país, la temporada de zafra arranca durante el mes de abril y dependiendo de distintos factores, puede extenderse hasta fines de Noviembre, período durante el cual la actividad es extremadamente intensa, a los fines de procesar toda la caña disponible sin detener la operación del Ingenio, lo cual acarrearía altos costos y bajas tasas de productividad.

A modo de resumen del proceso, la caña de azúcar que llega proveniente de los distintos frentes de cosecha, es analizada para determinar su calidad (rendimiento porcentual de azúcar, humedad, pureza etc.), para lo cual se utiliza un moderno sistema de extracción de muestras, conocido como Core Sampler. El objetivo es lograr que solo ingrese a la molienda materia prima en condiciones óptimas.

La elaboración del azúcar consta básicamente y a grandes rasgos de los siguientes procesos:

- cosecha de la caña de azúcar en fincas propias o de terceros;
- traslado a planta fabril por parte de transportes propios o de terceros y descarga;
- preparación de la caña y obtención del azúcar y derivados;
- empaque y almacenamiento para su posterior distribución.

La preparación de la caña consiste en un complejo proceso para dividirla finamente, sin pérdida de jugo. Posteriormente se extrae la sacarosa presente en la caña preparada, tarea que se realiza mediante lavados sucesivos con agua caliente o jugo diluido en un difusor. Como dato adicional, el Ingenio produce más de 135 mil toneladas de azúcar al año, en forma casi ininterrumpida durante toda la temporada de zafra.

Para tener una idea numérica de lo que la operación significa:

- Para no detener la planta, se necesita abastecerla a un ritmo promedio de 450 tns. de caña por hora;
- Suponiendo que cada camión tenga una capacidad promedio de 30 tns. se necesita dar ingreso a aproximadamente 15 camiones por hora;
- 15 camiones por hora significan un ritmo de ingreso de 1 camión cada 4 minutos;
- Adicionalmente, cada camión pasa dos veces por la báscula para pesar carga completa primero y tara después.



Para soportar tamaño proceso es necesario un sistema lo suficientemente ágil y aceitado como para no detenerlo.

En un entorno de TI centralizado, los clientes remotos pueden correr con algunas desventajas. En el caso puntual del Ingenio Azucarero, era una de las principales características, incluyendo a los procesos que se entendía como críticos para el negocio. Como consecuencia, la operación estaba altamente expuesta a cortes en las comunicaciones. A todo esto se sumaba que la infraestructura local era inadecuada o escasa. Actualmente sigue existiendo solo un proveedor con capacidad de proveer enlaces de comunicaciones, con lo cual antes de la implementación, la dependencia era total. Si bien se habían tomado algunas acciones para mitigar el riesgo contratando un enlace secundario, seguía siendo susceptible a cortes como el ocurrido durante Abril de 2013 que dejó sin servicios a todo el norte del país.

Como en todo diseño centralizado, la infraestructura remota es compartida y cualquier ventana de corte, sea programada o de emergencia significaba un corte en la operación del Ingenio, un alto impacto principalmente durante la temporada de zafra. El riesgo era conocido y se tomaron algunos recaudos como una pseudo-solución de contingencia hecha en casa, que en más de una vez permitieron mantener la operatoria ante problemas de disponibilidad de la infraestructura productiva.

La idea original

Si bien fueron varias las cuestiones que llevaron al diseño final, el disparador fue cierto concepto vertido alguna vez al implementar una solución de WMS (WareHouse Management System).

En breves palabras y para enfocar solo en lo que a este trabajo respecta, este producto proponía entre otras cosas la posibilidad de operar depósitos de productos terminados en forma sincrónica o asincrónica respecto de los sistemas centrales. Esto solucionaba la dispersión geográfica de los mismos y los inconvenientes asociados a la dependencia con los sistemas centrales y las comunicaciones.

Trabajando contra un servidor local, una desconexión del Sistema Central no significaba mayores inconvenientes para la operación y era totalmente transparente para los clientes.

Simplificadamente, una aplicación es cliente para un servidor, el cual a su vez es cliente para un segundo servidor, generalmente remoto.

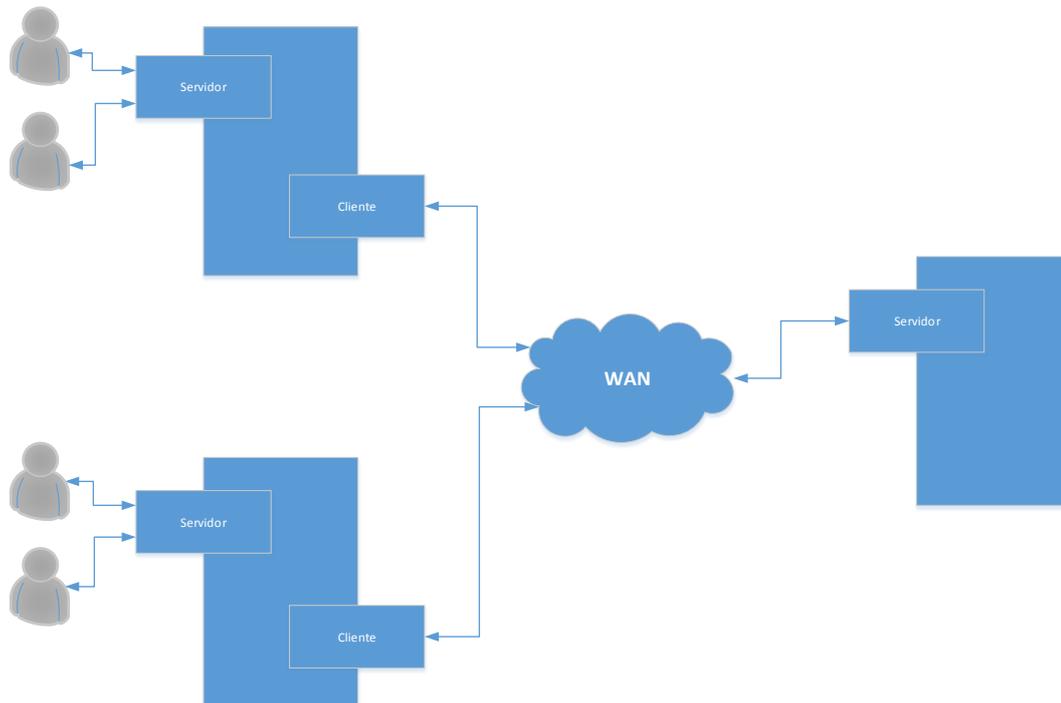


Fig. 2 – Arquitectura Cliente Servidor/Cliente-Servidor: Esquema Top Level

Breve descripción de la solución

Se necesitó realizar entonces un:

- Completo análisis de la situación;
- Identificación de puntos de falla;
- Identificación de los procesos que componían el circuito crítico, entendiéndose por ello a aquellos procesos que de fallar podían parar la operatoria del Ingenio;
- Identificación de los componentes de TI con los cuales se contaba;
- Diseñar una solución que combinara los componentes bajo algún esquema que permitiera operar aún en forma aislada del resto de la compañía.

A los fines de:

- Garantizar el normal abastecimiento de caña de azúcar al Ingenio;
- Minimizar los riesgos de una parada por inconvenientes en el sistema;
- Independizar a las funcionalidades que componen el circuito crítico de los inconvenientes que pudieran no disponibilizar el Sistema Central.

Si bien fue pasando por distintas fases de ajustes y adecuaciones el diseño final basando en el concepto anteriormente mencionado quedó como el diagrama de bloques que sigue:

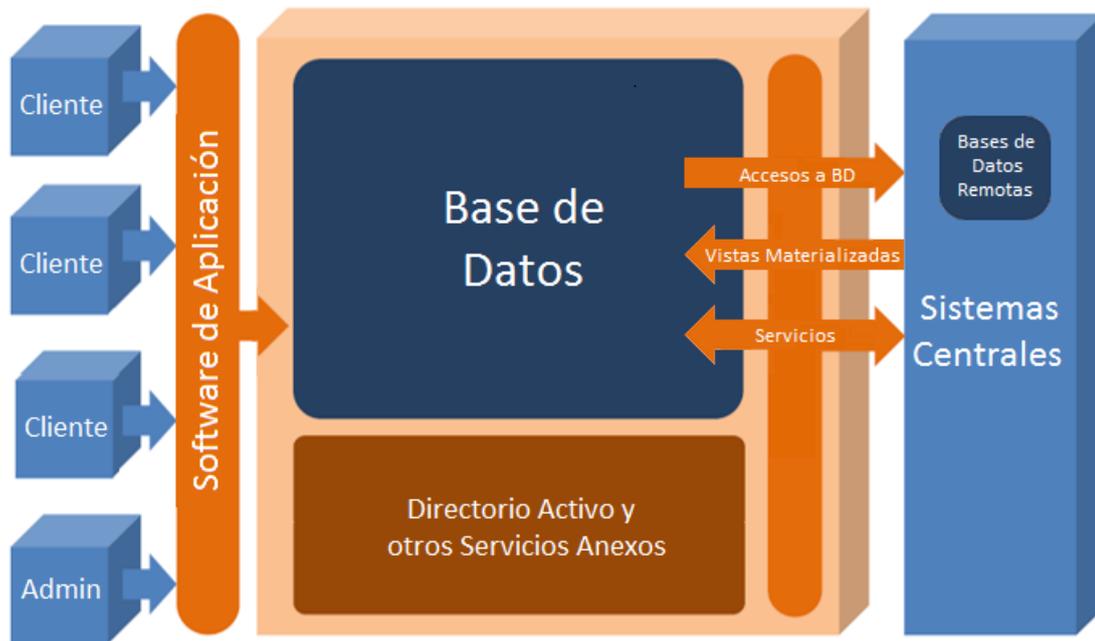


Fig. 3 – Esquema Solución Ingenio Azucarero

En este esquema simplificado se identificaron a grandes rasgos las funcionalidades contempladas como parte de la solución, cada una de las cuales cumplen una determinada función, evitando sus nombres comerciales. Al mismo tiempo sirve de punta pie inicial para comenzar a desglosar la solución en componentes, con el fin de generalizar la idea.

La solución es una solución cliente-servidor. La capa lógica y la capa de presentación se encuentran en cada cliente instalado en las PCs de escritorio de los usuarios. Los aspectos relacionados con la comunicación entre el cliente y el servidor se dan por sobre entendidos ya que se trata de software de aplicación que se implementa sobre una red TCP/IP común y corriente y no ofrece mayor complejidad técnica que la configuración mínima para su correcto funcionamiento.

La capa de datos reside en un servidor de datos accedido en forma local a través de la aplicación de tecnologías de acceso a Bases de Datos. Adicionalmente, existen servicios que complementan la solución, como un directorio activo, servicios DHCP para los clientes, backups de las bases de datos, deployment services, servicios de procesos de negocios etc.

Los sistemas centrales remotos son accedidos a través de los servicios WAN proporcionados por la compañía. En la interacción con los mismos fue necesario clasificar la ubicación donde el dato se genera, modifica o elimina:

- Dato generado localmente en el Ingenio;
- Dato generado remotamente en los sistemas remotos y que necesitan estar disponibles localmente;
- Datos generados en cualquiera de los extremos pero susceptibles de ser modificados en cualquiera de los mismos;



y posteriormente evaluar la forma en la cual cada uno de los casos sería resuelto. De acuerdo al diseño, y en caso de operación sincrónica con los sistemas centrales:

- Dato generado localmente en el Ingenio: solucionado a través de la gestión de transacciones distribuidas, escribiendo el dato localmente, remotamente y por último realizando el commit;
- Dato generado remotamente en los sistemas remotos y que necesitan estar disponibles localmente: solucionado mediante la utilización de vistas materializadas, funcionalidad propia de los Sistemas de Gestión de Base de Datos. El dato se replica desde los sistemas centrales al sistema local utilizando tiempos de refresco pre especificados, y no se modifican localmente;
- Datos generados en cualquiera de los extremos pero susceptibles de ser modificados en cualquiera de los mismos: solucionado mediante el desarrollo de un servicio que parametrizado con un tiempo de ejecución configurable y lógica propia del negocio. En caso que pudieran presentarse conflictos por controles de compromisos, la lógica contempla la regla de resolución del mismo.

Para el caso de operación asincrónica, es decir, bajo situaciones de no disponibilidad de los sistemas centrales:

- Dato generado localmente en el Ingenio: el dato se escribe localmente y la sentencia SQL que aplica el dato en forma remota se escribe en una tabla caché, que almacena todas las transacciones que deberían aplicarse remotamente y que por no disponer del sistema central no es posible realizarlo;
- Dato generado remotamente en los sistemas remotos y que necesitan estar disponibles localmente como maestros: son de rara actualización y generalmente permiten la operación mediante métodos manuales;
- Datos generados en cualquiera de los extremos pero susceptibles de ser modificados en cualquiera de los mismos: bajo condiciones de operación asincrónica, no existe posibilidad de actualización remota.

La solución está diseñada de tal manera que automáticamente se detecta una condición de sincronismo o no con el sistema remoto, por lo que es capaz de cambiar por si misma el modo de operación. Cuando el cambio es en el sentido “sincrónico → asincrónico”, la aplicación simplemente escribe en una tabla caché las sentencias SQL en el orden correcto de todo lo que debería escribir en forma remota. Cuando el cambio es en el sentido “asincrónico → sincrónico”, un servicio comienza a aplicar en forma remota y en el orden correspondiente las sentencias SQL almacenadas en el caché durante la operación asincrónica.



Capítulo

3

Arquitectura Empresarial

Este capítulo recopila conceptos previos necesarios para el análisis del objeto en estudio y relacionados con la disciplina de la Arquitectura Empresarial, base para entender las cuestiones sobre las cuales el trabajo propone echar luz. Realiza un breve recorrido por ciertos conceptos muchas veces sobre entendidos, pero sobre entendidos a medias o no dentro de lo que al alcance del trabajo respecta, tratando de alinearlos en el sentido que se le pretende dar.

Arquitectura Empresarial

La arquitectura empresarial es una práctica que guía el análisis, diseño, planeamiento e implementación de procesos empresariales, utilizando un enfoque holístico en todo momento, para procurar el desarrollo y la ejecución de la empresa en forma exitosa. Aplica principios y prácticas que guían a la empresa a sus objetivos de negocios, de información, de procesos y de cambios tecnológicos necesarios para llevar a cabo la estrategia, utilizando varios aspectos de la organización para identificarlos, motivarlos y alcanzarlos.

Desde el punto de vista del negocio, la Arquitectura Empresarial se lleva a cabo en todo momento, se tenga conciencia de ello o no, o esté bien definida o no el área que la lleva a cabo dentro de la estructura de la organización. Relaciona los procesos de negocio con la tecnología que la soporta, permitiendo evaluar fortalezas y debilidades, compatibilidades y alternativas que permiten delinear las estrategias que deriven en un modelo arquitectónico de la solución que se aplique.

La Arquitectura Empresarial modela los procesos de negocios, haciendo foco no solamente en las estrategias, objetivos y metas empresariales, sino también en la forma en la cual la organización trabaja. Esto no significa que en el 100% de los casos esos modelos alcancen a la tecnología tal cual se la utiliza hoy en día. Para los casos en los cuales sí, se contemplan o deberían contemplar además estándares de TI propios y de mercado, componentes existentes, integración de los nuevos; etc. Sin embargo, en este punto ya estaríamos con un pie en el campo de la Arquitectura de TI, una de las razones por las cuales el trabajo del Arquitecto de TI debe estar 100% integrado al del Arquitecto Empresarial, aunque para los casos donde no se involucren aspectos tecnológicos el trabajo de este último podría no estar relacionado con el de Arquitectura de TI.

El propósito de la arquitectura empresarial es optimizar mediante el diseño los procesos de negocios, sean manuales o automáticos, en un ambiente integrado adaptable a nuevos contextos, e integrable a nuevos procesos, siempre apoyado en la estrategia de la empresa.

Uno de los factores fundamentales para el éxito de una organización es la gestión efectiva de la información. La Arquitectura Empresarial se enfoca en esta necesidad proporcionando un contexto para la evolución de los sistemas de información en respuesta a la dinámica de los negocios, permitiendo un correcto balance entre la eficiencia de los sistemas de TI y la innovación en los negocios.

Las ventajas de un correcto diseño acarrearán importantes beneficios a las organizaciones:



- Operación más eficiente mediante:
 - Reducción de costos de operación;
 - Organización más ágil;
 - Reducción de costos en la gestión de cambios;
 - Mano de obra más flexible;
 - Optimización de la productividad;
 - Gestión de la información más eficiente;
 - Reducción de costos en el desarrollo, soporte y mantenimiento de aplicaciones de software;
 - Incremento en la portabilidad de aplicaciones de software (estandarización);
 - Mejora en la habilidad para identificar y soportar incidentes o fallos;
 - Facilidad para realizar upgrades o cambios de determinados componentes;
 - Mejora el retorno de inversión, reduciendo el riesgo de futuras inversiones;
 - Reducción de la complejidad del negocio y de los sistemas de TI;
 - Flexibilidad para desarrollar, comprar o tercerizar negocios y/o soluciones de TI;
 - Implementación de Nuevos negocios más rápido, más simple y más barato;
 - Mejora la calidad de la toma de decisiones;
 - Integración de ambientes heterogéneos.

Si bien no es objetivo específico de este trabajo ahondar en las distintas opciones que se pueden encontrar, a continuación sigue un breve repaso de las alternativas más representativas que podemos encontrar en el mercado; de las cuales se han extraído distintos elementos para la elaboración de este trabajo.

Marco Referencial / Framework

Si bien dependiendo del estándar de Arquitectura Empresarial que se trate el concepto podría variar levemente, en general entenderemos por marco de referencia, Marco Referencial o Framework a la estructura fundacional o conjunto de estructuras básicas que pueden ser usadas para el desarrollo de un amplio rango de arquitecturas relacionadas. Proporciona métodos para alcanzar un estado de la organización a futuro, en términos de bloques de construcción, describiendo las relaciones entre los mismos y los componentes que lo integren. Adicionalmente, podría incluir un conjunto de herramientas que provean un vocabulario en común y estándares recomendados y productos compatibles que pueden ser usados en la construcción de los bloques.

Si bien esta definición general está enfocada en el Marco Referencial en relación a la disciplina de la Arquitectura Empresarial, similar significado será utilizado en este trabajo respecto de la Arquitectura de TI, y en particular de la Arquitectura Cliente-Servidor / Cliente-Servidor que se describirá.

EABOK – EA Body of Knowledge

La Arquitectura Empresarial como disciplina está en rápida evolución, pero aún es algo inmadura. De todas formas existe una considerable cantidad de conocimientos y experiencias disponibles. Basándose en ello, el EABOK desarrollado por la firma MITRE (MITRE's Center for Innovative Computing and Informatics) y



por Agencias Gubernamentales de los Estados Unidos, es una guía para el diseño y desarrollo de arquitecturas empresariales, alcanzando a las relaciones entre los elementos que la componen, describiendo como se integran entre sí los procesos de negocio, las organizaciones, las Tecnologías de la Información, las capacidades y la infraestructura, y como podrían hacerlo en el futuro.

También describe como los elementos en transición soportan los planes estratégicos de la organización. Define las necesidades de interoperabilidad y estándares y estructura los contenidos en un conjunto de vistas consistentes. Como todo cuerpo de conocimientos, se subdivide en áreas de conocimiento:

- **Planeamiento:** provee información y guías para la definición de los objetivos específicos del diseño y del alcance, que servirá de guía para el desarrollo. El planeamiento debería alinear y soportar los objetivos y metas de la organización, considerando factores anexos tales como estándares, políticas y leyes;
- **Administración:** alcanza al establecimiento, monitoreo y control del proyecto a lo largo de todo su ciclo de vida, comunicando el plan a los stakeholders, asegurando el soporte y la visión estratégica. Todo esto a través de mecanismos y estructuras de gestión y control. El administrador traduce el plan en acciones concretas tales como el monitoreo, la programación de tareas, la conformación de los equipos de trabajo, asignación de responsabilidades, manejo de expectativas, presupuesto etc.;
- **Desarrollo:** incluye a todas las actividades asociadas con la creación y en mantenimiento de la arquitectura empresarial en relación a propósitos específicos. Provee una guía para llevar a la empresa desde el estado actual hasta el estado deseado, siempre en línea con sus objetivos organizacionales. Este estado deseado podría alcanzar a cambios organizacionales, procesos de transformación de los negocios, integración de datos, reingeniería de sistemas etc.

Esta área de conocimiento también incluye:

- **Principios:** líneas generales de comportamiento bajo situaciones específicas;
- **Modelos:** aproximaciones, abstracciones o representaciones de actores, procesos, sistemas, información y/o tecnologías;
- **Metodologías y procesos:** conjunto de métodos, reglas y postulados para estructurar, planificar e implementar procesos de desarrollo, en un número de fases determinado por los datos de entrada y salida necesarios;
- **Marcos de referencia:** convenciones, principios y prácticas para la descripción de arquitecturas dentro de un dominio específico de aplicación, una estructura lógica para la clasificación y organización de información compleja;
- **Herramientas:** software de aplicación que ayuda al desarrollo, manipulación y gestión de los componentes necesarios;
- **Estándares:** que describen mejores prácticas, guías, especificaciones técnicas o criterios de funcionamiento para el gobierno del desarrollo y la implementación de una arquitectura empresarial;
- **Modelos de Referencia y Arquitecturas:** un modelo de referencia es un modelo abstracto o dominio específico de un conjunto claro y definido de conceptos. Puede representar componentes que sean parte de una idea, de funciones de negocios o de un sistema, permitiendo una comunicación clara y efectiva.

- **Utilización:** esta área de conocimiento provee referencias y guías para la correcta utilización de la arquitectura empresarial. Podría alcanzar a casos de estudio acerca de su utilización para acompañar decisiones relacionadas con los negocios, la administración, la ingeniería, las operaciones o finanzas. Incluye herramientas y técnicas para el análisis de costo/beneficio, diseño e ingeniería de sistemas, mejoras a la operación o a la visualización de datos de la arquitectura;
- **Medidas de Impacto:** a los fines de alcanzar los objetivos organizacionales, un programa de arquitectura empresarial necesita ser constantemente medido, revisado y ajustado, a los fines de evitar costos excesivos de desarrollo o producir y/o mantener artefactos obsoletos. Esta área de conocimientos provee referencias y guías acerca de cómo medir el impacto de un proyecto de Arquitectura Empresarial, incluyendo modelos de referencia y de madurez, lineamientos para recolectar e interpretar las mediciones, y sugerencias para cuando las mismas no superen las expectativas.

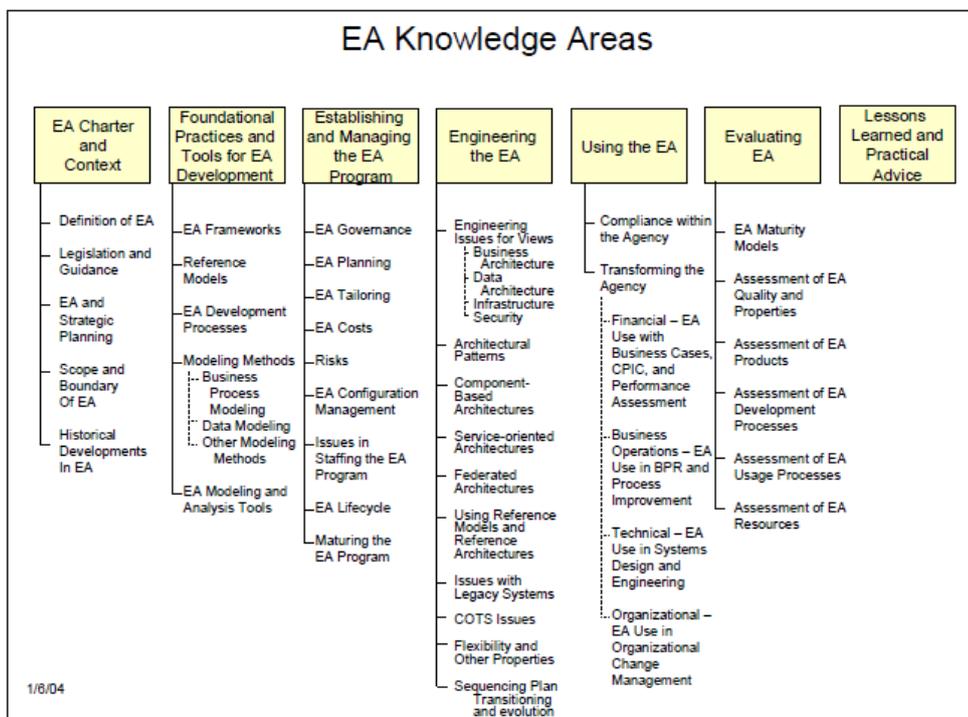


Fig. 4 – Áreas de Conocimiento del EA BOK

Fuente: <http://www.mitre.org>

Framework de Zachman

De cierta forma, el framework de Zachman es la formalización de la manera tradicional en la cual se diseñaban los procesos. Consiste en una matriz donde cada una de las columnas representan primitivas básicas de la comunicación, comunes para todos los diseños:



- Que;
- Como;
- Cuando;
- Quien;
- Donde;
- Porqué;

y las filas representan a cada una de las instancias de transformación de una idea abstracta:

- Identificación
- Definición
- Representación
- Especificación
- Configuración
- Instanciación.

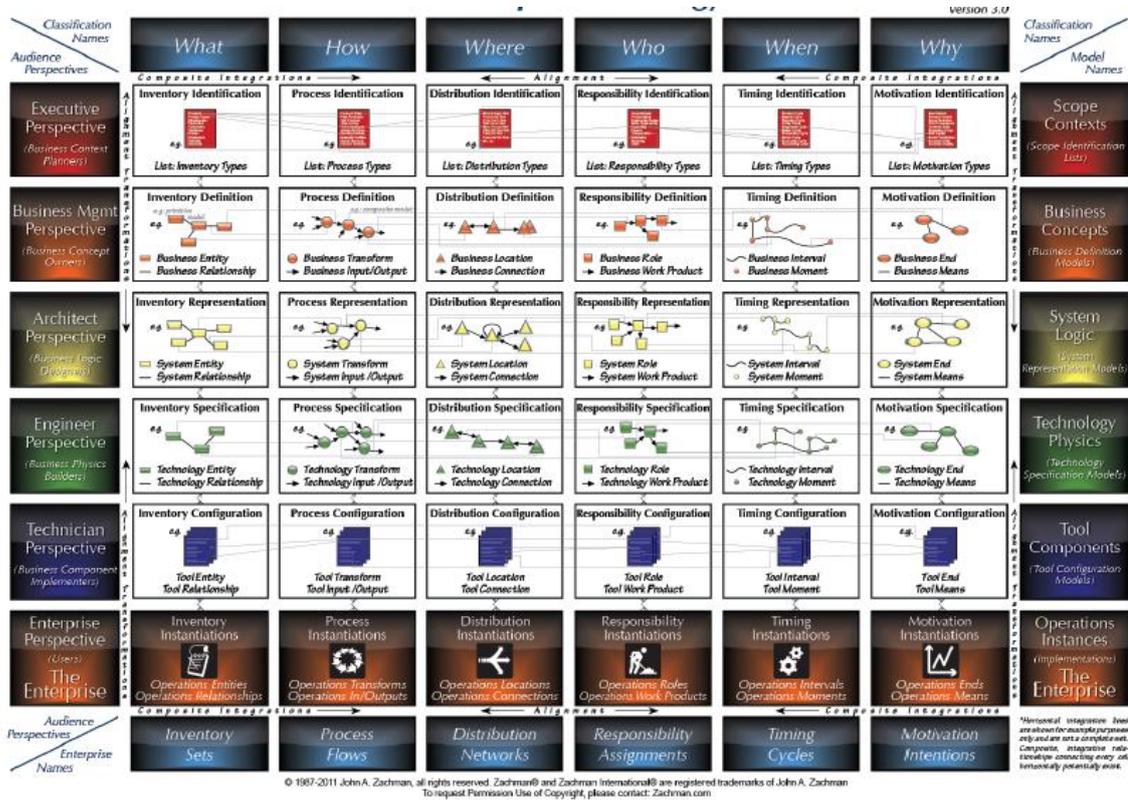


Fig. 5 – Metodología Zachman

Las distintas clasificaciones están representadas por la intersección de estas dos dimensiones, las cuales alcanzan al conjunto total de descripciones relevantes que podrían ser utilizadas para modelar cualquier cosa, y en particular, una empresa.



Este modelo se apoya en la teoría de la existencia de un conjunto estructurado de componentes esenciales de un objeto, para los cuales se pueden explicitar expresiones para su creación, operación y cambio. Ese objeto en particular puede ser una empresa, un departamento, una cadena de valor, una solución, un proyecto, un edificio o lo que sea, y como veremos más adelante, el diseño de una arquitectura para una aplicación de misión crítica.

Es necesario aclarar en este punto que el Framework de Zachman no es una metodología para la implementación o instanciación del objeto al cual se aplique. Un framework o marco referencial es la descripción de la manera en la cual se relacionan los elementos que dan forma al objeto, una estructura; mientras que una metodología es un proceso. Por un lado, una estructura establece una definición, mientras que por el otro, un proceso establece una transformación. Esta diferencia es fundamental para la comprensión del modelo: los procesos basados en estructuras ontológicas (que describen las relaciones entre los elementos que conforman un objeto), son predecibles y dan lugar a resultados repetibles. Las que no, son específicas, fijas y dependientes del dominio de alguna habilidad especial.

El framework de Zachman es un meta-modelo y a diferencia de una metodología, no implica nada respecto de:

- si se construyen modelos primitivos (intersección entre las dimensiones de la matriz) o si simplemente se construyen modelos específicos que puedan contener elementos primitivos;
- cualquier forma en la cual se diseñe la arquitectura;
- que tanta flexibilidad se pretende obtener en las implementaciones (modelos primitivos);

todo esto a pesar de que estas alternativas son significantes, identificables y metodológicas.

Fuente: <http://www.zachman.com>

The Open Group Architectural Framework (TOGAF)

Una muy buena práctica al diseñar una Arquitectura Empresarial es usar un modelo de referencia para lo cual TOGAF (The Open Group Architecture Framework) es una excelente alternativa. TOGAF es un marco de trabajo que propone un enfoque para el diseño, planificación, implementación y gobierno de una arquitectura empresarial; y que además provee una metodología para el diseño: el Método para el Desarrollo de Arquitecturas (ADM – Architecture Development Method).

La norma ISO / IEC 42010 define arquitectura como: "La organización fundamental de un sistema, enfocada en sus componentes, las relaciones entre cada uno de ellos y con su entorno y los principios que rigen su diseño y evolución". TOGAF extiende esta definición, ya que arquitectura puede tener dos significados, dependiendo del contexto:

- Una descripción formal de un sistema, o un plan detallado del sistema a nivel de componentes para guiar su implementación;
- La estructura de componentes, sus interrelaciones, y los principios y directrices que rigen su diseño y evolución en el tiempo.

