

ESCUELA POLITECNICA DEL EJÉRCITO

DPTO. DE CIENCIAS DE LA COMPUTACIÓN

CARRERA DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

**Diseño, Construcción e Implementación del Núcleo de
Procesamiento de la Versión Automatizada de Paraná, Bajo un
Modelo de Arquitectura Orientada a Servicios**

Previa a la obtención del Título de:

INGENIERO DE SISTEMAS E INFORMÁTICA

POR: ABDELRAHAMAN ALY TOVAR

SANGOLQUÍ, Febrero de 2010

CERTIFICACION

Certifico que el presente trabajo fue realizado en su totalidad por el Sr. ABDELRAHAMAN ALY TOVAR como requerimiento parcial a la obtención del título de INGENIERO DE SISTEMAS E INFORMATICA

Sangolquí, Febrero del 2010

Ing. Rolando Reyes

DEDICATORIA

Si cuando en entre tus sueños y esperanzas te encuentres buscando la luz, preguntándote si al final se puede obtener una victoria. Recuerda que este trabajo es un testimonial de aquello. Esta, va dedicada a ti, compañero del camino, y espero que tanto tu como yo logremos llegar a la luz al final de toda historia.

Abdelrahaman Aly

AGRADECIMIENTO

La vida siempre resulta en una compleja aventura por alcanzar momentos y circunstancias. Al final el esfuerzo que demanda es imposible de llevar para cualquier hombre persé.

Mis más profundos agradecimientos a todos aquellos que durante este corto camino, por más noble o insignificante que pueda en sí mismo significar, estuvieron de mi lado.

A mis compañeros en la vida profesional, a mis compañeros en mi vida estudiantil, a mis maestros, a los que desearon mi bien y a los que desearon mi mal, ambos me motivaron cada día a superarme.

A mi familia por estar siempre ahí, al pendiente de lo que yo fuese capaz de lograr.

Las circunstancias cambian las formas y los esquemas, al final del camino, jamás olvidaré los detalles en los que tantos destacaron su apoyo y la estima que me tenían. El nombrarlos a todos requeriría de otro tomo de la presente, como olvidar a Ramiro Delgado y sus clases de Electrónica, Lina Forero y su profesionalismo, Danilo Martínez y la paciencia para forjarnos como desarrolladores, Hugo Muñoz y el desafío que significo conocer una persona así en nuestras vidas, Rolando Reyes y el prototipo de profesionales que deseábamos ser, Santiago y Adrián, a ustedes les debo el ver el mundo como un gran sueño, la lucha constante por algo que trasciende a lo financiero, el demostrarnos siempre que podemos ser lo mejor, a Microsoft Corp. Por su invaluable apoyo durante toda la carrera, etc.

Las anécdotas son interminables, la experiencia enriquecedora, algo que ni en mis mejores sueños imagine vivir, algo que me estaba vetado, que jamás pensé conseguir, es en este momento en el que puedo entender al menos por un breve instante el significado del precio mi paz, que al final fue sobre él.

Mi corazón se estremece, jamás olvidaré esta, la mejor experiencia que he disfrutado, deseándoles a todos que puedan compartir hoy mi alegría. Siempre los llevaré en mi corazón.

Espero que conmigo todos, desde ahora en más busquemos el futuro.

Abdelrahman Aly

ÍNDICE

INDICE DE FIGURAS.....	xi
INDICE DE TABLAS	xii
INDICE DE ILUSTRACIONES	xiv
RESUMEN.....	xv
1. CAPÍTULO: INTRODUCCIÓN.....	1
1.1. Tema.....	1
1.2. Antecedentes	1
1.3. Justificación e Importancia.....	3
1.4. Objetivos	6
1.4.1. Objetivo General.....	6
1.4.2. Objetivos Específicos.....	6
1.5. Alcance.....	7
2. CAPÍTULO: MARCO TEÓRICO.....	10
2.1. Introducción.....	10
2.2. Situación Actual con respecto a la Aplicación de la filosofía.....	10
2.3. Conceptos Básicos.....	11
2.4. Filosofía y Paradigma	11
2.4.1. Indicadores Medioambientales.....	12
2.4.2. Núcleo operacional.....	13
2.4.3. Interfaces Acoplables.....	13
2.5. Filosofía Paraná.....	14

2.5.1.	Orígenes	14
2.5.2.	Objetivos de la Filosofía	17
2.5.3.	Indicadores	18
2.6.	Modelo SOA	18
2.6.1.	Fundamentos de SOA	18
2.6.2.	Integración Punto a Punto, EAI y BPM	20
2.6.3.	Modelo de Arquitectura Referencial	27
2.6.4.	Ventajas de la Aplicación y uso de SOA.....	32
2.7.	Metodología extreme Programming (Programación Extrema).....	33
2.7.1.	Introducción a extreme Programming	33
2.7.2.	Definición y descripción de extreme Programming.....	34
2.7.3.	Introducción al uso de extreme Programming	35
2.7.4.	Características de extreme Programming	37
2.7.5.	Fases de extreme Programming.....	43
2.8.	Herramientas de Desarrollo y Diseño	49
2.8.1.	Herramienta de Diseño: Microsoft Visio 2007, Enterprise Architect 7.5, VizSequence 0.11	49
2.8.2.	IDE: Visual Studio 2008.....	50
2.8.3.	ataforma: Microsoft .Net Framework Versión 3.5.....	52
2.8.4.	Plataforma Para el Desarrollo de Web Services : Windows Communication Foundation.....	57
2.8.5.	Servidor de Base de Datos: Microsoft SQL Server 2005.....	59

2.8.6.	Acceso a Información: Live Services y Fuentes de Terceros.....	61
2.8.7.	Servidor de Aplicaciones, Web: Internet Information Server 7.....	62
2.8.8.	Enterprise Service Bus: Aqualogic Service Bus 3.....	63
3.	CAPÍTULO: PLANIFICACIÓN (ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE PARA EL DESARROLLO DEL NÚCLEO OPERACIONAL DE PARANÁ) ...	66
3.1.	Introducción.....	66
3.1.1.	Propósito.....	66
3.1.2.	Alcance.....	67
3.1.3.	Definiciones, Acrónimos y Abreviaturas	67
3.1.4.	Visión General.....	68
3.2.	Descripción.....	68
3.2.1.	Funciones Principales.....	68
3.2.2.	Restricciones Generales	69
3.2.3.	Suposiciones y Dependencias	70
3.3.	Requerimientos Específicos	71
3.3.1.	Historias de Usuario.....	71
3.4.	Plan de Entregas	85
3.6.	Iteraciones	88
3.6.1.	Iteración 1	88
3.6.2.	Iteración 2	94
3.6.3.	Iteración 3	98
3.6.4.	Iteración 4	102

3.7.	Rotaciones.....	104
3.8.	Reuniones.....	105
4.	CAPÍTULO: DISEÑO DEL NÚCLEO OPERACIONAL DE PARANÁ.....	106
4.1.	Metáfora del Sistema	106
4.2.	DISEÑO ARQUITECTÓNICO DE LA APLICACIÓN	107
4.2.1.	Arquitectura Lógica	107
4.2.2.	Arquitectura Física.....	108
4.3.	Tarjetas CRC.....	109
4.4.	Soluciones Puntuales	117
4.5.	Funcionalidad Mínima.....	118
5.	CAPÍTULO: IMPLEMENTACIÓN DEL NÚCLEO OPERACIONAL DE PARANÁ	119
5.1.	Programación en Parejas.....	119
5.2.	Estándares de Codificación	120
5.3.	Modelo de Datos.....	122
5.4.	Modelo de Clases.....	124
5.5.	Generación de Proxy Services	130
5.6.	Diagrama de Mapeo de LINQ.....	132
5.7.	Pruebas Unitarias	133
6.	CAPÍTULO: CONCLUSIONES Y RECOMENDACIONES.....	135
6.1.	Conclusiones.....	135
6.2.	Recomendaciones.....	138
	ANEXOS.....	141

Anexo A: Diccionario de Datos.....	141
Anexo B: Script de Creación de Base de Datos.....	150
Anexo C: Manual Despliegue de Proxy Services sobre Aqualogic Service Bus 3	162
Anexo D: Manual de Creación de Interfaces Genéricas	171
ANEXO E: WSDLs de Salida de Proxy Services	187
ANEXO F: Establecimiento Archivos de Administración y Configuración.....	196
GLOSARIO.....	201
BIBLIOGRAFIA	205

m

INDICE DE FIGURAS

Figura 2.5.1.1.1. (Descripción de Modelo de Trabajo Ponit-to-Point).....	20
Figura 2.5.1.1.2 (Descripción del Modelo de Trabajo con un Bus de Mensajería EAI)	22
Figura 2.5.2.1 (Modelo de arquitectura referencial SOA)	27
Figura 2.7.3.1 (Transcurso de .Net a lo largo de la historia)	53

INDICE DE TABLAS

Tabla 3.3.1.1 (C. U. Inserción de Usuarios)	72
Tabla 3.3.1.2 (C. U. Identificación de Usuarios)	72
Tabla 3.3.1.3 (C. U. Identificar entorno de Usuario)	73
Tabla 3.3.1.4 (C. U. Inserción de Grupos Sociales)	74
Tabla 3.3.1.5 (C. U. Identificación de Grupo Social)	75
Tabla 3.3.1.6 (C. U. Nombre: Identificación de Consumos)	76
Tabla 3.3.1.7 (C. U. Nombre: Identificación de Zona geográfica).....	77
Tabla 3.3.1.8 (C. U. Nombre: Inserción de Consumos de un Usuario)	78
Tabla 3.3.1.9 (C. U. Nombre: Identificación Pormenorizada de Datos de Usuario)	79
Tabla 3.3.1.10 (C. U. Nombre: Identificación de Usuarios Para Clientes Livianos)	80
Tabla 3.3.1.11 (C. U. Nombre: Identificación de Consumos de Usuario)	81
Tabla 3.3.1.12 (C. U. Nombre: Identificación de consumos por su Tipo)	82
Tabla 3.3.1.13 (C. U. Nombre: Identificación de Puntaje por Zonas)	83
Tabla 3.3.1.14 (C. U. Nombre: Identificación de Puntaje de Zona Por Usuario) ..	84
Tabla 3.6.1.1.1 (P.A. PA001).....	88
Tabla 3.6.1.1.2 (P.A. PA002).....	89
Tabla 3.6.1.1.3 (P.A. PA003).....	91
Tabla 3.6.1.1.4 (P.A. PA004).....	91
Tabla 3.6.1.1.5 (P.A. PA005).....	92
Tabla 3.6.1.1.6 (P.A. PA006).....	93
Tabla 3.6.1.1.7 (P.A. PA007).....	95
Tabla 3.6.1.1.8 (P.A. PA008).....	96

Tabla 3.6.1.1.9 (P.A. PA009).....	97
Tabla 3.6.1.1.10 (P.A. PA010).....	99
Tabla 3.6.1.1.11 (P.A. PA011).....	99
Tabla 3.6.1.1.12 (P.A. PA012).....	100
Tabla 3.6.1.1.13 (P.A. PA013).....	102
Tabla 3.6.1.1.14 (P.A. PA014).....	103
Tabla 4.3.1 (C.R.C. CityManagerImpl)	109
Tabla 4.3.2 (C.R.C. ConsumptionManagerImpl)	110
Tabla 4.3.3 (C.R.C. ConsumptionRegistrationManagerImpl)	111
Tabla 4.3.4 (C.R.C. CountryManagerImpl)	112
Tabla 4.3.5 (C.R.C. InscriptionProcessManagerImpl)	112
Tabla 4.3.6 (C.R.C. MangroveCreationManagerImpl)	113
Tabla 4.3.7 (C.R.C. MangroveManagerImpl).....	113
Tabla 4.3.8 (C.R.C. MobileOperationManagerImpl)	114
Tabla 4.3.9 (C.R.C. ProfileManagerImpl)	115
Tabla 4.3.10 (C.R.C. SimulationManagerImpl).....	116
Tabla 5.2 (M.E. archivos)	120
Tabla 5.2 (M.E. clases)	121
Tabla 5.7.1. (Cuadro de Resultados Final de Pruebas Unitarias Ejecutadas)...	134

INDICE DE ILUSTRACIONES

Ilustración 3.4.1 (Cronograma del Plan de Entregas).....	85
Ilustración 3.5.1 (Cronograma del Plan de Iteraciones)	87
Ilustración 4.2.1.1 (Arquitectura Lógica de la Aplicación).....	107
Ilustración 4.2.2.1 (Diagrama de Arquitectura Física)	108
Ilustración 5.3.1 (Modelo Lógico)	122
Ilustración 5.3.2 (Modelo SQL Server)	123
Ilustración 5.4.1 (M.C. ParanaSoft.Parana.BusinessEntities).....	124
Ilustración 5.4.2 (M.C. ParanaSoft.Parana.DataAccessObjects).....	125
Ilustración 5.4.3 (M.C. ParanaSoft.Parana.Manager).....	126
Ilustración 5.4.3 (M.C. ParanaSoft.Parana.ManagerImplementation)	127
Ilustración 5.4.3 (M.C. ParanaSoft.Parana.Utils y ParanaSoft.Parana.Server)	128
Ilustración 5.4.3 (M.C. ParanaSoft.Parana.WCFServices).....	129
Ilustración 5.5.1 (M.C. Service Proxy)	131
Ilustración 5.6.1 (M.C. mapping de LINQ)	132

RESUMEN

Durante el último siglo, la población mundial ha sido testigo de un conjunto de cambios dramáticos en su típico estilo de vida, los que le han permitido mejorar su experiencia de vida, y al final el cumplir de mejor manera, el objetivo que se ha planteado como especie, la supervivencia. Nuestra inteligencia superior se vuelve una ventaja evolutiva clara, esta nos ha permitido aprovechar de forma exponencial cada recurso ofrecido por el planeta, algo que ninguna especie anterior habría conseguido, pero es tal vez esta sobre explotación sellada en nosotros como nuestro típico modelo mental la futura causante de nuestra desaparición como ha pasado con el 99.9% de las especies que han caminado sobre la tierra. Paraná es una filosofía que busca alterar dicha realidad, y el presente proyecto es una iniciativa que persigue en su etapa inicial el automatizar tareas fundamentales de la filosofía, para que de esta manera, su implementación a gran escala sea real con la construcción por parte de usuarios finales alrededor del mundo de interfaces genéricas que interactúen con los componentes que se encargan de la administración de la lógica de negocio. Los usuarios podrían ser grupos ambientalistas como entidades gubernamentales, lo que permitirá ampliar la base de usuarios, piedra angular para garantizar el éxito de la ya mencionada filosofía que persigue el objetivo ya dicho, empleando como herramienta el cambio de paradigmas mentales.

1. CAPÍTULO: INTRODUCCIÓN

1.1. Tema

Diseño, construcción e implementación del núcleo de procesamiento de la versión automatizada de Paraná, bajo un Modelo de Arquitectura Orientada a Servicios.

1.2. Antecedentes

A partir de la era industrial, el género humano ha sufrido grandes trastornos en su comportamiento típico, fruto de invenciones que modificaron el estilo de vida del ser humano promedio, se puede nombrar al automóvil cómo el mayor ejemplo, pero sin dejar a un lado hechos como la masificación de la ganadería como fuente alimentaría y el modelo fabril que adoptaron cientos de centros de producción que a partir de ese momento se denominaron fábricas.

Estos hechos han enmarcado cientos de cambios radicales en el comportamiento de varios modelos climáticos¹ en especial, cambios que en la actualidad resultan visibles, la proximidad de eventos climatológicos relacionados con la variación en la temperatura de los océanos, El Niño y La Niña en las costas del Ecuador, movimientos migratorios en especial de aves y peces en búsqueda de temperaturas aptas para sus procesos reproductivos, periodos de actividad atípicamente alta de formación de huracanes y tifones, entre otras manifestaciones.

¹ Un modelo climático es un conjunto de ecuaciones que persiguen el describir el comportamiento del mundo real y anticipar así el comportamiento del clima en los próximos años

Esto en esencia por el nuevo comportamiento de variables que antes no influenciaban de manera notable estos modelos, para ser acordes con la presente descripción se puede considerar las emisiones de CO₂ que se enmarcan en niveles claramente superiores a los de siglos anteriores, pero todavía más preocupante es el aumento en la atmosfera de niveles gas metano, mayores a los registrados en milenios, este es producido básicamente por el ganado vacuno ya que es de origen biológico en mayor dosis, en la tierra, y que resulta el principal causante del efecto invernadero.

Todos estos fenómenos se han visto intensificados en especial en el último cuarto de siglo, lo que ha generado un comportamiento climático que puede ser definido entonces como anómalo. Es por esta razón que en la década pasada se ha visto el apareamiento de cientos de indicadores que señalan de manera diversa, los efectos negativos que tendrían sobre el normal desempeño de modelos tan complejos como los climáticos, el mantener comportamientos generalizados como los actuales, es más, estos índices resultan totalmente insostenibles a mediano y largo plazo, proyecciones sobre mantenibilidad alimentaria, constante incremento de la temperatura del mar, incremento de los niveles del mismo, la constante expansión de zonas desérticas como el Sáhara y el Gobi, entre otros.

Estos nuevos valores de las variables que consideran los modelos climáticos se originan en los niveles actuales de consumo de recursos naturales actuales y la utilización no administrada de las nuevas alternativas tecnológicas como factores incidentes, en consecuencia se puede determinar que inclusive parando el constante incremento de su uso y aprovechamiento, al día de hoy, el mundo no lograría satisfacer de manera efectiva las necesidades básicas e

inclusive la supervivencia de todos sus habitantes en cortos periodos de tiempo mucho menos a mediano y largo plazo. Entre los indicadores que guardan el secreto del comportamiento de estos modelos en años posteriores se encuentra el footPrint², que efectivamente es el encargado de medir la huella que se deja en el planeta. Bajo esta óptica, muchos individuos y grupos de voluntariado han adaptado filosofías que en teoría cambiarían el comportamiento humano típico para redefinir la afectación a los ya tan mencionados modelos, entre estas se encuentra Paraná, cuyo objetivo es la reducción de consumos de los grupos humanos involucrados.

1.3. Justificación e Importancia

Dadas las proyecciones de los modelos climáticos actuales, se puede identificar claramente que los problemas medioambientales se originan de manera sustancial por la manera deshabitada, libre de todo tipo de limitación, del consumo de recursos tanto renovables como no renovables, del miembro típico de la comunidad. El uso de filosofías que buscan el implantar paradigmas mentales, se han convertido en el arma ideal para lidiar con esta problemática, es aquí en donde Paraná entra a la escena.

Paraná tiene su origen en la conjugación de una serie de principios guaraníes, que han sido orquestados por grupos estudiantiles de tono ecologista, persiguiendo como objetivo el implantar un mejor relacionamiento del hombre con el medio circundante reduciendo sus consumos. Los pilares en los que usa son apenas cuatro:

² Indicador que señala la demanda del género humano sobre el ecosistema. Basado en la comparación de la demanda de un ser humano con la real capacidad ecológica del planeta para regenerar los recursos consumidos. Este representa el total de área marítima y de terreno cultivable junto con su capacidad de absorción del daño producido por el desgaste de los mismos

Arandú: Etimológicamente viene de Sabiduría, y versa básicamente en el aglutinamiento del conocimiento, es la agrupación de la información.

Ikatú: No tiene una traducción exacta al español, su significado más aproximado se puede identificar como el Conector Universal Entre Todas las Personas, y eso es lo que persigue, la interrelación, para la consecución y formación de una sabiduría colectiva

M'boe: Enseñar, es cómo el mostrar a los demás no solo lo aprendido sino también lo conseguido, ambos se integran con la ya mencionada sabiduría colectiva

Kuaa: Conocer, visto desde una óptica pluralista fundamentada en el respeto a los demás, es decir que no es sólo el identificar información de terceros, sino el brindar mi información a terceros para que pueda recibir ayuda

Es por la interrelación de estos elementos que la filosofía recibe su nombre, Paraná quiere decir el Pariente del Mar, el mar es visto cómo un símil, que identifica no solo poder sino a toda la raza humana, en síntesis Paraná es la unión de los conocimientos y las enseñanzas de todos los individuos involucrados para producir sabiduría colectiva, que al involucrar a los sujetos en su desarrollo de una manera tan comprometida, siendo parte de sus propias vivencias, establece la inmediata adopción dentro de sus hábitos de vida cotidianos, generando un impacto real sobre sí y el medio circundante consiguiendo así vencer la influencia del medio social, en nuestro presente caso enfocado sin dudas a la reducción de consumos.

La implementación del núcleo de procesamiento que embeberá la lógica propia de la filosofía, sin dudas se convierte en el primer paso dado para la masificación del uso de la ya mencionada filosofía Paraná, permitiendo que a futuro cualquiera pueda desarrollar interfaces capaces de interactuar con el presente proyecto, ocasionando una reacción en cadena que permita la creación de nuevos instrumentos que busque la reducción del amplio porcentaje de consumos de productos suntuarios y nocivos para el medio ambiente.

El presente proyecto y de manera adicional, cobra relevancia tecnológica al implementar una serie de algoritmos matemáticos diseñados específicamente para este motor con el fin estricto de evaluar el progreso individual de cada usuario y no así cantidades subjetivas que en realidad no señalan el contexto del progreso real de un individuo. Adicional a la presente implementación del algoritmo mencionado se plantea la utilización de herramientas de última generación, de cuyo uso este proyecto será pionero e integrador a nivel nacional, tecnologías tales como las Foundation de Microsoft ³que resultan sumamente viables para el actual proyecto adicional de la inédita integración con el Bus de Aqualogic⁴, algo nunca antes presentado en un proyecto de investigación de la ESPE.

En cuanto a su trascendencia política, se puede señalar la manera en la que encaja entre los objetivos nacionales, dado que resulta ser política del gobierno actual, definida por la Secretaría Nacional de Planificación Senplades, la disminución del consumo eléctrico Nacional, además de otros aspectos como la

³ Microsoft Foundation referido al conjunto de elementos y conocimientos base para la implantación tecnológica de un concepto todo bajo la utilización de herramientas Microsoft, ejemplos de esto son WCF, WWF, WPF, entre otros.

⁴ Se refiere al BEA Aqualogic Service Bus, que es un ESB que permite la interacción entre servicios, ruteo de relaciones, transformación de data o mensajes y aplicación de políticas

concordancia con lo expuesto en la Constitución vigente con respecto a derechos básicos de acceso a recursos naturales como el agua por ejemplo., de todo ciudadano ecuatoriano. Por estos motivos entre otros varios es que se señala la total concordancia del presente proyecto con los objetivos nacionales y su contribución real y tangible al cumplimiento de los mismos.

En cuanto a la viabilidad, se debe de tomar en cuenta que se trabajará de manera coordinada con los creadores de la filosofía aquí mencionada, además de la interacción con Secretarías Estatales a fines de manera directa, resultando factible el contacto y la relación entre el proveedor de información y nosotros.

1.4. Objetivos

1.4.1. Objetivo General

Diseñar, construir e Implementar el núcleo (core) de procesamiento de la versión automatizada de Paraná, bajo un Modelo de Arquitectura Orientada a Servicios, que permita el control, medición y disminución de los consumos de recursos naturales de las personas promedio.

1.4.2. Objetivos Específicos

- Definir de manera consistente en qué versa la filosofía “Paraná”, estableciendo si está repercute en la disminución de los consumos de las personas promedio, implantando un modelo de aplicación.
- Diseñar el núcleo de acuerdo a los principios que rigen a SOA, obteniendo componentes de bajo nivel de

acoplamiento y alto de cohesión para así propender a la integración en el uso de información de sistemas de terceros.

- Desarrollar los cuatro componentes elementales del núcleo bajo el marco referencial de la filosofía alcanzando al final del proceso un conjunto de servicios clasificados que permitan recopilar los principales elementos y funciones de la misma.
- Implantar el núcleo y exponer “Proxy Services⁵”, los mismos que deben permitir el libre acoplamiento con interfaces desarrolladas o que puedan desarrollarse en el futuro.

1.5. Alcance

El alcance definido en para el proyecto actual resulta en la implementación de todo el núcleo central de procesamiento de los términos enmarcados dentro de Paraná , factibles de automatización, dentro de la ya mencionada filosofía, y la publicación bajo un modelo Arquitectónico Orientado a Servicios (SOA), de componentes consumibles por las interfaces que a futuro puedan implementarse utilizando para esto tanto WCF⁶ como el propio Bus de Integración de Aqualogic⁷ que será el encargado de Distribuir toda interacción entre los componentes lógicos y servicios tanto del núcleo de Paraná cómo de terceros, con la información entrantes de las interfaces, proceso conocido como Orquestación.

Los Service Proxies que podrán ser consumidos por las interfaces y que embeberán toda la funcionalidad del núcleo se pueden clasificar en 4 Grupos

⁵ Un Proxy Service es el producto final de un ESB, es el servicio entregado por Bus tal que en su interior contiene desde orquestación hasta transformación de data.

⁶ Windows Communication Foundation, nueva plataforma de Microsoft para el manejo y construcción de integración SOA.

⁷ Aqualogic es la anterior línea de Bea Systems encargada de productos de línea SOA Middleware y BPM.

base, estos comprenden los cuatro pilares de la filosofía y cumplen responsabilidades y tareas a fines, estos son los siguientes:

Arandú

Se encarga de la administración de todo el “conocimiento” e información registrada sobre el núcleo de procesamiento, contendrá componentes básicos para mantener relación con interfaces de tipo Web 2.0 y redes sociales además del registro de ciertos ítems que podrán ser usados para calificar el comportamiento de los usuarios.

Ikatú

Estos serán los servicios encargados de definir datos sumamente puntuales bajo interacción directa con el usuario, además de entregar información sumamente puntual sobre este en específico, cómo valores estadísticos que señalan su desempeño dentro de los principios de la filosofía, que podrán servir para el desarrollo de pequeñas interfaces de usuario que señalen el rendimiento del participante, como Gadgets o aplicaciones móviles sin importar su plataforma, cómo podrían ser Android, Windows Mobile y tecnologías propias de Apple.

Mboe

Estos servicios entregarán la data de usuarios seleccionados contrastada con el de sus similares bajo parámetros estrictamente definidos, como podrán ser sus vecinos e inclusive amigos, esto en base a su localización y los promedios que entregue el núcleo en base a los algoritmos de calificación ya mencionados.

Kuaa

Finalmente este grupo de Servicios se refieren de manera estricta a aquellos que me permitirán recolectar y entregar data real sobre los consumos de usuarios que lo autoricen, en base a estos es que efectivamente se obtendrán valores definitorios que me permitirán establecer el comportamiento de un usuario.

2. CAPÍTULO: MARCO TEÓRICO

2.1. Introducción

Como ya se ha mencionado con anterioridad y de manera reiterada, la crisis provocada por los altos niveles de consumo ha despertado el interés no solo de instituciones de renombre

2.2. Situación Actual con respecto a la Aplicación de la filosofía

Como ya se ha mencionado con anterioridad y de manera reiterada, la crisis provocada por los altos niveles de consumo ha despertado el interés no solo de instituciones de renombre y alcance mundial, sino también de grupos sociales que promueven el consumo responsable.

La filosofía fue diseñada como se explicará más adelante por un grupo de estudiantes universitarios, sin que su aplicación haya sido registrada con anterioridad en sistemas automatizados, por lo tanto incursiones en el mundo informático de la misma no se registran al momento.

Los creadores de la filosofía al día de hoy practican y respetan sus principios pero carecen de formas de difusión y utilización a gran escala, no puede ser empleada como medio dada la no existencia de un vehículo que se constituya en el medio fehaciente para el empleo y adopción masiva no solo de la filosofía sino de su continua práctica.

El poder digitalizar dichos mecanismos propios de la filosofía, en un motor ágil que registre niveles de cohesión sumamente altos permitirá la utilización a gran escala de estos, la construcción de interfaces acordes a las necesidades de

las comunidades que adopten dichos principios serán ventanas que mantendrán concordancia con las políticas básicas trazadas por la presente filosofía.

2.3. Conceptos Básicos

Con la finalidad de brindar al lector del presente documento, la oportunidad de comprender y analizar las características de fondo y forma del presente proyecto, así como sus reales beneficios, se presenta a continuación una compilación de términos que pueden resultar de una u otra manera confusos, los mismos que se emplearán en capítulos posteriores.

2.4. Filosofía y Paradigma

Etimológicamente viene del latín “*philosophia*” que tiene como significado literal el “amor a la sabiduría”. Su aplicabilidad a lo largo de los años posteriores a su establecimiento por Platón y Pitágoras ha cambiado, optando por tendencias cada vez más materializadas, es así que se puede ver como la forma en la que un problema es abordado en base a conocimientos obtenidos adquiridos luego de un profundo análisis de este. En síntesis es una forma. Una manera un camino de libre elección, de ver y enfrentar un obstáculo presentado o una cuestión en específico, entender un principio y valorar un aspecto trascendental en la concepción que mantenemos sobre un punto. Como ejemplo podemos citar a Paraná como una filosofía, ya que es la respuesta al análisis de un grave problema medioambiental ya analizado anteriormente.

Paradigma en cambio etimológicamente procede del griego “*paradeigma*”, cuya significancia es “modelo”, y se refiere a un patrón predefinido bajo cualquier forma epistemológica, en palabras sencillas es nada más que una manera de

realizar una tarea de forma típica, un modelo esquematizante, se usa para referirse a elementos u objetos que presenten similitudes. En el presente trabajo investigativo su concepción debe obligatoriamente pasar por campos psicológicos haciendo referencia pura al banco de acepciones ideológicas, pensamientos generalizados y creencias que se incorporan a nuestra psique a tempranas edades y q se aceptan como postulados falsos o verdaderos sin necesidad de ningún tipo de análisis ni validación.

2.4.1. Indicadores Medioambientales

Un indicador no es más que un instrumento, que permite el relacionar un fenómeno medible y evaluable que además permite el señalar la existencia de ciertos comportamientos o características específicas, en el presenta caso de estudio, estas se consideran solo en el campo medioambiental.

La Unión Europea, en Gotemburgo en junio del 2001, en uno de los principales congresos de análisis climático del mundo moderno adoptó un plan estratégico para el desarrollo sostenible en el que incluyó una amplia lista de indicadores medioambientales necesarios para el crecimiento sostenible, un objetivo crucial del presente proyecto, entre ellos y como motivo de ejemplo podemos señalar a:

- Las emisiones de gases de efecto invernadero, emisiones por habitante, por sector y por unidad de PIB;
- El consumo energético según el modo de transporte;
- Exposición de poblaciones urbanas a la contaminación atmosférica;

- Niveles de emisiones de contaminantes atmosféricos como el dióxido de azufre;
- Sostenibilidad de la pesca artesanal y comercial de ciertas especies;

Existen otras organizaciones sociales sin fines de lucro que utilizan otro tipo de indicadores, ya que para estos el apoyo y la fuerza gubernamental son esenciales, en el caso del presente trabajo en específico, la utilización del “*footprint*”, es la opción viabilizada, ya que este indicador es usado ampliamente por estas organizaciones.

2.4.2. Núcleo operacional

Consiste en un conjunto de paquetes de software destinados a cumplir y administrar la información de los agentes fundamentales de negocio, en el se realiza de manera integra la transformación edición y manejo de la data, para luego exponerla de manera estandarizada para que pueda ser consumida por interfaces adecuadas para el fin.

2.4.3. Interfaces Acoplables

Las interfaces acoplables son todos aquellos agentes diseñados y construidos con la finalidad de consumir los recursos de un repositorio de información pre procesado de tal suerte que su intervención sobre la lógica del negocio sea mínima, delegando dichas funcionalidades al núcleo operacional, los vínculos que establecen los puentes de comunicación deben de ser estandarizados.

2.5. Filosofía Paraná

2.5.1. Orígenes

A finales del año 2007 un grupo de estudiantes universitarios de varias nacionalidades, principalmente estudiantes de la Escuela Politécnica del Ejército, con la colaboración de técnicos especializados, bajo el auspicio de esta Institución Educativa y de empresas como Microsoft Corp⁸. Motivados por los trastornos climáticos ocurridos a partir del año 2005, emprendieron en un análisis profundo sobre las principales causas de estos hechos, y como poder enfrentarlos con herramientas tan preciadas como el pensamiento, la imaginación y la esperanza.

Su análisis arrojó como resultado que medidas podían ser tomadas para alterar dicha realidad alterando de manera positiva el medio ambiente disminuyendo el nivel de consumo de recursos no renovables así como de gran impacto social, los mejores ejemplos son los combustibles de origen fósil y el agua.

En este punto la primera disyuntiva aparece a la vista, siendo que si bien existe una forma real de parar la continua alteración de los modelos climáticos, el análisis demostró que los seres humanos como individuos no se encuentran preparados para alterar su modus vivendi típico a pesar de que este puede causar su desaparición en los próximos años.

La implementación de la solución debía por tanto saltar esta vicisitud surgida de manera tal que un marco de personas considerable reduzca

⁸ Microsoft corp. Es la primera productora de Software a nivel mundial en varios campos de la industria del software, dominador de la misma durante el último cuarto de siglo

drásticamente sus consumos. Por tanto la alteración del modelo mental típico de los individuos era la alternativa más adecuada.

El siguiente escollo detectado se encontraba en las características propias de las sociedades de consumo, que son proclives a dar mayor importancia al individuo antes que al grupo social, lo que genera un proceso de segmentación cada vez más granular, lo que dificultaba nuevamente la vigencia de una alternativa de solución real.

Cuando el conjunto de dificultades se definió y la meta se fijó, el análisis dio paso al diseño de alternativas que permitan sortear los problemas y disminuir de manera significativa y representativa los consumos de las personas. El grupo de trabajo ahora integrado con grupos juveniles como el Club Ecológico de la Espe definió una serie de principios básicos que se debían aplicar para obtener el ya citado resultado y los cotejó con principios de la cultura Guaraní, el segundo pueblo nativo más grande de Sudamérica, estos son:

Arandú: Sabiduría

Ikatú: Vínculo entre todas las personas

Mboe: Enseñar

Kuaa: Conocer

Estos pilares fundamentales fueron definidos como indispensables para vencer las dificultades ya descritas. A continuación el grupo definió que todos estos deberían entrelazarse, para obtener resultados, a la unión de los principios se la denominó **Paraná**, cuyo significado en guaraní implica Pariente del Mar.

A la integración de los principios y a la manera en la que estas interactúan, una vez con un cuerpo bañado de lógica se la denominó filosofía. Esta en resumen reza:

Cada individuo per se tiene conocimiento introspectivo, que viene a ser el conocimiento elemental, básico, el conocimiento de sí mismo (Kuaa), estos conocimientos deben ser enseñados (Mboe) para ser enriquecidos y al interrelacionarlos con el de todos los demás individuos (Ikatú), se puede generar Sabiduría (Arandú). La implicación de Paraná cobra vigencia al usar Mar como una figura literaria que representa a toda la humanidad, por lo tanto la denominación de la presente sinergia se debe al producto de la misma, movimientos de impacto global.

Si bien Paraná como se dijo posee un significado etimológico real, que es el nombrado, para el pueblo Guaraní, al ser este su principal afluente representaba su vida misma, lo que para la presente filosofía es completamente plausible, la fuerza que tiene la sinergia puede cambiar el rumbo de la vida de todos los involucrados, para todo el Mar de personas que habitan este globo azul.

Como se puede ver la filosofía resulta ser una herramienta para forjar paradigmas mentales, se podría decir que editarlos, y por tanto puede ser aplicado en cualquier ámbito del pensamiento y comportamiento humano, en este caso el grupo responsable de su diseño lo imaginó cambiando los esquemas mentales de la sociedad de consumo para que esta busque el reducir sus consumos.

2.5.2. Objetivos de la Filosofía

La filosofía guarda un conjunto de objetivos básicos y hasta ciertos puntos elementales, estos deben de ser considerados y tenidos en cuenta ya que se relacionan con los tres problemas básicos que han sido explicados anteriormente, estos son:

- Provocar unidad de criterio y pensamiento entre todos sus participantes.
- Forjar sociedades enfocadas a objetivos comunes.
- Excitar movimientos sociales que produzcan cambios radicales en el comportamiento humano en masa (El grupo que identificó la filosofía lo denominó generación de sismas).
- Forjar nuevamente los paradigmas sociales que habitan en la mente del ciudadano promedio para darles un enfoque renovado y diferente acerca de su relación con el entorno social típico.

Componentes

Sus componentes ya han sido expuestos con anterioridad en el presente documento y en repetidas ocasiones, a continuación son enumerados:

Arandú: Conjunto de Conocimientos colectivos acumulados durante un periodo de tiempo dado.

Ikatú: Ente vinculante entre cada individuo miembro de la comunidad.

Mboe: La enseñanza impartida por miembros de la colectividad tanto de su conocimiento introspectivo como del conjunto de conocimientos acumulados por la comunidad.

Kuaa: Clara referencia al conocimiento, comenzando por la unidad más elemental del mismo, este es el conocimiento introspectivo, visto con sinceridad y franqueza.

2.5.3. Indicadores

El indicador sustancial bajo el cual se fijó el modelo filosófico es el “footprint”, el mismo expresa niveles de consumo actuales versus futuros, en el ámbito medioambiental, según la óptica bajo la cual se aplique la filosofía, el indicador sustancial deberá de ser redefinido en función del área de impacto. En base a este indicador se determina la efectividad del cambio de paradigma, ya que componentes de medición social sobre el cambio paradigmático de la mente humana resulta indetectables.

Existen mediciones y consideraciones que se pueden realizar en base al estado actual de niveles de consumo por ejemplo, para determinar el real impacto en áreas específicas, pero el señalar y definir los indicadores que se pueden obtener de la aplicación de la filosofía adicional al señalado y aquellos que determinan el impacto deben de ser definidos a través de un proyecto de similar envergadura que el presente.

2.6. Modelo SOA

2.6.1. Fundamentos de SOA

Durante décadas, cada área de soporte de IT ha encontrado dos únicas alternativas, como soluciones viables a cada problemática encontrada al momento de automatizar sus procesos de negocio, el uso de componentes prefabricados, como versiones de ERPs o CRMs comerciales, alimentando la industria del

Software de Aplicación empaquetado, y potenciando empresas como SAP y Oracle, dos de los principales proveedores de este tipo de producto, y como segunda alternativa, en casos exclusivos en los que la lógica del proceso de negocio no podía o no resultaba adaptable a una aplicación preconcebida, las aplicaciones “custom-built” o a la medida eran la salida predilecta por parte de la gerencia de IT.

La integración del proceso de negocio, dado el mix de herramientas y recursos utilizado típicamente se realiza a nivel de la data existente en estas, resultando en un proceso de complejidad alta, ya que el desarrollo de software tradicional en lo que contempla a integración ha sido más bien rígido y no basado en estándares, siendo este un traspie severo para IT al momento de dar soporte a los procesos de negocio más dinámicos.

Cuando se habla de instituciones de gran volumen, se puede encontrar un ciclo continuo repetido y cimentado, en el que aplicaciones se desarrollan para compartir y consolidar información de negocio, proveniente de una o más fuentes e inclusive otras aplicaciones; de manera sencilla se podría decir que la solución típica para la integración de la información es el desarrollo. Es por esto que los procesos de consolidación de data dentro de una organización conllevan de manera típica el desarrollo de componentes, su posterior ensamblado, conexiones entre sistemas back-end, implementaciones de Workflows, finalmente el desarrollo de nuevas interfaces de usuario, pruebas y finalmente debugging.

Para solventar lo que hasta este punto ha sido narrado, se diseñaron varias metodologías de integración, las tres principales son las siguientes:

- Point-to-Point,

- EMB (Enterprise Message Bus) mencionado por algunas empresas como Middleware, a esta también se lo como EAI (Enterprise Application Integration), e
- Integración de Procesos de Negocio.

2.6.2. Integración Punto a Punto, EAI y BPM

Se procederá al breve análisis de las ya nombradas metodologías de integración comenzando por la que dio orígenes a los modelos posteriores:

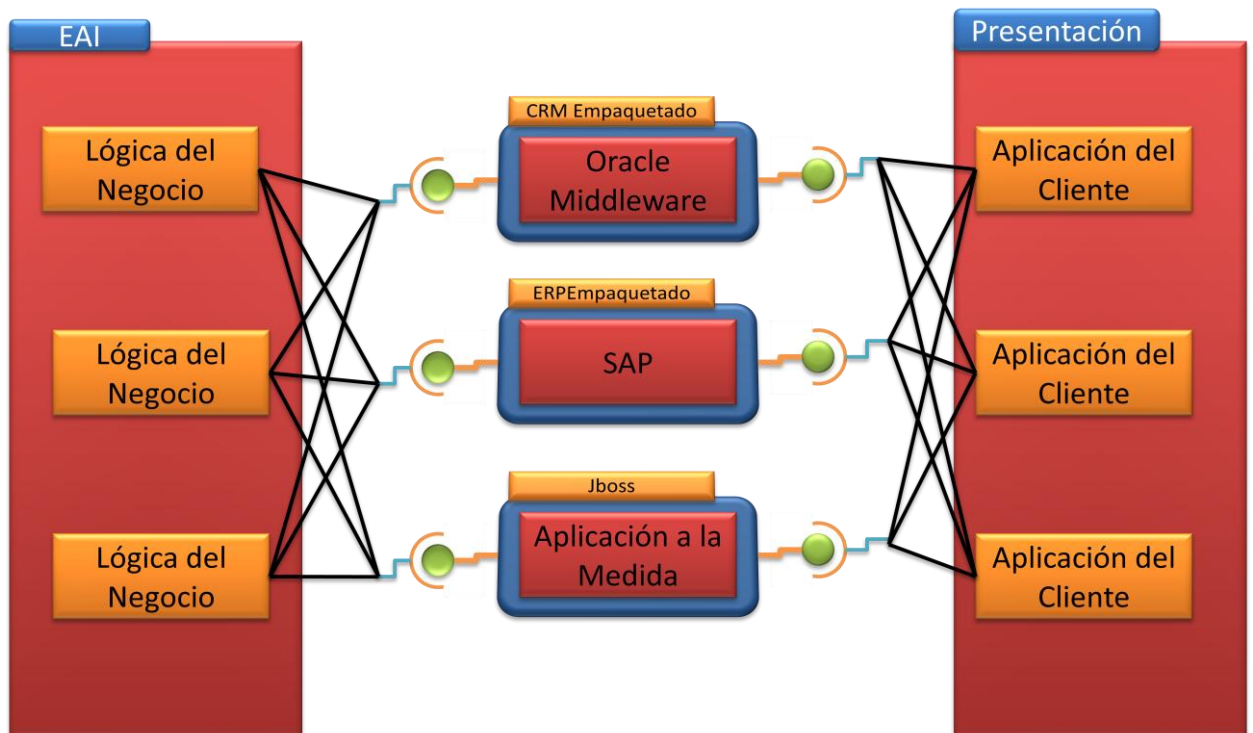


Figura 2.5.1.1.1. (Descripción de Modelo de Trabajo Point-to-Point)

La integración Point-to-Point es aquella que, como señala la **figura 2.5.1.1.1.** busca el conseguir la integración final de aplicativos utilizando como medio APIs, por lo tanto no se usan estándares abiertos, generalmente terminan convirtiéndose en acoplamientos sumamente rígido necesitando personal altamente calificado aumentando los costos, como estas implementaciones

dependen de la adaptación, inclusive de la creación de dichas APIs en ciertos casos, esto redundaba en un tiempo de respuesta poco efectivo.

Point-to-Point fue ampliamente adoptado por instituciones de toda índole, inclusive se usa hasta la actualidad al ser una metodología de fácil adopción aunque, pero por su complejidad, altos costos y la grave dificultad que resulta en redes de sistemas complejos el detectar y corregir anomalías, se implementaron modelos que buscaban mejorar los pobres resultados de Point-to-Point.

Se plantearon alternativas de solución que pudieran ser consideradas viables para resolver los problemas típicos de Point-to-Point, a la conjunción de las alternativas planteadas, se las denominó EAI, estas se basaban en el establecimiento de un bus de mensajería intermedia, también denominado “bróker” o middleware, en donde cada aplicación que necesitase compartir información empleaba APIs propietarias.

Fue entonces cuando se comprendió que esta aproximación resolvía varios problemas de comunicación entre aplicaciones, así que tomándolo como base, se generaron una serie de mejoras funcionales, generando un modelo basado en flujos de negocio, en donde el bus antes nombrado se acompaña y entremezcla con un motor de flujo de negocio, como lo indica la **figura 2.5.1.1.2**, que será empleado como orquestador de los flujos de los procesos que pueden ejecutar operaciones multi-transaccionales interactuando con varios sistemas.

