

Instituto Tecnológico de Costa Rica

Escuela de Ingeniería en Computación

Northek Software S.A.

**“Prototipo de Implementación GIS-PDA. -
Sistema Integrado de Conectividad y Consultas
Municipalidad de San Carlos.”**

**Informe de Proyecto de Graduación para optar por el grado de
Bachiller en Ingeniería en Computación**

Víctor Andrés Garro Jarquín.

San Carlos, Noviembre 2007

Índices

1	DESCRIPCIÓN DEL PROBLEMA	1
1.1	ENUNCIADO DEL PROBLEMA	1
1.1.1.a	Prototipo GIS PDA:	1
1.1.1.b	Sistema Integrado de Consultas y Conectividad	2
1.2	ENUNCIADO DE LA SOLUCIÓN	4
1.2.1.a	Prototipo GIS-PDA	4
1.2.1.b	Sistema Integrado de Consultas y Conectividad	4
1.3	PERSPECTIVAS, SUPUESTOS Y DEPENDENCIAS DEL PROYECTO	7
1.3.1.a	Perspectivas:	7
1.3.1.b	Supuestos:	7
1.3.1.c	Dependencias:	8
2	OBJETIVOS Y ALCANCES	9
2.1	GENERAL	9
2.2	ESPECÍFICOS	9
2.3	ALCANCES	9
2.3.1	<i>Prototipo GIS-PDA</i>	9
2.3.2	<i>Sistema de Conectividad</i>	10
3	ESPECIFICACIÓN DE LOS SISTEMAS	11
3.1	REQUERIMIENTOS NO FUNCIONALES	11
3.1.1.a	Prototipo GIS-PDA:	11
3.1.1.b	Sistema de Conectividad:	11
3.2	CARACTERÍSTICAS GENERALES	12
3.2.1.a	Prototipo GIS – PDA	12
3.2.1.b	Sistema de Conectividad	12
4	METODOLOGÍA DE REQUERIMIENTOS, DESARROLLO Y PRUEBAS	12
4.1	LEAN PROCESS Y LEAN PROGRAMMING	12
4.2	AGILE DEVELOPMENT	13
4.3	XTREME PROGRAMMING	18
4.4	TEST DRIVEN DEVELOPMENT	19
5	SOLUCIÓN: PROTOTIPO DE IMPLEMENTACIÓN GIS-PDA	21
5.1	PROCESO DE DESARROLLO	21
5.1.1	<i>Análisis del Proyecto.</i>	21
5.1.1.a	Propuesta de realización:	21
5.1.1.b	Requerimientos:	22
5.2	MODELO DE DISEÑO - DIAGRAMA CONCEPTUAL	22
5.3	DIAGRAMA DE USER STORIES	24

5.4	USER STORIES	24
5.5	DIAGRAMAS DE SECUENCIA	26
5.6	MODELO DE DOMINIO	28
5.7	DIAGRAMA DE CLASES	29
5.8	IMPLEMENTACIÓN	31
5.8.1	<i>Investigación:</i>	31
5.8.1.a	Análisis Evaluativo.	31
5.8.1.b	Resultados	32
5.8.2	<i>Entregables</i>	32
5.8.2.a	Manual de Sincronización	32
5.8.2.b	Prototipo Desarrollado	33
6	SOLUCIÓN: SISTEMA INTEGRADO DE CONECTIVIDAD Y CONSULTAS MUNICIPALIDAD DE SAN CARLOS.	35
6.1	PROCESO DE DESARROLLO	35
6.2	ANÁLISIS DEL PROYECTO	36
6.2.1	<i>Requerimientos</i>	36
6.2.1.a	Features y User Stories	36
6.2.2	<i>Modelo de Diseño – Diagrama Conceptual</i>	38
6.2.3	<i>Diagrama User Stories</i>	43
6.2.4	<i>User Stories</i>	44
6.2.5	<i>Diagramas de Secuencia</i>	47
6.2.6	<i>Modelo de Dominio</i>	55
6.2.7	<i>Diagrama de Clases</i>	56
6.2.8	<i>Implementación</i>	60
6.2.8.a	Módulo de Sitio Web	60
6.2.8.b	Módulo Servicios Web	63
6.2.9	<i>Resultados:</i>	65
7	CONCLUSIONES	66
8	BIBLIOGRAFÍA	68

Índices de Tablas y Figuras

ILUSTRACIÓN 1 EJEMPLO INFORMACIÓN DE UN FEATURE	17
ILUSTRACIÓN 2 EJEMPLO UN STORY	17
ILUSTRACIÓN 3 EJEMPLO TASKS SOBRE UN STORY	17
ILUSTRACIÓN 4 DIAGRAMA GENERAL DE FUNCIONES E INTERRELACIONES	23
ILUSTRACIÓN 5 USER STORIES PARA EL USUARIO CON PDA.....	24
ILUSTRACIÓN 6 DIAGRAMA DE SECUENCIA USER STORY Nº 1.....	26
ILUSTRACIÓN 7 DIAGRAMA DE SECUENCIA USER STORY Nº 2.....	26
ILUSTRACIÓN 8 DIAGRAMA DE SECUENCIA USER STORY Nº 3.....	26
ILUSTRACIÓN 9 DIAGRAMA DE SECUENCIA USER STORY Nº 4.....	27
ILUSTRACIÓN 10 DIAGRAMA DE SECUENCIA USER STORY Nº 5.....	27
ILUSTRACIÓN 11 DIAGRAMA DE DOMINIO	28
ILUSTRACIÓN 12 DIAGRAMA DE CLASES	30
ILUSTRACIÓN 13 IMAGEN DE SINCRONIZACIÓN	34
ILUSTRACIÓN 14 MODELO CONCEPTUAL SITIO DE CONSULTAS EN LÍNEA	39
ILUSTRACIÓN 15 DIAGRAMA CONCEPTUAL SERVICIOS WEB	41
ILUSTRACIÓN 16 DIAGRAMA DE INTEGRACIÓN MÓDULOS.....	42
ILUSTRACIÓN 17 DIAGRAMA DE USER STORIES, MÓDULO SITIO DE CONSULTAS	43
ILUSTRACIÓN 18 DIAGRAMA USER STORIES, MÓDULO SERVICIOS WEB	43
ILUSTRACIÓN 19 DIAGRAMA DE SECUENCIA USER STORY Nº 1.....	47
ILUSTRACIÓN 20 DIAGRAMA DE SECUENCIA USER STORY Nº 2.....	47
ILUSTRACIÓN 21 DIAGRAMA DE SECUENCIA USER STORY Nº 3.....	48
ILUSTRACIÓN 22 DIAGRAMA DE SECUENCIA USER STORY Nº 4.....	48
ILUSTRACIÓN 23 DIAGRAMA DE SECUENCIA USER STORY Nº 5.....	48
ILUSTRACIÓN 24 DIAGRAMA DE SECUENCIA USER STORY Nº 7.....	49
ILUSTRACIÓN 25 DIAGRAMA DE SECUENCIA USER STORY Nº 8.....	49
ILUSTRACIÓN 26 DIAGRAMA DE SECUENCIA USER STORY Nº 9.....	49
ILUSTRACIÓN 27 DIAGRAMA DE SECUENCIA USER STORY Nº 10.....	50
ILUSTRACIÓN 28 DIAGRAMA DE SECUENCIA USER STORY Nº 11.....	50
ILUSTRACIÓN 29 DIAGRAMA DE SECUENCIA USER STORY Nº 12.....	50
ILUSTRACIÓN 30 DIAGRAMA DE SECUENCIA USER STORY Nº 13.....	51
ILUSTRACIÓN 31 DIAGRAMA DE SECUENCIA USER STORY Nº 14.....	51
ILUSTRACIÓN 32 DIAGRAMA DE SECUENCIA USER STORY Nº 15.....	52
ILUSTRACIÓN 33 DIAGRAMA DE SECUENCIA USER STORY Nº 16.....	52
ILUSTRACIÓN 34 DIAGRAMA DE SECUENCIA USER STORY Nº 17.....	52
ILUSTRACIÓN 35 DIAGRAMA DE SECUENCIA USER STORY Nº 18.....	53
ILUSTRACIÓN 36 DIAGRAMA DE SECUENCIA USER STORY Nº 19.....	53

ILUSTRACIÓN 37 DIAGRAMA DE SECUENCIA USER STORY Nº 20.....	54
ILUSTRACIÓN 38 DIAGRAMA DE SECUENCIA USER STORY Nº 21.....	54
ILUSTRACIÓN 39 DIAGRAMA DE DOMINIO DE LA SOLUCIÓN	55
ILUSTRACIÓN 40 DIAGRAMA DE CLASES. MÓDULO SITIO WEB	57
ILUSTRACIÓN 41 DIAGRAMA DE CLASES. MÓDULO WEB SERVICES.....	59

Resumen Ejecutivo

El presente documento es el informe de esta práctica de especialidad, caracterizada por basarse en dos proyectos excluyentes.

El primero, es un la realización de un prototipo funcional de una aplicación GIS (Sistema de Información Geográfica), enfocado a la interpretación de datos proporcionados por dispositivos GPS (Sistemas de Posicionamiento Global).

Este prototipo se encuentra dentro de un ámbito muy peculiar, en el cual se generan los supuestos de la alimentación externa de los datos y el almacén de estos en estructuras similares a los estándares en desarrollo por IEEE. También encuentra como limitante el manejo de estos datos sobre dispositivos PDA, y la actualización de la información de forma periódica.

La segunda parte del la práctica, consiste en la realización de un Sistema Integrado de Conectividad y Consultas para la Municipalidad de San Carlos.

Este sistema integrado, se divide en dos subsistemas. Uno de ellos es la creación de servicios Transparentes basados en comunicaciones vía Web (Web Services). El otro es la realización de un sitio Web, en el cual los contribuyentes podrán realizar consultas varias sobre sus estados de Tributos en la municipalidad.

Para ambos proyectos se deben aplicar nuevas metodologías de trabajo por implantar en la empresa.

1 Descripción del Problema

1.1 Enunciado del Problema

1.1.1.a Prototipo GIS PDA:

Debido a que este es un proyecto cuyo producto final será para el cliente interno, se plantea el problema de la necesidad de implementar un prototipo funcional de una aplicación GIS alimentada de datos GPS.

La implementación restringe el ámbito a una investigación previa de los principales productos comerciales disponibles para tal hecho. La adopción de alguno de estos es clave para el desarrollo satisfactorio de un prototipo, y su eventual compra para la puesta en marcha de una aplicación formal en este ámbito.

El prototipo debe ser la implementación parcial o total de un *framework*¹ comercial, o bien la propia implementación de un sistema similar.

Este prototipo debe ser capaz de sincronizar datos con un repositorio de información compatible, además de ser una aplicación para dispositivos PDA de la familia de Pocket PC. Además debe tener las capacidades básicas de cualquier sistema GIS.

Dentro de los detalles investigativos, se deben evaluar distintas soluciones que se encuentran en el mercado: opciones existentes sobre la plataforma JAVA, u opciones sobre la plataforma Microsoft. Aparte de la conclusión del mejor proceder: Adquirir una herramienta con funciones GIS PDA o desarrollar una propia para tal hecho.

La investigación debe poder concluir cual de toda la rama de opciones es la mejor, y a partir de ahí, concentrar los esfuerzos en realizar una evaluación de cualquier solución elegida, y proceder a desarrollar el prototipo correspondiente.

Por ejemplo, en caso de elegir una plataforma JAVA utilizando librerías de terceros, se deberá desarrollar un prototipo sobre esta plataforma (J2ME²), y en cuyas funciones principales debe integrar las funcionalidades específicas para este prototipo, utilizando las librerías aportadas por este software de terceros.

Este prototipo, deberá ser el reflejo del uso de una librería específica por un tercero, o la implementación de un desarrollo desde cero. El cual, deberá manejar

¹ Conjunto de funciones y librerías que realizan tareas determinadas en una plataforma específica.

² J2ME: Java 2 Micro Edition: Edición dos de Java embebida para dispositivos móviles.

información relacionada con elementos parametrizables de un sistema geográfico de información real, y el cual representará la situación real de una actividad de negocio en específico, ayudado distintas variables externas (repositorios, archivos planos, XML³, GPS, etc.).

1.1.1.b Sistema Integrado de Consultas y Conectividad

Este proyecto es un producto a la medida, del cual se derivan dos módulos fundamentales: Módulo de Consultas y Módulo de Conectividad mediante Web Services.

Los requerimientos no funcionales de este proyecto son varios, dentro de los principales se encuentra la integración de los sistemas en un repositorio de datos centralizado y que ya se encuentra en uso, por lo que no será necesario realizar más que adiciones a este repositorio para el correcto funcionamiento del sistema.

Otro de los requerimientos funcionales, es el manejo de las correctas transacciones que se realicen en el sistema; ya que este proyecto será parte del repositorio central de información, debe tener la consistencia lógica sobre las transacciones realizadas (transacciones sobre movimientos de cuentas de los contribuyentes de la municipalidad). Ya que estas se verán reflejadas en la recolección de impuestos a los contribuyentes.

Este sistema será un producto basado en una tecnología extensible, con el uso de plataformas de punta. Es por eso que se utiliza la tecnología orientada al servicio, llamado Web Service, con una plataforma de punta por excelencia: J2EE.

Para el desarrollo del módulo de Consultas, se han especificado requerimientos no funcionales como facilidad de navegación, baja carga de de contenido y otros. También se debe tener en cuenta que el módulo de Consultas debe reflejar las transacciones reales que realiza un Contribuyente en las entidades financieras preferidas, por lo que la información debe estar siempre actualizada.

Dentro de las restricciones que posee este proyecto, existe la necesidad de transparencia en la creación de los Web Services, la interoperabilidad entre los

³ XML: Extensible Markup Language. Estándar desarrollado por W3C para manejo de transacciones y datos en un lenguaje neutral y extensible basado en Objetos.

navegadores Web más comunes sobre el mercado, y el garantizar la integridad de los datos transaccionales.

Otro punto importante, es que debe ser seguro. Esto puede ser muy subjetivo, pero en el ambiente de Web Services, existen varios estándares para garantizar distintos tipos de seguridad en cuando al transporte, transacciones y otros. Entre estos se pueden ver:

- Nivel de Capa de Transporte:
 - Protocolos de Seguridad Basados en protocolos http seguros (http y SSL)
 - Certificados de Autenticidad de Emisores y Receptores.
 - Validaciones de certificados de Autenticidad.
 - Seguridad en Capas Físicas de Transporte: (Firewalls⁴, Autorizaciones, etc.)
- Nivel de Lógica de Seguridad:
 - Autenticación (¿Quién es usted?)
 - Autorización (¿A qué tiene permiso usted?)
 - Integridad (¿Lo que usted me envía es lo mismo que yo recibo?)
 - Disponibilidad (¿Está el servicio siempre disponible?).
 - No Repudio. (¿Me asegura que usted me envió respuesta?).
 - Confidencialidad. (¿Alguien más puede ver los datos que envió?).
- Nivel de Encriptación:
 - Extensibilidad. (¿Los algoritmos utilizados van a soportar el paso del tiempo?)
 - Adaptabilidad. (¿Los algoritmos utilizados causarán fallas en futuras implementaciones?).
- Nivel de Transporte XML SOAP:
 - Codificación. (¿Están los datos del XML encriptados?).
 - Certificación. (¿Los mensajes SOAP tienen mecanismos de integridad?)
 - Carga. (¿Qué tamaño máximo podrán tener estos mensajes SOAP?).

⁴ Medio Lógico o Físico que protege una Red contra peticiones externas (Internet) o accesos indeseados.

1.2 Enunciado de la Solución

1.2.1.a Prototipo GIS-PDA

Se pretende crear un prototipo basado en un *framework* existente, el cual contiene librerías y funciones específicas para el manejo de datos GIS, mas aun, se deben implementar las estructuras lógicas necesarias, interfaces de aplicación GUI especializadas para dispositivos de la familia de Pocket PC, y la correcta integración con acceso a datos de esta aplicación.

Dentro de las soluciones esperadas, se plantea la utilización de bases de datos orientadas a ficheros (para el dispositivo móvil). Este motor de base de datos es Microsoft SQL CE (Compact Edition), el cual provee de la integración adecuada con dispositivos PDA de la familia de Pocket PC y con el sistema operativo compatible.

Para la visualización de información, se utiliza lenguaje C# implementando una plataforma basada en Windows 32 bits (Microsoft .Net).

Debido a que la aplicación debe mantener la integridad de los datos, se debe implementar un mecanismo de sincronización de esta base de datos con una centralizada, haciendo persistente la información requerida, y tomando en cuenta cuestiones como rendimiento y espacio. Para esto se utilizan las herramientas de Integración de datos propia de Microsoft SQL Server 2005 y Microsoft SQL CE llamada replicaciones mediante las publicaciones. Servidor Web intermedio: Microsoft IIS⁵ v 6.0.