El Método para el Desarrollo de Arquitecturas, define una secuencia de varias fases, cada una de las cuales contiene una serie de pasos para el desarrollo de una arquitectura. Es un proceso iterativo donde la interacción entre las distintas áreas es cada vez mayor, produciendo entregables que se van enriqueciendo en la medida que se avanza con el diseño. Cada iteración agrega recursos a un repositorio gestionado por un software de aplicación provisto por la organización TOGAF misma.

El proceso se enfoca en lo que agrega valor para la organización, entendiéndose como un proceso cíclico y continuo que agrega más y más contenidos y recursos al repositorio, sea de modificaciones a elementos existentes (objetos, artefactos, componentes etc.) o a nuevos, sin dejar de lado como los mismos se integran a la organización y relacionan con el resto de los componentes.

La estructura básica del método es la siguiente:

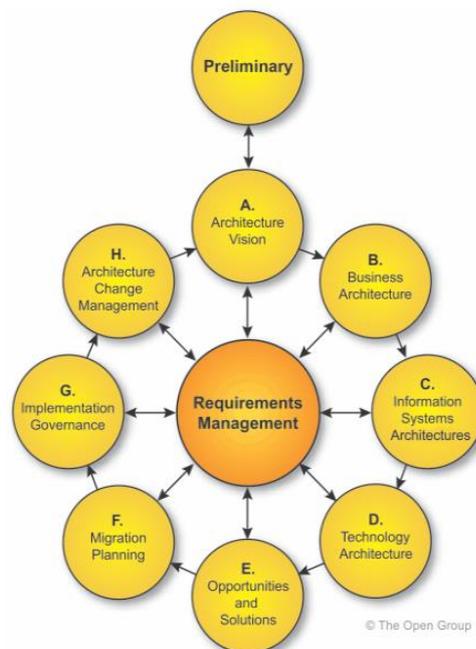


Fig. 6 – Metodología TOGAF

El resultado de este método es una Arquitectura Empresarial aplicable a la organización en estudio. La implementación de cualquier funcionalidad en cualquier sector de la organización y en cualquier momento del ciclo de vida de la misma, podría requerir el diseño o cambio de una o más de cuatro dominios de arquitectura:

- Arquitectura de Negocios: define la organización, estrategia y gobierno del negocio enfocado en procesos;
- Arquitectura de Datos: describe la estructura de una organización desde la perspectiva lógica y física de los datos y la gestión de los mismos;
- Arquitectura de Aplicación: define lineamientos para las aplicaciones que serán desplegadas, sus interacciones y relaciones con los procesos de negocio de la organización;
- Arquitectura de TI: que describe como los recursos de software y hardware deben ser requeridos para soportar el despliegue de los procesos, datos, servicios, etc.



En definitiva y respecto del Framework de Zachman, el estándar TOGAF no solamente provee una metodología para el diseño de la Arquitectura Empresarial sino que además provee herramientas de software para apoyar el proceso de construcción, gestión de cambios y repositorio de la información relacionada al mismo.

Fuente: <http://www.opengroup.org>

The Federal Enterprise Architecture Framework (FEAF)

Este framework fue desarrollado por el gobierno norteamericano para el diseño de Arquitecturas Empresariales de Agencias del Estado. Alcanza a organizaciones gubernamentales y socios del estado. Apoyado sobre el modelo de Zachman, define el negocio y la información necesaria para operarlo, la tecnología para soportarlo y los procesos transaccionales para la modificación de componentes existentes y la implementación de los nuevos componentes.

Define ocho componentes base:

- Factores de Arquitectura: representa dos tipos de agentes:
 - Factores del negocio, tales como leyes, iniciativas, presupuestos etc.;
 - Factores del diseño, tales como software de aplicación, hardware o una combinación de ambos;
- Dirección estratégica: guía al desarrollo de la arquitectura para la agencia objetivo, incluyendo su visión, principios, metas y objetivos. El modelo define los negocios, los datos, aplicaciones y la arquitectura tecnológica que le dará soporte.
Arquitectura actual: define como es la empresa actualmente. Es la representación de la capacidad y tecnología actual y se divide en dos partes:
 - Negocio actual;
 - Diseño de la arquitectura: datos, aplicaciones, tecnología etc.
- Arquitectura objetivo: define la arquitectura a construir. Representa la capacidad y tecnología que se pretende obtener a futuro:
 - Negocio objetivo;
 - Diseño de la arquitectura: datos, aplicaciones, tecnología etc.
- Procesos de Transición: Soporta la migración desde la arquitectura actual a la objetivo, incluyendo presupuestos, plan de migración, configuración, control de cambios etc.;
- Segmentos de Arquitectura: representa a los esfuerzos de arquitectura cros-áreas, como por ejemplo sistemas administrativos comunes. Representa una porción de la arquitectura total;
- Modelos de Arquitecturas: Define el negocio y los modelos de diseño que alcanzan a los segmentos mencionados anteriormente;
- Estándares: se refieren a todos los estándares, guías y mejores prácticas tomados como base por la organización.

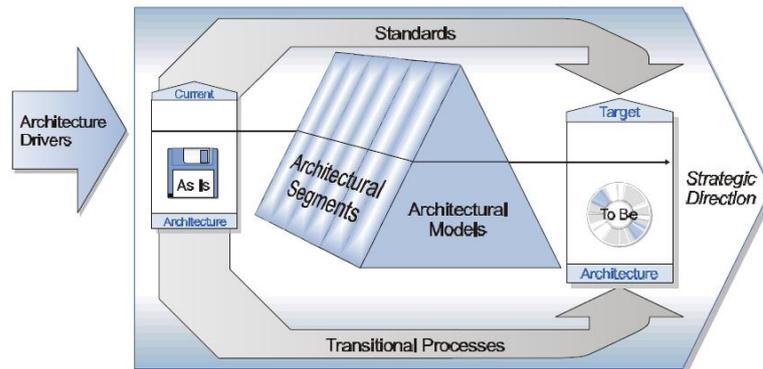


Fig. 7 – Metodología FEAF

FEA utiliza toma elementos de varios modelos de referencia que proporcionan una taxonomía y ontología común para describir los recursos de TI:

- Modelo de Referencia para el Rendimiento (Performance reference model - PRM): Modelo de referencia para medir el rendimiento de las soluciones de TI en su conjunto. Tiene tres propósitos básicos:
 - Producir información de rendimiento para contribuir a la estrategia y la toma de decisiones diarias;
 - Contribuir a articular las entradas con las salidas, para obtener los resultados deseados;
 - Identificar oportunidades de mejora del rendimiento.

PRM está compuesto por cuatro tipos de medidas:

- Misión y Resultados de los Negocios;
 - Resultados de Clientes
 - Procesos y Actividades
 - Tecnología.
- Modelo de Referencia para los Negocios (Business Reference Model – BRM): modelo de referencia para la operación del negocio para el Gobierno, independientemente de las agencias que la componen. Provee un marco organizado jerárquicamente que describe el día a día de las operaciones gubernamentales utilizando un enfoque funcional. Es la primera capa de la Arquitectura Federal y la más importante para el análisis de datos, componentes y tecnología.

Está compuesto por cuatro áreas:

- Servicios a la ciudadanía;
 - Modo de entrega;
 - Soporte a la entrega de Servicios;
 - Administración de Recursos del Gobierno.
- Modelo de Referencia de Componentes de Servicios (Service Component Reference Model – SRM): Clasifica a los componentes de servicios respecto de cómo soportan al negocio y a los objetivos de rendimiento. Se estructura a través de dominios verticales y horizontales,



independientemente de la función de negocio que desempeñen, permitiendo la reutilización de aplicaciones, funcionalidades, componentes, servicios etc.

Adicionalmente, cada uno de los dominios mencionados se dividen en tipos de servicio:

- Servicios al Cliente:
 - Gestión de Relaciones con los Clientes;
 - Preferencias del Cliente;
 - Asistencia al Cliente;
- Servicios de Automatización de Procesos:
 - Seguimiento y flujos de trabajo;
 - Routeo y programación de la ejecución;
- Servicios de Administración de Negocios;
 - Gestión de Procesos;
 - Gestión Organizacional;
 - Gestión de las Inversiones;
 - Administración de la cadena de Abastecimiento;
- Servicios de Activos Digitales
 - Administración de Contenidos;
 - Administración de Documentos;
 - Gestión del Conocimiento;
 - Gestión de Registro;
- Servicios de Análisis de Negocio;
 - Análisis y Estadísticas;
 - Visualización;
 - Conocimiento;
 - Inteligencia de Negocios
 - Reportes;
- Servicios de Back Office:
 - Gestión de Datos;
 - Recursos Humanos;
 - Gestión Financiera;
 - Administración de Activos;
 - Desarrollo e Integración;
 - Gestión del Capital Humano;
- Servicios de Soporte:
 - Gestión de la Seguridad
 - Colaboración
 - Búsquedas;
 - Comunicación;
 - Administración de Sistemas



- Administración de Formularios.
- Modelo de Referencia de Datos (Data Reference Model – DRM): Describe a los datos y a la información que soportan los programas del Gobierno y la operación de los negocios. Categoriza la información gubernamental en grandes niveles de detalle, a fin de estandarizar el intercambio de información entre procesos internos y externos a la organización;
- Modelo de Referencia Técnica (Technical Reference Model – TRM): Este modelo de referencia está basado en componentes técnicos, categorizando los estándares y tecnologías necesarias para soportar y permitir la entrega de componentes de servicios y funcionalidades, permitiendo la reutilización y estandarización de la tecnología y los servicios desde la perspectiva del Gobierno.

Consiste en:

- Áreas de Servicio: que representan la capa técnica que soporta la construcción segura, intercambio y entrega de los componentes de Servicio.;
- Categorías de Servicios: Clasifica niveles de tecnología y estándares respecto de los negocios o tecnologías que soportan, pudiendo alcanzar a más de un estándar;
- Estándares de Servicios: Define los estándares y tecnologías soportados por las categorías de Servicios.

Fuente: <http://www.enterprise-architecture.info>
<http://www.wikipedia.com>

The Gartner Methodology

Desde el enfoque que Gartner infiere, la Arquitectura empresarial es una disciplina proactiva y holística para dar respuesta a las fuerzas disruptivas a las cuales la organización está sometida, mediante la identificación y el análisis de los cambios necesarios para la visión y objetivos que se persiguen.

Bajo la premisa de que los proyectos de Arquitectura Empresarial exitosos están enfocados a los procesos, Gartner comenzó el desarrollo de su metodología definiendo primero un Modelo de Procesos para la Arquitectura Empresarial. El resultado fue un modelo no lineal, multi fase e iterativo, enfocado en el desarrollo de procesos empresariales, en su evolución y migración, y compuesto por subprocesos organizacionales, de gobierno y administración. Representa una síntesis de las mejores prácticas de como las organizaciones más exitosas han desarrollado, implementado y mantenido sus diseños de arquitectura.

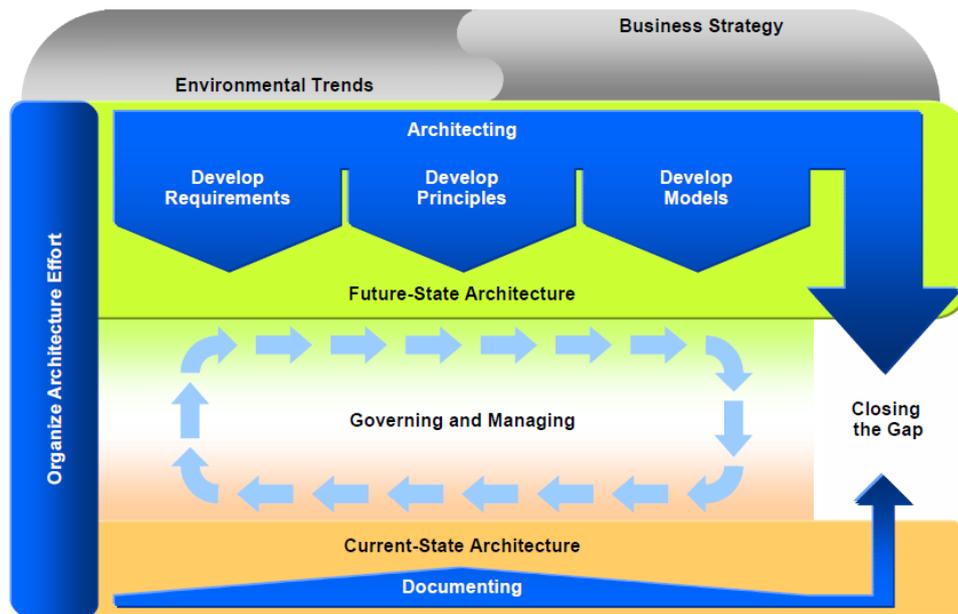


Fig. 8 – Metodología de Gartner

Definido el Modelo de Procesos, Gartner comenzó a desarrollar el Modelo de Referencia para Arquitecturas Empresariales, un complejo diseño con componentes abstractos. Este framework provee un contexto bajo el cual la organización puede ser estructurada de una forma consistente, definiendo los componentes y sus relaciones, dictando lineamientos para la formación de los equipos de trabajo y los roles de cada uno de ellos y proveyendo un conjunto de reglas semánticas y conceptos que permiten describirla.

Para el diseño del framework, Gartner adoptó un enfoque compatible con la norma IEEE 1471 (Recommended Practice for Architectural Description of Software Intensive Systems), estándar que fue reemplazado posteriormente por la norma ISO/IEC/IEEE 42010:2011 - Systems and Software Engineering — Architecture Description.

El framework de Gartner define un contexto, que alcanza a la estrategia de negocios y a las fuerzas externas que conforman e influyen en el contexto general en el cual se desarrolla la organización. Este contexto se puede abordar desde la combinación de tres puntos de vista:

- Puntos de Vista del Negocio: que alcanza a los procesos y organización de los negocios;
- Puntos de vista de la Información: que alcanza a la información que fluye por la organización;
- Puntos de vista de la Tecnología: que alcanza a los componentes de hardware y software de la empresa que dan soporte a la operación de los negocios.

Un enfoque orientado a aspectos, permite la articulación con puntos de vista adicionales que caractericen a la organización bajo diseño.

Aunque tanto el Modelo de Proceso como el framework han cambiado significativamente desde su versión original, la solución de arquitectura, donde los sistemas que soportan la operación del negocio se especifican y definen, se da en la intersección de los distintos puntos de vista que conformen el modelo.



Gartner maneja el concepto de “ladrillos, especies de unidades de construcción o “bloques” elementales, y de “patrones” o “templates” de arquitectura, para la reutilización de componentes comunes.

Fuente: <http://www.gartner.com>

Conclusiones

Es imposible determinar si un estándar es mejor que otro. Se puede afirmar que existen ciertos aspectos de cada uno que podrían aplicar mejor a ciertos tipos de organizaciones. De igual forma cualquier comparación que se pueda establecer corre el riesgo de caer en la parcialidad o subjetividad de quien lo evalúe.

A continuación se detallan los doce aspectos más utilizados para comparar y evaluar metodologías de arquitectura empresarial, según la bibliografía disponible. Si bien EA BOK podría ser evaluada de igual forma, trata de una recopilación de lecciones aprendidas, con un enfoque distinto en su concepción respecto de las otras metodologías. La evaluación de estos aspectos podría variar de organización en organización dependiendo de las características de las mismas y lo abarcativo del proceso de adopción:

- Taxonomía: se refiere a cuan bien la metodología puede ser utilizada para clasificar artefactos arquitectónicos.
- Procesos: se refiere a que tan bien el paso a paso de la metodología aplica a la creación de la arquitectura empresarial.
- Modelo de Referencia: se refiere a qué tan útil es la metodología para la construcción de modelos de referencia.
- Prácticas: se refiere a que tanto la metodología asimila las características de la organización y desarrollo una cultura en la cual es valorada y utilizada.
- Modelo de Madurez: se refiere a que tanto la metodología evalúa la eficacia y madurez de la organización en el uso de la arquitectura empresarial;
- Enfoque de negocios: se refiere a que tanto la metodología se centra en el uso de la tecnología para impulsar el valor del negocio;
- Guía para el Gobierno: se refiere a que tanto la metodología aportará al entendimiento y creación de modelos de gobierno;
- Guía para el Particionamiento: refiere a que tanto la metodología guía el particionamiento de la organización en unidades autónomas que faciliten su gestión por separado;;
- Catálogo: se refiere a cuan bien la metodología permite construir un catálogo de activos de arquitectura que podrían ser reutilizados en el futuro;



- Independencia: se refiere a la probabilidad de que la adopción de la metodología bajo evaluación produzca una dependencia total de la consultora que brinde el servicio;
- Disponibilidad de la información: se refieren a la cantidad y calidad de la información disponible acerca de la metodología en evaluación, ya sea en forma gratuita o de bajo costo;
- Retorno del Valor: se refiere al tiempo que se necesita trabajar en la metodología antes de que empiece a devolver valor.

Cada aspecto fue evaluado de acuerdo a la siguiente definición:

1. Muy Pobre
2. Inadecuado
3. Aceptable
4. Muy Bueno

En Resumen:

Aspectos	Valoración			
	Zachman	TOFAG	FEAF	Gartner
Taxonomía	4	2	2	1
Proceso	1	4	2	3
Modelo de Referencia	1	3	4	1
Prácticas	1	2	2	4
Modelo de Madurez	1	1	3	2
Enfoque de negocios	1	2	1	4
Guía para el Gobierno	1	2	3	3
Guía de Particionamiento	1	2	4	3
Catálogo	1	2	4	2
Neutralidad del Vendor	2	4	3	1
Disponibilidad de Información	2	4	2	1
Tiempo para devolver valor	1	3	1	4

Fig. 9 – Comparación de Metodologías para la Arquitectura Empresarial

Para determinar cuál de las metodologías aplica mejor a una organización y partiendo del cuadro anterior, se podría:

- Eliminar del cuadro los criterios que no se consideran importantes para la organización objetivo;
- Agregar cualquier otro criterio que se considere importante y evaluarlo en el marco de la organización objetivo;
- Cambiar cualquier valoración del cuadro con la cual no se esté de acuerdo para cualquier metodología.

De esta forma se podrá obtener una visión rápida y clara de cuál de las metodologías mejor aplica a la organización. De igual forma, podría no haber una metodología que se ajuste específicamente a las características de la organización. Justamente una de las fortalezas de este tipo de disciplinas es la



posibilidad de conjugar elementos de cada una para dar lugar a una nueva que aplica puntualmente a lo que se está necesitando. Inclusive, es factible su aplicación parcial para atacar un determinado tipo de necesidad.

Fuente: <http://www.microsoft.com>

Resumen

Un Marco referencial para la Arquitectura Empresarial, es una representación gráfica, abstracta y pragmática que permite describir a la organización en su conjunto y a los procesos que permiten operar sus negocios. Este marco referencial ayuda a los arquitectos (y para este trabajo en particular a los Arquitectos de TI) a clasificar y/o seleccionar distintos modelos aplicables, clarificando distintos puntos de vista y niveles de detalle, lo cual permite establecer claramente límites y responsabilidades.

Las metodologías utilizadas por la disciplina de la Arquitectura empresarial, describe los diferentes pasos y entregables necesarios y los resultados que se esperan de esos procesos. La existencia de distintos estándares no indica que posean una metodología determinada o específica ni que se deba optar por uno u otro a la hora de describir una empresa, sino que se pueden combinar para un mejor resultado.

Se puede entender entonces a la Arquitectura empresarial, como la descripción de la empresa y la definición abstracta de sus procesos y al Marco de Referencia como un modelo de comunicación para su desarrollo, un conjunto de modelos, principios, servicios, enfoques, estándares, diseños conceptuales, componentes, y configuración que guía el desarrollo de un aspecto particular de la Arquitectura Empresarial.

Como ventajas de su aplicación y utilización se puede mencionar:

- Mejora la adaptación de las áreas de TI enfocándolas al negocio;
- Se enfoca en los objetivos de la organización;
- Mejor visibilidad del trabajo individual en correlación a como ayuda a alcanzar los objetivos de la organización;
- Reduce el número de implementaciones fallidas;
- Reduce la complejidad de los sistemas;
- Agiliza a la organización para la adopción e integración de nuevos sistemas;
- Mejora el alineamiento entre los entregables de TI y los requerimientos de negocio.

Más información: <http://www.iso-architecture.org/ieee-1471/afs/frameworks-table.html>



Capítulo

4

Arquitectura de TI

En este capítulo se define como las tareas propias de la Arquitectura de TI se insertan dentro de la disciplina empresarial descrita en el capítulo anterior, y particularmente a los fines de describir el objeto en estudio, se definen varios conceptos relacionados a la disciplina en cuestión que proveerán al lector no solo de información sino de orientación en cuanto al sentido y alcance que se les dará en el resto del trabajo.

Al mismo tiempo y como parte de la Arquitectura de TI se introduce en los conceptos mínimos necesarios para entender el concepto de Aplicaciones de Misión Crítica, objetivo de la Arquitectura que se pretende describir.

Arquitectura de TI

Según Gartner y en términos de TI, la arquitectura se define como “el diseño de los componentes lógicos y físicos de cómputo y las relaciones entre estos. Define el hardware, software, métodos de acceso y protocolos usados a través del sistema. También define un marco de trabajo y un conjunto de lineamientos para construir un nuevo sistema, o un proyecto que está desarrollado, implementado, mantenido y usado para describir y orientar la manera como el área de Tecnología de Información (TI), y los elementos para gestionar la información están relacionados con el fin de cumplir con los objetivos de la empresa.”

En relación a esto y siempre según Gartner, la Arquitectura Empresarial define vistas generales de la arquitectura, a saber:

- Arquitectura de Negocio: incluye la misión, la visión, la estrategia y los procesos de negocio de la empresa, entre otros;
- Arquitectura de la Información: incluye modelos ontológicos, información, datos, representación, almacenamiento, entre otros;
- Arquitectura de Aplicaciones/Software: incluye las funcionalidades y aplicaciones;
- Arquitectura de Tecnología: incluye el hardware, comunicaciones, servidores de aplicación, gestión de bases de datos, entre otros.

Hablaremos de Arquitectura de TI como disciplina que integra estas cuatro vistas generales y que en su interacción con la Arquitectura Empresarial, está alineada a los mismos objetivos y proporciona los elementos necesarios para la construcción de soluciones robustas, flexibles, reutilizables y escalables. Aunque pueda ser aplicado a otros fines o pueda ser utilizado de otras formas, el foco para el trabajo tendrá eje alrededor de las aplicaciones de misión crítica.

Una solución de negocios basada en componentes tecnológicos necesita estar apoyada sobre una Arquitectura de TI que permita describir, estructurar y optimizar el diseño de los sistemas de información, lo cual permite a la organización:



- Obtener un conocimiento global sobre sus activos de TI;
- Establecer principios eficaces de gobierno de TI;
- Desarrollar una arquitectura de TI específica y planificar la transformación necesaria;
- Permitir la operación de los procesos de negocios manteniendo estándares tecnológicos;
- Minimizar la posibilidad de contar con componentes backlevel y por ende falta de soporte de los fabricantes de esos componentes;
- Permitir a los arquitectos de software construir aplicaciones robustas que satisfagan las necesidades de negocios, apoyadas sobre una arquitectura sólida, flexible y escalable, permitiéndoles la reutilización de componentes en los casos que sea aplicable y la incorporación de los nuevos, integrándolos en forma transparente.

Los arquitectos empresariales y más específicamente los Arquitectos de TI, son responsables de obtener una visión clara de la Arquitectura de TI de la organización y establecer su gobierno. Diversos autores y estándares abordan esta temática de manera aproximada y con algunas variantes entre sí, mezcla de varias cosas, lo cual puede llevar a la confusión. Adicionalmente, si se sigue un estándar específico para un modelo de negocio en particular, se corre el riesgo de dejar fuera de alcance procesos de negocios específicos de la organización, y/o la inclusión de los mismos dentro de ciertos límites y estándares. Lo importante en este punto para el Arquitecto de TI hacia arriba, es estar alineado a las definiciones que bajan desde la Arquitectura Empresarial, y hacia abajo diseñar e implementar una Arquitectura robusta conociendo las necesidades del negocio y las soluciones tecnológicas que pueden aplicar en ambos sentidos.

Es muy difícil establecer los límites entre la Arquitectura del Negocio, de la Información, de las aplicaciones o de la tecnología. En vez de tratar de hacerlo, conviene entender a la Arquitectura de TI como una sola entidad interdisciplinaria, con una permanente interacción entre sus miembros. Es por ello que cuando se menciona al “Arquitecto”, no se está hablando directamente de una sola persona, sino de un equipo de trabajo cuyo número podría variar dependiendo de la magnitud y la complejidad de la organización a la que se aplique. Inclusive, puede ser que sus miembros no sean permanentes, sino que el equipo de trabajo se conforme para trabajar sobre Arquitectura, y una vez definida vuelvan a sus labores cotidianas, para volver a reunirse cuando cuestiones relacionadas lo ameriten.



Fig. 10 – Alcances de Arquitectura

Lo expuesto en el gráfico anterior define los alcances de cada área en particular y no una escala jerárquica, todo dentro de lo que a la disciplina de Arquitectura como tal alcanza. Es más, un Arquitecto



Empresarial puede desarrollar funciones dentro del alcance del área de la Arquitectura de TI y/o dentro de la Arquitectura de Negocios o viceversa, sin perjuicio de sus atribuciones funcionales y sin que ello signifique diferencias jerárquicas respecto del resto del equipo. Esta interacción entre los distintos niveles de Arquitectura que podemos encontrar dentro de la organización es lo que define a la actividad como una actividad interdisciplinaria.

Si bien el concepto de elementos primitivos se desarrollará más adelante, a partir de ahora se debe tener en claro (y será eje conceptual dentro de este trabajo) que las actividades de arquitectura tienen que ver con el diseño, construcción, administración y cambios de modelos primitivos, es decir, modelos indivisibles que tienen una razón de ser que los definen en sí mismo. Si bien es cierto que pueden dividirse en componentes menos complejos, perderían la razón de ser.

Al objeto de este trabajo y para facilitar su comprensión e interrelación con la Arquitectura Empresarial, se desagregará a la Arquitectura de TI utilizando las primitivas básicas de comunicación que utilizara Zachman para su framework de Arquitectura Empresarial, aunque bien podrían utilizarse para definir cualquier otro tipo de disciplina, y de ahí su gran valor conceptual.

- **¿Qué es la Arquitectura de TI?:** de acuerdo a lo expuesto, a lo largo de este trabajo se entenderá por Arquitectura de TI a todas aquellas tareas relacionadas con el diseño, construcción, administración y cambios de modelos primitivos relacionados con la tecnología de la información, lo cual alcanza a los procesos de negocios, a la información en sí, a las aplicaciones y a la tecnología, para dar lugar mediante la combinación de distintos elementos a soluciones que satisfagan necesidades propias de la operación del negocio.

Las tareas propias de la arquitectura en general y de TI en particular, se definen normalmente en términos de elementos estructurales (primitivos) y las relaciones entre ellos. El diseño y desarrollo de Arquitecturas de TI alcanza, entre otras a:

- La descomposición del modelo en elementos estructurales, componentes de arquitectura, dominios y/o subsistemas, atendiendo a estándares empresariales que bajan desde la Arquitectura Empresarial y haciendo foco en estándares tecnológicos y de diseño, con el objetivo de garantizar la robustez, flexibilidad, escalabilidad e integración de las soluciones que se apoyen sobre la misma. Una buena descomposición del modelo en partes, satisface el principio de acoplamiento débil entre ellas, facilitado por las interfaces que simplifican cuestiones complejas en partes simples que pueden abordarse por separado;
- La estructura como resultado, la cual debe soportar las funcionalidades para las cuales ha sido desarrollada o que motivaron su diseño y construcción con eficacia y eficiencia. No solo se debe tener en cuenta la característica dinámica de los sistemas en general al momento del diseño, sino también la disponibilidad de infraestructura para soportarla y lograr que todas las piezas se integren perfectamente para lograr la funcionalidad deseada;
- Cuestiones transversales, debido a las cuales el impacto de un cambio es difícil de anticipar, medir o determinar, por características propias de los sistemas en general y debido a la interrelación de las partes. Es por ello que la función del Arquitecto no está

perfectamente delimitada, sino que debe contar con conocimientos periféricos a su función que le permitan interrelacionarse con el resto de la organización para alcanzar su cometido.

- **¿Cómo** se diseña la Arquitectura de TI?: la Arquitectura de TI como tal, es una disciplina que fue ganándose un lugar a lo largo del tiempo en la misma medida que las tecnologías de la información fueron volviéndose cada vez más complejas, se fueron insertando cada vez con mayor profundidad en las cuestiones de la organización y por ende, el impacto de un cambio cada vez fue más difícil de determinar. A pesar que en un primer momento no existía como disciplina, en las organización y de la misma forma que la Arquitectura Empresarial, alguien siempre estaba haciendo tareas relacionadas con la Arquitectura de TI, aunque talvez no metodológica y/o formalmente, o con los mismos objetivos que hoy se conocen para ella.

Cómo se diseña la Arquitectura de TI no solamente tiene relación con la metodología que utilice el arquitecto para hacerlo, sino también con los requerimientos y alcances que le bajen desde la Arquitectura Empresarial desde distintas perspectivas según el framework adoptado, y las necesidades de negocio hacia abajo, las cuales serán satisfechas mediante la implementación de las soluciones de TI que se soporten.

El resultado es un meta modelo, un modelo lógico conceptual de cómo la Arquitectura de TI es compatible con el diseño de la Arquitectura Empresarial sobre la cual se apoya y por otro lado, una especificación, una descripción de como las soluciones basadas en esa Arquitectura de TI deben ser diseñadas, desarrolladas e implementadas.

Agregando información al gráfico anterior:

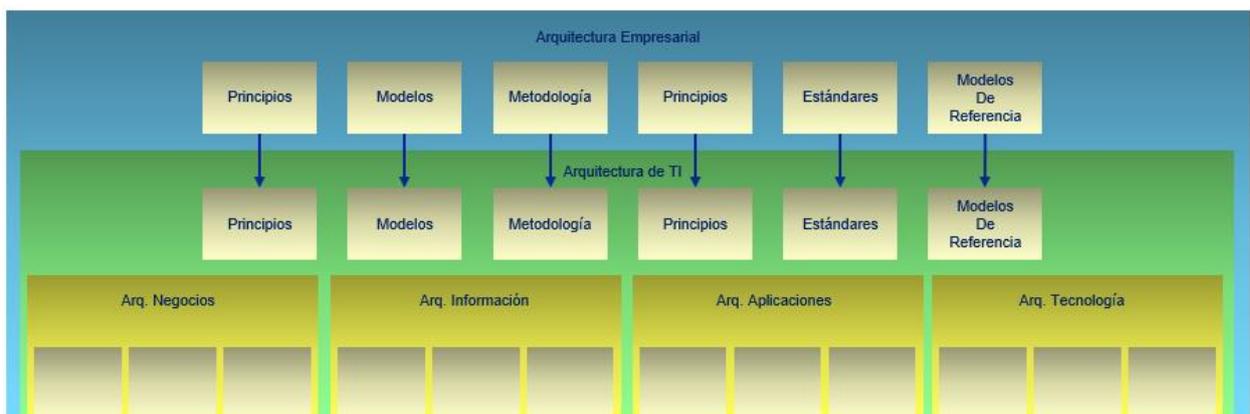


Fig. 11 – ¿Cómo se diseña la Arquitectura de TI?