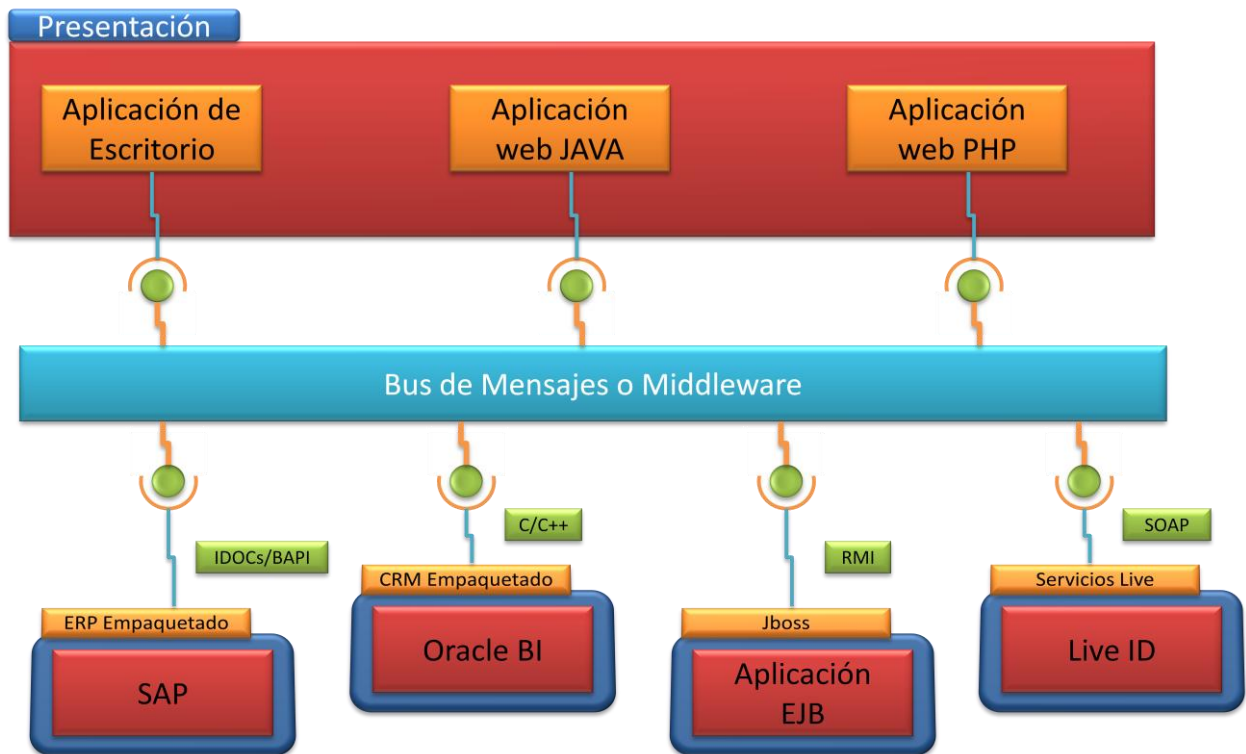


Figura 2.5.1.1.2 (Descripción del Modelo de Trabajo con un Bus de Mensajería EAI)

Las tres respuestas a las necesidades planteadas, como se puede ver requieren en la actualidad de la utilización en algunos casos de APIs propietarias, o en su defecto de la utilización de tipos de dato que no resultan genéricos en cada punto de enlace distinto. Adicionalmente en los dos últimos revisados se encuentra que las soluciones que se desarrollan se encuentran fuertemente atadas al bus. Adicional la integración entre estos sistemas resulta granular al ser necesario el comprender el funcionamiento interno de todas las aplicaciones integradas, adicional, las suites de BPM generalmente usadas, no permiten el acoplar componentes de terceros limitando su usabilidad y rendimiento.

2.6.2.1. Definiciones de SOA

SOA se plantea como un modelo que busca el superar las falencias de los mecanismos de integración típicamente empleados, su definición reza:

*La Arquitectura Orientada a Servicios es una estrategia de IT que busca el organizar las funciones discretas contenidas en aplicaciones empresariales en servicios interoperables, basados en estándares que puedan ser combinadas y usadas de manera rápida para satisfacer las necesidades del negocio.*⁹

En su esencia por tanto se puede entrever la presencia de un amplio set de patrones y buenas prácticas que resultan en recomendaciones para conectarse a servicios. Estos patrones señalan un conjunto de pasos que permiten el identificar a un servicio, para publicar y descubrir un servicio y finalmente para conectarse a un servicio.

Generalmente componentes de Middleware típicamente usados para la comunicación de aplicativos, como son RPC, CORBA componentes DCOM, los mismos EJBs, pasando por RMI utilizan patrones muy similares, inclusive los mismo en algunos casos, pero sus implementaciones presentan lo que se podría describir como “debilidades”. SOA se diseñó cuando estas tecnologías se encontraban en plena vigencia, es por estos motivos, que SOA trata de corregir dichas “debilidades”.

Como puede evidenciarse, cada una de las plataformas tecnológicas mencionadas al ser vistas de manera granular, se manejan de forma totalmente aislada y diferente, las “funciones” de los componentes RPC, o los “objetos” de los

⁹Bea Systems, Inc, “BEA AquaLogic Service Bus 3: Designing and Integrating Services for SOA Student Guide”, Oracle University, pp 16, ed. 1.0

componentes CORBA, la granularidad adecuada es inexistente en los "Servicios" de SOA, ya que en realidad un Servicio se adecua a las necesidades de sus usuarios pudiendo ser, una función, un objeto, un conjunto de objetos, etc. Ya que el modelo resulta adaptable a cualquier plataforma ya existente no obliga a ningún sistema al uso de un nivel o tipo de granularidad específico, es por esto que la recomendación de la utilización de estándares es tan repetitiva en el modelo.

Como puede entenderse hasta el presente punto SOA busca no el modificar la arquitectura actual de aplicaciones funcionales, sino permitir su reutilización convirtiéndolas en servicios ágiles. SOA no solo se encarga de administrar dentro de las líneas de gobierno a las aplicaciones preexistentes y de sistemas legacy, sino que también adapta y reutiliza la funcionalidad y data propia de la infraestructura de IT como son la seguridad u búsquedas indexadas, permitiendo así la reducción del tiempo de desarrollo de nuevas aplicaciones permitiéndoles el adicionar dichas funcionalidades.

2.6.2.2. Enterprise Service Bus

Como se ha revisado hasta el momento, se puede observar que la necesidad básica del modelo es el intercambio de mensajes, a mayor número de aplicaciones incluidas en el modelo, un mayor número de mensajes se necesitarán, los mismos deben de ser entregados de manera rápida y eficiente, el modelo como ha sido planteado en la presente claramente busca ser una alternativa "end-to-end" y por lo tanto, la velocidad de transmisión de estos es esencial.

El motor de mensajería usado de manera genérica por el modelo es el ESB (Enterprise Service Bus), este se encuentra constituido por una serie de paquetes de software. Es un componente del modelo, encargado de la coordinación de la mensajería proveyendo Niveles de Servicio o QoS (Quality of Service), constituyéndose en la ruta o camino por el que tanto como la información como las instrucciones circulan dentro del modelo.

En el momento en el que cualquier componente del modelo desea interactuar con el Bus, este debe conectarse al mismo, entregarle el mensaje en el formato correcto, esto debe de definirse de manera previa, entregándole tanto la data como la metadata correspondiente para poder identificar el destinatario, el ESB por tanto será el encargado realizar todas las funciones que correspondan para ubicarlo enviar el mensaje.

2.6.2.3. Modelo de domino para SOA

El modelo de Domino se refiere en términos genéricos al ámbito de acción de SOA, explicándolo a detalle se puede decir, que durante los primeros años de vigente aplicación de IT en las empresas en general, se pudo notar una severa separación entre las líneas y procesos de negocio y las Tecnologías de Información hablando entre ambos solo en casos específicos y bajo necesidades puntuales. Para mejorar la eficiencia evidentemente se necesitan reducir los ciclos de comunicación surgiendo la necesidad de alinearlos de una mejor forma, dadas las necesidades actuales de reingeniería de procesos para obtener menores ventanas de tiempo en su desempeño.

Si bien esta aproximación entre IT y los procesos de negocio es la que de manera original despliega la necesidad del surgimiento del modelo, el dominio, o

ámbito de este incluye en total 6 áreas clave contando la ya relatada, todas explicadas brevemente a continuación:

Aproximación entre IT y los Procesos de Negocio: SOA Provee un ambiente consistente para enlazar y evaluar mediante procesos de medición las TI y los procesos de negocio.

Arquitectura: Los ambientes de IT que se basan en estándares, distribución de carga, bajos niveles de acoplamiento y el mapeo de procesos de negocio diseñados para responder e integrarse con cualquier cambio funcional a cualquier nivel empresarial.

Construcción Modular: Una vez fijados los principios arquitecturales se llega a maximizar el potencial para alcanzar el éxito a través de la reutilización de componentes.

Aplicaciones y Proyectos: Categorizar y refactorizar funcionalidades propias de los sistemas y aplicaciones varias dentro de la institución para así estandarizar la forma en la que la funcionalidad es ofertada para así evitar la redundancia y promoviendo la consistencia en la ejecución de procesos de negocio.

Gobernanza: Interviene el modelo en la estructura organizacional y ejerce autoridad estandarizando la entrega de requerimientos de TI para así asegurar la cobertura de las necesidades de la institución maximizando el uso de funcionalidades existentes.

Costos y Beneficios: La planeación y ejecución de implementaciones de TI para generar valor de manera sustentable que se encuentre sólidamente apalancada en inversión de TI preexistentes.

2.6.3. Modelo de Arquitectura Referencial

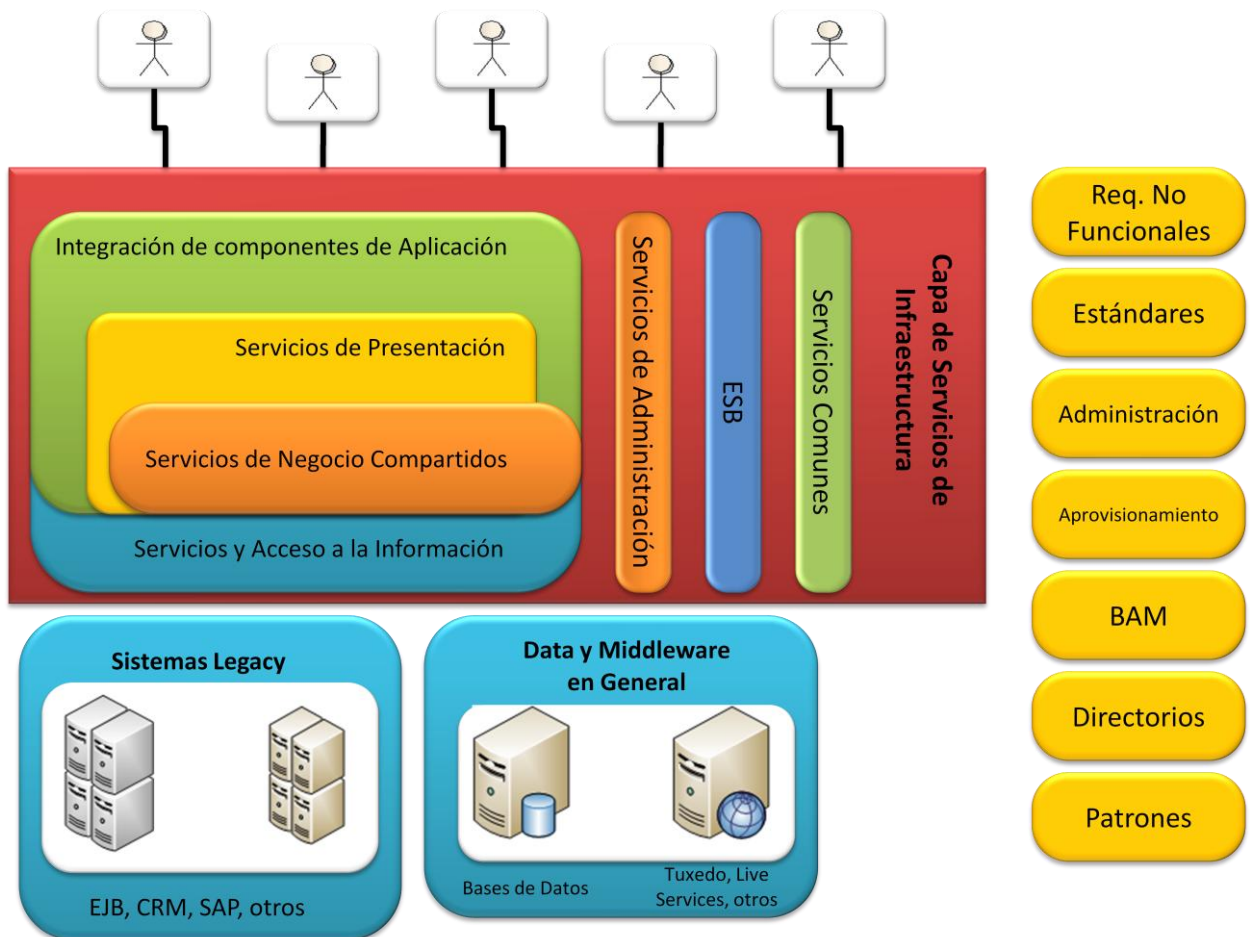


Figura 2.5.2.1 (Modelo de arquitectura referencial SOA)

La arquitectura busca el mantener separados a los usuarios de funcionalidades empresariales de aquellos sistemas responsables de proveer dichas funcionalidades, colocando infraestructura propia de servicios y entrega de estos entre ambos.

Como se puede observar en la **figura 2.5.2.1** la separación existente se genera a través de la capa de servicios de Infraestructura.

Servicios de Infraestructura basados en estándares componen la plataforma común de la que todos los demás tipos de servicios se desplegarán, estos servicios de Infraestructura comunes, vigentes para todos a los que se refiere son por lo general de tipo QoS, es decir condiciones de failover, independencia, etc. El ESB como ya se revisó es el encargado de proveer el ruteo para todos los servicios además de transformación de data de la misma manera en la que un típico bróker en middleware lo haría. Los servicios de administración permiten la gestión eficiente de los mismos.

Servicios que por lo general deben de ser de uso común a lo largo de toda organización, como auditoria y logging pueden ser añadidos a este esquema siempre y cuando sean servicios empresariales que cubran a toda la organización.

Finalmente cabe recalcar que conceptos como estos de infraestructura común para todas las aplicaciones lleva poco tiempo en el medio y se encuentra en proceso de adopción, pero es pieza fundamental para la construcción de una institución con orientación a servicios.

El concepto “Layered Architectural Model” es la base del modelo de referencia analizado y permite dividir las funciones básicas de una arquitectura SOA en las capas que se señalan en la **figura 2.5.2.1**.

2.6.3.1. Capa de Servicios y Acceso a la Información

La presente capa representa a la funcionalidad que existe actualmente en la institución que aplicará el modelo, estos servicios no solo se crean sino que de cierta forma se puede decir que son cosechados de aquellos recursos preexistentes para ser desplegados en la ya mencionada infraestructura común o compartida de la arquitectura y así poder cumplir con políticas institucionales de QoS. Es aquí en donde se estandariza el acceso tanto a las funcionalidades reusables como a la información misma de la empresa o institución. Tanto la información como las funcionalidades se deben de encapsular para que todo lo que consume sus servicios no necesite conocimientos granulares de la plataforma base de esta, ya sea funcionalidad o canal de acceso a la información.

La granularidad a este nivel es claramente visible al hablar del origen de cada funcionalidad o canal, en especial cuando se comienza a referir a aplicaciones que usan modelos de nivel empresarial, un claro ejemplo es JEE en el que sus conectores y adaptadores de conexión deben seguir la especificación que corresponde, en este caso J2CA.

Es por este motivo esencial que el aplicar correctamente los principios de esta capa proveerá a futuro de una base común para el acceso a los recursos empresariales propios de la institución sobre la que se podrá construir e integrar servicio que brinden mayor valor agregado y permitan la integración y fácil entendimiento con comunidades que sirven de apoyo en la construcción de mejores modelos de software.

2.6.3.2. Capa de Servicios de Negocio Compartidos

La presente capa representa el núcleo central de la funcionalidad del negocio. Su principal diferencia con la capa anteriormente revisada subyace en que esta ya opera sobre la información que provee la capa anterior, consolidándose como un servicio superior al no ser solo un proveedor de información.

Cabe recalcar que en un modelo escalonado en capas como este, cada capa encapsula sus procesos internos pasando a ser un proveedor de información para capas superiores y un consumidor de información para capas inferiores, por lo tanto el sentido de la presente diferenciación radica en los principios del modelo, al considerar que tanto las funcionalidades de los sistemas legacy como la información propia de la empresa son la data primitiva del modelo, estos gracias a los proceso de encapsulación de la capa de Servicios y Acceso a la Información no forman parte de la implementación del modelo y por tanto toda transformación y proceso sobre la información que proveen viene a ser considerada la primera en hacerse dentro del modelo. Este funcionamiento resulta familiar al compararlo al Modelo TCP y sus dos primeras capas.

Siguiendo la naturaleza del modelo, esta capa tiene la propiedad de ser compartida, es decir que además de ser desplegada sobre la infraestructura común, provee funcionalidad de negocio que resulta participativa y generalmente usada como un servicio común.

2.6.3.3. Capa de Presentación

Esta capa busca el escenificar a los componentes generales de presentación que usan las dos capas anteriormente vistas de forma colaborativa o de manera independiente para interactúa con recursos empresariales, un ejemplo común son los portlets. Un ejemplo sumamente típico puede ser el de los portlets que muestran la información del consumidor o cliente registrado en un portal empresarial una vez que este se encuentra logoneado.

2.6.3.4. Capa de Integración de Componentes de Aplicación

La presente es la encargada de orquestar los demás servicios y funcionalidades provenientes de cualquiera de las capas anteriormente revisadas para así proporcionar aplicaciones con funcionalidades de nivel superior y útiles para la institución. Aquí se encontrará las funcionalidades del negocio que se acoplan a lo que los técnicos en el negocio piensan que la tecnología debería proporcionar. En breves palabras implementa lógica de negocio de alto nivel y multi escalonada.

Un ejemplo típico puede ser el de un portal empresarial, u la creación de un proceso para crear y registrar productos. En este punto se puede hablar ya de aplicaciones que al embeber procesos de negocio permiten que este sea administrado y medido para así mantenerlo sumamente apegado a la línea de negocio.

2.6.3.5. Capa de Servicios de Infraestructura

Contando con las diversas capas de servicios hasta aquí analizados y la infraestructura común sobre la que se despliegan estas, existe a la par un vasto conjunto de requerimientos tecnológicos que deben de ser satisfechos para dar cumplimiento real al modelo. Servicios de infraestructura comunes como el empaquetamiento, despliegue, versionamiento y control de cambios deben seguir estándares comunes para así brindarle una mayor consistencia a la plataforma.

2.6.4. Ventajas de la Aplicación y uso de SOA

Las ventajas se pueden observar con claridad a lo largo de los capítulos anteriores, básicamente se ha separado las que generan un mayor impacto, sin dudas los listados taxativos de los beneficios de SOA pueden ser inmensos pero los que se han seleccionado son los que en una institución que decida adoptarla debiera considerar.

Reutilización de Componentes. El modelo permite el seleccionar de entre todas las funcionalidades a las mayormente usadas en el día a día, empaquetarlas usando estándares para emplearlo en futuras aplicaciones, una y otra vez, pudiendo dotar a nuevas aplicaciones con las mejores características de las aplicaciones legacy.

Expansión e integración con Clientes, Proveedores y Partners. El modelo busca el establecer estándares, estos se consisten en un lenguaje común, bajo el cual todos pueden entenderse y comunicarse, SOA se convierte en el marco adecuado, bajo el cual todos pueden comunicarse y entenderse.

Enfoque en la Función, no en la forma de conseguirlo. Tal vez en la balanza esta resulta ser la mayor ventaja de la aplicación del modelo, ya que este permite que los procesos de negocio se dediquen a pensar en los procesos de negocio y no en la forma, SOA permite ocultar los medios tecnológicos y dar el foco principal al negocio, la institución no se busca el definir alternativas tecnológicas ya que SOA se encuentra por detrás brindando acceso a las funcionalidades, en lugar de aquello esta se dedica a mejorar sus proceso de negocio.

2.7. Metodología extreme Programming (Programación Extrema)

2.7.1. Introducción a extreme Programming

Generalmente, en el momento en el que un individuo aprende a desarrollar, su principal actividad es la lectura, define posteriormente características del diseño, deduce la manera más apropiada de hacerlo, intue algo de Orientación a Objetos y crea un poco de código por sí solo para ir cimentando conocimientos, en periodos de tiempo regulados.

Los profesores que buscan orientarte, además de sugerir temas, te señalan la necesidad de tener prácticas más profesionales, como diagramar en UML la estructura, aunque este código generado semana a semana se represente de manera elemental y sencilla.

La sinergia que surge de estas combinaciones permite el analizar y albergar en el interior del naciente desarrollador las mejores características que le han sido mostradas, estas van desde pruebas hasta versionamiento, al conjunto de estas características intrínsecas del desarrollador se las conoce como la

noción de la ingeniería de Software, que de manera rústica en la mente del Desarrollador se forma e integra.

De forma común se puede ver cómo en las diferentes escuelas de ingeniería de software como usan metáforas comparativas con la construcción de edificios o la edificación de puentes, ya que se puede asemejarlos sumamente bien, en todos se tiene que inicialmente identificar las necesidades para posteriormente generar un diseño que pueda satisfacer estas, un proceso de construcción, finalmente probar su resistencia y demás.

La velocidad de entrega de resultados por tanto no solo dependerá del tamaño del proyecto sino de la estrategia que se use para acelerar los ciclos del proceso, en esto consiste eXtreme Programming, en constituirse en una estratagema de aceleración de procesos, una metodología acelerada, si el término cabe.

2.7.2. Definición y descripción de extreme Programming

Los señalados padres de XP nombrados típicamente son Ward Cunningham, como su inventor, Kent Beck como su articulador y Ron Jeffries como su mentalizador. Beck la diseñó mientras trabajaba en Chrysler, él describe a la metodología de la siguiente manera:

“Liviana, eficiente, de bajo riesgo, flexible, predecible, científica, una manera divertida de hacer Software”¹⁰

A su vez Ron Jeffries fue compañero de Beck mientras este trabajaba en Chrysler en el primer proyecto de gran envergadura en el que la metodología fue

¹⁰ Kent Beck, “Extreme Programming Explained: Embrace Change”, ed.2

aplicada, adicionalmente pasó por largos periodos de tiempo enseñando los principios de la metodología. De los principios obtenidos entre Beck y Jeffries se puede describir a eXtreme Programming como:

Una disciplina del desarrollo de Software que se basa en la simplicidad del pensamiento del desarrollador, utilizando como medio la comunicación y la retroalimentación pero sobre todo el valor de enfrentar los cambios dentro del diseño, cuya forma de trabajo se fundamenta en el trabajo en equipo de todo el grupo humano a través de prácticas simplificadas, bajo las suficientes fuentes de información como para que el equipo se ubique en el contexto y para acoplar el conjunto de principios del desarrollo que se encuentren usando a su caso en específico.

Como se ha dicho entonces sus bases radican en la colaboración tanto del equipo de trabajo, entre sí como la del usuario final, sus técnicas, prácticas y recomendaciones habitan en estos principios.

2.7.3. Introducción al uso de extreme Programming

XP se encuentra diseñado y dividido en procesos sumamente cortos ya que estos sencillos de planear, sencillos de ejecutar, sencillos de evaluar pero sobre todas estas características la primordial es que estos procesos pequeños son fáciles de reversar.

Este enfoque permite mantener los principios ya revisados entre otros presentes en la metodología, es decir mantiene la simplicidad, retroalimentación, reducción de riesgos, predictibilidad, entre otros.

Para poder avanzar al siguiente proceso el anterior debe de estar totalmente resuelto, lo que permite obtener como resulta, software que siempre es capaz de mostrar resultados, efectivamente compila y cumple de manera sencilla con los procesos planteados, lo que le permite el cumplir con los puntos anteriormente revisados.

A lo largo de la práctica laboral de un ingeniero de software, este puede encontrar varios proyectos de software que terminan en litigios severos, ejecución de multas y demás inconvenientes de carácter legal que se presentan en el modelo de trabajo adecuado a la realidad nacional y que al final terminan generando tensiones en todo el grupo de trabajo. Dichos conflictos radican en la grave dificultad de predecir el comportamiento de los esquemas de trabajo, sin dudas generando incumplimiento de expectativas, y decepciones en el cliente. La ventaja en frente de esta realidad que brinda XP es sin dudas que el cliente ya no se encuentra al otro lado de la mesa, más bien es parte integral del equipo de desarrollo, vive con los desarrolladores los miles de problemas que pueden presentarse, es el responsable de redirigir los esfuerzos y orientarlos al cumplimiento de sus expectativas, en lugar de motivar una guerra de defensa de puntos de contratos que al final no benefician a nadie, es más el calendario de entregas no puede retrasarse ya que el cliente se encuentra totalmente al tanto de su proceso.

Para finalizar la presente sección, cabe el sugerir que todos los desarrolladores poseen la misma responsabilidad y autoridad en cuanto a las herramientas de software que se administren y la metodología en sí, quien detecte un problema por lo tanto se encuentra facultado para resolverlo, quien se encuentre desarrollando un módulo determinado es el responsable directo de su

diseño y desarrollo, quien encuentre la forma de mejorar un proceso es el responsable de implementarlo. Ya que no hay un proceso de diseño integro y formal, cada par de desarrolladores, es el responsable del diseño de sus procesos y tareas y por tanto la estructura final solo se puede obtener con el producto final completamente desarrollado.

2.7.4. Características de extreme Programming

La metodología inicialmente nos permitía el identificar doce puntos de valor que subyacen y se alimentan de los valores anteriormente expuestos, tomando en cuenta el hecho presente de su amplia utilización, y por respeto a sus principios esta sigue evolucionando, creciendo y cambiando. Muchas de sus prácticas han sido refactorizadas para brindar mayor eficiencia, han modificado su enfoque, han sido divididas, han sido trastornadas. A pesar de que cada punto actúa por sí solo, su interrelación permite la generación de cierta sinergia que admite el producir un producto de alta calidad. A continuación se anotaran estas y aunque cada una es capaz de autodescribirse se analizarán a detalle a posteriori y revisando los cambios que esas han sufrido en la línea del tiempo, de manera adicional se desea hacer notar que su compleja interrelación es la que en realidad cobra relevancia.

Las doce características inicialmente identificadas eran:

1. La planeación,
2. entregas pequeñas,
3. metáfora,
4. diseño simplificado,
5. pruebas,

6. re-edición,
7. programación en paralelo,
8. autoría colectiva del fuente,
9. integración continua,
10. cuarenta horas a la semana,
11. usuario in sito y,
12. estandarización de codificación.

Con los años cuatro de estas características han mutado, primero se comenzará hablando de las pruebas, ya que estas se han dividido en dos separadas, aquellas que realiza el cliente y aquellas que se realizan durante el desarrollo. La re-edición ha cambiado nombre, ahora hay que referirse a esta como el mejoramiento del diseño para enfatizar el enfoque que le otorga al diseño la metodología. El término de cuarenta horas a la semana, ha sido reemplazado por el de lapso adecuado, dado que en realidad el tiempo para el desarrollo no puede ser encasillado de esta manera, puede ser más e inclusive menos, todo dependerá del escenario de trabajo en el que se encuentra Finalmente el concepto referido al usuario in sito ha sido modificado para que albergue no solo a un cliente sino a todo el equipo de trabajo, este puede ir desde administrador hasta el desarrollador junior integrado.

Al final de este proceso se han obtenido las trece características fundamentales para la implementación de XP y como se anotó con anterioridad aquí constan especificadas a detalle:

Todo el equipo (inicialmente identificado como cliente in sito): Xp se basa en la disolución de la barrera natural existente entre el cliente y el equipo de desarrollo integrándolo, la visión ampliada de esta característica, criterio conlleva la búsqueda de la integración de cada miembro del mismo, desde el desarrollador de menor estirpe hasta el gerente de proyecto.

Metáfora: El equipo completo debe de expresarse acerca del sistema o aplicación en términos de una figura común, una similar imagen mental que abraque la funcionalidad global del sistema, esto les permite a todos comunicar de mejor manera sus ideas, en especial entre miembros del equipo de áreas especializadas y disímiles, prestando una valiosa colaboración a la mantención de la integridad del sistema.

Planeación: Dentro de XP la planeación permite definir el cómo del siguiente paso en el continuo ir y venir de las entregas, basados en la participación directa del usuario, ya que a través de las historias de usuario se define luego de cada entrega, que nuevos requerimientos se cumplirán en la siguiente entrega, bajo la libre elección del cliente, en el marco tecnológico que corresponda al momento.

Diseño simplificado: El nivel de complejidad del diseño de una aplicación debe de mantenerse tan simple como la presente nivel funcional lo permita. El principio aquí es mantener el diseño incremental y si esta es la meta este no debe abarcar más de aquello que señale la presente iteración. Como se verá en los siguientes párrafos, si el diseño de una funcionalidad se volviese inmanejable siempre existe el refactorización.

Entregas pequeñas: Cada equipo de trabajo que emplee la metodología debe de liberar software probado y funcional, que cumpla con las características especificadas por el cliente de manera muy frecuente, por sentido común estas liberaciones son muy pequeñas ya que se realizan por iteración, las mismas fluctúan en periodos generalmente de dos semanas como máximo. El resultado es analizado a profundidad por el usuario, que luego de comprender el impacto sobre el sistema de la presente entrega deberá decidir los siguientes puntos a entregar por parte del equipo de trabajo.

Pruebas por parte del cliente: Estas son aquellas pruebas que lleva a cabo el cliente sobre las entregas de software que se realizan en cada iteración, en ellas el usuario busca el medir cuán apegadas se encuentran estas liberaciones a las historias de usuario y son usada a posteriori por el equipo de desarrollo para dar por cumplida una determinada historia de usuario.

Programación en paralelo: Técnica que consiste en colocar a dos desarrolladores de similares características técnicas a trabajar sobre un único equipo, cada línea de código en la medida de lo posible debe de ser implementada de esta manera al trabajar con XP. Así mientras un individuo es el encargado de manejar el teclado, su compañero realiza tareas de apoyo, que consisten básicamente en corregir errores tipográficos y sugerir mejoras al proceso, en cualquier instante ambos pueden intercambiar sus lugares.

Pruebas durante el desarrollo: Al momento en el que el desarrollador se fija un meta dentro del desarrollo, la que deberá ser pequeña en envergadura, antes de iniciar el proceso de codificación, deberá escribir las pruebas correspondientes que verificarán su comportamiento, esto disminuye la incidencia de errores y por tanto su impacto final en la eficiencia del proceso de desarrollo.

Mejoramiento del diseño: Si el equipo encuentra en cualquier momento alguna falencia en el diseño, simplemente lo mejora, así se puede mantener un proceso de pulido incremental del código. Al final de todo el ciclo, que inicia con diseños bastante simples por lo general XP entrega código con valiosas características en cuanto a diseño, rendimiento y eficiencia.

Autoría colectiva del fuente: El equipo como unidad es el que posee la autoría sobre el software. Si bien cada pequeño componente y línea de código fue escrita en principio por un desarrollador, este hecho pierde importancia y vigencia, al otorgársele a cada miembro del equipo la facultad de modificar cualquier sección de código el momento que desee y considere va a significar una mejora al proceso, con esto no se desvirtúa el uso de un controlador de versionamiento, que resulta sumamente útil en casos específicos, como al momento en el que varios grupos trabajan sobre un mismo proceso, este ítem más bien se refiere a que al final del proceso el desarrollador piense que puede encontrar sus fuentes en el mismo estado en el que los dejó la primera vez que los escribió.

Integración continua: A todo momento, sin considerar su actual nivel funcional, el sistema o aplicativo deberá compilar, poder ejecutarse y además pasar todas las pruebas descritas hasta el momento ya que este también es un proceso continuo. Cada sección de código agregada al sistema deberá cumplir con los mismos criterios, y una vez terminada la integración nuevamente la aplicación debe poder compilar, ejecutarse y pasar todas las pruebas hasta el momento descritas.

Lapso adecuado: La carga de trabajo existente debe de ser similar a lo largo de cada iteración. Excepciones en el tiempo conducen al desgaste, aumento de frecuencia en la ocurrencia de errores críticos, entre otros. Mantener por tanto un lapso adecuado para realizar el trabajo permite a los desarrolladores el crear software de mejor calidad en periodos de tiempo cada vez menores.

Estandarización de codificación: el resultado final no puede ni debe verse como la amalgama de varios estilos de codificación. El equipo completo debe de aprender a adherirse a un conjunto de normas y estándares completos y vigentes para cada miembro del equipo, los mismos que deberán de ser empleados por todos sus miembros, línea tras línea.

Estas que se acaban de revisar son las características consideradas comunes de XP, si bien a lo largo del tiempo cada proyecto conlleva necesidades diferentes, está en los líderes de estos el definir que características de XP va a emplear en cada etapa, según su criterio y convicción el seleccionar los más adecuados, cabe recalcar que al final ninguna metodología resulta perfecta y que esta debe de acomodarse a necesidades puntuales. A pesar de esto es de

notable importancia el anotar que la sinergia que producen todos estos elementos al trabajar juntos resulta altamente sorprendente y debería de ser experimentada antes de realizar cualquier tipo de edición al proceso de XP.

2.7.5. Fases de extreme Programming

2.7.5.1. Planificación dentro de extreme Programming

La fase de planificación nos permite establecer los principios básicos sobre los que se trabajará a lo largo del proyecto. Esta entrega una serie de procedimientos, algunos que al final se transforman en hitos que permiten identificar de forma rápida las principales necesidades de los usuarios, estos son:

Historias de Usuarios: Su función resulta similar a la de los casos de uso y su función principal es colaborar con data suficiente como para realizar estimaciones, y tomar conciencia de los requerimientos del aplicativo, reemplazando así a grandes documentos de requerimientos que si bien permiten la detallada detección de necesidades, demoran gran cantidad de tiempo en su elaboración. Cada historia debe de expresar la forma en la que las necesidades del usuario son satisfechas, por lo tanto debe elaborarse por el mismo usuario. Cabe aclarar que estas no consisten en meras descripciones de las interfaces de usuario, ya que pueden abarcar inclusive procesos.

Planeación de Entregas: Se refiere a una reunión en la que se define el plan general de entregas, en esta se deben de llegar a acuerdos como las frecuencias de las iteraciones, entre otros puntos de acuerdo sobre los puntos a desarrollar. En esta los desarrolladores evalúan el tiempo que puede tomar la realización de una historia de usuario, esto bajo el criterio propio de los

desarrolladores, definiendo así que se implementará en la siguiente iteración. Esta reunión se puede realizar varias veces gracias a lo iterativo de la metodología.

Entregas Periódicas: El proyecto se divide en iteraciones cortas que van de una a tres semanas de trabajo.

Planeación de Iteraciones: en cada ocasión que una iteración se dé por iniciada se deberá llevar a cabo una reunión en la que se fijen los puntos comunes, metas y tiempos a alcanzar empleando principios como Spike.

2.7.5.2. Diseño dentro de extreme Programming

En la presente fase se debe de aterrizar los objetivos encontrados en las historias de usuario en hechos concretos generando los esquemas base de la aplicación, diferenciándose al aplicar los siguientes parámetros:

Simplicidad: Diseño simplificado, ya que lo que extreme Programming busca es el reducir los tiempos de construcción, bajo la consideración básica de que el diseño mientras más elemental, mas rápido puede ser desarrollado, si se encuentran en ideas abstractas que busquen el reducir líneas de código, la elección en este caso será el simplemente repetirlos sin llegar a dichas abstracciones.

Metáfora: saber escoger una metáfora adecuada al sistema, esta es en sí misma una idea general de lo que se quiere en la aplicación con ciertas características especiales. Su cualidad preponderante es el poder demostrar el diseño del sistema a personas que apenas toman conocimiento de la aplicación, sin la necesidad de explorar gran cantidad de documentación técnica para

comprenderlo. Así se puede conseguir que nuevo personal se involucre y produzca de manera casi inmediata

Tarjetas CRC: Se deben de emplear las llamadas tarjetas CRC (Class, Responsabilites, Collaboration). Su principal función es promover la colaboración al permitir que la gran parte de los equipos de desarrollo colaboren al diseño de mejores aplicaciones. Cada tarjeta representa una clase específica, en ella se detallan sus principales funciones y las clases que interactúan con ellas, se revisa el proceso que las envuelve y su interacción con el medio, esto permite pulir el proceso y el flujo de información entre objetos.

Soluciones Spike: Para comprobar posibles rutas de diseño se deben de generar pequeños programas para dar respuesta a problemas elementales y así detectar soluciones válidas. Estas soluciones solo nos permiten el poner el problema bajo la lupa de la exanimación e ignora variables ambientales que influyen de una u otra forma el comportamiento de la solución

No Considerar Funcionalidades en etapas iniciales: Limitar la aparición de características funcionales que no tengan que ver con el funcionamiento central del aplicativo es una fuente de tiempo sumamente fiable, ya que apenas el 10% de todas las consideraciones funcionales hechas en esta etapa terminan plasmándose en el proyecto al final del mismo.

Constante Refactorización: Durante el proceso de software generalmente acelerado que conlleva el emplear la presente metodología, se encuentran a desarrolladores que se sumergen dentro de su código y pierden la orientación y el enfoque al momento en que dificultades comienzan a surgir, de manera constante debe de evaluarse todo aquello que se encuentre demás, eliminarlos y pulirlos

haciendo al código de fácil lectura y acceso. Cada desarrollador debe asegurarse de que su código fue escrito tan solo una vez y en un solo lugar.

2.7.5.3. Desarrollo dentro de extreme Programming

La metodología en este punto busca esquematizar de la mejor manera posible la serie de normas y proceso que el desarrollador debe llevar a cabo al momento de comenzar la codificación, entre estas y las recomendaciones, principios e hitos:

Comunicación Constante con el Cliente: En extreme Programming resulta de vital importancia el tener al usuario siempre disponible, no solo para llevar a cabo tareas de apoyo sino como parte del equipo de desarrollo ya que todas las fases de la metodología requieren de contacto directo con este. La cercanía con el cliente permite entregar un producto acorde a sus necesidades, establecido de la manera en la que el usuario lo definió en las Historias de etapas anteriores y corrigiendo cualquier anomalía en el presente.

Codificar de acuerdo a Estándares: Previo a iniciar la codificación se deben de llegar a acuerdos sobre las normas de codificación que se emplearan, para que así todo el equipo hable un mismo idioma, y el resultado pueda ser racionalmente entendido.

Programación en Paralelo: Todo el código y mientras sea posible debe de ser generado utilizando este método que consiste en el trabajo en equipo de dos desarrolladores, uno es el encargado de proveer ideas e iniciativas, el otro adicional a esto tiene la obligación de tipear el contenido. Cabe aclarar que no se trata de una especie de asesoría, sino más bien de trabajo comunitario.

Integración Secuencial: Al momento esperado del fin de la codificación en una iteración determinada llega la crítica hora de aunar esfuerzos para unificar el código, el problema que conlleva dicha actividad subyace en que pruebas de integración no se pueden realizar si la integración al igual que el desarrollo se realiza en paralelo, y aunque estas resulten consistentes con versiones anteriores del proyecto, al ser la propiedad del código comunitaria, una variación en el mismo a cualquier momento puede ocasionar serios inconvenientes. Los grupos de trabajo suelen para solucionarlo definir grupos dedicados únicamente a la integración, o usar subversionamiento, pero ambas alternativas presentan problemas sugiriendo el mantener la integración en paralelo, es por esto que en XP se sugiere en ese caso utilizar un principio ya mencionado en esta sección, la propiedad comunitaria del código, que unida a la integración secuencial, cada grupo que quiera introducir nuevo código debe el turnarse y realizar pruebas de integración, si todo va bien entonces pasa el siguiente grupo y así de forma subsiguiente.

Integración Continua: Los desarrolladores divididos en pares deben de actualizar el repositorio con nuevas secciones de código probadas constantemente, en lo posible de hecho evitar el dejar pasar un solo día con cambios que reportar al repositorio.

Autoría colectiva Del Fuente: Este punto es sin dudas una vital característica de XP, de hecho su uso se encuentra fuertemente recomendado, ya que permite el flujo de nuevas ideas y revitaliza los procesos de refactorización, sobre todo permite la resolución de bugs agregando modelos de funcionalidad al código existente.

2.7.5.4. Pruebas dentro de extreme Programming

El proceso de pruebas que se considera dentro de extreme Programming tiene como característica identificadora el ser sumamente extenso, pero no se refiere a un lapso posterior al de la codificación, en el que se dedican incesantes horas a la verificación de comportamientos, más bien es un tema que pasa por la certificación de que cada línea de código integrada posee todas las características técnicas como para no filtrar ningún error relevante. Los puntos a considerar dentro del periodo de pruebas son:

Pruebas Unitarias: Las pruebas unitarias se consisten en la base de XP al tener que ser ejecutadas como requisito indispensable para la integración. Se recomienda el usar framework especializado para las mismas, adicionalmente de validar con este tipo de pruebas toda clase y método perteneciente a la aplicación al momento de ser creadas y definidas, adicionalmente se recomienda el definir casos de prueba de forma anterior a la codificación, cabe recalcar que métodos típicos como los que componen propiedades y accesores se omiten por obvias razones. Como nota adicional se puede decir que todo el código debe de pasar las pruebas unitarias antes de que este pueda ser integrado.

Detección de Bugs: En el momento en el que el equipo detecte la existencia de un bug en cualquier sección de código ya compilado e integrado, recordando que la pertenencia del código es universal a todo el grupo de trabajo, se debe de proceder a crear un caso de prueba para el garantizar no solo que este se solucione de manera satisfactoria sino que tal inconveniente no vuelva a aparecer. El definir pruebas de aceptación como se analizará en el siguiente acápite, permitirá el graficar correctamente la problemática a los desarrolladores

por parte de los usuarios y así lograr dirigir el total de su esfuerzo a un punto en específico para lograr resultados. Adicionalmente en el caso de un fracaso los test unitarios permitirían desde un inicio el tener un feedback inmediato de la problemática.

Pruebas de Aceptación: Estas debe de ejecutarse de manera corriente, nacen además de las historias de usuario. A partir de las historias seleccionadas para la presente funcionalidad, se debe de considerar por parte del usuario todo escenario que deba de ser evaluado, cabe recalcar que pruebas de aceptación definidas por el usuario no deben de tener ninguna limitante, su existencia totalizada se remite únicamente al criterio del usuario, y sus resultados deberán de ser publicados, ya que el principal objetivo de extender las pruebas a este nivel es precisamente el evitar a toda costa al tener que corregir errores en fases terminales del proyecto.

2.8. Herramientas de Desarrollo y Diseño

2.8.1. Herramienta de Diseño: Microsoft Visio 2007, Enterprise Architect 7.5, VizSequence 0.11

Estas herramientas permiten la rápida diagramación de esquemas de UML así como diseños estructurales informáticos. Los tres resultan en herramientas altamente útiles para la generación de diagramas técnicos con métodos contemporáneos actualizados de manera constante, con técnicas que incluyen ingeniería reversa y mapeos sincronizados con proyectos de software bajo la plataforma actual.

Cada una de las herramientas aquí mencionadas permite la especialización de la diagramación y la segmentación de tareas, es así que Microsoft Office Visio

2007 permite el trazar diseños estructurales, Enterprise Architect, nos permitirá el obtener diagramas de UML típicos, apoyado por VizSequence, con el que se obtendrá diagramas provenientes de los Assemblies, esta combinación, por la metodología de trabajo de eXtreme Programming resulta ideal para la obtención de los ya mencionados diagramas y diseños varios.

2.8.2. IDE: Visual Studio 2008

El galardonado como mejor IDE (Integrated Development Enviroment) del mundo en la gala de InfoWorld en el 2009, se constituye en la mejor alternativa para el desarrollo de software bajo plataforma .Net, expandiendo sus funcionalidades anteriores para incluir un completo soporte para SOA, desarrollo sobre WPF (Windows Presentation Foundation) y Web 2.0, entre otras. Entre las pieza más importantes incluidas se encuentra su fácil integración con la DATA a través de su soporte para LINQ, y su habilidad destacada de interactuar con Silvelight, permite un rápido acceso a todos los posibles enfoques de una aplicación comercial end-to-end, ratificándolo como una valiosa alternativa al momento de enfocarse en el desarrollo.