1.2.1.b Sistema Integrado de Consultas y Conectividad

Se planea la creación de un proyecto conformado por dos módulos. Estos módulos deben ser totalmente compatibles entre ellos, y deben ser persistentes con los sistemas actuales implantados.

La solución planteada se divide por módulos:

1. Módulo de Consultas

Se pretende realizar un sistema basado en consultas de información ya existente y almacenada.

⁵ Internet Information Services (IIS)

Este sistema deberá poseer características específicas de seguridad y consistencia de los datos, así como la interoperabilidad entre navegadores, rendimiento y acceso persistente de los datos.

Entre las soluciones tecnológicas por implementar se especifican:

Publicador de Sitio Web	Apache Tomcat ⁶
Plataforma	J2EE ⁷
Manejador de Pruebas	JUnit ⁸
Otros	Spring ⁹ Hibernate ¹⁰ , Webwork ¹¹ , SOAP ¹²

2. Módulo de Servicios Web

Como parte del otro módulo de Conectividad e integración, se debe mantener una transparencia en este tipo de conectividad y paso de mensajes. Es por eso que se implementan distintas funcionalidades basadas en la misma plataforma, pero orientadas a su trascendencia al paso del tiempo, transparencia de plataformas, extensibilidad y estándar. Es por eso que, aparte de las tecnologías antes mencionadas, se implementan comunicaciones basadas en SOAP, cuyo objetivo primordial es mantener una transparencia absoluta entre plataformas, y brindar la mayor extensibilidad posible al sistema realizado.

Con respecto a los aspectos de seguridad, se implementan las siguientes políticas para garantizar que los datos que viajan sean seguros y consistentes.

3. Nivel de Capa de Transporte

- HTTP sobre SSL: Se utiliza el motor llamado Tomcat, como publicador de los servicios Web. Este a su vez es configurado para que posea su propio certificado

⁶ Apache Software Group.

⁷ Java 2 Enterprise Edition.

⁸ Un manejador de Pruebas para plataformas Java. (www.junit.org).

⁹ Un framework diseñado para inyección de objetos tipo Java Beans (www.springframework.org).

¹⁰ Un producto libre proporcionado por JBoss Corporation (www.hibernate.org).

¹¹ Un producto libre proporcionado por OpenSymphony (www.opensymphony.com).

¹² SOAP: Simple Object Access Protocol. Estándar desarrollado por W3C (<http://www.w3.org/TR/soap/>)

SSL basado en *keystores*¹³ y certificados de Autenticidad que deberán ser validados por alguna autoridad relevante (Por ejemplo, VerySign).

- Control de acceso al servidor de Web Service: Se utiliza una seguridad integrada basada en el motor de publicación de los Web Services llamado AXIS.
- Utilización de SHA-1 como estándar de certificación de datos para llaves públicas, y X.509 para llaves privadas.

4. Nivel de Lógica de Seguridad:

- Se utilizan métodos muy específicos y hechos a la medida, para autenticación, autorización y certificación de los datos enviados y recibidos.
- Se utilizan hasta tres distintos métodos de encriptación de datos basados en llaves públicas, privadas y otros.

5. Nivel de Transporte XML SOAP

- Se realizan encriptaciones de todos los datos enviados y recibidos.

¹³ Almacén de Datos de Proveedor del Certificado, basado en estándares de autenticación por LDAP y encriptaciones de datos utilizando SHA-1 como método transparente.

1.3 Perspectivas, Supuestos y Dependencias del Proyecto

1.3.1.a Perspectivas:

Con respecto al prototipo GIS – PDA, se tiene como expectativa principal la implantación de los siguientes objetivos:

- Implementación de un medio en el cual se pueda sincronizar información mediante protocolos basados en TCP.
- Implementación de una aplicación altamente eficiente y embebida para dispositivos PDA.
- Tener una perspectiva amplia y real del esfuerzo que involucra crear una aplicación de este tipo.
- Analizar los factores técnicos que dan a un tipo de aplicación de este tipo, el valor agregado que se merece.

Como parte de los supuestos que tiene el proyecto de Conectividad y Consultas, se desea lo siguiente:

- Contar con un sistema robusto de consultas en línea, para asociados a la municipalidad de San Carlos.
- Analizar el costo beneficio que conlleva la realización de un producto completo utilizando las metodologías que se están aplicando.
- Contar con una primera perspectiva empresarial del trabajo y habilidad técnica que merece un sistema de Conectividad por Web Services.
- Contar con un sistema totalmente integrado entre sus módulos de conectividad y consultas.
- Garantizar la seguridad del sistema, según estándares actuales.

1.3.1.b Supuestos:

Prototipo GIS - PDA

- Se supone la alimentación de datos por parte de una fuente externa.
- Se mantiene la estructura lógica implementada en la solución en forma de Prototipo.

Para el Sistema integrado de Conectividad:

- Se posee una base de datos relacional de la cual se parte para la realización de los procesos de consulta y transacciones.
- Se posee de la plataforma tecnológica compatible para la implantación de la solución.
- Se provee de los recursos tecnológicos para el monitoreo, instalación y publicación de las soluciones creadas.

1.3.1.c Dependencias:

- Tecnologías:
 - SQL Server 2005 como repositorio de datos GPS.
 - SQL Management Studio como manejador de Base de Datos.
 - SQL CE, repositorio local Embebido para aplicaciones PDA
 - .Net Framework 2.0. Conjunto de librerías estándar para el manejo de ambientes basados en Microsoft.
 - Lenguaje C# para el manejo de transacciones y objetos.

 - SQL Server como motor de Base de Datos
 - Sybase como actual repositorio de datos.
 - Sybase Central, Embarcadero Studio, SQL Server Management Studio 2005, como motores para manejo de Base de Datos.
 - Apache Tomcat. Motor de publicación Web para plataformas basadas en Java.
 - Apache Axis como publicador de Web Services.
 - JSP como principal implementado de páginas activas.
 - SOAP, Idioma transaccional transparente para el manejo de datos en protocolos basados en Internet y TCP.
 - JSSE. Java Security Standard Edition. Para el manejo de seguridad.
 - Certificados LDAP.
 - Hibernate, Spring, WebWork: Frameworks encargados de estandarizar el trabajo de desarrollo y garantizar la persistencia transaccional con cualquier tipo de plataforma.
 - J2EE como plataforma básica de desarrollo.
 - JUnit como Integrador de Calidad y Casos de Prueba

- Eclipse 3.2 como herramienta de Desarrollo.
- Navegadores Web (Mozilla Firefox, Internet Explorer)

2 Objetivos y Alcances

2.1 General

Implementar una aplicación para dispositivos PDA, que sea capaz de representar Datos GPS en un sistema GIS complejo y coherente.

Crear un sistema de consultas sobre pendientes de pago, en conjunto un Protocolo de Acceso a información e intercambio de datos entre emisor de servicios y Agentes Recaudadores (Bancos) a través de Web Services.

2.2 Específicos

- Logar un prototipo funcional como sistema integrado GIS para el modelado adecuado de información relacionada.
- Implementar un sitio Web orientado a las consultas.
- Implementar servicios transparentes para manejo de transacciones.
- Implementar transacciones basadas en mensajes RCP sobre protocolos TCP orientados a objetos y metadatos SOAP.
- Aplicar nuevas metodologías de diseño, especificación, toma de requerimientos, desarrollo y pruebas.

2.3 Alcances

2.3.1 Prototipo GIS-PDA

- El sistema debe ser capaz de sincronizar los datos con un servidor principal.
- El sistema debe modelar la información presente, en forma de figuras lógicas a un nivel comprensible para el usuario.
- El sistema debe contar con funciones de Acercamiento.
- El sistema debe implementarse sobre un dispositivo PDA o emulador compatible.

2.3.2 Sistema de Conectividad

- El sistema debe poder consultar las cuentas pendientes de un usuario determinado.
- El sistema debe autorizar solamente a los usuarios registrados la consulta de la información.
- El sistema debe poder consultar los conceptos de cada pendiente en específico.
- El sistema debe poder consultar el histórico de Pagos que se han realizado en un período determinado.
- El sistema debe poder realizar transacciones vía Internet a modo de pago, no pago, pago por adelantado, etc.
- El sistema debe poder realizar transacciones vía Internet a modo de consulta.
- El sistema debe ser transparente en cuando a la plataforma / navegador utilizado.
- El sistema debe ser seguro, basado en estándares actuales.
- El sistema debe ser consistente con las transacciones realizadas en las entidades bancarias y en el sitio de Consultas.
- El sistema debe acoplarse a los datos ya existentes, ya que coexiste con información y sistemas presentes en la municipalidad.
- El sistema debe ser robusto, sólido y extensible.

3 Especificación de los Sistemas

3.1 Requerimientos No Funcionales

3.1.1.a Prototipo GIS-PDA:

1. Escalabilidad:

El prototipo debe estar diseñado para soportar su posible escalabilidad en un producto más complejo.

3.1.1.b Sistema de Conectividad:

1. Seguridad:

Este sistema debe ser totalmente seguro para manejar las transacciones por conceptos de pagos entre entidades Recaudadoras y la Municipalidad de San Carlos.

2. Modularidad:

El sistema de Consultas no debe depender de los otros sistemas en paralelo que se realicen; ni viceversa.

3. Escalabilidad:

El sistema y módulos generados deben ser mantenibles al paso del tiempo y parametrizables en todo lo necesario.

4. Transparencia.

El sistema de conectividad debe ser totalmente transparente a la tecnología cliente que lo consulte.

5. Consistencia:

El sistema debe mantener la consistencia lógica de la información sobre la que interactúe.

3.2 Características Generales

3.2.1.a Prototipo GIS – PDA

El sistema contará con una sincronización basada en protocolos TCP.

El sistema trabajará sobre dispositivos PDA o emulaciones similares.

3.2.1.b Sistema de Conectividad

El sistema contará con las siguientes características generales:

- Contará con un sitio Web en el que se podrán realizar consultas por conceptos específicos.
- El sistema Web Contará con métodos de autenticación, autorización y tiempos de caducidad.
- El sistema de Conectividad poseerá procedimientos secretos de encriptación.
- El sistema de Conectividad poseerá métodos de autenticación y autorización de entidades Recaudadoras.
- El sistema de Conectividad, asegurará la confidencialidad de los datos enviados por el canal de comunicación llamado protocolo http.
- El sistema de Conectividad será transparente.

4 Metodología de Requerimientos, Desarrollo y Pruebas

Northek Software, en su programa de mejora continua, ha venido implementando una metodología basada en *Lean Process*, el cual trata básicamente en la reducción de desperdicios operacionales de sus productos (llámese gastos operacionales a todas las actividades propias de la manufactura de productos Informáticos).

4.1 Lean Process y Lean Programming

Por medio de esta metodología, los procesos tradicionales conocidos en la producción de un software, se vuelven inadecuados y hasta inapropiados para una visión basada en la productividad. Es por eso que se han venido implementando distintas metodologías que acompañan al *Lean Process* y en específico en su variante de la generación de productos de software, para que esta sea lo más eficiente, eficaz y de mayor calidad posible.

Las bases del *Lean Process* son muchas, pero en especial se quieren rescatar las siguientes:

- Reduzca el Desperdicio.
- Mantenga una Postura de Inteligencia de Grupo.
- Cambie las reglas.
- Optimice el valor de su trabajo.
- No planee todo antes de Empezar. Acepte Retroalimentaciones y Ajuste.
- Empoderamiento

Es por eso que, parte de esta práctica de especialidad se basa en la implementación de estas metodologías en el desarrollo de productos informáticos, sustitutas de los procesos tradicionales de recolección de requerimientos, planeamiento, levantamiento de requerimientos, desarrollo de casos de uso, creación y planeamiento del diseño, etapas de pruebas y sobre todo, implementación de la solución.

Estas metodologías sustitutas de las tradicionales son específicamente:

- *Lean Programming*. (Programación Liviana)
- *Agile Development*. (Desarrollo Ágil)
- *Xtreme Programming*. (Programación Extrema)
- *Test Driven Development*. (Desarrollo orientado a la Prueba / Calidad)

Lean Programming, basa su metodología en la actitud que tienen los Líderes de Proyecto, Coordinadores de Proyectos, Gerentes de Producción, Aseguradores de Calidad y hasta los mismos Desarrolladores hacia el trabajo continuo y diario en la empresa. Cambiando actitudes, procesos y agregando calidad a todas las actividades que se realizan diariamente, restándole al mismo tiempo, cualquier desperdicio que se pueda generar de tal hecho.

4.2 Agile Development

Agile [Ficher, RB. 2004], tiene su propia metodología aplicada al manejo de requerimientos, tareas y otros en productos específicos de software. [Ficher, RB. 2004].

Trata cada requerimiento como una unidad general que enmarca un entorno un poco abierto, y no lo enfoca en todos los pasos a seguir para alcanzar un objetivo. Este requerimiento tiene el nombre de *Feature*, y puede cambiar con el paso del tiempo.

Características de un Feature:

- Es mutable con el paso del tiempo.
- Enmarca un conglomerado de funciones lógicamente interrelacionadas.
- Puede variar en cuanto a tamaño dependiendo de las especificaciones de cada cliente.
- Es el máximo exponente que debe quedar listo en cada ciclo.

Agile Development, está orientado a la interacción y manejo la prioridad y orden lógico que tiene cada requerimiento en el ciclo de vida del desarrollo del (los) producto(s) de un proyecto determinado.

Una vez definidos los requerimientos iniciales, se deben extraer los que son de mayor importancia para el cliente, dejando el margen inicial para la implantación de tecnologías propias del desarrollo de la solución, puesta en marcha de servicios necesarios, implantaciones de almacenes de datos, copias de información necesaria y los aspectos de inicialización que se consideren necesarios.

Es por eso que una vez que se tienen los requerimientos, se realiza un reordenamiento de las prioridades para el cliente, y se estiman entregables, avances y los demás procesos que tienen que ver con cada producto terminado.

Algo muy importante acerca de *Agile*, es que trabaja con productos terminados. O sea con entregables en fechas específicas o “*death lines*”. Esto permite ver el avance real del proyecto en general en fechas específicas, sin importar que tanto esfuerzo se necesite para la implementación en otras etapas o entregables. También se considera una ventaja, pues se trabajan cada story desde su inicio hasta su completitud, de modo que nada queda pendiente (o al menos no debería), y se genera toda la funcionalidad necesaria para el correcto funcionamiento de ese story en específico.

Los elementos que incluyen la metodología *Agile*, son bastantes, pero en este momento, los más caracterizados e importantes son **Features, User Stories** y **Tasks**.

1. Feature:

Un elemento caracterizado como *Feature*, corresponde a un nivel importante de abstracción dentro de los procesos fundamentales que debe cumplir un entregable en un tiempo definido. Se podría ver como un módulo pequeño con una serie de funcionalidades específicas y determinadas dentro de un contexto lógico coherente.

Características de un *Feature*:

Podrían definirse como Conjuntos de Funcionalidades lógicamente enlazadas.

No posee descripciones generales, sino que se explican por sí solos.

Son el mayor rango de completitud que se debe alcanzar en etapas de desarrollo.

2. User Stories:

Agile posee, además de *Features*, lo que se llaman *User Stories* o *Stories*. Este concepto es un poco más específico que un *Feature*, ya que enmarca una serie de funciones caracterizadas por un funcionamiento específico (sin especificar pasos en detalle).

Este *Story* posee características de mutabilidad y suele ser definido por una descripción no muy detallada del funcionamiento general que debe tener una funcionalidad, ante una acción – reacción determinada.

Características de un *User Story*:

- Debe ser explicativo, pero exactamente lo necesario. Ni más ni menos.
- Tiene que ser orientado hacia el Cliente, no hacia la función que realiza.
- Debe ser algo sencillo, fácil de explicar en 30 segundos, debe ser lo necesario para introducir a procesos más detallados de qué en específico se debe hacer.
- Deben facilitar la escritura de un correcto caso de prueba.
- No debe contener pasos específicos, sino funcionalidades en general.

3. Tasks

El *Task*, es la especificación más pequeña que puede tener *Agile*. Este existe solamente en el término de un *Story*, que a su vez existe solamente dentro de un *Feature*.

Este *Task* posee el nivel de detalle más específico posible, describiendo pasos o procedimientos muy característicos de una funcionalidad en específico, mas aun no debe describir procedimientos detallados de la implementación.

Cabe mencionar, que No todos los *Stories* poseen *Tasks*, y cada *Story* puede contener desde ninguno, hasta cualquier cantidad de *tasks*. Solamente que no se recomienda tener muchos *tasks* dentro de un solo *Story*, ya que se pierde la idea del *Story*.