- **¿Quién** hace Arquitectura de TI?: es obvio que quien realiza tareas relacionadas a la Arquitectura de TI es el Arquitecto de TI. Pero como se había expuesto anteriormente, los límites no están perfectamente identificados y es correcto que así sea. El rol de Arquitecto de TI requiere de habilidades interdisciplinarias y conocimientos periféricos que le permitan establecer una comunicación de igual a igual con los Arquitectos Empresariales y los Arquitectos de Negocios, de la Información, de Aplicaciones y de Tecnología. Adicionalmente, dependiendo del tipo de organización que se trate, podrían existir otras áreas de Arquitectura bajo el alcance de la



Arquitectura de TI o Arquitecturas al mismo nivel para otras áreas de la empresa, con las cuales el Arquitecto necesita relacionarse.

Es por ello que el Arquitecto de TI o el grupo de personas que conformen la función debe tener excelentes habilidades de comunicación, abstracción, gestión, negociación e influencias para llevar adelante sus tareas. De hecho, en muchas organizaciones el rol del Arquitecto de TI no es una función a tiempo completo, y una vez definida la Arquitectura de TI, solo se conforma el equipo de trabajo para resolver cuestiones que afecten al diseño o rediseño, construcción, administración o cambios de los modelos primitivos diseñados originalmente, o cuando la dinámica del negocio lo exija.

Existe mucho material respecto del alcance específico y bien determinado del rol de Arquitecto de TI, y otros no tanto. Sin embargo y a los fines del presente trabajo, se entenderá como Arquitecto de TI a la persona o conjunto de personas que llevan adelante el diseño o rediseño, construcción, administración o cambios de los modelos primitivos de Arquitectura de TI.

- **¿Porqué** se necesita la Arquitectura de TI?: una Arquitectura Empresarial eficiente y exitosa arrastra a una Arquitectura de TI eficiente y exitosa, y por correspondencia directa, a cualquier Arquitectura basadas en ellas, en una especie de efecto dominó, lo cual confiere una ventaja estratégica a la organización que se aplique. Por el contrario, la falta de Principios, Modelos, Metodologías, Marcos de Referencia y estándares, consecuentemente conducen a la ineficiencia, la falta de control, falta de flexibilidad para amortiguar los cambios del contexto y por ende a la pérdida de oportunidades de negocio, las cuales en el peor de los casos, lleva al fracaso de la empresa.

Para las organizaciones con más tiempo en el mercado, la Arquitectura heredada y por ende, las soluciones basadas en ellas constituyen una compleja telaraña de sistemas desordenados, ineficientes, no estandarizados y poco optimizados, difíciles y costosos de mantener y cambiar, pero que por contraposición, su sustitución amenaza la vida de la organización.

Las motivaciones para la Arquitectura de TI pueden variar de organización en organización. De entre ellas se puede destacar que permite:

- Comunicar el diseño de alto nivel;
 - Proveer el contexto sobre el cual las soluciones deberán ser desplegadas;
 - Correlacionar objetivos con requerimientos y resultados;
 - Fácil adecuación a cambios e integración de elementos nuevos;
 - Estandarización;
 - Minimizar costos de propiedad y optimizar retornos de las inversiones;
 - Modularización de las soluciones;
 - Reutilización y Flexibilidad de componentes.
- **¿Dónde** encaja la Arquitectura de TI?: La Arquitectura de TI alcanza a todos los niveles de la organización en la misma medida en la cual las tecnologías de la información alcancen a sus procesos. Es por eso habitual confundir en las organizaciones tecnológicamente avanzadas a la Arquitectura de TI como la estructura de la cual depende el funcionamiento de la organización. Muy lejos de ello, la Arquitectura de TI define los modelos primitivos sobre los cuales construir



las soluciones de TI que darán soporte a la operación de los negocios. Cuando erróneamente se sostiene que el funcionamiento de la organización depende de la Arquitectura de TI, en realidad lo que se pretende significar no es que la organización funcione o no de acuerdo al funcionamiento o no de la misma, sino a las implementaciones que sobre ella se realicen.

De estas afirmaciones se pueden concluir al menos tres cuestiones:

- Una Arquitectura Empresarial mal definida o no adecuada al tipo de organización o desajustada a los negocios que llevará a cabo, invariablemente conducirán a un mal diseño de la Arquitectura de TI. Aún y cuando los problemas en el diseño de la Arquitectura Empresarial sean absorbidos en mayor o menor medida por la Arquitectura de TI, estos desajustes tarde o temprano pondrán en riesgo el funcionamiento de la organización o de los procesos de negocios alcanzados, o al menos serán poco eficaces e ineficientes;
 - De igual manera, una Arquitectura de TI mal diseñada o no adecuada al tipo de organización o procesos que de soporte, invariablemente desembocará en malas implementaciones que pondrán en riesgo el funcionamiento de la organización o de los procesos de negocios afectados;
 - Por simple asociación, las implementaciones sobre una Arquitectura de TI mal diseñada o ineficiente conducirán a soluciones ineficientes, mal diseñadas y difíciles de mantener o ajustar.
- **¿Cuándo** trabajar sobre Arquitectura de TI?: En este punto es preciso detenerse específicamente sobre las motivaciones mencionadas anteriormente que conllevan en sí mismas las necesidades por las cuales se diseña, construye, administra y se cambia una Arquitectura de TI, aunque se podrían generalizar las mismas a cualquier otro tipo de Arquitectura. En un extremo, para la organización recién constituida y luego de definidos los procesos de negocios y cuestiones primitivas que permitirán la operación de los procesos de negocios es necesario diseñar y construir la Arquitectura de TI basadas en las mismas. Sin embargo se había comentado que la Arquitectura de TI alcanzaba al menos cuatro arquitecturas:
 - Negocios;
 - Aplicaciones;
 - Tecnología;
 - Información;

con una fuerte interdependencia entre ellas. Si bien el ciclo de vida puede cambiar de estándar en estándar, lo ideal es que la organización defina el ciclo de vida de la arquitectura en base a sus propias necesidades, tomando los mejores elementos de cada una de las disponibles y ajustándolos a sus requerimientos.

Sin embargo, un elemento en común es que la Arquitectura de TI se interrelaciona tanto con con la Arquitectura Empresarial como con las arquitecturas mencionadas anteriormente.

Replanteando el gráfico anterior:

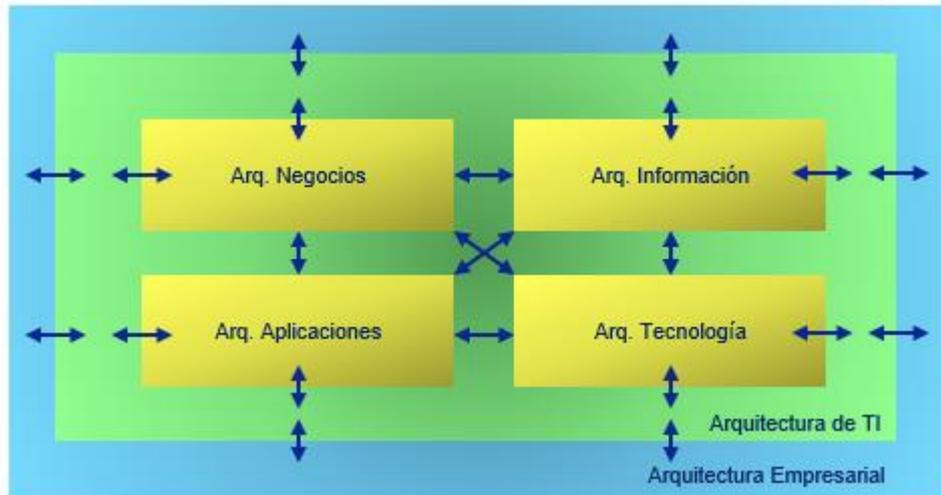


Fig. 12 – ¿Cuándo trabajar sobre la Arquitectura de TI?: Interrelaciones

Entonces, las tareas relacionadas con la Arquitectura de TI comienzan desde el momento mismo en el cual se termina de definir la Arquitectura Empresarial, con lo cual se define el momento de inicio del ciclo de vida de su proceso.

Adicionalmente es necesario tener una visión aproximada de la Arquitectura de Negocios que se pretende. Es por ello que mucha bibliografía indica como inicio del ciclo de vida justamente, a los negocios. En el caso particular en estudio, se agregará como particularidad relacionada, los casos en la cual la Arquitectura es heredada y se debe realizar un trabajo minucioso para transicionar de un modelo existente, tal vez empírico y basado en necesidades puntuales a un modelo estratégico.

Aplicaciones de Misión Crítica

Se puede definir a las aplicaciones de Misión Crítica como aquellas necesarias para la supervivencia y/o continuidad de la operación de los negocios de la organización. Cuando una aplicación considerada de misión crítica falla, interrumpe su ejecución o es detenida por otros motivos, significa que de alguna forma los negocios se verán impactados, generando costos de no operación muchas veces no contemplados en los presupuestos, pérdidas económicas relacionadas, caídas de índices de productividad etc., cuando no inconvenientes relacionados como por ejemplo problemas sindicales o de suspensión de actividades.

Si bien las aplicaciones de Misión Crítica podrían encajar dentro de lo subjetivo, limitaremos la definición a lo mencionado en el párrafo anterior.

Se puede agregar además, que con mayores o menos variantes el ciclo de vida de las aplicaciones, sean de misión crítica o no, este contempla fases tales como la captura de requisitos, el análisis, el diseño, las pruebas y la puesta en producción para el caso particular del Proceso Unificado de Desarrollo de Software, pero podrían aplicarse otros procesos o más aún, una combinación de ellos. En cada una de estas etapas y de acuerdo a la complejidad y criticidad de las necesidades de negocio que se pretendan satisfacer, el proceso deberá estar ajustado a un Modelo de Arquitectura Empresarial y de TI predefinido. En la práctica



significa que un Arquitecto de Aplicaciones no podrá diseñar una solución basada en una tecnología de base de datos que no sea estándar para la compañía o que no tenga soporte del fabricante, por ejemplo.

Adicionalmente, y también en distintas fases del ciclo de vida del proceso, se podría necesitar la redefinición de Modelos Primitivos, estén relacionados estos al ámbito de la Arquitectura Empresarial para el caso de modelado de negocios nuevos o adaptaciones profundas en los existentes, o al ámbito de la Arquitectura de TI, bien como consecuencia de ajustes a los primeros, o por otros cambios específicos como podría ser un cambio de estándar de base de datos, para seguir con el ejemplo del párrafo anterior.

Todas estas tareas tienen que ver con lo que se mencionaba dentro de las motivaciones para el diseño, rediseño, administración y gestión de Modelos Arquitectónicos, a esta altura ya entendidos como las tareas principales de la disciplina.

En particular a lo que ocupa a este trabajo y como se desarrollará más adelante, las Aplicaciones de Misión Crítica o parte de las funcionalidades que las compongan y que se consideren necesarias para la operación continua del negocio, deberán ser debidamente identificadas, dimensionadas en materia de costos de no operación, retrabajo, y otras cuestiones no económicas y puestas en consideración a la hora proveer alternativas para los casos de no disponibilidad. Al mismo tiempo, los arquitectos deberán determinar los puntos posibles de falla o los componentes cuya disponibilidad es estrictamente necesaria para operación continua, a los fines de minimizar la posibilidad de fallos o minimizar la no disponibilidad de los mismos, en una tarea interdisciplinaria y colaborativa que requiere el esfuerzo de todos los miembros del equipo.

Conclusiones

La Arquitectura de TI es una disciplina que enrola a otras Arquitecturas con las cuales se interrelaciona, y que se inserta dentro de la Arquitectura Empresarial como el conjunto de tareas destinadas al diseño, rediseño, construcción, administración y cambio de Modelos Arquitectónicos Primitivos, a los fines de entre otros objetivos y metas, alinear la estrategia de TI a la empresarial, estandarizar soluciones tecnológicas, reducir costos, lograr la flexibilidad y la adecuación a los cambios de las aplicaciones basadas en esta arquitectura, minimizar las probabilidades de fallo etc.

Las aplicaciones de Misión Crítica o las funcionalidades de ellas que se consideren crítica para la operación continua de los negocios de la organización se construyen sobre la base de la Arquitectura de TI definida anteriormente, con el fin de dar respuesta a los requerimientos de la empresa para la obtención de sus objetivos y metas.



Capítulo

5

ISO/IEEE 42010

En este módulo se da un rápido repaso a la formulación del estándar ISO/IEEE 42010 y sus implicancias en relación a la arquitectura bajo análisis, dando un acercamiento a sus fundamentos, objetivos y alcances, y particularmente enfocándose en el draft P1471 en cuanto a los lineamientos para la descripción de arquitecturas y en especial la utilización de UML como ADL.

IEEE 42010

El estándar “IEEE 42010 Systems And Software engineering – Architecture description” está enfocado en la creación, análisis y mantenimiento de la arquitectura de sistemas mediante la utilización de descripciones de la arquitectura. Provee una ontología base para una correcta especificación de:

- Descripciones de arquitectura;
- Puntos de vista de la arquitectura;
- Modelos de referencia de la arquitectura;
- Lenguajes de Descripción de la arquitectura,

Que pueden ser utilizadas para la práctica coherente en el desarrollo de arquitecturas en el contexto del ciclo de vida y sus procesos.

Fundamentos

IEEE 42010 entiende a un sistema como aquellas entidades cuyas arquitecturas son de interés, abarcando:

- Sistemas, de acuerdo a lo que define la norma ISO/IEC 15288: sistemas compuestos por hardware, software, datos, personas, procesos, procedimientos, facilidades, materiales y otras entidades;
- Productos y servicios de software, como son descriptos por la norma ISO/IEC 12207;
- Sistemas con uso intensivo de software, como lo describe la norma IEEE 1471.

Con una serie de propósitos determinados basados en un interés en particular, los stakeholders tienen interés en un sistema. Cada propósito es un tipo específico de interés. El sistema está inmerso en un contexto que determina las influencias sobre el mismo a lo largo de su ciclo de vida, incluyendo sus mutuas interrelaciones. El contexto puede contener a otros sistemas. Estos sistemas exponen una arquitectura determinada la cual puede estar descrita por alguna descripción de arquitectura.

En estos aspectos la cardinalidad es muy importante y por tal motivo debe ser considerada. Inclusive, una organización puede desenvolverse en más de un contexto determinado por distintas características que le imprimen su condición de contexto, al menos para la organización que lo considera.

La figura siguiente representa conceptos claves relacionados con la arquitectura de sistemas y el contexto en el cual se desarrolla la práctica de la descripción de la arquitectura:

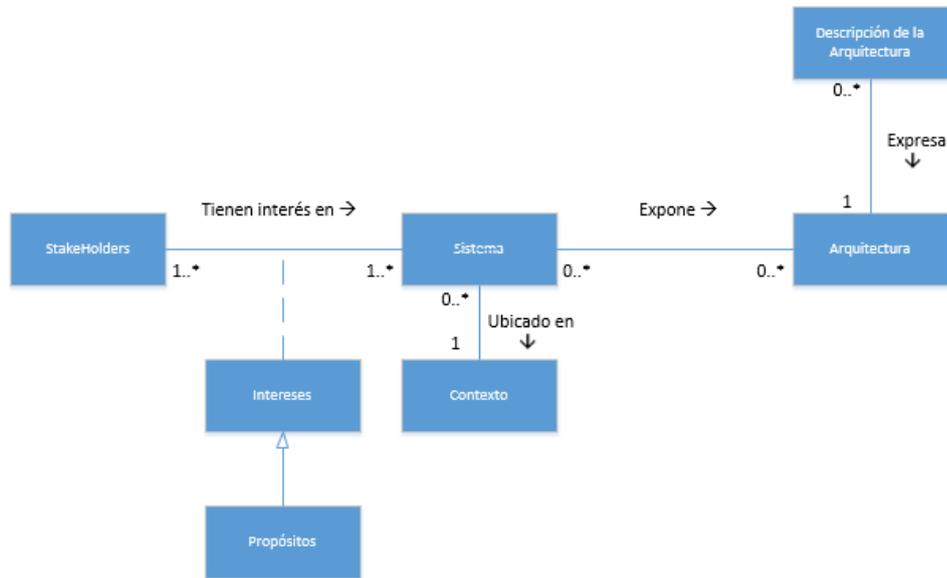


Fig. 13 – IEEE 42010: Contexto de la descripción de Arquitectura

Modelo Conceptual de la Descripción de Arquitecturas

La descripción de la arquitectura el producto del trabajo de los arquitectos de negocios, de información, aplicación o de tecnología.

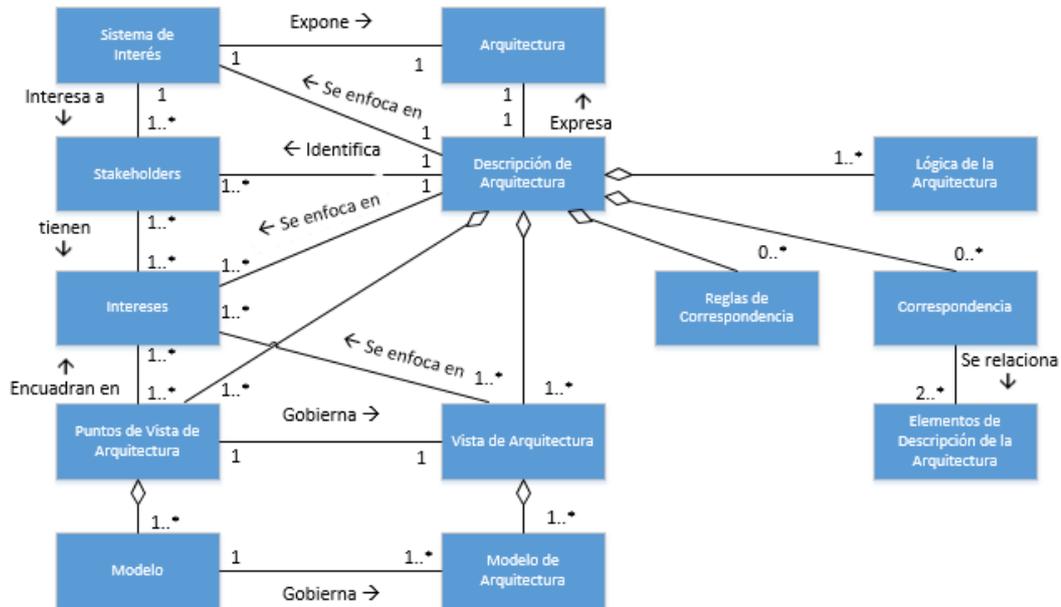


Fig. 14 – IEEE 42010: Modelo Conceptual de la Descripción de Arquitecturas



En el gráfico anterior se representan los conceptos relacionados con la práctica de la descripción de arquitecturas cuando se aplica este estándar, de tal forma de producir una descripción de la arquitectura que represente una arquitectura para el sistema bajo análisis.

Es necesario entender las diferencias entre arquitectura de un sistema y la descripción de esa arquitectura. La descripción es un producto de trabajo, una concepción abstracta compuesta de conceptos y propiedades. Los estándares definen ciertas convenciones y requerimientos para las descripciones de arquitectura, pero no para la arquitectura, sistemas o contextos. Tampoco se definen técnicas, notación, procesos o métodos que deban ser utilizados para la descripción, aunque el borrador IEEE P1471 propone algunas prácticas recomendables para ello.

En base a lo expuesto en la figura anterior, los stakeholders tienen intereses centrados en el sistema bajo análisis, el cual está inserto en un contexto determinado. Estos intereses van apareciendo a lo largo del ciclo de vida en base a las necesidades y requerimientos del sistema en diseño, decisiones de implementación, operación y otras cuestiones. Estos intereses se pueden manifestar como:

- relaciones entre uno o más stakeholders,
- necesidades,
- objetivos,
- expectativas,
- responsabilidades;
- requerimientos;
- restricciones;
- dependencias;
- atributos de calidad;
- decisiones de arquitectura;
- riesgos y otras cuestiones relacionadas al sistema en particular.

Una descripción de arquitectura incluye una o más vistas de la arquitectura. Una vista enfoca a uno o más intereses que los stakeholders puedan tener en relación al sistema.

Una vista de la arquitectura representa la arquitectura del sistema que se está construyendo, de acuerdo con un punto de vista. Un punto de vista enmarca a uno o más intereses, y a su vez uno o más intereses pueden estar enfocado en más de un punto de vista. Una vista es gobernada por su punto de vista, el cual establece las convenciones necesarias para la construcción, interpretación y análisis de la vista. Finalmente, una vista de la arquitectura está compuesta por uno o más modelos arquitectónicos., cada uno de los cuales utilizan convenciones para el modelado.

Un elemento de descripción de la arquitectura es un elemento primitivo. La correspondencia define una relación entre al menos dos elementos que componen la descripción de la arquitectura. Las reglas de correspondencia gobiernan esta correspondencia entre elementos de la arquitectura. La lógica de la arquitectura registra la explicación, justificación y razones acerca de las decisiones que se tomen en relación a la arquitectura adoptada.

Modelo Conceptual del Framework para la Arquitectura

De acuerdo al IEEE 42010 un framework para la arquitectura establece criterios comunes para la práctica, interpretación, análisis y uso de una o más descripciones de arquitectura, dentro de un dominio de aplicación en particular.

El uso de un framework para la arquitectura incluye:

- la creación de descripciones de arquitecturas;
- el desarrollo de herramientas y métodos para el modelado de arquitecturas;
- el establecimiento de procesos que faciliten la comunicación y el entendimiento entre las partes intervinientes y su interrelación a lo largo y ancho de la organización.

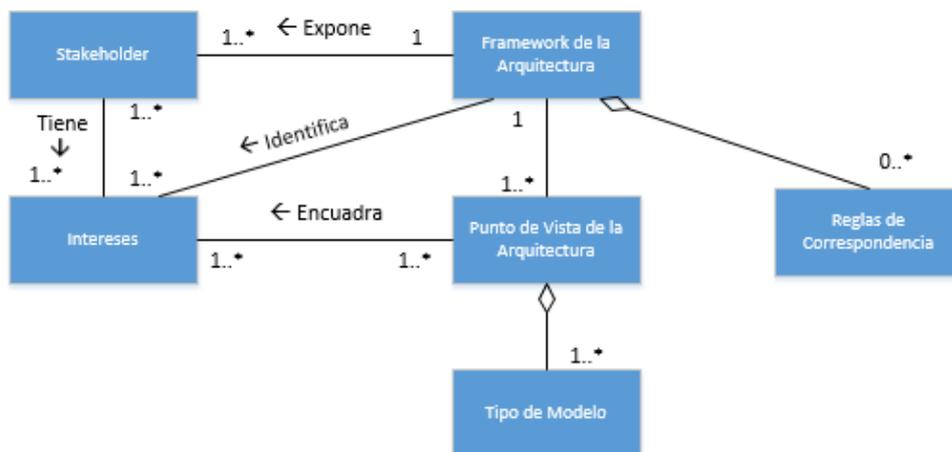


Fig. 15 – IEEE 42010: Modelo Conceptual de la Descripción de Arquitecturas

Lenguaje para la Descripción de Arquitecturas – ADL

Los lenguajes de descripción de arquitectura o ADLs por sus siglas en inglés son utilizados por muchas disciplinas tales como la Ingeniería de Sistemas, de Software, Arquitectura Empresarial y otras Ingenierías y Arquitecturas relacionadas con el uso intensivo de tecnologías de la información.

La norma “ISO/IEC/IEEE 42010:2011, Systems and software engineering — Architecture description” define a los lenguajes de descripción de arquitecturas como “cualquier forma de expresión para usar en la descripción de arquitecturas” y particularmente, establece “Prácticas Recomendadas para la Descripción de Arquitecturas” en su borrador IEEE P1471, el cual establece un marco conceptual sobre el cual fundamentar lenguajes de Descripción de Arquitecturas, y en especial UML.

Los objetivos de IEEE P1471 son:

- Lograr una significado común acerca de lo que es “arquitectura” como disciplina aplicable a sistemas con uso intensivo de tecnologías de la información;
- Establecer un marco conceptual y un vocabulario común para facilitar el entendimiento en cuestiones de arquitectura de sistemas;



- Identificar y promover prácticas de arquitectura estándar;
- Facilitar la evolución de las prácticas relacionadas a la tecnología.

IEEE P1471 es:

- Un conjunto de definiciones para la descripción, vistas y puntos de vista de la arquitectura;
- Marco conceptual que establece los términos de uso en el contexto de un sistema en construcción;
- Un conjunto de requerimientos para la descripción arquitectónica de un sistema.

De acuerdo a IEEE P1471, la “Descripción de la Arquitectura” es cualquier colección de artefactos que pretenda describir y documentar la arquitectura de un sistema con uso intensivo de tecnologías de la información. No se especifica un formato o medio para la descripción, sino los requerimientos mínimos indispensables que una descripción de arquitectura debe tener en base a prácticas comunes de la industria. Por ello, IEEE P1471 está diseñado para ser independiente de cualquier técnica o metodología y podría ser aplicable a cualquier modelo de referencia, proyecto, organización, procesos, métodos, prácticas o herramientas en particular, lo que le imprime cualidades importantes a la hora de compartir y entender diseños.

La descripción de una arquitectura podría estar organizada en una o más vistas arquitectónicas, cada una de las cuales podrían expresar uno o más puntos de vista. La cantidad de vistas y puntos de vistas por cada vista podrían variar dependiendo la metodología que se utilice.

Una vista es una colección de modelos que representan a un sistema como un todo en relación con ciertos aspectos predefinidos o determinados. Un punto de vista captura las reglas para la construcción y análisis de una vista en particular.

Resumen

El estándar ISO/IEEE 42010 Systems and software engineering — Architecture description, define una serie de normas que fundamentan las bases para la descripción de la arquitectura, alineando con mayor o menor similitud los conceptos desarrollados para otras disciplinas de la arquitectura, pero enfocadas especialmente en el campo de la Arquitectura de TI y sus ramas arquitectónicas dependientes.

Finalmente, el draft P1471 propone a UML como lenguaje para la descripción de arquitecturas, por estar ampliamente extendida su utilización y ser generalmente entendido y aceptado por la comunidad de TI.



Capítulo

6

Arquitectura Propuesta

Este capítulo profundiza en los aspectos necesarios para entender la arquitectura que se propone como objeto de estudio, avanzando en la descripción de los elementos que la componen en un orden progresivo que permita su comprensión en detalle. Se detallan los elementos que en forma separada o interrelacionada configuran las funcionalidades para las cuales la misma fue pensada, como base para un mayor entendimiento de los temas que siguen.

Introducción

Si bien puede no estar explícitamente mencionado entre lo que a conceptos previos respecta, la Arquitectura Empresarial es una disciplina que necesariamente tiene que enlazar a otras a los fines de obtener una visión estratégica de los negocios y a especificar y/o diseñar la forma en la cual los mismos serán operados.

Siempre relacionado a la Organización en general, es importante en este punto entender que a diferencia de otras áreas alcanzadas por los distintos sabores de Arquitecturas que podamos encontrar, como podrían ser Arquitectura de Soluciones, Arquitectura de TI, Arquitectura de Negocios etc., el dominio de la Arquitectura Empresarial alcanza a la especificación y priorización de las necesidades de negocio de la organización en su totalidad, y engloba otras disciplinas y metodologías relacionados, las cuales sin entrar en mayores detalles y/o excepciones, generalmente están enfocadas en la implementación de necesidades específicas. Estas tareas SIEMPRE deberán estar alineadas bajo lo que líneas arriba se describía como Arquitectura Empresarial.

Si bien en mayor o menor medida las organizaciones venían realizando tareas relacionadas con lo que hoy conocemos como Arquitectura Empresarial, en realidad esta se comenzó a formalizar como disciplina a mediados de los 80, y su utilización se aceleró en la misma medida en la cual las soluciones de TI comenzaron a expandirse a todos los negocios, una de las razones de su estrecha vinculación. Básicamente, la Arquitectura Empresarial como disciplina formal apareció en un momento en el cual alguna forma de Arquitectura Empresarial ya estaba definida y explícitamente implantada en la organización.

Algo parecido sucedió con la Arquitectura de TI en general, la cual fue cobrando importancia en la medida en la cual las soluciones de TI no solamente alcanzaban a más y más negocios, sino que además, se comenzaban a tornar críticas para la operación del negocio y por ende, para su éxito económico, fundamental para su supervivencia.

Pero al igual que en el caso de la Arquitectura Empresarial, la Arquitectura de TI (y otras disciplinas relacionadas) aparece en un momento en el cual cierta forma de Arquitectura de TI ya estaba implantada en las organizaciones.

En un primer momento estas soluciones mencionadas en el párrafo anterior tenían un alcance más o menos regional, altamente distribuido, y difícilmente se aplicaban a procesos corporativos. La interrupción de



un servicio afectaba solo a ciertas áreas de la organización y consolidar información de la organización como un todo era una tarea extremadamente compleja que significaba un esfuerzo importante en recursos de todo tipo. Sin embargo, en la medida que la tecnología fue facilitando el acceso a servicios remotos mediante los avances en materia de comunicaciones, la consolidación en puntos de acceso único y data centers de categoría mundial, apropiadamente certificados y por ende especialmente preparados comenzó a ser un paso normal.

Fue necesario entonces empezar a pensar de otra forma y aunar distintas perspectivas de la empresa bajo una misma estrategia y visión, a los fines de optimizar los esfuerzos y enfocarlos en las necesidades que eran esenciales para los negocios, por las cuestiones que la organización considerara necesario tener en cuenta. Estas son las necesidades requeridas o servicios brindados que la organización considera críticos para su operación, rentabilidad, supervivencia etc. y que no puede darse el lujo de interrumpir su disponibilidad, sea a los clientes internos o externos.