Las versiones del IDE son varias en esta edición, aquí se presenta un listado de las mismas, ordenada por el soporte de cada una de ellas a características y recursos adicionales para el desarrollador, comenzando desde sus versiones básicas a la más completas:

- Visual web Developer 2008 Express Edition,
- Visual Basic 2008 express Edition,
- C# 2008 express Edition,
- C++ 2008 express Edition,

- Visual Studio 2008 Standard Edition,
- Visual Studio 2008 Professional Edition,
- Visual Studio Team System 2008 Architecture Edition,
- Visual Studio Team System 2008 Database Edition,
- Visual Studio Team System 2008 Development Edition,
- Visual Studio Team System 2008 Test Edition,
- Visual Studio Team System 2008 Team Suite.

Las diferentes ediciones y su contenido resulta sumamente extenso y un tema de análisis posiblemente motivo de una investigación de carácter mucho más profundo que el presente marco, pero en síntesis las versiones Express que constan al principio de la lista y que resultan gratuitas brindan soporte a funciones elementales del Framework y carecen de recursos como el soporte para dispositivos móviles e para LINQ, como ejemplo, la versión Developer es la primera por la que se paga un monto, y da soporte a todos los lenguajes soportados por el framework, adicionalmente la versión profesional, lleva las características del IDE a un nivel superior al ofrecer ciertas características específicas para la elaboración de casos de prueba entre otros. Las versiones Team en cambio son el conjunto completo de características que puede ofrecer el Visual Studio Foundation a un tema en específico, mientras que la versión Team Suite es el conjunto completo d todo el bagaje de herramientas provenientes de VSF.

Como características adicionales a las ya mencionadas hasta el momento se puede agregar que el presente IDE además del soporte para todas las plataformas Foundation de .Net y a los lenguajes entendidos por el compilador JIT

(Just in Time), permite el compilar cada proyecto bajo cualquier versión anterior del framework de .Net permitiéndole al usuario el escoger el desde el inicio el modelo arquitectónico.

2.8.3. Plataforma: Microsoft .Net Framework Versión 3.5

Desde que el desarrollo de software se constituyó en una industria lucrativa, los cambios sobre los paradigmas y su enfoque motivados por ingentes capitales de inversión han permitido un constante cambio en ellos, nuevas visiones que se integran y mejoran la manera de administrar y ver el software, junto con su desempeño y escalabilidad. Es por esto que al menos en periodos de tiempo considerables, la magnitud del impacto del cambio es radical, esto puede verse aproximadamente una vez por década, en el presente caso analizado, Microsoft ha presentado variaciones fundamentales en el diseño de sus modelos arquitectónicos para la entrega de aplicaciones a la medida en lapsos equivalentes, el último trastorno fundamental en su enfoque se presentó justamente al inicio de la década del 2000, la introducción del framework y la migración de todo el soporte de la tecnología de Microsoft al framework que tuvo su primer release oficial en su versión 1.0, continuo con su versión estabilizada, la 1.14 y así hasta la fecha en la que se cuenta ya con su versión 4.0.

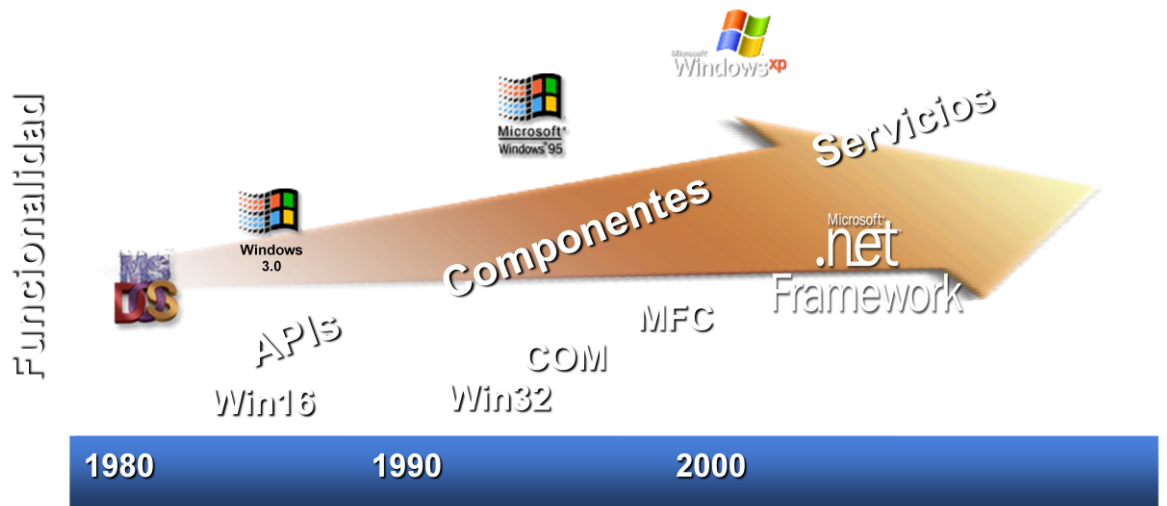


Figura 2.7.3.1¹¹ (Transcurso de .Net a lo largo de la historia)

Esta evolución puede observarse claramente en la **figura 2.7.3.1** tomada del programa de aprendizaje de Microsoft Corp. En el mismo se puede divisar el surgimiento de la plataforma como parte de un cambio de ciclo en la forma de ver y usar paradigmas, como se mencionó anteriormente.

La motivación para el cambio de enfoque al interior de Microsoft Corp. Surge por la creciente necesidad de satisfacción de nuevas demandas por parte de los usuarios de internet y nuevas factores que se introdujeron en el juego a principios de la década, entre ellos y por anotar algunos:

- Acceso a Internet en masificación, surgiendo así conexiones de alta velocidad accesibles al público en general.
- Equipos tecnológicos diseñados para la telecomunicación, como celulares y pocket PCs.

¹¹ Microsoft corp., “Programa Desarrollador cinco Estrellas: Introducción a Microsoft .NET”, diap. 6, MSDN en Español

- El aumento del poderío de x86 como arquitectura, y su común uso y adopción.
- Inundación de estándares sobre Internet, lo que permite establecer enlaces de comunicación estabilizados entre plataformas disímiles.

La presente plataforma al ser la alternativa de Microsoft para estos años venideros se convierte en una plataforma de desarrollo y ejecución. Desarrollo porque contiene una serie de recursos y servicios que le permiten al desarrollador el construir una variada suite de aplicaciones que abarcan aspectos granulares en orientación del software contemporánea. Ejecución, ya que proporciona a las aplicaciones de mecanismos y herramientas que resultan eficientes y adecuadas para poder poner en marchas todo desarrollo bajo la plataforma.

Para poder alcanzar dicho nivel de funcionalidad, el framework se construyó de manera modular, aquí se anotan algunos de sus principales componentes genéricos:

- Un ambiente de ejecución unificado al que también se lo denomina “Runtime”, este es un conjunto de paquetes de software encargados de mantener el ciclo de vida de la aplicación completo, este es el que se encarga de administrar su relación con el Sistema operativo para que este no tenga que arreglárselas directamente con este.

- Serie de APIs que poseen funcionalidad que puede considerarse básica, y componentes para rehusos que típicamente son usadas, todos estos componentes vienen compilados y empaquetados.
- Compiladores y Linkers para una gran cantidad de lenguajes, que le permiten al desarrollador escoger el más acorde a sus necesidades.
- Programas utilitarios que debieran permitir el simplificar el desarrollo de aplicaciones.
- Documentación y guías básicas sobre el diseño arquitectónico de la plataforma, junto con patrones y mejores prácticas que se recomiendan seguir en el desarrollo de aplicaciones con la presente plataforma.

El framework de .Net, es la base de toda la línea .NET, convirtiéndose en un elemento indispensable tanto para ejecutar como para desarrollar aplicaciones bajo cualquier entorno.

La plataforma tiene tres variantes principales, cualquiera de estas en cualquier versión puede ser descargada de forma gratuita del Internet, estas son:

- .NET Framework Redistributable Package
- .NET Framework SDK
- .NET Compact Framework

En síntesis el framework toma el código escrito en lenguajes de alto nivel, lo pre-compila y lo transforma a un lenguaje común basándose en una extensa especificación de lenguaje común obteniendo como resultado el fuente en un lenguaje intermedio, denominado MSIL (Microsoft Intermediate Language), este

código a su vez es interpretado por el entorno de ejecución común para finalmente obtener el resultado que se puede observar.

Los componentes típicos del framework para su ejecución única son los ya nombrados, es decir:

- Lenguajes de Alto Nivel,
- Common Language Specification (Especificación de Lenguaje Común),
- Librerías de Clases de .NET en cualquiera de sus variaciones dependiendo de la versión del framework, y
 - Common Language Runtime (Entorno de Ejecución Común).

La evolución tecnológica durante la década ha permitido la adecuación del framework .NET, que ha presentado de manera posterior a su salida en la versión 2.0 cambios fundamentales, agregando a los componentes actuales que han sido escuetamente mencionados con anterioridad los siguientes componentes, eso a partir de su versión 3, y mejorados para su versión 3.5:

- Windows Presentation Foundation,
- Windows Workflow Foundation,
- Windows Communication Foundation,
- Windows CardSpace

Cada una de las cuales se erige como una fuerte plataforma estructural para soportar carga de trabajo sobre:

- Desarrollo de aplicaciones en 3D,

- Diseño e implementación de Workflows (Contendiente principal del uso de BPM),
- soporte para el modelo arquitectónico SOA,
- consolidar el manejo de seguridad en aplicaciones, entre otros.

Entre otras características ahora soportadas por el framework en su release 3.5, se encuentra el soporte a LINQ, entre otros.

2.8.4. Plataforma Para el Desarrollo de Web Services : Windows Communication Foundation

Este proyecto de Microsoft de nombre clave “Indigo”, que hoy por hoy es una realidad en la industria, es la última tendencia diseñada por Microsoft Corp. para la creación y publicación de aplicaciones distribuidas. En breves palabras es un modelo de programación unificado diseñado para la implementación de aplicaciones de arquitectura distribuida.

Con la globalización, como se ha visto anteriormente la necesidad de comunicar aplicaciones de plataformas de diversas características se volvió una necesidad imperante. WCF fue concebido para asimilar dichas necesidades y darles simplicidad a los procesos de desarrollo de soluciones distribuidas.

Su proceso de desarrollo como se mencionó es unificado, lo que permite el dar solución a problemas complejos de una manera común, evitando la repetición de proceso innecesaria.

La interoperabilidad mencionada que permite el trabajo entre plataformas viene de la fortaleza en la que resulta el trabajar con SOAP, ya que administra así

XML. Se podría decir que es una evolución de los Web Services tradicionales de Microsoft, al utilizar los mismos estándares.

En cuanto al transporte de mensaje, WCF permite el definir un protocolo neutral con formato de mensajes neutral, permitiendo escoger al gusto del diseñador ambos, desde HTTP hasta TCP o cualquier mecanismo que resulte escogido. El proveedor del transporte resulta indiferente al servicio. Esta característica fundamental de separar el código del transporte le permite el mantener las funciones de capas altamente separadas permitiendo así una mejor gobernanza, además de facilitar el trabajo del desarrollador, que no tendrá que dedicarle horas al re-ensamblaje de la trama al cambiar de protocolo de comunicación.

La metodología de trabajo de un WCF permite la separación de tareas por los así llamados “contratos”, que son definiciones técnicas de aspectos específicos relevantes al servicio, los mismos definen no solo las tareas básicas del servicio sino otras características funcionales, estos tipos de contratos son:

Data Contract: Permite el definir las estructuras de los datos abstractos que manejará el servicio, es así que si se tiene un dato abstracto denominado persona, con atributos propios como nombre y edad, estos deberán de estar definidos dentro del Data Contract para poder ser tomados en cuenta.

Message Contract: Este a su vez permite el determinar el formato del mensaje cuando se envía o se recibe, como ejemplo se puede anotar, que datos se esperan tanto en la cabecera como en el cuerpo en una respuesta típica.

Service Contract: En el presente contrato se debe de fijar exactamente lo que hace el servicio. En él se enumeran el conjunto de acciones de este. Al grupo conformado por estas acciones propias del servicio se las denomina operaciones del servicio. Como ejemplo se puede anotar a un Service Contract denominado ObetnerPais el mismo realizaría la tarea de la obtención del país de un usuario común. Finalmente cabe el mencionar que un servicio basado en WCF debe de tener por lo menos un Service Contract para poder funcionar.

Policy & Binding: Es en este contract en el que precisamente se define la manera en la que este se comunica con el mundo. Aquí se anotan las especificaciones del transporte así como requerimientos de seguridad y confiabilidad, características que le permiten el brindar QoS.

2.8.5. Servidor de Base de Datos: Microsoft SQL Server 2005

Motor de base de datos perteneciente a la casa Microsoft Corp. Con varios años en el mercado, posee una fuerte orientación empresarial, convirtiéndose en la alternativa predilecta de usuarios no Oracle siendo su principal rival en su línea estrella de bases de datos.

El motor maneja una variación de SQL conocida como Transact-SQL, lenguaje implementado para la construcción de objetos dentro del motor, el mismo que es usado de manera exclusiva dentro de ambientes Microsoft.

Su nivel de integración con la plataforma nivel es nativo, a tal punto que un motor de limitadas capacidades viene agregado a la instalación de ciertas versiones de Visual Studio a partir de su versión 2005. Es la elección estandarizada por defecto en construcciones atadas a la plataforma ya que pueden aumentar el rendimiento de aplicación de stack Microsoft de manera significativa.

Entre sus características repotenciadas y otras vistas en motores empresariales comerciales y adoptados por el motor a partir de la versión 2005 se pueden enumerar:

- Soporte de Transacciones,
- Inclusión y Soporte a Objetos propios de la base,
- Entorno gráfico ágil y de gran calidad,
- Administración remota, entre otros.

Aunque existen ya versiones publicadas del motor muchos más recientes gran parte de usuarios consideran hoy en día que la versión 2005 resuelve muchos de los problemas de sus antecesores y que no es necesario un proceso de migración, Microsoft Corp. A su vez brinda programas de migración acordes a las necesidades de sus clientes, entendiendo que las características de la ya nombrada versión proporcionan un nivel de integración muy satisfactorio con los componentes que generalmente usan estos motores.

La base en la actualidad cuenta ya con su tercer Service Pack, que son conjuntos de actualizaciones que buscan corregir bugs registrados.

Las versiones de Microsoft SQL Server 2005 son:

- SQL Server 2005 Developer Edition,
- SQL Server 2005 Compact Edition,
- SQL Server 2005 Express Edition,
- SQL Server 2005 Workgroup Edition,
- SQL Server 2005 Standard Edition, y
- SQL Server 2005 Enterprise Edition

Estas versiones se diferencian entre sí por el nivel de prestación, número de usuarios, niveles de escalabilidad y accesibilidad, y niveles de administración y monitoreo, que separan a sus versiones Enterprise de las versiones Express.

2.8.6. Acceso a Información: Live Services y Fuentes de Terceros

Live Services son una vasta gama de servicios provistos por Microsoft Corp. Que permiten a los desarrolladores que los usen, el integrar la información proveniente de las experiencias de casi 700 millones de clientes registrados en todo el mundo de aplicaciones Windows Live con la que generan sus propias aplicaciones, permitiendo así el uso de un banco de datos considerable. Uno de los ejemplos típicos que se dan para representar a los servicios Live, es Live ID, servicio de identificación de usuarios, pionero en SSO (Single Sign On), sobre la red, que permite con el administrar el acceso a una aplicación genérica con usuarios pertenecientes a Live ID como cuentahabientes de Hotmail que por defecto se integran a Live ID. Otros ejemplos son Live Messenger para mensajería instantánea, entre otros que conforman un extenso framework de servicios que permiten el compartir información entre usuarios.

Estos servicios pueden ser integrados en modelos arquitectónicos SOA, de hecho como toda fuente de terceros publicada bajo formatos estandarizados, puede ser consumida por un motor de orquestación o el formar parte de cualquiera de las capas del modelo SOA, de hecho generalmente son sumamente usadas y empleadas para el ensamblaje de porlets por ejemplo.

2.8.7. Servidor de Aplicaciones, Web: Internet Information Server 7

Internet Information Services, en versión 7 es el servidor de aplicaciones que se liberó junto con Windows Vista y Windows 2008 Server, es la última versión del ya conocido servidor Microsoft. Se encuentra construido sobre la base de tres principios fundamentales que son la escalabilidad, la confiabilidad y la seguridad, para poder conseguirlo Microsoft Corp. desarrollo las siguientes características:

- **Seguridad:** Arquitectura modular, personalizable, se puede controlar el total de componentes que se instalan, es decir que solo aquellos indispensables se integran, mientras que componentes de seguridad como el URLScan vienen integrados.
- **Capacidad de Ampliación:** Se ha definido una nueva API de Win32 que permite dar soporte a componentes HttpModules, y permite definir nuevos módulos o componentes para el IIS a través de la creación de aplicaciones ASP.NET
- **Configuración:** Permite realizar configuraciones bajo XML, que pueden ser distribuidas y delegables, así otorgar privilegios de administración directamente al propietario de la aplicación.

- **Administración del Sistema:** Renovación sobre IIS Manager, actualmente existe acceso vía línea de comandos, y permite administración remota a través de HTTP y HTTPS.
- **Diagnóstico:** Se instaló una nueva API de diagnóstico que permite determinar y rastrear todo inconveniente de tipo modular.

2.8.8. Enterprise Service Bus: Aqualogic Service Bus 3

Aqualogic Service Bus es un ESB basado en configuraciones y administrado en base a políticas. Este a su vez provee una consola con características enriquecedoras para la administración de servicio y configuración de políticas, así como para el monitoreo de sistemas tareas operacionales. El presente bus facilita la implementación de arquitecturas de bajo nivel de acoplamiento, permeabiliza el rehúso de servicios empresariales y centraliza la administración permitiendo la convivencia del concepto de gobernanza. La consola del presente bus adicionalmente se encuentra diseñada para dar rápida respuesta de manera efectiva a todos los requerimientos que puedan surgir en ambientes orientados a servicios.

El ALSB puede realizar transportaciones de tipo mix-and-match lo que le permite garantizar entregas de información en aplicaciones end-to-end.

Los protocolos que típicamente soporta el ALSB para mensajería son:

- HTTP (S),
- SOAP,
- JMS,
- File/FTP/POP/SMTP/IMAP,

- Tuxedo,
- EJB,
- DSP y
- Native NQ

De manera adicional, se puede decir que los siguientes:

- JMS con cabeceras ,
- Emails (con o sin anexos),
- MFL (Message format Language),
- Texto,
- SOAP (con o sin anexos) y
- XML (con o sin anexos).

El presente bus permite:

- Transparentar la ubicación del servicio,
- Ruteo dinámico,
- Realizar Transformaciones,
- Orquestación de Servicios,
- Enriquecimiento de Mensajes,
- Administración de Seguridad y
- Uso de SLA (Service Level Agreements)

2.8.9. Servidor de Integración: Weblogic Server 10

Oracle WebLogic Server, es el Servidor de middleware del stack tecnológico de Oracle ya que lo adquirió en el 2008 al comprar BEA Systems. Viene distribuido de manera comercial en versiones Standard y Enterprise, diseñado esencialmente para brindar alta disponibilidad en aplicaciones y servicios, se encuentra dotado de excelentes herramientas de diagnóstico. La diferencia básica entre las versiones comerciales se sitúa en la posibilidad de usar clustering, administración múltiple de dominios, y en varias herramientas de diagnóstico que pueden o no hacerse presentes. El clustering permite el migrar entre servidores y manejo de failover entre estos, asegurando la continuidad de procesos de negocio críticos. Las herramientas de diagnóstico también referidas aquí son las que permiten detectar funciones críticas en cuestiones como el uso de memoria, permitiendo la detección temprana de las causas raíces que originan la problemática.

Al ser el servidor de middleware de Oracle, en el se despliegan la gran mayoría de componentes que conforman su stack middleware junto con soporte nativo para el despliegue de aplicaciones bajo framework Oracle, es así como el Bus de integración de Aqualogic así como todos los componentes BPM y SOA de AquaLogic deben de colocarse en WebLogic como servidor de integración.

En la actualidad WebLogic se encuentra en versión 11G, con levantamiento constante de nuevos realeses, la presente versión estudiada es aquella resultante de la fusión Oracle-Bea, denominada Versión de Integración, esta es la primera emitida por ambas empresas en conjunto y permite el uso de sus funcionalidades ya integradas con recursos Oracle.

3. CAPÍTULO: PLANIFICACIÓN (ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE PARA EL DESARROLLO DEL NÚCLEO OPERACIONAL DE PARANÁ)

3.1. Introducción

El presente es un documento que busca el identificar y definir las necesidades básicas a cubrir por parte del presente aplicativo, el núcleo operacional de la versión automatizada de Paraná. Este documento ha sido elaborado teniendo presente los principios filosóficos que el núcleo desea implementar así como también las características que los futuros constructores de las interfaces de acceso desean obtener de los componentes de bajo acoplamiento. Adicionalmente este documento se basa en los pilares de la etapa de planificación de la metodología extreme Programming.

3.1.1. Propósito

El objetivo claro del documento aquí presentado y de la etapa de planificación bajo la visión de extreme Programming es el de definir de la manera más clara y frontal, todas las funcionalidades y restricciones que el núcleo a desarrollar presentará. El presente documento se elabora con el fin de orientar al equipo de implementación así como a todos los futuros usuarios de los componentes sobre el camino y el alcance real del presente núcleo y así definir lo que se debe esperar al momento de su conclusión. Posterior a su elaboración, la presente se constituirá en la base misma sobre la cual todo el proyecto es construido y planificado, incluyendo todas sus características de relevancia tanto

de hardware como de software, para constituirse en la directriz del proceso a posteriori.

3.1.2. Alcance

El núcleo operacional es una aplicación construida bajo el modelo de referencia arquitectónica SOA y como tal sigue sus principios y estandarizaciones básicos. Adicionalmente la aplicación persigue como meta el brindarle las herramientas de procesamiento adecuadas a los desarrolladores que a posteriori empleen el presente core, para ejercer al común de las personas el controlar, medir y disminuir el consumo de recursos provenientes de la naturaleza, tanto renovables como no renovables; proveer de los medios adecuados para construir y mantener una red social que involucre a cada usuario que participe y se registre en el core; agilizar los proceso y principios de SOA con el uso de elementos de bajo acoplamiento que permitan una escalabilidad real en ambos sentidos, tanto de lado del usuario final como de los recursos y servicios que proveen de información al usuario.

Con lo anteriormente expuesto se asegura la automatización de los principales pilares dentro de Paraná ya descritos con anterioridad.

3.1.3. Definiciones, Acrónimos y Abreviaturas

- **XP.-** eXtreme Programming.
- **CRUD.-** Siglas de Create, Update y Delete.

3.1.4. Visión General

El documento a presentar consta de dos partes fundamentales, en la primera constan elementos descriptivos que buscan el proporcionar detalles genéricos sobre las principales características del presente desarrollo, en ella se detalla de manera sucinta las características funcionales del aplicativo; la segunda y apegándose al modelo metodológico adquiere una forma y visión un tanto diferente, al aplicar la relación con el cliente que en este caso presente es sumamente técnico, que es lo que la metodología requiere con la aparición y uso de las historias de usuario.

3.2. Descripción

3.2.1. Funciones Principales

Las siguientes son las funciones principales que el sistema debe reclutar para resultar de utilidad práctica para los implementadores de interfaces genéricas según reuniones mantenidas con los posibles desarrolladores de las mismas:

- Creación de usuarios por parte de las interfaces genéricas.
- Debe contener accesores a la información almacenada.
- Cuando un usuario es creado por primera vez este debe de inscribirse en un mangrove de manera obligatoria, en el caso de no registrarse se creará un mangrove con su propio nombre.

- Las interfaces tendrán el derecho de aumentar por su cuenta el valor del promedio si ellos así lo desearan y no utilizar el algoritmo de evaluación desarrollado.
- El puntaje tanto de la zona geográfica así como del grupo social se debe de realizar en base a la calificación de sus individuos miembros.
- La información que se refiera y corresponda a un tipo de consumo debe de ser accesible a todo momento para su análisis por parte de las interfaces genéricas.
- Al momento de registrarse un nuevo puntaje para el usuario los promedios tanto de su grupo social o mangrove como de su zona geográfica deberán de actualizarse.

3.2.2. Restricciones Generales

Luego de un profundo análisis de la visión y alcance del presente proyecto en etapas iniciales, se definieron los siguientes puntos como restricciones básicas y fundamentales, para su construcción y desarrollo.

- El proyecto presentado no incluirá la administración y procesos de gobernanza típicos en capas específicas del modelo, en especial al momento del intercambio de mensajes Soap en el ESB, ya que la envergadura del análisis de tiempos de respuesta mínimos administración de timeouts, niveles de conectividad, distribución de carga y excepciones y faltas a profundidad determinaría un proyecto posterior por sí solo, por tanto para estas instancias solo se considerará el ruteo en el

bus, permitiendo así en futuras etapas del proyecto anexar esas características con total facilidad.

- La forma de exposición de todas las características funcionales como estándar usarán exclusivamente SOAP como protocolo de comunicación, la implementación de colas bajo el uso de estándares diferentes como JMS y otros no será soportada en esta primera versión.

- Solo se desarrollarán las características enumeradas en las historias de usuario, ya que estas cubren de manera global las necesidades básicas de los futuros desarrolladores de interfaces.

- La actualización de cualquier punto del proceso de procesamiento y administración del core no debe involucrar en etapas posteriores del proceso cambios en los servicios expuestos y que son consumidos por las interfaces genéricas.

3.2.3. Suposiciones y Dependencias

De manera adicional a lo aquí considerado se deben de anotar la serie de suposiciones de las que parte el presente desarrollo, estas son:

Los constructores de interfaces genéricas son capaces de implementar componentes que puedan intercambiar mensajería SOAP, genérica.

Los servidores de los constructores genéricos de interfaces proporcionarán los recursos de infraestructura que se necesiten para mantener los canales de comunicaciones con sus interfaces y estas buscaran el disminuir su consumo de ancho de banda en caso de ser diseñadas para dispositivos móviles.

3.3. Requerimientos Específicos

3.3.1. Historias de Usuario

Las siguientes son las historias de usuario identificadas con cada uno de los probables futuros constructores de interfaces genéricas en reuniones sostenidas en la etapa de planificación del presente proyecto:

Historia de Usuario	
Número: 1	Nombre: Inserción de Usuarios
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: En el momento en el que la interfaz lo requiera podrá generar un nuevo usuario, el mismo que deberá ser ingresado con sus datos básicos necesarios:</p> <ul style="list-style-type: none">• Código único Identificante (Proveniente de Servicios Live)• Email• Nombres• Fecha de Inscripción• Nick• Ubicación <p>Con la opción de enviar datos complementarios:</p> <ul style="list-style-type: none">• Fecha de Nacimiento• Dirección• Clasificación• Fotografía <p>Estos datos deberán ser almacenados y el usuario deberá ser activado, en caso de que los datos ingresados sean válidos (La validación comprende en verificar la unicidad de la data como el email).</p>	

Observaciones: **En caso de abandonar el uso de servicios Live en un futuro las interfaces podrán enviar un password si se necesitase.**

Adicionalmente los datos que se ingresaron del usuario pueden ser modificados con el tiempo, las interfaces deben de tener la posibilidad real de modificarlos

Tabla 3.3.1.1 (C. U. Inserción de Usuarios)

Historia de Usuario	
Número: 2	Nombre: Identificación de Usuarios
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: Las interfaces deben de poder solicitar el identificar a los usuarios registrados en el sistema bajo diferentes condiciones de ubicación indiferentemente de la cantidad de usuarios que dichos pedidos retornen. El patrón de búsqueda deberá devolver un usuario al buscarlo por su código identificador o su email, y en listas de datos genéricas al momento de buscarlas por nombre, grupo social o ubicación.</p>	
<p>Observaciones: Las listas de usuarios que se devuelvan a las interfaces deben de usar un formato estandarizado</p>	

Tabla 3.3.1.2 (C. U. Identificación de Usuarios)

Historia de Usuario	
Número: 3	Nombre: Identificar Entorno de Usuario
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: Las interfaces deben de poder obtener data del entorno que rodea al usuario, es decir, su grupo social, invitaciones a grupos y su ubicación geográfica, a través de la data propia del usuario en caso de tenerla.</p>	
<p>Observaciones: Las interfaces pueden solicitar cualquiera de la data de entorno a través del código identificante o de cualquier código vinculado según sea necesario</p>	

Tabla 3.3.1.3 (C. U. Identificar entorno de Usuario)

Historia de Usuario	
Número: 4	Nombre: Inserción de Grupos Sociales
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: Las interfaces podrán crear en cualquier momento un grupo social (Mangrove), estos deberán de manera obligatoria estar atados a un usuario que será el creador del mismo, adicional deberá de constar con los siguientes parámetros de manera obligatoria:</p> <ul style="list-style-type: none"> • Código Identificante • Fecha de inscripción • Nombre <p>La relación con sus usuarios se fijarán de manera posterior</p>	
<p>Observaciones: El fundador puede en cualquier momento desvincularse en cualquier momento del grupo social.</p>	

Tabla 3.3.1.4 (C. U. Inserción de Grupos Sociales)

Historia de Usuario	
Número: 5	Nombre: Identificación de Grupo Social
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: La interfaz debe de poder identificar a un Grupo Social (Mangrove), de manera individual o colectiva utilizando listas genéricas para este fin, esto a través del código identificante del grupo social, del usuario miembro, y en listas de data genérica por su nombre y su promedio de calificación basado en el algoritmo.</p>	
<p>Observaciones: Las listas de grupos deberán ser devueltas a las interfaces en formatos estandarizados.</p>	

Tabla 3.3.1.5 (C. U. Identificación de Grupo Social)

Historia de Usuario	
Número: 6	Nombre: Identificación de Consumos
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 1
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: Las interfaces podrán pedir en cualquier momento el conjunto de data sobre un consumo en específico, la petición de debe de ser en base al código identificante del consumo, ya que el usuario final no debe de interferir en el proceso de trabajo con un consumo en específico sino que la interfaz debe de administrar dicha data, valores como gravedad y peso se deben de incluir en el traspaso de la data.</p>	
<p>Observaciones: El envío de la información debe de ser en formato estándar, y la respuesta de igual manera.</p> <p>La administración de los consumos no deberá de ser expuesta a través de interfaces de acceso genéricas al representar el conjunto de data de reglas de negocio.</p>	

Tabla 3.3.1.6 (C. U. Nombre: Identificación de Consumos)

Historia de Usuario	
Número: 7	Nombre: Identificación de Zona Geográfica
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 2
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Medio (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: Las interfaces genéricas deberán de poder recibir data referente a la ubicación geográfica específica, es decir, que según la distribución geográfica generada y escalable, las interfaces podrán conocer de manera específica los datos de una ciudad, país y zona delimitada por su código identificante, y listas genéricas a través del envío del nombre de cualquiera de estas. De manera adicional las interfaces podrán obtener la lista de ciudades en un país, y zonas en una ciudad, y el total de países.</p>	
<p>Observaciones: Tanto el envío como la recepción de la data debe de ser en formato estandarizado.</p> <p>La administración de las zonas geográficas no debe de ser expuestas a las interfaces genéricas dado que estas encapsulan las reglas del negocio.</p>	

Tabla 3.3.1.7 (C. U. Nombre: Identificación de Zona geográfica)

Historia de Usuario	
Número: 8	Nombre: Inserción de Consumos de un Usuario
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 2
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Alto (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: Toda interfaz genérica como parte central del proceso deben de ser capaces de calificar a un usuario, esta calificación se realiza a través del ingreso de un nuevo consumo de un usuario en específico, este dato posteriormente permitirá el recalcular del valor de la calificación de cada usuario, grupo social y zona geográfica.</p>	
<p>Observaciones: La delimitación de la zona geográfica puede variar entre el país, ciudad y zona</p>	

Tabla 3.3.1.8 (C. U. Nombre: Inserción de Consumos de un Usuario)

Historia de Usuario	
Número: 9	Nombre: Identificación Pormenorizada de Datos de Usuario
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 2
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: Las interfaces genéricas que necesiten baja carga de transmisión de data, deberán de constar salidas que les permitan obtener datos pormenorizados del usuario, estos constarán con proveedores del puntaje del usuario, su nombre entre otros.</p>	
<p>Observaciones: Para la solicitud de la data, las interfaces genéricas deben de enviar simplemente el código identificante del usuario y duplicar salidas preexistentes en otras historias de usuario según dichos clientes las necesiten</p>	

Tabla 3.3.1.9 (C. U. Nombre: Identificación Pormenorizada de Datos de Usuario)

Historia de Usuario	
Número: 10	Nombre: Identificación de Usuarios Para Clientes Livianos
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 3
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Alto (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: Los usuarios deberán de ser plenamente identificables para varios tipos de clientes livianos, dicha localización se debe realizar en base a su ubicación geográfica para la toma de datos, del listado de países, de este la obtención de ciudades y de este la obtención de usuarios es indispensable. Adicionalmente de permitir el envío en formato XML de todo el conjunto de datos de entorno y consumos relacionado con un usuario en específico o conjunto de estos</p>	
<p>Observaciones: Se deben determinar accesorios individuales para cada una de las funciones relatadas con anterioridad</p>	

Tabla 3.3.1.10 (C. U. Nombre: Identificación de Usuarios Para Clientes Livianos)

Historia de Usuario	
Número: 11	Nombre: Identificación de Consumos de Usuario
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 3
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: Cada usuario de una manera específica deberá tener determinado un conjunto de consumos básicos y adicionales de los cuales dependerá su calificación, las interfaces genéricas deben de tener la posibilidad de acceder al listado de dichos consumos para realizar sus evaluaciones respectivas</p>	
<p>Observaciones: En etapas iniciales del proyecto todos los usuarios registrarán todos los consumos registrados</p>	

Tabla 3.3.1.11 (C. U. Nombre: Identificación de Consumos de Usuario)

Historia de Usuario	
Número: 12	Nombre: Identificación de consumos por su Tipo
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 3
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Bajo (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: Las interfaces genéricas deberán de poder obtener el listado de consumos según su tipo en especial en etapas iniciales del proyecto, ya que así la identificación de la clasificación de estos, que determina su peso, será de gran ayuda al momento de trabajar con clientes livianos para el registro de la data de consumos reales dado que todos los usuarios estarán asociados a todos los consumos.</p>	
<p>Observaciones: Los tipos no deberán de ser administrables a través de las interfaces genéricas.</p>	

Tabla 3.3.1.12 (C. U. Nombre: Identificación de consumos por su Tipo)

Historia de Usuario	
Número: 13	Nombre: Identificación de Puntaje por Zonas
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 4
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Alto (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: Las interfaces genéricas podrán acceder a datos calculados sobre promedios de zonas geográficas, dado que la data necesaria para realizar este cálculo en un punto ocupará un gran volumen de almacenamiento y su transmisión resulta inviable. Las interfaces podrán acceder al promedio de un grupo de zonas o de una zona individualizada a través del envío de su identificador único de zona.</p>	
<p>Observaciones: En etapas iniciales del presente proyecto el único grupo de zonas al que podrán acceder las interfaces genéricas es todos, es decir el resumen de todas las zonas.</p>	

Tabla 3.3.1.13 (C. U. Nombre: Identificación de Puntaje por Zonas)

Historia de Usuario	
Número: 14	Nombre: Identificación de Puntaje de Zona Por Usuario
Usuario: Autor	
Modificación de Historia Número:	Iteración Asignada: 4
Prioridad en Negocio: Alta (Alta / Media / Baja)	Puntos Estimados:
Riesgo en Desarrollo: Alto (Alto / Medio / Bajo)	Puntos Reales:
<p>Descripción: Las interfaces genéricas deberán de poder identificar la calificación del grupo al que pertenece un usuario individualizado, para efectos de presentación como dato de entorno adicional y especial al ser este también un valor calculado. Las interfaces deberán de enviar el identificador único del usuario</p>	
<p>Observaciones: Este dato debe de estar disponible para cualquier simulador visual e incluido en cualquier salida estándar para datos calculados</p>	

Tabla 3.3.1.14 (C. U. Nombre: Identificación de Puntaje de Zona Por Usuario)

lógica del negocio. A continuación se relata la manera y consideraciones al respecto:

En el plan de entregas presentado con antelación constan los tiempos de entrega de las historias de usuario 1, 2, 3, 4, 5 y 6 estas conforman la primera iteración, las mismas que en conjunto tomaron alrededor de 15 días laborables para su construcción.

La segunda iteración comprende las historias 7, 8, y 9, todas estas se encuentran altamente compenetradas y contienen un nivel de complejidad elevado, éstas tomaron en total 12 días laborables en su implementación.

La tercera iteración que también involucra la implementación de lógica de negocio esencial considerando las observaciones anteriores, e involucra las historias de usuario 10, 11, y 12 con un total de 11 días en su elaboración.

La cuarta y última iteración se planificó bajo las premisas anteriormente citadas, es decir agrupándolas por la complejidad que las embarga y su correlación, este último punto es el elemento principal que orilló a utilizar como recurso adicional el generarlas 2 historias de usuario, 13, y 14 en una nueva iteración, su tiempo total se registro en 10 días laborables.

3.6. Iteraciones

3.6.1. Iteración 1

3.6.1.1. Pruebas de Aceptación

Número: PA001	Historia de Usuario: 1
Descripción: Existen funciones para el ingreso de nuevos usuarios a partir de sus datos individuales, por lo tanto se desea medir su correcto desempeño	
Condiciones de Ejecución: <ol style="list-style-type: none">1. Eliminar la Data Previa en la Base de Datos2. Construir una interfaz genérica para el ingreso de la Data	
Entrada: <ol style="list-style-type: none">1. Ingresar los siguientes usuarios<ul style="list-style-type: none">• Abdelrahman Aly (001)• Santiago Pesantez (002)• Adrián Cevallos (003)	
Resultado Esperado: <ol style="list-style-type: none">1. No debe de retornarse ninguna falta ni excepción2. Deben de aparecer tres registros con los nombres mencionados en la base de datos	
Evaluación de la Prueba: <ol style="list-style-type: none">1. No se levantó excepción ni falta alguna2. Los tres registros aparecieron en la base de datos	

Tabla 3.6.1.1.1 (P.A. PA001)

Número: PA002	Historia de Usuario: 2
<p>Descripción: La data almacenada del usuario se encuentra expuesta de manera genérica para proporcionar listas y usuarios individuales según corresponda.</p>	
<p>Condiciones de Ejecución:</p> <ol style="list-style-type: none"> 1. Eliminar la Data Previa en la Base de Datos 2. Construir una interfaz genérica para el ingreso de la Data 	
<p>Entrada:</p> <ol style="list-style-type: none"> 1. Ingresar los siguientes usuarios <ul style="list-style-type: none"> • Abdelrahman Aly (001) • Santiago Pesantez (002) • Adrián Cevallos (003) • Santiago Alvear (004) • Adrián Pesántez (005) • Soledad Erazo (006) 2. Solicitar el Usuario de Código 002 3. Solicitar el Usuario de Nombre Santiago 	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. No debe de retornarse ninguna falta ni excepción 2. Debe retornar a Abdelrahman Aly 3. Debe Retornar una lista compuesta por dos usuarios, 002 y 004 	
<p>Evaluación de la Prueba:</p> <ol style="list-style-type: none"> 1. No se levantó excepción ni falta alguna 2. Se mostró el usuario Abderlahaman Aly 3. Se mostró la lista de 2 usuarios 	

Tabla 3.6.1.1.2 (P.A. PA002)

Número: PA003	Historia de Usuario: 3
<p>Descripción: Al proporcionar acceso a la información de un usuario, se debe de mostrar la data de su zona geográfica y grupo social de manera separada. Con la consideración de que cada usuario al momento de crearse genera un grupo social con su nombre.</p>	
<p>Condiciones de Ejecución:</p> <ol style="list-style-type: none"> 1. Eliminar la Data Previa en la Base de Datos 2. Construir una interfaz genérica para el ingreso de la Data 	
<p>Entrada:</p> <ol style="list-style-type: none"> 1. Ingresar los siguientes usuarios <ul style="list-style-type: none"> • Abdelrahman Aly (001) • Santiago Pesantez (002) • Adrián Cevallos (003) 2. Ingresar Lista de Países <ul style="list-style-type: none"> • Colombia • Perú • Ecuador 3. Ingresar Lista de Ciudades Relacionadas Con los países que correspondan: <ul style="list-style-type: none"> • Bogotá • Quito • Lima 4. Ingresar una zona por ciudad y país vinculado: <ul style="list-style-type: none"> • Zona1 (001) • Zona2 (002) • Zona3 (003) 5. Relacionar a Abdelrahman Aly con la Zona1 6. Relacionar a Santiago Pesantez con la Zona2 7. Relacionar a Adrián Cevallos con la Zona3 8. Buscar el grupo social de Abdelrahman Aly 9. Buscar el grupo social de Santiago Pesantez 10. Buscar el grupo social de Adrián Cevallos 11. Buscar la zona geográfica de Abdelrahman Aly 12. Buscar la zona geográfica de Santiago Pesantez 13. Buscar la zona geográfica de Adrián Cevallos 	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. Retorno del grupo social de Abdelrahman Aly (Abdelrahman Aly) 2. Retorno del grupo social de Santiago Pesantez (Santiago Pesantez) 3. Retorno del grupo social de Adrián Cevallos (Adrián Cevallos) 4. Retorno de la zona geográfica de Abdelrahman Aly (Zona1) 5. Retorno de la zona geográfica de Santiago Pesantez (Zona2) 6. Retorno de la zona geográfica de Adrián Cevallos (Zona3) 	
<p>Evaluación de la Prueba:</p>	

<ol style="list-style-type: none"> 1. No se levantó excepción ni falta alguna 2. Retornó el grupo social de Abdelrahaman Aly (Abdelrahaman Aly) 3. Retornó el grupo social de Santiago Pesantez (Santiago Pesantez) 4. Retornó el grupo social de Adrián Cevallos (Adrián Cevallos) 5. Retornó la zona geográfica de Abdelrahaman Aly (Zona1) 6. Retornó la zona geográfica de Santiago Pesantez (Zona2) 7. Retornó la zona geográfica de Adrián Cevallos (Zona3)
--

Tabla 3.6.1.1.3 (P.A. PA003)

Número: PA004	Historia de Usuario: 4
<p>Descripción: Existen funciones para el ingreso de nuevos grupos sociales a partir de sus datos individuales, por lo tanto se desea medir su correcto desempeño</p>	
<p>Condiciones de Ejecución:</p> <ol style="list-style-type: none"> 1. Eliminar la Data Previa en la Base de Datos 2. Construir una interfaz genérica para el ingreso de la Data 	
<p>Entrada:</p> <ol style="list-style-type: none"> 1. Ingresar el Siguiete Grupo Social: <ul style="list-style-type: none"> • Mangrove01 	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. No debe de retornarse ninguna falta ni excepción 2. Debe de aparecer un Mangrove con el nombre Mangrove01 3. Debe de aparecer una invitación al creador del grupo 4. Debe de aparecer el autor relacionado con Mangrove01 	
<p>Evaluación de la Prueba:</p> <ol style="list-style-type: none"> 1. No se levantó excepción ni falta alguna 2. Aparece un Mangrove con el nombre Mangrove01 3. Aparece una invitación al creador del grupo 4. Aparece el autor relacionado con Mangrove01 	

Tabla 3.6.1.1.4 (P.A. PA004)

Número: PA005	Historia de Usuario: 5
<p>Descripción: La data almacenada del grupo social se encuentra expuesta de manera genérica para proporcionar listas grupos sociales individuales según corresponda.</p>	
<p>Condiciones de Ejecución:</p> <ol style="list-style-type: none"> 1. Eliminar la Data Previa en la Base de Datos 2. Construir una interfaz genérica para el ingreso de la Data 	
<p>Entrada:</p> <ol style="list-style-type: none"> 1. Ingresar los siguientes Grupos sociales <ul style="list-style-type: none"> • Mangrove01 (001) • Mangrove02 (002) • Mangrove03 (003) • Mangrove01 (004) 2. Solicitar el grupo social 001 3. Solicitar el grupo social de nombre Mangrove01 	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. No debe de retornarse ninguna falta ni excepción 2. Debe retornar un único registro Mangrove01 3. Debe Retornar una lista compuesta por dos grupos sociales, 001 y 004 	
<p>Evaluación de la Prueba:</p> <ol style="list-style-type: none"> 1. No se levantó excepción ni falta alguna 2. Se mostró el grupo social Mangrove01 3. Se mostró la lista de 2 grupos sociales 	

Tabla 3.6.1.1.5 (P.A. PA005)

Número: PA006	Historia de Usuario: 6
<p>Descripción: Existen funciones para la exposición de consumos e identificar un consumo, una porción o todos.</p>	
<p>Condiciones de Ejecución:</p> <ol style="list-style-type: none"> 1. Eliminar la Data Previa en la Base de Datos 3. Construir una interfaz genérica para el ingreso de la Data 	
<p>Entrada:</p> <ol style="list-style-type: none"> 1. Ingresar los siguientes consumos de manera manual y clasificarlos en un grupo 1: <ul style="list-style-type: none"> • Electricidad (001) • Agua (002) 2. Ingresar los siguientes consumos de manera manual y clasificarlos en un grupo 2: <ul style="list-style-type: none"> • Gas (003) • Gasolina (004) 3. Solicitar el consumo de código 001 4. Solicitar los consumos del grupo 2 5. Solicitar todos los consumos 	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. No debe de retornarse ninguna falta ni excepción 3. Debe de retornar un objeto de tipo consumo con Electricidad (001) 4. Debe de retornar una lista con los consumos del grupo 2 que son: <ul style="list-style-type: none"> • Gas (003) • Gasolina (004) 5. Debe retornar una lista con los consumos del grupo 1 y 2: <ul style="list-style-type: none"> • Electricidad (001) • Agua (002) • Gas (003) • Gasolina (004) 	
<p>Evaluación de la Prueba:</p> <ol style="list-style-type: none"> 1. No se levantó excepción ni falta alguna 2. Retornó el objeto determinado 3. Retornó la lista de consumos determinada 4. Retorno los 4 consumos 	

Tabla 3.6.1.1.6 (P.A. PA006)

3.6.1.2. Incidencias

Durante la ejecución de las pruebas de aceptación no se registraron incidencias que ameriten una corrección en el diseño de la presente solución, todo lo contrario los resultados indican un comportamiento satisfactorio una vez terminadas las pruebas de la presente iteración.