Por último, un ejemplo.

Feature: Administración

Story: Un usuario Autorizado podrá administrar información de Clientes.

Task: Registrar un Cliente

Task: Consultar un Cliente.

Task: Al eliminar una cuenta, se debe mantener la integridad lógica ingresando este Cliente y su último estado dentro de la Bitácora del sistema.

Additional Info	
Estimated Effort	0 h
Developer(s) Effort	0 h
Tasks Effort	6 h
Total Effort	6 h
Progress	100%
Time Spent	6 h
Time ToDo	0 h
Business Value	Must Have
State	<u>Done</u>
Assignments	Developer(s) V. Garro
Feature	Consultas
Release	RTM 1.0 / Finish: 28-sep-07
Iteration	Iteration #2 / Finish: 30-jul-07
Owner	
Creation Date	
Completion Date	

Ilustración 1 Ejemplo Información de un Feature

STORY #3. Bienes Inmuebles

General | **Tasks** | Times | Bugs | Test Cases | Source | History

Como usuario, quiero poder consultar el monto a pagar por concepto del impuesto de Bienes Inmuebles utilizando mi número de cédula

Attachments

[Attach a file](#)

Comments [[add](#)]

Ilustración 2 Ejemplo un Story

STORY #3. Bienes Inmuebles

General | **Tasks** | Times | Bugs | Test Cases | Source | History

Name	State	Effort	Progress	Time Spent	Time Remain	Assignments
TASK Consulta de históricos	Done	3 h	100%	3 h	0 h	Developer(s) V. Garro
TASK Consulta de pendiente	Done	3 h	100%	3 h	0 h	Developer(s) V. Garro

Ilustración 3 Ejemplo Tasks sobre un Story

4.3 Xtreme Programming

Como parte de las metodologías que se utilizan, para la especificación detallada de los procesos específicos que deben cumplir cada requerimiento, se utiliza la *Xtreme Programming*.

Esta metodología es muy poderosa cuando se combina con otras (como *Agile*) ya que permite tener una interacción directa con el cliente y con las necesidades que debe solventar en el menor tiempo posible (parte de las características de *Agile*).

La idea básica de Extreme, es conocer el ámbito de lo que se debe realizar en el momento en que se debe realizar, no antes.

Esto deja problemas como la no incorporación de funcionalidades específicas de un requerimiento puntual, al no tener el ámbito general de lo que debe hacer esta funcionalidad. Todo esto porque se extrae del cliente (mediante diversos métodos) directamente el cómo quiere que se implemente este requerimiento, y todos los pasos que debe tener.

Podría ser un problema en caso de la incorporación de nuevos requerimientos no previstos en un inicio. Pero el impacto sobre el avance total de proyecto estimado no es visible, ya que se manejan nuevos requerimientos como un punto y aparte.

Otra de las técnicas de Extreme, es implementar exactamente lo que se necesita implementar. no más. Y esto quiere decir que desde que se lee por primera vez el requerimiento, se deben realizar etapas como levantamiento de funcionalidades específicas, diseño, diagramas, interfaces, aspectos de GUI y hasta estructuras estáticas de datos, si se requieren. O sea, no existe todo un proceso de diseño de todo el proyecto, ni creación de modelos de clases de todo, sino más bien, un diseño y modelo de lo que se necesita en el momento que se necesita; pero tomando en cuenta toda la estabilidad y el diseño que debe tener todo producto de software de calidad.

4.4 Test Driven Development

Esta metodología es una completa implementación de políticas altamente productivas y orientadas a la producción eficiente y de alta calidad en el ámbito de software. [Robert C 2005].

La idea básica de esta metodología es crear las pruebas antes de realizar la parte programada del sistema o aplicación. Creando exclusivamente lo necesario para cumplir con estas pruebas.

Como parte de los pasos para una correcta implementación de la metodología *Test Driven*, es tener todo el entorno de desarrollo completamente listo para cualquier ajuste.

Luego, se procede a crear los casos de prueba y su correspondiente “*runner*”, para hacer funcionar estos casos; luego se realiza la programación funcional del sistema basado en lo que dicen estos casos.

Como bien se menciona anteriormente, es totalmente contrario a la metodología habitual de creación de software. En este caso, se crean las pruebas de tal modo que satisfagan un escenario específico, luego otro y así consecutivamente.

Ventajas:

- Implementación rápida y orientada a la producción ágil.
- No existe desperdicio de tiempo en aspectos de diseño.
- Cumple exactamente lo que el cliente desea, ya que el caso de prueba atiende exactamente el escenario reflejado.
- Abstrae en un mayor nivel los requerimientos no funcionales.
- Asegura la calidad e interoperabilidad con otros módulos en el paso del tiempo (desarrollo).
- Provee la guía para que el programador realice las tareas específicas que debe realizar, sin malgastar tiempo en otros asuntos menos importantes.
- Provee la primera herramienta en detección de errores y aseguramiento de calidad en funcionalidades específicas.

Desventajas:

- Se requiere más tiempo en etapas iniciales, puesto que se debe averiguar todos los posibles escenarios.

- Se debe ajustar con el paso del tiempo, debido a dependencias lógicas.
- Se necesita mayor disponibilidad del cliente, algo que es complicado dadas las restricciones de tiempo que el cliente pueda tener

Como parte de la implementación, se realiza un estudio de todos los casos de excepción que se pueden dar en una funcionalidad específica. Una vez realizado esto, se procede a configurar y dejar listo el ambiente en el cual estas pruebas podrán ejecutarse satisfactoriamente (creación de objetos, estructuras, dependencias, etc.). Luego se crean los procedimientos, invocaciones u otros, en los cuales se generarán accesos a objetos creados o resultados de las funciones especificadas en los requerimientos.

Esto genera un caso de prueba, el cual nunca va a compilar inmediatamente, puesto que no se han generado las funciones para acceso a objetos, acceso a datos, capa de negocios, transacciones, capa de presentación, etc.

A partir de que se tienen los casos de prueba (o un solo caso de prueba a la vez, dependiendo de la metodología), se procede a programar cada llamada, capa de negocios, objeto u otro, en orden de necesidad según su precedencia.

Es así como en el momento en que se tiene un requerimiento terminado, se tiene también el caso de prueba para asegurar su calidad y certificar su funcionamiento. Además se tiene previsto que en el momento que se genere otra funcionalidad extra, estos casos de prueba se ejecutarán para que se asegure que todo esté funcionando bien.

5 Solución: Prototipo de Implementación GIS-PDA

5.1 Proceso de Desarrollo

Para el proceso de desarrollo de este prototipo de implementación, se ha elegido utilizado una selección y combinación de las distintas metodologías mencionadas anteriormente. Específicamente, se ha preferido el desarrollo utilizando una técnica de requerimientos basada en *Xtreme Programming*, y guiada por *Lean Process* y *Agile Development*.

Estas metodologías permiten:

- *Xtreme Programming*: Conocer los detalles técnicos y de proceso de cada requerimiento, en el momento que será desarrollado.
- *Lean Process*: Mantener el planeamiento del proyecto y los ciclos de desarrollo de la forma más liviana y sin gasto de recursos.
- *Agile Development*: Asegurar que se realiza exactamente lo necesario, de la mejor forma, y con el ahorro necesario y requerido para los procesos productivos de la unidad (desarrollo).

5.1.1 Análisis del Proyecto.

5.1.1.a Propuesta de realización:

Se requiere de un prototipo de simulación de una aplicación GIS, basada en alimentación externa de datos GPS por medio de otras fuentes. Este prototipo simulará la aplicación de propiedades específicas en forma de atributos sobre secciones conceptualizadas dentro de un esquema basado en meta datos y almacenes de información tipo GIS, en donde cada entidad funcional de segmentación total de un conglomerado de instancias pertenecientes a una propiedad lógica, será caracterizada por la especialización específica según tareas relacionadas (atributos) y comportamientos específicos (parámetros) y otros que se consideren necesarios. Aparte de dar la opción real de visualización simulada del escenario real basado en datos correctos. El prototipo debe ser una aplicación para dispositivos móviles de la familia de las Pocket PC.

5.1.1.b Requerimientos:

Para poder dar una aproximación real de los requerimientos como tales, es necesario aclarar que un *User Stories* está totalmente enfocado en procesos del negocio y hacia el cliente, y no hacia la funcionalidad propia de la solución; por lo tanto estos requerimientos se especifican de la siguiente forma:

User Stories:

- Como usuario, deseo alimentar la información de fuentes externas de almacén.
- Como usuario deseo poder sincronizar la información almacenada y actualizada.
- Como usuario podré ver Unidades de Producción con sus respectivos atributos y parámetros.
- Como usuario podré aplicar filtros acerca de cuales unidades de producción podré analizar.
- Como usuario deseo poseer mi aplicación en mi dispositivo Pocket PC.
- Como administrador deseo poder limpiar la información contenida en el dispositivo PDA.

5.2 Modelo de Diseño - Diagrama Conceptual

A continuación, se muestra un diagrama conceptual de los principales procesos que se ven involucrados dentro del accionar de la aplicación GIS – PDA¹⁴, y en donde muchos factores externos incluyen directamente en el resultado final.

En el siguiente diagrama se denotan procesos de sincronización, manejo de mapas geo-referenciados mediante librerías y/o procesos totalmente desarrollados; así como los roles de almacén de datos que tiene todo el sistema en su gran ámbito. El flujo de información sigue una tendencia lineal desde la alimentación y los servicios que se otorgan a nivel de los subsistemas. De este modo, el proceso de adquisición de Datos GPS no es tomado en cuenta, aunque sí es necesario este proceso, dentro del funcionar de la solución.

Luego sucede un servicio, llamado servicio de sincronización, el cual puede ser llamado manualmente “Sinc. Manual”, o puede ser un proceso interno de la aplicación de escritorio.

¹⁴ La definición GIS y PDA ya se ha definido en documentos anteriores.

Es importante mencionar que este servicio de sincronización, es uno de los requerimientos originales del proyecto, y que su implementación se hizo mediante la investigación contenida en este documento.

Luego, se puede denotar que la aplicación PDA contiene distintas funcionalidades en las cuales interviene un DAO¹⁵ para una réplica del repositorio original, pero almacenada en la PDA.

Por otro lado, la solución de escritorio, posee así mismo un DAO, pero este se encarga automáticamente de la sincronización interna con los datos. También existen Funcionalidades específicas para la aplicación de Escritorio.

Por otro lado, existe también una entidad inherente a ambas soluciones, la cual se encarga de el mapeo visible a nivel de un Mapa Físico, mediante la vectorización de coordenadas reales; y es aquí donde todo se une de forma visible al usuario final.

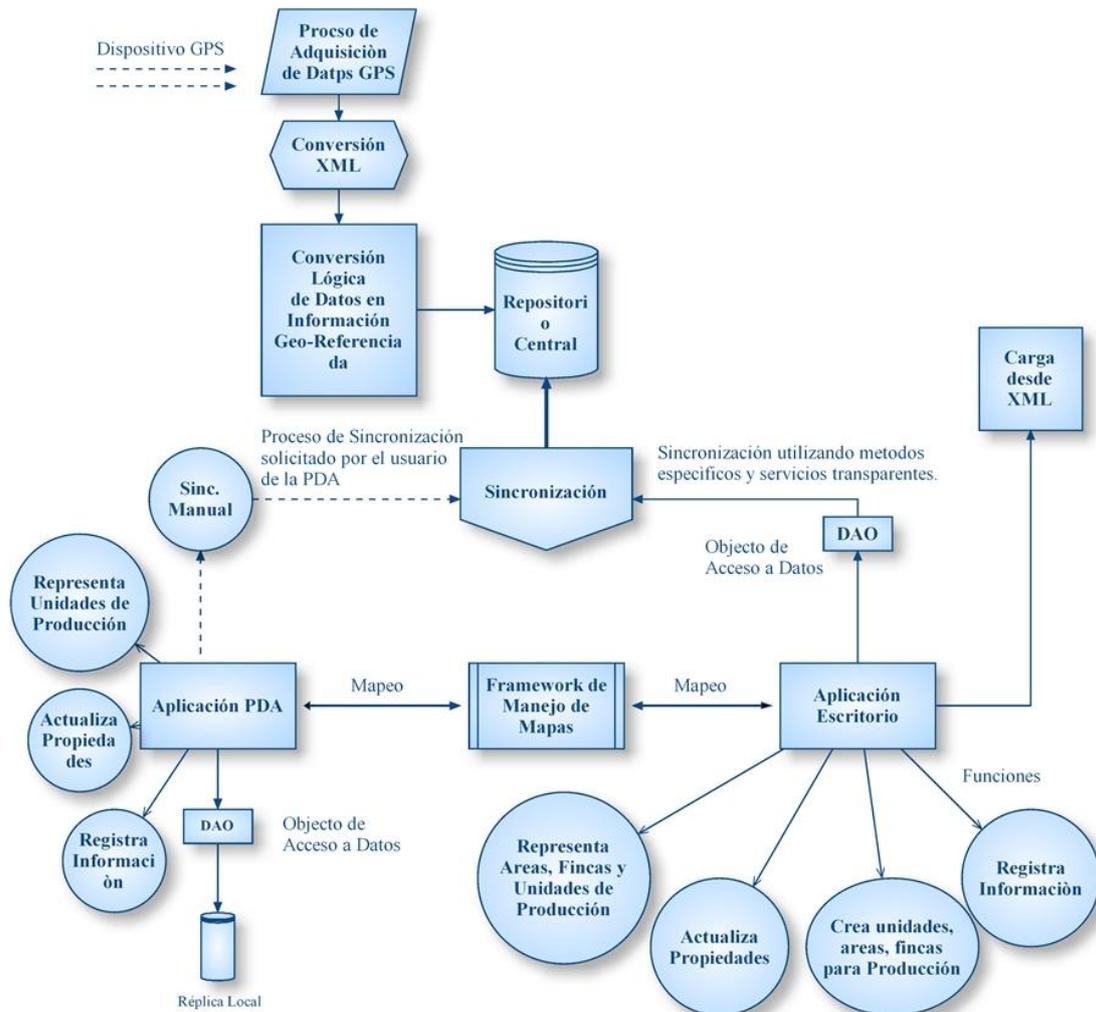


Ilustración 4 Diagrama General de Funciones e Interrelaciones

¹⁵ DAO : Data Access Object. Objeto de Acceso a Datos.

5.3 Diagrama de User Stories

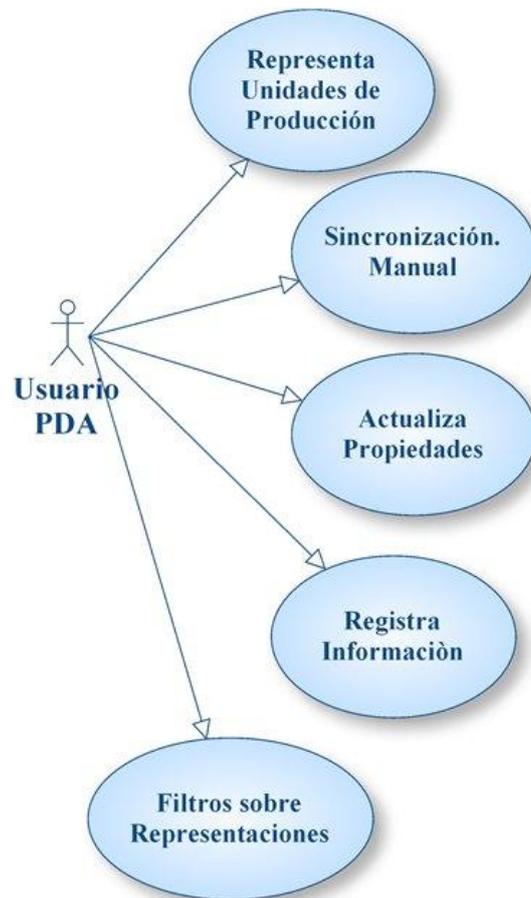


Ilustración 5 User Stories para El usuario con PDA

5.4 User Stories

User Story N°: 1

Nombre: Representar Unidades de Producción.

Actor: Usuario de Dispositivo PDA.

Descripción: Como usuario, deseo poder ver las principales unidades de producción, representadas como áreas de propiedades Geo-Referenciadas reales.

User Story N°: 2

Nombre: Sincronización Manual.

Actor: Usuario de Dispositivo PDA.

Descripción: Como usuario, deseo poder sincronizar la información actualizada o antigua de mi dispositivo PDA, con el repositorio central de información.

User Story N°: 3

Nombre: Actualizar Propiedades

Actor: Usuario de Dispositivo PDA.