En general, estas necesidades requeridas o servicios brindados cumplen con alguna combinación de las siguientes características:

- Requieren disponibilidad en todo momento o algún procedimiento de contingencia;
- Una interrupción no puede ser tolerada;
- Su operación generan rentabilidad o reduce costos asociados a su no operación;
- Es necesario para otros servicios;

Una interrupción de la operación:

- Inmediatamente reduce la rentabilidad;
- Automáticamente genera costos;
- Produce daños comerciales o de reputación de la empresa;

Gran parte de todo lo expuesto anteriormente, es razón por la cual la Arquitectura de TI no solamente debe alinearse a la estrategia de la organización a través de la Arquitectura Empresarial, sino que además debe estar estrechamente relacionada con las disciplinas relacionadas, a los fines de satisfacer las necesidades organizacionales. La Arquitectura Cliente-Servidor / Cliente Servidor apunta a satisfacer necesidades relacionadas a la disponibilidad de servicios considerados críticos. Se debe entender a esta propuesta como un modelo de referencia o framework para:

- estandarizar y simplificar el desarrollo,
- asegurar la disponibilidad de los servicios;
- reutilizar componentes;
- permitir escalar la solución, el servicio, la arquitectura etc. con mínimo impacto;
- integrar el resto de las soluciones;

El desarrollo de una solución de negocio que atienda una necesidad en particular puede llegar a ser una tarea muy compleja. Pero en el contexto de un diseño arquitectónico efectivo y alineado a estándares de TI y a la estrategia organizacional sus componentes pueden ser utilizados en ambientes heterogéneos, reutilizados para atender más de una necesidad, la integración será más transparente y su mantenimiento mucho más sencillo, dando como resultado funcionalidades que satisfacen los requerimientos del negocio de un forma más simple y efectiva.

El siguiente diagrama de bloques destaca la ubicación de la Arquitectura Cliente-Servidor / Cliente-Servidor que se propone dentro del contexto de la Arquitectura Empresarial:

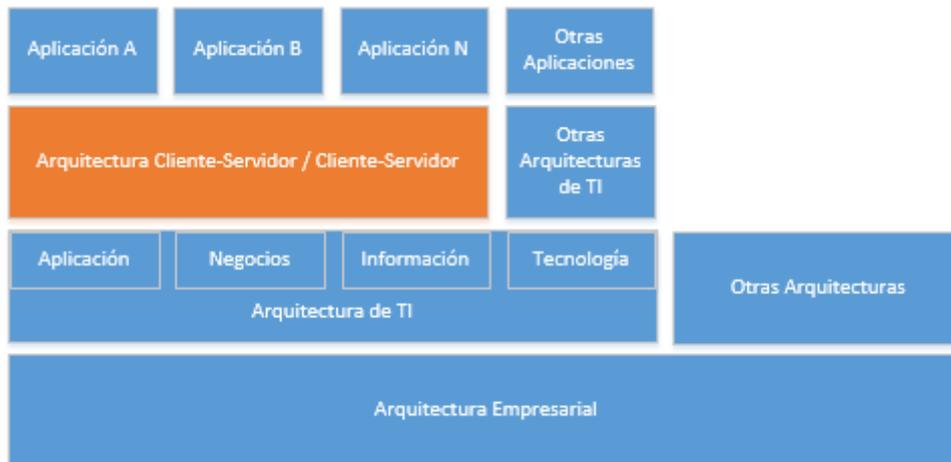


Fig. 16 – Arquitectura Cliente-Servidor / Cliente-Servidor en el contexto de Arquitectura Empresarial

Modelo de Capas

Bajo este contexto, es preciso entonces desagregar no solo como la arquitectura que se propone se enmarca dentro de lo que a la Arquitectura Empresarial alcanza, sino también como las soluciones que se desarrollen sobre la misma soportan, interactúan y se integran con funcionalidades ya existentes, y al mismo tiempo proporcionan capacidades que permitan operar los procesos de negocios alcanzados bajo situaciones de no disponibilidad de los servicios centrales o remotos, no disponibilidad que se pretende mitigar con soluciones basadas en la arquitectura objeto de este trabajo.

En un primer nivel de desagregación y con el objetivo de delimitar o modular las aplicaciones basadas en la arquitectura Cliente-Servidor / Cliente-Servidor, se definen al menos dos capas (en principio) lógicas sobre las cuales los elementos que compongan las soluciones que se apoyen sobre la misma serán desplegadas:

- Capa Local: que comprende todos los componentes de la solución alcanzables y disponibles en un área geográficamente delimitada por la capacidad de conexión o disponibilidad a los servicios que los mismos brindan. Si bien depende el ámbito de esta capa depende de varios factores como el alcance del proyecto, la habilidad del arquitecto, disponibilidad de infraestructura, puntos de falla a considerar etc., a modo de ejemplo, podría ser un segmento de red, un conjunto de redes lógicamente/físicamente conectadas, un data center de nivel 2 etc.;
- Capa Remota: que comprende todos los componentes de la solución no alcanzables localmente y no disponibles bajo situaciones de interrupción de los servicios.

Definidos los conceptos, queda más que claro que ambas capas podrían estar lógicas y/o físicamente (geográficamente) separadas, y por eso la salvedad anterior. Sin embargo y aún sin definir lo que en el marco de este trabajo se entenderá como componente, se hará solamente mención (por ahora) a “componente/s de comunicación” como medio precisamente, para la comunicación entre ambas capas.

Como aclaración, a esta altura y a los efectos de abstraer gráficamente la arquitectura que se pretende describir, se esquematiza la misma en módulos que paulatinamente se irán desagregando con el objetivo de profundizar cada vez con mayor grado de detalle cada uno de los elementos intervinientes, en concordancia con el marco metodológico prefijado y con base teórica en los modelos de referencia y estándares mencionados anteriormente.

Un esquema inicialmente basado en hexágonos, permitirá conceptualizar gráficamente la idea de un conjunto de elementos estrechamente relacionados entre sí, con el objetivo de llevar a cabo una funcionalidad determinada y prefijada con anterioridad, y con ciertas reglas que determinarán su comportamiento bajo distintas circunstancias. Finalmente, permitirá ir evolucionando el modelo hacia un diseño final, mediante el cual se pretende proporcionar la idea de un todo formado por partes interrelacionadas con fines primitivos y últimos, en pos de configurar un modelo consistente.

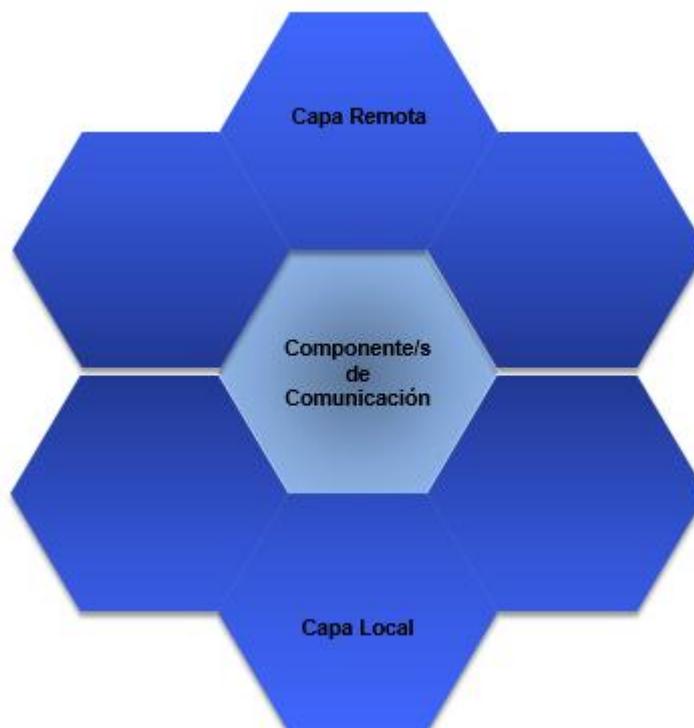


Fig. 17 – Arquitectura Cliente-Servidor / Cliente-Servidor: esquema de dos capas y componente/s de comunicación

Modos de Operación

Precisamente, la disponibilidad o no de los servicios cualquiera sea su causa, da lugar a que se definan dos modos de operación de la solución a la que se aplique:

- Modo Sincrónico: cuando los servicios locales en su totalidad funciona en línea con los servicios expuestos desde la capa remota, sincronizando la información generada localmente con su equivalente remoto, en una dirección, en otra, o en ambas;



- Modo Asíncrono: cuando se produce una no disponibilidad de algún servicio expuesto por la capa remota y el mismo es asumido por los servicios brindados por la capa local. En este punto es preciso al menos comentar situaciones que se deberán tener en cuenta:
 - Los servicios locales pueden ser consumidos por clientes locales;
 - Los servicios remotos pueden ser consumidos por clientes remotos;
 - Bajo una no disponibilidad de servicios locales los clientes locales pueden consumir servicios remotos;
 - Bajo una no disponibilidad de servicios remotos los clientes remotos pueden consumir servicios locales.

Estos dos conceptos sugieren o establecen un par de supuestos básicos:

- ambos modos de operación son excluyentes entre sí: la solución como tal podrá estar operando en modo síncrono o en modo asíncrono, pero no en ambos;
- la presencia de dos modos de operación excluyentes entre sí, supone la existencia una operación de cambio de un modo a otro y en cualquier dirección, que se define con el nombre de “cambio de modo de operación” o simplemente “switch”, por su acepción en inglés.

De la misma forma en la cual el concepto de Modo asíncrono da por entendido operar desconectado de “algo”, la idea de sincronismo implícitamente indica su existencia. Ambos mundos (que previamente se habían definido como capas lógicas), necesariamente tiene que estar conectados por algo que por el momento se había definido como “componente/s de comunicación”.

Bajo el esquema anterior, se puede entender implícitamente que operando en modo síncrono se supone el correcto funcionamiento y disponibilidad de los componentes de comunicación, y por ende, como medio de alcanzar los servicios brindados por los componentes ubicados en la capa remota, la disponibilidad de esta última.

Análogamente, se puede entender implícitamente que operando en modo asíncrono se supone un funcionamiento defectuoso y/o no disponibilidad de los componentes de conexión, y por ende, como medio de alcanzar los servicios brindados por los componentes ubicados en la capa remota, la disponibilidad de esta última.

El cambio de modo de operación puede ser invariablemente y debido a las circunstancias que se presenten, en uno de dos sentidos:

- Modo síncrono → Modo asíncrono: supone la interrupción de un servicio o la no disponibilidad de un componente esencial para la prestación del mismo, sea por cuestiones técnicas o no;
- Modo asíncrono → Modo síncrono: supone la recuperación de un servicio o la disponibilidad de un componente esencial para la prestación del mismo, sea por cuestiones técnicas o no.



Como sea, la operación en cualquiera de los dos modos deberá alcanzar al menos a las funcionalidades y/o procesos que delimitan a las aplicaciones de misión crítica y que permiten la operación del negocio aún bajo condiciones de total aislamiento.

Perspectiva

Definiendo a los clientes (sean remotos o no) como componentes que consumen servicios, bajo la combinación de posibilidades o circunstancias mencionadas, es necesario aclarar una cuestión de perspectiva a los simples efectos de que sea necesaria su consideración bajo las circunstancias en las cuales se diseñe la solución:

- los clientes locales que consuman servicios locales, serán clientes locales para los servicios brindados por componentes en la capa local;
- de igual forma, los clientes locales que consuman servicios remotos, serán clientes remotos para los servicios brindados por componentes ubicados en la capa remota.

Idéntica reciprocidad merecen los componentes ubicados en una capa local respecto de los componentes ubicados en la capa remota y viceversa; y extensivamente, la misma consideración merecen los servicios brindados por los mismos.

En definitiva, cuando se hace relación a un cliente y no se especifique ningún detalle adicional, invariablemente se podría estar haciendo referencia a un componente o conjunto de componentes, servicio o conjunto de servicios, otras aplicaciones y sistemas, usuarios, etc. que consumen de alguna forma algún servicio. Adicionalmente y como ya se ha visto, un cliente puede ser local o remoto dependiendo de la perspectiva expuesta anteriormente.

De igual manera se define a un servidor como un componente o conjunto de componentes que proporcionan servicios a clientes (remotos o no) en base a peticiones de servicios realizados por estos, peticiones que son procesadas por el servidor en base a cierta lógica predeterminada, devolviendo los resultados a quien corresponda.

Componentes, Servicios y Estados

Hasta ahora se han mencionado un par de conceptos dando por sobreentendido su significado y función. Sin embargo, a esta altura y con algunas cuestiones ya aclaradas se pueden abordar con más detalles:

- **Componente:** si bien la palabra componente tiene un sin número de acepciones, en el contexto del objeto bajo estudio en el presente trabajo se entiende como tal a todo elemento lógico o físico que encapsula una o más funcionalidades. Estas funcionalidades pueden ser entendidas por sí solas o por en su conjunto como uno o más servicios.

Además los componentes cumplen con ciertas características que los identifican claramente:



- Son modulares y cohesivos, lo cual significa que varios componentes pueden trabajar en conjunto, secuencialmente o en paralelo, para atender distintas fases de un mismo proceso o el proceso en su conjunto, actuando como un todo;
- Tienen propiedades, relaciones y restricciones que condicionan, limitan o expanden su funcionamiento;
- Adicionalmente, podrían estar dotados de cierta lógica (de negocios o no) que le agregan inteligencia.

En este trabajo y cuando sea necesario, se hará referencia a los componentes en forma general nombrándolos simplemente como “componente/s”, o en forma específica agregando una clasificación que agregue información respecto de su función, como por ejemplo “componente de comunicación.

- Servicio: de igual forma para el contexto del objeto bajo estudio en el presente trabajo, se entiende como servicio al conjunto de funcionalidades que un componente o conjunto de componentes brindan o exponen para su consumo a uno o más clientes, sean estos remotos o no. Las características de un servicio pueden estar definidas por:
 - Los requerimientos funcionales y no funcionales de la solución a implementar;
 - Las necesidades de negocio que se pretende satisfacer;
 - Las restricciones que limitan su funcionamiento;
 - Cierta lógica de negocio necesaria para alcanzar sus objetivos;
 - Ciertas características de seguridad que garantiza que el servicio sea consumido por clientes autorizados bajo determinadas circunstancias.

A los fines de tener en cuenta ciertos aspectos que podrían presentarse en base a las características y particularidades de las soluciones que se diseñen y construyan, los servicios merecen una clasificación en base a su localización:

- Servicios Locales: son los servicios configurados por componentes locales, solo expuestos por la capa local en forma autónoma, independientemente que puedan ser consumidos por clientes locales, remotos o combinación de ellos y bajo Modos de Operación sincrónico o asincrónico;
- Servicios Remotos: son los servicios configurados por componentes remotos, solo expuestos por la capa remota en forma autónoma, independientemente que puedan ser consumidos por clientes locales, remotos o combinación de ellos. Desde la perspectiva del cliente local, solo pueden ser consumidos bajo situaciones de disponibilidad de los componentes de comunicación, generalmente en Modos de Operación sincrónico. Bajo modo de operación asincrónico se supone la posibilidad de no disponibilidad de los mismos;
- Servicios Mixtos: son los servicios configurados mediante una combinación de componentes locales y remotos. Debido a la existencia de al menos un componente remoto, merecen un tratamiento similar al de los servicios remotos, ya que la no disponibilidad de al menos un componente remoto podría suponer la no disponibilidad del servicio o de al menos una de sus funcionalidades.

Al hablar de Componentes y Servicios, es necesario tener en cuenta el concepto de estado, tanto para el correcto funcionamiento de un componente, como para el del/los servicio/s que configuran. A los fines del trabajo, se entiende como estado, a la propiedad de un componente o servicio (o conjunto de ellos), en base a la cual se establece un determinado comportamiento del mismo.

Avanzando en el modelo, en el siguiente diagrama se podrán observar como las capas están conectadas mediante componentes de conexión, y la forma en la cual uno o más componentes de servicios (numerados del 1 al 11) configuran diferentes servicios locales y remotos (nombrados desde la letra A a la G), siempre teniendo en cuenta el concepto de perspectiva:

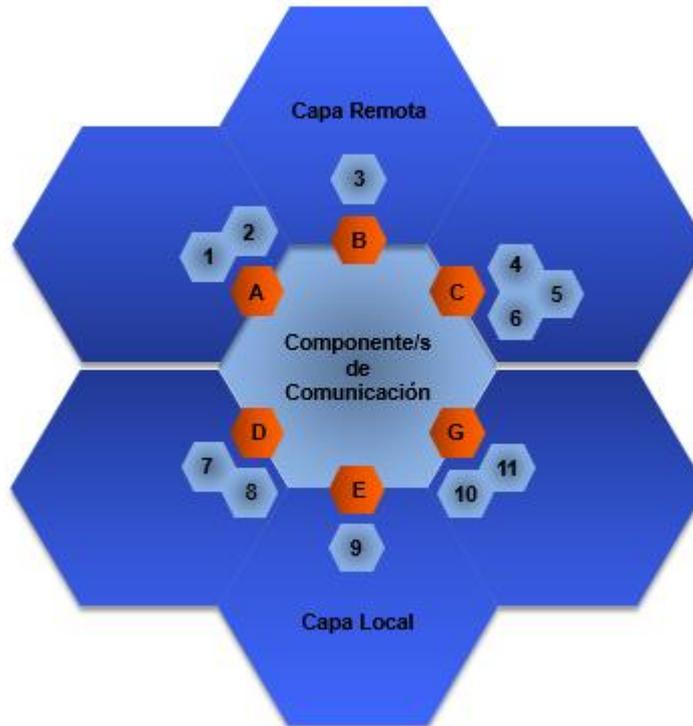


Fig. 18 – Arquitectura Cliente-Servidor / Cliente-Servidor: esquema de dos capas, componente/s de conexión, y componentes de servicios

Modelo de N Capas

Un elemento más que agrega complejidad al diseño tiene que ver con que podrían definirse más de dos capas. Si bien esto agrega algo más de trabajo a la hora del análisis, el tratamiento para cada capa es el mismo. Inclusive y dependiendo de las necesidades o características de las aplicaciones de misión crítica que se quiera atender, la capa local puede definirse o no. En caso que se defina, y por las cuestiones de perspectiva mencionadas anteriormente, siempre será una y solo una sola. A los fines de este trabajo y también por cuestiones de fijar una perspectiva desde el punto de vista del cliente de aplicaciones de misión crítica, se asume que siempre habrá una capa local.

Para graficar el hecho de nombrar a la capa local como tal, es objetivo de este trabajo dar soporte a las aplicaciones de misión crítica de tal manera de que filiales, oficinas o departamentos geográficamente distribuidos puedan continuar operando bajo cualquier circunstancia, aún aislados de las oficinas u

aplicaciones centrales. Esto abre el juego a un sinnúmero de combinaciones de elementos donde todo depende de la complejidad lógica de los procesos de negocio, los medios tecnológicos con los cuales se cuentan y la habilidad e imaginación del arquitecto, tal vez el recurso más valioso.

Esta complejidad lógica en los modelos de n capas está directamente asociada a la complejidad lógica de la solución a implementar y es función de:

- las necesidades, requerimientos y expectativas de los clientes que se quiera dar cobertura;
- direccionalidad de la información y posibilidades técnicas de implementarla;
- características propias de la solución.

En la figura que sigue, se puede observar como el “Esquema A” representa un modelo de una capa local y dos capas remotas, mientras que el “Esquema B” representa un modelo de N capas remotas. Para cualquiera de los dos casos, los servicios configurados mediante los componentes que lo forman, son expuestos y accedidos (o no).

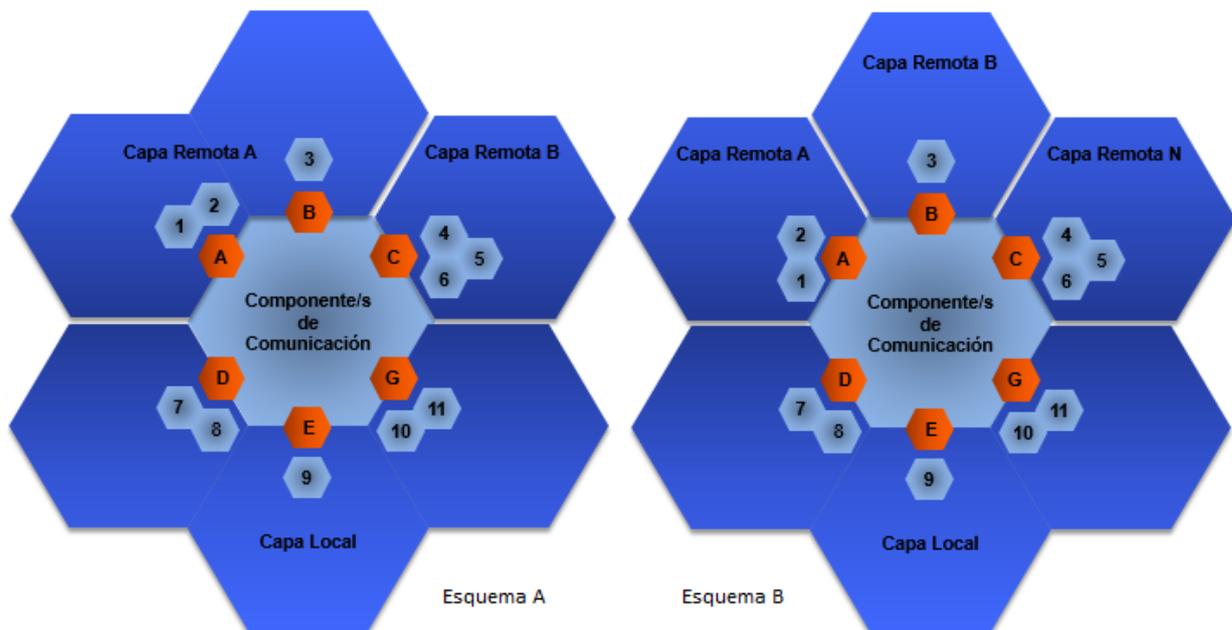


Fig. 19 – Arquitectura Cliente-Servidor / Cliente-Servidor: esquema de n capas, componente/s de conexión, componentes y servicios

Direccionalidad de la Información

Esto desencadena una lista cada vez más extensa de combinaciones de elementos y conceptos, estrechamente relacionado con los factores mencionados anteriormente, lo que agrega complejidad por un lado, pero flexibilidad para atender los requerimientos por el otro.

De igual forma, esto sugiere la idea de direccionalidad de la información, más allá de los métodos o componentes para alcanzarlo. Supone un flujo continuo de datos con tres posibilidades, siempre teniendo en cuenta un origen y un destino de la misma y desde la perspectiva del cliente local:

- Capa Local → Capa Remota: información generada localmente necesita estar disponible remotamente. La información solo se actualiza localmente;
- Capa Remota → Capa Local: información generada remotamente necesita estar disponible localmente. La información solo se actualiza remotamente;
- Capa Local ↔ Capa Remota: la información puede ser generada, actualizada o eliminada tanto en forma local como en forma remota. Es la más compleja de las situaciones y requiere un análisis

Adicionalmente se puede (o debe) considerar otra clasificación relacionada con la información que contrariamente a lo expuesto, solamente hace a los efectos de brindar servicios localmente, sea desde la perspectiva de un usuario local o de un cliente remoto:

- Información solo necesaria en forma local: los clientes remotos, sean usuarios, sistemas, servicios u otros componentes no necesitan ni necesitarán esa información;
- Información solo necesaria en forma remota: los clientes locales, sean usuarios, sistemas, servicios y otros componentes no necesitan ni necesitarán esa información.

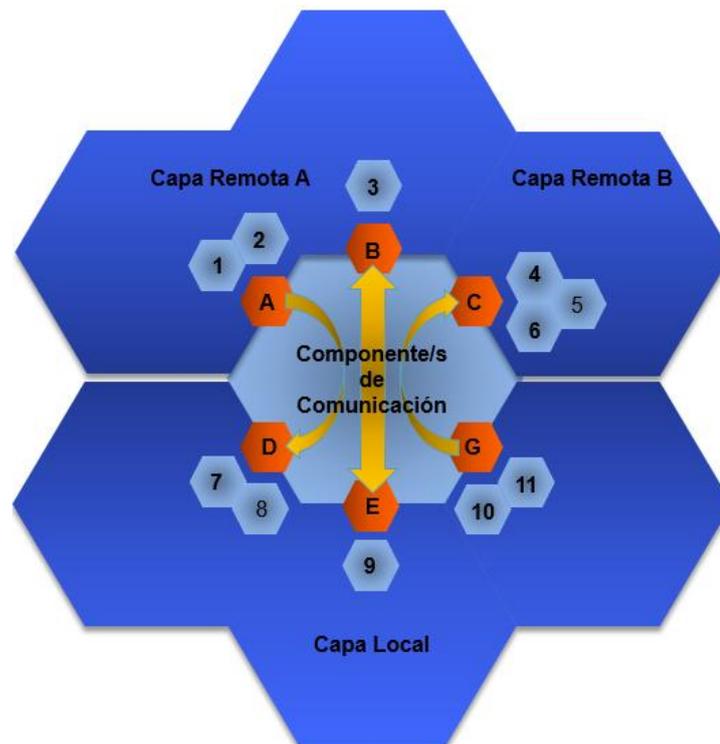


Fig. 20 – Arquitectura Cliente-Servidor / Cliente-Servidor: Direccionalidad de la Información

Primitivas, Sentencias, Relaciones y Procesos

Si al momento de diseñar la arquitectura de TI no se siguen principios de diseño relacionados con primitivas, modelos de referencia y/o modelos celulares, seguramente será muy complejo alcanzar objetivos de alineamiento, integración, reusabilidad, flexibilidad, etc. Si lo que se construye, diseña, administra y cambia no son modelos primitivos, entonces no se están realizando tareas relacionadas con Arquitectura. Lo que se



están haciendo son implementaciones, es decir diseño (o rediseño), construcción, administración y/o cambios sobre la instanciación de la Arquitectura o de los objetos/componentes que la integran. Es por ello que la Arquitectura de TI se desprende de la Arquitectura Empresarial, con sus propios modelos, principios y particularidades para instanciar la operación de los procesos descriptos por esta última.

Se definirá a la relación entre componentes mediante la utilización de sentencias con una sintaxis básica previamente definida. Las sentencias estarán compuestas al menos por:

- Uno o más componente que relacionan;
- Una primitiva o conjunto de primitivas, a las cuales se definirán como predicados elementales relevantes para la existencia, descripción, relación y comportamiento entre dos componentes, incluyendo sus restricciones, estados posibles y reglas;
- Uno o más componentes que son relacionados.

El primer y el último punto no solamente establecen la dirección de la relación sino además la cardinalidad de la sentencia. Adicionalmente se definirán como componente "Source" al componente que inicia la relación y componente "Target" al componente objetivo de la misma. En relaciones bidireccionales, cualquier componente podría ser source o target para la relación que se especifique.

Un proceso estará determinado entonces por una o más sentencia cuyo conjunto de primitivas relacionan uno o más componentes con el objeto de realizar una o más transformaciones sobre algún componente o conjunto de componentes, recordando que uno o más componentes podrían conformar un servicio o una parte del mismo.

De esta forma se configura una estructura ontológica, una teoría que establece la existencia de un conjunto estructurado de componentes esenciales para un objeto en el cual las expresiones explícitas de éstas son básicas e incluso obligatorias para su descripción, creación, operación, relación, cambios y comportamiento en general.

A modo gráfico, en el ejemplo que sigue se describe el servicio SC, el cual está integrado por tres componentes nombrados C4, C5 y C6 respectivamente. Por cada N instancias del componente C4 existe una instancia del componente C6 en una relación bidireccional representada por la relación $R(C4,C5)$ donde la flecha \leftrightarrow indica la dirección de la relación. De igual forma por cada N instancias del componente C5 existe una instancia del componente C6 en una relación bidireccional representada por la relación $R(C6,C5)$, donde la flecha \leftrightarrow indica la dirección de la relación. Por último, cada instancia del componente C5 configura una instancia del servicio SC, el cual es expuesto para su consumo a los clientes, sean locales o remotos.

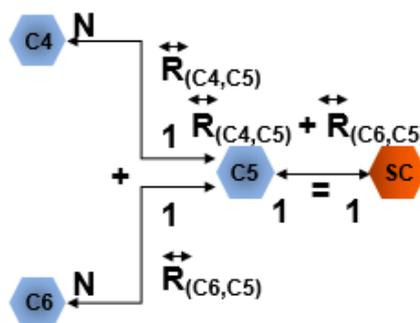


Fig. 21 – Arquitectura Cliente-Servidor / Cliente-Servidor: Macro Descripción de un Servicio



En lenguaje común, podríamos describir el servicio SC como:

“La relación entre N instancias del componente source/target C4 hacia una única instancia del componente source/target C5 o la relación entre N instancias del componente source/target C6 hacia una única instancia del componente source/target C5 configuran una única instancia del servicio SC”

Se podría describir el servicio SC de una forma más abstracta mediante una expresión más formal, entendiendo a la doble flecha como una compuerta lógica OR (+):

$$C5 = \overset{\leftrightarrow}{R}_{(C4,C5)} + \overset{\leftrightarrow}{R}_{(C6,C5)} = SC$$

Fig. 22 – Expresión formal de una relación en estado primitivo

Supóngase una aplicación (el cliente) en donde ingresados una serie de datos, se consume un servicio (SC) que está escuchando nuevas conexiones y en el cual de acuerdo a sentencias que regulan el comportamiento del componente C5 (componente source), podría disparar N consultas o una sola en un rango desde-hasta, sea en el componente C4 (componente Target) o en el componente C6 (componente Target) dependiendo de, por ejemplo, un determinado dato.

Arquitectura Cliente-Servidor / Cliente-Servidor

En su concepción tradicional, la arquitectura cliente-servidor es un modelo de sistemas distribuidos en donde múltiples clientes realizan múltiples peticiones a múltiples servidores, que procesan la información y devuelven los resultados a quien corresponda. Es preciso recordar en este punto que, en la línea que se viene desarrollando, tanto los clientes como los servidores pueden ser modelados como componentes o conjunto de componentes (de software o de hardware), y por lo tanto podrían estar relacionados de alguna forma. Cada uno de ellos podría contener cierta lógica de negocios, propiedades, estados y reglas que definan su comportamiento. La diferencia está en que los clientes son demandantes o consumidores de servicios mientras que los servidores los exponen para que sean consumidos.

Si bien cliente y servidor podrían estar geográficamente distribuidos, la separación entre ellos es pura y exclusivamente lógica. No se avanzará en detalles acerca de diseño de soluciones de software en dos capas, tres capas o N capas, ya que si bien no es un tema menor, cualquier diseño que se utilice podría montarse sin inconvenientes sobre la arquitectura en análisis y no constituye un factor condicionante.

En una Arquitectura de TI sobre la cual se pueda desplegar una solución, tanto cliente como servidor pueden ser modelados como componentes, ya que en la descripción que se pretende obtener constituyen elementos lógicos (o físicos en fase de implementación del ciclo de vida de la solución que aplique) y que como se había mencionado más arriba, encapsulan una o más funcionalidades.