3.6.2. Iteración 2

3.6.2.1. Pruebas de Aceptación

Número: PA007	Historia de Usuario: 7
Descripción: La data almacenada de la zona geográfica se encuentra expuesta de manera genérica para proporcionar listas de países, ciudades por país, y zonas por ciudad	
Condiciones de Ejecución: <ol style="list-style-type: none">1. Eliminar la Data Previa en la Base de Datos2. Construir una interfaz genérica para el ingreso de la Data	
Entrada: <ol style="list-style-type: none">1. Ingresar Lista de Países<ul style="list-style-type: none">• Colombia (001)• Ecuador (002)• Perú (003)2. Ingresar Lista de Ciudades Relacionadas Con los países que correspondan:<ul style="list-style-type: none">• Bogotá (0001)• Quito (0002)• Guayaquil (0003)• Sangolquí (0004)• Lima (0005)3. Ingresar una zona por ciudad y país vinculado:<ul style="list-style-type: none">• Zona1 (001)• Zona2 (002)• Zona3 (003)• Zona4 (004)• Zona5 (005)4. Solicitarla zona de código identificante 0015. Solicitar la lista de ciudades pertenecientes al país de código identificante 0026. Solicitar la lista del total de países7. Solicitar la lista de zonas de la ciudad de código	

(0004)
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. No debe de retornarse ninguna falta ni excepción 2. Debe de retornar la Zona001 con la data relacionada a Bogotá y Colombia 3. Debe retornar el total de ciudades que pertenecen al Ecuador, estas son: <ul style="list-style-type: none"> • Quito • Guayaquil • Sangolquí 4. Debe de retornar el total de todos los países ingresados es decir: <ul style="list-style-type: none"> • Colombia • Ecuador • Perú 5. Debe retornar la lista de zonas de la ciudad de Sangolquí compuesta de un único objeto
<p>Evaluación de la Prueba:</p> <ol style="list-style-type: none"> 1. No retornó ninguna falta ni excepción 2. Retornó la Zona001 con la data relacionada a Bogotá y Colombia 3. Retornó el total de ciudades que pertenecen al Ecuador, estas son: <ul style="list-style-type: none"> • Quito • Guayaquil • Sangolquí 4. Retornó el total de todos los países ingresados es decir: <ul style="list-style-type: none"> • Colombia • Ecuador • Perú 5. Retornó la lista de zonas de la ciudad de Sangolquí compuesta de un único objeto

Tabla 3.6.1.1.7 (P.A. PA007)

Número: PA008	Historia de Usuario: 8
<p>Descripción: Existen funciones dedicadas a permitir el ingreso de consumos de un determinado usuario, estas debe de registrar dicha información de manera coherente</p>	
<p>Condiciones de Ejecución:</p> <ol style="list-style-type: none"> 1. Eliminar la Data Previa en la Base de Datos 2. Construir una interfaz genérica para el ingreso de la Data 	
<p>Entrada:</p> <ol style="list-style-type: none"> 1. Ingresar los siguientes consumos de manera manual y clasificarlos en un grupo 1: <ul style="list-style-type: none"> • Electricidad (001) • Agua (002) 2. Ingresar los siguientes consumos de manera manual y clasificarlos en un grupo 2: <ul style="list-style-type: none"> • Gas (003) • Gasolina (004) 3. Ingresar un usuario de nombre Abdelrahman Aly (001) 4. Ingresar un consumo de grupo 2 (004) de usuario 001 	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. No debe de retornarse ninguna falta ni excepción <ul style="list-style-type: none"> • Debe de Registrar un consumo para el usuario 001 de gasolina 	
<p>Evaluación de la Prueba:</p> <ol style="list-style-type: none"> 1. No se levantó excepción ni falta alguna 2. Se registró un consumo del usuario 001 en el consumo 001 	

Tabla 3.6.1.1.8 (P.A. PA008)

Número: PA009	Historia de Usuario: 9
<p>Descripción: Existen funciones para el ingreso de nuevos usuarios a partir de sus datos individuales, por lo tanto se desea medir su correcto desempeño</p>	
<p>Condiciones de Ejecución:</p> <ol style="list-style-type: none"> 1. Eliminar la Data Previa en la Base de Datos 2. Construir una interfaz genérica para el ingreso de la Data 	
<p>Entrada:</p> <ol style="list-style-type: none"> 1. Ingresar los siguientes usuarios <ul style="list-style-type: none"> • Abdelrahman Aly (001) • Santiago Pesantez (002) • Adrián Cevallos (003) 2. Solicitar el nombre de usuario 001 3. Solicitar puntaje usuario 003 	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. No debe de retornarse ninguna falta ni excepción 2. Debe de retornar Abdelrahman Aly 3. Debe de retornar 0 	
<ol style="list-style-type: none"> 1. No retornó ninguna falta ni excepción 2. Retornó Abdelrahman Aly 3. Retornó 0 	

Tabla 3.6.1.1.9 (P.A. PA009)

3.6.2.2. Incidencias

En la presente iteración y luego del respectivo análisis del comportamiento durante las pruebas de aceptación, el comportamiento abstracto de los siguientes elementos fue modificado:

- El promedio en caso de no tener data sobre puntaje se redefine a 0
- Se encontraron inconvenientes a la hora de generar la vinculación automática de un usuario con un mangrove de su nombre lo mismo que fue corregido para que la historia pudiese pasar la prueba de aceptación.

- El análisis profundo sobre el último caso se pudo realizar a fondo gracias a la utilización de la opción del envío detallado de data seleccionado randómicamente ya que existían funciones adicionales que realizaban tareas similares.

Luego de atender las presentes incidencias se procedió a la siguiente iteración.

3.6.3. Iteración 3

3.6.3.1. Pruebas de Aceptación

Número: PA010	Historia de Usuario: 10
Descripción: La data almacenada de la zona geográfica se encuentra relacionada al usuario y debe de ser accesible relacionándola con el usuario, adicional se debe de generar un documento XML que contenga la información de consumos de un usuario en una zona geográfica determinada	
Condiciones de Ejecución:	
<ol style="list-style-type: none"> 1. Eliminar la Data Previa en la Base de Datos 2. Construir una interfaz genérica para el ingreso de la Data 	
Entrada:	
<ol style="list-style-type: none"> 1. Ingresar Lista de Países <ul style="list-style-type: none"> • Colombia (001) • Ecuador (002) • Perú (003) 2. Ingresar Lista de Ciudades Relacionadas Con los países que correspondan: <ul style="list-style-type: none"> • Bogotá (0001) • Quito (0002) • Guayaquil (0003) • Sangolquí (0004) • Lima (0005) 3. Ingresar una zona por ciudad y país vinculado: <ul style="list-style-type: none"> • Zona1 (001) • Zona2 (002) • Zona3 (003) • Zona4 (004) • Zona5 (005) 4. Ingresar los siguientes usuarios <ul style="list-style-type: none"> • Abdelrahman Aly (001) • Santiago Pesantez (002) • Adrián Cevallos (003) 5. Vincular a usuario 001 con la zona 001 	

<ol style="list-style-type: none"> 6. Vincular a usuario 002 con la zona 002 7. Vincular a usuario 003 con la zona 003 8. Solicitar lista de usuarios por ciudad 0001 9. Solicitar la lista de consumos a registrar para la lista de usuarios obtenida
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. No debe de retornarse ninguna falta ni excepción 2. Debe de retornar a usuario 001 3. Debe de retornar un stream XML con todos los consumos de tipo registrables del usuario 001, en etapas iniciales son todos los existentes
<p>Evaluación de la Prueba:</p> <ol style="list-style-type: none"> 1. No retornó ninguna falta ni excepción 2. Retornó al usuario 001 3. Retornó una XML en el formato esperado

Tabla 3.6.1.1.10 (P.A. PA010)

Número: PA011	Historia de Usuario: 11
<p>Descripción: Existen salidas genéricas para obtener el listado de consumos por usuario, el cual en etapas iniciales deberá devolver el total de consumos</p>	
<p>Condiciones de Ejecución:</p> <ol style="list-style-type: none"> 1. Eliminar la Data Previa en la Base de Datos 2. Construir una interfaz genérica para el ingreso de la Data 	
<p>Entrada:</p> <ol style="list-style-type: none"> 1. Ingresar los siguientes consumos de manera manual y clasificarlos en un grupo 1: <ul style="list-style-type: none"> • Electricidad (001) • Agua (002) 2. Ingresar los siguientes consumos de manera manual y clasificarlos en un grupo 2: <ul style="list-style-type: none"> • Gas (003) • Gasolina (004) 3. Ingresar un usuario de nombre Abdelrahman Aly (001) 4. Solicitar la lista de consumos de usuario 001 	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. Debe devolver el total de consumos registrados 	
<p>Evaluación de la Prueba:</p> <ol style="list-style-type: none"> 1. No se levantó excepción ni falta alguna 2. Devolvió el total de consumos registrados 	

Tabla 3.6.1.1.11 (P.A. PA011)

Número: PA012	Historia de Usuario: 12
<p>Descripción: Existen salidas genéricas para obtener el listado de consumos esta vez por su tipo y clasificación</p>	
<p>Condiciones de Ejecución:</p> <ol style="list-style-type: none"> 1. Eliminar la Data Previa en la Base de Datos 2. Construir una interfaz genérica para el ingreso de la Data 	
<p>Entrada:</p> <ol style="list-style-type: none"> 1. Ingresar los siguientes consumos de manera manual y clasificarlos en un grupo 1: <ul style="list-style-type: none"> • Electricidad (001) • Agua (002) 2. Ingresar los siguientes consumos de manera manual y clasificarlos en un grupo 2: <ul style="list-style-type: none"> • Gas (003) • Gasolina (004) 3. Consultar los consumos de grupo 2 	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. Debe devolver el total de consumos registrados en el grupo 2, estos son: <ul style="list-style-type: none"> • Gas (003) • Gasolina (004) 	
<p>Evaluación de la Prueba:</p> <ol style="list-style-type: none"> 1. No se levantó excepción ni falta alguna 2. Devolvió el total de consumos registrados en el grupo 2 es decir: <ul style="list-style-type: none"> • Gas (003) • Gasolina (004) 	

Tabla 3.6.1.1.12 (P.A. PA012)

3.6.3.2. Incidencias

La presente iteración durante su proceso de evaluación mediante el uso de la herramienta de pruebas de aceptación detecto las siguientes incidencias:

- Las pruebas sobre accesos dispares a grupos y clasificaciones de consumos por usuarios al tratarse y encontrarse diseñado para soportar a todos los usuarios dentro de todos los consumos retorno los resultados esperados pero no se pueden realizar verificaciones posteriores para cotejar dichos resultados con aquellos que se desearán en etapas posteriores del proyecto.
- El documento XML de salida que fue probado en las pasadas pruebas de aceptación tuvo que ser corregido en varias ocasiones hasta ser emitido en un formato estandarizado y comprensible esto según los propios interesados en la construcción de las interfaces genéricas, es por tal virtud que se aprobó la salida en XML.

Luego de atender las presentes incidencias se procedió a la siguiente iteración.

3.6.4. Iteración 4

3.6.4.1. Pruebas de Aceptación

Número: PA013	Historia de Usuario: 13
Descripción: El puntaje de la zona es definido en función de los usuarios que pertenecen a la misma, del promedio de cada uno de ellos.	
Condiciones de Ejecución: <ol style="list-style-type: none">1. Eliminar la Data Previa en la Base de Datos2. Construir una interfaz genérica para el ingreso de la Data	
Entrada: <ol style="list-style-type: none">1. Ingresar la siguiente Lista de Países:<ul style="list-style-type: none">• Colombia (001)• Ecuador (002)• Perú (003)2. Ingresar Lista de Ciudades Relacionadas Con los países que correspondan:<ul style="list-style-type: none">• Bogotá (0001)• Quito (0002)• Guayaquil (0003)• Sangolquí (0004)• Lima (0005)3. Ingresar una zona por ciudad y país vinculado:<ul style="list-style-type: none">• Zona1 (001)• Zona2 (002)• Zona3 (003)• Zona4 (004)• Zona5 (005)4. Ingresar los siguientes usuarios<ul style="list-style-type: none">• Abdelrahman Aly (001) con puntaje 100• Santiago Pesantez (002) con puntaje 50• Adrián Cevallos (003) con puntaje 305. Vincular a usuario 001 con la zona 0016. Vincular a usuario 002 con la zona 0017. Vincular a usuario 003 con la zona 0038. Solicitar el puntaje de la zona geográfica 0019. Solicitar el puntaje de todas las zonas geográficas	
Resultado Esperado: <ol style="list-style-type: none">1. No debe de retornarse ninguna falta ni excepción2. Debe de retornar 75 puntos3. Debe de retornar 60 puntos	
Evaluación de la Prueba: <ol style="list-style-type: none">1. No retornó ninguna falta ni excepción2. Retornó 75 puntos3. Retornó 60 puntos	

Tabla 3.6.1.1.13 (P.A. PA013)

Número: PA014	Historia de Usuario: 14
<p>Descripción: Los grupos sociales también almacenan un puntaje este puede ser obtenido a partir del código identificante que le pertenece al usuario.</p>	
<p>Condiciones de Ejecución:</p> <ol style="list-style-type: none"> 1. Eliminar la Data Previa en la Base de Datos 2. Construir una interfaz genérica para el ingreso de la Data 	
<p>Entrada:</p> <ol style="list-style-type: none"> 1. Ingresar los siguientes Grupos sociales <ul style="list-style-type: none"> • Mangrove01 (001) • Mangrove02 (002) • Mangrove03 (003) • Mangrove01 (004) 2. Ingresar los siguientes usuarios <ul style="list-style-type: none"> • Abdelrahman Aly (001) con puntaje 100 • Santiago Pesantez (002) con puntaje 50 • Adrián Cevallos (003) con puntaje 30 3. Vincular a usuarios 001 y 002 con Mangrove01 4. Vincular usuario 003 a Mangrove02 5. Solicitar el puntaje del grupo social del usuario 001 6. Solicitar el puntaje del grupo social del usuario 002 7. Solicitar el puntaje del grupo social del usuario 003 	
<p>Resultado Esperado:</p> <ol style="list-style-type: none"> 1. No debe de retornarse ninguna falta ni excepción 2. Debe retornar 75 puntos 3. Debe retornar 75 puntos 4. Debe retornar 30 puntos 	
<p>Evaluación de la Prueba:</p> <ol style="list-style-type: none"> 1. No se levantó excepción ni falta alguna 2. Retornó 75 puntos 3. Retornó 75 puntos 4. Retornó 30 puntos 	

Tabla 3.6.1.1.14 (P.A. PA014)

3.6.4.2. Incidencias

Las siguientes son las notas observatorias, resultado de las pruebas de aceptación que se llevaron a cabo sobre el presente aplicativo en la última iteración registrada:

- Los presentes cálculos que serán utilizados para simulaciones de ambiente deberán ser ampliados de manera necesaria en etapas posteriores al presente proyecto ya que en la actualidad es imposible el definir más allá de lo que hasta ahora se ha hecho las futuras perspectivas de simulación por parte de los constructores de interfaces genéricas.
- El cálculo para los valores se constato a mano resultando ser similar al aplicar el algoritmo de calificación en etapas iniciales.

3.7. Rotaciones

Las rotaciones resultan en un elemento de relevancia al hablar de eXtreme programming y persiguen como objetivo como ya se ha revisado anteriormente el refrescar los procesos de implementación y sobre todo permitir a los desarrolladores el mantener un conocimiento globalizado de la aplicación.

Al tratarse de un proyecto elaborado por un solo recurso para su desarrollo, esta tarea no pudo ser llevada a cabo, pero su espíritu se mantuvo a lo largo de su desarrollo.

3.8. Reuniones

A lo largo del proceso de desarrollo se mantuvieron constantes reuniones informales de planificación y análisis con los futuros usuarios del núcleo así como con usuarios finales de las interfaces genéricas que podrían desarrollarse para establecer lineamientos generales.

De manera adicional, se llevaron a cabo reuniones semanales y en épocas específicas del proyecto, durante el proceso de desarrollo, éstas fueron diarias con los posibles usuarios del core para definir niveles de importancia y prioridades en las tareas a realizarse.

4. CAPÍTULO: DISEÑO DEL NÚCLEO OPERACIONAL DE PARANÁ

4.1. Metáfora del Sistema

La metáfora del sistema considerada es la que a continuación se anota:

El núcleo de procesamiento de Paraná es un contenedor de lógica de negocios capaz de proveer de manera estandarizada información preprocesada, calculada o accesada, a interfaces genéricas diseñadas para embeber la funcionalidad que les brinda la serie de servicios que exponen el conjunto de características necesarias detectadas para automatizar las características principales de la filosofía.

4.2. DISEÑO ARQUITECTÓNICO DE LA APLICACIÓN

4.2.1. Arquitectura Lógica

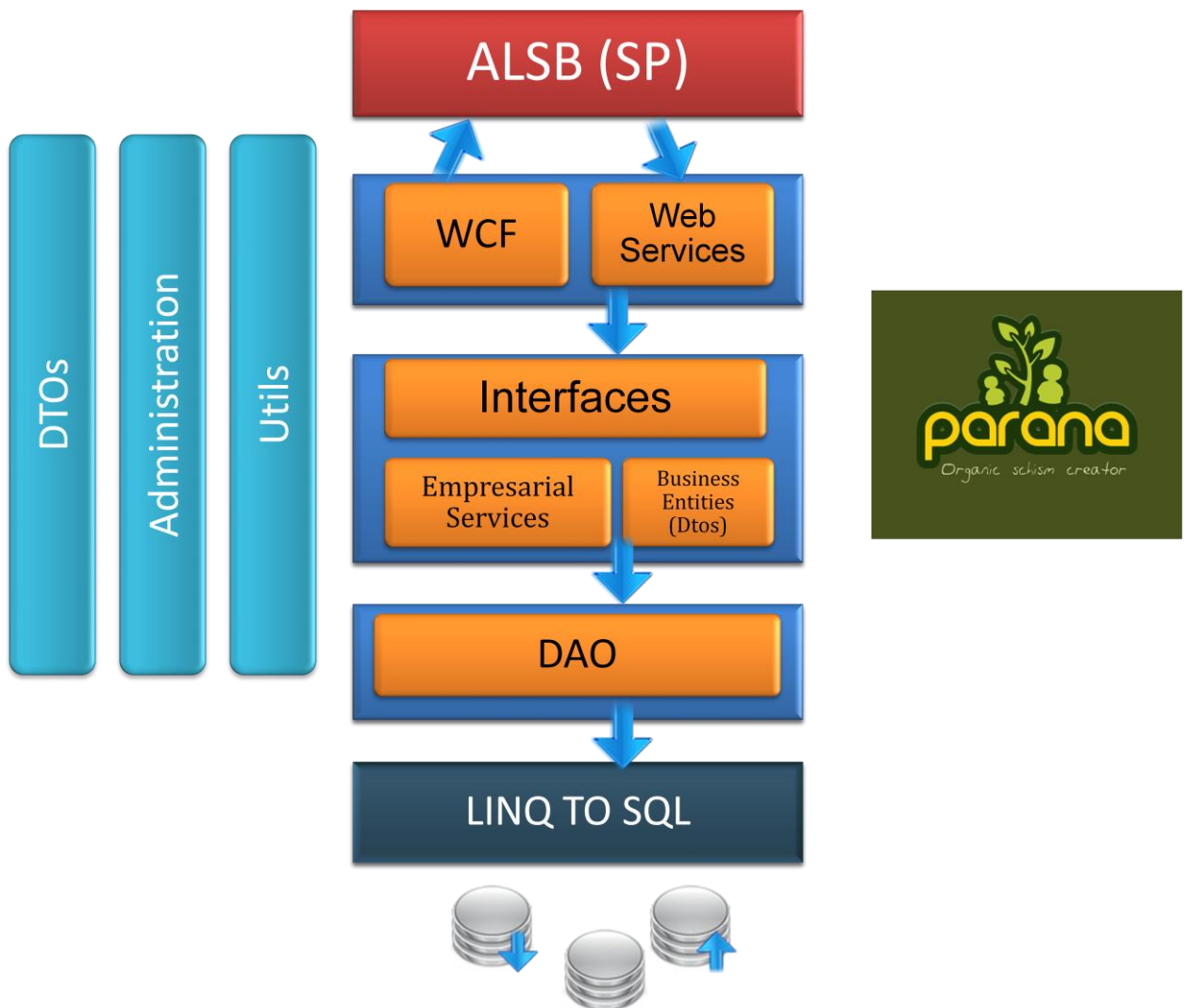


Ilustración 4.2.1.1 (Arquitectura Lógica de la Aplicación)

4.2.2. Arquitectura Física

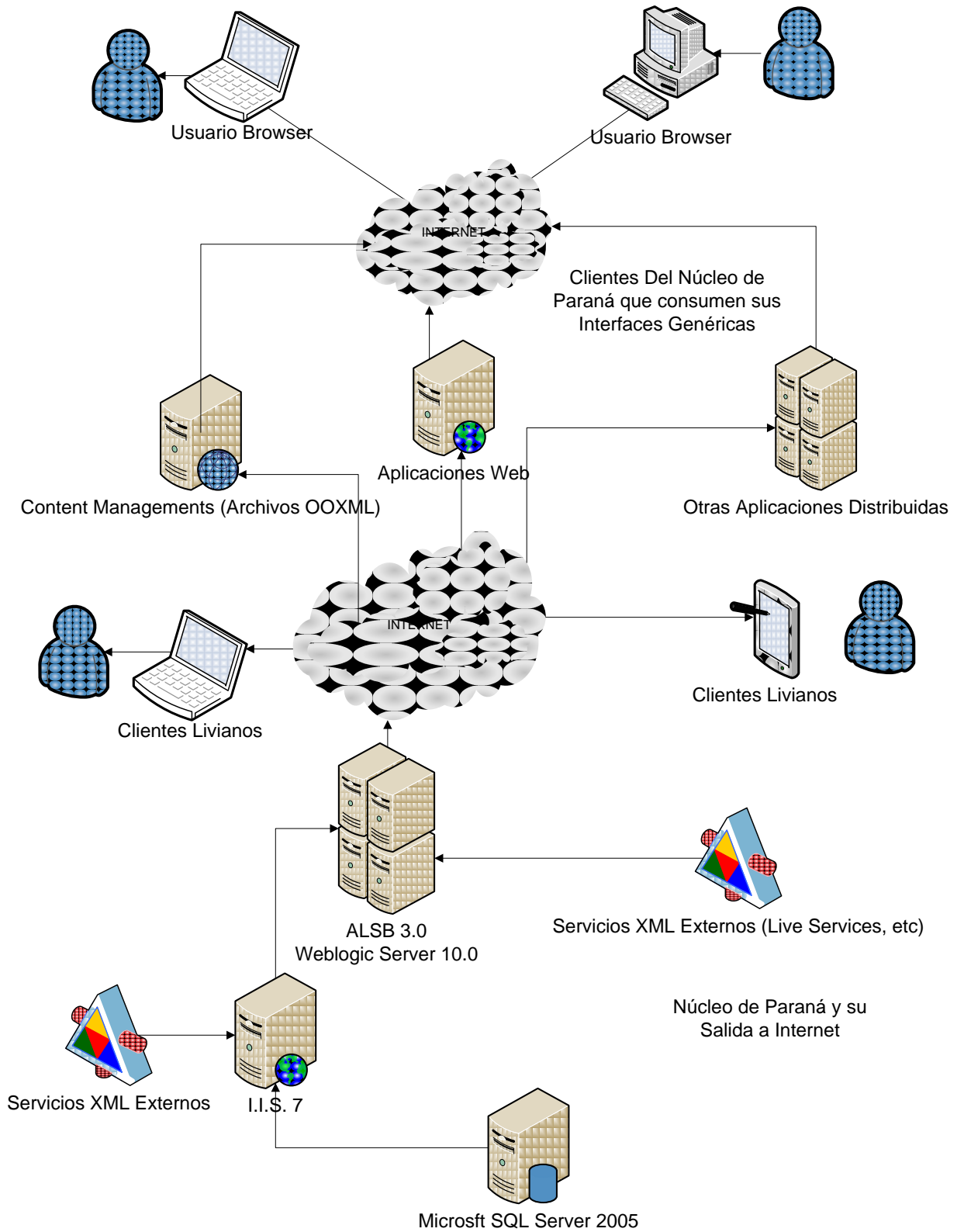


Ilustración 4.2.2.1 (Diagrama de Arquitectura Física)

4.3. Tarjetas CRC

Las siguientes son el total de tarjetas CRC definidas para la presente aplicación y que se convierten en las contenedoras de la lógica de negocio, las mismas se desarrollaron en las reuniones que se llevaron a cabo con los posibles usuarios y se describen a continuación:

CityManagerImpl	
Super Clases: CityManager	
Sub Clases: Ninguna	
Descripción: Contiene la funcionalidad base para la automatización de toda acción relacionada con la administración de las ciudades dentro del núcleo	
Atributos:	
Nombre	Descripción
cityDao	Accesor a la data de la ciudad, con el se trabajara para realizar tareas Crud
Responsabilidades:	
Nombre	Colaborador
Listar la ciudad por su país	CityDao
Encontrar la ciudad por su código identificante	CityDao
Listar Ciudades por su nombre	CityDao

Tabla 4.3.1 (C.R.C. CityManagerImpl)

ConsumptionManagerImpl	
Super Clases: ConsumptionManager	
Sub Clases: Ninguna	
Descripción: Contiene la funcionalidad base para la automatización de toda acción relacionada con la administración de los consumos dentro del núcleo	
Atributos:	
Nombre	Descripción
consumptionDao	Accesor a la data del consumo, con el se trabajara para realizar tareas Crud
Responsabilidades:	
Nombre	Colaborador
Obtener la data de un consumo específico	ConsumptionDao
Obtener los consumos por su tipo	ConsumptionDao

Tabla 4.3.2 (C.R.C. ConsumptionManagerImpl)

ConsumptionRegistrationManagerImpl	
Super Clases: ConsumptionRegistrationManager	
Sub Clases: Ninguna	
Descripción: Contiene la funcionalidad base para la automatización de toda acción relacionada con el ingreso y registro de todo consumo ocasionado por el usuario y recalculation de valores	
Atributos:	
Nombre	Descripción
userConsumptionDao	Accesor a la data del consumo de un usuario determinado, con el se trabajara para realizar tareas Crud
consumptionDao	Accesor a la data de un consumo, con el se trabajara para realizar tareas Crud
mangroveDao	Accesor a la data de un grupo social determinado, con el se trabajara para realizar tareas Crud
userDao	Accesor a la data de un usuario determinado, con el se trabajara para realizar tareas Crud
invitationDao	Accesor a la data de las invitaciones a grupos sociales de los usuarios, con el se trabajara para realizar tareas Crud
zoneDao	Accesor a la data de una zona determinada, con el se trabajara para realizar tareas Crud
Responsabilidades:	
Nombre	Colaborador
Insertar un consumo de un usuario determinado con datos predefinidos	UserConsumptionDao
Insertar un consumo de un usuario en detalle recalculando su calificación	UserConsumptionDao, InvitationDao, UserDao, MangroveDao, ZoneDao
Calcular la variación de una calificación.	Ninguna
Obtener el impacto de una zona determinada	ZoneDao

Tabla 4.3.3 (C.R.C. ConsumptionRegistrationManagerImpl)

CountryManagerImpl	
Super Clases: CountryManager	
Sub Clases: Ninguna	
Descripción: Contiene la funcionalidad base para la automatización de toda acción relacionada con la administración de los países dentro del núcleo	
Atributos:	
Nombre	Descripción
countryDao	Accesor a la data del país, con el se trabajara para realizar tareas Crud
Responsabilidades:	
Nombre	Colaborador
Listar el país por su nombre	CountryDao
Encontrar el país por su código identificante	CountryDao
Listar todos los países	CountryDao

Tabla 4.3.4 (C.R.C. CountryManagerImpl)

InscriptionProcessManagerImpl	
Super Clases: InscriptionProcessManager	
Sub Clases: Ninguna	
Descripción: Contiene la funcionalidad base para la automatización de toda acción relacionada con el proceso de inscripción de usuarios dentro del núcleo	
Atributos:	
Nombre	Descripción
userDao	Accesor a la data del usuario, con el se trabajara para realizar tareas Crud
mangroveDao	Accesor a la data del mangrove, con el se trabajara para realizar tareas Crud
zoneDao	Accesor a la data de la zona, con el se trabajara para realizar tareas Crud
countryDao	Accesor a la data del país, con el se trabajara para realizar tareas Crud
invitationDao	Accesor a la data de la invitación de un usuario, con el se trabajara para realizar tareas Crud
Responsabilidades:	
Nombre	Colaborador
Crear un nuevo usuario	UserDao, MangroveDao, ZoneDao, CountryDao, InvitationDao

Tabla 4.3.5 (C.R.C. InscriptionProcessManagerImpl)

MangroveCreationManagerImpl	
Super Clases: MangroveCreationManager	
Sub Clases: Ninguna	
Descripción: Contiene la funcionalidad base para la automatización de toda acción relacionada con el proceso de inscripción de nuevos mangroves dentro del núcleo	
Atributos:	
Nombre	Descripción
mangroveDao	Accesor a la data del mangrove, con el se trabajara para realizar tareas Crud
invitationDao	Accesor a la data de la invitación de un usuario, con el se trabajara para realizar tareas Crud
Responsabilidades:	
Nombre	Colaborador
Crear un nuevo grupo social	MangroveDao, InvitationDao

Tabla 4.3.6 (C.R.C. MangroveCreationManagerImpl)

MangroveManagerImpl	
Super Clases: MangroveManager	
Sub Clases: Ninguna	
Descripción: Contiene la funcionalidad base para la automatización de toda acción relacionada con la administración de los grupos sociales dentro del núcleo	
Atributos:	
Nombre	Descripción
mangroveDao	Accesor a la data del grupo social (Mangrove), con el se trabajara para realizar tareas Crud
Responsabilidades:	
Nombre	Colaborador
Obtener la información de un grupo social en específico	MangroveDao
Listar grupos sociales filtrados por su nombre	MangroveDao
Listar todos los grupos sociales por su puntaje	MangroveDao

Tabla 4.3.7 (C.R.C. MangroveManagerImpl)

MobileOperationManagerImpl	
Super Clases: MobileOperationManager	
Sub Clases: Ninguna	
Descripción: Contiene la funcionalidad base para la automatización de toda acción relacionada con la comunicación con dispositivos móviles y clientes livianos dentro del núcleo	
Atributos:	
Nombre	Descripción
userDao	Accesor a la data del usuario, con el se trabajara para realizar tareas Crud
cityDao	Accesor a la data de la ciudad, con el se trabajara para realizar tareas Crud
consumptionDao	Accesor a la data del consumo, con el se trabajara para realizar tareas Crud
countryDao	Accesor a la data del país, con el se trabajara para realizar tareas Crud
userConsumptionDao	Accesor a la data de los consumos de un usuario, con el se trabajara para realizar tareas Crud
consumptionRegistrationManager	Encargado del procesamiento de la lógica de negocio relacionada con el registro de consumos del usuario
Responsabilidades:	
Nombre	Colaborador
Listar los usuarios de una ciudad	UserDao
Listar todos los países	CountryDao
Listar todas las ciudades de un país	CityDao
Fijar los consumos ingresados remotamente en formato XML bajo un XSD específico	ConsumptionRegistrationManagerImpl, UserConsumptionDao, ConsumptionDao
Listar los consumos de un usuario	ConsumptionDao, UserConsumptionDao

Tabla 4.3.8 (C.R.C. MobileOperationManagerImpl)

ProfileManagerImpl	
Super Clases: ProfileProcessManager	
Sub Clases: Ninguna	
Descripción: Contiene la funcionalidad base para la automatización de toda acción relacionada con el proceso de administración del perfil de un usuario dentro del núcleo	
Atributos:	
Nombre	Descripción
userDao	Accesor a la data del usuario, con el se trabajara para realizar tareas Crud
mangroveDao	Accesor a la data del mangrove, con el se trabajara para realizar tareas Crud
zoneDao	Accesor a la data de la zona, con el se trabajara para realizar tareas Crud
invitationDao	Accesor a la data de la invitación de un usuario, con el se trabajara para realizar tareas Crud
Responsabilidades:	
Nombre	Colaborador
Actualizar la data de un usuario en específico, incluyendo su grupo social y zona actual	UserDao
Listar el total de usuarios registrados en un grupo social	UserDao, MangroveDao
Encontrar el total de usuarios que comparten el mismo nombre	UserDao
Crear un nuevo grupo social	MangroveDao
Obtener la data de una zona en específico	ZoneDao
Obtener la data del grupo social de un usuario en específico	UserDao, InvitationDao, MangroveDao
Obtener la data de un usuario en específico	UserDao

Tabla 4.3.9 (C.R.C. ProfileManagerImpl)

SimulationManagerImpl	
Super Clases: SimulationProcessManager	
Sub Clases: Ninguna	
Descripción: Contiene la funcionalidad base para la automatización de toda acción relacionada con todo proceso de simulación y obtención de data calculada dentro del núcleo	
Atributos:	
Nombre	Descripción
zoneDao	Accesor a la data de la zona, con el se trabajara para realizar tareas Crud
Responsabilidades:	
Nombre	Colaborador
Obtener todos los puntos de una zona	ZoneDao
Obtener el promedio de todas las zonas	ZoneDao
Obtener el promedio de una zona a través de la identificación de un usuario vinculado a esta	ZoneDao

Tabla 4.3.10 (C.R.C. SimulationManagerImpl)

4.4. Soluciones Puntuales

La presente arquitectura así como el diseño en conjunción contiene una serie de soluciones específicas a puntos trascendentales para el buen trabajo del conjunto de la aplicación, entre ellas y las que se dispusieron son las siguientes:

- Usar el patrón de facade para la exposición final de servicios que interactúen directamente con el bus lo que facilitará la construcción de servicios proxy de bypass.
- Utilizar el patrón Manager con interfaces para encapsular la lógica de negocio en las clases implementadas, así se subdivide mejor forma la aplicación y se le da mayor integridad en la consulta y acceso a cada funcionalidad del core en específico.
- Emplear LINQ como lenguaje de acceso a data de objetos y no solo para el acceso a la data relacional, así la disminución del volumen de las líneas de código es sustancial sin aumentar necesariamente la complejidad ciclomática.
- Dado que las técnicas designadas para hacer el mapeo de la data relacional a data de objetos almacena todo cambio real sobre los ya mencionados objetos para reflejarlos en la base de datos, se debe de implementar objetos DTO (Data Transfer Objects) para transformar toda la data que la aplicación requiera entre sus diferentes capas.
- Utilizar puertos de salida predefinidos por el Bus y tanto el IIS 7 como cualquier servidor dedicado concebido con el fin de albergar servicios WCF.

- Utilizar endpoint bajo HTTPBasic en la configuración de cada WCF de salida ya que es recomendación Microsoft el usarlos para mejorar la confiabilidad en el envío de paquetería SOAP.
- Crear un servidor dedica para así no depender exclusivamente de IIS versión 7 y que sea en realidad sumamente portable el core aquí presentado.

4.5. Funcionalidad Mínima

La funcionalidad mínima deseada es aquella que se indica en la lista de requerimientos mencionados adicionándose la correcta interacción comprobada con la lógica y ruteo del ESB que se dispuso como herramienta de exhibición final de los servicios de proxy.

5. CAPÍTULO: IMPLEMENTACIÓN DEL NÚCLEO OPERACIONAL DE PARANÁ

5.1. Programación en Parejas

La programación en parejas es sin dudas uno de los pilares de la presente metodología empleada y persigue como claro objetivo el no permitir que los desarrolladores cometan errores livianos tanto en el diseño como en la implementación de microsoluciones a problemas y adversidades encontradas en el proceso de desarrollo de la aplicación.

Lastimosamente al ser la presente desarrollada de manera integral por un solo desarrollador, el proceso de programación en parejas no se pudo aplicar, pero su espíritu y sus objetivos se buscaron y obtuvieron a través de la revisión y refactorización continua permitiendo así la reevaluación de aquellos elementos construidos y diseñados con anterioridad en etapas de diseño, mejorándolos inclusive de manera continua.