Descripción: Como usuario, deseo poder cambiar, actualizar, agregar o quitar propiedades de cada unidad de producción o área representada dentro del mapa geo-referenciado.

User Story N°: 4

Nombre: Registro de Información.

Actor: Usuario de Dispositivo PDA.

Descripción: Como usuario, quiero poder registrar la información representativa a una unidad de producción específica, por los parámetros especiales establecidos.

User Story N°: 5

Nombre: Filtros sobre La representación.

Actor: Usuario de Dispositivo PDA.

Descripción: Como usuario, quiero poder aplicar filtros de visualización sobre las distintas unidades de producción reflejadas en el mapa geo-referenciado.

5.5 Diagramas de Secuencia

User Story N°: 1

Nombre: Representar Unidades de Producción.

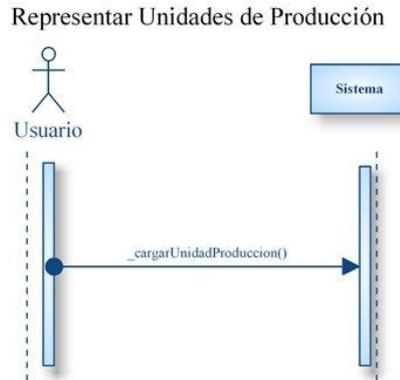


Ilustración 6 Diagrama de Secuencia User Story N° 1

User Story N°: 2

Nombre: Sincronización Manual.

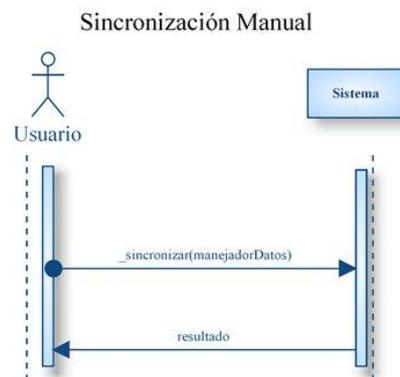


Ilustración 7 Diagrama de Secuencia User Story N° 2

User Story N°: 3

Nombre: Actualizar Propiedades

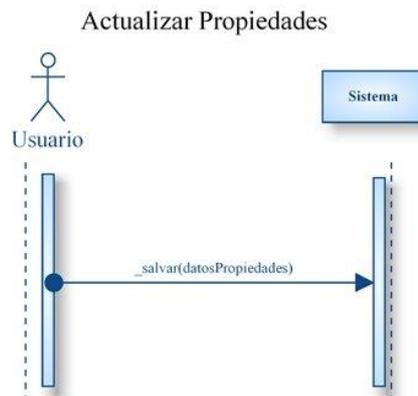


Ilustración 8 Diagrama de Secuencia User Story N° 3

User Story N°: 4

Nombre: Registro de Información.



Ilustración 9 Diagrama de Secuencia User Story N° 4

User Story N°: 5

Nombre: Filtros sobre La representación.

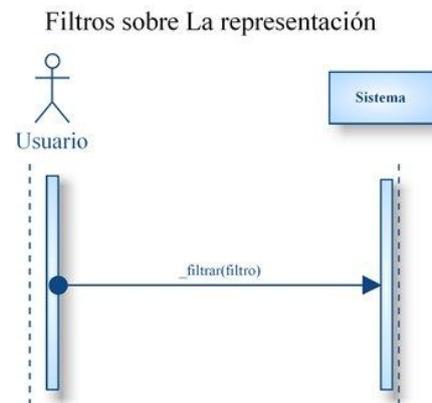


Ilustración 10 Diagrama de Secuencia User Story N° 5

5.6 Modelo de Dominio

En el siguiente diagrama, se representa el dominio completo en el cual, muy simplificadamente, intervienen las distintas entidades de las soluciones.

En primera instancia, los datos GPS externos, son la base de todo este sistema, por lo que se encuentran en la cumbre del diagrama.

En un segundo nivel, se tiene la especialización de soluciones: PDA o Escritorio. Cada una de ellas utiliza un repositorio de datos, que se debe ser sincronizado. Este repositorio puede ser local o remoto, pero debe estar sincronizado de alguna manera.

Luego, también existe el *framework* de mapeo, el cual muestra visiblemente las acciones realizadas por las distintas soluciones. Este se encuentra en el centro del diagrama, debido a que es parte de ambas soluciones, y es el resultado de la investigación hecha.

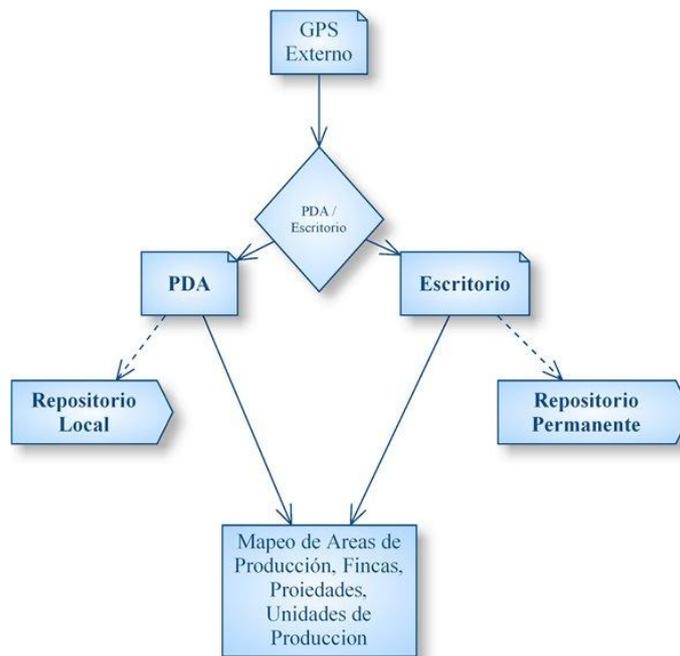


Ilustración 11 Diagrama de Dominio

5.7 Diagrama de Clases

El siguiente diagrama de Clases, representa la abstracción que se realizó para la implementación tanto del prototipo a nivel de aplicación de escritorio, como el prototipo a nivel de aplicación PDA.

En primera instancia, se abstrae la capa de GUI (Interfaz Gráfica) de las capas de negocios. Aquí interviene el *framework* de mapeo producto de la investigación mencionada, ya que este se encarga de representar los datos gráficamente.

Luego se tiene una capa de servicios, la cual es parte de las capas de negocios, pero es utilizada para la interfaz (comunicación) entre los procesos específicos del negocio, y presentación.

Para la parte de negocios, se realizan las distintas funciones y procesos, mediante invocaciones. Algunas de estas operaciones son las llamadas CRUD¹⁶. Estas hacen uso de los objetos DAO, que son encargados de mantener la persistencia con el repositorio de datos. Existe así mismo, un proceso o servicio encargado de realizar la sincronización mediante la utilización de estos DAO.

Una vez entendido esto, se tienen las distintas entidades propias del negocio, que serán objetos y colecciones reflejo de la estructura del repositorio de datos.

La serialización de datos, es un tema complejo y consta de varios detalles de internases con objetos, manejos mediante sesiones y otros procesos inherentes a la implementación en sí; es por eso que se ha llamado serialización a esta abstracción de funciones, abstracciones y extensiones.

¹⁶ CRUD: Create, Retrieve, Update, Delete. (Operaciones de Crear, Obtener, Actualizar y Eliminar)

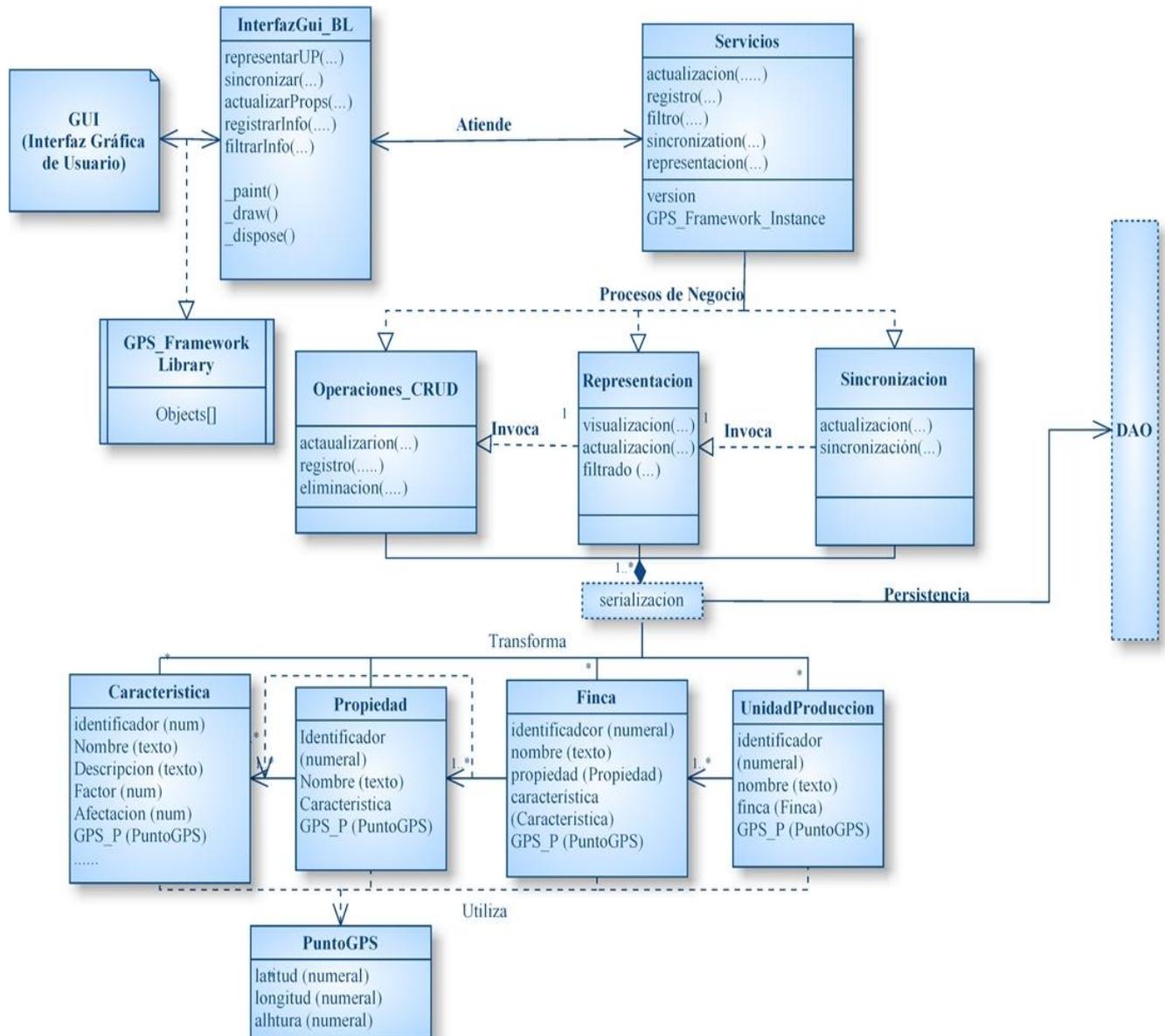


Ilustración 12 Diagrama de Clases

5.8 Implementación

Para la implementación de esta solución, se recurre a las tecnologías de Microsoft Visual Studio .Net, versión 2005 como herramienta de Programación y Emulación, Microsoft SQL Server 2005 Team Edition.

El desarrollo implementado de esta etapa, tiene varios entregables, en distintas etapas:

- Investigación:
 - Documento Evaluativo de las Herramientas y Plataformas más óptimas para desarrollar una versión completa de este prototipo.
- Implementación
 - Manual de Sincronización
 - Prototipo desarrollado

5.8.1 Investigación:

5.8.1.a Análisis Evaluativo.

Durante el proceso de investigación, se encontraron distintos productos de las Líneas de Software Propietario, y de Software Libre.

Este proceso evaluativo, basa su investigación en tres campos específicos: Tecnología, Interoperabilidad, Evaluación.

Tecnología: En este punto, se investiga por tecnologías populares, actuales, escalables y que tengan en menor impacto posible para un eventual proceso de desarrollo. Por este motivo se enfoca la investigación en tres ramas: Microsoft Visual Studio .Net, JAVA Sun Microsystems (J2ME, J2SE), Otras alternativas.

Dentro de estas ramas, se debe atender a la curva de aprendizaje del desarrollador de la solución en aprender a utilizar el producto, capacidades del producto, documentación, soporte, etc.

Interoperabilidad: Con respecto a esto, se investigan tecnologías que soporten el paso del tiempo, así como que brinden productos basados en estándares internacionales, Borradores de estándares (“*Drafts*” [IEEE 1998]) u otros mecanismos. Así mismo, se

busca que la interoperabilidad y soporte dado para dispositivos GPS basados en GPSML¹⁷,

Evaluación: Este segmento se trata de una evaluación de Ventajas, Desventajas acerca de los tres mejores productos encontrados en las distintas Plataformas, y de los cuales se extrae el principal por cada rama, y a partir de este se debe realizar un prototipo funcional.

5.8.1.b Resultados

Con respecto a esta etapa, se han concentrado los resultados, en un documento de análisis de las tecnologías, de carácter confidencial y de uso interno exclusivo para Northek Software S.A.

Los únicos puntos que se pueden dar a conocer, son las tecnologías elegidas como mejores, y la utilización de una en específico para la realización del prototipo funcional. Este es el caso de GIS.NET¹⁸, producto de *GeoFrameworks* para plataforma .Net.

5.8.2 Entregables

5.8.2.a Manual de Sincronización

Este documento denominado Manual de Sincronización, responde a la necesidad de una sincronización de los datos en un repositorio central, dentro del entorno de una aplicación PDA.

Este punto es de vital importancia, porque de aquí parte el desarrollo o No desarrollo de la aplicación PDA, ya que la persistencia de los datos es vital para cualquier actividad de negocio, y para esta en específico adquiere un valor aun mayor.

Es por eso que se realiza primero la investigación / manual de sincronización de datos dentro de un entorno de aplicación PDA, con un motor de base de datos SQL-CE, hacia un ambiente robusto de base de Datos SQL –Server basado en plataforma PC .

¹⁷ GPSML significa, al igual que XML, GPS Markup Lenguaje.

¹⁸ <http://www.geoframeworks.com/Products/GIS/>

Como se puede suponer, la sincronización se pudo lograr de la mejor forma posible, y abogando específicamente a la utilización de herramientas y servicios de alta fidelidad basados en TCP. Específicamente la utilización de réplicas mediante suscripciones y publicaciones en servicios TCP interpolados con Microsoft IIS.

El entregable de estos procesos, es un manual de Instalación y Publicación de un servicio activo dentro un servidor Web basado en IIS, que pueda realizar sincronizaciones de datos, en correspondencia a una petición específica de un cliente determinado y autenticado.

5.8.2.b Prototipo Desarrollado

La solución de prototipo que implemente los métodos y tecnologías investigadas, se da dentro de un ambiente de funciones básicas especificado anteriormente por los casos de Uso.

Este proyecto se desarrolla sobre la plataforma Visual Studio .Net, en lenguaje C#, y este módulo se divide en dos aplicaciones:

Aplicación PDA

Esta aplicación funciona sobre dispositivos PDA de la familia de los Pocket PC, y debe realizar todas las funciones que se describen en la especificación de requerimientos y en el diseño de estos requerimientos.

El servicio denominado como sincronización debe desarrollarse para esta aplicación de forma que utilice las soluciones investigadas y desarrolladas en procesos previos, e implemente todas las capacidades de la solución en específico.

Aplicación Escritorio

Esta aplicación realiza todas las mismas funciones que realiza la aplicación PDA, pero en un ambiente de PC, y se utiliza para realizar pruebas de funcionalidad, sincronización y utilización de la solución tecnológica elegida en la etapa de investigación.

Resultados:

Dentro de los resultados entregables, se encuentra el prototipo debidamente desarrollado, aparte de los manuales necesarios para que estas aplicaciones se implanten debidamente.

Durante las pruebas, se utilizaron datos representativos No reales, y datos escalables reales basados en coordenadas válidas dentro del mapa cartesiano costarricense.

Aquí una pequeña imagen de las aplicaciones, lado a lado, después de un proceso de dibujo y sincronización en ambos lados.