Si bien podría ser utilizada con otros fines ajenos a este trabajo, la Arquitectura en análisis se enfoca en las aplicaciones de misión crítica que debido a cuestiones de distribución geográfica o particularidades regionales de los clientes que la utilizan, necesitan tenerlas disponibles todo el tiempo, de tal forma de operar el negocio en forma continua independientemente de los riesgos indentificados que se pretendan mitigar o eliminar con su implementación.

Adicionalmente y previo análisis, invariablemente se podrá desplegar localmente la aplicación de misión crítica completa o únicamente aquellos procesos necesarios de tener disponibles. Lo importante es que operando en Modo Asíncrono, los procesos de negocios puedan seguir siendo operados de tal forma de mantener la continuidad del negocio. Obviamente no escapa a este análisis la forma en la cual la información deberá

Si bien el diseño de una arquitectura para dar soporte a este tipo de necesidades puede variar tanto como la habilidad y la imaginación del arquitecto lo limiten, lo que define la particularidad y por caso la novedad de la arquitectura propuesta, es la doble función del componente servidor que se ubique en la capa local en relación a su par en la capa remota. Gráficamente:

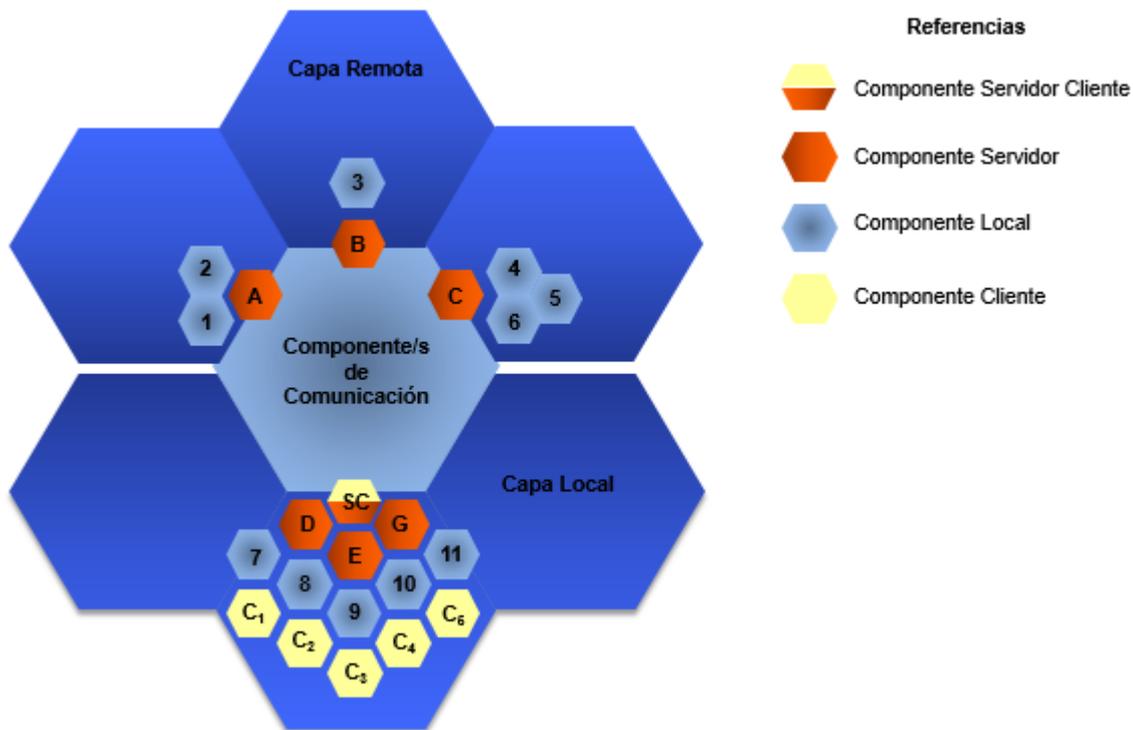


Fig. 23 – Arquitectura Cliente-Servidor / Cliente-Servidor

En el diagrama anterior podemos interpretar como clientes locales conectan con servicios locales que podrían estar configurados por uno o más componentes locales, y que en su conjunto configuran un componente (podrían ser más) denominado “Servidor Cliente”, ubicado en el borde lógico de la capa local. Este componente es Servidor para los clientes locales que solicitan sus servicios pero vía componente/s de conexión, es cliente (remoto) para los servidores que proporcionan servicios desde la capa remota. Esta desagregación lógica respecto de la capa remota es la que, dependiendo de la profundidad del análisis y el alcance que se le pretenda dar a la solución que aplique, es la que permitirá a la Capa Local trabajar en forma aislada de la capa remota (Modo de Operación Asíncrono). Precisamente debido a esta consideración es necesario incluir dentro del análisis de las funciones de los componentes Servidor-Cliente la lógica que permita sincronizar la información en base a la direccionalidad que la misma necesite, incluyendo las reglas que permitan resolver los conflictos e involucrando a los componentes que sean necesario.

Está claro que dependiendo de la complejidad de la solución que se implemente sobre esta arquitectura y en base a la criticidad de los procesos que se pretendan disponibilizar en forma continua, es posible y probable que expandiendo la perspectiva del cliente local a la del cliente remoto podamos obtener modelos funcionales y consistentes, dando lugar a una arquitectura modular, flexible, robusta y escalable, aplicable también a ambientes heterogéneos:

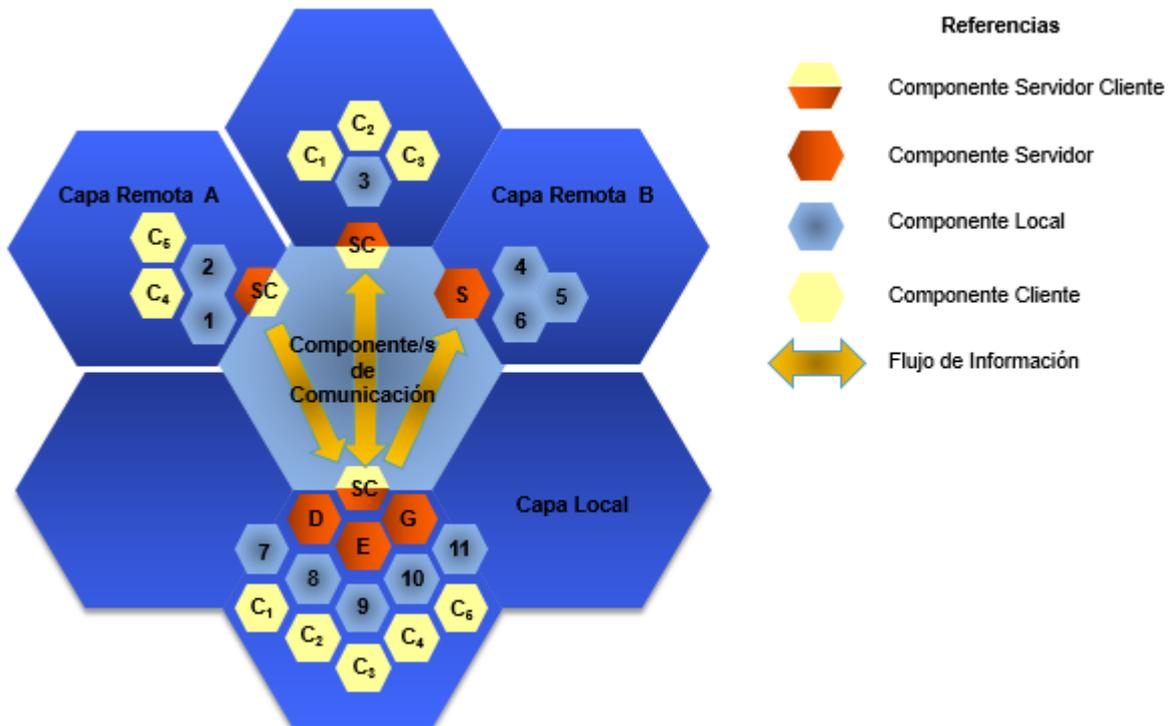


Fig. 24 – Variante Arquitectura Cliente-Servidor / Cliente-Servidor

Resumen

En este capítulo se ha visto como desagregando o bajando en niveles de detalle a partir de la definición de un modelo de dos capas lógicas, se logra un modelo de N capas sobre el cual, si bien faltan algunas cuestiones en las que se ahondará en los próximos capítulos, se pueden desplegar las aplicaciones de misión crítica de la compañía, logrando un modelo flexible y consistente, cuya modularidad permitirá no solo operar en forma continua los procesos de negocios, sino además atender cuestiones relacionadas con la no disponibilidad de servicios.



Capítulo

7

Objetos Primitivos

En un nivel de abstracción más profundo, este capítulo desagrega en elementos primitivos algunos de los elementos descritos en el capítulo anterior, ofreciendo a modo de ejemplo una formalización semántica propia de la comunicación, a los fines de establecer relaciones entre dos o más objetos.

Se avanza en la definición de objetos primitivos como elementos necesarios para la construcción de componentes de mayor nivel a los cuales se le puede asociar cierta lógica de funcionamiento o comportamiento a los fines de obtener las funcionalidades deseadas, diferenciándola claramente de la lógica propia del negocio.

Introducción

En el capítulo anterior se habían definido una serie de elementos necesarios para la descripción de la arquitectura objeto, generalizando de tal manera que esos mismos elementos pudieran ser utilizados en la descripción de cualquier otra arquitectura. También se había mencionado que estos componentes se relacionaban mediante sentencias con una sintaxis definida, cada una de ellas compuestas por un conjunto de primitivas, con el fin de ejecutar una transformación (proceso) determinada.

También se había clasificado a los componentes como Source o Target desde el punto de vista de quien inicia y quien es afectado por una relación. Inclusive podemos encontrarnos con ciertas cuestiones de perspectiva, donde un componente es target para una relación con un componente x, pero es source para la relación con un componente y.

Los componentes entonces se relacionan entre sí mediante un conjunto de primitivas que configuran una o más sentencias y que determinan el comportamiento de los primeros. Una agrupación de componentes tienen una finalidad en particular que dan lugar a servicios, un tipo especial de componente el cual, en base a peticiones de un cliente (otro tipo especial de componente), realiza procesa el requerimiento en base a ciertas reglas lógicas y devuelve un resultado. Este último punto configuraría el comportamiento de un Modelo Cliente-Servidor.

De entre todos los elementos considerados y, enfocados en la descripción particular de una Arquitectura Cliente-Servidor / Cliente-Servidor que permita la operación continua de un negocio (aplicaciones de misión crítica), se había considerado el caso especial de ciertos componentes con la particularidad de ser componentes servidor para una determinada cantidad de clientes, pero cliente para algún otro servidor. Las posibilidades de combinar esta variedad de elementos es lo que proporciona

No se debe confundir en este punto la lógica del negocio con la lógica que se utilice para la relación entre los componentes. Si bien son dos cuestiones que no deben estar desconectadas, la primera merece un análisis y enfoque en particular (y no por ello muy distinto), peor es imprescindible tener bien definido primero la arquitectura sobre la cual se va a apoyar. Es cierto también, que dependiendo de ciertas cuestiones, requerimientos y/o necesidades, la lógica del negocio y las herramientas que se utilicen para su



implementación determinan ciertas cuestiones arquitectónicas. Sin embargo, es necesario recordar en este punto que si estas cuestiones arquitectónicas no implican la construcción, el diseño, la administración o el cambio de elementos primitivos, entonces no se están realizando tareas de Arquitectura, sino implementaciones, acciones sobre la instanciación de objetos que componen la Arquitectura sobre la cual se montan.

Modelos Primitivos

Con la premisa de que se trabaja en Arquitectura cuando se trabaja sobre modelos primitivos, se debería definir primero qué se entiende por Modelo Primitivo, a esta altura sobreentendiendo de que se habla cuando se habla de Modelo. En el contexto de este trabajo, se entenderá por primitivo a todo aquel objeto indivisible y cuya razón de ser y función se definen por sí mismo. Para funcionalidades particulares, un objeto primitivo podría estar compuesto por más de un elemento primitivo o conjunto de elementos primitivos, lo que supone la posibilidad de dividirlo. Sin embargo y para que sean primitivos, esa división provocaría la pérdida de la razón de ser del mismo.

Un ejemplo práctico a los fines de graficar la idea es la del Arquitecto que diseña una casa. La descripción de la arquitectura de la casa es un objeto primitivo y para el caso, equivale al plano sobre el cual se bosqueja su idea. La instanciación de la arquitectura también es un modelo primitivo: la casa en sí, alcanzando paredes, techo, piso, aberturas. Inclusive podrían existir N instancias de la casa para el caso que se construya un barrio siguiendo un único modelo. Cualquier cambio que se realicen sobre aspectos tales como el color de las paredes, los materiales de las puertas, la decoración o disposición de los muebles son cambios relacionados a la implementación y no a su arquitectura, modelo o estructura. Esto refuerza la idea de que la disciplina de la arquitectura trabaja sobre modelos primitivos.

La definición de las relaciones que asocian a dos objetos (o componentes dependiendo de la fase del ciclo de vida en la cual se encuentre), se construyen en base a sentencias compuestas por primitivas, vinculadas a través de constructores predefinidos que le dan un significado lógico. Estos constructores lógicos son los operadores lógicos tradicionales: OR, AND, NOR, NAND, etc., pero no es objetivo de este trabajo ahondar en sus detalles y se da por entendido su significado. Así y todo, en la construcción de un modelo arquitectónico podrían definirse otro tipo de operadores, combinándolos o acotando la utilización a un par de ellos.

Entonces: en la construcción de una arquitectura es necesario definir el conjunto de primitivas a partir de las cuales se construirán las sentencias, las cuales vinculadas mediante la utilización de constructores lógicos, describirán la funcionalidad que los objetos relacionados desarrollarán para el modelo en construcción.

Este panorama se extiende considerablemente si se introduce la idea de recursividad y la utilización de variables, para mediante su determinación, obtener un comportamiento u otro. Esto se logra mediante la utilización de operadores condicionales, tales como IF, FOR, CASE, etc. Obviamente, la idea de trabajar con variables sugiere la posibilidad de utilizar los constructores lógicos mencionados anteriormente y otros constructores matemáticos primitivos (SUMA, RESTA, MULTIPLICACIÓN, DIVISIÓN) y otras funciones que la tecnología que se aplique podría resolver.



Sin embargo corremos el riesgo de mezclar las tareas propias de la Arquitectura con tareas propias de la implementación. Si bien es posible en la práctica que algún componente primitivo propio de la arquitectura sea utilizado para resolver cuestiones lógicas, es mejor mantener separadas las aguas, y como se mencionaba más arriba, trabajar sobre los modelos primitivos con la idea de enfocar el trabajo y no perder generalidad, aunque no por ello flexibilidad mediante su combinación.

Primitivas

En lo que al bosquejo de la Arquitectura de TI Cliente-Servidor / Cliente-Servidor en particular se refiere, es necesario definir el conjunto de primitivas que serán necesarias para comenzar a delinear las sentencias que configurarán el comportamiento y función de los objetos que se relacionarán, y que configurarán los componentes sobre las cuales se implementarán las aplicaciones de Misión Crítica, cada uno de ellos con su función específica.

En el modelo más básico, se habían mencionado la presencia de al menos dos capas lógicas que podrían estar física o geográficamente separadas o no, pero que estaban comunicadas a través de un componente de comunicación que a esta altura del trabajo ya podemos identificar como primitivo, al menos para la arquitectura objeto de análisis. Se deberá comenzar entonces, a partir de la existencia de tres elementos primitivos:

- Capa local, que se identificará como A;
- Componente de comunicación, que se definirá como B;
- Capa remota, que se identificará como C.

Si bien las posibilidades son variables en base a la finalidad que se persiga o a las necesidades que se pretendan satisfacer, en concordancia con lo expuesto y a los fines de ejemplificar, se pueden definir algunas primitivas para configurar el objeto “componente de comunicación” y su relación con los otros dos elementos.

Desde el punto de vista arquitectónico, la idea es que cualquier componente de comunicación que pongamos en medio para relacionar la capa local A con la capa Remota C, cumpla con las primitivas establecidas para el modelo. Para el ejemplo, se define la primitiva “Comunicar”, con lo cual de acuerdo a la sintaxis que se había establecido en el capítulo anterior la relación quedaría formalizada como:

$$\overset{\leftrightarrow}{B}_{(A,C)}$$

Si bien es factible utilizar la Arquitectura Cliente-Servidor / Cliente-Servidor para otras cuestiones como por ejemplo la operación en ambientes heterogéneos donde los servicios son provistos por servicios distribuidos geográficamente, la razón de ser y alcance de este trabajo es dar soporte a las aplicaciones de Misión Crítica permitiendo la operación continua de los procesos de negocio claves. Esto supone la posibilidad de que el componente de comunicación no esté disponible. Cualquier Arquitecto que entienda la criticidad de este punto tomará cartas en el asunto tratando de protegerse de una falla o una contingencia de este tipo.

Para seguir con el ejemplo, se agregará algo de lógica a la relación B que configura el componente de comunicación. Hasta ahora, el componente de comunicación relaciona dos objetos A y C mediante una sentencia compuesta por una sola primitiva: “comunicar”. Se incorporarán entonces al modelo, las siguientes:



- El constructor condicional estándar “IF THEN”;
- El operador matemático “=” para determinar el estado de un objeto o asignar un valor;
- Los objetos B1 y B2: que representarán dos canales de comunicación, uno primario y el otro secundario y configuran componentes primitivos en sí mismos;
- Los estados “Alcanzable” y “No Alcanzable”: como propiedad que definen la disponibilidad o no llegar a C;
- La primitiva “usando”, para canalizar la comunicación a través de B1 o B2;
- El elemento primitivo “Modo Operación” que podría tener dos valores posibles: sincrónico y asincrónico.

Ahora, la relación podría estar determinada de la siguiente manera:

```
“IF C = <Alcanzable> <usando> B1 THEN <comunicar> <usando> B1”  
“<Modo Operación> = <Sincrónico>”  
“IF C = <No Alcanzable> <usando> B1 THEN <comunicar> <usando> B2”  
“<Modo Operación> = <Sincrónico>”  
“IF C = <No Alcanzable> THEN <Modo Operación> = <Asincrónico>”
```

Literalmente se está expresando:

Si se pueden consumir los servicios expuestos por C, entonces establecer la comunicación utilizando el canal B1, en cuyo caso el Modo de Operación será sincrónico.

Si no se pueden consumir los servicios expuestos por C, entonces establecer la comunicación utilizando el canal B2, en cuyo caso el Modo de Operación también será sincrónico.

En caso no estén disponibles los servicios expuestos por C, entonces establecer el Modo de Operación en Asincrónico.

Formalizando:

$$\overleftrightarrow{R}_{(A,B1)} + \overleftrightarrow{R}_{(A,B2)} = C + A$$

Esto significa que se necesitará comunicación a través de B1 o B2 para operar en Modo Sincrónico (C disponible o alcanzable), sino se deberá trabajar en Modo Asincrónico.

Gráficamente:

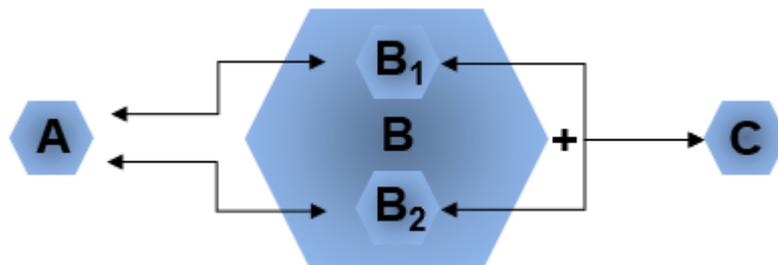


Fig. 25 – Arquitectura Cliente-Servidor / Cliente-Servidor: Relaciones entre Objetos Primitivos



Lo expuesto anteriormente, no significa que se estén sentando las bases para un nuevo lenguaje de descripción de arquitecturas (ADL por sus siglas en inglés), ya que no es intención del presente trabajo. De hecho, existen metodologías extensas, aplicables a cualquier tipo de arquitectura. Lo que se intenta es graficar como los fundamentos primitivos de la comunicación y la lógica proposicional pueden contribuir a lograrlo y como la determinación de elementos primitivos ayudará a construir arquitecturas robustas y flexibles mediante la utilización de sentencias bien formadas.

Conclusiones

Un elemento primitivo es un elemento indivisible y cuya razón de ser y función se definen por sí mismo. La relación entre primitivas se establece mediante la utilización de constructores, los cuales podrían alcanzar a operadores lógicos, matemáticos u otros constructores compuestos o no que sean necesario definir para la descripción de que se trate. Un conjunto de elementos primitivos relacionados por constructores configuran sentencias que construyen componentes, una clase particular de objetos que agrupan funcionalidades a forma de contenedor y que tienen ciertas características y propiedades a través de las cuales basan su comportamiento.

Si bien pueden construirse componentes primitivos desagregables o divisibles en unidades más pequeñas, los elementos primitivos pierden razón de ser y ya no son definibles por sí mismos. Las tareas propias de la Arquitectura de la cual se trate, pero en particular de la Arquitectura de TI están relacionadas a Modelos primitivos como conjunto de elementos primitivos interrelacionados.



Capítulo

8 Componentes y Servicios

Como continuación del capítulo anterior, este módulo avanza en el análisis de las relaciones entre elementos primitivos que configuran componentes que le dan funcionalidad a la Arquitectura: Componentes y Servicios. Si bien es cierto que se podrían definir elementos primitivos adicionales, a los fines del trabajo, estos dos representan un muy buen ejemplo de como la combinación de elementos primitivos pueden configurar componentes complejos.

Componentes y Servicios Primitivos

Es innumerable la cantidad de componentes y/o servicios que se pueden considerar o tener en cuenta al momento de diseñar una arquitectura. En organizaciones complejas o que tienen sus negocios diversificados puede tornarse muy difícil contemplar todos los objetos que se podrían necesitar a futuro. Es por eso que la tarea de Arquitectura es extremadamente dinámica.

Antes de profundizar con el análisis de estos elementos, es preciso diferenciar los componentes y servicios propios de la Arquitectura (primitivos) y los propios de la solución (primitivos o no), si bien en lo que a la Arquitectura en Análisis respecta su composición y función no varía, solo difieren en que unos son objetos elementales con una razón de ser que los define por sí mismo, y los otros no. Además esta diferencia permite delimitar o identificar aquellas tareas que son propias de Arquitectura, es decir aquellas que se aplican sobre elementos primitivos, de aquellas que son propias de las Implementaciones, que aplican sobre elementos no primitivos.

Aquellos elementos primitivos propios de la aplicación se definen en tiempos de diseño de la Arquitectura de la Aplicación y merecen igual tratamiento que los elementos primitivos que fueron diseñados al momento de delinear la Arquitectura de TI. De la misma forma que el diseño de la Arquitectura de TI debe estar alineada a la Arquitectura Empresarial, la Arquitectura de la Aplicación debe estarlo respecto de la Arquitectura de TI, en una permanente interacción entre ellas. Lo mismo debería ocurrir respecto de la Arquitectura de la Información, de Negocios, de Tecnología y las que se definan como área de alcance de la Arquitectura de TI en el ámbito de la Arquitectura Empresarial.

Se definirá un componente como una agrupación lógica de sentencias que encapsulan una única y determinada transformación, siempre recordando que las sentencias están conformadas por primitivas relacionadas por constructores lógicos, matemáticos, de decisión o cualquier otro tipo que se pueda sumar a la definición. Una primera clasificación de los componentes podría determinarse de la siguiente forma:

- Componentes simples: agrupación lógica de sentencias que encapsulan una única y determinada transformación, tal cual lo definido en el párrafo anterior;
- Componentes compuestos: agrupación de componentes que encapsulan una determinada funcionalidad o conjunto de funcionalidades, entendiéndose por funcionalidad, a un conjunto de procesos que actúan de manera sincronizada para un mismo fin.



Se está en condiciones ahora de definir a un servicio como un componente o conjunto de componentes que ejecutan una funcionalidad o conjunto de funcionalidades, con la particularidad de que atienden requerimientos que provienen de componentes clientes, los procesan y devuelven un resultado como objeto de su función.

Entonces, una segunda clasificación de los componentes podría determinarse de acuerdo a algunas ya mencionadas anteriormente:

- Componentes Clientes: componentes o conjunto de componentes que realizan peticiones de servicio a componentes servidor;
- Componentes Servidor: componentes o conjunto de componentes que atienden requerimientos de componentes clientes, los procesan y devuelven un resultado;
- Componentes de comunicación: componentes que vinculan a dos o más componentes y que permiten el intercambio de información y la petición y atención de servicios.

No es objetivo mencionar todos los tipos de componentes que se puedan encontrar en una Arquitectura promedio, sino dar la idea de que se pueden clasificar en tantos como funcionalidades necesite la Arquitectura que se está describiendo en base a las necesidades de negocio que se pretenden satisfacer. En este caso en particular y bajo el dominio de este trabajo, se definirá como “componente Servidor-Cliente” al componente o conjunto de componentes que atienden requerimientos de componentes clientes, los procesan y, o bien devuelven un resultado, o bajo determinadas condiciones realizan una petición a otro componente servidor actuando como cliente en ese caso. Es el eje sobre el cual gira la arquitectura en análisis y es un concepto clave que determina operar en Modo Sincrónico o Asíncrono permitiendo la operación continua de la/s aplicación/es de misión crítica.

Identificación de Componentes

Si bien la identificación de componentes y/o servicios obedece a cuestiones metodológicas y a como se pretende separar el diseño en unidades lógicas bien definidas, es tarea del Arquitecto de TI no solamente tener pleno conocimiento de las cuestiones arquitectónicas propias de su alcance, sino también de las características propias de las soluciones que se apliquen sobre la misma, a los fines de diseñar el modelo que mejor se ajuste.

Es necesario destacar que estas tareas no podrían ser llevadas adelante sin cierta capacidad para interactuar con el Arquitecto de la Aplicación y otras áreas de alcance de la Arquitectura que podrían estar involucradas en la misma (Tecnología, Negocios, Información, etc.).

Un bosquejo inicial de la Arquitectura se podría iniciar listando las funcionalidades macro que necesiten ser satisfechas, que componentes se necesitarían para lograrlo, cuales son las transformaciones que necesitan ejecutar cada uno de los componentes, que sentencias lógicas deberían aplicarse y en que secuencia y cuál es el conjunto de primitivas y constructores que la implementan, para ir bajando en el nivel de detalle hasta encontrar elementos atómicos (primitivos), aplicando una metodología top-down.

De igual forma, aplicando una metodología down-top, el bosquejo inicial se podría iniciar definiendo el conjunto de primitivas y constructores sobre el cual se basan las sentencias lógicas que deberían aplicarse para ejecutar las transformaciones necesarias, agrupando las mismas en componentes que proporcionen las funcionalidades necesarias, en base a los requerimientos y necesidades de negocios.

Sin embargo en la práctica es difícil encontrar oportunidades en las cuales el diseño de la Arquitectura se pueda diseñar desde cero y/o de punta a punta para cualquiera de las dos metodologías mencionadas. Lo más probable es que se encuentren funcionalidades y componentes medianamente definidas y primitivas medianamente identificables a las cuales echar mano para rediseñar lo existente, y sea preciso definir algunas nuevas para soportar los procesos de negocios nuevos. Esto remite a las motivaciones que impulsan las tareas específicas de la Arquitectura y al estado actual de la misma y le imprimen su carácter dinámico.

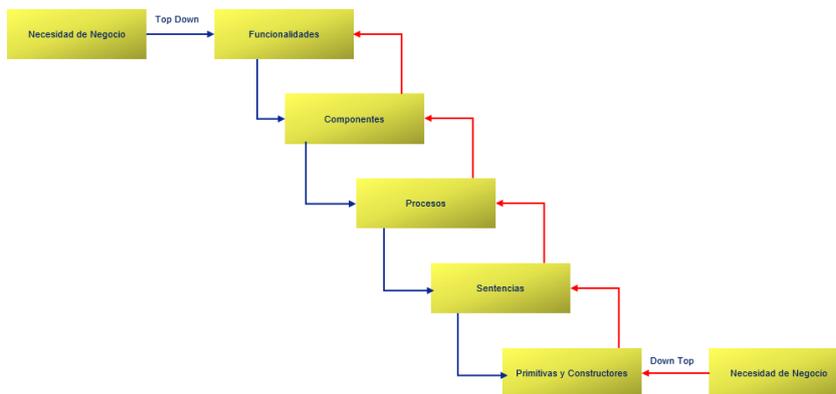


Fig. 26 – Arquitectura de TI: Metodologías de Componentes

También habla de la necesidad de identificar el punto desde el cual se debe disparar el proceso de diseño para los elementos nuevos de la arquitectura, rediseño de los existentes o adecuaciones de los nuevos a los existentes (o viceversa según corresponda). Para una solución como la planteada para el Ingenio azucarero, este es el punto donde Arquitectura de Negocios, Arquitectura de Aplicación, Arquitectura de TI y Arquitectura de la Información convergen y comienzan a interactuar en pos de satisfacer la necesidad de negocio específica, siempre en base a los lineamientos de Arquitectura de TI y Arquitectura Empresarial. Adicionalmente, la metodología y el ciclo de vida están determinados por los estándares de Gobierno de TI adoptados por la organización.

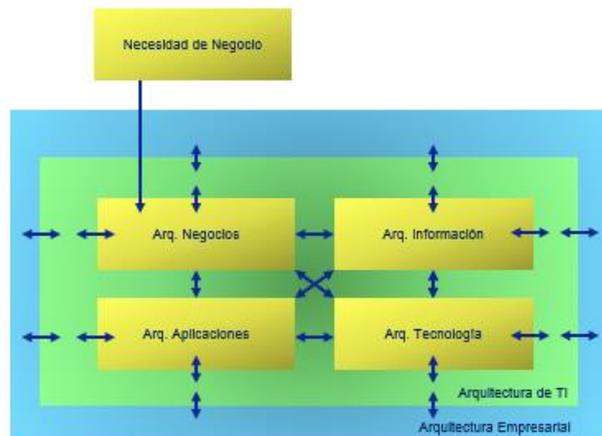


Fig. 27 – Arquitectura de TI: Identificación de Componentes



El diseño de una nueva solución no necesariamente implica realizar alguna tarea sobre la Arquitectura de TI existente, siempre y cuando se acople perfectamente a la misma. En esos casos, las tareas relacionadas con la Arquitectura de la Aplicación tienen que ver más con definir que componentes existentes llevarán a cabo una determinada funcionalidad que con operar sobre elementos primitivos, lo cual implica tareas de administración de dichos elementos. Para los casos en los cuales se necesite diseñarlos, rediseñarlos, o cambiarlos y como se mencionó anteriormente, se deberá identificar el punto a partir del cual se iniciará el proceso y aplicar la metodología que corresponda, involucrando a las áreas de Arquitectura alcanzadas.