5.2. Estándares de Codificación

La siguiente es la matriz detallada del conjunto de estándares utilizados para la construcción del núcleo, todos se siguieron a detalle y se encuentran clasificados en dos grupos, el primero contiene la especificación de estándares de nombres de objetos, la segunda el conjunto de nombres y estándares empleados para el desarrollo dentro de dichos objetos como se podrá observar a continuación:

Area	Standard	Observación	Ejemplo
Namespace	Se debe de emplear notación Pascal, no se debe de usar subguión. La raíz deberá ser el dominio de la aplicación. Si existen acrónimos en el nombre y estos tienen 3 letras o más se deberá de usar notación de pascal.	Facilita la lectura del código al ser la manera en la que la librería de clases del Framework de .Net nombra a sus nameSpaces	Org.Parana.BusinessEntities
Assemblies	En lo posible un assembly debe de contener un solo nameSpace, y deberá de llevar el mismo nombre que el nameSpace	Permite además de la rápida lectura y su acepción con las convenciones de .Net, el poder de manera eficaz identificar el contenido del assembly	Org.Parana.BusinessEntities.dll
Clases	Se debe de emplear notación Pascal, no se debe de usar subguión. No iniciarán con un señalador de clase. No iniciarán con I a menos que la siguiente letra se encuentre en minúscula. Las clases no deberán tener el mismo nombre que el namespace que las contiene. No habrá abreviaciones en nombres y siempre se usaran sustantivos.	Facilita la lectura del código al ser la manera en la que la librería de clases del Framework de .Net nombra a sus clases	CityDto, MangroveDao, TreeDto
Interfaces	Seguir los estándares fijados para las clases, con el condicionante de colocar una I al inicio para identificarla	Esencialmente permite el separar las clases de las interfaces para identificar plenamente tipos de datos disímiles	ICityManager, IConsumptionManager

Tabla 5.2 (M.E. archivos)

La siguiente tabla contiene especificaciones formales sobre el código propiamente dicho:

Area	Standard	Observación	Ejemplo
Métodos	Se debe de emplear notación Pascal, no se debe de usar subguión. Consideración si existe un handler. Evitar abreviaciones. Usar verbos en los nombres	Facilita la lectura del código al ser la manera en la que la librería de clases del Framework de .Net nombra a sus métodos	<code>public void CreateNewUser(UserDto userDto)</code>
Propiedades	Se debe de emplear notación Pascal, no se debe de usar subguión. No se deben usar abreviaturas, sus miembros deben de comenzar con minúscula	Facilita la lectura del código al ser la manera en la que la librería de clases del Framework de .Net nombra a sus propiedades	<code>public string IdCity</code>
Parámetros	Se debe de emplear notación Pascal, no se debe de usar subguión. No se deben usar abreviaturas, deben de comenzar con minúscula	Facilita la lectura del código al ser la manera en la que la librería de clases del Framework de .Net nombra a sus parámetros	<code>ref string idCity</code>
Variables en Métodos	Se debe de emplear notación Pascal, no se debe de usar subguión. No se deben usar abreviaturas, deben de comenzar con minúscula. Usar sustantivos.	Evita la colisión de nombres entre atributos de las clases y variables usadas en sus métodos	<code>string idCtiyAux;</code>
Constantes	Escribirlas íntegramente en mayúsculas separándolas con subguión. Se deben de usar nombres claramente descriptivos	Aunque este estándar contradice lo que típicamente señalan los libros de estándares de .Net la práctica cotidiana ha demostrado mayor utilidad en expresarlas de esta forma	<code>public const string INVITATION_ACTIVE_STATUS="A";</code>

Tabla 5.2 (M.E. clases)

5.3. Modelo de Datos

Modelo Lógico

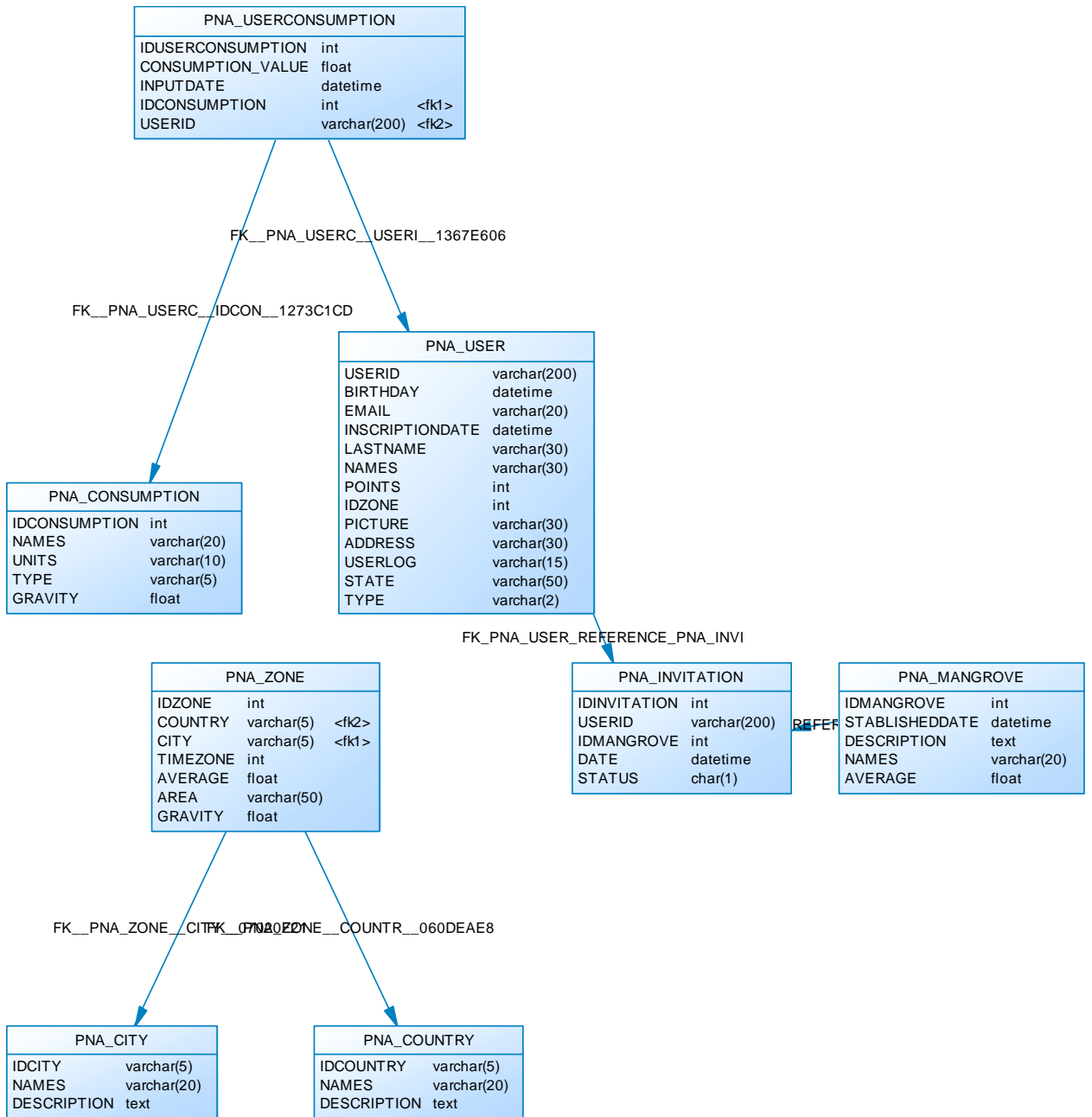


Ilustración 5.3.1 (Modelo Lógico)

Modelo SQL Server

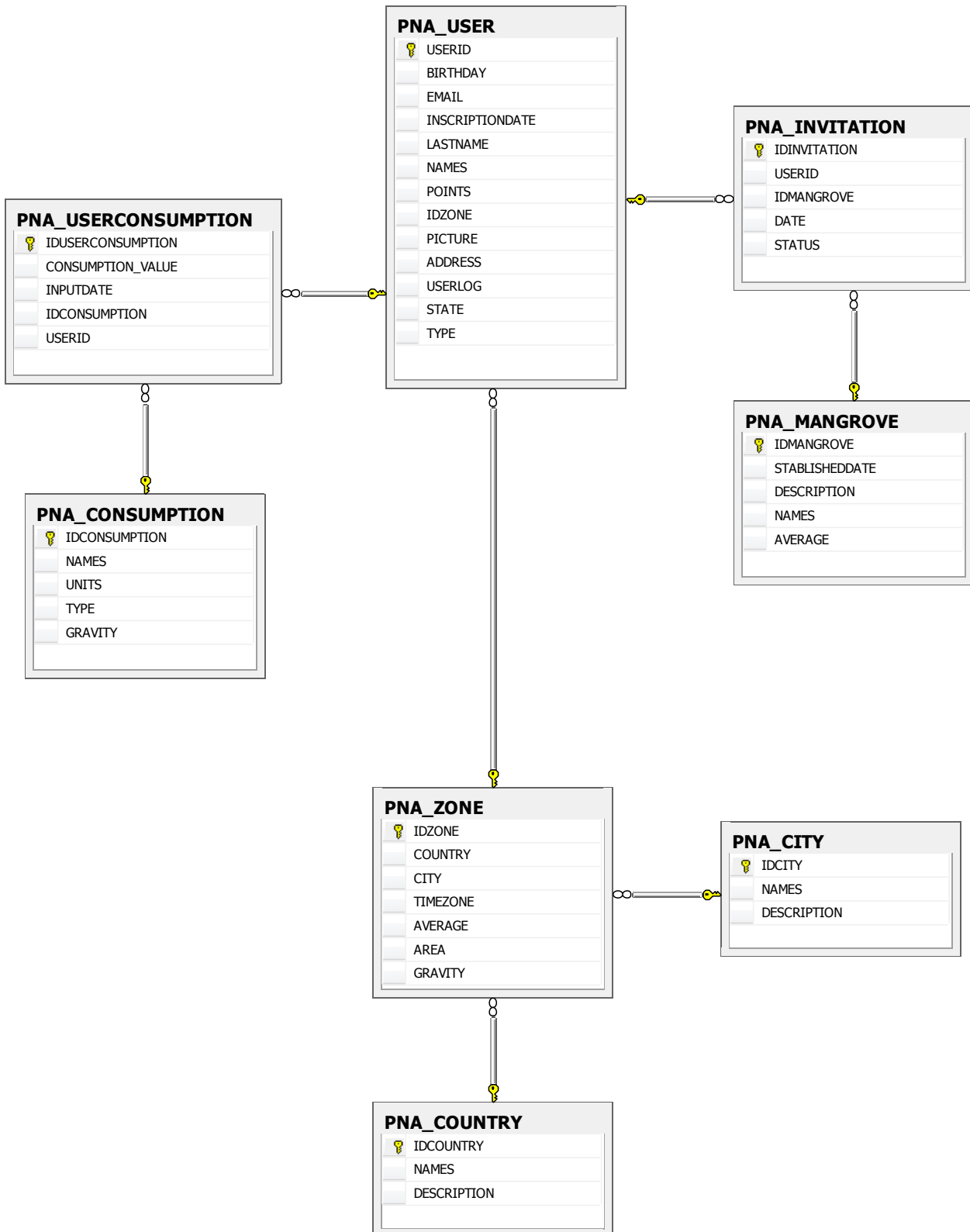
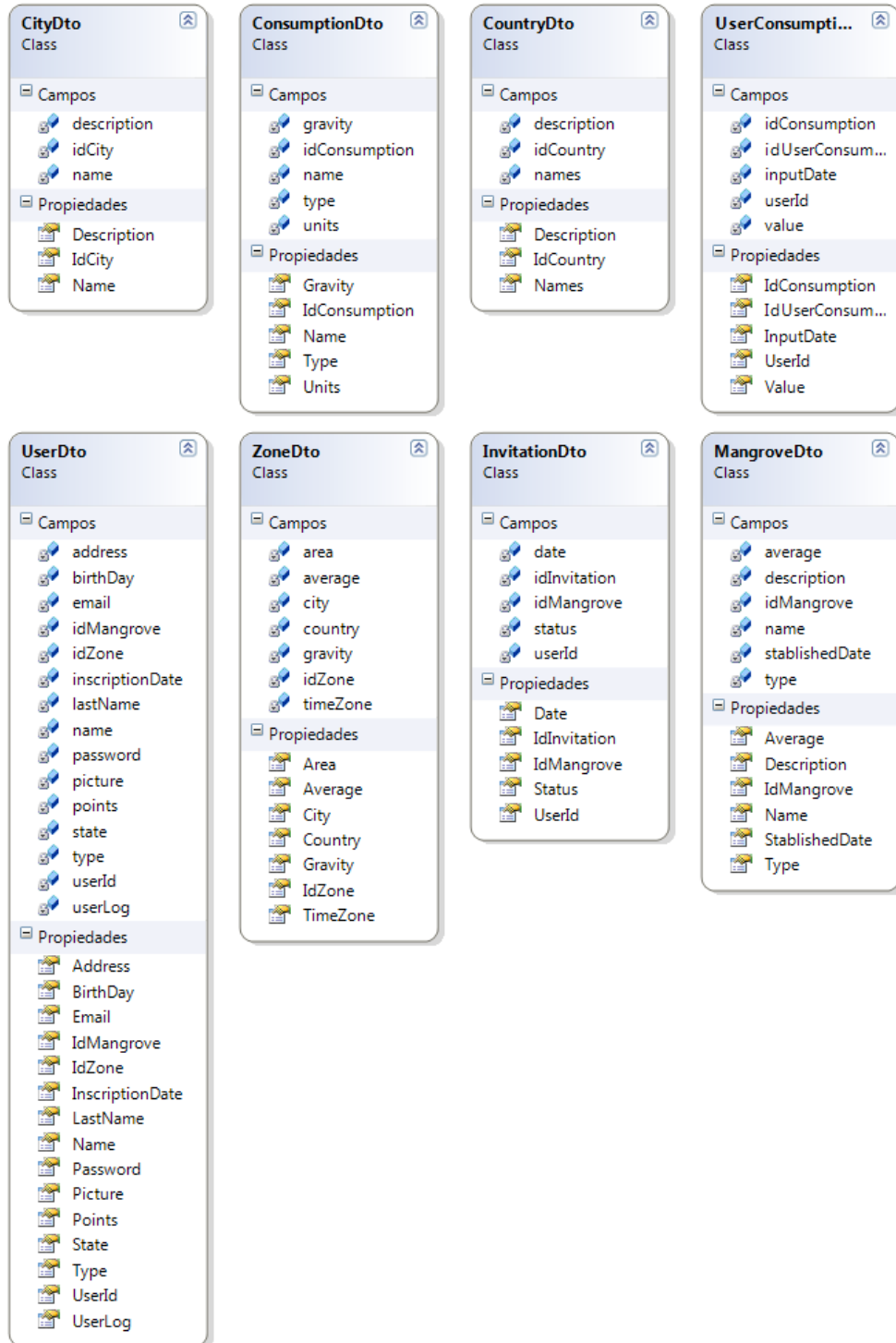


Ilustración 5.3.2 (Modelo SQL Server)

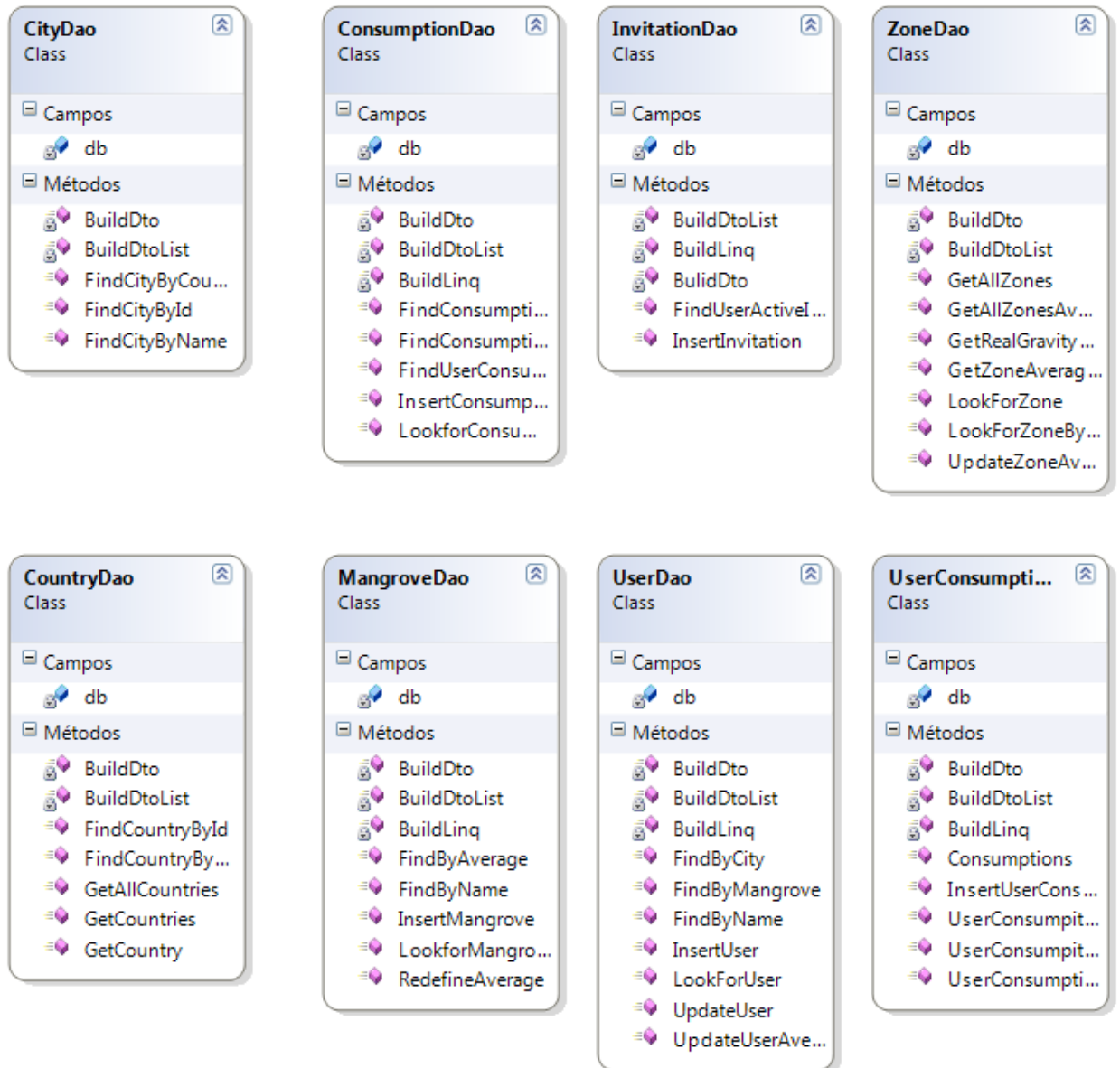
5.4. Modelo de Clases

ParanaSoft.Parana.BusinessEntities



Ilstración 5.4.1 (M.C. ParanaSoft.Parana.BusinessEntities)

ParanaSoft.Parana.DataAccessObjects



Ilstración 5.4.2 (M.C. ParanaSoft.Parana.DataAccessObjects)

ParanaSoft.Parana.Manager

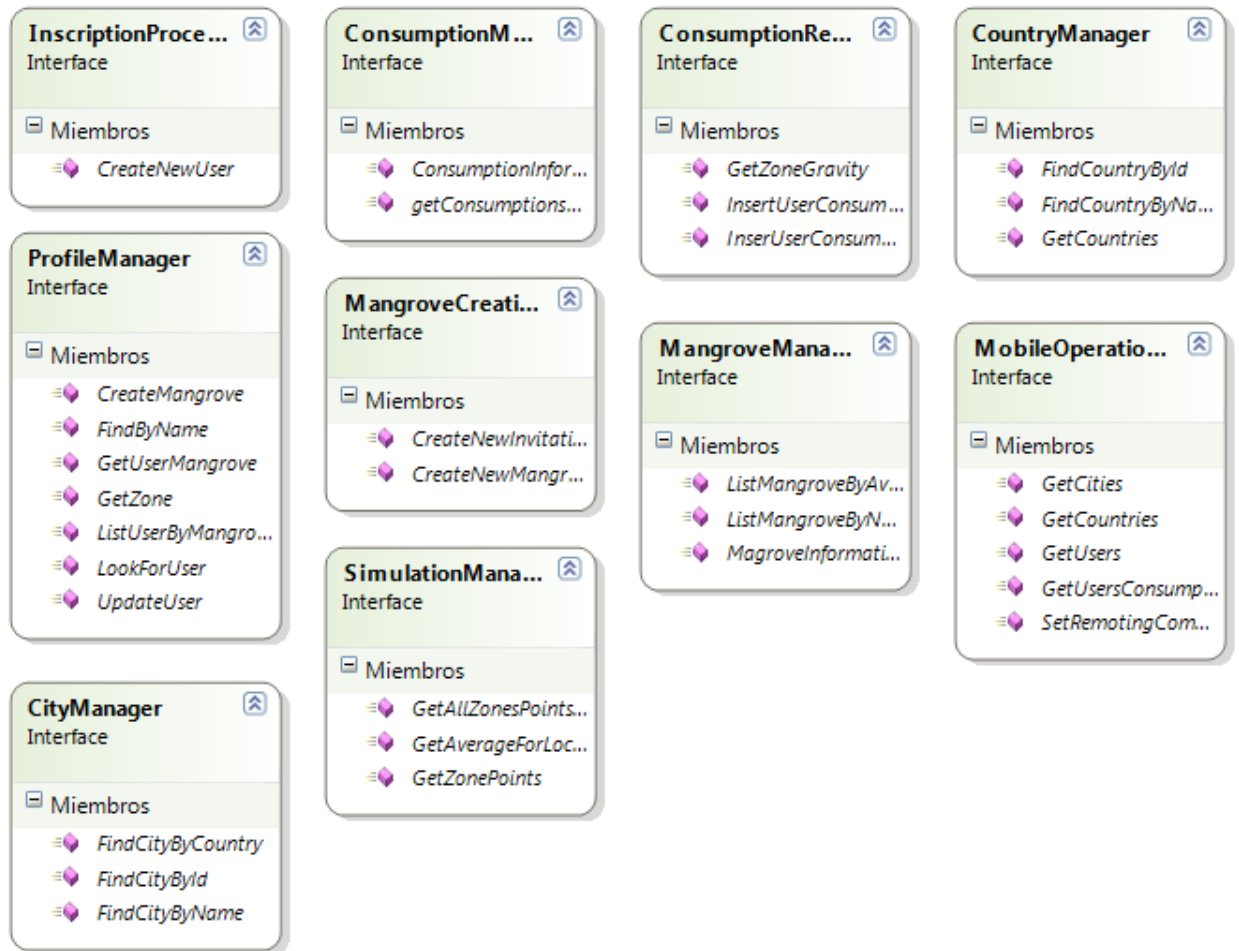
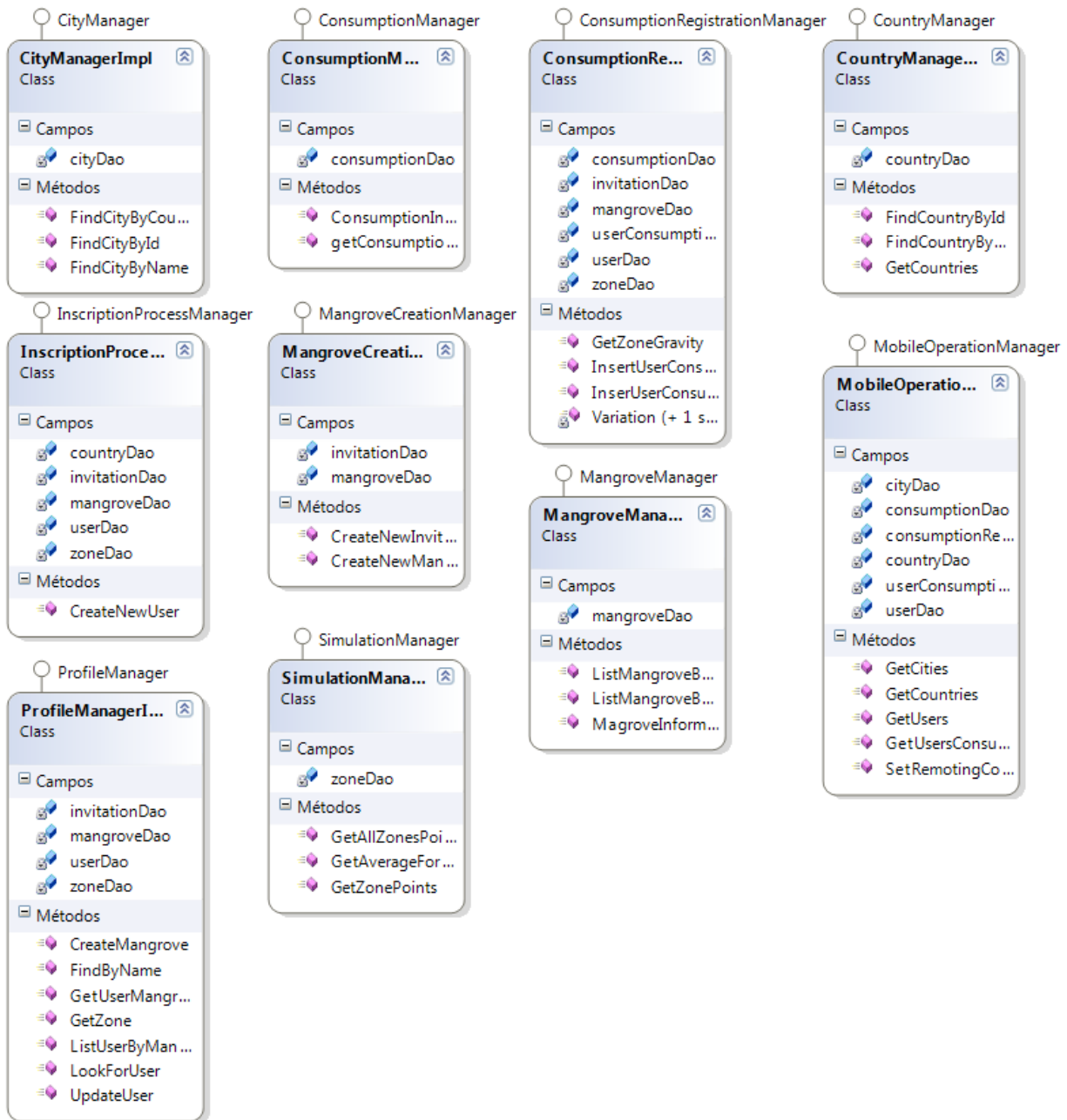


Ilustración 5.4.3 (M.C. ParanaSoft.Parana.Manager)

ParanaSoft.Parana.ManagerImplementation



Ilstración 5.4.3 (M.C. ParanaSoft.Parana.ManagerImplementation)

ParanaSoft.Parana.Utils y ParanaSoft.Parana.Server

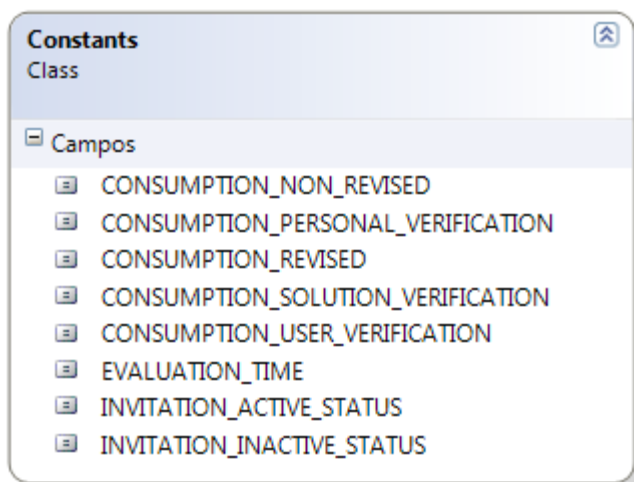
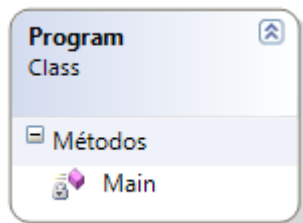


Ilustración 5.4.3 (M.C. ParanaSoft.Parana.Utils y ParanaSoft.Parana.Server)

ParanaSoft.Parana.WCFServices

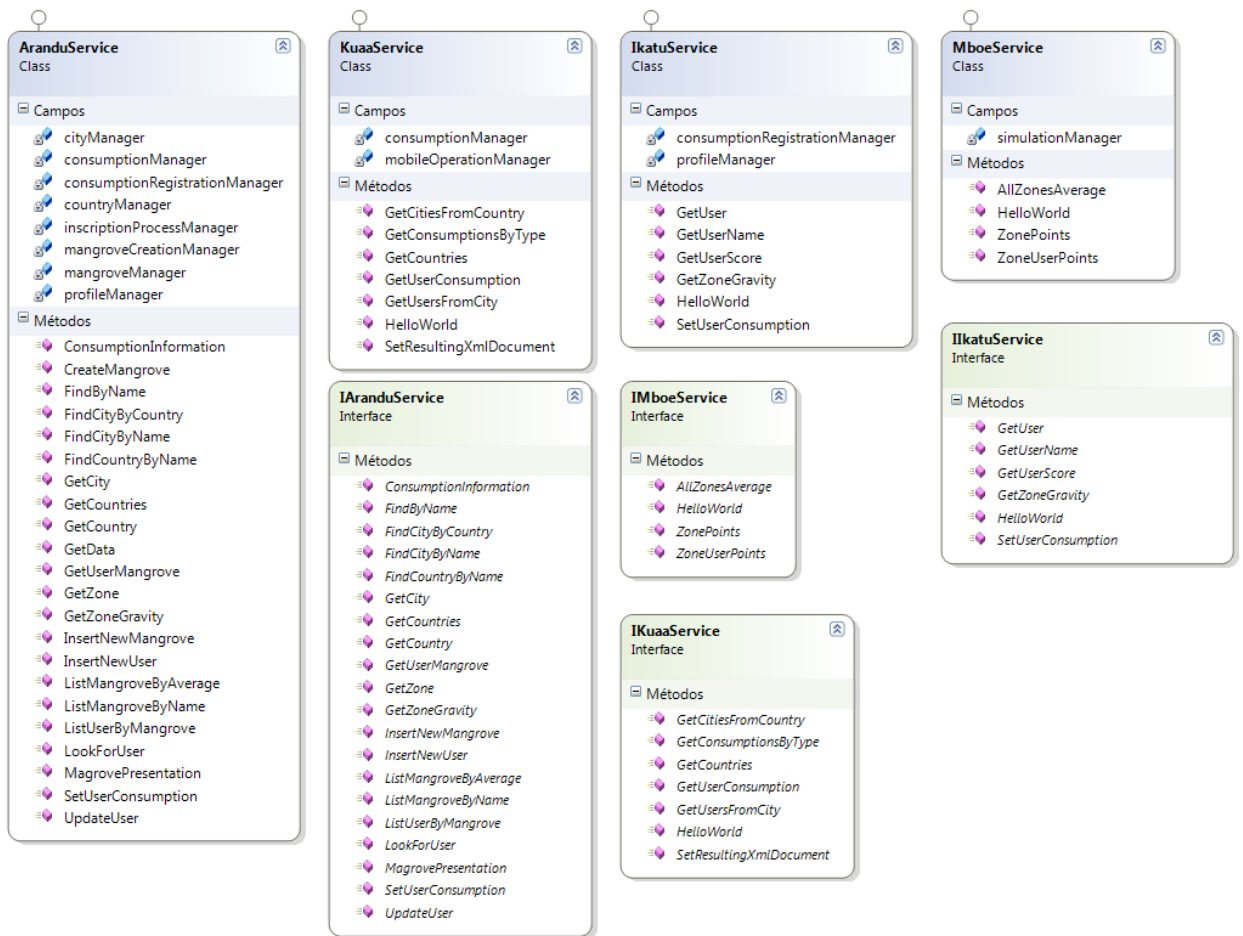
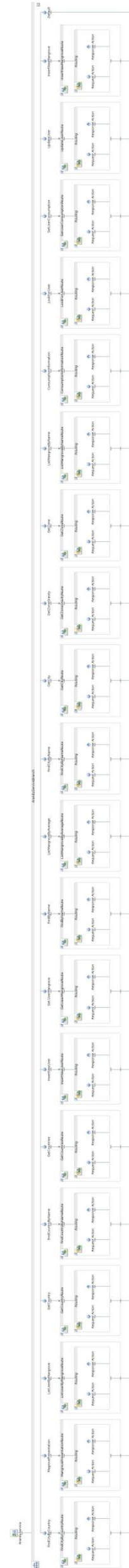


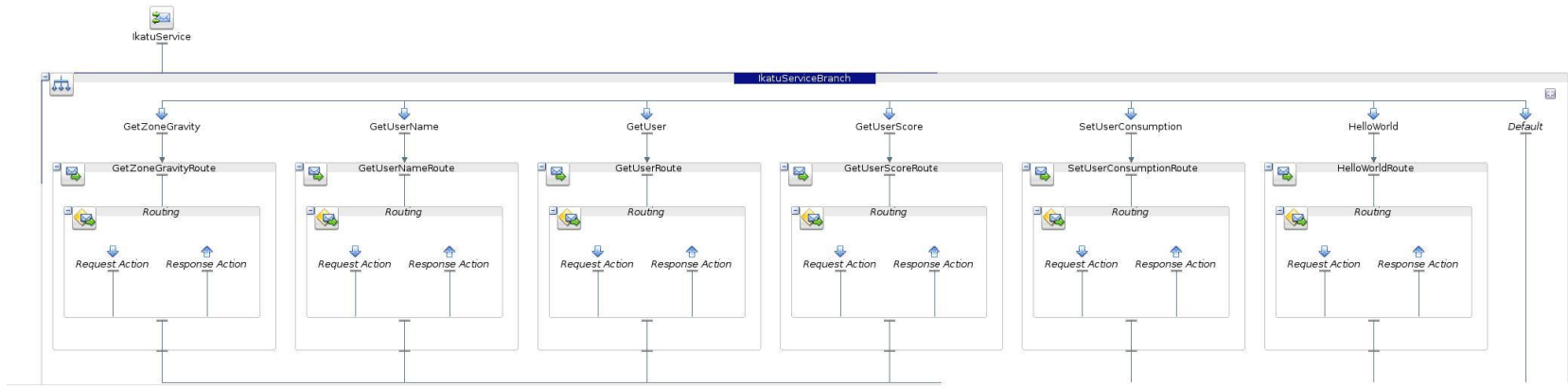
Ilustración 5.4.3 (M.C. ParanaSoft.Parana.WCFServices)

5.5. Generación de Proxy Services

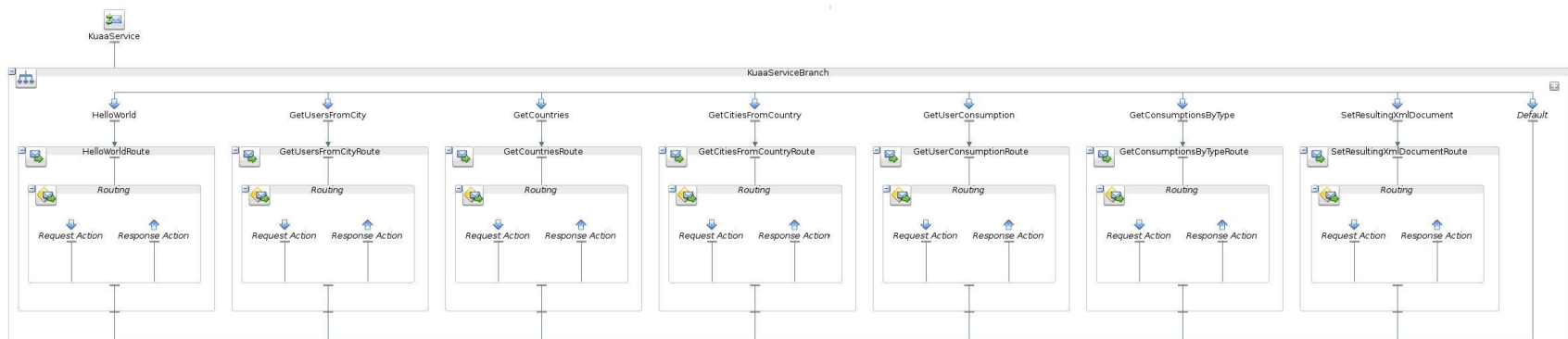
AranduServiceProxy



IkatuServiceProxy



KuaaServiceProxy



MboeServiceProxy

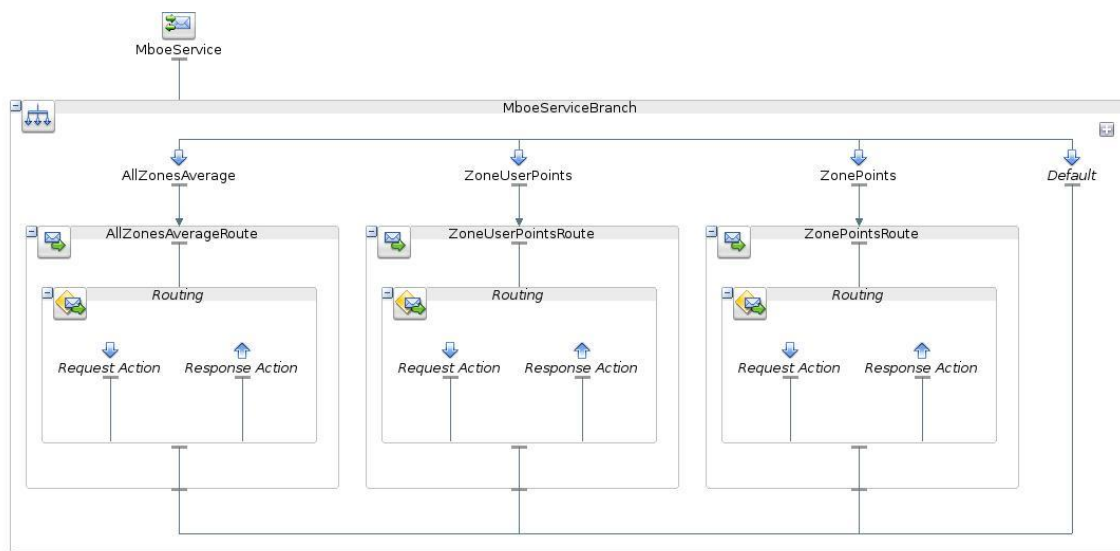


Ilustración 5.5.1 (M.C. Service Proxy)

5.6. Diagrama de Mapeo de LINQ

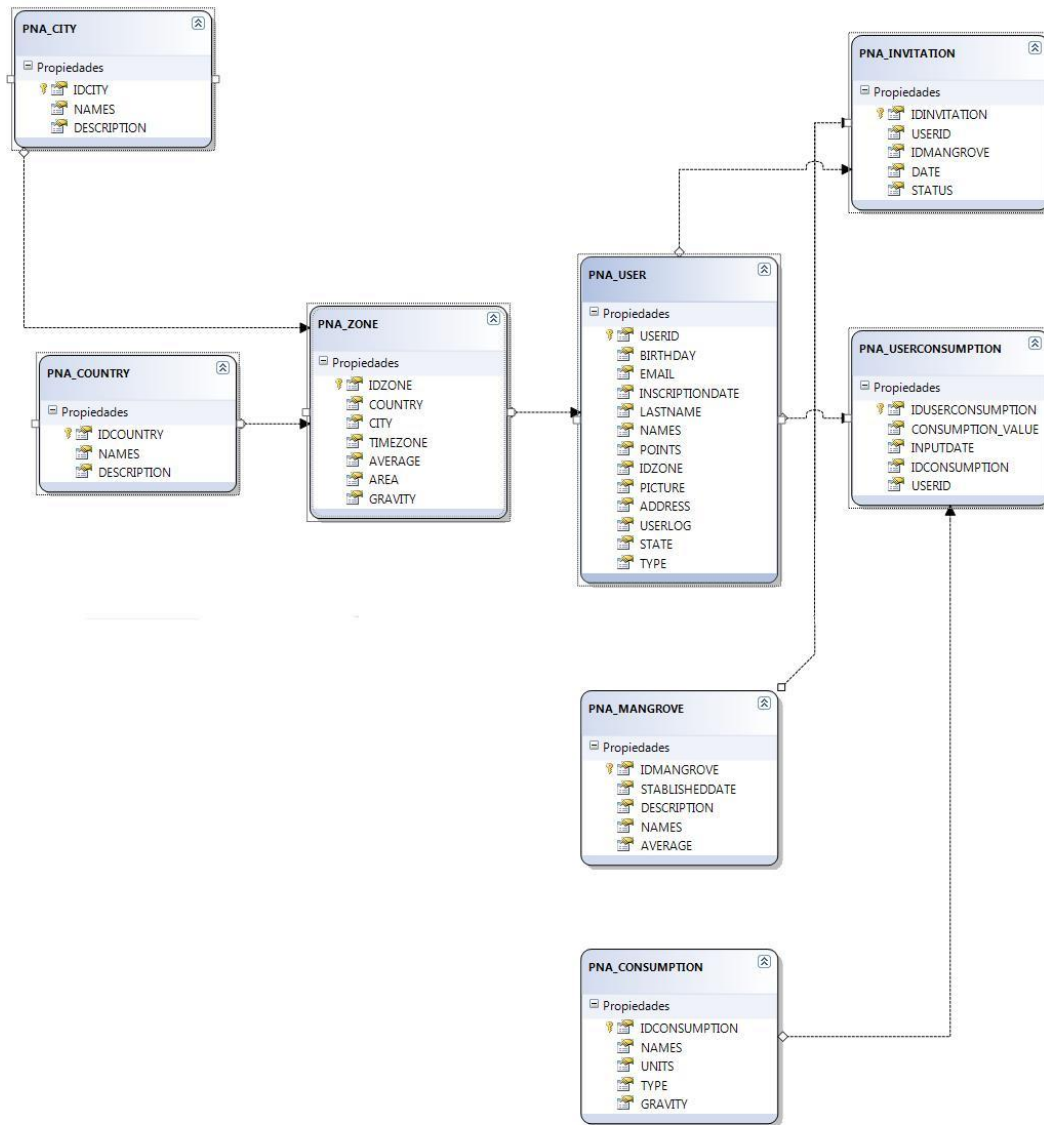


Ilustración 5.6.1 (M.C. mapping de LINQ)

5.7. Pruebas Unitarias

Las pruebas unitarias permiten a los desarrolladores el establecer el nivel de eficacia del código implementado para así poder garantizar su correcto funcionamiento de acuerdo a lo establecido en etapas de diseño en cualquier proyecto.

Es así como en el presente proyecto motivado esencialmente por la importancia de la garantía del correcto funcionamiento de cada uno de los elementos que componen la compleja lógica de negocio, se llevaron a cabo diferentes pruebas unitarias según la necesidad surgida en cada caso. Estas pruebas se llevaron a cabo bajo la vista y supervisión del manejador de pruebas incluido en la versión Professional de Visual Studio 2008, que en este caso fue el IDE empleado, con la ayuda de varios Asserts se creo un proyecto de pruebas denominado:

- ParanaSoft.Parana.ManagerImplTest

El mismo se empleo para la ejecución de pruebas sobre los métodos de las clases de negocio. En el presente trabajo se incluye el cuadro de resultados final de las pruebas incluidas en el ya mencionado proyecto (resumen de las principales), en el que se puede destacar que todas las funciones y métodos evaluados pasaron estas sin ningún inconveniente.

Resultado	Nombre de la Prueba	Proyecto	Error	Duración	Tipo de Prueba	Espera	Nombre de la Clase
Pasada	FindCityByCountryTest	ParanaSoft.Parana.ManagerImplTest		00:00.4	Prueba unitaria	0:00:00	CityManagerImplTest
Pasada	FindCityByIdTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	CityManagerImplTest
Pasada	FindCityByNameTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	CityManagerImplTest
Pasada	ConsumptionInformationTest	ParanaSoft.Parana.ManagerImplTest		00:00.1	Prueba unitaria	0:00:00	ConsumptionManagerImplTest
Pasada	getConsumptionsByTypeTest	ParanaSoft.Parana.ManagerImplTest		00:00.1	Prueba unitaria	0:00:00	ConsumptionManagerImplTest
Pasada	GetZoneGravityTest	ParanaSoft.Parana.ManagerImplTest		00:00.1	Prueba unitaria	0:00:00	ConsumptionRegistrationManagerImplTest
Pasada	InsertUserConsumptionTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	ConsumptionRegistrationManagerImplTest
Pasada	FindCountryByIdTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	CountryManagerImplTest
Pasada	FindCountryByNameTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	CountryManagerImplTest
Pasada	GetCountriesTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	CountryManagerImplTest
Pasada	ListMangroveByAverageTest	ParanaSoft.Parana.ManagerImplTest		00:00.1	Prueba unitaria	0:00:00	MangroveManagerImplTest
Pasada	ListMangroveByNameTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	MangroveManagerImplTest
Pasada	MagroveInformationPresentationTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	MangroveManagerImplTest
Pasada	GetUsersConsumptionsTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	MobileOperationManagerImplTest
Pasada	GetUsersTest	ParanaSoft.Parana.ManagerImplTest		00:00.1	Prueba unitaria	0:00:00	MobileOperationManagerImplTest
Pasada	GetCountriesTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	MobileOperationManagerImplTest
Pasada	GetCitiesTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	MobileOperationManagerImplTest
Pasada	LookForUserTest	ParanaSoft.Parana.ManagerImplTest		00:00.1	Prueba unitaria	0:00:00	ProfileManagerImplTest
Pasada	ListUserByMangroveTest	ParanaSoft.Parana.ManagerImplTest		00:00.1	Prueba unitaria	0:00:00	ProfileManagerImplTest
Pasada	GetZoneTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	ProfileManagerImplTest
Pasada	GetUserMangroveTest	ParanaSoft.Parana.ManagerImplTest		00:00.1	Prueba unitaria	0:00:00	ProfileManagerImplTest
Pasada	FindByNameTest	ParanaSoft.Parana.ManagerImplTest		00:00.1	Prueba unitaria	0:00:00	ProfileManagerImplTest
Pasada	GetAverageForLocalUserTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	SimulationManagerImplTest
Pasada	GetAllZonesPointsAverageForLocalOperationsTest	ParanaSoft.Parana.ManagerImplTest		00:00.0	Prueba unitaria	0:00:00	SimulationManagerImplTest

Tabla 5.7.1. (Cuadro de Resultados Final de Pruebas Unitarias Ejecutadas)

6. CAPÍTULO: CONCLUSIONES Y RECOMENDACIONES

6.1. Conclusiones

Se implementó el núcleo operacional de Paraná conforme a los requerimientos y peticiones plasmadas por los posibles futuros desarrolladores de interfaces genéricas recolectadas en las historias de usuario

El marco teórico resultó en una eficiente herramienta para comprender, aclarar y exponer las principales tendencias y herramientas a utilizar en la implementación del presente proyecto.

La planificación y diseño del presente proyecto se constituyeron en pilares indispensables para la construcción de la presente aplicación, permitiendo la correcta interpretación de las necesidades de los usuarios finales y brindándonos las herramientas para proporcionar soluciones que proporcionen calidad de servicio entre otros beneficios.

La metodología al ser de carácter iterativo resuelve problemas de eficiencia de código a través de la refactorización, acto que debe ser propiciado constantemente, si la eficiencia en el rendimiento del código fuente es el objetivo a conseguir como en la presente.

El conjunto de pruebas y evaluaciones realizadas durante la construcción del software, en especial a través del uso y aplicación de pruebas de aceptación y unitarias, permitió el detectar cualquier corrección y verificación de la que fuese

susceptible la aplicación para poder brindarle un nivel de satisfacción aceptable a los futuros desarrolladores de interfaces genéricas.

El uso de herramientas orientadas al modelo arquitectónico empleado (SOA), como es sin dudas el ESB de Aqualogic, permiten a los futuros usuarios del núcleo el aumentar su confianza en temas puntuales como escalabilidad de la aplicación, control de tiempos de respuesta, en fin, características propias de QoS, obteniendo criterios propia de gobernanza informática para el núcleo de procesamiento lo que se convierte en una garantía de eficacia y eficiencia pero sobre todo de alta fiabilidad.

El desarrollo del presente núcleo en la actualidad ya es una alternativa real para la colaboración con el usuario final en el manejo y administración de sus recursos convirtiéndose en una fuerte alternativa informática que soporta el cambio de modelo de pensamiento, esto al ver los resultados de los sujetos de prueba.

Una aplicación del presente tipo debe perseguir como objetivo claro la automatización de procesos core de negocio y sus tareas de apoyo, centralizando el procesamiento y esfuerzo en un solo punto, de una manera administrable cabe recalcar, facilitando al conjunto de usuarios el crear sus propias interfaces, ya que de esta manera, su uso , del núcleo, se vuelve más generalizado, sin que esto le implique al desarrollador de dichas interfaces una gran carga de trabajo, liberándolo del peso de implementar lógica de negocio altamente compleja.

La instrucción de los principios de la filosofía sobre la que descansa el presente core así como la de los componentes de las interfaces genéricas y

su forma de uso debe de ser difundida entre los futuros usuarios para su correcto uso.

LINQ posee severas limitaciones en cuanto a características propias como la carga de datos y la administración de conexiones, por tales motivos debe de ser monitoreado constantemente en el flujo de información y así evitar caídas que resulten relevantes.

La presente integración de servicios WCF de fabricante Microsoft con el Aqualogic Service Bus de fabricante Oracle – Bea, de componentes Java de fabricante Sun, se consiste en prueba suficiente para aseverar que la interconexión entre plataformas disímiles es completamente viable en el presente desarrollo presentado y por tanto la integración futura con diferentes servicios de diferentes proveedores es más que posible y probable.

Si es posible llevar a cabo proyectos de integración de alto nivel, de complejidad elevada, con tecnología de vanguardia y técnicas y patrones considerados como mejores prácticas, por parte de estudiantes locales, a pesar de las limitaciones que el medio pueda imponer, siempre y cuando el deseo y la voluntad de cumplir no con las teorías que se pueden inculcar en un aula de estudio, sino con alcanzar el ejemplo de aquellos que bien pueden llamarse profesores y amigos de una voluntad férrea por alcanzar hitos nunca antes sospechados.

6.2. Recomendaciones

Se recomienda a los usuarios del presente núcleo tener en claro la serie de funcionalidades expuestas por este, para que maximicen su desempeño y disminuyan el tiempo de implementación de toda interfaz genérica lo que va a redundar sin dudas mejorar las prestaciones en las que estas se presenten a los usuarios finales ya que menos tiempo en implementación de lógica redundante en un mayor tiempo de dedicación en características finales de usuario como el Look & Feel.