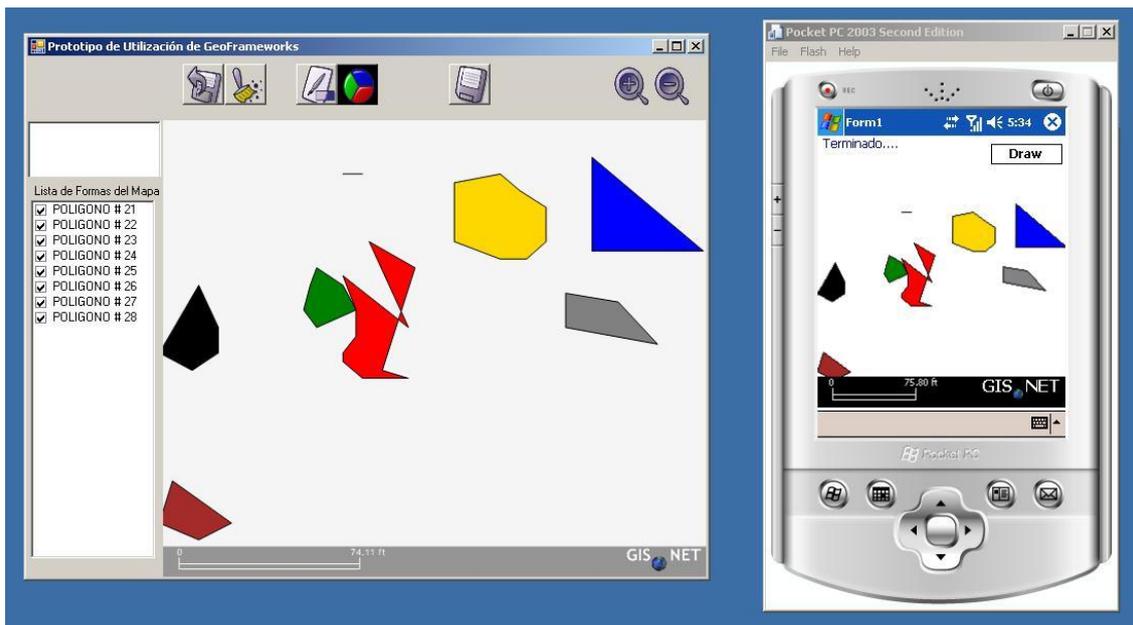


Ilustración 13 Imagen de Sincronización

6 Solución: Sistema Integrado de Conectividad y Consultas Municipalidad de San Carlos.

6.1 Proceso de Desarrollo

El sistema Integrado de Conectividad y Consultas, se subdivide en dos módulos que se han mencionado anteriormente.

Estos módulos son abordados desde una perspectiva un tanto distinta que el prototipo GIS – PDA, ya que en este caso, se utiliza una combinación mucho más exitosa de las metodologías analizadas en apartados anteriores, y se manejan desde distintas perspectivas dentro de los roles e involucrados dentro de la solución integral.

En específico se aborda la aplicación *de Lean Process* dentro de la parte administrativa del proyecto (disponibilidad de recursos, etc.).

Agile Development se aplica dentro de la coordinación de los procesos involucrados dentro del desarrollo de la solución (control de tareas, priorización de Stories, generación de entregables parciales, avances, reuniones con el (los) Cliente(s), planeamiento técnico general).

Xtreme Programming es aplicado en el día a día dentro de los procesos propios de desarrollo de la solución, ósea, los detalles meramente técnicos que son requeridos para poder realizar un correcto desarrollo de la solución necesaria por el (los) Cliente(s).

Test Driven Development, se utiliza en conjunto con *Xtreme Programming*, para trabajar sobre los procesos que son necesarios realizar. De este modo se realizan las pruebas de Unidad mucho antes de empezar a digitar código.

6.2 *Análisis del Proyecto*

6.2.1 **Requerimientos**

6.2.1.a **Features y User Stories**

Feature: Site

- **Login**

Como usuario, quiero poder ingresar a la sección de consultas utilizando mi número de cédula y una contraseña

- **Noticias**

Como usuario, quiero poder ver las noticias más recientes de la Municipalidad

- **Página de Registro**

Como usuario, quiero poder registrarme en el sitio para hacer uso de los servicios. La información a registrar es la información personal y de contacto del contribuyente. Además, debe permitirme editar los datos en cualquier momento.

- **Página de Contactos**

Cómo usuario, quiero poder ver una página con la información de contacto de la Municipalidad y un formulario de contacto

- **Página Principal**

Como usuario, quiero poder ver una página principal

- **Diseño de “Site”**

- **UI y framework**

Feature: Consultas

- **Bienes Inmuebles**

Como usuario, quiero poder consultar el monto a pagar por concepto del impuesto de Bienes Inmuebles utilizando mi número de cédula

- **Patentes**

Como usuario, quiero poder consultar el monto a pagar por concepto de Patentes utilizando mi número de cédula

- **Aguas**

Como usuario, quiero poder consultar el monto a pagar (pendiente) por concepto de Aguas utilizando mi número de cédula.

- **Mercados**

Como usuario, quiero poder consultar el monto a pagar por concepto de Mercados utilizando mi número de cédula.

- **Servicios**

Como usuario, quiero poder consultar el monto a pagar por concepto de Servicios de basura y limpieza de caños utilizando mi número de cédula

- **Cementerios**

Como usuario, quiero poder consultar el monto a pagar por concepto de Cementerios de basura y limpieza de caños utilizando mi número de cédula

Feature: WebServices - Entidades Bancarias

- **Cierre Diario**

Como entidad bancaria, quiero un WebService que me permita enviarle a la Municipalidad el lote de pagos realizados durante el día. La información a enviar es: usuario, fecha, recibo, serie, cédula y monto total.

- **Anulación de pago**

Como entidad bancaria, quiero un WebService que me permita anular un pago.

- **Realizar pago**

Como entidad bancaria, quiero un WebService que me permita registrar pagos de contribuyentes.

- **Consulta de pendiente**

Como entidad bancaria, quiero un WebService que me permita consultar el pendiente de un contribuyente por número de cédula

- **Consulta de pendiente Pago por Adelantado**

Como entidad bancaria, quiero un WebService que me permita consultar el pendiente de un contribuyente por número de cédula, generándolo hasta el mes de diciembre del año en curso.

- **Pago por Adelantado**

Como entidad bancaria, quiero un WebService que me permita pagar el pendiente Adelantado de un contribuyente por número de cédula

- **Anulación de Pago por Adelantado**

Como entidad bancaria, quiero un WebService que me permita anular el pago de un pago por adelantado.

- **Pago Parcial**

Como entidad bancaria, quiero un WebService que me permita pagar el parte de un contribuyente por número de cédula y un monto de dinero.

- **Anulación de Pago Parcial**

Como entidad bancaria, quiero un WebService que me permita anular el pago de un contribuyente por número de cédula

6.2.2 Modelo de Diseño – Diagrama Conceptual

A continuación, se muestran dos diagramas conceptuales de los elementos, procesos, entidades, subsistemas y otros involucrados en los dos módulos la solución. Tanto la sección de consultas en Línea, como la de Servicios Web, y su integración.

En el diagrama denominado como Ilustración 14, se muestra la representación en forma de procesos o funciones esenciales que debe realizar el Sistema de Consultas en línea.

En primera instancia, se tiene el “GUI-Sitio Web” que es la representación principal y canal de acceso a todo el sistema. En este Sitio Web, existen secciones de Información, Noticias, Publicaciones y el Diseño total del Sitio que son desarrolladas dentro de la misma solución.

Por otra parte, se tiene el acceso al sitio de Consultas en línea. Este acceso está basado en procesos de registro, depuración, autorización y otros.

Una vez completado este acceso, se procede a tener el acceso a un sitio de consultas en línea, el cual tendrá una serie de opciones de consulta para los contribuyentes. Estas opciones de consulta se resumen en cuatro grupos generales: Tributos, Históricos, Reportes, Misceláneos u Otros.

Todas estas opciones entregadas al Contribuyente, poseen características generales y comunes en cuanto a su implementación. Estas características generales son el mantenimiento adecuado de la consistencia y persistencia de los datos; además que se deben realizar procesos especializados para obtener la información del repositorio de datos.

Las últimas entidades se encargan de la parte análoga de la consistencia lógica y persistencia de los datos consultados.

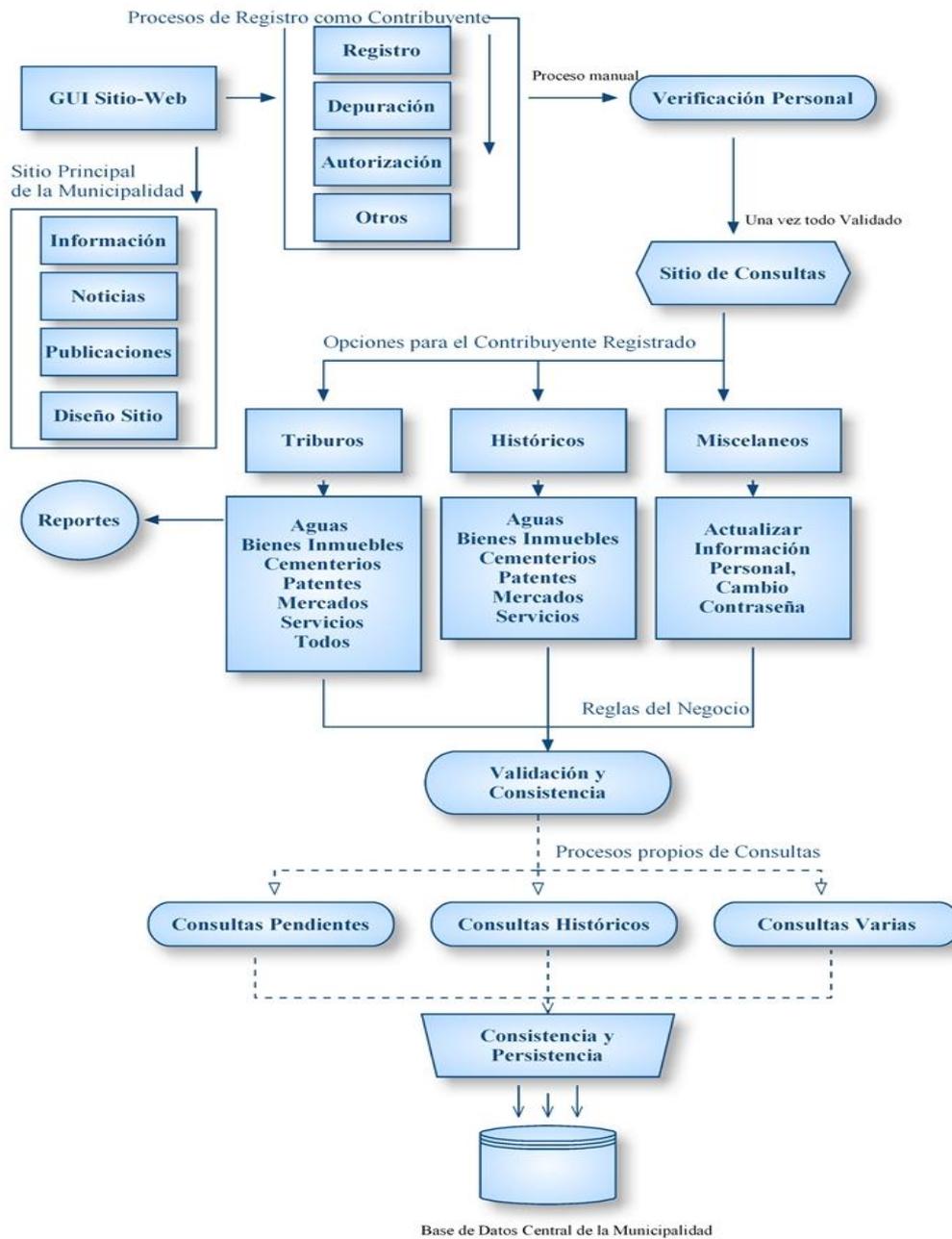


Ilustración 14 Modelo Conceptual Sitio de Consultas en Línea

En la Ilustración 15, se trata acerca de los Web Services (o Servicios Web), y su conceptualización.

En primera instancia, se debe considerar que son exactamente eso, servicios.

Estos servicios se publican por medio de un protocolo (como Internet) para ser consultados desde cualquier punto.

La entidad más importante en este tipo de sistemas, es la comunicación y el paso de mensajes desde un punto a otro. Estos puntos suelen llamarse Front-End y Back-End. A nivel del Front-End, se tiene como una entrada interpretada desde internet.

Esta entrada suele (y deber) ser codificada mediante algunos protocolos comunes (SSL) de certificación en el canal de comunicación. A esto se refiere la “Certificación SSL”.

Luego, los mensajes enviados y emitidos son y deben ser transparentes, orientados a objetos, extensibles, y coherentes. Es por eso que se utilizan plataformas orientadas a las transformaciones y manejo de mensajes SOAP.

Una vez hecho esto, se procede a una serie de filtros y procesos de certificación de seguridad adicional o lógica de la solución. Esto con el fin de evitar fuga de información sensible, posibles ataques externos, transacciones no autorizadas, u otros.

Es por eso que se aplican los niveles de seguridad de Codificación, Autenticación, Autorización y Certificación de Sesión para este aseguramiento de seguridad.

Luego se pasa a un filtro meramente de negocios, en el cual se verifican por condiciones especiales del contribuyente, se generan los registros necesarios en cualquier caso, se hace un manejo de excepciones más robusto, y se especializa el servicio requerido en consecuencia con los parámetros aportados.

Una vez realizado este paso, se realiza la transacción correspondiente y se manejan los posibles casos de excepción, en los cuales se retorna todo a un estado anterior.

Todas estas transacciones tienen una afectación real y consecuente sobre el repositorio central de datos.