Estrategias

Si bien diseñar, rediseñar, construir, e incluso identificar componentes y servicios está directamente relacionada con la forma en la cual los modelos primitivos en los cuales se basen fueron diseñados en cuanto a tareas de Arquitectura respecta, independientemente de la metodología que se implemente siempre es conveniente adoptar una estrategia que permita agruparlos lógicamente, aunque esto tenga que ver con aspectos metodológicos más que con tareas propia de la Arquitectura.

En toda solución de software que se aplique (y por tanto dentro del alcance del Arquitecto de Software), existirán clases de entidad que necesitan ser representadas de alguna forma en el modelo. Dichas clases son procesadas por clases de control que contienen la lógica de la solución. Sin embargo también existirán elementos ajenos a la solución a la cual aplique la arquitectura de software, y que necesitan ser representados para darle determinada funcionalidad o integrarla al universo de las aplicaciones de la organización.

En una primera instancia, clasificar los componentes en base al área de Arquitectura que la diseña, rediseña, construye, administra o cambia no está mal. Se podrán entonces encontrar:

- Componentes propios de la Arquitectura de la Información: repositorios de datos tales como las Bases de Datos, interfaces para la integración de la información y su tratamiento, etc.;
- Componentes propios de la Arquitectura de Tecnología: tales como los canales de comunicación que se utilicen, la tecnología de base de datos estándar que aplique, hardware necesario, etc.;
- Componentes propios de la Arquitectura de Negocios: que tengan que ver con la forma en la cual los mismos deban ser tratados;
- Componentes propios de la Arquitectura de Software: como por ejemplo las clases de entidad, las clases de control y su distribución en el modelo arquitectónico;

lo que da lugar a una nueva clasificación de los mismos.

De todas formas, en la práctica es casi imposible diseñar componentes en forma aislada y luego integrarlos. Por ello es necesario realizar estas tareas en forma interdisciplinaria y la determinación de los componentes a utilizar y la asignación de las responsabilidades debe ser un trabajo de equipo. Adicionalmente es necesario aclarar que dependiendo de la tecnología que se aplique, las metodologías, estándares de gobierno y otras políticas y principios organizacionales se podrían encontrar estos elementos con distintos nombres y/o bajo la responsabilidad de distintas áreas.



Conclusiones

Componentes y Servicios constituyen bloques de construcción que las áreas de Arquitectura diseñan, rediseñan, construyen, administran y cambian como parte de las propias de su alcance y responsabilidad y que ponen a disposición de los procesos de implementación para la satisfacción de las necesidades objeto de la solución. Para la identificación de los componentes necesarios es preciso determinar el punto de inicio del proceso que mejor se acomode a las necesidades o que facilite más las tareas de diseño, teniendo en cuenta las características interdisciplinarias y de trabajo en equipo del mismo.

Más allá de la clasificación que se haga y de las responsabilidades que se asignen a cada área, se necesita no perder este enfoque para garantizar la correcta integración de los elementos y su posterior correcto funcionamiento. La adopción de una metodología cualquiera sea esta mejora la identificación de componentes y ayuda a construir bloques de arquitectura sólidos, flexibles, reutilizables y consistentes, totalmente integrables entre sí, garantizando los objetivos para los cuales son creados.



Capítulo

9

Arquitectura CS/CS

En este módulo se formaliza la Arquitectura Cliente-Servidor/Cliente-Servidor en un lenguaje gráfico basado en UML, de forma tal que pueda ser aplicada a soluciones con similares necesidades. Si bien es lógico aclarar que cada solución merece su propio análisis y que la generalidad de su aplicabilidad y la concepción de los elementos que la componen podrían no ser (justamente) tan específica, es allí donde precisamente radica su fortaleza, destacando que la robustez y flexibilidad que se alcanza con este tipo de arquitectura permite soportar la operación continua de las aplicaciones críticas que operan el negocio de la organización.

Introducción

En línea con lo expuesto en capítulos anteriores, las tareas que desembocaron en la Arquitectura Cliente-Servidor / Cliente-Servidor fueron tareas propias de la arquitectura, es decir la operación sobre elementos primitivos. En los últimos años, han aparecido una serie de estándares de los cuales el IEEE42010 específicamente trata acerca de recomendaciones prácticas para la descripción de arquitecturas que hacen uso exhaustivo de tecnologías relacionadas con la información. Uno de los objetivos del estándar es promover un enfoque sistemático para la descripción de arquitecturas mediante la creación de vistas.

Una de las ventajas de su adopción, es que propone a UML como Lenguaje para la descripción de arquitecturas. Si bien esto no es mandatorio y podría utilizarse cualquier otro ADL inclusive combinándolos con elementos de UML, la popularidad de este último hace que sea un factor a considerar cuando se describe una arquitectura.

Lo importante en este punto es lograr un conjunto de elementos que hagan consistente y entendibles a la descripción de la arquitectura, mediante la utilización de representaciones significativas para los stakeholders objetivo. Es tarea del arquitecto realizar estas representaciones y seleccionar las que más se ajusten a lo que se pretende comunicar y explicar, de tal forma de demostrar que lo propuesto satisface las expectativas para las cuales ha sido desarrollada.

Principios

Los principios constituyen reglas generales y directrices perdurables en el tiempo y, rara vez, modificadas o adaptadas, que apoyan la forma en la cual una organización enmarca el cumplimiento de su misión. A su vez, son un solo elemento de un conjunto estructurado de ideas que en conjunto definen y orientan a la organización, partiendo de sus valores y a través de sus operaciones y resultados.

Dependiendo de la organización que se trate, los principios pueden ser categorizados en:

- Principios empresariales, que provee la base para la toma de decisiones en pos de alcanzar los objetivos organizacionales;



- Tecnologías de la Información, que proveen directrices para el diseño, creación, uso y despliegue de recursos de TI a lo largo y ancho de la organización, en línea con los principios empresariales y los objetivos organizacionales;
- Principios de Arquitectura, que componen un subconjunto de principios para las tareas relacionadas a la arquitectura que se trate. Se pueden sub dividir en:
 - Principios de gobierno de arquitectura, para el gobierno de los procesos de arquitectura;
 - Principios de gobierno de implementación, para los procesos de implementación.

En lo que a este trabajo en particular respecta, se mencionan principios relevantes para la conformación de la arquitectura Cliente-Servidor/Cliente-Servidor relacionados a las tecnologías de la información:

- Maximización de Beneficios y Minimización de costos;
- Reutilización de la infraestructura existente;
- Operación continua de los procesos de negocios críticos;
- Conformidad y alineamiento con los procesos de negocios y estándares de la organización;
- Independencia tecnológica;
- Facilidad de Uso, Implementación e integración;
- Utilización en ambientes heterogéneos;
- Modo de Operación Sincrónico o Asincrónico.

Evolución del Modelo de Referencia

La arquitectura Cliente Servidor que sirve como modelo de referencia sobre el cual operar hasta tener la arquitectura objetivo es un tipo de arquitectura de procesamiento cooperativo donde clientes y servidores colaboran en el proceso del total de las funcionalidades contempladas por una aplicación o conjunto de aplicaciones. Bajo este modelo, los clientes realizan peticiones de servicio (Requests) a los servidores, los cuales procesan estas solicitudes en base a cierta lógica y devuelven un resultado (Response) como parte de ese proceso.

A nivel físico (hardware) Clientes y servidores pueden compartir un mismo procesador, procesadores distintos, ubicados en el mismo servidor físico o en servidores virtuales, separados físicamente o no, lo que brinda un amplio abanico de posibilidades que permitirían adaptarlo casi a cualquier situación. Idéntica situación sucede con la memoria, disco y otros recursos tecnológicos que puedan contemplarse para la solución que se trate.

En general, los clientes son procesos/aplicaciones que realizan requerimientos de servicios y los servidores son procesos/aplicaciones que proveen esos servicios requeridos. Bajo este esquema, generalmente el cliente inicia la comunicación con uno o más servidores, aunque los servidores podrían proveer servicios sin necesidad de un requerimiento de un cliente, con una programación de una tarea por ejemplo. Adicionalmente, un servidor puede atender a n clientes a la vez y un cliente puede solicitar servicios a más de un servidor.

Cliente y Servidor pueden correr en un mismo hardware, en hardware distintos, o inclusive en redes distintas, estén física o lógicamente separados, lo que configura un esquema de capas lógicas en el cual

cada capa puede representar una función específica del modelo, ya sea por la función que representan para la aplicación o la función que representan para el modelo en sí.

El modelo cliente-servidor original de referencia es como el que se describe en la figura siguiente:

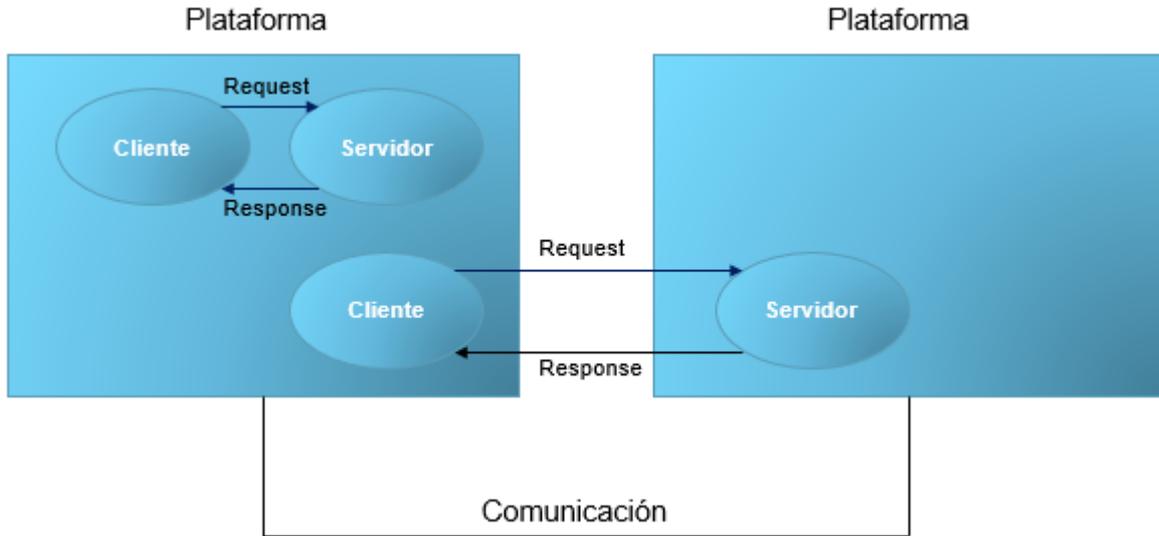


Fig. 28 – Arquitectura de TI: Arquitectura Cliente-Servidor básica

Mediante la ejecución de tareas relacionadas a la arquitectura, se pretende llegar a un modelo adaptado como el que sigue:

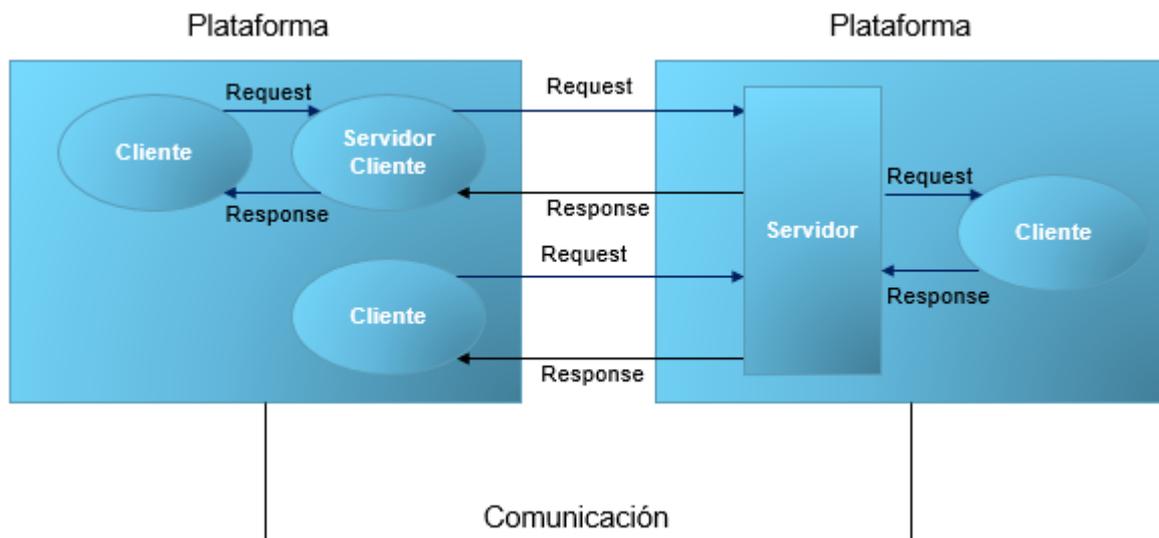


Fig. 29 – Arquitectura de TI: Arquitectura Cliente-Servidor / Cliente-Servidor básica



Vistas de la Arquitectura Cliente-Servidor / Cliente-Servidor

Una arquitectura puede ser representada de varias formas de tal modo de expresar un significado apropiado para el/los stakeholder/s objetivo, proporcionando una lectura consistente para todos que facilite su comunicación y entendimiento.

Si bien es probable que en la práctica (y como en este caso en particular) se esté representando una sola arquitectura específicamente, una sola vista de la misma puede no alcanzar para lograr el entendimiento que se pretende. Es por ello que una misma arquitectura puede ser representada por distintas vistas, las cuales puede proporcionar aspectos generales o expresar con más detalles ciertas cuestiones, en relación a distintos factores como pueden ser su comportamiento, sus componentes, relaciones o el público al cual esté dirigida y su conocimiento del tema, de tal forma que pueda ser entendidos por estos en un lenguaje apropiado.

A esta altura del trabajo ya se está en condiciones de representar el modelo Cliente-Servidor / Cliente-Servidor adoptando elementos de los estándares mencionados anteriormente. Adicionalmente, existen ciertos tipos de componentes que podrían ser considerados con más detalles en otras vistas o que podrían ser excluidos para no generar confusión. A los fines de aclarar e ingresar en el campo de la solución aplicada al Ingenio azucarero, se comenzarán a incluir componentes propios de esa solución, aunque como se mencionara anteriormente, todo depende de los elementos con los cuales el arquitecto cuente y las competencias técnicas, conocimiento del negocio y, en definitiva la imaginación de la cual disponga.

Vistas de la Arquitectura de TI:

El desarrollo de este conjunto de vistas tiene varias finalidades, entre ellas:

- Desagregar el modelo, partiendo desde una macro imagen con poco detalle;
- Mostrar la forma en la cual el mismo va evolucionando;
- Utilizar un lenguaje gráfico sencillo basado en UML y otros elementos que ayuden a comprender el significado;
- Correspondencia con las figuras y el desarrollo del capítulo 6 de este trabajo.

Adicionalmente, cada vista se corresponde con una fase determinada del ciclo de vida que se aplique, con lo cual logra cierta independencia del mismo.

Estos esquemas también permiten visualizar cómo sobre una arquitectura de TI ya existente se puede montar la arquitectura que se propone, en especies de incrementos sobre la misma, agregando componentes que brinden nuevas funcionalidades o reutilizando los existentes, factores que juegan un papel importante en el desarrollo de la Arquitectura Cliente-Servidor / Cliente-Servidor.

Modelo de dos Capas

Vista del modelo a gran nivel, donde se representan dos capas, una local y otra remota, con algún elemento para la comunicación entre ambas. La ubicación del actor determina que una capa sea local y la otra remota o viceversa.

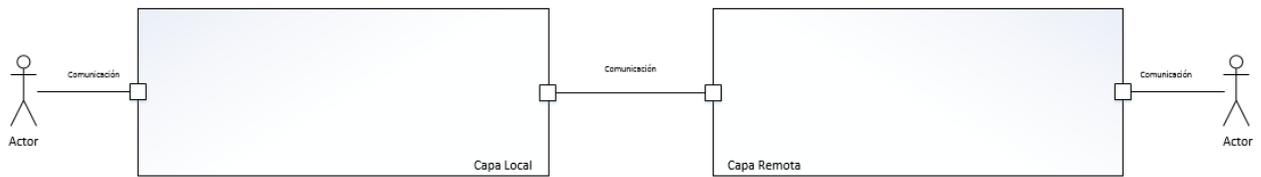


Fig. 30 – Arquitectura de TI: Vista del Modelo de dos Capas

Modelo de N Capas

Vista del modelo a gran nivel, donde se representan N capas, una local y N remotas, con algún elemento para la comunicación. La ubicación del actor determina que una capa sea local y las otras remotas. También se destaca que desde el punto de vista (perspectiva) del actor, la capa a la cual pertenece siempre será la capa local. Esta vista da una idea de aplicabilidad general cruzada entre N ambientes (capas).

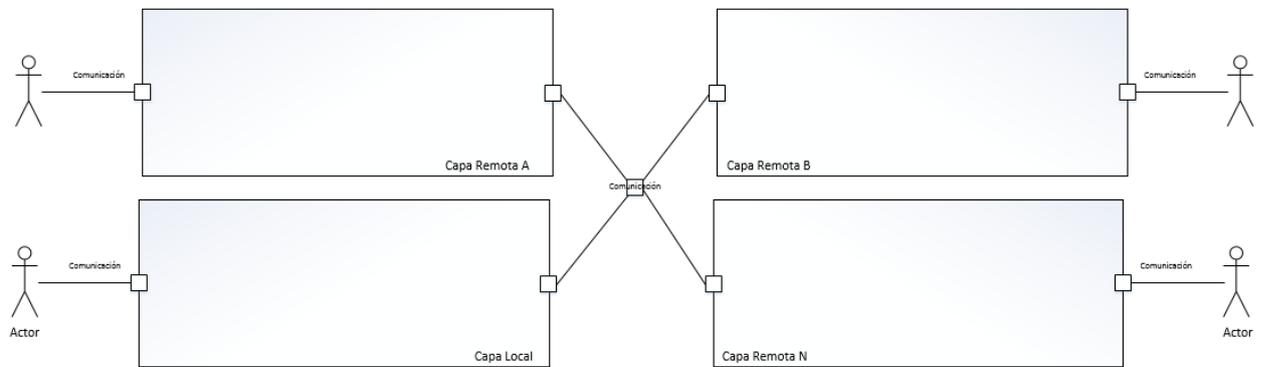


Fig. 31 – Arquitectura de TI: Vista del Modelo de N Capas

Modelo de 2 Capas con Distribución de Nodos

Vista del modelo a gran nivel, donde a partir de las vistas anteriores se desagregan para cada capa la distribución de nodos que soportarán las distintas capas lógicas de la aplicación. Sin embargo, esta distribución podría ser arbitraria y variar de aplicación en aplicación dependiendo de las necesidades y elementos disponibles.

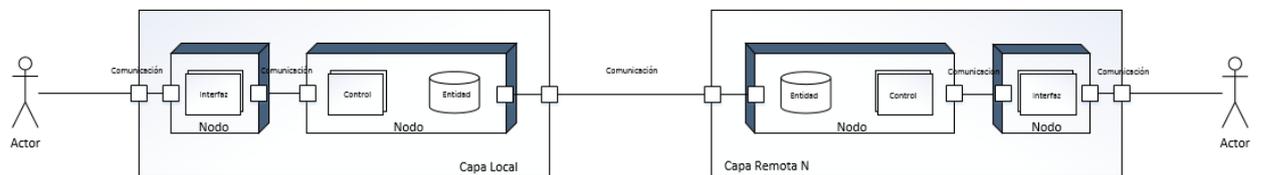


Fig. 32 – Arquitectura de TI: Modelo de dos Capas con Distribución de Nodos

Vistas de la Arquitectura de Aplicación:

El desarrollo de este conjunto de vistas tiene varias finalidades, entre ellas:

- Correlacionar las Vistas de la Arquitectura de TI expuestas en el apartado anterior con las Vistas de la Arquitectura de Aplicación, entendiendo por aplicación a cualquier aplicación de software en forma genérica;
- Desagregar el modelo, partiendo desde una macro imagen con poco detalle;
- Identificar componentes y servicios y su ubicación en el modelo, entendiendo siempre que la distribución podría variar dependiendo de las características de la solución que se aplique;
- Integrar Arquitectura de TI (infraestructura) con Arquitectura de la Aplicación, Datos e Información.

Modelo de Distribución de Componentes y Servicios

Vista del modelo con mayor detalle en donde se representan los componentes del software que aplica sobre la arquitectura existente. Se pueden observar claramente las capas lógicas que hacen a la aplicación y una distribución estándar de las mismas sobre distintos nodos, aunque en la práctica podría variar. Si bien cada componente referenciado como Interfaz, Control y Entidad representan clases para el software, desde el punto de vista que se pretende representar son tratados como componentes.

Cada uno de estos componentes exponen servicios, representados como círculos. Su ubicación en el diagrama también indica a qué otros componentes son expuestos. El servicio expuesto por el componente de Control, aparece entonces como candidato para cumplir funciones de servidor para la capa local y cliente para la capa remota. También se representa la interacción que los actores tendrán con el sistema mediante líneas de comunicación, las cuales ya evidencian un sentido lógico y direccionalidad del flujo de datos.

Esta distribución de componentes en nodos, comunicación entre los componentes y actores y exposición de servicios configura una arquitectura robusta y flexible compatible con las mejores prácticas para el desarrollo de software.

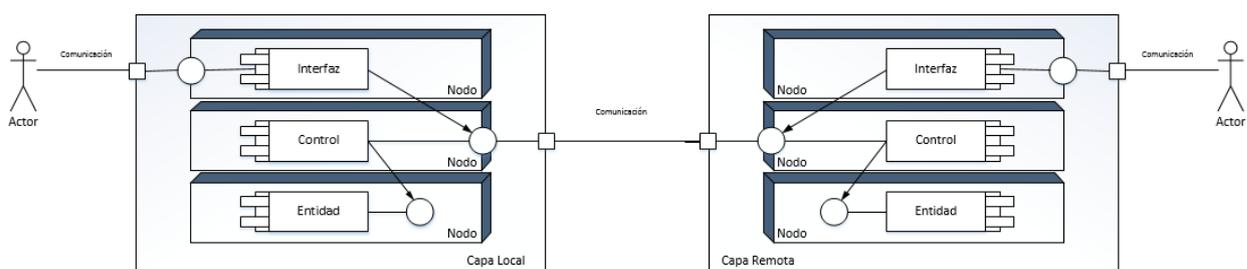


Fig. 33 – Arquitectura de TI: Modelo de Distribución de Componentes y Servicios

Modelo de Distribución de Componentes y Servicios de Alto Nivel

En esta vista se pueden apreciar componentes que cumplen determinadas funcionalidades y que exponen servicios con alcance local y/o con alcance remoto. Los servicios con alcance remoto son los

servicios candidatos para brindar funcionalidades de servidor a nivel local y cliente para las aplicaciones remotas.

Se mantiene la separación lógica en capas, aunque como se mencionara anteriormente esto podría variar de aplicación en aplicación. Se detallan funcionalidades a modo de componentes que a su vez pueden estar formadas por más de un componente.

Si bien en las vistas anteriores existían cuestiones relacionadas a la comunicación, en esta vista ya aparecen nociones técnicas relacionadas con la red LAN tanto local como remota, y la red WAN que las comunica, aunque bien podría haber otro tipo de comunicación, separada o no geográficamente. Lo importante es remarcar la separación lógica entre las capas (que por otro lado podría haber más de una y remota), y representar gráficamente funcionalidades que deberá cumplir la solución para configurar una arquitectura Cliente-Servidor / Cliente-Servidor que garantice la operación continua de las aplicaciones de misión crítica:

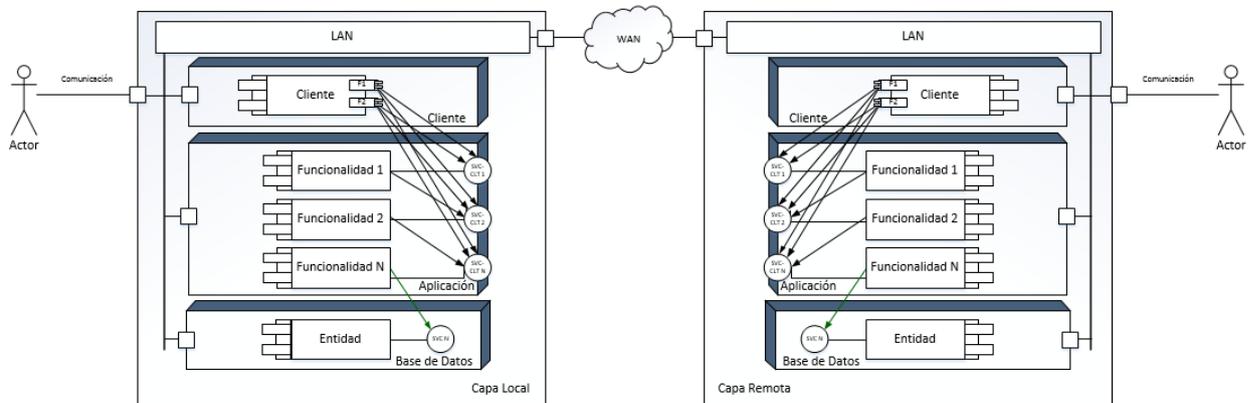


Fig. 34 – Arquitectura de TI: Modelo de Distribución de Componentes y Servicios de Alto Nivel

Modelo de Operación en Modo Asíncrono

En las vistas anteriores la presencia de vínculos de comunicación supone no solo su existencia sino también su correcto funcionamiento. Precisamente uno de los objetivos de la Arquitectura Cliente-Servidor / Cliente-Servidor es diseñar a las aplicaciones de Misión Crítica de tal forma de que puedan operar en forma continua, contemplando ciertas contingencias que se podrían presentar. En soluciones altamente dependientes de componentes de comunicación, es uno de los factores críticos de éxito a tener en cuenta.

Adicionalmente, no solo se deberá pensar la aplicación para operar en modo asíncrono, sino también contemplar aquellas funcionalidades que, una vez restablecida la comunicación, permitan re sincronizar la información en forma consistente.

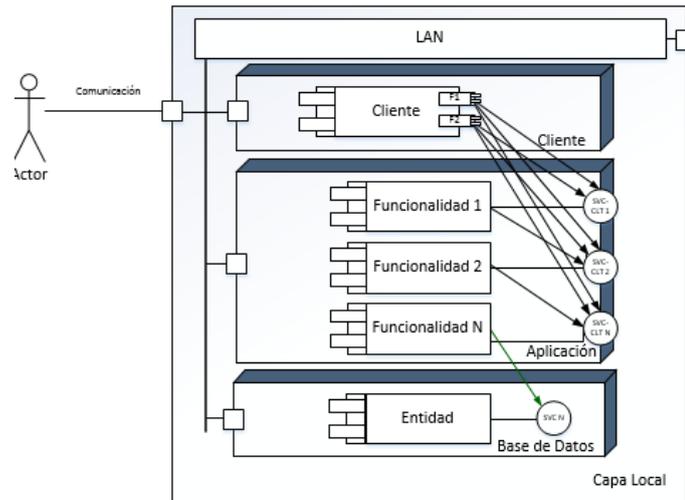


Fig. 35 – Arquitectura de TI: Modelo de Operación Asíncrono

Modelo de Flujo de Datos

En esta vista se representan componentes responsables de sincronizar la información, ya sea operando en modo síncrono o retomando el sincronismo luego de una contingencia. Se destaca la presencia de ciertas reglas que se deberán preestablecer para mantener la consistencia de la información. La ubicación de las reglas es lógica y se sobreentiende que la comunicación se establece a través de los mecanismos de comunicación existentes.

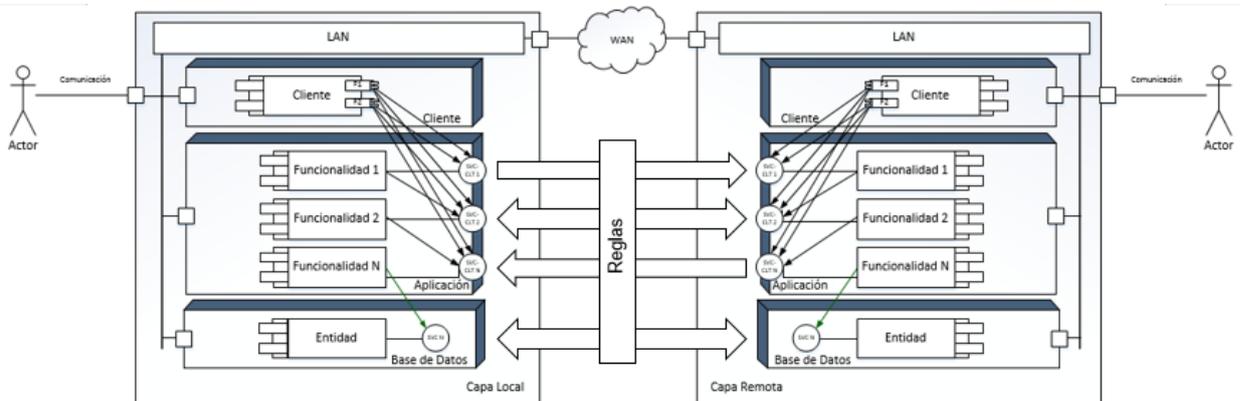


Fig. 36 – Arquitectura de TI: Modelo de Flujo de Datos

Relaciones entre componentes

Los distintos tipos de vistas definidos en el apartado anterior tienen que ver con distintos puntos de vista que colaboren al entendimiento de distintos stakeholders involucrados en la solución. De igual forma las

vistas anteriores tienen un fin técnico y seguramente no serán comprendidos completamente por los usuarios finales, lo cual está fuera del alcance de este trabajo, precisamente por tratarse de cuestiones técnicas.

De igual forma que la metodología que se aplique para la descripción de la arquitectura, esta no está dentro del alcance, pero existen ciertas consideraciones que a los fines prácticos y para redondear la idea en relación a la funcionalidad de los componentes es necesario aclarar.

En capítulos anteriores se había mencionado la existencia de elementos primitivos sobre los cuales operan las tareas relacionadas a la disciplina de la arquitectura, cualquiera sea la especialidad de la cual se trate. A los fines de ejemplificar y correlacionar con lo expuesto en el capítulo 7, se parte de una vista lo más abarcativa posible, de tal forma de ir desagregando en un par de niveles las posibles relaciones que podrían existir entre los elementos que la componen, los cuales en su conjunto darán soporte a la funcionalidad para la cual han sido diseñadas y se deberán desarrollar.

Este esquema no solamente ayuda a comprender un modelo de relación entre componentes determinados (que de hecho, podrían ser más de uno o podría ser uno completamente distinto), sino que además constituyen una base sólida para el desarrollo de la/s aplicación/es o la/s modificación/es a la/s existente/s.