La no utilización de componentes open source que no sean de amplio uso y además de contar con esquemas como el de suscripción para recibir soporte, esto se debe a que cualquier elemento que se introduzca dentro de plataformas de integración como la presente debe de hacerse de forma tal en la que se garantice por todos los medios asequibles el mantenimiento de los niveles de QoS y la no existencia de posibles fallas generalizadas que determinen un detrimento en las normales funciones del proyecto ya que es centralizado y una caída de este tipo ocasionaría pérdidas que podrían llegar a ser en un futuro cercano inconmensurables.

El administrador, encargado de ampliar la funcionalidad e involucrar nuevos servicios debería de tener un amplio conocimiento del Bus de Aqualogic, así como de la presente aplicación, ya que cualquier cambio de tipo plumbing debe de ser realizado de forma transparente para el usuario final y las interfaces ya implementadas no deben de sentir dicha variación.

El presente modelo Arquitectónico SOA plantea la división de funciones y tareas por componentes por tanto es recomendable el separar del presente núcleo la administración de esquemas de seguridad, relegando la tarea a servicios como Live ID antiguamente conocido como .Net Passport, que efectivamente cumplen con el principio de separación ya mencionado permitiendo así la especialización de los recursos del núcleo en tareas centralizadas de procesos de negocio y no en tareas de apoyo.

Las interfaces genéricas que se generen deben procurar su alto nivel de cohesión con el usuario final ya que de esta manera se puede aseverar el establecimiento de puentes de acceso para la masificación del uso del núcleo, las mismas que siempre debe en sí mismas reptar los principios establecidas por la filosofía.

Se recomienda el uso e integración con servicios que represente el uso de estándares como bibliotecas en OOXML, y el uso de feeds RSS para complementar a las interfaces genéricas y brindar así un servicio integral al usuario final.

Desarrollar nuevas etapas del proyecto que incluyan el conjunto de necesidades que surgirán por parte de los desarrolladores de interfaces genéricas, esto a ambos lados del Bus tanto de integración como de exposición, en un sentido integración con empresas públicas como eléctrica y agua potable y sobre esta la exposición de servicios REST y repositorios RSS, bibliotecas en línea bajo uso de estándares, entre otros.

Mantener un criterio open mind y confiar en que el camino que exponga la creatividad siempre resulta viables para así asegurar el desarrollo de futuras versiones del núcleo que cumplan con el objetivo tácito del presente proyecto, el asegurar nuestra supervivencia como especie, y como seres de pensamiento trascendiendo a nuestros propios deseos para alcanzar metas más nobles en conjunto, siempre en aras del bien común.

ANEXOS

Anexo A: Diccionario de Datos

PNA_CITY

Dominio de ciudades soportadas para las zonas geográficas

Nombre de la Columna	Descripción		Clave Primaria	Clave Foránea	Tipo de Dato	Tamaño	Presicion Numerica	Escala Numerica	Nullable	Calculado	Identidad
IDCITY	Identificador de Ciudades	unico	Sí	No	varchar	5	No	No	No	No	No
NAMES	Nombre de la ciudad registrada	descriptor de la ciudad	No	No	varchar	20	No	No	Sí	No	No
DESCRIPTION	Datos descriptivos e informativos de la ciudad	acerca	No	No	text	16	No	No	Sí	No	No

PNA_CONSUMPTION

Dominio de consumos, registro de todos los consumos

Nombre de la Columna	Descripcion	Clave Primaria	Clave Foránea	Tipo de Dato	Tamaño	Presicion Numerica	Escala Numerica	Nullable	Calculado	Identidad
IDCONSUMPTION	Identificador unico de Consumos	Sí	No	int	4	10	No	No	No	Sí
NAMES	Nombres Relacionados de los consumos	No	No	varchar	20	No	No	Sí	No	No
UNITS	Unidad de Medida para Evaluar a los Consumos	No	No	varchar	10	No	No	Sí	No	No
TYPE	Identifica datos propios de la clasificación del consumo	No	No	varchar	5	No	No	Sí	No	No
GRAVITY	Almacena el peso del consumo en el cálculo del promedio indicador	No	No	float	8	53	No	Sí	No	No

PNA_COUNTRY

Almacena el dominio de paises registrados

Nombre de la Columna	Descripcion	Clave Primaria	Clave Foránea	Tipo de Dato	Tamaño	Presicion Numerica	Escala Numerica	Nullable	Calculado	Identidad	
IDCOUNTRY	Identificador de Paises	unico	Sí	No	varchar	5	No	No	No	No	No
NAMES	Almacena el nombre descriptivo del pais registrado		No	No	varchar	20	No	No	Sí	No	No
DESCRIPTION	Guarda datos informativos con respecto al pais		No	No	text	16	No	No	Sí	No	No

PNA_INVITATION

Almacena el conjunto de invitaciones que un usuario tiene a un Mangrove en especifico

Nombre de la Columna	Descripcion	Clave Primaria	Clave Foránea	Tipo de Dato	Tamaño	Presicion Numerica	Escala Numerica	Nullable	Calculado	Identidad
IDINVITATION	Identificador unico de las invitaciones para usuarios	Sí	No	int	4	10	No	No	No	Sí
USERID	Identificador del usuario al que le perte	No	Sí	varchar	200	No	No	Sí	No	No
IDMANGROVE	Identificador del mangrove al que ha sido invitado el usuario	No	Sí	int	4	10	No	Sí	No	No
DATE	Fecha en la que la invitacion es enviada	No	No	datetime	8	23	3	Sí	No	No
STATUS	Establece si la invitacion se encuentra activa	No	No	char	Sí	No	No	Sí	No	No

PNA_MANGROVE

Almacena el total de mangroves creados y registrados

Nombre de la Columna	Descripcion	Clave Primaria	Clave Foránea	Tipo de Dato	Tamaño	Presicion Numerica	Escala Numerica	Nullable	Calculado	Identidad
IDMANGROVE	Identificador unico de Mangroves	Sí	No	int	4	10	No	No	No	Sí
STABLISHEDDATE	Indica la fecha en el que el mangrove fue establecido	No	No	datetime	8	23	3	Sí	No	No
DESCRIPTION	Almacena la descripcion del mangrove establecida por su creador	No	No	text	16	No	No	Sí	No	No
NAMES	Almacena el identificador del mangrove	No	No	varchar	20	No	No	Sí	No	No
AVERAGE	Almacena el promedio de calificaciones del mangrove, dato calculado	No	No	float	8	53	No	Sí	No	No

PNA_USER

Almacena a los usuarios registrados y sus datos de vinculacion conel ambiente de la base

Nombre de la Columna	Descripcion	Clave Primaria	Clave Foránea	Tipo de Dato	Tamaño	Presicion Numerica	Escala Numerica	Nullable	Calculado	Identidad
USERID	Identificador unico de Usuarios	Sí	No	varchar	200	No	No	No	No	No
BIRTHDAY	Fecha de nacimiento del usuario	No	No	datetime	8	23	3	Sí	No	No
EMAIL	direccion de correo electronico comporbado	No	No	varchar	20	No	No	Sí	No	No
INSCRIPTIONDATE	Fecha de registro en Parana	No	No	datetime	8	23	3	Sí	No	No
LASTNAME	Apellidos del Usuario	No	No	varchar	30	No	No	Sí	No	No
NAMES	Nombres del usuario	No	No	varchar	30	No	No	Sí	No	No
POINTS	almacena el total de puntos de un usuario	No	No	int	4	10	No	Sí	No	No
IDZONE	codigo identificador de la zona a la que pertenece actualmente el usuario	No	Sí	int	4	10	No	Sí	No	No

PICTURE	foto del usuario registrado	No	No	varchar	30	No	No	Sí	No	No
ADDRESS	direccion de domicilio actual del usuario registrado	No	No	varchar	30	No	No	Sí	No	No
USERLOG	nombre a mostrar del usuario registrado	No	No	varchar	15	No	No	Sí	No	No
STATE	indica si el usuario se encuentra activo o inactivo	No	No	varchar	50	No	No	Sí	No	No
TYPE	Indica el tipo al que el usuario pertenece	No	No	varchar	2	No	No	Sí	No	No

PNA_USERCONSUMPTION

Almacena la lista de consumos realizados por un usuario en específico

Nombre de la Columna	Descripcion	Clave Primaria	Clave Foránea	Tipo de Dato	Tamaño	Presicion Numerica	Escala Numerica	Nullable	Calculado	Identidad
IDUSERCONSUMPTION	Identificador unico de un consumo de un usuario	Sí	No	int	4	10	No	No	No	Sí
CONSUMPTION_VALUE	Valor total del consumo	No	No	float	8	53	No	Sí	No	No
INPUTDATE	Fecha de Registro del consumo	No	No	datetime	8	23	3	Sí	No	No
IDCONSUMPTION	Identificador unico de consumo	No	Sí	int	4	10	No	Sí	No	No
USERID	Identificador unico de usuario	No	Sí	varchar	200	No	No	Sí	No	No

PNA_ZONE

Almacena el total de zonas geograficas registradas para usuarios organizadas por ciudad

Nombre de la Columna	Descripcion	Clave Primaria	Clave Foránea	Tipo de Dato	Tamaño	Presicion Numerica	Escala Numerica	Nullable	Calculado	Identidad
IDZONE	Identificador unico de zonas	Sí	No	int	4	10	No	No	No	Sí
COUNTRY	Pais en el que se encuentra la zona	No	Sí	varchar	5	No	No	Sí	No	No
CITY	ciudad en la que se establece la zona	No	Sí	varchar	5	No	No	Sí	No	No
TIMEZONE	zona horaria en la que se encuentra la zona	No	No	int	4	10	No	Sí	No	No
AVERAGE	Promedio de la zona, cantidad calculada	No	No	float	8	53	No	Sí	No	No
AREA	Indica la especificacion de la zona	No	No	varchar	50	No	No	Sí	No	No
GRAVITY	Factor de multiplicacion de la zona geografica	No	No	float	8	53	No	Sí	No	No

Anexo B: Script de Creación de Base de Datos

```
USE [master]
GO
/***** Object: Database [PARANA] Script Date: 02/07/2010 22:18:28
*****/
CREATE DATABASE [PARANA] ON PRIMARY
( NAME = N'PARANA', FILENAME = N'C:\Base\PARANA.mdf' , SIZE = 3072KB ,
MAXSIZE = UNLIMITED, FILEGROWTH = 1024KB )
LOG ON
( NAME = N'PARANA_log', FILENAME = N'C:\Base\PARANA_log.ldf' , SIZE =
1024KB , MAXSIZE = 2048GB , FILEGROWTH = 10%)
COLLATE Modern_Spanish_CI_AS
GO
EXEC dbo.sp_dbcmptlevel @dbname=N'PARANA', @new_cmptlevel=90
GO
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))
begin
EXEC [PARANA].[dbo].[sp_fulltext_database] @action = 'disable'
end
GO
ALTER DATABASE [PARANA] SET ANSI_NULL_DEFAULT OFF
GO
ALTER DATABASE [PARANA] SET ANSI_NULLS OFF
GO
ALTER DATABASE [PARANA] SET ANSI_PADDING OFF
GO
ALTER DATABASE [PARANA] SET ANSI_WARNINGS OFF
GO
ALTER DATABASE [PARANA] SET ARITHABORT OFF
GO
ALTER DATABASE [PARANA] SET AUTO_CLOSE ON
GO
ALTER DATABASE [PARANA] SET AUTO_CREATE_STATISTICS ON
GO
ALTER DATABASE [PARANA] SET AUTO_SHRINK OFF
GO
ALTER DATABASE [PARANA] SET AUTO_UPDATE_STATISTICS ON
GO
ALTER DATABASE [PARANA] SET CURSOR_CLOSE_ON_COMMIT OFF
GO
ALTER DATABASE [PARANA] SET CURSOR_DEFAULT GLOBAL
GO
ALTER DATABASE [PARANA] SET CONCAT_NULL_YIELDS_NULL OFF
GO
ALTER DATABASE [PARANA] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [PARANA] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [PARANA] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [PARANA] SET DISABLE_BROKER
GO
```

```

ALTER DATABASE [PARANA] SET AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER DATABASE [PARANA] SET DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [PARANA] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [PARANA] SET ALLOW_SNAPSHOT_ISOLATION OFF
GO
ALTER DATABASE [PARANA] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [PARANA] SET READ_WRITE
GO
ALTER DATABASE [PARANA] SET RECOVERY FULL
GO
ALTER DATABASE [PARANA] SET MULTI_USER
GO
ALTER DATABASE [PARANA] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [PARANA] SET DB_CHAINING OFF

--Creacion de Tablas

USE [PARANA]
GO
/***** Objeto: Table [dbo].[PNA_CITY]      Fecha de la secuencia de
comandos: 02/07/2010 22:22:30 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PNA_CITY] (
    [IDCITY] [varchar] (5) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [NAMES] [varchar] (20) COLLATE Modern_Spanish_CI_AS NULL,
    [DESCRIPTION] [text] COLLATE Modern_Spanish_CI_AS NULL,
PRIMARY KEY CLUSTERED
(
    [IDCITY] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description',
@value=N'Identificador unico de Ciudades' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_CITY',
@level2type=N'COLUMN',@level2name=N'IDCITY'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Nombre
descriptor de la ciudad registrada' ,
@level0type=N'SHEMA',@level0name=N'dbo',

```

```

@level1type=N'TABLE',@level1name=N'PNA_CITY',
@level2type=N'COLUMN',@level2name=N'NAMES'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Datos
descriptivos e informativos acerca de la ciudad' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_CITY',
@level2type=N'COLUMN',@level2name=N'DESCRPTION'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Dominio
de ciudades soportadas para las zonas geograficas' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_CITY'

USE [PARANA]
GO
/***** Objeto: Table [dbo].[PNA_CONSUMPTION]      Fecha de la secuencia de
comandos: 02/07/2010 22:23:10 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PNA_CONSUMPTION] (
    [IDCONSUMPTION] [int] IDENTITY(1,1) NOT NULL,
    [NAMES] [varchar](20) COLLATE Modern_Spanish_CI_AS NULL,
    [UNITS] [varchar](10) COLLATE Modern_Spanish_CI_AS NULL,
    [TYPE] [varchar](5) COLLATE Modern_Spanish_CI_AS NULL,
    [GRAVITY] [float] NULL,
    CONSTRAINT [PK__PNA_CONSUMPTION__0CBAE877] PRIMARY KEY CLUSTERED
(
    [IDCONSUMPTION] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description',
@value=N'Identificador unico de Consumos' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_CONSUMPTION',
@level2type=N'COLUMN',@level2name=N'IDCONSUMPTION'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Nombres
Relacionados de los consumos' , @level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_CONSUMPTION',
@level2type=N'COLUMN',@level2name=N'NAMES'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Unidad de
Medida para Evaluar a los Consumos' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_CONSUMPTION',
@level2type=N'COLUMN',@level2name=N'UNITS'

```

```

GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description',
@value=N'Identifica datos propios de la clasificaci'on del consumo' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_CONSUMPTION',
@level2type=N'COLUMN',@level2name=N'TYPE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Almacena
el peso del consumo en el calculo del promedio indicador' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_CONSUMPTION',
@level2type=N'COLUMN',@level2name=N'GRAVITY'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Dominio
de consumos, registro de todos los consumos' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_CONSUMPTION'

USE [PARANA]
GO
/***** Objeto: Table [dbo].[PNA_COUNTRY]      Fecha de la secuencia de
comandos: 02/07/2010 22:23:35 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PNA_COUNTRY] (
    [IDCOUNTRY] [varchar](5) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [NAMES] [varchar](20) COLLATE Modern_Spanish_CI_AS NULL,
    [DESCRIPTION] [text] COLLATE Modern_Spanish_CI_AS NULL,
PRIMARY KEY CLUSTERED
(
    [IDCOUNTRY] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description',
@value=N'Identificador unico de Paises' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_COUNTRY',
@level2type=N'COLUMN',@level2name=N'IDCOUNTRY'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Almacena
el nombre descriptivo del pais registrado' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_COUNTRY',
@level2type=N'COLUMN',@level2name=N'NAMES'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Guarda
datos informativos con respecto al pais' ,

```



```

@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_COUNTRY',
@level2type=N'COLUMN',@level2name=N'DESCRIPTION'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Almacena
el dominio de paises registrados' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_COUNTRY'

USE [PARANA]
GO
/***** Objeto: Table [dbo].[PNA_INVITATION]      Fecha de la secuencia de
comandos: 02/07/2010 22:24:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PNA_INVITATION] (
    [IDINVITATION] [int] IDENTITY(1,1) NOT NULL,
    [USERID] [varchar] (200) COLLATE Modern_Spanish_CI_AS NULL,
    [IDMANGROVE] [int] NULL,
    [DATE] [datetime] NULL,
    [STATUS] [char] (1) COLLATE Modern_Spanish_CI_AS NULL,
    CONSTRAINT [PK__PNA_INVITATION__1B0907CE] PRIMARY KEY CLUSTERED
(
    [IDINVITATION] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description',
@value=N'Identificador unico de las invitaciones para usuarios' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_INVITATION',
@level2type=N'COLUMN',@level2name=N'IDINVITATION'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description',
@value=N'Identificador del usuario al que le perte' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_INVITATION',
@level2type=N'COLUMN',@level2name=N'USERID'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description',
@value=N'Identificador del mangrove al que ha sido invitado el usuario' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_INVITATION',
@level2type=N'COLUMN',@level2name=N'IDMANGROVE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Fecha en
la que la invitacion es enviada' ,
@level0type=N'SCHEMA',@level0name=N'dbo',

```

```

@level1type=N'TABLE',@level1name=N'PNA_INVITATION',
@level2type=N'COLUMN',@level2name=N'DATE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Establece
si la invitacion se encuentra activa' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_INVITATION',
@level2type=N'COLUMN',@level2name=N'STATUS'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Almacena
el conjunto de invitaciones que un usuario tiene a un Mangrove en
especifico' , @level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_INVITATION'
GO
ALTER TABLE [dbo].[PNA_INVITATION] WITH CHECK ADD CONSTRAINT
[FK_PNA_INVIT_IDMAN_1CF15040] FOREIGN KEY([IDMANGROVE])
REFERENCES [dbo].[PNA_MANGROVE] ([IDMANGROVE])
GO
ALTER TABLE [dbo].[PNA_INVITATION] CHECK CONSTRAINT
[FK_PNA_INVIT_IDMAN_1CF15040]
GO
ALTER TABLE [dbo].[PNA_INVITATION] WITH CHECK ADD CONSTRAINT
[FK_PNA_INVIT_USERI_1BFD2C07] FOREIGN KEY([USERID])
REFERENCES [dbo].[PNA_USER] ([USERID])
GO
ALTER TABLE [dbo].[PNA_INVITATION] CHECK CONSTRAINT
[FK_PNA_INVIT_USERI_1BFD2C07]

USE [PARANA]
GO
/***** Objeto: Table [dbo].[PNA_MANGROVE] Fecha de la secuencia de
comandos: 02/07/2010 22:24:31 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PNA_MANGROVE] (
[IDMANGROVE] [int] IDENTITY(1,1) NOT NULL,
[STABLISHEDDATE] [datetime] NULL,
[DESCRIPTION] [text] COLLATE Modern_Spanish_CI_AS NULL,
[NAMES] [varchar](20) COLLATE Modern_Spanish_CI_AS NULL,
[AVERAGE] [float] NULL,
CONSTRAINT [PK_PNA_MANGROVE_7E6CC920] PRIMARY KEY CLUSTERED
(
[IDMANGROVE] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description',
@value=N'Identificador unico de Mangroves' ,
@level0type=N'SHEMA',@level0name=N'dbo',

```

```

@level1type=N'TABLE',@level1name=N'PNA_MANGROVE',
@level2type=N'COLUMN',@level2name=N'IDMANGROVE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Indica la
fecha en el que el mangrove fue establecido' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_MANGROVE',
@level2type=N'COLUMN',@level2name=N'STABLISHEDDATE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Almacena
la descripcion del mangrove establecida por su creador' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_MANGROVE',
@level2type=N'COLUMN',@level2name=N'DESCRIPTION'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Almacena
el identificador del mangrove' , @level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_MANGROVE',
@level2type=N'COLUMN',@level2name=N'NAMES'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Almacena
el promedio de calificaciones del mangrove, dato calculado' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_MANGROVE',
@level2type=N'COLUMN',@level2name=N'AVERAGE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Almacena
el total de mangroves creados y registrados' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_MANGROVE'

```

```
USE [PARANA]
```

```
GO
```

```
/****** Objeto: Table [dbo].[PNA_USER] Fecha de la secuencia de
comandos: 02/07/2010 22:24:57 *****/
```

```
SET ANSI_NULLS ON
```

```
GO
```

```
SET QUOTED_IDENTIFIER ON
```

```
GO
```

```
SET ANSI_PADDING ON
```

```
GO
```

```
CREATE TABLE [dbo].[PNA_USER] (
    [USERID] [varchar] (200) COLLATE Modern_Spanish_CI_AS NOT NULL,
    [BIRTHDAY] [datetime] NULL,
    [EMAIL] [varchar] (20) COLLATE Modern_Spanish_CI_AS NULL,
    [INSCRIPTIONDATE] [datetime] NULL,
    [LASTNAME] [varchar] (30) COLLATE Modern_Spanish_CI_AS NULL,
    [NAMES] [varchar] (30) COLLATE Modern_Spanish_CI_AS NULL,
    [POINTS] [int] NULL,
    [IDZONE] [int] NULL,
    [PICTURE] [varchar] (30) COLLATE Modern_Spanish_CI_AS NULL,
    [ADDRESS] [varchar] (30) COLLATE Modern_Spanish_CI_AS NULL,
    [USERLOG] [varchar] (15) COLLATE Modern_Spanish_CI_AS NULL,
    [STATE] [varchar] (50) COLLATE Modern_Spanish_CI_AS NULL,

```

```

        [TYPE] [varchar] (2) COLLATE Modern_Spanish_CI_AS NULL,
    CONSTRAINT [PK__PNA_USER__7C8480AE] PRIMARY KEY CLUSTERED
    (
        [USERID] ASC
    ) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
    OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
    ) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description',
@value=N'Identificador unico de Usuarios' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'USERID'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Fecha de
nacimiento del usuario' , @level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'BIRTHDAY'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'direccion
de correo electronico comporbado' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'EMAIL'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Fecha de
registro en Parana' , @level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'INSCRIPTIONDATE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Apellidos
del Usuario' , @level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'LASTNAME'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Nombres
del usuario' , @level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'NAMES'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'almacena
el total de puntos de un usuario' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'POINTS'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'codigo
identificador de la zona a la que pertenece actualmente el usuario' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'IDZONE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'foto del
usuario registrado' , @level0type=N'SHEMA',@level0name=N'dbo',

```

```

@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'PICTURE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'direccion
de domicilio actual del usuario registrado' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'ADDRESS'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'nombre a
mostrar del usuario registrado' , @level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'USERLOG'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'indica si
el usuario se encuentra activo o inactivo' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'STATE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Indica el
tipo al que el usuario pertenece' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER',
@level2type=N'COLUMN',@level2name=N'TYPE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Almacena
a los usuarios registrados y sus datos de vinculacion conel ambiente de la
base' , @level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USER'
GO
ALTER TABLE [dbo].[PNA_USER] WITH CHECK ADD CONSTRAINT
[FK_PNA_USER_PNA_ZONE] FOREIGN KEY ([IDZONE])
REFERENCES [dbo].[PNA_ZONE] ([IDZONE])
GO
ALTER TABLE [dbo].[PNA_USER] CHECK CONSTRAINT [FK_PNA_USER_PNA_ZONE]

USE [PARANA]
GO
/***** Objeto: Table [dbo].[PNA_ZONE] Fecha de la secuencia de
comandos: 02/07/2010 22:25:49 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PNA_ZONE] (
[IDZONE] [int] IDENTITY(1,1) NOT NULL,
[COUNTRY] [varchar](5) COLLATE Modern_Spanish_CI_AS NULL,
[CITY] [varchar](5) COLLATE Modern_Spanish_CI_AS NULL,
[TIMEZONE] [int] NULL,
[AVERAGE] [float] NULL,
[AREA] [varchar](50) COLLATE Modern_Spanish_CI_AS NULL,
[GRAVITY] [float] NULL,
CONSTRAINT [PK__PNA_ZONE__0519C6AF] PRIMARY KEY CLUSTERED
(

```

```

        [IDZONE] ASC
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description',
@value=N'Identificador unico de zonas' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_ZONE',
@level2type=N'COLUMN',@level2name=N'IDZONE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Pais en
el que se encuentra la zona' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_ZONE',
@level2type=N'COLUMN',@level2name=N'COUNTRY'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'ciudad en
la que se establece la zona' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_ZONE',
@level2type=N'COLUMN',@level2name=N'CITY'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'zona
horaria en la que se encuentra la zona' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_ZONE',
@level2type=N'COLUMN',@level2name=N'TIMEZONE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Promedio
de la zona, cantidad calculada' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_ZONE',
@level2type=N'COLUMN',@level2name=N'AVERAGE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Indica la
especificacion de la zona' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_ZONE',
@level2type=N'COLUMN',@level2name=N'AREA'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Factor de
multiplicacion de la zona geografica' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_ZONE',
@level2type=N'COLUMN',@level2name=N'GRAVITY'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Almacena
el total de zonas geograficas registradas para usuarios organizadas por
ciudad' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_ZONE'
GO
ALTER TABLE [dbo].[PNA_ZONE] WITH CHECK ADD CONSTRAINT
[FK__PNA_ZONE__CITY__07020F21] FOREIGN KEY ([CITY])
REFERENCES [dbo].[PNA_CITY] ([IDCITY])
GO
ALTER TABLE [dbo].[PNA_ZONE] CHECK CONSTRAINT
[FK__PNA_ZONE__CITY__07020F21]

```

```

GO
ALTER TABLE [dbo].[PNA_ZONE] WITH CHECK ADD CONSTRAINT
[FK__PNA_ZONE__COUNTR__060DEAE8] FOREIGN KEY ([COUNTRY])
REFERENCES [dbo].[PNA_COUNTRY] ([IDCOUNTRY])
GO
ALTER TABLE [dbo].[PNA_ZONE] CHECK CONSTRAINT
[FK__PNA_ZONE__COUNTR__060DEAE8]

USE [PARANA]
GO
/***** Objeto: Table [dbo].[PNA_USERCONSUMPTION] Fecha de la secuencia
de comandos: 02/07/2010 22:26:30 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
SET ANSI_PADDING ON
GO
CREATE TABLE [dbo].[PNA_USERCONSUMPTION] (
    [IDUSERCONSUMPTION] [int] IDENTITY(1,1) NOT NULL,
    [CONSUMPTION_VALUE] [float] NULL,
    [INPUTDATE] [datetime] NULL,
    [IDCONSUMPTION] [int] NULL,
    [USERID] [varchar] (200) COLLATE Modern_Spanish_CI_AS NULL,
    CONSTRAINT [PK__PNA_USERCONSUMPT__117F9D94] PRIMARY KEY CLUSTERED
(
    [IDUSERCONSUMPTION] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY =
OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
SET ANSI_PADDING OFF
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Identificador unico de un consumo de un usuario' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USERCONSUMPTION',
@level2type=N'COLUMN',@level2name=N'IDUSERCONSUMPTION'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Valor total del consumo' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USERCONSUMPTION',
@level2type=N'COLUMN',@level2name=N'CONSUMPTION_VALUE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Fecha de Registro del consumo' , @level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USERCONSUMPTION',
@level2type=N'COLUMN',@level2name=N'INPUTDATE'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Identificador unico de consumo' ,
@level0type=N'SCHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USERCONSUMPTION',
@level2type=N'COLUMN',@level2name=N'IDCONSUMPTION'
GO

```

```

EXEC sys.sp_addextendedproperty @name=N'MS_Description',
@value=N'Identificador unico de usuario' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USERCONSUMPTION',
@level2type=N'COLUMN',@level2name=N'USERID'
GO
EXEC sys.sp_addextendedproperty @name=N'MS_Description', @value=N'Almacena
la lista de consumos realizados por un usuario en especifico' ,
@level0type=N'SHEMA',@level0name=N'dbo',
@level1type=N'TABLE',@level1name=N'PNA_USERCONSUMPTION'
GO
ALTER TABLE [dbo].[PNA_USERCONSUMPTION] WITH CHECK ADD CONSTRAINT
[FK__PNA_USERC__IDCON__1273C1CD] FOREIGN KEY ([IDCONSUMPTION])
REFERENCES [dbo].[PNA_CONSUMPTION] ([IDCONSUMPTION])
GO
ALTER TABLE [dbo].[PNA_USERCONSUMPTION] CHECK CONSTRAINT
[FK__PNA_USERC__IDCON__1273C1CD]
GO
ALTER TABLE [dbo].[PNA_USERCONSUMPTION] WITH CHECK ADD CONSTRAINT
[FK__PNA_USERC__USERI__1367E606] FOREIGN KEY ([USERID])
REFERENCES [dbo].[PNA_USER] ([USERID])
GO
ALTER TABLE [dbo].[PNA_USERCONSUMPTION] CHECK CONSTRAINT
[FK__PNA_USERC__USERI__1367E606]

```


Anexo C: Manual Despliegue de Proxy Services sobre Aqualogic Service Bus 3

Manual Despliegue de Proxy Services sobre Aqualogic Service Bus 3

Información General	
Proyecto:	Diseño, construcción e implementación del núcleo de procesamiento de la versión automatizada de Paraná, bajo un Modelo de Arquitectura Orientada a Servicios
Versión:	1.0
Documentos Relacionados:	Volúmenes del presente trabajo de Investigación

Control de Cambios			
Fecha	Autor	Versión	Observaciones
27 – 01 – 2010	AAAt.	1.0	Creación inicial del documento.

Resumen
Este documento muestra en detalle el conjunto de procedimientos a seguir para el despliegue de los componentes de la aplicación pertenecientes a la capa de integración sobre el ya mencionado Bus, siendo esta forma no restrictiva al existir otros métodos que podrían ser empleados para conseguir el fin ya señalado.

Prerrequisitos

1. *Instalar Weblogic server en versiones 10.x*
2. *Instalar el Aqualogic Service Bus en versiones 3.x*
3. *Instalar Bea Workspace Studio en versión 1.x*
4. *Configurar Bea Worspace Studio para que tenga acceso al Servidor en*

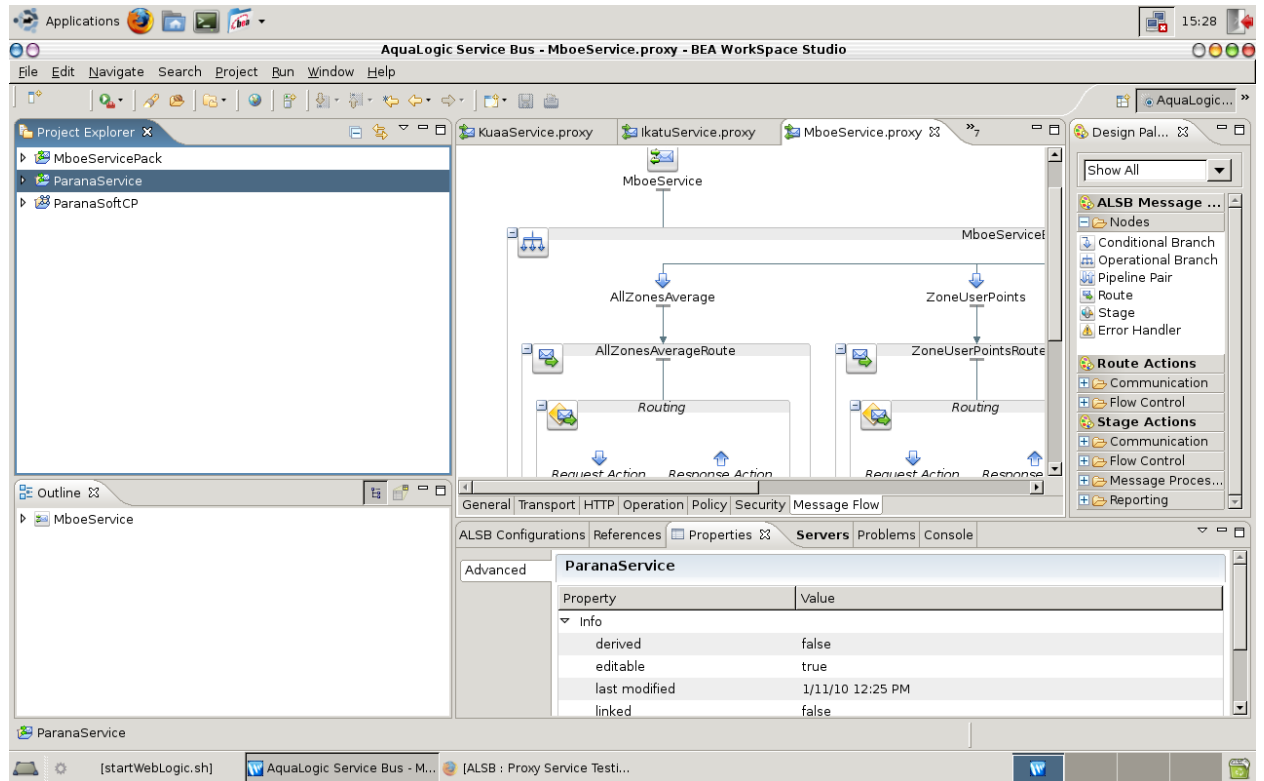
donde se encuentre instalado el ALSB.

Procedimiento

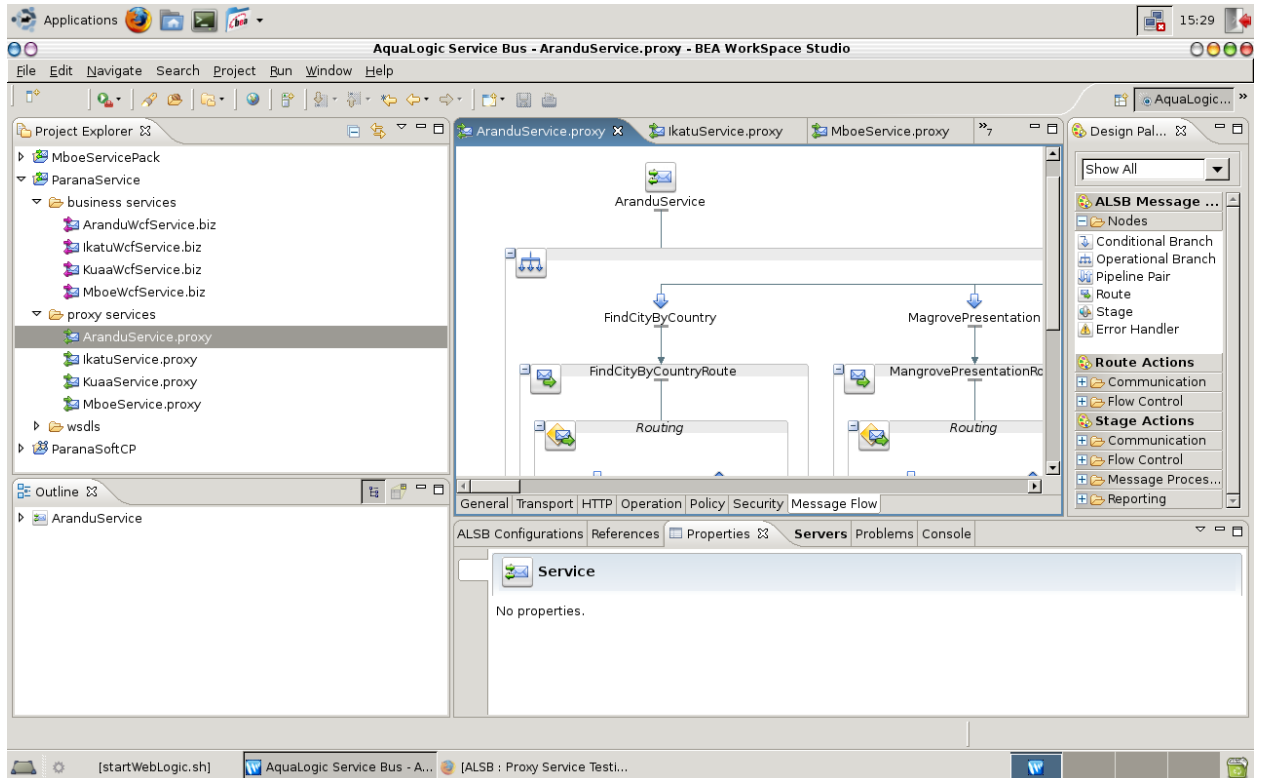
1. Iniciar una instancia de BEA Workspace Studio del menú tal como se ve en la imagen:



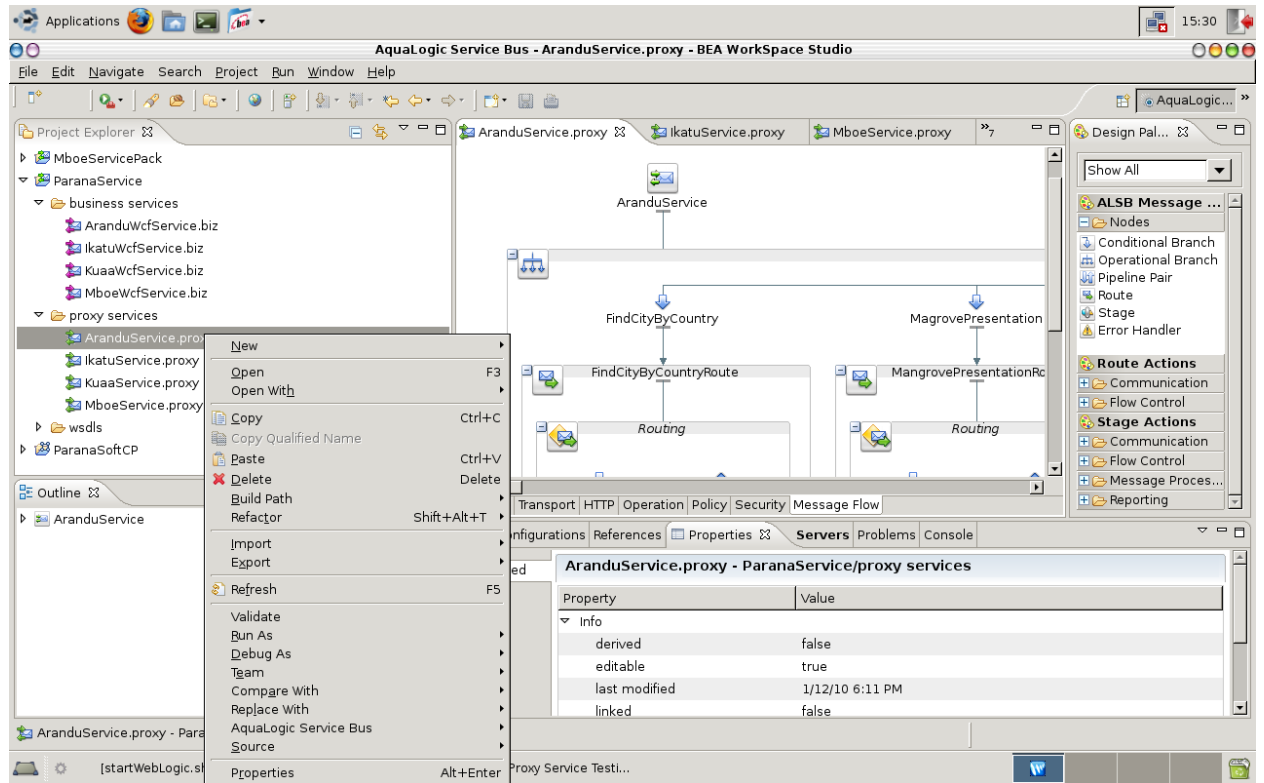
2. Seleccionar el proyecto del que se ha planteado el publicar los servicios de integración, cabe recalcar que debe ser el proyecto contenedor de tipo ALSB y no el proyecto de configuración de LASB que resulta genérico a todas las aplicaciones desplegadas.



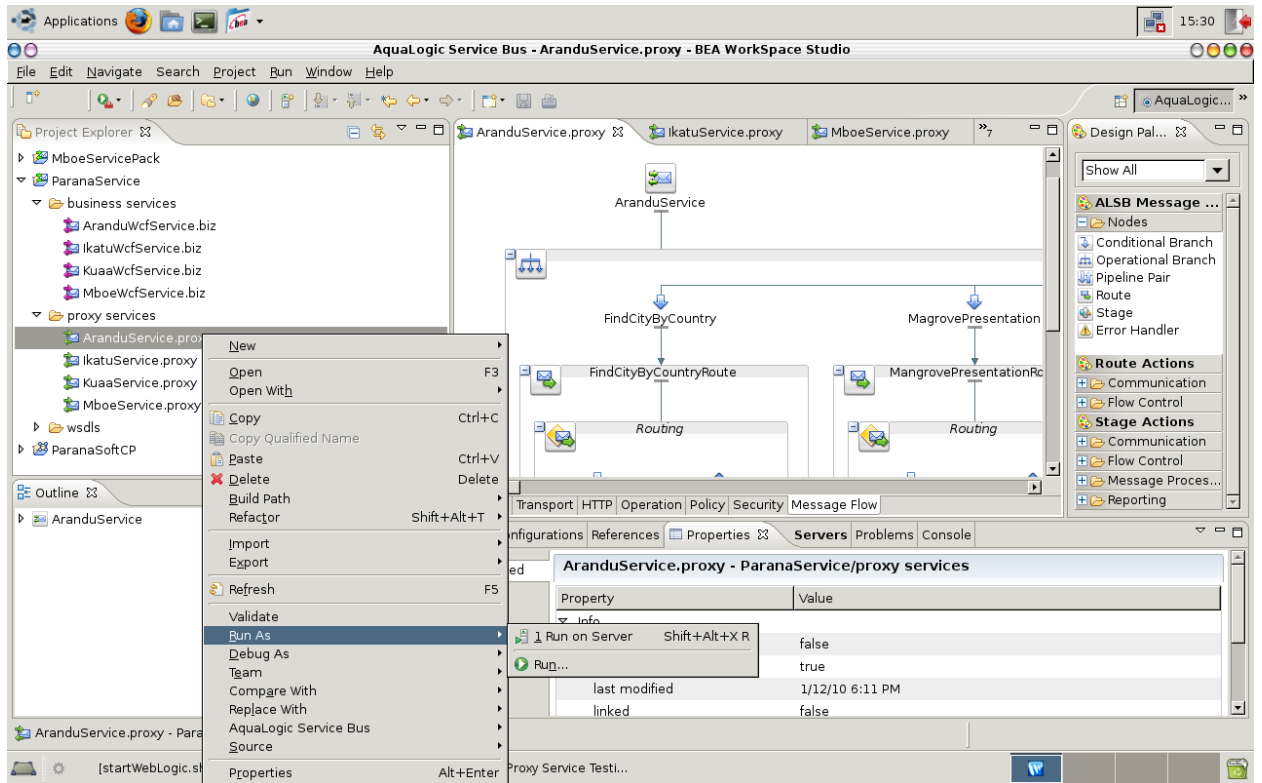
3. Seleccionar el componente de tipo proxy que se desea desplegar en el servidor, que como cabe señalar ya debió de haber sido configurado en BEA Workspace Studio,



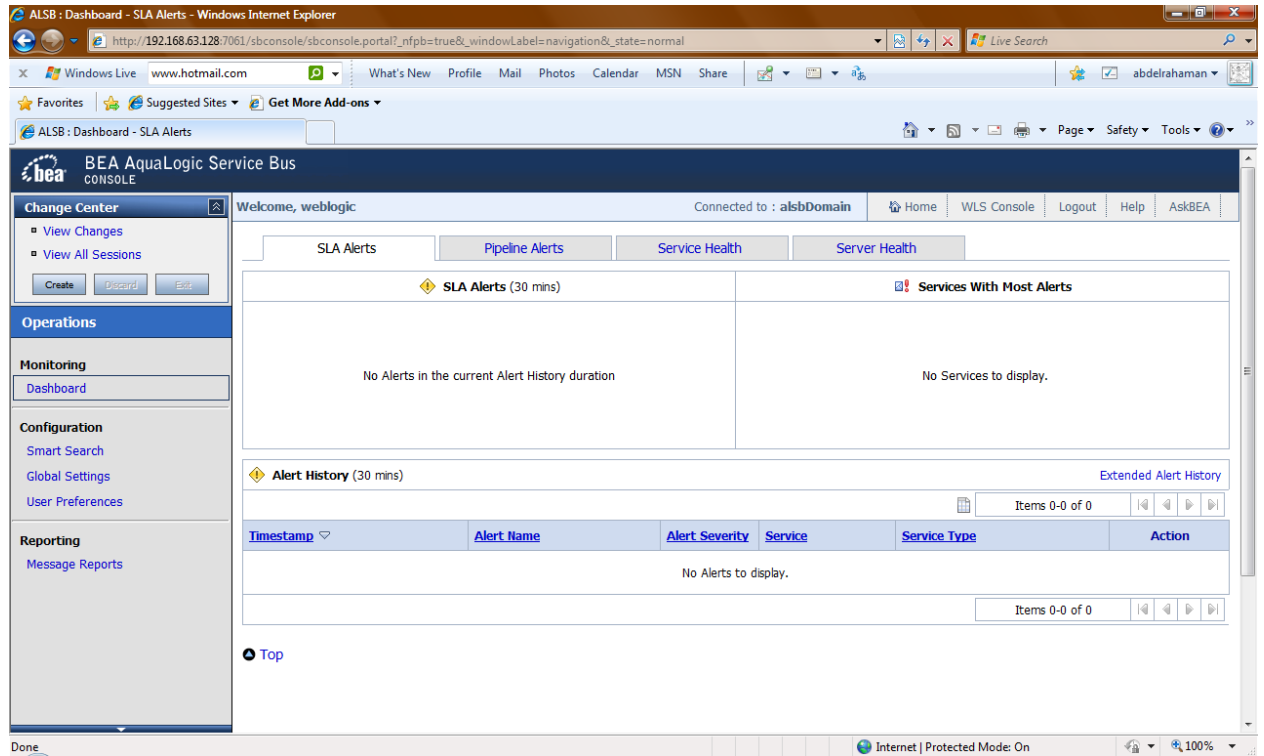
4. Abrir el menu contextual, en el cual consta la opción de “Run As”.