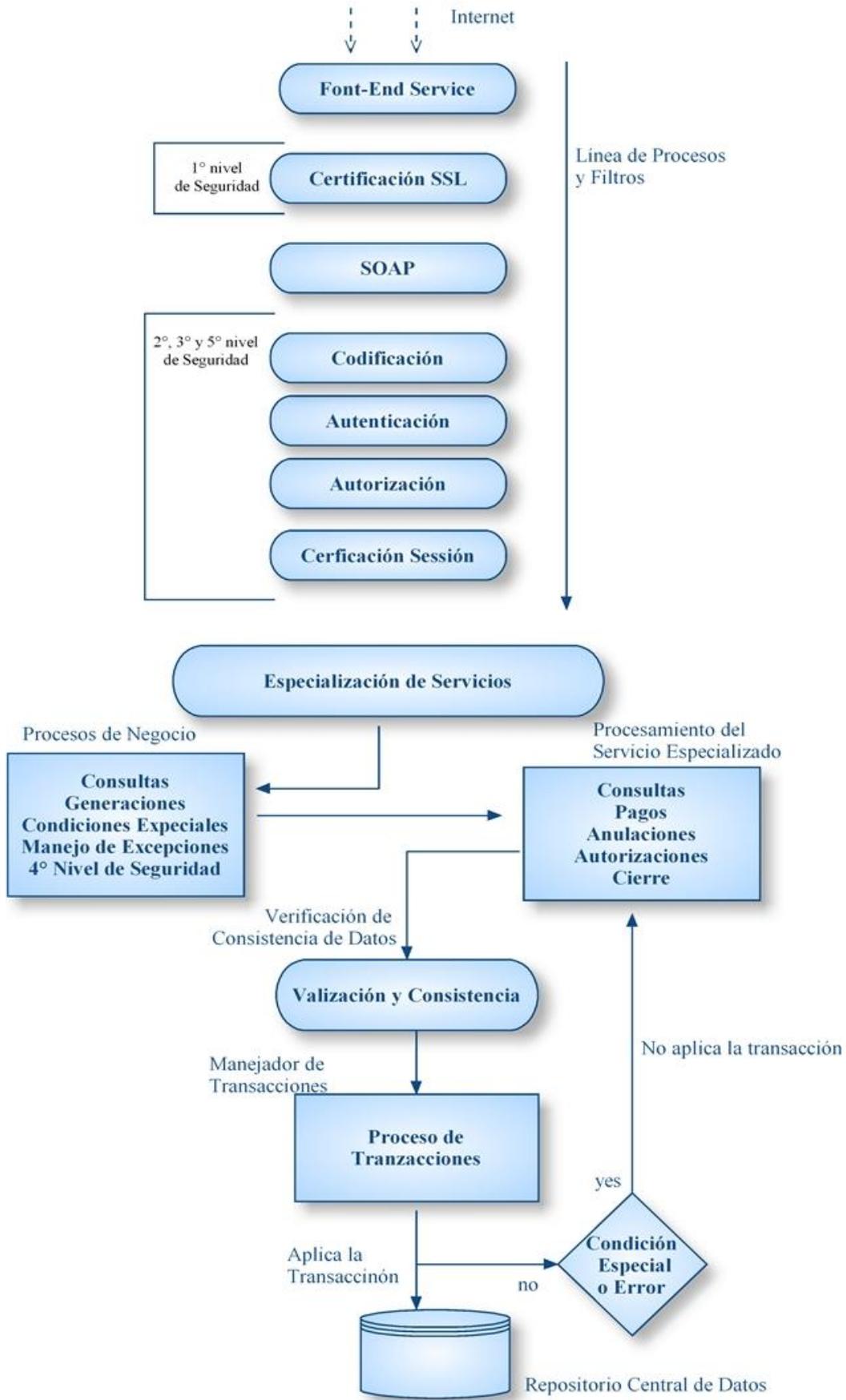


Ilustración 15 Diagrama Conceptual Servicios Web

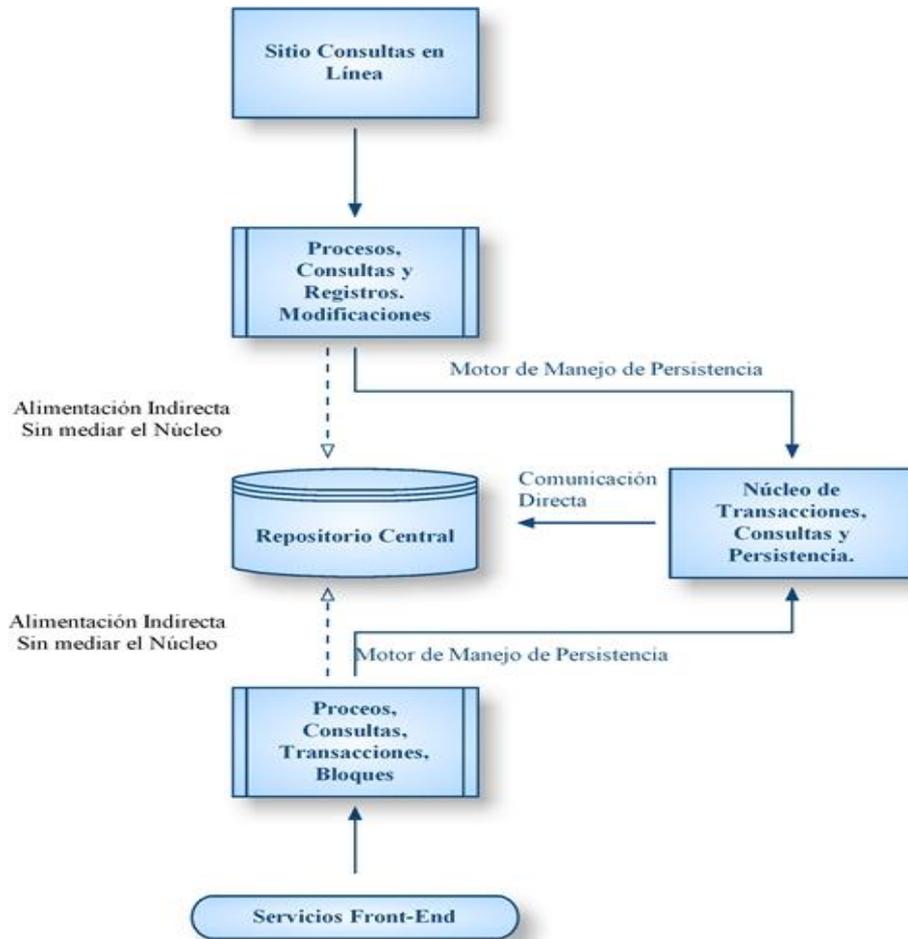


Ilustración 16 Diagrama de Integración Módulos

Por último, el diagrama de Integración de los módulos (Ilustración 16) expresa, a nivel muy macro, el comportamiento de cada Módulo con el repositorio central de datos, y como ambos Módulos comparten un núcleo de transacciones.

6.2.3 Diagrama User Stories

El siguiente, expresa las participaciones de los involucrados en ambos módulos.

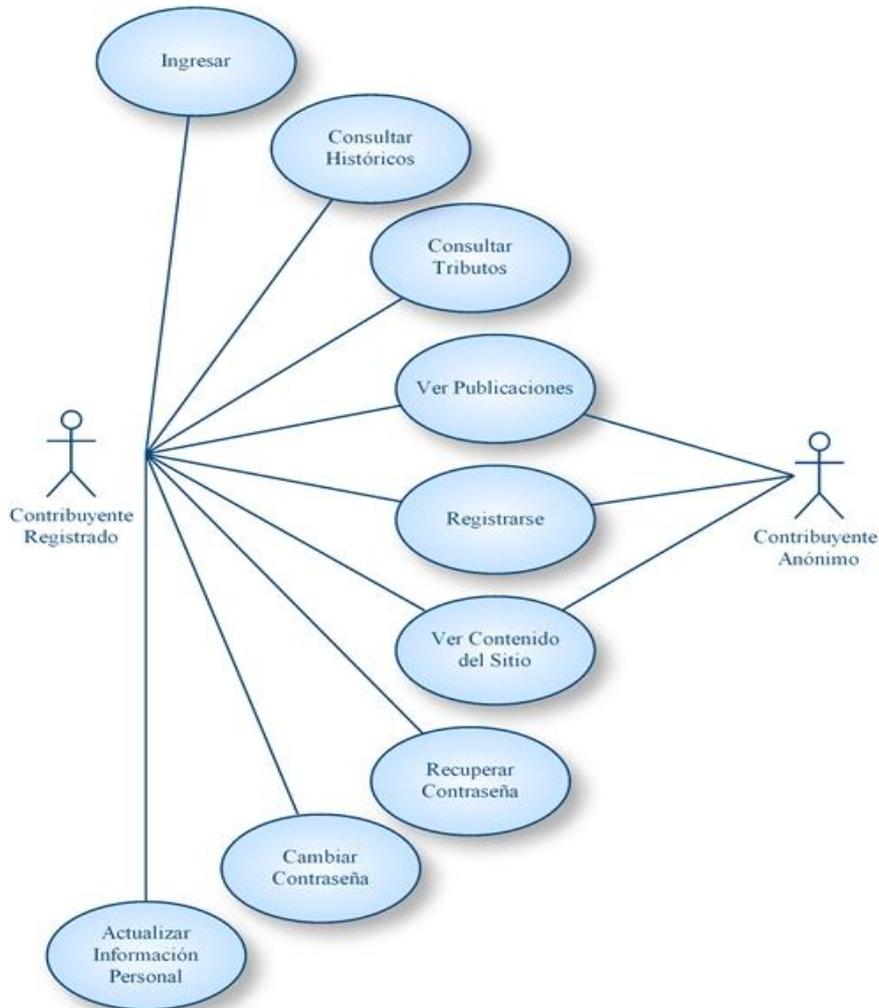


Ilustración 17 Diagrama de User Stories, Módulo Sitio de Consultas

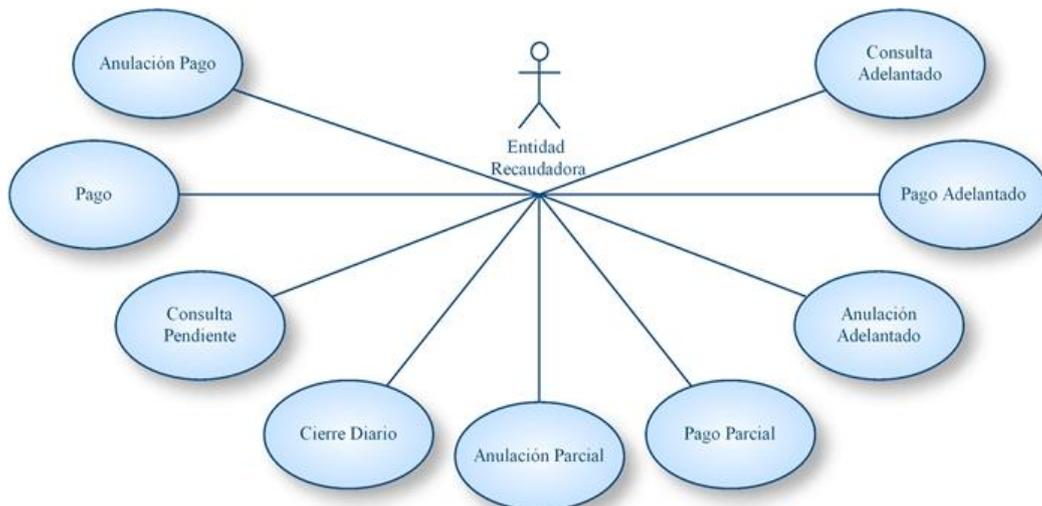


Ilustración 18 Diagrama User Stories, Módulo Servicios Web

6.2.4 User Stories

Feature: Site

User Story N°: 1

Nombre: Login.

Actor: Contribuyente Registrado.

Descripción: Como usuario, quiero poder ingresar a la sección de consultas utilizando mi número de cédula y una contraseña.

User Story N°: 2

Nombre: Noticias.

Actor: Contribuyente Anónimo.

Descripción: Como usuario, quiero poder ver las noticias más recientes de la Municipalidad

User Story N°: 3

Nombre: Página Registro.

Actor: Contribuyente Anónimo.

Descripción: Como usuario, quiero poder registrarme en el sitio para hacer uso de los servicios. La información a registrar es la información personal y de contacto del contribuyente. Además, debe permitirme editar los datos en cualquier momento.

User Story N°: 4

Nombre: Página de Contactos.

Actor: Contribuyente Anónimo / Contribuyente Registrado.

Descripción: Como usuario, quiero poder ver una página con la información de contacto de la Municipalidad y un formulario de contacto.

User Story N°: 5

Nombre: Página de Principal.

Actor: Contribuyente Anónimo / Contribuyente Registrado.

Descripción: Como usuario, quiero poder ver una página principal.

User Story N°: 6

Nombre: Diseño de Site.

Actor: Contribuyente Anónimo / Contribuyente Registrado.

Descripción: UI y framework.

Feature: Consultas

User Story N°: 7

Nombre: Bienes Inmuebles

Actor: Contribuyente Registrado.

Descripción: Como usuario, quiero poder consultar el monto a pagar por concepto del impuesto de Bienes Inmuebles utilizando mi número de cédula

User Story N°: 8

Nombre: Patentes

Actor: Contribuyente Registrado.

Descripción: Como usuario, quiero poder consultar el monto a pagar por concepto de Patentes utilizando mi número de cédula

User Story N°: 9

Nombre: Aguas

Actor: Contribuyente Registrado.

Descripción: Como usuario, quiero poder consultar el monto a pagar (pendiente) por concepto de Aguas utilizando mi número de cédula.

User Story N°: 10

Nombre: Mercados

Actor: Contribuyente Registrado.

Descripción: Como usuario, quiero poder consultar el monto a pagar por concepto de Mercados utilizando mi número de cédula.

User Story N°: 11

Nombre: Servicios

Actor: Contribuyente Registrado.

Descripción: Como usuario, quiero poder consultar el monto a pagar por concepto de Servicios de basura y limpieza de caños utilizando mi número de cédula

User Story N°: 12

Nombre: Cementerios

Actor: Contribuyente Registrado.

Descripción: Como usuario, quiero poder consultar el monto a pagar por concepto de Cementerios de basura y limpieza de caños utilizando mi número de cédula.

Feature: WebServices - Entidades Bancarias

User Story N°: 13

Nombre: Cierre Diario

Actor: Entidad Recaudadora.

Como entidad bancaria, quiero un WebService que me permita enviarle a la Municipalidad el lote de pagos realizados durante el día. La información a enviar es: usuario, fecha, recibo, serie, cédula y monto total.

User Story N°: 14

Nombre: Anulación de pago

Actor: Entidad Recaudadora.

Como entidad bancaria, quiero un WebService que me permita anular un pago.

User Story N°: 15

Nombre: Realizar pago

Actor: Entidad Recaudadora.

Como entidad bancaria, quiero un WebService que me permita registrar pagos de contribuyentes.

User Story N°: 16

Nombre: Consulta de pendiente

Actor: Entidad Recaudadora.

Como entidad bancaria, quiero un WebService que me permita consultar el pendiente de un contribuyente por número de cédula

User Story N°: 17

Nombre: Consulta de pendiente Pago por Adelantado.

Actor: Entidad Recaudadora.

Como entidad bancaria, quiero un WebService que me permita consultar el pendiente de un contribuyente por número de cédula, generándolo hasta el mes de diciembre del año en curso.

User Story N°: 18

Nombre: Pago por Adelantado.

Actor: Entidad Recaudadora.

Como entidad bancaria, quiero un WebService que me permita pagar el pendiente Adelantado de un contribuyente por número de cédula

User Story N°: 19

Nombre: Anulación de Pago por Adelantado

Actor: Entidad Recaudadora.

Como entidad bancaria, quiero un WebService que me permita anular el pago de un pago por adelantado.

User Story N°: 20

Nombre: Pago Parcial

Actor: Entidad Recaudadora.

Como entidad bancaria, quiero un WebService que me permita pagar el parte de un contribuyente por número de cédula y un monto de dinero.

User Story N°: 21

Nombre: Anulación de Pago Parcial

Actor: Entidad Recaudadora.

Como entidad bancaria, quiero un WebService que me permita anular el pago de un contribuyente por número de cédula

6.2.5 Diagramas de Secuencia

Feature: Site

User Story N°: 1

Nombre: Login.

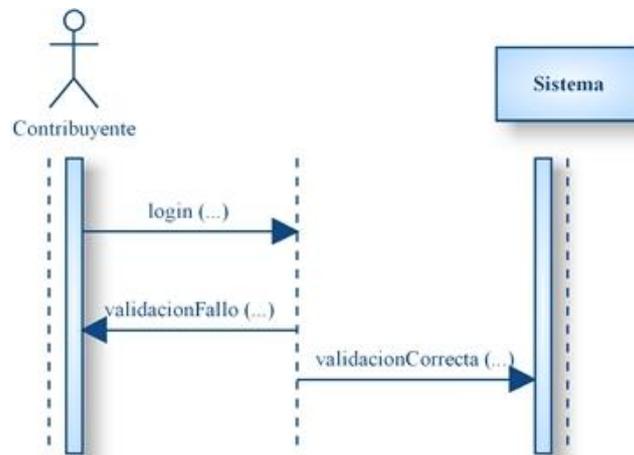


Ilustración 19 Diagrama de Secuencia User Story N° 1

User Story N°: 2

Nombre: Noticias.

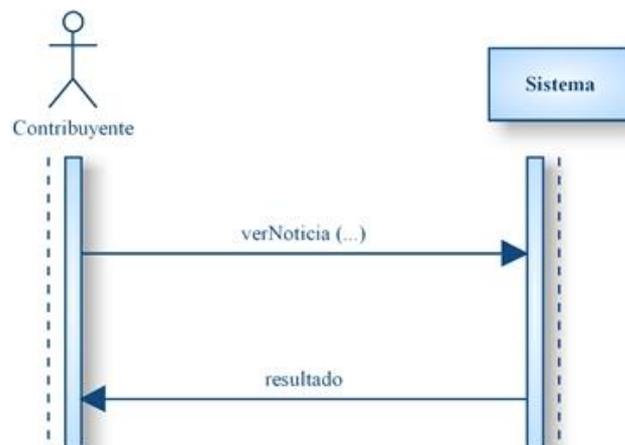


Ilustración 20 Diagrama de Secuencia User Story N° 2

User Story N°: 3

Nombre: Página Registro.

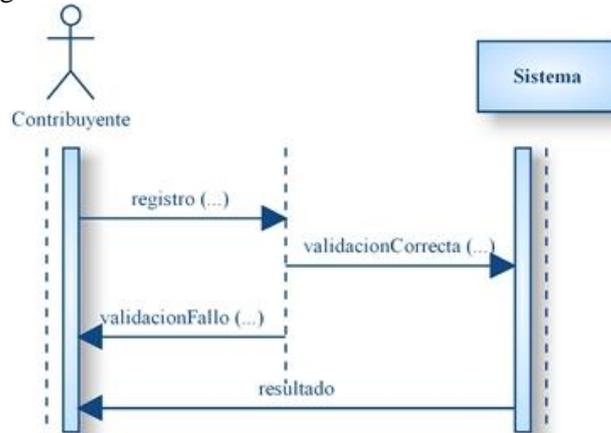


Ilustración 21 Diagrama de Secuencia User Story N° 3

User Story N°: 4

Nombre: Página de Contactos.

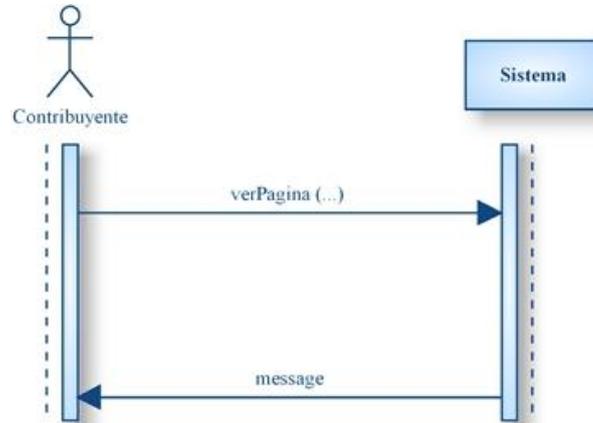


Ilustración 22 Diagrama de Secuencia User Story N° 4

User Story N°: 5

Nombre: Página de Principal.