En la vista del Modelo de Flujo de datos se podía intuir no solamente los componentes que hacen a la solución sino también algunas relaciones entre ellos, indicadas por líneas. Inclusive para algunos casos, se podía observar la direccionalidad de la relación mediante la utilización de flechas, lo que da una idea del origen de la relación y su destino. También se representaban otros tipos de relaciones con determinada funcionalidad, como las flechas direccionales que indicaban la dirección del flujo de datos, contemplando para el caso las tres alternativas posibles. Por último, en la vista mencionada anteriormente se podían ver representados y bien definidos ciertos elementos que a primera vista aparecían como primitivos. Si se comprendió así, precisamente era esa la finalidad.

La vista que sigue no es solo una adaptación de la vista Modelo de Flujo de Datos, sino que además constituye un incremento sobre la misma al contemplar la identificación con una nomenclatura determinada de ciertos elementos primitivos. Adicionalmente, podrían existir N incrementos sobre el mismo. Es necesario aclarar que en el gráfico y en su posterior desarrollo se omiten ciertas relaciones obvias, para dar prioridad a aquellas que tienen relevancia para la arquitectura que se está describiendo.

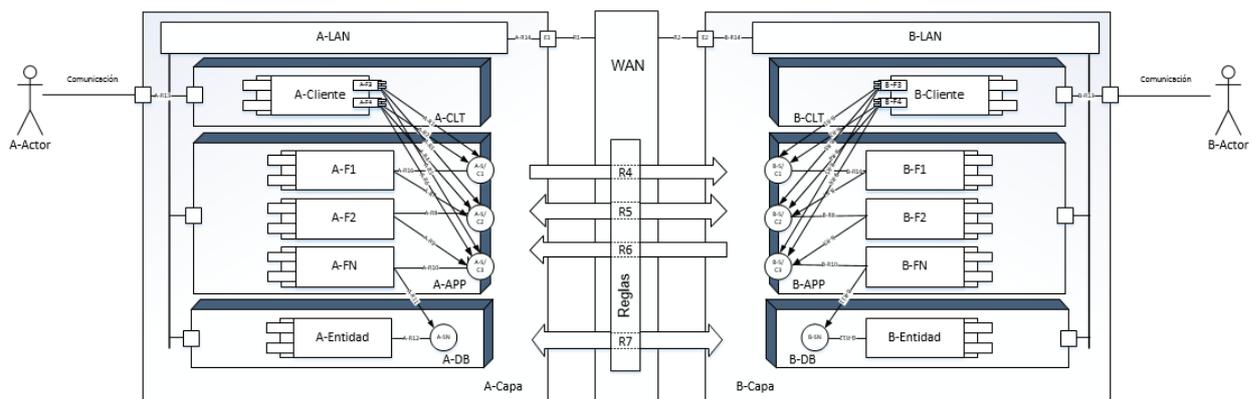


Fig. 37 – Arquitectura de TI: Vista de la Arquitectura CS/CS



Primitivas y Conectores

El catálogo de primitivas y conectores establece el catálogo de primitivas y conectores que se deberá utilizar para la construcción de las sentencias que representen tanto las relaciones entre componentes como su comportamiento y estado propios del modelo. Las tareas propias de la arquitectura alcanzan al diseño, construcción, administración y cambio de estas primitivas y conectores. Cualquier reutilización o instanciación que se realicen sobre las mismas corresponde al campo de la implementación y escapan a las tareas propias de la Arquitectura.

La definición de estos elementos puede realizarse antes, durante o después de la confección del modelo, pero es una tarea que necesariamente debe ser llevada a cabo para dar consistencia al diseño y favorecer la reutilización y flexibilidad de componentes.

A continuación se definirán una serie de primitivas y conectores bases para la construcción de sentencias que compondrán las relaciones entre los componentes que se identifiquen. Estas primitivas pueden representar acciones o palabras que representen el estado o el comportamiento de los componentes relacionados y pueden ser descompuestas dependiendo el nivel de detalle que se pretenda alcanzar en la descripción. A modo de ejemplificar, se utilizará un nivel de detalle que permita la comprensión del concepto sin que esto signifique mayor o menor nivel de abstracción.

Adicionalmente, los conectores operan sobre las primitivas y dan sentido lógico a una relación.

Catálogo de Primitivas	
Nombre	Descripción
<<Expone>>	Un elemento o componente expone un componente o conjunto de componentes que configuran una funcionalidad
<<Consume>>	Un elemento o componente consume un componente o conjunto de componentes que configuran una funcionalidad
<<Conecta con>>	Un elemento o componente conecta con un componente en una relación uno a uno
<<Disponible>>	Propiedad de un elemento
<<No Disponible>>	Propiedad de un elemento
<<Modo Operación>>	Define una determinada modalidad de operación
<<Sincrónico>>	Propiedad del elemento Modo de Operación
<<Asincrónico>>	Propiedad del elemento Modo de Operación
<<Sincroniza con>>	Define una actividad de sincronización bidireccional entre dos elementos
<<Datos Locales>>	Define datos que son generados localmente y necesitan ser replicados al entorno remoto
<<Datos Remotos>>	Define datos remotos que son generados remotamente y se necesitan tener disponibles localmente
<<Datos Compartidos>>	Identifica a datos que pueden ser generados tanto local como remotamente y que necesitan prevalecer uno sobre otro en base a determinadas reglas que necesitan ser definidas previamente, para su correcta sincronización e integridad de datos.



Catálogo de Conectores/Operadores	
Nombre	Descripción
IF ... THEN .. ELSE	Condición lógica “Si Entonces ...SI NO ...”
=	Operador lógico para determinar el valor de una propiedad de un elemento
:=	Operador para asignar valor a la propiedad de un objeto
<>	Operador de comparación equivalente a “no igual a”
AND	Operador lógico Y

Identificación de Componentes Primitivos

La identificación de Componentes primitivos documenta los componentes propios del modelo. Las tareas propias de la arquitectura alcanzan al diseño, construcción, administración y cambio de los mismos. Cualquier reutilización o instanciación que se realicen sobre ellos corresponde al campo de la implementación y escapan a las tareas propias de la Arquitectura.

Si bien la nomenclatura puede variar de aplicación en aplicación o de organización en organización, la adoptada es a los fines de ejemplificar e identificar claramente el elemento con el cual se está trabajando.

Brevemente:

- El prefijo “A-“ identifica a un elemento de la Capa A, en el caso la capa local;
- Lo que sigue al prefijo anterior identifica el tipo de elemento con el cual se está trabajando:
 - “Fx”, donde x corresponde a un número secuencial, identifica una funcionalidad determinada. Por ejemplo, el elemento “B-F1” se entiende como la funcionalidad 1 de la capa local. Adicionalmente, una funcionalidad puede estar compuesta por más de un componente y relaciones entre estos;
 - “Rx”, donde x corresponde a un número secuencial, identifica a una relación entre dos elementos. Para los casos donde los elementos “Relación” no tienen prefijo, significa que la relación es de alcance global;
 - “Ex”, donde x corresponde a un número secuencial, identifica un elemento cuyo estado es necesario conocer para un tipo determinado de relación que configure un comportamiento;
 - Un nombre determinado para el tipo de componente en base a su funcionalidad específica.

Es necesario aclarar que se pueden identificar tantos componentes como sea necesario al nivel de detalle que se considere pertinente para una correcta y consistente definición de la arquitectura. Sobre estas definiciones primitivas se basan las tareas de arquitectura y sientan las bases para las soluciones que se soportarán sobre los mismos.

Catálogo de Componentes	
Nombre	Descripción
WAN	Elemento primitivo que identifica a los medios de comunicación que se utilicen para conectar capas lógicamente separadas
E1	Elemento primitivo que conecta una capa a la WAN y cuyo estado es necesario



	conocer, para el ejemplo, para operar en Modo sincrónico o asincrónico.
E2	Elemento primitivo que conecta una capa a la WAN y cuyo estado es necesario conocer, para el ejemplo, para operar en Modo sincrónico o asincrónico.
R4	Elemento primitivo que representa una relación unidireccional entre dos componentes Cliente-Servidor/Servidor-Cliente, los cuales actúan como cliente uno y como servidor el otro o viceversa.
R5	Elemento primitivo que representa una relación bidireccional entre dos componentes Cliente-Servidor/Servidor-Cliente.
R6	Elemento primitivo que representa una relación unidireccional entre dos componentes Cliente-Servidor/Servidor-Cliente, los cuales actúan como cliente uno y como servidor el otro o viceversa.
R7	Elemento primitivo que representa una relación bidireccional entre dos componentes que no necesariamente puedan ser de tipo cliente o servidor. Para el modelado de la aplicación podría representar un actor más (por ejemplo otra aplicación o funcionalidad de otra aplicación).
Reglas	Elemento primitivo que representa el conjunto de reglas conformado por R4, R5, R6 y R7 y que tiene por fin mantener la consistencia de la información entre las N capas de las cuales trate la solución.
A-Capa	Elemento primitivo que representa la capa local desde la perspectiva del elemento A-Actor
A-LAN	Elemento primitivo que representa los activos de red local que permiten la comunicación entre los distintos elementos, desde la perspectiva de A-Actor
A-Actor	Usuario final que opera procesos de negocios, para cuya necesidad se diseñan las funcionalidades de la aplicación y cuya continuidad debe mantenerse.
A-Cliente	Elemento primitivo que idéntica la o las clase/s de Interfaz contra las cuales interactuar el elemento A-Actor
A-App	Elemento primitivo que identifica la o las clase/s de Control que componen la aplicación que soporta la operación de los procesos de negocios. Básicamente, allí se ejecutan las aplicaciones de misión crítica cuya continuidad es necesario asegurar.
A-DB	Elemento primitivo que representa a la o las clase/s de Entidad, donde residen los datos y cuya consistencia es necesario garantizar bajo situaciones de operación en modo sincrónico o asincrónico.
A-F1	Elemento primitivo que representa a la funcionalidad 1 específicamente
A-F2	Elemento primitivo que representa a la funcionalidad 2 específicamente
A-F3	Elemento primitivo que representa a la funcionalidad 3 específicamente
A-F4	Elemento primitivo que representa a la funcionalidad 4 específicamente
AF-N	Elemento primitivo que representa a la funcionalidad N específicamente
A-Entidad	Elemento primitivo que representa a las funcionalidades que brindan los sistemas gestores de bases de datos.
A-S/C 1	Elemento primitivo Servidor/Cliente desde la perspectiva de A-Capa, pieza fundamental que define la característica principal de la arquitectura y de cuya funcionalidad depende la operación continua de algún proceso crítico de negocio.
A-S/C 2	Elemento primitivo Servidor/Cliente desde la perspectiva de A-Capa, pieza fundamental que define la característica principal de la arquitectura y de cuya



	funcionalidad depende la operación continua de algún proceso crítico de negocio.
A-S/C 3	Elemento primitivo Servidor/Cliente desde la perspectiva de A-Capa, pieza fundamental que define la característica principal de la arquitectura y de cuya funcionalidad depende la operación continua de algún proceso crítico de negocio.
A-S N	Elemento primitivo que identifica servicios expuestos por el elemento primitivo A-Entidad, por caso una base de datos con un puerto en escucha y un listener que administra las peticiones.
B-Capa	Elemento primitivo que representa la capa local desde la perspectiva del elemento B-Actor. Capa Remota para el ejemplo.
B-LAN	Elemento primitivo que representa los activos de red local que permiten la comunicación entre los distintos elementos, desde la perspectiva de B-Actor
B-Actor	Usuario final que opera procesos de negocios, para cuya necesidad se diseñan las funcionalidades de la aplicación y cuya continuidad debe mantenerse. Para el ejemplo no necesariamente se supone la posibilidad de operar en forma continua al residir allí las aplicaciones consolidadas, por ejemplo, en un data center de clase mundial.
B-Cliente	Elemento primitivo que idéntica la o las clase/s de Interfaz contra las cuales interactuar el elemento B-Actor
B-App	Elemento primitivo que identifica la o las clase/s de Control que componen la aplicación que soporta la operación de los procesos de negocios. Es posible que cierta lógica de negocios se deba replicar a A-APP de alguna forma.
B-DB	Elemento primitivo que representa a la o las clase/s de Entidad, donde residen los datos y cuya consistencia es necesario garantizar bajo situaciones de operación en modo sincrónico o resincronizar al retomar la operación en modo sincrónico. Son los datos que se necesita tener disponibles para operar.
B-F1	Elemento primitivo que representa a la funcionalidad 1 específicamente
B-F2	Elemento primitivo que representa a la funcionalidad 2 específicamente
B-F3	Elemento primitivo que representa a la funcionalidad 3 específicamente
B-F4	Elemento primitivo que representa a la funcionalidad 4 específicamente
BF-N	Elemento primitivo que representa a la funcionalidad N específicamente
B-Entidad	Elemento primitivo que representa a las funcionalidades que brindan los sistemas gestores de bases de datos.
B-S/C 1	Elemento primitivo Servidor/Cliente desde la perspectiva de B-Capa, pieza fundamental que define la característica principal de la arquitectura y de cuya funcionalidad depende la operación continua de algún proceso crítico de negocio.
B-S/C 2	Elemento primitivo Servidor/Cliente desde la perspectiva de B-Capa, pieza fundamental que define la característica principal de la arquitectura y de cuya funcionalidad depende la operación continua de algún proceso crítico de negocio.
B-S/C 3	Elemento primitivo Servidor/Cliente desde la perspectiva de B-Capa, pieza fundamental que define la característica principal de la arquitectura y de cuya funcionalidad depende la operación continua de algún proceso crítico de negocio.
B-S N	Elemento primitivo que identifica servicios expuestos por el elemento primitivo B-

Trabajo Final de Grado

Arquitectura Cliente-Servidor/Cliente-Servidor en aplicaciones de Misión Crítica

Autor: Cristián Bircher – DNI: 22614222



	Entidad, por caso una base de datos con un puerto en escucha y un listener que administra las peticiones.
--	---

Identificación de Relaciones entre Componentes

La identificación de las relaciones entre componentes documenta las relaciones establecidas entre los distintos componentes del modelo. Las tareas propias de la arquitectura alcanzan al diseño, construcción, administración y cambio de las relaciones definidas en este catálogo. Cualquier reutilización o instanciación que se realicen sobre las mismas corresponde al campo de la implementación y escapan a las tareas propias de la Arquitectura.

Catálogo de Relaciones Primitivas	
Relación	Descripción
A-R1	Elemento primitivo que relaciona al componente A-F3 con el A-S/C 1
A-R2	Elemento primitivo que relaciona al componente A-F4 con el A-S/C 1
A-R3	Elemento primitivo que relaciona al componente A-F3 con el A-S/C 2
A-R4	Elemento primitivo que relaciona al componente A-F4 con el A-S/C 2
A-R5	Elemento primitivo que relaciona al componente A-F3 con el A-S/C 3
A-R6	Elemento primitivo que relaciona al componente A-F4 con el A-S/C 3
A-R7	Elemento primitivo que relaciona al componente A-F1 con el A-S/C 2
A-R8	Elemento primitivo que relaciona al componente A-F2 con el A-S/C 2
A-R9	Elemento primitivo que relaciona al componente A-F2 con el A-S/C 3
A-R10	Elemento primitivo que relaciona al componente A-FN con el A-S/C 3
A-R11	Elemento primitivo que relaciona al componente A-FN con el A-SN
A-R12	Elemento primitivo que relaciona al componente A-Entidad con el A-SN
A-R13	Elemento primitivo que relaciona al componente A-CLT con el A-Actor
A-R14	Elemento primitivo que relaciona al componente A-LAN con el componente WAN
B-R1	Elemento primitivo que relaciona al componente A-F3 con el B-S/C 1
B-R2	Elemento primitivo que relaciona al componente B-F4 con el B-S/C 1
B-R3	Elemento primitivo que relaciona al componente B-F3 con el B-S/C 2
B-R4	Elemento primitivo que relaciona al componente B-F4 con el B-S/C 2
B-R5	Elemento primitivo que relaciona al componente B-F3 con el B-S/C 3
B-R6	Elemento primitivo que relaciona al componente B-F4 con el B-S/C 3
B-R7	Elemento primitivo que relaciona al componente B-F1 con el B-S/C 2
B-R8	Elemento primitivo que relaciona al componente B-F2 con el B-S/C 2
B-R9	Elemento primitivo que relaciona al componente B-F2 con el B-S/C 3
B-R10	Elemento primitivo que relaciona al componente B-FN con el B-S/C 3
B-R11	Elemento primitivo que relaciona al componente B-FN con el B-SN
B-R12	Elemento primitivo que relaciona al componente B-Entidad con el B-SN
B-R13	Elemento primitivo que relaciona al componente B-CLT con el B-Actor
B-R14	Elemento primitivo que relaciona al componente B-LAN con el componente WAN



Construcción de Relaciones

El Catálogo de relaciones documenta las relaciones establecidas entre los distintos componentes del modelo. Las tareas propias de la arquitectura alcanzan al diseño, construcción, administración y cambio de estas relaciones. Cualquier reutilización o instanciación que se realicen sobre las mismas corresponde al campo de la implementación y escapan a las tareas propias de la Arquitectura.

Estas relaciones constituyen una base sólida y consistente para el diseño y desarrollo de la aplicación. A modo de ejemplo se resumen algunas de ellas otorgando funcionalidades genéricas al modelo:

Catálogo de Relaciones Primitivas	
Relación	Descripción
A-R1	A-F3 <<consume>> A-S/C 1
A-R2	A-F4 <<consume>> A-S/C 1
A-R3	A-F3 <<consume>> A-S/C 2
A-R4	A-F4 <<consume>> A-S/C 2
A-R5	A-F3 <<consume>> A-S/C 3
A-R6	A-F4 <<consume>> A-S/C 3
A-R7	A-F1 <<consume>> A-S/C 2
A-R8	A-F2 <<expone>> A-S/C 2
A-R9	A-F2 <<consume>> A-S/C 3
A-R10	A-FN <<expone>> A-S/C 3
A-R11	A-FN <<consume>> A-SN
A-R12	A-Entidad <<expone>> A-SN
A-R13	A-Actor <<conecta con>> A-CLT
A-R14	IF E1 = <<Disponible>> THEN A-LAN <<conecta con>> WAN AND <<Modo Operación>> := <<Sincrónico>> ELSE IF E1 = <<Disponible>> THEN <<Modo Operación>> := <<Asincrónico>>
A-R15	A-F1 <<expone>> A-S/C 1
R4	A-S/C 1 <<Conecta con >> B-S/C 1
R5	A-S/C 2 << Sincroniza con>> B-S/C 2
R6	B-S/C 3 <<Conecta con>> A-S/C 3
R7	A-SN <<Conecta con>> B-SN
Reglas	IF <<Datos Locales>> THEN R4 IF <<Datos Compartidos>> THEN R5 IF <<Datos Remotos>> THEN R6 IF <<Datos Locales>> = <<Datos Remotos>> THEN R7

Las relaciones de la capa remota podrían tener idéntica conformación que las relaciones de la capa local, aunque no necesariamente tienen que ser iguales. Para el caso se omiten las mismas, ya que en caso de ser iguales sería tan simple como copiar y pegar, y en caso de ser distintas escaparían al alcance específico del trabajo.

Es necesario aclarar que las relaciones identificadas son a modo de ejemplo y podrían variar de aplicación en aplicación. Persiguen un fin genérico de situaciones comunes que se pueden presentar en



cualquier entorno de aplicación de Misión Crítica y corresponden a la funcionalidad Cliente-Server / Cliente-Servidor que se pretende describir.

Como se podrá observar, existen relaciones genéricas heredadas de la Arquitectura Cliente-Servidor que sirven como modelo de referencia base para la construcción de la Arquitectura. Sin embargo, a medida que se va avanzando en el análisis de las relaciones se comienzan a vislumbrar con mayor o menor detalle aquellas que hacen específicamente a este tipo de arquitecturas y que constituyen la forma de trabajo de la misma y su especial particularidad frente al resto de las alternativas similares.

A excepción de la relación identificada como A-R14, Las relaciones genéricas A-R1 a A-R15 son relaciones típicas de cualquier arquitectura cliente-servidor donde componentes que cumplen distintas tareas configuran distintas funcionalidades afines a los objetivos de la solución.

Sin embargo, la relación A-R14, define conceptos propios de la Arquitectura bajo análisis, que tienen que ver con los modos de operación sincrónico para situaciones de operación en línea con los sistemas centrales, y el modo de operación asincrónico para situaciones en modo de operación fuera de línea de los sistemas centrales. Como se podrá correlacionar, esto es compatible con lo ejemplificado en las vistas del “Modelo de Operación en Modo Asincrónico” y cualquiera de las otras vistas (para la operación en modo sincrónico).

Adicionalmente, las relaciones R4, R5, R6 y R7 definen tres situaciones que se encontrarán en cualquier situación que se aplique la arquitectura cliente-servidor / cliente-servidor, ya que es factor crítico de éxito determinar que tratamiento se le da a los datos, clasificándolos según la capa donde se originan y la necesidad de disponer de los mismos en una u otra capa:

- Datos originados localmente y que necesitan ser dispuestos remotamente para otros procesos, como por ejemplo número de transacciones, código de región de la transacción, hora local etc.;
- Datos originados remotamente y que necesitan ser dispuestos localmente para procesos críticos, como pueden ser tablas maestras de clientes, tablas de control, numeración etc.
- Datos originados en cualquiera de las dos capas que necesitan ser sincronizados de cierta forma siguiendo determinadas reglas preestablecidas.

Es justamente la relación REGLAS la que configura estas últimas. En el ejemplo, las reglas para la sincronización utilizan lógica de las relaciones previamente establecidas para los componentes involucrados en cualquiera de las situaciones. En la práctica, es posible implementar este tipo de comportamiento mediante:

- Lógica de programación en los servicios que escuchan peticiones internas (servicios locales) y realizan peticiones a servicios remotos (clientes remotos). Para el ejemplo, los componentes Servidor-Cliente A-S/C 1, 2 y 3;
- Utilización de dblinks para escritura remota mediante la utilización de controladores ODBC, OLEDB, JDBC o cualquier otro tipo de conexión similar dependiendo la tecnología utilizada;
- Utilización de vistas materializadas. Para el ejemplo, R6 establece la materialización de vistas en la dirección Capa Remota → Capa local que haría disponibles localmente datos remotos en forma local en tiempo real;
- Herramientas de sincronización propias de cada Sistema Gestor de Base de Datos. Para el ejemplo podría estar representada sobre los componentes A-Entidad y B-Entidad, funcionalidad que podría ser definida mediante la relación R7;

- O una combinación de todo lo mencionado anteriormente.

Funcionalidades

La Arquitectura Cliente-Servidor / Cliente-Servidor consiste entonces en el diseño, construcción, administración y cambios de los elementos primitivos que configuren las funcionalidades necesarias para la operación de los procesos de negocio. Aplicado específicamente a las aplicaciones de Misión Crítica, estas funcionalidades consisten en configurar una capa intermedia entre los sistemas centrales y la red local, de tal forma que, desde la perspectiva del cliente, se trabaje siempre contra la capa intermedia, actuando esta como Servidor, mientras que desde la perspectiva de los Sistemas Centrales esta capa intermedia es cliente para las funcionalidades que se brinden desde la capa remota.

Adicionalmente, las funcionalidades que se deberán diseñar y desarrollar mediante la utilización y combinación de los elementos primitivos ejemplificados en los apartados anteriores, alcanzan no solamente a los componentes Servidor-Cliente que se identifiquen de acuerdo a las necesidades del negocio, sino también a los modos de operación que permitan operar bajo situaciones en las cuales los Sistemas Centrales no estén disponibles.

Esto supone diagramar la capacidad de las aplicaciones de tal forma que:

- puedan soportar la operación continua de los procesos críticos en forma fuera de línea respecto de los sistemas centrales;
- puedan soportar la operación continua de los procesos críticos en línea con los sistemas centrales;
- puedan re sincronizar la información en la dirección que el flujo de datos lo amerite en base a las características de los procesos, teniendo como objetivo mantener la integridad de los datos.

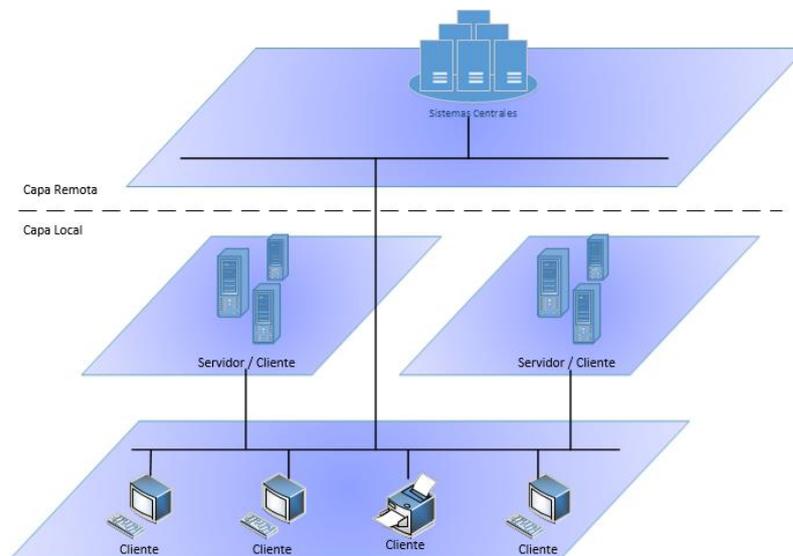


Fig. 38 – Arquitectura de TI: Vista de la Arquitectura CS/CS



En un nivel de análisis más profundo, los arquitectos no solamente deberán tener en cuenta estos factores sino también, no perder de foco la infraestructura con la que cuentan, su diseño, capacidad y necesidades anexas. La solución a diseñar y desarrollar no solamente debe tener una buena relación costo / beneficio sino, además de cumplir con las expectativas para las cuales se decide realizar el esfuerzo, ser lo más transparente posible para el usuario e impactar lo menos posible con una adecuada gestión del cambio.

Un diseño consistente debe tener en cuenta TODOS los factores anexas que escapan al alcance específico de lo que es la Arquitectura de TI, en una tarea interdisciplinaria que requiere mucho esfuerzo e iteración con otras áreas de servicios para lograrlo. De nada serviría implementar un diseño consistente de la Arquitectura Cliente-Servidor / Cliente-Servidor si la configuración eléctrica de los puestos de trabajo no pueden soportar un simple corte de energía.

Resumen

En resumen de cuentas, la aplicación de una Arquitectura Cliente-Servidor / Cliente-Servidor aporta valor a bajo costo mediante la reutilización de componentes ya existentes en la Infraestructura de TI, aportando simplicidad y flexibilidad a las operaciones del Negocio.

El diseño basado en principios organizacionales y de TI provee un alto alineamiento a los estándares de TI y procesos de negocios, conformando una evolución de la Arquitectura Cliente-Servidor adaptada a Aplicaciones de Misión Crítica.

La formalización de la Arquitectura basada en distintas Vistas y adaptada a los distintos puntos de vista de los stakeholders, brinda altas capacidades para la comprensión del modelo, el entendimiento entre las partes y aporta valiosa información para el desarrollo de las aplicaciones que sobre esta arquitectura se apliquen.



Capítulo

10

Aplicación Práctica

En este capítulo se posiciona al lector respecto de la experiencia adquirida y algunas lecciones adquiridas en la aplicación práctica de la Arquitectura Cliente-Servidor / Cliente-Servidor, en base al proyecto que le dio origen. Se reseñan particularidades propias del Ingenio Azucarero, junto con los objetivos perseguidos, los alcances, problemas existentes antes de la implementación y motivos que justificaron su desarrollo.

Adicionalmente se realiza una breve descripción de algunas de las alternativas consideradas, y factores que motivaron el diseño de una nueva arquitectura aplicable a la infraestructura existente que se integrara con las aplicaciones centralizadas. Finalmente, se da detalle de algunas soluciones tecnológicas del mercado y se justifican algunas de las razones de su no adopción.

Breve Reseña – Ingenio Azucarero

La actividad principal del Ingenio es la producción de azúcar, proceso en el cual también se elabora y comercializa la melaza, la cual se utiliza en la producción de alcohol y levaduras. Otra actividad no menor relacionada es la comercialización de las fibras de bagazo, materia prima para la producción de papel.

La caña de azúcar, proveniente de los distintos frentes de cosecha que llega al Ingenio, es analizada para determinar su calidad (rendimiento porcentual de azúcar, pureza etc.), para lo cual se utiliza un moderno sistema de extracción de muestras, conocido como Core Sampler. El objetivo es lograr que solo ingrese a la molienda materia prima en condiciones óptimas.

La elaboración del azúcar consta básicamente de los siguientes procesos:

- cosecha de la caña de azúcar en fincas propias o de terceros;
- traslado a planta fabril por parte de transportes propios o de terceros y descarga;
- preparación de la caña;
- empaque y almacenamiento para su posterior distribución.

La preparación de la caña consiste en un complejo proceso para dividirla finamente, sin pérdida de jugo. Posteriormente se extrae la sacarosa presente en la caña preparada, tarea que se realiza mediante lavados sucesivos con agua caliente o jugo diluido en un difusor.

Como dato adicional, el Ingenio produce más de 135 mil toneladas de azúcar al año, en forma casi ininterrumpida durante toda la temporada de zafra. Justamente, uno de los objetivos del trabajo fue alcanzar la operación ininterrumpida, o al menos, que las interrupciones no sean a causa de las herramientas de Software utilizadas.



La temporada de zafra en Tucumán inicia alrededor de Abril / Mayo. A partir de allí y hasta que termina la misma el trabajo no para en intensidad. El ritmo de ingreso de la caña de azúcar es incesante. Sólo a título informativo, los siguientes números permitirán dimensionar el volumen de la actividad e imaginar lo que supondría en materia económica y operativa para la producción por cualquier cuestión:

- 450 toneladas / hora: es la cantidad mínima de necesaria de ingreso de Caña de Azúcar para mantener el proceso productivo en funcionamiento;
- 30 toneladas: es el carga completa máxima promedio de un camión con chasis y acoplado;
- $450\text{tn} / 30\text{tn} = 15$ camiones con chasis y acoplado por hora es la cantidad mínima de camiones que deberán recorrer el circuito completo de abastecimiento:
 - Extracción de muestra y análisis de calidad de la muestra;
 - Espera en canchón;
 - Acceso a Planta;
 - Pesada carga completa;
 - Descarga;
 - Pesada de tara.
- $60 \text{ minutos} / 15 \text{ camiones} = 4$ camiones por minuto: es el ritmo promedio con el cual deberán ingresar los camiones con chasis y acoplado para sostener el proceso de producción.

Con estos números queda bastante claro que el más mínimo fallo provocaría la acumulación de camiones, y superado el pulmón que otorga el canchón (playa de estacionamiento donde los camiones que superaron los análisis de calidad esperan ser autorizados para el ingreso), la acumulación de camiones esperando ingresar se convertiría en pocos minutos en una gran congestión de camiones, y por supuesto una parada de la producción y todos los costos que la no operación significa.