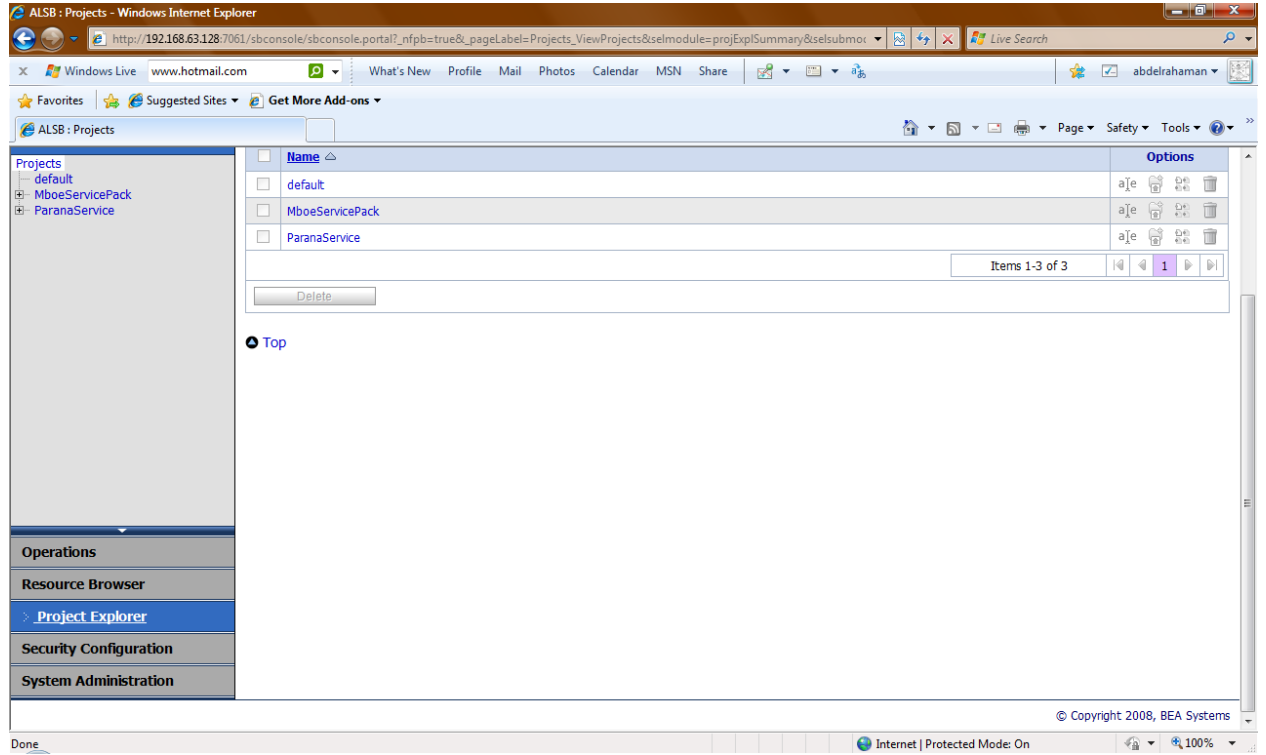
5. Seleccionar "Run As" para posteriormente oprimir la opción "Run on Server"



6. Abrir una ventana de explorador en cualquier equipo que pueda tener acceso al Servidor de Aqualogic, para proceder a ingresar a la consola de administración del bus que para el presente caso ocupará la URL:
<http://192.168.63.128:7061/sbconsole>



7. Seleccionarla opción de exploración del proyecto, en donde aparecerán todos las aplicaciones contenedoras que han sido desplegadas en el servidor.



8. Explorar en el árbol hasta localizar los servicios tipo proxy publicados.

The screenshot shows a web browser window displaying a management console. The address bar shows a URL starting with 'http://192.168.63.128'. The left sidebar contains a tree view with the following structure:

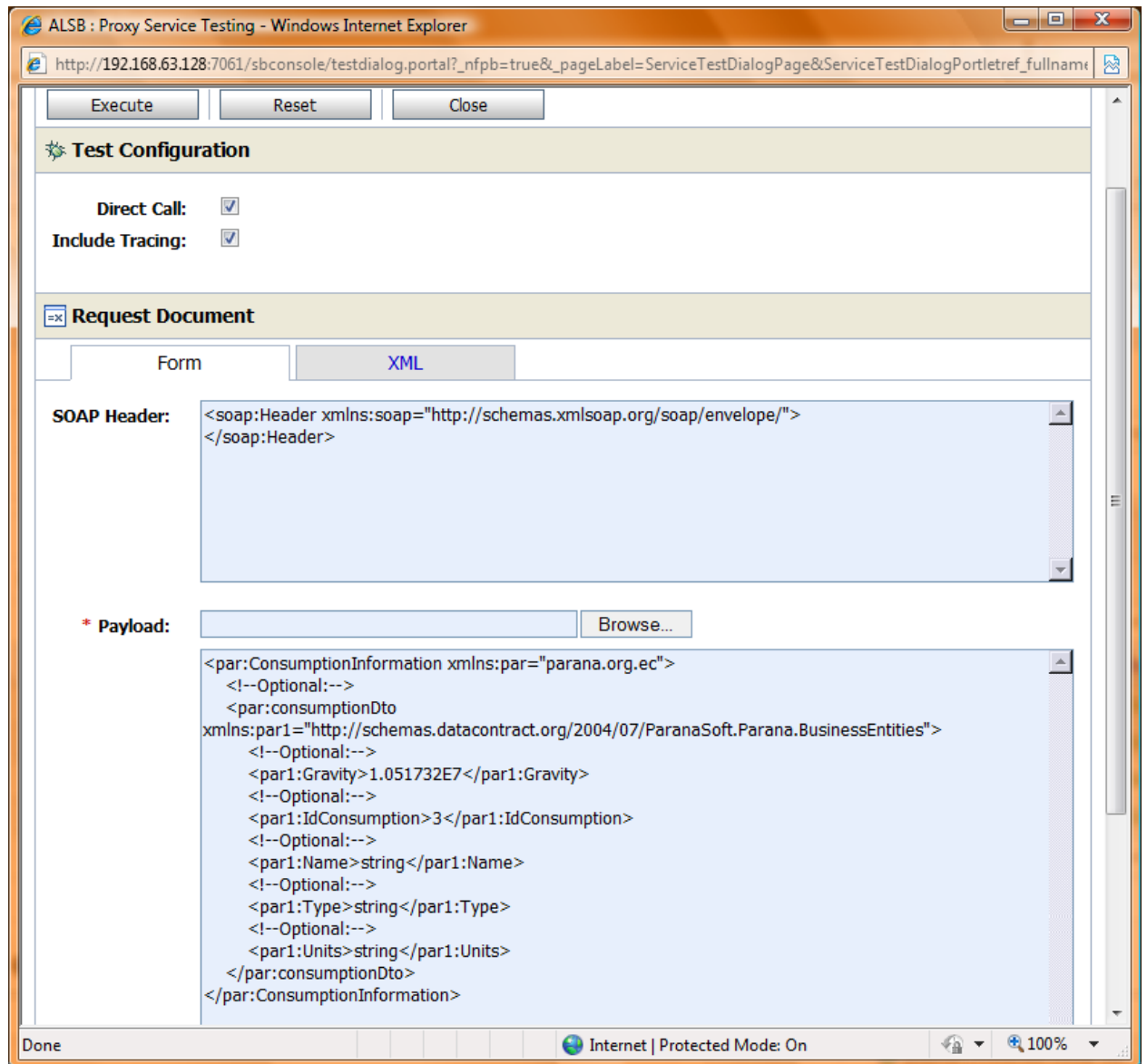
- MboeServicePack
 - ParanaService
 - business services
 - proxy services
 - wsdl

The main content area is titled 'proxy services' and contains a 'Resources' section. It features a table with the following data:

Name	Resource Type	Actions	Options
AranduService	Proxy Service	[Icons]	[Icons]
IkatuService	Proxy Service	[Icons]	[Icons]
KuaaService	Proxy Service	[Icons]	[Icons]
MboeService	Proxy Service	[Icons]	[Icons]

The browser's status bar at the bottom indicates 'Internet | Protected Mode: On' and '100%' zoom level. Copyright 2008, BEA Systems is visible in the footer.

9. Oprimir la opción de “Test” para así validar su correcto funcionamiento.



Anexo D: Manual de Creación de Interfaces Genéricas

Manual de Creación de Interfaces Genéricas

Información General	
Proyecto:	Diseño, construcción e implementación del núcleo de procesamiento de la versión automatizada de Paraná, bajo un Modelo de Arquitectura Orientada a Servicios
Versión:	1.0
Documentos Relacionados:	Volúmenes del presente trabajo de Investigación

Control de Cambios			
Fecha	Autor	Versión	Observaciones
27 - 01 - 2010	AAAt.	1.0	Creación inicial del documento.

Resumen
<p>Este documento muestra en detalle el procedimiento para la creación de un cliente de los componentes del núcleo, bajo tecnología Microsoft utilizando para el presente fin al Servicio Kuaa, un componente esencial del núcleo colocado sobre ALSB, y la creación de una aplicación de consola, con la respectiva verificación de la interconexión existente entre ambos</p>

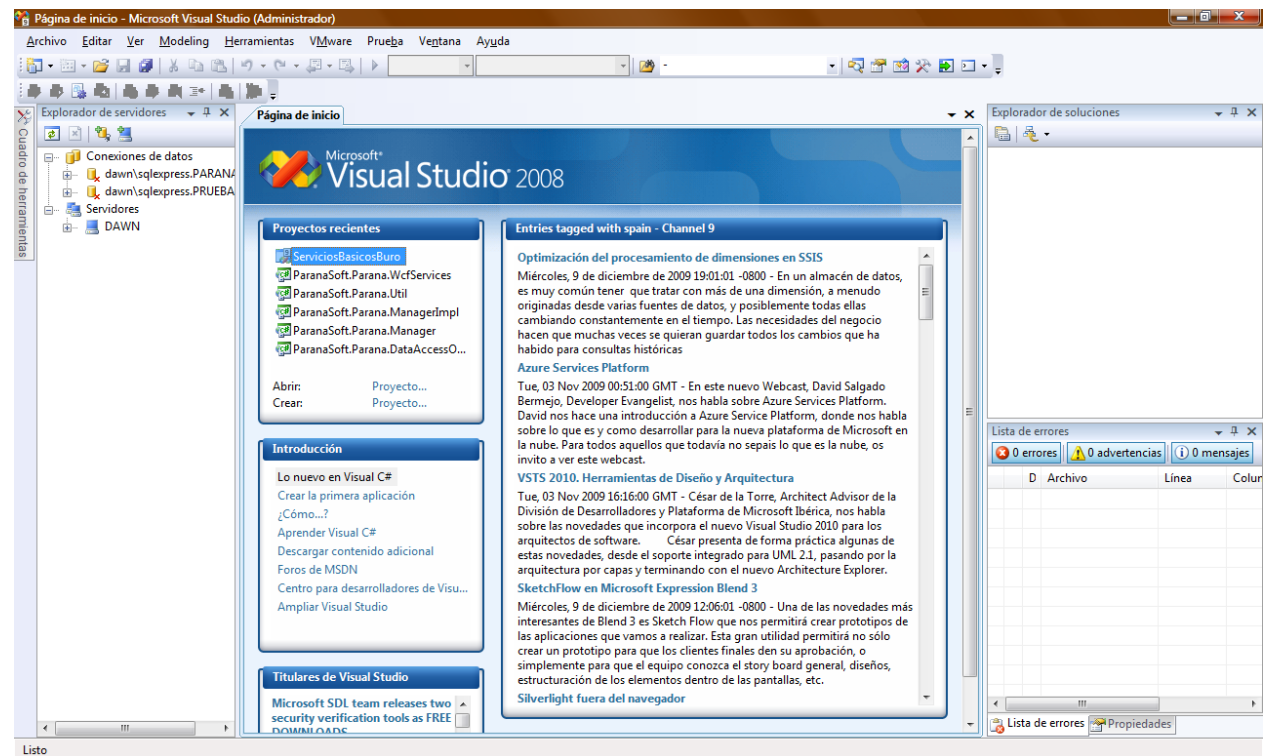
Prerrequisitos

5. *Instalar Visual Studio 2008*
6. *Instalar el y configurar el Núcleo de Procesamiento de Paraná*
7. *Instalar Bea Workspace Studio en versión 1.x*
8. *Configurar Bea Worspace Studio para que tenga acceso al Servidor en*

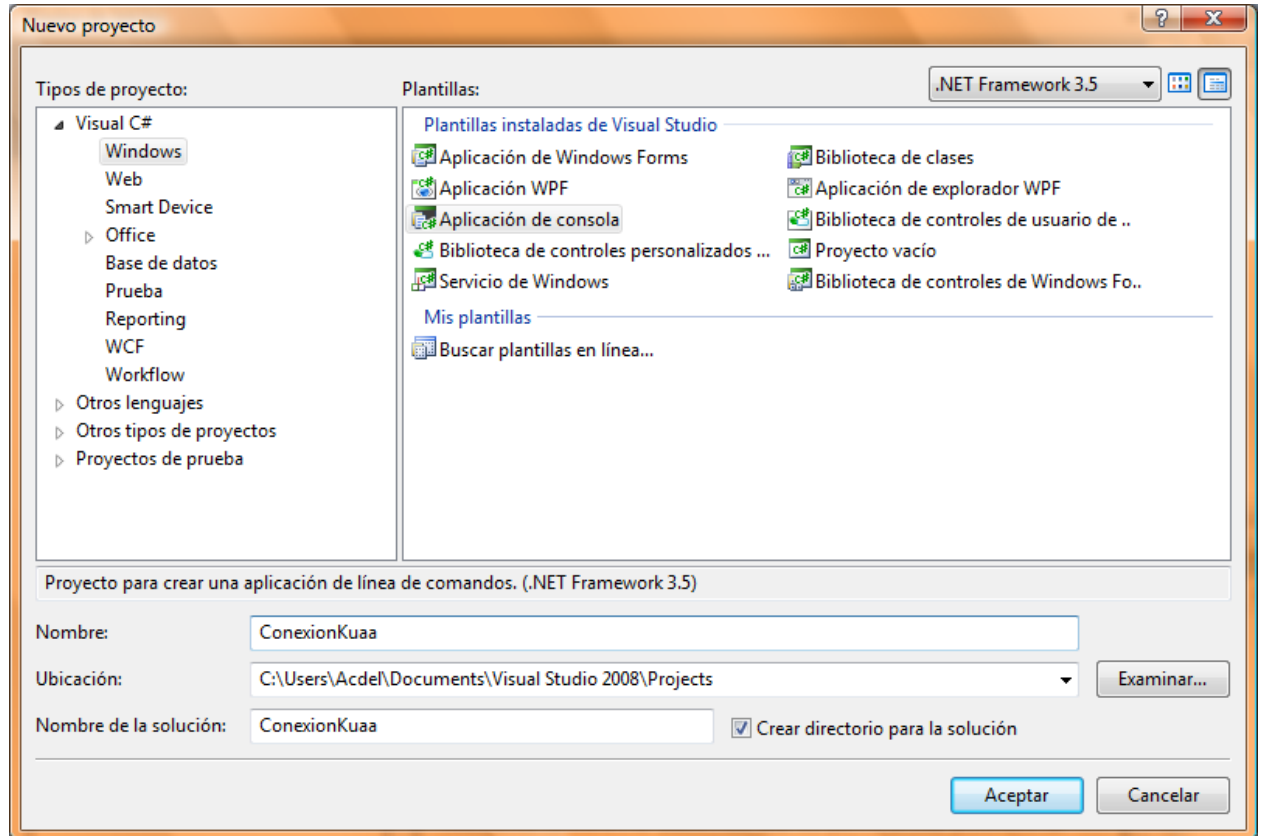
donde se encuentre instalado el ALSB.

Procedimiento

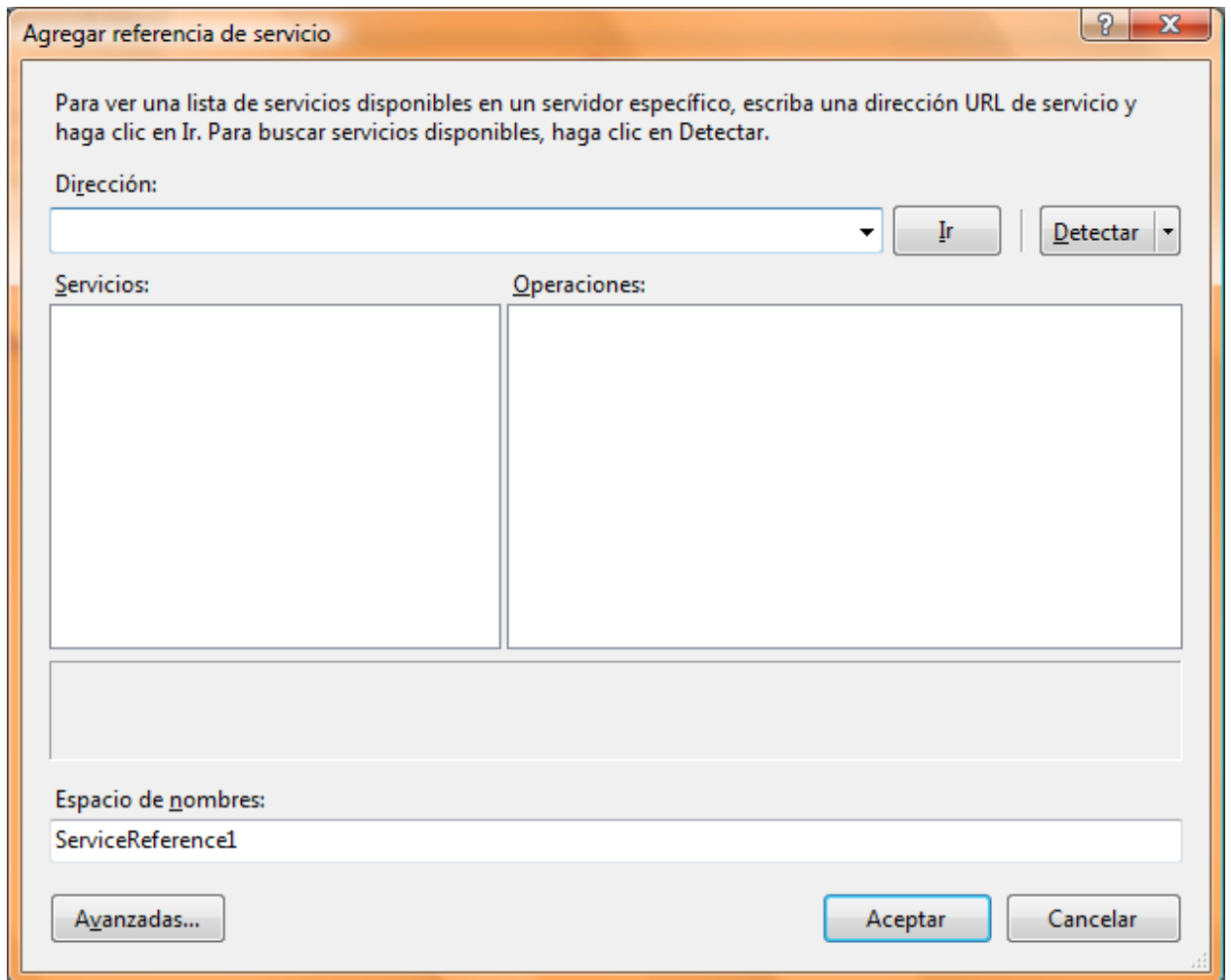
1. Abrir Visual Studio 2008 como administrador según se muestra en la gráfica



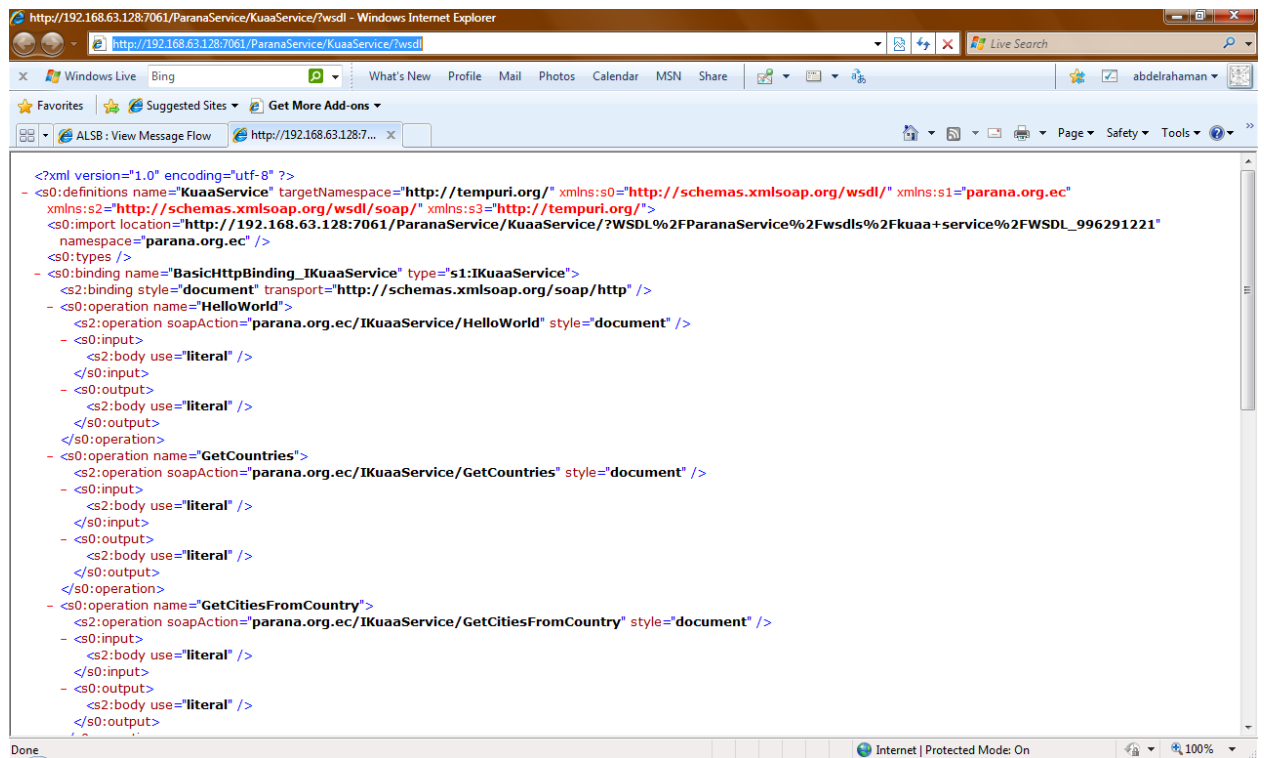
2. Crear un Nuevo proyecto de tipo consola compilado bajo cualquier Framework, ya que la plataforma resulta indiferente, en la presente se empleará la versión 3.5 del Framework, siendo que para la presente prueba se realizará una conexión con el componente Kuaa del núcleo, la aplicación llevará el nombre de ConxionKuaa.



3. A continuación al proyecto generado por defecto se le deberá agregar una nueva referencia web, oprimiendo un clic sobre la opción referencias y referencias web después, a continuación en pantalla se deberá desplegar la pantalla que se muestra

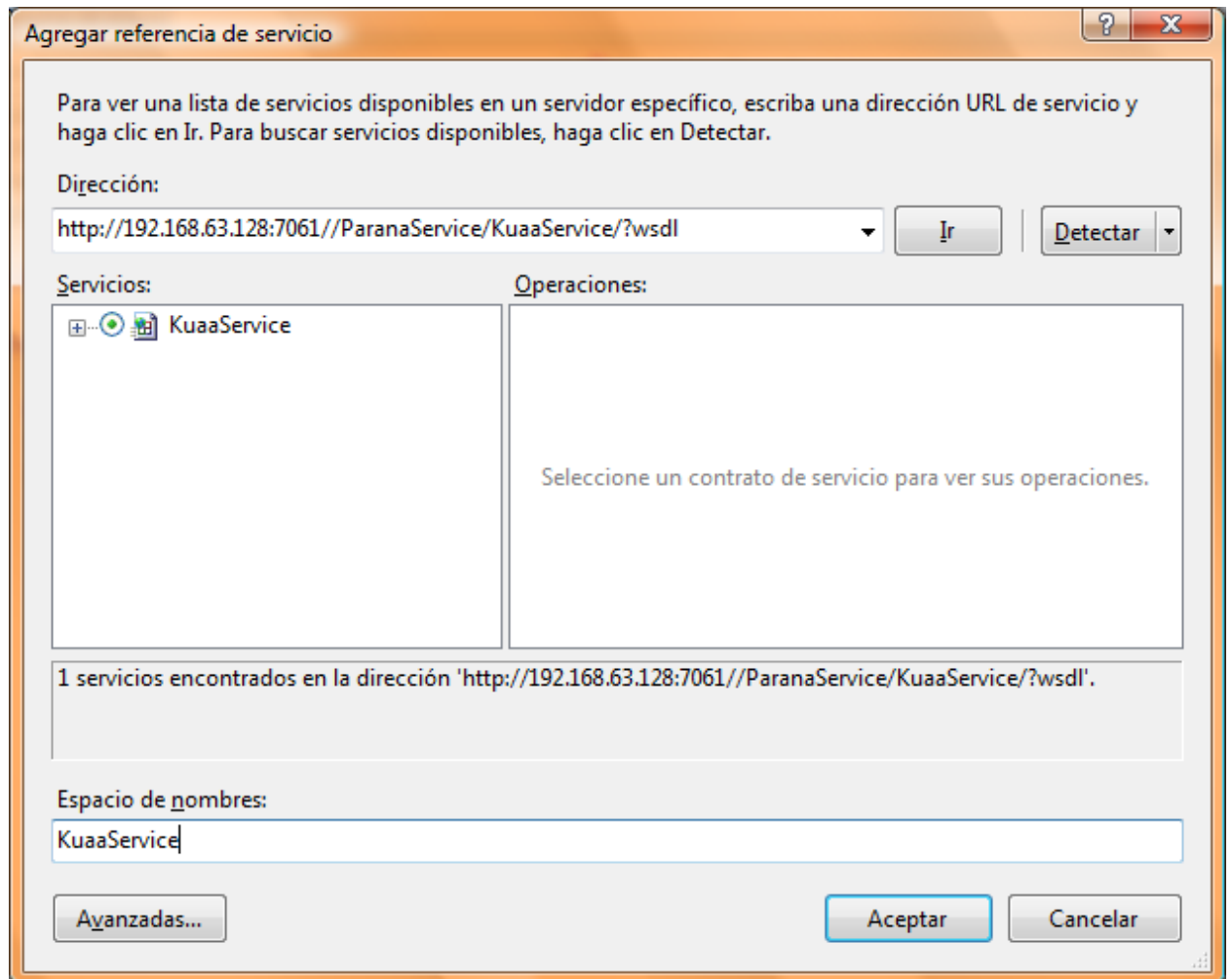


4. Se debe verificar el acceso al Componente, esto se debe de realizar a través de cualquier browser, accediendo al wsdl, si el resultado de digitar la URL que para el presente caso es: <http://192.168.63.128:7061/ParanaService/KuaaService/?wsdl>, señala el contenido que se muestra, entonces no existe ningún inconveniente ni impedimento para generar la interfaz desde el presente equipo, caso contrario deberá verificar problemas de conexión.

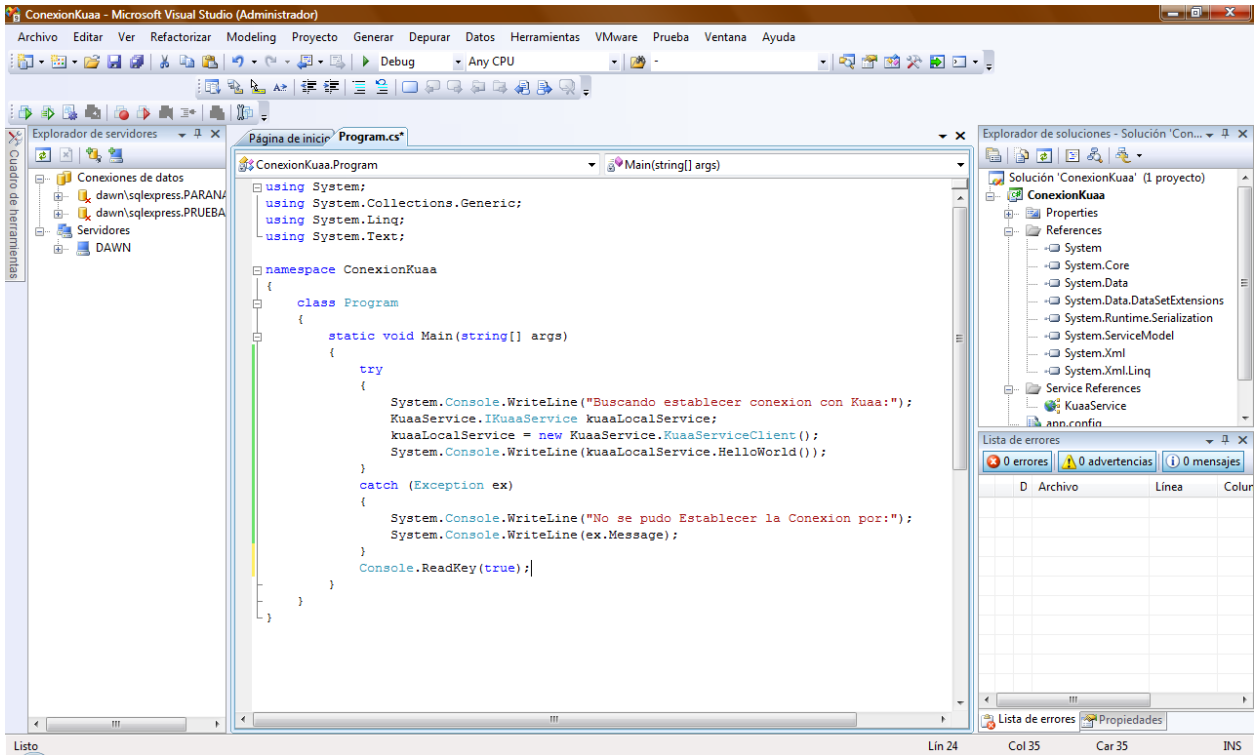


```
<?xml version="1.0" encoding="utf-8" ?>
- <s0:definitions name="KuaaService" targetNamespace="http://tempuri.org/" xmlns:s0="http://schemas.xmlsoap.org/wsdl/" xmlns:s1="parana.org.ec"
  xmlns:s2="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:s3="http://tempuri.org/"
  <s0:import location="http://192.168.63.128:7061/ParanaService/KuaaService/?WSDL%2FParanaService%2Fwsdl%2FKuaa+service%2FWSDL_996291221"
    namespace="parana.org.ec" />
  <s0:types />
- <s0:binding name="BasicHttpBinding_IKuaaService" type="s1:IKuaaService">
  <s2:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
- <s0:operation name="HelloWorld">
  <s2:operation soapAction="parana.org.ec/IKuaaService/HelloWorld" style="document" />
- <s0:input>
  <s2:body use="literal" />
</s0:input>
- <s0:output>
  <s2:body use="literal" />
</s0:output>
</s0:operation>
- <s0:operation name="GetCountries">
  <s2:operation soapAction="parana.org.ec/IKuaaService/GetCountries" style="document" />
- <s0:input>
  <s2:body use="literal" />
</s0:input>
- <s0:output>
  <s2:body use="literal" />
</s0:output>
</s0:operation>
- <s0:operation name="GetCitiesFromCountry">
  <s2:operation soapAction="parana.org.ec/IKuaaService/GetCitiesFromCountry" style="document" />
- <s0:input>
  <s2:body use="literal" />
</s0:input>
- <s0:output>
  <s2:body use="literal" />
</s0:output>
```

5. Se debe completar la pantalla en Visual Studio con los datos que se muestran en el gráfico, los mismos que señalan la dirección de acceso del Componente, de manera adicional se debe de observar el nombre al final, KuaaService, que será el que usará finalmente la referencia. Al final debe de hacer clic en Aceptar.



6. Se debe de adicionar en la clase Program.cs en la función Main el código necesario para la invocación hasta que la clase se vea como consta en la siguiente imagen.



Para una mayor apreciación el código colocado en la clase Program, en la función Main, para la creación de la interfaz es el siguiente:

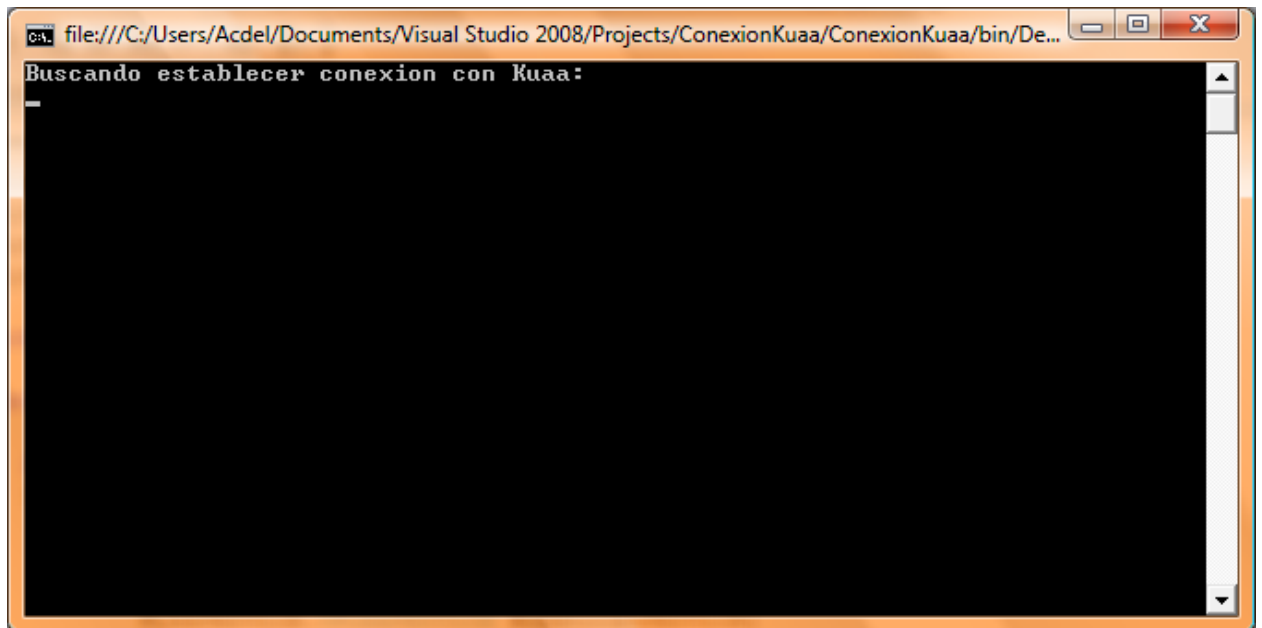
```

static void Main(string[] args)
{
    try
    {
        System.Console.WriteLine("Buscando establecer conexion con Kuaa:");
        KuaaService.IKuaaService kuaaLocalService;
        kuaaLocalService = new KuaaService.KuaaServiceClient();
        System.Console.WriteLine(kuaaLocalService.HelloWorld());
        Console.ReadKey(true);
    }
    catch (Exception ex)
    {
        System.Console.WriteLine("No se pudo Establecer la Conexion por:");
        System.Console.WriteLine(ex.Message);
        Console.ReadKey(true);
    }
}
}

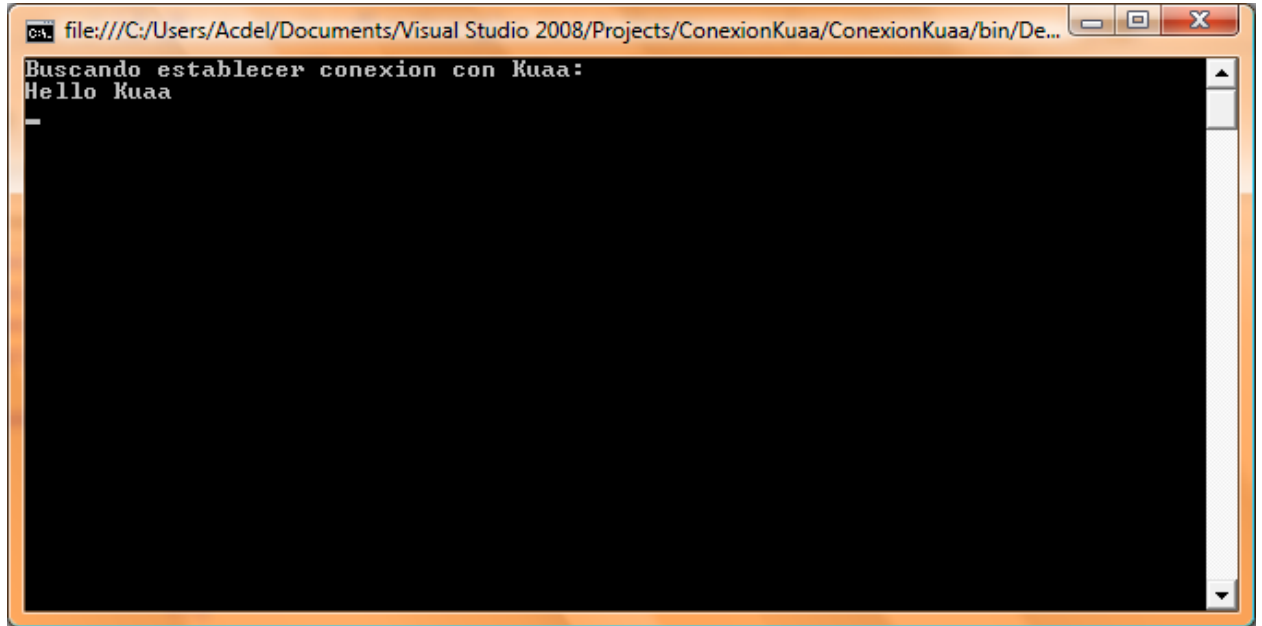
```

El presente código permitirá el utilizar la opción del componente HelloWorld diseñada para pruebas de conectividad, la misma que deberá devolver la leyenda "Hello Kuaa", en caso de existir conectividad, caso contrario se otendrá una excepción la misma que es contralada y además muestra el mensaje proveniente del error.