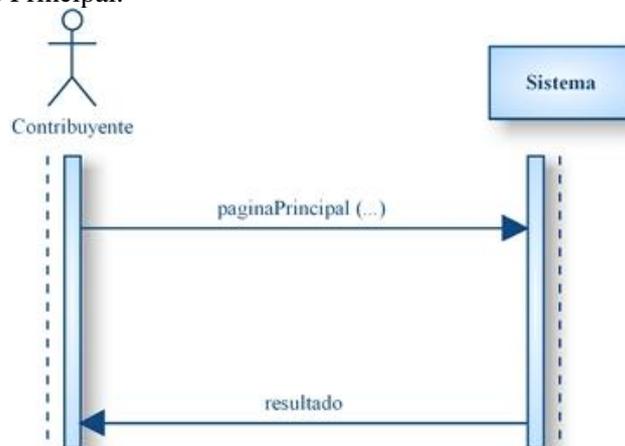


Ilustración 23 Diagrama de Secuencia User Story N° 5

Feature: Consultas

User Story N°: 7

Nombre: Bienes Inmuebles

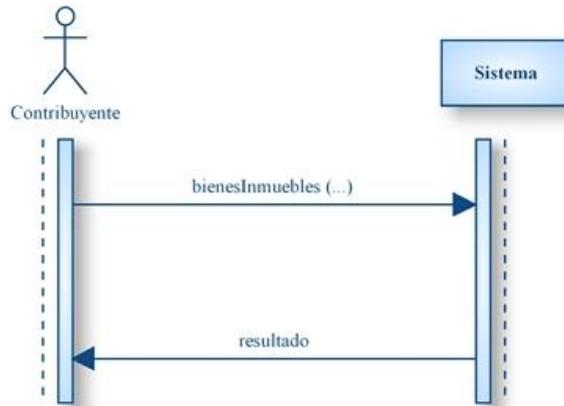


Ilustración 24 Diagrama de Secuencia User Story N° 7

User Story N°: 8

Nombre: Patentes

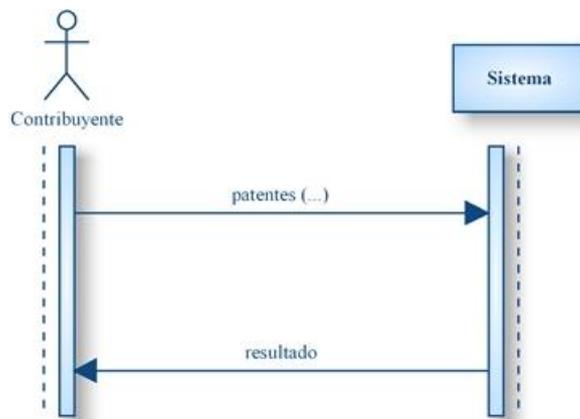


Ilustración 25 Diagrama de Secuencia User Story N° 8

User Story N°: 9

Nombre: Aguas

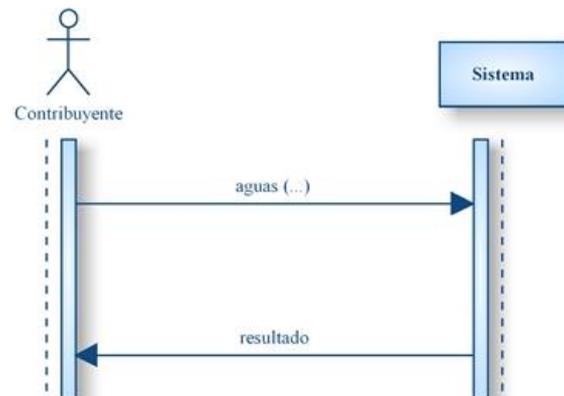


Ilustración 26 Diagrama de Secuencia User Story N° 9

User Story N°: 10

Nombre: Mercados

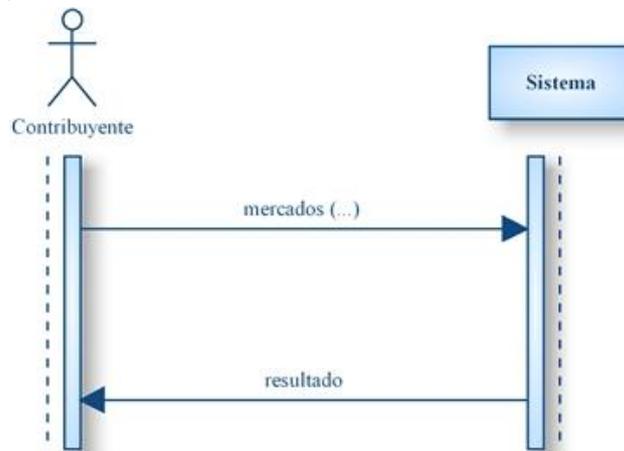


Ilustración 27 Diagrama de Secuencia User Story N° 10

User Story N°: 11

Nombre: Servicios

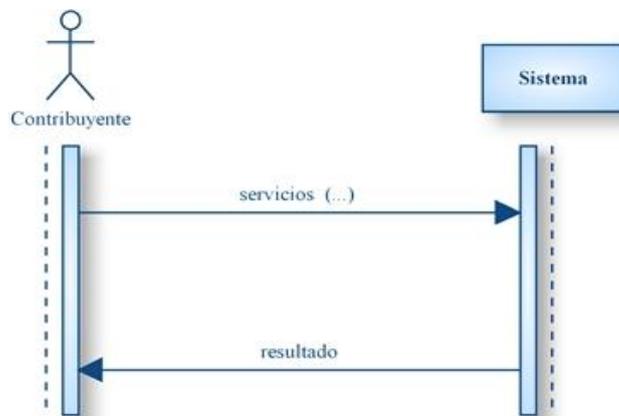


Ilustración 28 Diagrama de Secuencia User Story N° 11

User Story N°: 12

Nombre: Cementerios

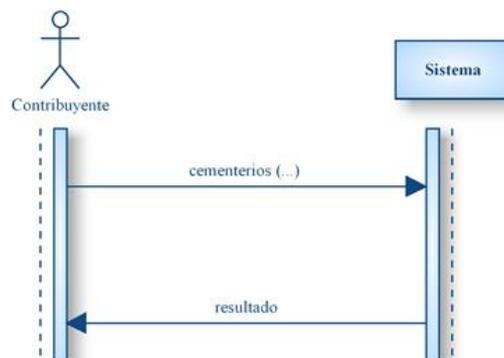


Ilustración 29 Diagrama de Secuencia User Story N° 12

Feature: WebServices - Entidades Bancarias

User Story N°: 13

Nombre: Cierre Diario

Actor: Entidad Recaudadora.

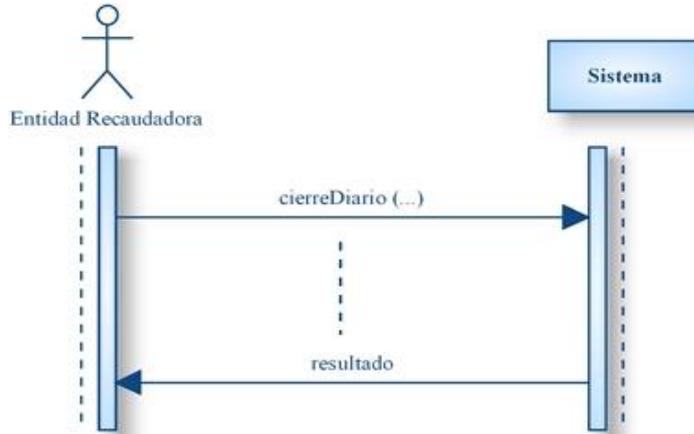


Ilustración 30 Diagrama de Secuencia User Story N° 13

User Story N°: 14

Nombre: Anulación de pago

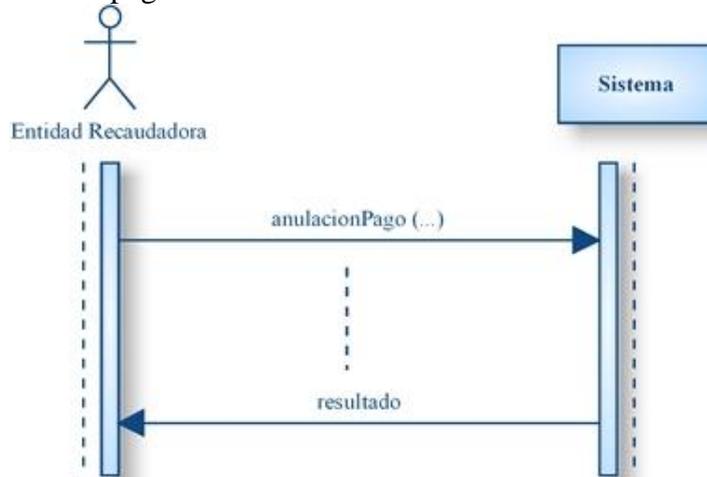


Ilustración 31 Diagrama de Secuencia User Story N° 14

User Story N°: 15

Nombre: Realizar pago

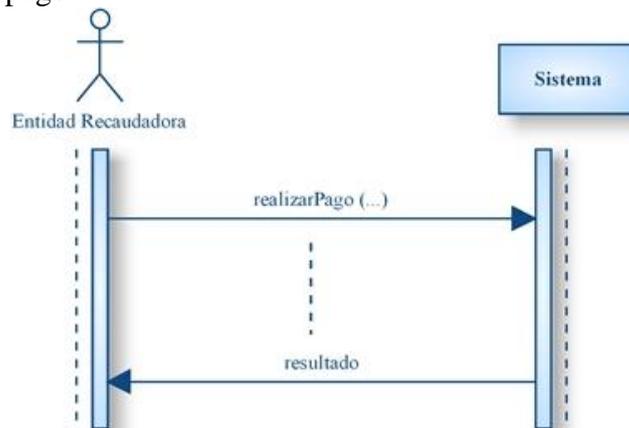


Ilustración 32 Diagrama de Secuencia User Story N° 15

User Story N°: 16

Nombre: Consulta de pendiente

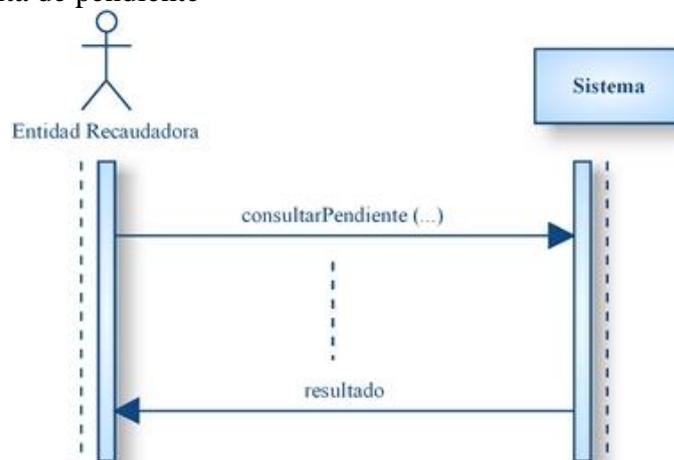


Ilustración 33 Diagrama de Secuencia User Story N° 16

User Story N°: 17

Nombre: Consulta de pendiente Pago por Adelantado.

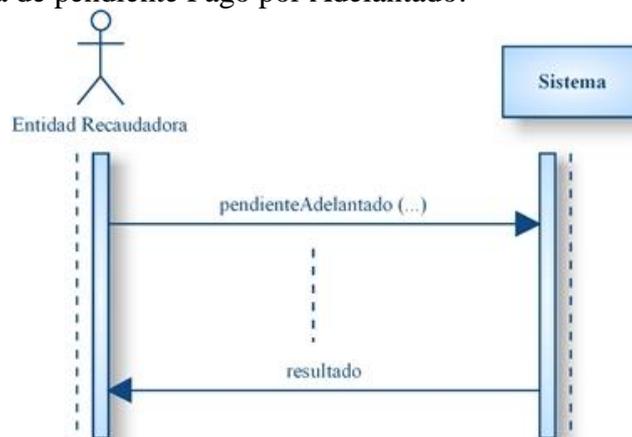


Ilustración 34 Diagrama de Secuencia User Story N° 17

User Story N°: 18

Nombre: Pago por Adelantado.

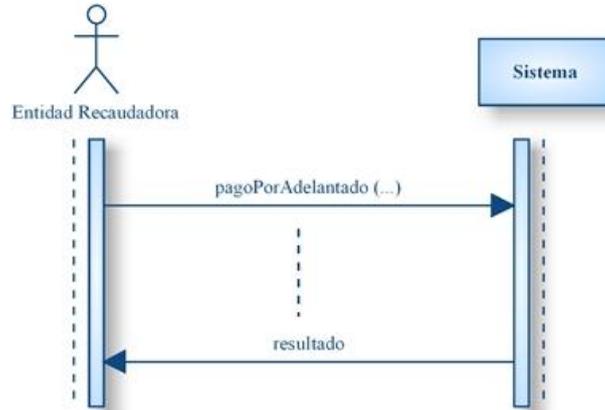


Ilustración 35 Diagrama de Secuencia User Story N° 18

User Story N°: 19

Nombre: Anulación de Pago por Adelantado

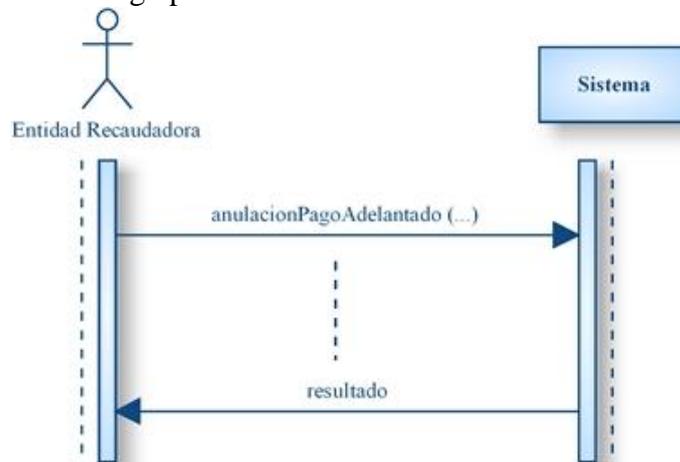


Ilustración 36 Diagrama de Secuencia User Story N° 19

User Story N°: 20

Nombre: Pago Parcial

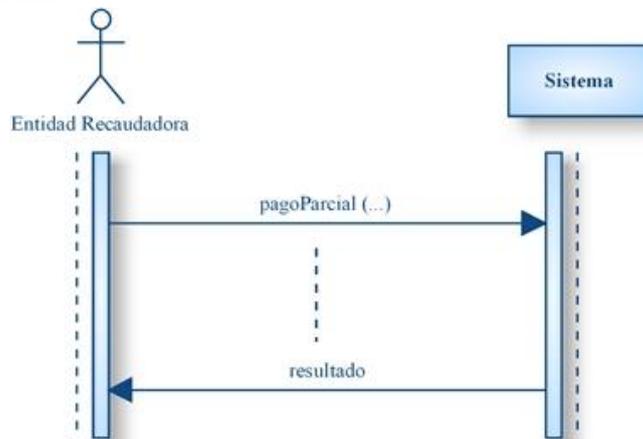


Ilustración 37 Diagrama de Secuencia User Story N° 20

User Story N°: 21

Nombre: Anulación de Pago Parcial

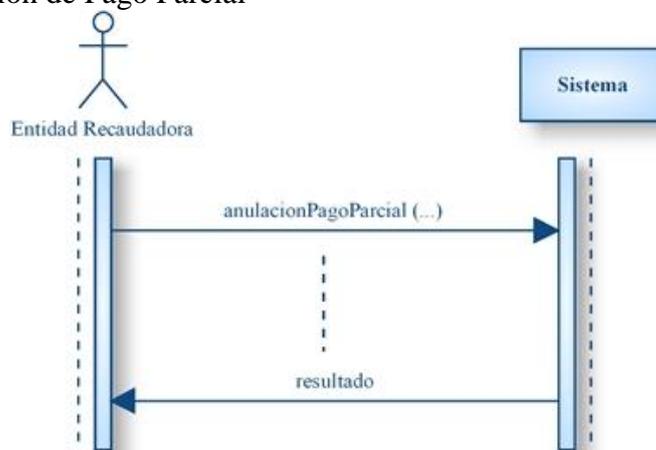


Ilustración 38 Diagrama de Secuencia User Story N° 21

6.2.6 Modelo de Dominio

En este caso, la Ilustración 39 expresa las distintas partes que entran en juego dentro de los dos Módulos, que poseen una Integración lógica con base en un repositorio central.