A esto también es necesario agregarle otros condimentos propios de la actividad y la región y de la forma de operar de la compañía:

- Un gran porcentaje del ingreso de la materia prima se realiza mediante los tradicionales “carros”, que heredaron el nombre de las tradicionales “carretas cañeras”, tan identificadas con lugares cercanos como la vecina localidad de Simoca, por ejemplo. Estos “carros” están compuestos por un tractor y una cantidad variable de acoplados, que van desde uno a cinco o seis, conformando verdaderos trencitos de carga que suelen inundar la ruta 38 en temporadas de zafra, transportando la caña desde las plantaciones a los distintos Ingenios de la zona;
- Mesas de alimentación: el Ingenio posee dos mesas de alimentación de caña de azúcar a la planta, una para caña entera, otra para caña molida;
- Canchón o Playa de Espera: el Ingenio cuenta con un canchón en el cual los camiones que ya han sido identificados en el sistema y cuya muestra está en análisis, esperen su turno de llamado para ingreso. Estando completo este espacio, el Ingenio tiene una capacidad de operación de aproximadamente una hora antes de parar la producción, en el caso que se corte el ingreso de camiones;



- La gestión de turnos de ingreso no está alcanzada por la solución, ya que existe una solución local;
- Por definición, las funcionalidades que no alcanzan al circuito crítico operan en forma remota desde el Sistema de Gestión de Cañas actual;
- Algunas funcionalidades no críticas se incorporaron a la solución para poder dar continuidad al proceso, como por ejemplo la gestión de choferes. Por caso ejemplo, un camión con un chofer nuevo, no debería esperar a que se reestablezcan los servicios centrales para poder ingresar la caña que transporta;

Por otro lado:

- Una parada por mantenimiento o imprevista del sistema transaccional supondría una parada de la producción, lo cual supone un alto impacto operativo y económico para la compañía;
- Si bien en temporadas de zafra el proceso de producción opera 7x24 y bajo condiciones normales se cuenta con ciertos niveles de stock de seguridad, es necesario mencionar que la producción de azúcar del Ingenio es materia prima para la producción de las demás plantas productivas de la compañía, de allí también su criticidad “hacia adelante” en el resto de los procesos productivos de la compañía;

Adicionalmente, el grupo cuenta con una configuración de sus sistemas altamente centralizada en un datacenter de clase mundial, hosteando allí las aplicaciones que permiten la operación de sus negocios, incluyendo hasta ese momento los del Ingenio; pero por otro lado, sus dependencias están geográficamente muy distribuidas, con 42 plantas productivas en Latinoamérica, y oficinas comerciales en todo el mundo.

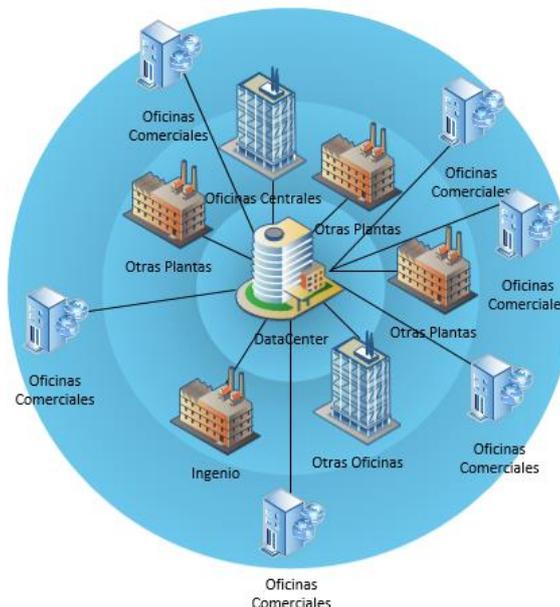


Fig. 39 – Infraestructura altamente centralizada con dependencias geográficamente distribuidas



Objetivos del Trabajo realizado

- Garantizar el normal abastecimiento de caña de azúcar al Ingenio;
- Minimizar los riesgos de una parada por inconvenientes en el sistema;
- Independizar a las funcionalidades que componen el circuito crítico de los inconvenientes que pudieran no disponibilizar el Sistema Central.

Alcances

El proyecto alcanzó a:

- Procesos que componen el circuito crítico de la caña de azúcar, y que permiten el normal abastecimiento de esta materia prima al Ingenio;
- Funcionalidades anexas que deban ser consideradas para mantener el alcance mencionado en el punto anterior.

Justificación

Analizadas detenidamente las alternativas en conjunto con la Gerencia de Sistemas y la Gerencia del Ingenio, se llegó a la conclusión de que dado el contexto tecnológico en particular en el cual se haya ubicado el complejo y la situación en la que se encuentra enmarcada la compañía, no existe una solución mejor que pueda no solamente alcanzar los objetivos propuestos, con un bajo costo y bajo impacto en su utilización por parte de los usuarios, sino que además, brinde posibilidades ciertas y concretas de mantenerse como solución en un futuro, cuando los planes de actualización se concreten.

Por otro lado, la solución propuesta cumple con el criterio de “mientras tanto”, lo que ayuda a la estrategia tecnológica a esperar el momento oportuno para el cambio, cumpliendo con los objetivos para los cuales se lo ha propuesto.

Adicionalmente, se concluyó que ningunas de las alternativas propuestas son económicamente viables.

Problemas

- El Sistema de Cañas era accedido en forma remota:
 - Causa: la obsolescencia del equipo AS/400 anteriormente hospedado localmente y la imposibilidad de conseguir partes de repuesto, obligaron a migrar el Sistema de Cañas a un nuevo AS/400 hospedado remotamente;
 - Consecuencias: la operación del Sistema de Cañas quedó altamente expuesto a cortes en las comunicaciones;
 - Mitigación: contratación de un enlace secundario provisto por la misma empresa, pero que llega a través de un troncal de fibra óptica distinto al del enlace primario. Sin embargo es susceptible a cortes, como el ocurrido durante Abril de 2013 (18 hs. de corte);



- Alternativas: la actualización tecnológica del equipo AS/400 anteriormente hosteado localmente fue descartada de plano por cuestiones de costos y planes de actualización del Sistema de Cañas en el mediano plazo.
- La infraestructura local es inadecuada o escasa y existe solo un proveedor con capacidad de proveer enlaces de comunicaciones, con lo cual la dependencia es total;
 - Causa: escasa demanda de servicios en la zona geográfica en la cual se encuentra el Ingenio;
 - Consecuencias: Solamente una compañía proveedora de enlaces de comunicaciones llega hasta la zona;
 - Mitigación: al tratarse de cuestiones externas, la mitigación propuesta fue expuesta en el problema anterior;
 - Alternativas: gestión a largo plazo para que Claro llegue a la región con una amortización de la inversión conveniente tanto para la empresa como para Claro.
- La infraestructura remota es compartida y por lo tanto susceptible a cortes por mantenimiento programados o no:
 - Causa: dentro del modelo centralizado actual, se consolidaron en un mismo equipo AS/400 las aplicaciones basadas en dicha infraestructura;
 - Consecuencias: los cortes por mantenimiento programado o no en temporada de zafra son de alto impacto para la operatoria del Ingenio;
 - Mitigación: se implementó una alta disponibilidad hecha en casa. Sin embargo las maniobras para realizar un switch over generalmente exceden a las ventanas de mantenimiento programadas, y más aún si se suman los tiempos de switch back.
 - Alternativas: las soluciones de alta disponibilidad de terceros, si bien son más eficientes y automáticas tienen tiempos de switch importantes y además son costosas.

Alternativas

Por un lado, la idea surge como consecuencia de arquitecturas similares observadas en aplicaciones que la compañía fue sumando a su catálogo de soluciones. Por otro lado, la solución a implementar requería ideas creativas e innovadoras que integraran no solo conocimientos sino también tecnologías disponibles, principalmente ante la imposibilidad de aplicación inmediata de las siguientes alternativas analizadas:

Plan de Actualización Tecnológica:

Durante el año 2012 el Grupo finalizó el proyecto de Renovación Tecnológica para los Negocios de Argentina. Este proyecto, que se inició en el año 2009 tuvo como objetivo redefinir, actualizar, simplificar e integrar los principales procesos operativos del grupo, basados en el funcionamiento estándar de la última versión disponible de JD Edwards. En este marco, en 2012 se terminó de incluir las operaciones logísticas y comerciales de consumo masivo y, entre octubre y diciembre se incorporó la gestión de todos los procesos del Negocio Packaging.



Durante el 2013 y como parte del mismo proceso se extendió el modelo a los países con filiales comerciales con el objetivo de contar con una herramienta integrada y moderna para gestionar su operación, generando sinergia por integración de procesos y centralización de datos y una mayor capacidad de replicar soluciones modelo a todos los Negocios.

Además, se actualizó la plataforma de comunicación interna, para mejorar la performance y la gestión del conocimiento de los colaboradores. Junto con esto se desarrollaron diversas iniciativas de movilidad para tomar valor en los procesos de Negocio de las tecnologías disponibles.

El negocio de la producción de azúcar es un negocio muy particular que requiere de soluciones especializadas en la materia. Dentro del Plan de Trabajo macro se prevé el análisis de soluciones alternativas al Sistema de Cañas para el año 2015 y una hipotética implementación para el año 2016. Sin embargo el desempeño del actual Sistema es tan bueno que la prioridad para avanzar en tal sentido es baja.

En este contexto, la solución a implementar no solamente constituye un “mientras tanto” dentro del Plan Macro, sino que además, entendiendo que el esquema final estará acorde al modelo corporativo y por ende manteniendo los problemas enumerados anteriormente, tratado cuidadosamente, con algunos ajustes e interfaces de por medio, se podría mantener para asegurar los objetivos de este proyecto.

Actualización de Hardware:

Si bien las soluciones de hardware actuales son mucho más versátiles, la solución de IBM que reemplaza a los viejos y nobles AS/400, iSeries o System-i, siguen teniendo un alto costo por varias sencillas razones particulares al caso que nos ocupa:

- Al menos se necesitan dos equipos Power Systems: uno productivo y otro como contingencia;
- El precio de las licencias de software es altísimo, y además se deberían multiplicar por dos;
- Se necesita mano de obra especializada para el soporte técnico de hardware, por lo cual se requeriría un contrato de soporte;
- IBM no garantiza la compatibilidad hacia atrás de los compiladores Cobol y RPG, lo que obligaría a revisar cada uno de los desarrollos del Sistema de Cañas;
- Volviendo a la alternativa anterior, se estaría saliendo del modelo centralizado.

Alta disponibilidad

Como solución a la no disponibilidad del Sistema Central, una solución de alta disponibilidad cubriría el problema en ventanas de mantenimiento del mismo, programadas o no. Sin embargo sería ineficaz ante una caída del enlace, con lo cual se hace inevitable pensar en una solución local.

A pesar de ello y a los fines de tomar acción en forma proactiva, las tareas que se desarrollaron a principios del 2012 en esta materia, contemplaron a grandes rasgos:

- Traslado del Sistema de Cañas desde el AS/400 productivo en el Ingenio a un AS/400 productivo, ubicado físicamente en un data center que la firma IBM posee en Martínez, Provincia de Buenos Aires;
- A los fines de minimizar el impacto en las aplicaciones productivas y segmentar los servicios, el Sistema de Cañas productivo fue implementado en una partición lógica de AS/400 de Test;



- Desafectación del equipo AS/400 del Ingenio del servicio activo ;
- La alta disponibilidad del Sistema de Cañas fue implementada contra la partición lógica de AS/400 de Producción, ubicada físicamente en otro data center del mismo complejo tecnológico de la firma IBM.

Arquitectura Cliente-Servidor / Cliente-Servidor vs. Otras Arquitecturas

La posibilidad de que existan inconvenientes que desconecten al Ingenio del resto de la organización implícitamente planteaba la necesidad de trabajar en forma autónoma, con lo cual la solución a aplicar exigía características en tal sentido.

Al mismo tiempo se necesitaba alimentar a los sistemas centrales con la información en tiempo real que se producía durante la operación del Ingenio, con lo cual bajo situaciones normales esto exigía cierta sincronización con los mismos.

Varios factores confluyeron para el diseño y desarrollo de una solución adecuada;

- Imposibilidad de encontrar aplicaciones enlatadas que contemplaran el 100% de las necesidades;
- Características particulares de la operación del Ingenio;
- Características y Topología de la Infraestructura existente;
- Estándares para la tecnologías de la información adoptados por la organización;
- Integración con los Proyectos Tecnológicos en curso;
- Conocimiento interno de las operaciones;
- Y obviamente, cuestiones de presupuesto, alcances y fundamentalmente tiempo, ante la inminente zafra 2014.

En contraste con cualquier otra arquitectura, la Arquitectura Cliente-Servidor / Cliente-Servidor proporciona un marco de referencia aplicable a Infraestructura existente, para el desarrollo de aplicaciones fundamentalmente de misión crítica, aunque podría ser aplicable a otro tipo de soluciones. Otros tipos de arquitectura, y debido a los factores mencionados anteriormente, hubieran precisado de un gran esfuerzo de adaptación, adquisición de conocimientos, tiempos para integrar, incorporación de nuevos elementos, y cuestiones no contempladas que hubieran puesto en riesgo el proyecto en cuanto a tiempos, costos y alcances, sin cubrir totalmente las expectativas de los clientes internos.

Justamente, uno de los logros más importantes de la implementación fue haberla logrado en forma prácticamente transparente para los usuarios, ya que la adaptación de las interfaces gráficas respecto de las existentes (basadas en AS/400) fue mínima, y el cliente interno solo encontró colores donde antes había solo pantallas verdes y negras.

El diseño de una nueva arquitectura Cliente-Servidor / Cliente-Servidor que se integrara perfectamente a la infraestructura y aplicaciones existentes, surge como consecuencia de todos estos factores, y proporciona un modelo conceptual no solo para el desarrollo de soluciones sino para la integración de las mismas con el resto de las aplicaciones que componen el portfolio de servicios de cualquier organización.



Solución Propuesta

Los factores intervinientes que dieron origen a la idea inicial son muchos, pero se pueden destacar los siguientes:

- Obsolescencia tecnológica de la solución actual;
- Plan de Actualización Tecnológica en el mediano plazo;
- Costos y Necesidades de Mantenimiento de Hardware;
- Operación remota de la solución actual;
- Disponibilidad de las herramientas tecnológicas para desarrollar la idea;
- Necesidad de abastecer de caña de azúcar al Ingenio en forma ininterrumpida, manteniendo los requisitos de calidad, el ritmo de abastecimiento y el flujo de información necesaria;
- Escasas posibilidades de la Infraestructura local.

A grandes rasgos, la solución implementada funciona a modo de una capa de software local que cubre las funcionalidades de lo que, en un análisis previo que escapa a este trabajo, se definió como circuito crítico, es decir las funcionalidades mínimas que se requieren para el abastecimiento normal de la caña de azúcar al Ingenio. Estas contemplan:

- Actividades que se desarrollan en el laboratorio (Core Sampler):
 - Ingreso de la caña de Azúcar;
 - Extracción de Muestra;
 - Asociación de Muestras con contratos existentes;
 - Asociación de Muestras con el cosechero correspondiente;
 - Asociación de Muestras con el transportista/chofer que corresponda;
 - Registro de resultados de los análisis de laboratorio que se le efectúan a la muestra y que permiten su ingreso al Ingenio.
- Actividades que se desarrollan en la portería:
 - Concretamente, luego que el transportista espere en Canchón, en base a los resultados de los análisis se permite el acceso para descarga, el acceso para desecho, o se rechaza el ingreso, dependiendo de distintas condiciones;
 - Adicionalmente, se deberán contemplar casos de excepción en donde bajo determinadas condiciones, muestras que no superen los análisis podrían ingresar para descarga, previa autorización de la Gerencia y previo tratamiento para mejorar la calidad, con algún porcentaje de trash.
- Actividades que se desarrollan en la balanza:
 - Pesada Manual de Carga Completa;
 - Pesada Automática de Carga Completa;
 - Pesada Manual de Tara;
 - Pesada Automática de Tara;
 - Carga de porcentaje de trash a descontar por número de muestra.
- Actividades de Administración:

- Alta de usuarios en base al directorio activo;
- Asignación de Roles;
- Alta de Temporada de Zafra.
- Sincronización de datos para casos de paso asíncrono a síncrono.

De esta forma, el usuario final trabaja contra la mencionada solución de software para las tareas que componen el circuito crítico, y para el resto de las funcionalidades, trabaja en forma remota contra el Sistema de Cañas tradicional (Sistema Central), hosteado en un Data Center de IBM, en Martínez, Provincia de Buenos Aires.

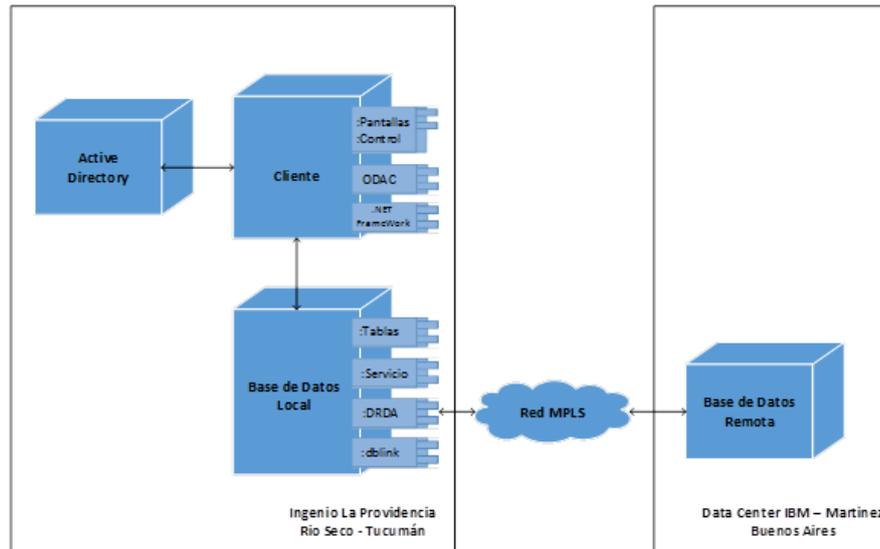


Fig. 40 – Vista de Alto Nivel de la Arquitectura de la Solución

En caso de contingencia, ya sea por no disponibilidad del Sistema Central, por cuestiones de comunicaciones o por lo que fuera, al tratarse de una solución local, el usuario continúa transaccionando y por lo tanto se podrá mantener el normal abastecimiento de la caña de azúcar.

Ver Anexo 1: “Anexo 1 - Pre-Grado - Ante Proyecto + Formulario B.doc”

Ver Anexo 2: “Anexo 2 - Pre-Grado - Plan de Trabajo.mpp”

Otras herramientas de mercado

Las particulares características de operación de un Ingenio Azucarero, sumado a la existencia de sólo 15 Ingenios de tales dimensiones (todos con características operativas distintas), hacen que sea imposible encontrar una herramienta de software que se ajuste perfectamente a los procesos.

De hecho, la solución utilizada hasta el momento es un viejo desarrollo propio y la compañía se encuentra en un proceso de análisis de herramientas de software donde en todos los casos existen procesos que tienen que ajustarse al software, con escasas posibilidades de que el software se ajuste a los procesos por cuestiones de estandarización e integración con el resto de las aplicaciones que conforman el portfolio de aplicaciones de la organización.

A continuación se detallan algunas de esas herramientas las cuales, como se podrá deducir, no están pensadas para un Ingenio Azucarero directamente, sino que cubren una necesidad específica que puede ser utilizada por la herramienta de software que se aplique.

Mimix

Mimix es una herramienta para sistemas basados en AS/400 que permite la replicación en tiempo real de datos, objetos del sistema y aplicaciones desde una máquina source a una máquina target, mediante la utilización de diarios y receptores de diarios, una especie de objeto propio de la plataforma que permite almacenar los cambios, a modo de re-do logs propios de cualquier sistema gestor de bases de datos.

Esta herramienta permite Alta Disponibilidad de las aplicaciones y continuidad de los negocios, asegurando una rápida recuperación ante una contingencia, realizando en forma automática o manual el pasaje de la operación desde la máquina source a la máquina target donde quiera que físicamente esta se encuentre, dando vuelta automáticamente el sentido de la replicación cuando esto ocurre.



Fig. 41 – Arquitectura Básica de Mimix

Una vez que la máquina source está nuevamente disponible y la información correctamente resincronizada es posible volver atrás en forma automática o manual, intercambiando los roles nuevamente (source x target o target x source).

Adicionalmente cuenta con funcionalidades de auditoría, reporting, y administración que permiten personalizar la instalación a las necesidades del negocio y a las características de las aplicaciones.



```
MIMIX Intermediate Main Menu                                     System: AS1
MIMIX
Select one of the following:
  1. Work with data groups                                     WRKDG
  2. Work with systems                                       WRKSYS
  3. Work with messages                                     WRKMSGLOG
  4. Work with monitors                                     WRKMON
 11. Configuration menu
 12. Compare, verify, and synchronize menu
 13. Utilities menu
 31. Product management menu                                LAKEVIEW/PRDMGT
                                                                Bottom
Selection or command
====>
F3=Exit  F4=Prompt  F9=Retrieve  F21=Assistance level  F12=Cancel
(C) Copyright Lakeview Technology Inc., 1990, 2007.
1  >> 21/7
```

Fig. 42 – Consola de Administración de Mimix

Con amplia experiencia en la solución producto de años de utilización, la compañía estaba en condiciones de asegurar que cumplía con las expectativas que se esperaba de una solución para el Ingenio. Sin embargo la mudanza de las aplicaciones al data center centralizado sumado a la baja del ya obsoleto equipo que se tenía en el Ingenio, exigían la compra de un nuevo equipo. Esto sumado al conocimiento de los costos de las licencias, hicieron que la alternativa apenas se mencionara como posibilidad.

Fuentes: www.visionsolutions.com

Oracle Golden Gate

Oracle Golden Gates es una herramienta con un perfil completamente distinto al de Mimix, ya que fue concebida como una herramienta de integración, como evolución a anteriores productos como por ejemplo Oracle Transparent Gateway y Oracle DRDA, herramientas para la integración en ambientes heterogéneos, que finalmente se terminaron utilizando como parte de la solución para el Ingenio Azucarero.

Sin embargo, la posibilidad de mantener una copia fresca de los datos en ubicaciones remotas sumado a la posibilidad de establecer reglas para sincronización y la direccionalidad del flujo de datos, más la posibilidad de sincronizar bases de datos de distintas tecnologías hacen de Golden Gate un producto sumamente versátil y potente a la hora de brindar este tipo de soluciones.

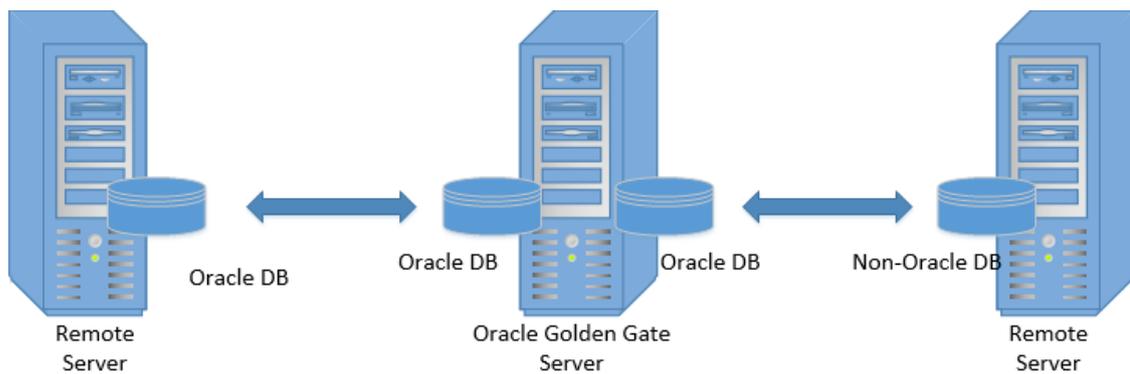


Fig. 43 – Oracle Golden Gate: Configuración Básica

Ventajas:

- Reduce los costos de infraestructura y favorece la disponibilidad de datos en forma remota y en el formato que se lo necesite;
- Asegura la integridad de transacciones;
- Posee un mecanismo de check-pointing, para el caso de interrupciones y cortes de servicio, a fin de continuar con el movimiento de datos desde dónde se interrumpió reduciendo el riesgo de pérdida de datos que puede afectar la operación del negocio;
- Soporta una amplia variedad de bases de datos;
- Permite establecer la direccionalidad del flujo de datos para distintos tipos de topología;
- Se integra con Oracle GoldenGate Veridata para la comparación de datos en tiempo real entre bases de datos heterogéneas sin afectar la disponibilidad;

Si bien el producto no existía como tal al momento del proyecto para el Ingenio azucarero, un par de cuestiones habría hecho caer la posibilidad concreta de utilizarlo:

- Altos costos de implementación;
- Conocimiento técnico no existente en la organización;
- Complejidad de configuración;
- A diferencia de Mimix que también replica datos y objetos del sistema (como pueden ser perfiles de usuario, dispositivos de impresión, pantallas, ejecutables etc.), Golden Gate solo se ocupa de la capa de datos.

Fuentes: www.oracle.com

Conclusiones

Finalizada la primera temporada de Zafra la solución demostró cualidades innegables de facilidad de uso. Una de las razones de ello fue que el diseño de las interfaces fueron pensadas lo más parecidas al sistema anterior, de forma tal de facilitar su adopción.



Al mismo tiempo y durante un par de oportunidades, se presentaron problemas en la comunicación con los sistemas centrales, sin que el usuario percibiera tal inconveniente, al menos en relación a los procesos de misión crítica. Mientras duró el corte, la producción fue normal.

Las funcionalidades desarrolladas y los servicios expuestos cumplieron a la perfección las tareas para las cuales fueron diseñadas. La aplicación fue capaz de operar en modo asincrónico cuando se perdió comunicación con los sistemas centrales y re sincronizó correctamente la información una vez retomada la operación sincrónica.

En definitiva, la aplicación demostró tener un diseño con una relación costo / beneficio extremadamente alta, con lo cual se convirtió en uno de los factores de éxito de la temporada de zafra 2014.



Capítulo

11

Conclusiones

Para cualquier organización, las posibilidades de innovar sobre la arquitectura existente son innumerables y dependen no solamente del tipo de solución que se esté implementando, sino también de la necesidad de hacerlo. Sin embargo, lograrlo en forma transparente, con bajo impacto y una relación costo / beneficio extremadamente alta supone mucho más que un desafío.

Lamentablemente en el ambiente de TI de hoy en día pareciera ser un axioma el hecho de que las soluciones más caras tienen que ser las mejores, o que suponen un esfuerzo menor su implementación al ser un paquete que se instala y queda funcionando casi mágicamente. Si bien hay excepciones que confirman la regla, las soluciones que permiten dar continuidad al negocio escapan al costo económico que los administradores están dispuestos a asumir y al esfuerzo que las áreas de TI están dispuestas a sacrificar.

En otras ocasiones, estas soluciones mágicas constituyen una auténtica caja negra hacedora de milagros que, si bien no significa que lo hagan mal, tienen la desventaja de que el conocimiento no se encuentra dentro de la organización, y en situaciones de contingencia o no disponibilidad se vuelve extremadamente crítico mantener el control. De otra forma, las posibilidades de volver atrás o retomar la operación se derrumbarán lentamente, aunque no tanto como la posición de sus responsables.

Es por eso y otras cosas, que una Arquitectura Cliente-Servidor / Cliente-Servidor enfocada en aplicaciones de misión crítica se constituye como una herramienta válida, principalmente para aquellos procesos de negocios en donde por sus especiales características, no existen herramientas en el mercado que puedan satisfacer el 100% de las necesidades.

Esta forma de diseñar soluciones actúa a modo de capa intermedia entre el usuario y los sistemas centrales: el usuario trabaja contra la capa intermedia actuando esta como servidor, y la capa intermedia actúa contra los sistemas centrales actuando como cliente para la misma, lo cual la convierte en una herramienta especial para soportar aplicaciones de misión crítica que requieran operar en forma continua.

En la práctica, superados los primeros momentos de incertidumbre típicos de cada implementación, en la práctica y cumplida la primer temporada de zafra, la solución implementada en el Ingenio Azucarero con base en esta Arquitectura, logró convertirse en una herramienta de valor para la organización, que permitió salvar más de una no disponibilidad, garantizando la continuidad de los procesos críticos en forma totalmente transparente para los usuarios, y de a poco comenzó a erigirse en una herramienta fundamental para el éxito de la temporada.

Dada la nula información al respecto, este trabajo surgió como continuidad del proyecto para el Ingenio Azucarero, haciendo extensiva la arquitectura como marco de referencia general para cualquier solución de misión crítica que se aplique, y con un mínimo de formalismo que otorgue sentido lógico a la descripción.

Finalmente, la inclusión de material relacionado con la Arquitectura Empresarial en diversas formas pretendió darle cierto nivel de alineamiento a los estándares organizaciones, de procesos y de TI,



configurando un marco tanto para su inserción dentro de la Organización como así también su interrelación y complementación con otras ramas de la arquitectura de TI.

Bibliografía y Material de Soporte

Durante la confección de este trabajo se utilizó entre otras, información de las siguientes fuentes:

- Framework de Zachman - Arquitectura Empresarial: <http://www.zachman.com>
- The Open Group Architectural Framework (TOGAF): <http://www.opengroup.org>
- The Federal Enterprise Architecture: <http://www.mitre.org>
- The Gartner Methodology: <http://www.gartner.com>
- <http://www.enterprise-architecture.info>
- ITIL V3
- ISO/IEC 42010
- IEEE P1471–2000
- Microsoft MSDN: <http://www.microsoft.com>
- Oracle
- IBM
- Wikipedia
- Otros estándares, materiales e información disponible en la web



FORMULARIO B



I NSTITUTO
U NIVERSITARIO
A ERONAUTICO

Facultad de Ciencias de la Administración

Departamento Desarrollo Profesional

Lugar y fecha: 28 de Febrero de 2014

INFORME DE ACEPTACIÓN del ANTEPROYECTO del TRABAJO FINAL

Título del Trabajo Final: Arquitectura Cliente-Servidor / Cliente-Servidor en Aplicaciones de Misión Crítica

Integrantes: (Apellido, Nombre y Carrera) Cristián Bircher - Ingeniería de Sistemas de Información

Profesor Tutor del TF: Natalia Mira

Miembros del Tribunal evaluador:

.....

Resolución del Tribunal Evaluador

- El AP puede aceptarse en su forma actual sin modificaciones.
- El AP puede aceptarse pero el/los alumno/s debería/n considerar las Observaciones sugeridas a continuación.
- Rechazar debido a las Observaciones formuladas a continuación.

Observaciones:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....