7. Se debe de ejecutar el aplicativo el mismo que de manera inicial intentará establecer la conexión con el servidor Weblogic que contiene el despliegue de los componentes como se puede observar:



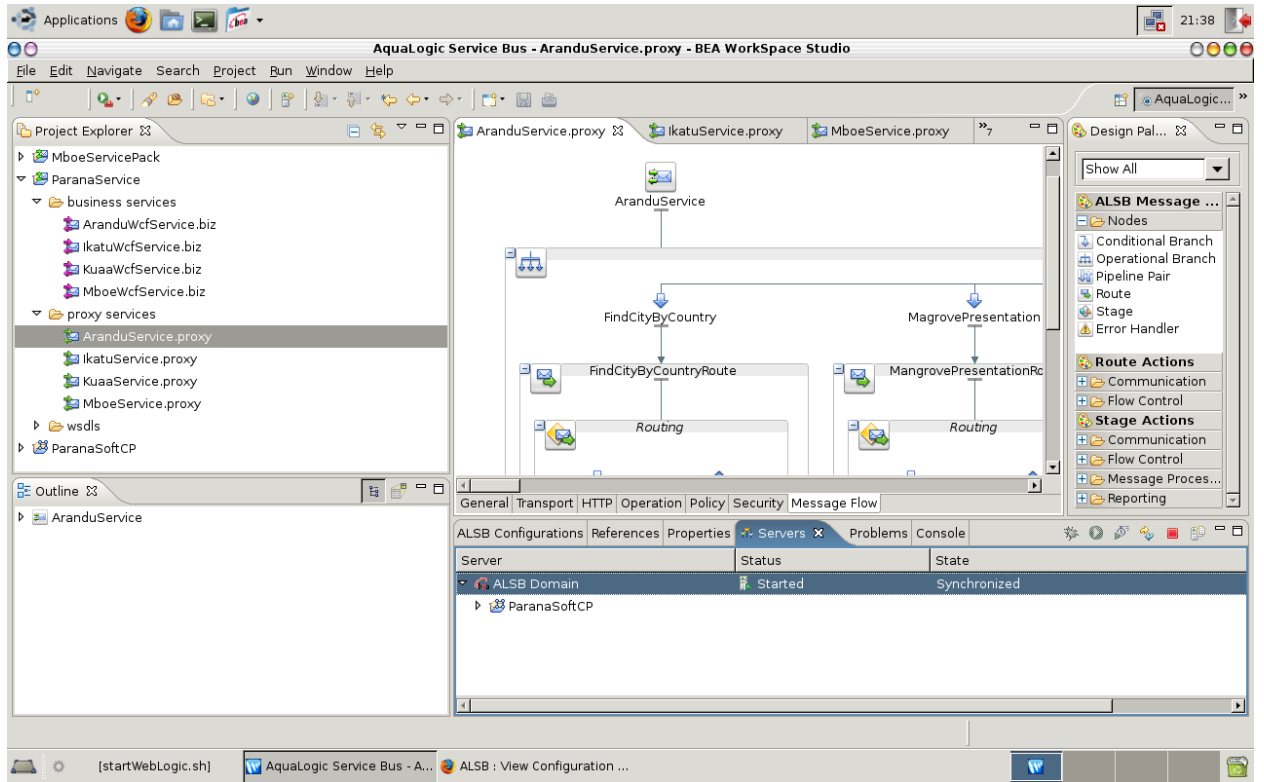
8. Luego de unos breves segundos con la variación del desempeño propio del aplicativo, se obtendrá la respuesta deseada como se expone en la pantalla siguiente:



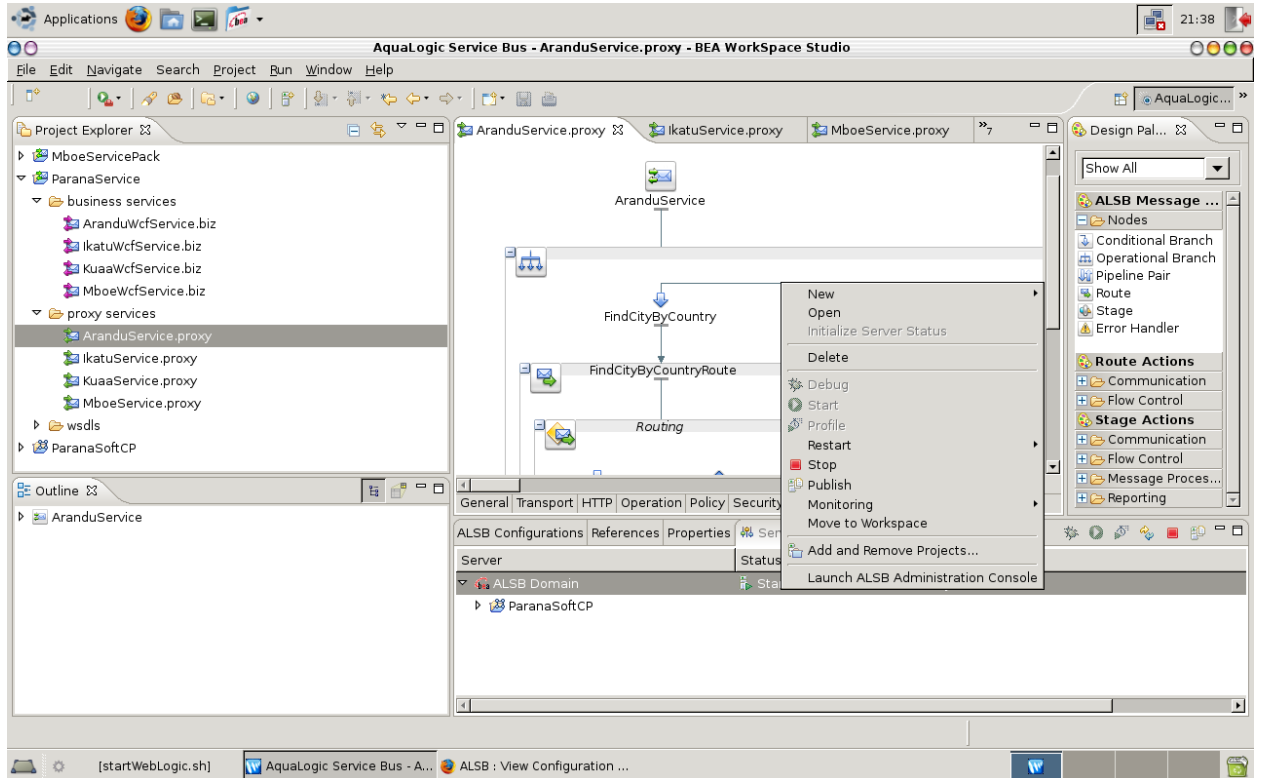
```
file:///C:/Users/Acdel/Documents/Visual Studio 2008/Projects/ConexionKuaa/ConexionKuaa/bin/De...
Buscando establecer conexion con Kuaa:
Hello Kuaa
```

Comprobación

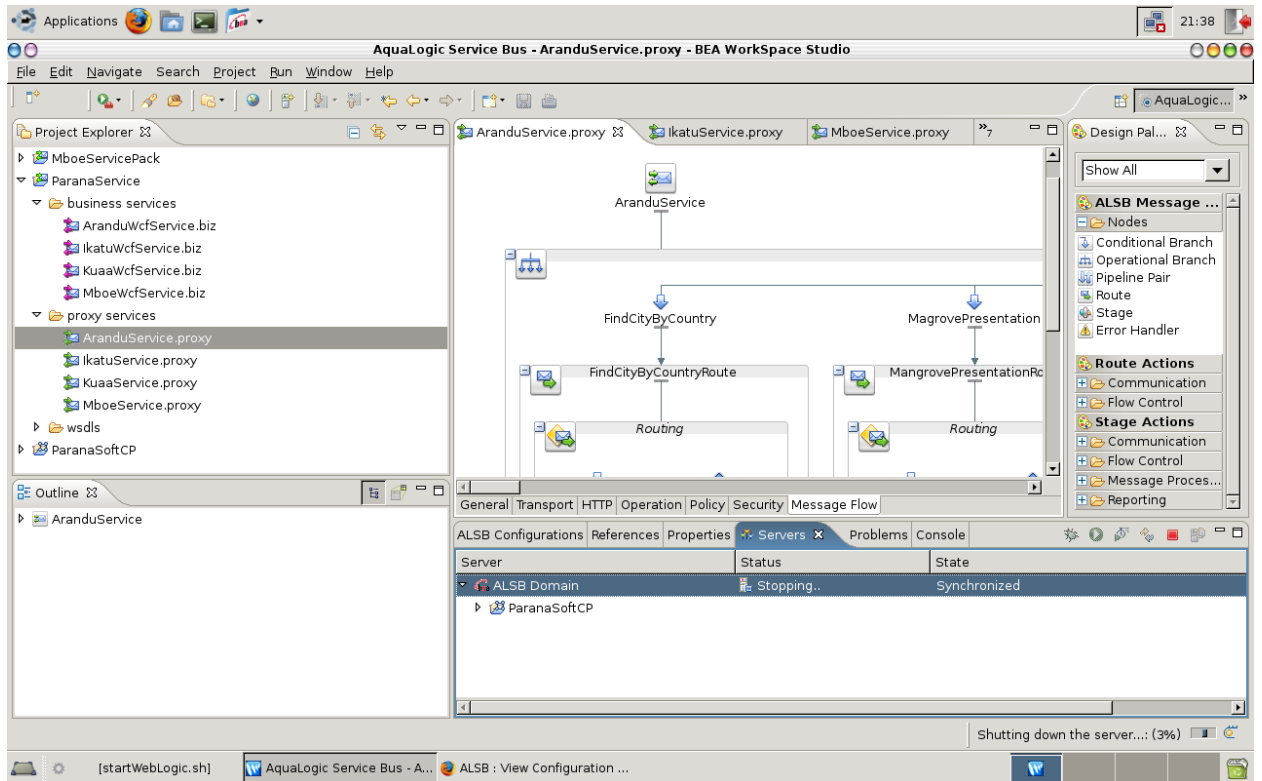
9. Para verificar que efectivamente la respuesta aquí presentada provenga del ALSB se debe proceder a detener el servidor de Aqualogic, esto se puede hacer bajo diferentes métodos, pero para la presente demostración se lo hará a través de la consola de Bea WorkStudio, por lo tanto se debe de abrir el mismo como se muestra en el gráfico (notar que este procedimiento se realiza sobre el servidor linux):



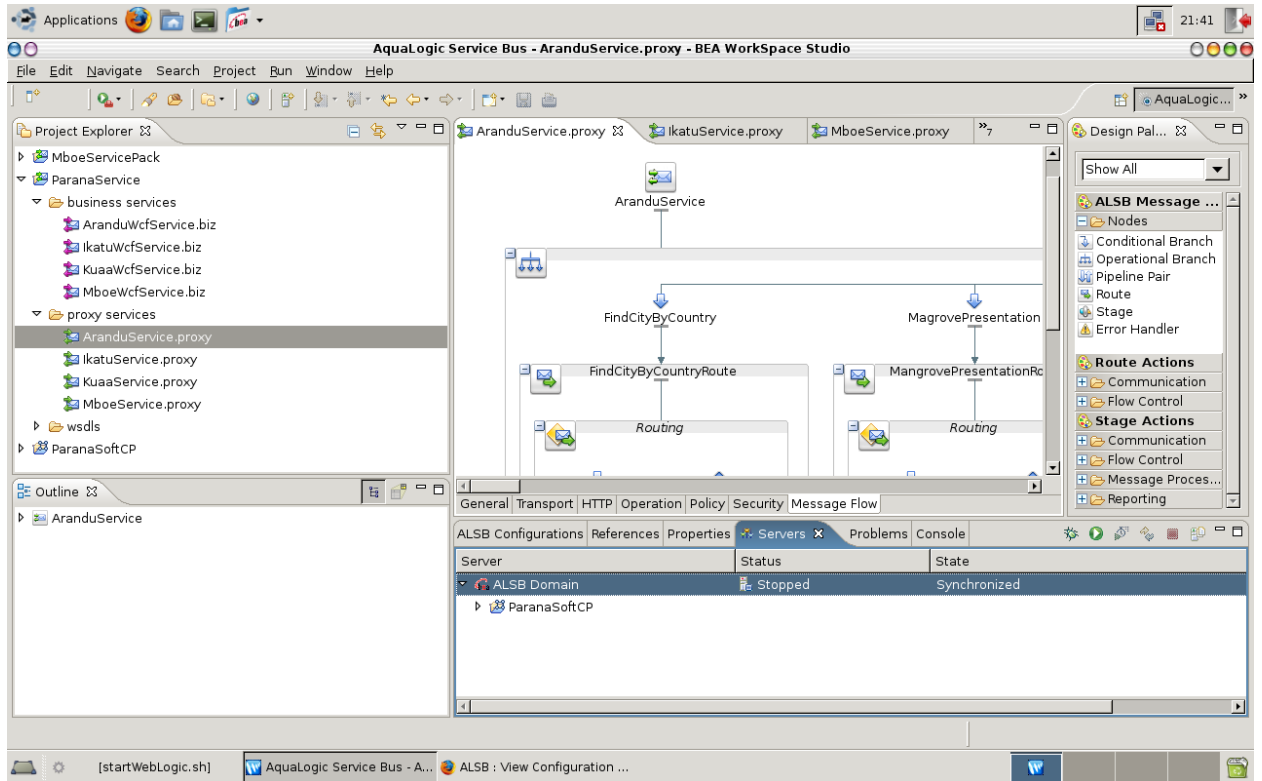
10. En la viñeta de “Servers”, se debe desplegar el menú contextual del servidor deseado, en el constan todas las posibles acciones a realizar sobre el servidor, entre ellas el detenerlo



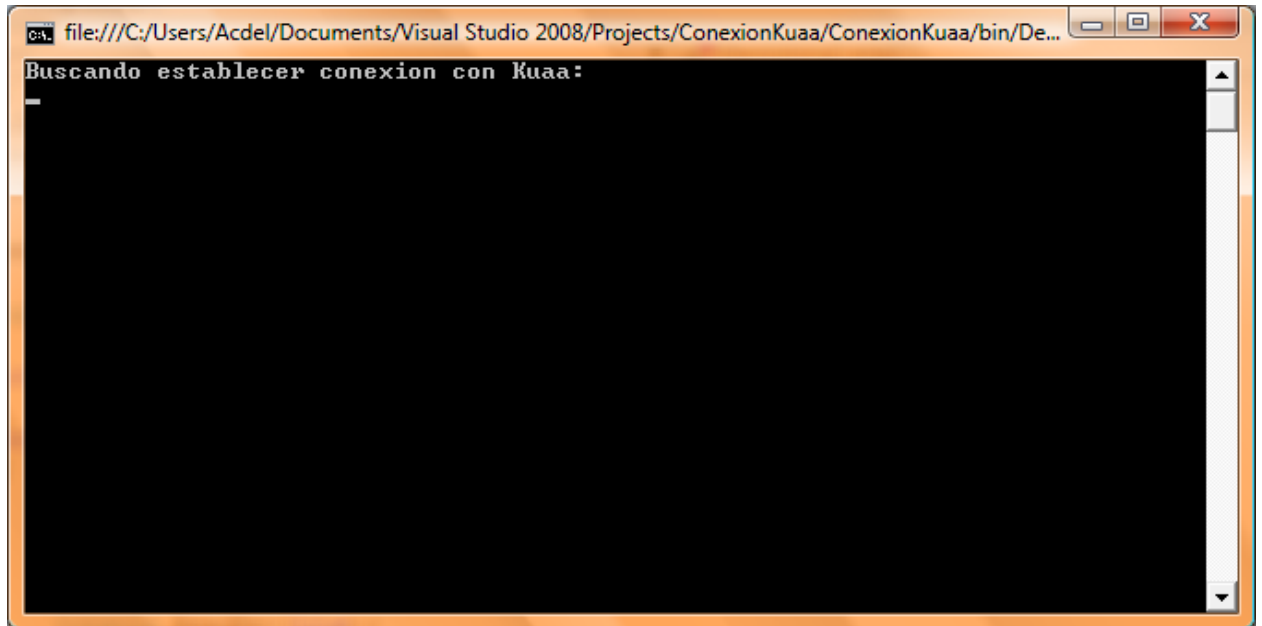
11. Se debe de seleccionar la opción para detener el servidor, hecho lo cual Bea WorkStudio debe de mostrarse según el presente gráfico:



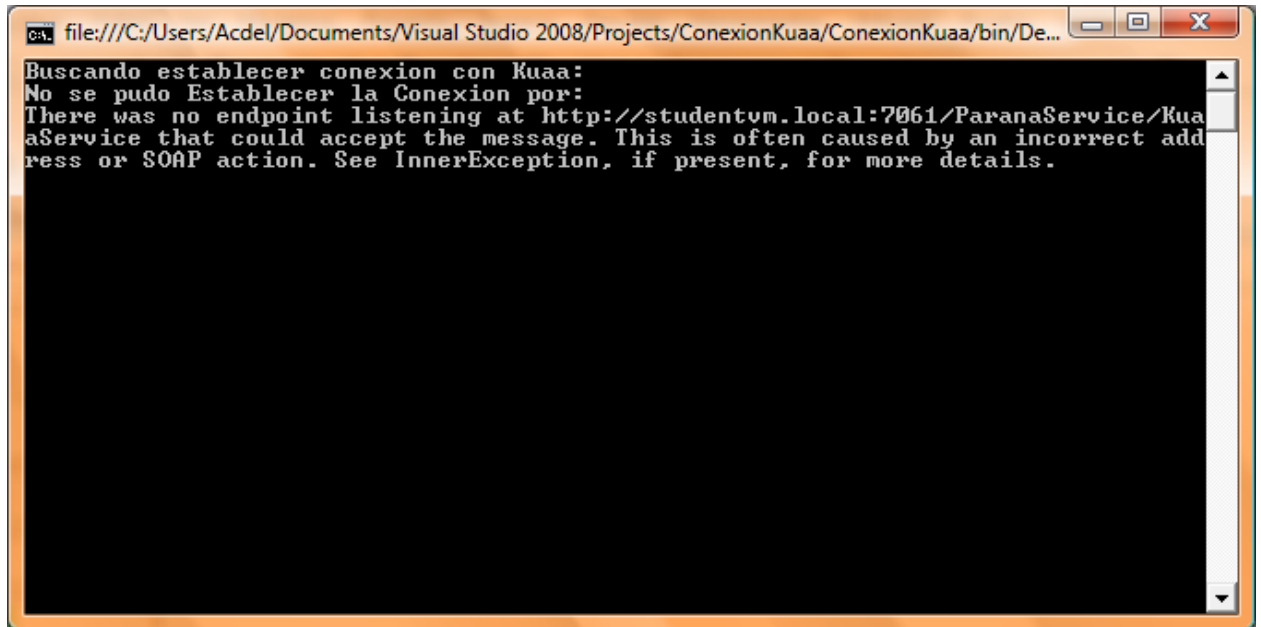
12. Luego de unos instantes el servidor finalmente se detendrá por si solo mostrando su estatus en la misma viñeta en la que unos momentos antes señalaba su proceso de para , según se puede ver en la siguiente gráfica:



13. Una vez más se levanta el proyecto de evaluación, que comenzará a rastrear al Componente como se muestra en la pantalla, operación que no tendrá éxito.



14. Luego de unos instantes en pantalla se mostrará en lugar de la leyenda esperada un mensaje de error ocasionado por la conexión fallida de la aplicación y el componente, comprobando de esta manera la colaboración existente entre la presente aplicación de prueba y el ALSB, tal como se muestra en la pantalla.



```
file:///C:/Users/Acdel/Documents/Visual Studio 2008/Projects/ConexionKuaa/ConexionKuaa/bin/De...
Buscando establecer conexion con Kuaa:
No se pudo Establecer la Conexion por:
There was no endpoint listening at http://studentvm.local:7061/ParanaService/Kua
aService that could accept the message. This is often caused by an incorrect add
ress or SOAP action. See InnerException, if present, for more details.
```

ANEXO E: WSDLs de Salida de Proxy Services

AranduService

```
URL: http://192.168.63.128:7061/ParanaService/AranduService/?wsdl
<?xml version="1.0" encoding="utf 8" ?>
  <s0:definitions name="AranduService"
targetNamespace="http://tempuri.org/"
xmlns:s0="http://schemas.xmlsoap.org/wsdl/" xmlns:s1="parana.org.ec"
xmlns:s2="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s3="http://tempuri.org/">
    <s0:import
location="http://192.168.63.128:7061/ParanaService/AranduService/?WSDL%2FPa
ranaService%2Fwsdls%2Farandu+service%2FWSDL_ 1906947983"
namespace="parana.org.ec" />
    <s0:types />
    <s0:binding name="BasicHttpBinding_IAranduService"
type="s1:IAranduService">
        <s2:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
        <s0:operation name="InsertNewUser">
            <s2:operation soapAction="parana.org.ec/IAranduService/InsertNewUser"
style="document" />
                <s0:input>
                    <s2:body use="literal" />
                </s0:input>
                <s0:output>
                    <s2:body use="literal" />
                </s0:output>
            </s0:operation>
            <s0:operation name="LookForUser">
                <s2:operation soapAction="parana.org.ec/IAranduService/LookForUser"
style="document" />
                    <s0:input>
                        <s2:body use="literal" />
                    </s0:input>
                    <s0:output>
                        <s2:body use="literal" />
                    </s0:output>
                </s0:operation>
            <s0:operation name="ListUserByMangrove">
                <s2:operation
soapAction="parana.org.ec/IAranduService/ListUserByMangrove"
style="document" />
                    <s0:input>
                        <s2:body use="literal" />
                    </s0:input>
                    <s0:output>
                        <s2:body use="literal" />
                    </s0:output>
                </s0:operation>
            <s0:operation name="FindByName">
                <s2:operation soapAction="parana.org.ec/IAranduService/FindByName"
style="document" />
                    <s0:input>
```

```

        <s2:body use="literal" />
    </s0:input>
    <s0:output>
        <s2:body use="literal" />
    </s0:output>
</s0:operation>
    <s0:operation name="UpdateUser">
        <s2:operation soapAction="parana.org.ec/IAranduService/UpdateUser"
style="document" />
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />
        </s0:output>
    </s0:operation>
    <s0:operation name="GetUserMangrove">
        <s2:operation
soapAction="parana.org.ec/IAranduService/GetUserMangrove" style="document"
/>
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />
        </s0:output>
    </s0:operation>
    <s0:operation name="GetZone">
        <s2:operation soapAction="parana.org.ec/IAranduService/GetZone"
style="document" />
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />
        </s0:output>
    </s0:operation>
    <s0:operation name="MagrovePresentation">
        <s2:operation
soapAction="parana.org.ec/IAranduService/MagrovePresentation"
style="document" />
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />
        </s0:output>
    </s0:operation>
    <s0:operation name="ListMangroveByName">
        <s2:operation
soapAction="parana.org.ec/IAranduService/ListMangroveByName"
style="document" />
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />

```

```

        </s0:output>
    </s0:operation>
    <s0:operation name="ListMangroveByAverage">
        <s2:operation
soapAction="parana.org.ec/IAranduService/ListMangroveByAverage"
style="document" />
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />
        </s0:output>
    </s0:operation>
    <s0:operation name="InsertNewMangrove">
        <s2:operation
soapAction="parana.org.ec/IAranduService/InsertNewMangrove"
style="document" />
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />
        </s0:output>
    </s0:operation>
    <s0:operation name="ConsumptionInformation">
        <s2:operation
soapAction="parana.org.ec/IAranduService/ConsumptionInformation"
style="document" />
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />
        </s0:output>
    </s0:operation>
    <s0:operation name="FindCityByName">
        <s2:operation
soapAction="parana.org.ec/IAranduService/FindCityByName" style="document"
/>
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />
        </s0:output>
    </s0:operation>
    <s0:operation name="GetCity">
        <s2:operation soapAction="parana.org.ec/IAranduService/GetCity"
style="document" />
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />
        </s0:output>
    </s0:operation>
    <s0:operation name="FindCityByCountry">

```

```

    <s2:operation
soapAction="parana.org.ec/IAranduService/FindCityByCountry"
style="document" />
    <s0:input>
    <s2:body use="literal" />
    </s0:input>
    <s0:output>
    <s2:body use="literal" />
    </s0:output>
</s0:operation>
    <s0:operation name="FindCountryByName">
    <s2:operation
soapAction="parana.org.ec/IAranduService/FindCountryByName"
style="document" />
    <s0:input>
    <s2:body use="literal" />
    </s0:input>
    <s0:output>
    <s2:body use="literal" />
    </s0:output>
</s0:operation>
    <s0:operation name="GetCountry">
    <s2:operation soapAction="parana.org.ec/IAranduService/GetCountry"
style="document" />
    <s0:input>
    <s2:body use="literal" />
    </s0:input>
    <s0:output>
    <s2:body use="literal" />
    </s0:output>
</s0:operation>
    <s0:operation name="GetCountries">
    <s2:operation soapAction="parana.org.ec/IAranduService/GetCountries"
style="document" />
    <s0:input>
    <s2:body use="literal" />
    </s0:input>
    <s0:output>
    <s2:body use="literal" />
    </s0:output>
</s0:operation>
    <s0:operation name="SetUserConsumption">
    <s2:operation
soapAction="parana.org.ec/IAranduService/SetUserConsumption"
style="document" />
    <s0:input>
    <s2:body use="literal" />
    </s0:input>
    <s0:output>
    <s2:body use="literal" />
    </s0:output>
</s0:operation>
    <s0:operation name="GetZoneGravity">
    <s2:operation
soapAction="parana.org.ec/IAranduService/GetZoneGravity" style="document"
/>
    <s0:input>

```

```

        <s2:body use="literal" />
    </s0:input>
    <s0:output>
        <s2:body use="literal" />
    </s0:output>
</s0:operation>
</s0:binding>
<s0:service name="AranduService">
    <s0:port binding="s3:BasicHttpBinding_IAranduService"
name="BasicHttpBinding_IAranduService">
        <s2:address
location="http://studentvm.local:7061/ParanaService/AranduService" />
    </s0:port>
</s0:service>
</s0:definitions>

```

IkatuService

URL:

http://192.168.63.128:7061//ParanaService/proxy_services/IkatuService/?wsdl

```

<?xml version="1.0" encoding="utf 8" ?>
    <s0:definitions name="IkatuService" targetNamespace="http://tempuri.org/"
xmlns:s0="http://schemas.xmlsoap.org/wsdl/" xmlns:s1="parana.org.ec"
xmlns:s2="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s3="http://tempuri.org/">
        <s0:import
location="http://192.168.63.128:7061//ParanaService/proxy_services/IkatuSer
vice/?WSDL%2FParanaService%2Fwsdls%2Fikatu+service%2FWSDL_1346332800"
namespace="parana.org.ec" />
        <s0:types />
        <s0:binding name="BasicHttpBinding_IIkatuService"
type="s1:IIkatuService">
            <s2:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
            <s0:operation name="HelloWorld">
                <s2:operation soapAction="parana.org.ec/IIkatuService/HelloWorld"
style="document" />
                <s0:input>
                    <s2:body use="literal" />
                </s0:input>
                <s0:output>
                    <s2:body use="literal" />
                </s0:output>
            </s0:operation>
            <s0:operation name="GetUserScore">
                <s2:operation soapAction="parana.org.ec/IIkatuService/GetUserScore"
style="document" />
                <s0:input>
                    <s2:body use="literal" />
                </s0:input>
                <s0:output>
                    <s2:body use="literal" />
                </s0:output>
            </s0:operation>
            <s0:operation name="GetUserName">

```

```

        <s2:operation soapAction="parana.org.ec/IIkatuService/GetUserName"
style="document" />
        <s0:input>
        <s2:body use="literal" />
        </s0:input>
        <s0:output>
        <s2:body use="literal" />
        </s0:output>
    </s0:operation>
    <s0:operation name="GetUser">
    <s2:operation soapAction="parana.org.ec/IIkatuService/GetUser"
style="document" />
    <s0:input>
    <s2:body use="literal" />
    </s0:input>
    <s0:output>
    <s2:body use="literal" />
    </s0:output>
    </s0:operation>
    <s0:operation name="SetUserConsumption">
    <s2:operation
soapAction="parana.org.ec/IIkatuService/SetUserConsumption"
style="document" />
    <s0:input>
    <s2:body use="literal" />
    </s0:input>
    <s0:output>
    <s2:body use="literal" />
    </s0:output>
    </s0:operation>
    <s0:operation name="GetZoneGravity">
    <s2:operation soapAction="parana.org.ec/IIkatuService/GetZoneGravity"
style="document" />
    <s0:input>
    <s2:body use="literal" />
    </s0:input>
    <s0:output>
    <s2:body use="literal" />
    </s0:output>
    </s0:operation>
</s0:binding>
<s0:service name="IkatuService">
    <s0:port binding="s3:BasicHttpBinding_IIkatuService"
name="BasicHttpBinding_IIkatuService">
    <s2:address
location="http://studentvm.local:7061/ParanaService/proxy_services/IkatuSer
vice" />
    </s0:port>
</s0:service>
</s0:definitions>

```

KuaaService

```
URL: http://192.168.63.128:7061/ParanaService/KuaaService/?wsdl
<?xml version="1.0" encoding="utf 8" ?>
  <s0:definitions name="KuaaService" targetNamespace="http://tempuri.org/"
xmlns:s0="http://schemas.xmlsoap.org/wsdl/" xmlns:s1="parana.org.ec"
xmlns:s2="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s3="http://tempuri.org/">
  <s0:import
location="http://192.168.63.128:7061/ParanaService/KuaaService/?WSDL%2FPara
naService%2Fwsdls%2Fkuaa+service%2FWSDL_996291221"
namespace="parana.org.ec" />
  <s0:types />
  <s0:binding name="BasicHttpBinding_IKuaaService"
type="s1:IKuaaService">
  <s2:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
  <s0:operation name="HelloWorld">
  <s2:operation soapAction="parana.org.ec/IKuaaService/HelloWorld"
style="document" />
  <s0:input>
  <s2:body use="literal" />
  </s0:input>
  <s0:output>
  <s2:body use="literal" />
  </s0:output>
  </s0:operation>
  <s0:operation name="GetCountries">
  <s2:operation soapAction="parana.org.ec/IKuaaService/GetCountries"
style="document" />
  <s0:input>
  <s2:body use="literal" />
  </s0:input>
  <s0:output>
  <s2:body use="literal" />
  </s0:output>
  </s0:operation>
  <s0:operation name="GetCitiesFromCountry">
  <s2:operation
soapAction="parana.org.ec/IKuaaService/GetCitiesFromCountry"
style="document" />
  <s0:input>
  <s2:body use="literal" />
  </s0:input>
  <s0:output>
  <s2:body use="literal" />
  </s0:output>
  </s0:operation>
  <s0:operation name="GetUsersFromCity">
  <s2:operation
soapAction="parana.org.ec/IKuaaService/GetUsersFromCity" style="document"
/>
  <s0:input>
  <s2:body use="literal" />
  </s0:input>
  <s0:output>
```



```

        <s2:body use="literal" />
    </s0:output>
</s0:operation>
    <s0:operation name="GetUserConsumption">
        <s2:operation
soapAction="parana.org.ec/IKuaaService/GetUserConsumption" style="document"
/>
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />
        </s0:output>
    </s0:operation>
    <s0:operation name="SetResultingXmlDocument">
        <s2:operation
soapAction="parana.org.ec/IKuaaService/SetResultingXmlDocument"
style="document" />
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />
        </s0:output>
    </s0:operation>
    <s0:operation name="GetConsumptionsByType">
        <s2:operation
soapAction="parana.org.ec/IKuaaService/GetConsumptionsByType"
style="document" />
        <s0:input>
            <s2:body use="literal" />
        </s0:input>
        <s0:output>
            <s2:body use="literal" />
        </s0:output>
    </s0:operation>
</s0:binding>
    <s0:service name="KuaaService">
        <s0:port binding="s3:BasicHttpBinding_IKuaaService"
name="BasicHttpBinding_IKuaaService">
            <s2:address
location="http://studentvm.local:7061/ParanaService/KuaaService" />
        </s0:port>
    </s0:service>
</s0:definitions>

```

MboeService

URL:

```
http://192.168.63.128:7061//ParanaService/proxy_services/MboeService/?wsdl
<?xml version="1.0" encoding="utf 8" ?>
  <s0:definitions name="MboeService" targetNamespace="http://tempuri.org/"
xmlns:s0="http://schemas.xmlsoap.org/wsdl/" xmlns:s1="parana.org.ec"
xmlns:s2="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:s3="http://tempuri.org/">
  <s0:import
location="http://192.168.63.128:7061//ParanaService/proxy_services/MboeServ
ice/?WSDL%2FParanaService%2Fwsdls%2Fmboe+service%2FWSDL_1973001077"
namespace="parana.org.ec" />
  <s0:types />
  <s0:binding name="BasicHttpBinding_IMboeService"
type="s1:IMboeService">
  <s2:binding style="document"
transport="http://schemas.xmlsoap.org/soap/http" />
  <s0:operation name="ZonePoints">
  <s2:operation soapAction="parana.org.ec/IMboeService/ZonePoints"
style="document" />
  <s0:input>
  <s2:body use="literal" />
  </s0:input>
  <s0:output>
  <s2:body use="literal" />
  </s0:output>
  </s0:operation>
  <s0:operation name="AllZonesAverage">
  <s2:operation soapAction="parana.org.ec/IMboeService/AllZonesAverage"
style="document" />
  <s0:input>
  <s2:body use="literal" />
  </s0:input>
  <s0:output>
  <s2:body use="literal" />
  </s0:output>
  </s0:operation>
  <s0:operation name="ZoneUserPoints">
  <s2:operation soapAction="parana.org.ec/IMboeService/ZoneUserPoints"
style="document" />
  <s0:input>
  <s2:body use="literal" />
  </s0:input>
  <s0:output>
  <s2:body use="literal" />
  </s0:output>
  </s0:operation>
  </s0:binding>
  <s0:service name="MboeService">
  <s0:port binding="s3:BasicHttpBinding_IMboeService"
name="BasicHttpBinding_IMboeService">
  <s2:address
location="http://studentvm.local:7061/ParanaService/proxy_services/MboeServ
ice" />
  </s0:port>
```

```
</s0:service>
</s0:definitions>
```

ANEXO F: Establecimiento Archivos de Administración y Configuración

Archivo de configuración del Servidor Local:

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.web>
    <compilation debug="true" />
  </system.web>
  <!-- Al implementar el proyecto de la biblioteca de servicios, el
  contenido del archivo de configuración se debe agregar al archivo
  app.config del host. La configuración del sistema no admite archivos de
  configuración en las bibliotecas. -->
  <system.serviceModel>
    <services>
      <service
behaviorConfiguration="ParanaSoft.Parana.WcfServices.AranduServiceBehavior"
  name="ParanaSoft.Parana.WcfServices.AranduService">
        <endpoint address="" binding="basicHttpBinding"
contract="ParanaSoft.Parana.WcfServices.IAranduService">
          <identity>
            <dns value="DAWN" />
          </identity>
        </endpoint>
        <host>
          <baseAddresses>
            <add
baseAddress="http://192.168.63.1:8081/ParanaSoft/Parana/WcfServices/AranduS
ervice/" />
          </baseAddresses>
        </host>
      </service>
      <service
behaviorConfiguration="ParanaSoft.Parana.WcfServices.IkatuBehavior"
  name="ParanaSoft.Parana.WcfServices.IkatuService">
        <endpoint address="" binding="basicHttpBinding"
contract="ParanaSoft.Parana.WcfServices.IIkatuService">
          <identity>
            <dns value="DAWN" />
          </identity>
        </endpoint>
        <host>
          <baseAddresses>
            <add
baseAddress="http://192.168.63.1:8081/ParanaSoft/Parana/WcfServices/IkatuSe
rvice/" />
          </baseAddresses>
        </host>
      </service>
      <service
behaviorConfiguration="ParanaSoft.Parana.WcfServices.KuaaBehavior"
  name="ParanaSoft.Parana.WcfServices.KuaaService">
```

```

        <endpoint address="" binding="basicHttpBinding"
contract="ParanaSoft.Parana.WcfServices.IKuaaService">
        <identity>
            <dns value="DAWN" />
        </identity>
    </endpoint>
    <host>
        <baseAddresses>
            <add
baseAddress="http://192.168.63.1:8081/ParanaSoft/Parana/WcfServices/KuaaSer
vice/" />
            </baseAddresses>
        </host>
    </service>
</service>
behaviorConfiguration="ParanaSoft.Parana.WcfServices.MboeBehavior"
name="ParanaSoft.Parana.WcfServices.MboeService">
    <endpoint address="" binding="wsHttpBinding"
contract="ParanaSoft.Parana.WcfServices.IMboeService">
        <identity>
            <dns value="DAWN" />
        </identity>
    </endpoint>
    <host>
        <baseAddresses>
            <add
baseAddress="http://192.168.63.1:8081/ParanaSoft/Parana/WcfServices/MboeSer
vice/" />
            </baseAddresses>
        </host>
    </service>
</services>
<behaviors>
    <serviceBehaviors>
        <behavior
name="ParanaSoft.Parana.WcfServices.AranduServiceBehavior">
            <serviceMetadata httpGetEnabled="true" />
            <serviceDebug includeExceptionDetailInFaults="true" />
        </behavior>
        <behavior name="ParanaSoft.Parana.WcfServices.IkatuBehavior">
            <serviceMetadata httpGetEnabled="true" />
            <serviceDebug includeExceptionDetailInFaults="true" />
        </behavior>
        <behavior name="ParanaSoft.Parana.WcfServices.KuaaBehavior">
            <serviceMetadata httpGetEnabled="true" />
            <serviceDebug includeExceptionDetailInFaults="true" />
        </behavior>
        <behavior name="ParanaSoft.Parana.WcfServices.MboeBehavior">
            <serviceMetadata httpGetEnabled="true" />
            <serviceDebug includeExceptionDetailInFaults="true" />
        </behavior>
    </serviceBehaviors>
</behaviors>
</system.serviceModel>
</configuration>

```

Archivo de configuración de servicios WCF

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <system.web>
    <compilation debug="true" />
  </system.web>
  <!-- Al implementar el proyecto de la biblioteca de servicios, el
  contenido del archivo de configuración se debe agregar al archivo
  app.config del host. La configuración del sistema no admite archivos de
  configuración en las bibliotecas. -->
  <system.serviceModel>
    <services>
      <service
behaviorConfiguration="ParanaSoft.Parana.WcfServices.AranduServiceBehavior"
        name="ParanaSoft.Parana.WcfServices.AranduService">
        <endpoint address="" binding="basicHttpBinding"
contract="ParanaSoft.Parana.WcfServices.IAranduService">
          <identity>
            <dns value="DAWN" />
          </identity>
        </endpoint>
        <host>
          <baseAddresses>
            <add
baseAddress="http://192.168.63.1:8080/ParanaSoft/Parana/WcfServices/AranduS
ervice/" />
          </baseAddresses>
        </host>
      </service>
      <service
behaviorConfiguration="ParanaSoft.Parana.WcfServices.IkatuBehavior"
        name="ParanaSoft.Parana.WcfServices.IkatuService">
        <endpoint address="" binding="basicHttpBinding"
contract="ParanaSoft.Parana.WcfServices.IIkatuService">
          <identity>
            <dns value="DAWN" />
          </identity>
        </endpoint>
        <host>
          <baseAddresses>
            <add
baseAddress="http://192.168.63.1:8080/ParanaSoft/Parana/WcfServices/IkatuSe
rvice/" />
          </baseAddresses>
        </host>
      </service>
      <service
behaviorConfiguration="ParanaSoft.Parana.WcfServices.KuaaBehavior"
        name="ParanaSoft.Parana.WcfServices.KuaaService">
        <endpoint address="" binding="basicHttpBinding"
contract="ParanaSoft.Parana.WcfServices.IKuaaService">
          <identity>
            <dns value="DAWN" />
          </identity>
        </endpoint>
      </service>
    </services>
  </system.serviceModel>
</configuration>
```

```

        <host>
            <baseAddresses>
                <add
baseAddress="http://192.168.63.1:8080/ParanaSoft/Parana/WcfServices/KuaaSer
vice/" />
            </baseAddresses>
        </host>
    </service>
    <service
behaviorConfiguration="ParanaSoft.Parana.WcfServices.MboeBehavior"
    name="ParanaSoft.Parana.WcfServices.MboeService">
        <endpoint address="" binding="basicHttpBinding"
contract="ParanaSoft.Parana.WcfServices.IMboeService">
            <identity>
                <dns value="DAWN" />
            </identity>
        </endpoint>
        <host>
            <baseAddresses>
                <add
baseAddress="http://192.168.63.1:8080/ParanaSoft/Parana/WcfServices/MboeSer
vice/" />
            </baseAddresses>
        </host>
    </service>
</services>
<behaviors>
    <serviceBehaviors>
        <behavior
name="ParanaSoft.Parana.WcfServices.AranduServiceBehavior">
            <serviceMetadata httpGetEnabled="true" />
            <serviceDebug includeExceptionDetailInFaults="true" />
        </behavior>
        <behavior name="ParanaSoft.Parana.WcfServices.IkatuBehavior">
            <serviceMetadata httpGetEnabled="true" />
            <serviceDebug includeExceptionDetailInFaults="true" />
        </behavior>
        <behavior name="ParanaSoft.Parana.WcfServices.KuaaBehavior">
            <serviceMetadata httpGetEnabled="true" />
            <serviceDebug includeExceptionDetailInFaults="true" />
        </behavior>
        <behavior name="ParanaSoft.Parana.WcfServices.MboeBehavior">
            <serviceMetadata httpGetEnabled="true" />
            <serviceDebug includeExceptionDetailInFaults="true" />
        </behavior>
    </serviceBehaviors>
</behaviors>
</system.serviceModel>
</configuration>

```

Archivo de configuración de Conexión de LINQ

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <configSections>
  </configSections>
  <connectionStrings>
    <add
name="ParanaSoft.Parana.LinqObjects.Properties.Settings.PARANAConnectionStr
ing"
      connectionString="Data Source=DAWN\SQLEXPRESS;Initial
Catalog=PARANA;Persist Security Info=True;User ID=sa;Password=master"
      providerName="System.Data.SqlClient" />
  </connectionStrings>
</configuration>
```

GLOSARIO

JEE.- Java Enterprise Edition, edición empresarial del ya conocido framework de desarrollo, que contiene una serie de convenciones sobre mejores prácticas al momento de la implementación de soluciones de este tipo. Actualmente se encuentra en su versión 6, con implementaciones sobre Glassfish, aunque la versión comercialmente viable es la 5.

J2CA.- J2EE Connector architecture, es el patrón arquitectónico, componente de JEE, que busca el estandarizar la construcción de conectores entre diversas plataformas con JEE.

Modelo TCP.- Se consiste en la base misma del Internet, posibilita la interconexión de equipos computacionales en un modelo distribuido en capas.

Corp.- Siglas de corporación.

XP.- Siglas de Extreme Programming

UML.- Siglas de *Unified Modeling Language*, resulta en el lenguaje de modelo de software por excelencia, este comprende una gran variedad de elementos gráficos que permiten su fácil acoplamiento a las principales metodologías de diseño empleadas en la actualidad.

ALSB.- Siglas de Aqualogic Service Bus que para el presente trabajo se referira de manera exclusive a su versión 3

MFL.- Siglas de Message Format Language, es un lenguaje propietario de Bea ahora parte del suite de Oracle, empleado en el ALSB, el mismo es el responsable de la migración de data binario a data tipado

IDE.- Siglas de Integrated Development Enviroment, que resulta en software dedicado para la utilización de frameworks de Desarrollo de Software, en el presente proyecto la referencia se orienta al trato de Visual Studio, dedicado a trabajar con .Net Framework y a Eclipse, dedicado al trabajo con JEE.

SOA.- Siglas de Service Oriented Architecture, es un modelo arquitectónico que define el uso de servicios para satisfacer las necesidades de software empresarial.

WPF.- Siglas de Windows Presentation Foundation, es la base de lo que hoy se conoce como Aero y conforma gran parte del Nuevo API de interfaz de Microsoft que reemplaza a Win32.

LINQ.- Siglas de Language Integrated Query, es una característica integrada en el Framework de .Net a a partir de su versión 3, el que permite el insertar queries de tipo SQL directamente en el código de sus lenguajes administrados

VSF.- Siglas de Visual Studio Foundation, encarga del desarrollo de dicho IDE.

Just In Time.- Nombre que toma el proceso de interpretación del código en la Java Virtual Machine, también es el nombre del interprete de .Net Framework que traslada el c'dogio administrado a lenguaje intermedio.

ASP.- Siglas de Active Server Pages, es una tecnología propietaria de Microsoft Corp. La misma permite ejecutar páginas web de manera dinámica del lado del servidor.

HTTP.- Siglas de *HyperText Transfer Protocol*, que consiste en el protocolo de comunicación usado en cada transacción, siendo esencialmente un protocolo sin estado y sumamente liviano.

HTTPS.- Siglas de HTTP Secure, la versión encriptada del protocolo HTTP, ampliamente usado en la actualidad sobre la Web.

SDK.- Siglas de Software Development Kit, y viene a constituirse al conjunto de recursos provistos por un framework para el desarrollo de aplicaciones sobre el mismo.

SSO.- Siglas de Secure Socket Layer, es un protocolo de conexión segura que persigue la encriptación de la información.

CRUD.- Siglas de Create, Read, Update and Delete, el término por lo general es empleado para referirse al conjunto de cuatro operaciones básicas que se pueden realizar a través de SQL.

API.- Siglas de Application Development Interface, es un conjunto de recursos para la implementación de características específicas dentro de un paquete de software.

Point-to-Point.- Siglas de Punto a Punto, nombre que se le da al tipo de conexión por su característica intrínseca de construir un único canal entre los dos puntos a conectar.

EAI.- Siglas de Enterprise Application Integration, es el nombre que se le brinda a las iniciativas de conexión de aplicaciones empresariales que no resultan punto a punto dentro de una misma institución a través de paquetes de software, algunos sostienen que SOA es una implementación del presente paradigma.

EJB.- Siglas de Enterprise Java Bean, y se refiere a la implementación de lo redactado en JEE sobre la implementación de la provisión de objetos provenientes del Servidor.

RPC.- Siglas de *Remote Procedure Call*, es un protocolo que permite la ejecución remota de código proveniente de otra aplicación sin tener que preocuparse por la mantención de la conexión entre ambos

CORBA.- Siglas de *Common Object Request Broker Architecture*, que definen un modelo y conjunto de APIs para el desarrollo de aplicaciones distribuidas con acceso remoto a métodos de una manera estandarizada.

IT.- Siglas de Information Technologies, se refiere al conjunto de herramientas que se permiten el tratamiento y transporte de data.

BIBLIOGRAFIA

- **Stoecker**, Matthew A. Microsoft .NET Framework 3.5 – Windows Presentation Foundation (MCTS EXAM 70 - 502). Microsoft Press, 2008
- **Jhonson**, Bruce. Microsoft .NET Framework 3.5 – Windows Communication Foundation (MCTS EXAM 70 - 503). Microsoft Press, 2008
- **Steinberg**, Daniel H. Extreme Programming Engineering: Hands on Approach. Pearson Education, Inc, 2004
- **Barabási Albert-László**. Linked. First Plume Printing, Penguin Group. 2003
- **Hurwitz, Judith**. Service Oriented Architecture for Dummies. Wiley Publishing, Inc. 2006
- **Oracle University**. BEA Aqualogic Service Bus 3: Design & integrate Services for SOA (Student Guide). Oracle University. 2008
- **Oracle University**. BEA Aqualogic Service Bus 3: Design & integrate Services for SOA (Practice Guide). Oracle University. 2008
- **Oracle University**. Oracle BPEL Process Manager: Services Orchestration (Student Guide - Volume 1). Oracle University. 2008
- **Delgado Landazuri Franklin, otros**. Análisis, Diseño, construcción y puesta en marcha del sistema de información académica para la Unidad Educativa Liceo del Valle a través de la Web. Dpto. De Ciencias de la Computación, ESPE. 2008
- **WIKIMEDIA FOUNDATION, INC.**, Information Technology Infrastructure Library
- <http://es.wikipedia.org/> en Enero del 2010

- **Wells Done**, eXtreme programming
<http://www.extremeprogramming.org/rules.html> en 2009
- **Irrtated Vowel**, Estándares en .Net
<http://www.irritatedvowel.com/Programming/Standards.aspx> en a;o2009
- **MSDN**, consultas variadas sobre tecnología .net
[http://msdn.microsoft.com/en-us/library/yyaad03b\(VS.71\).aspx](http://msdn.microsoft.com/en-us/library/yyaad03b(VS.71).aspx) en año 2009
<http://social.msdn.microsoft.com/Forums/en-US/wcf/thread/90716451-fd29-4917-a5ca-f7f3bcaa96c9> en año 2009