Por un lado, el contribuyente interactúa directamente con el Módulo de Consultas, denominado en el diagrama como “Sitio Web”, y por su puesto, con todos sus procesos hijos.

Por otro lado, el módulo de Servicios Web, posee una entidad llamada seguridad, debido a que entran en juego Factores externos que se encargan de verificar la seguridad a nivel de capa de transporte de los mensajes enviados y recibidos de / por la Municipalidad de San Carlos. Luego existen procesos hijos que también entran en juego dentro del dominio general.

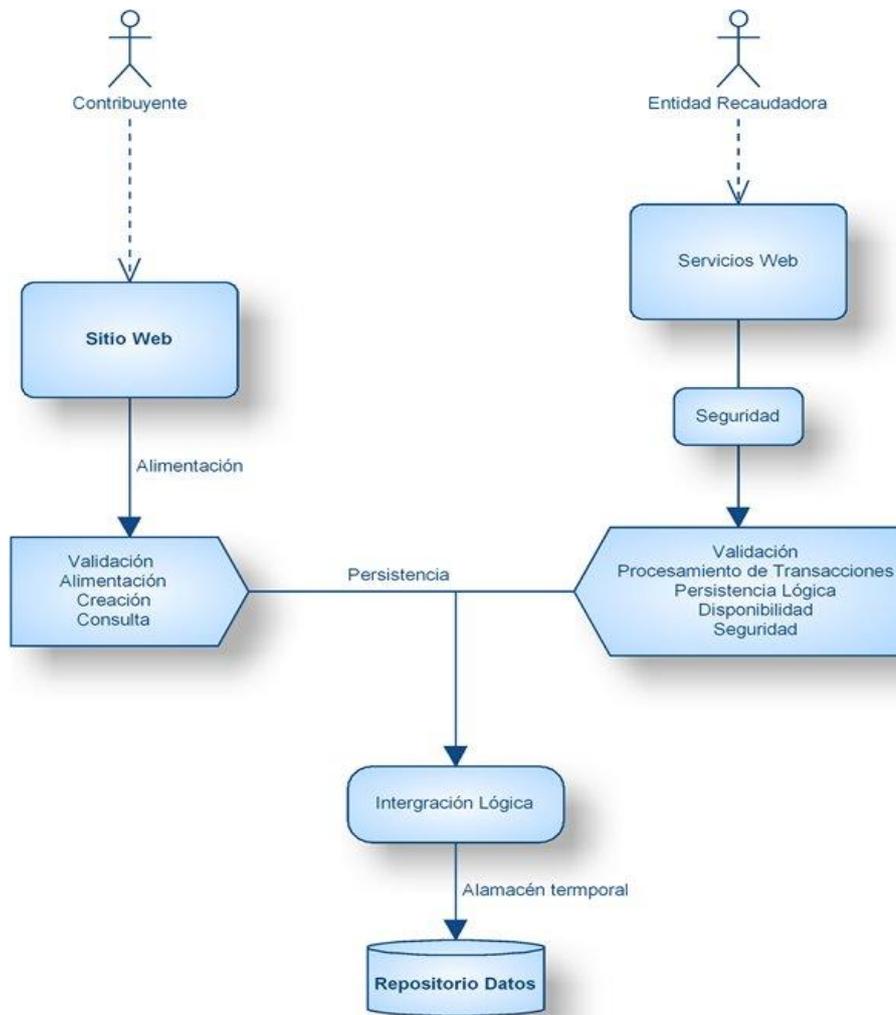


Ilustración 39 Diagrama de Dominio de la Solución

6.2.7 Diagrama de Clases

En la Ilustración 40, se expresa a grandes rasgos, la implementación de clases utilizada en esta solución.

En primera instancia, se tiene la parte de GUI, que es la que podrá ser vista por el Contribuyente de esta solución. Inmediata a esta capa, se tiene una abstracción y mapeo de eventos relacionados con la capa de presentación, en donde se involucran *frameworks* para tales usos como lo es *sitemesh*, *xwork*, y *webwork*. Estos se encargan (a muy grandes rasgos) de encapsular la visualización de páginas y convertirlas en eventos mediante xml, utilizar plantillas para evitar repeticiones de código y un mejor desarrollo, aplicación (o no) de plantillas a distintos tipos de acciones, y una colección impresionante de librerías que facilitan la aplicación y visualización de datos a nivel de lenguaje html.

Todos estos eventos, se transforman en Acciones o Procesos del sistema. Estas acciones pueden ser de tipo CRUD¹⁹ (en algunos casos), o simples reportes. Dentro de estas acciones también se adjunta la navegabilidad por todo el sistema y otros servicios paralelos como envío de correos electrónicos y otros.

Estas acciones a su vez, disparan Eventos que se transforman en transacciones propias de procesos de negocio, en las cuales se deben mantener condiciones específicas, estados especiales y otros. Todo esto utiliza las clases propias de objetos de negocio, que a su vez realizan procesos de acceso a datos mediante DAO's hacia un repositorio central.

También se utilizan algunas clases y paquetes útiles o misceláneos, que se utilizan de forma común o muy rara vez, dentro del sistema. Estas pueden ser manejo de sesiones, generación de reportes, conversiones de tipos, transformaciones varias, Criptografía (seguridad), y otras relacionadas con validación mediante Expresiones Regulares.

Paralelo a todo el sistema, se encuentra un robusto sistema de registro de acciones mediante *logs* que es inherente a cualquier proceso o clase dentro de la solución.

¹⁹ CRUD: Create, Retrieve, Update, Delete: Crear, consultar, actualizar o eliminar.

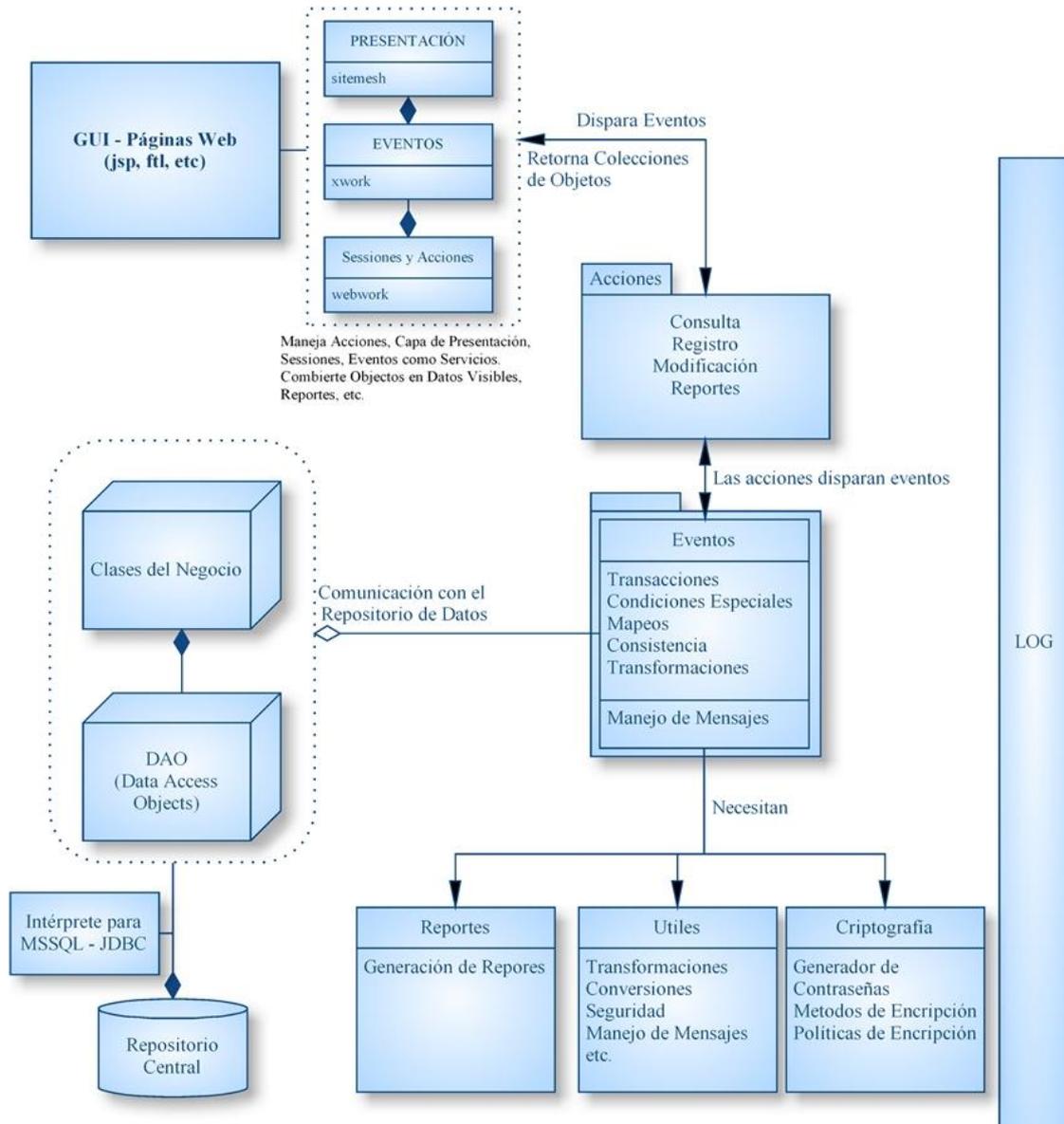


Ilustración 40 Diagrama de Clases. Módulo Sitio Web

En el diagrama 41 (Ilustración 41), se representa la interacción de las clases dentro de la solución Web Services.

En primer término, las capas SSL y SOAP son importantes de mencionar, ya que aunque no se desarrollen sistemas para interpretar y utilizar este tipo de tecnologías, sí se deben implementar configuraciones específicas para el uso de estas; así como incluir algunas librerías.

Luego de esto, la interpretación de los servicios es inherente a la publicación de los servicios.

Una vez completada esta etapa, se pasa a las clases de seguridad, en las cuales se procesan las peticiones de servicio mediante filtros de autenticación, autorización, certificación, transparencia y persistencia de los datos de entrada. Estos procesos de seguridad utilizan otros servicios de mensajería y funciones comunes de criptografía especializada para esta solución.

Luego que se ha asegurado la seguridad de los datos, suceden una serie de servicios de validación de la entrada de los datos, en donde se verifican las entradas por su congruencia, reglas específicas del negocio, estados especiales y otros.

Cuando se ha pasado este filtro, se realiza(n) la(s) transacciones necesarias. Estas transacciones utilizan acceso a clases de Negocio que a su vez utilizan DAO's de acceso a un repositorio central.

También estos procesos utilizan clases útiles o misceláneas.

Y al igual que el módulo de Consultas, se utiliza un robusto sistema de registro de *logs*.

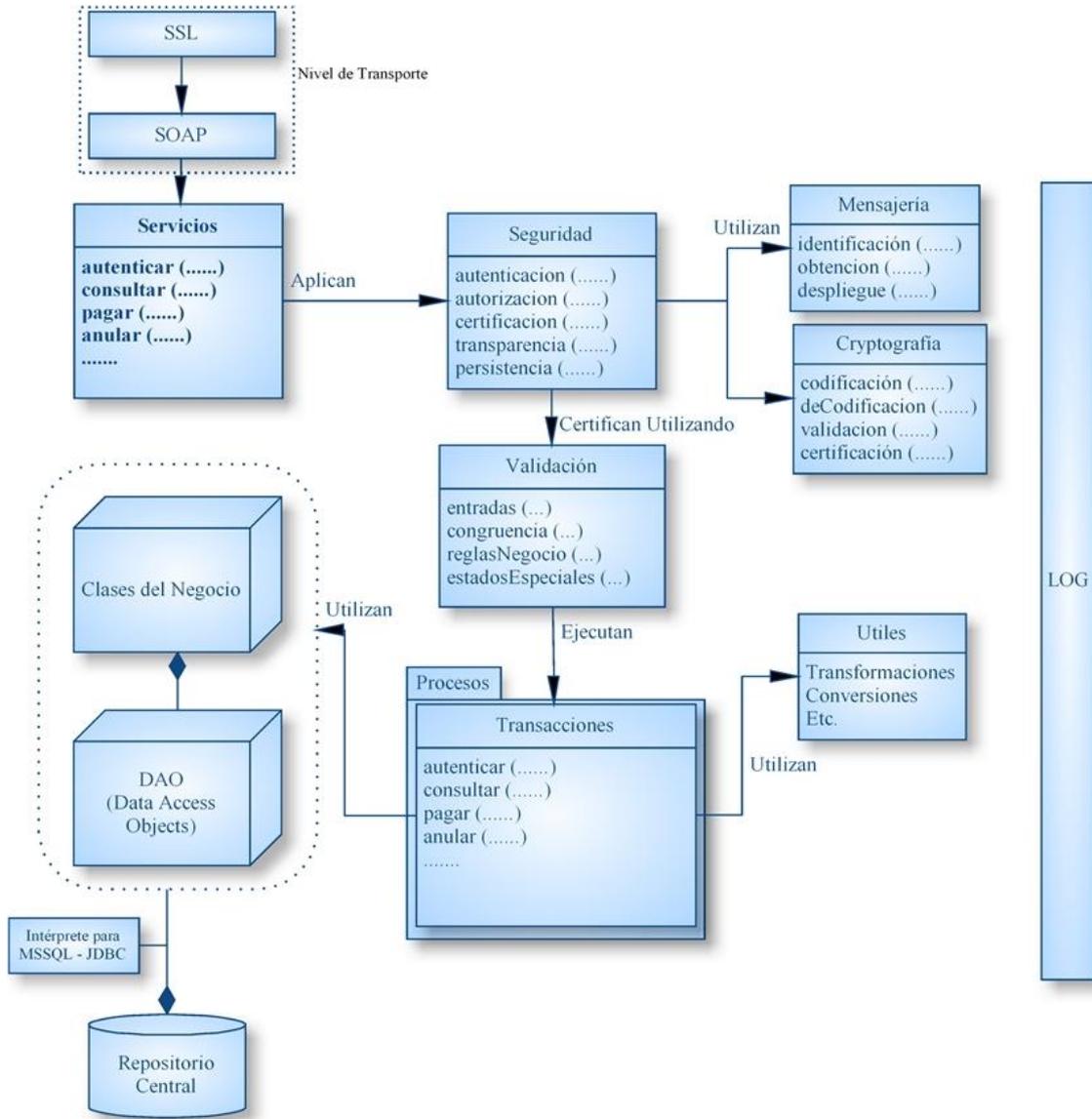


Ilustración 41 Diagrama de Clases. Módulo Web Services

6.2.8 Implementación

Para la implementación de esta solución, en ambos módulos, se recurre a las tecnologías Java, específicamente J2EE (Enterprise Edition), J2SE (Standard Edition), JSSE (Java Security Socket Extensión), y las librerías y *Frameworks* como lo son: WebWork, XWork, Jasper Reports, SiteMesh²⁰ y otros.

6.2.8.a Módulo de Sitio Web

Capa de Presentación

Para este módulo, se aplican las tecnologías de la familia de OpenSymphony, para el manejo de eventos disparados por acciones de usuario vía Web. Esto significa que se abstrae totalmente la estructura interna de la aplicación Web, dando paso a una mayor seguridad en cuando a control de acceso, alteración indebida del sitio, fuga de información importante, autorización de acciones y prácticas como inyección de código.

También el uso de tecnologías de este tipo, permite una gran extensibilidad del sitio y soporte para el paso del tiempo, puesto que utiliza configuraciones globales basadas en ficheros XML y plantillas pre-compiladas, por lo que al cambiar una sola plantilla, se puede cambiar el comportamiento total del sitio, sin aplicar re-ingeniería. Además supone de las herramientas adecuadas para realizar validaciones de formularios por expresiones regulares y otros al lado del cliente, y esto es una gran mejora en rendimiento y funcionalidad de las aplicaciones Web.

Otro aspecto importante, es que este tipo de tecnologías se basan en la aplicación de Acciones a Eventos específicos. Estas acciones pueden estar condicionadas por aspectos de sesión, autorización u otras variables del ambiente local. Estas pre-condiciones pueden ser configuradas para una o un conjunto de acciones en específico, y a nivel global. Estas acciones suelen ser llamadas “*Interceptors*”, y realizan tareas de interceptar las acciones antes de “disparar” otras, por lo que se pueden configurar estas variables y realizar otras acciones varias.

²⁰ Frameworks de la Familia de Opensymphony Organization.

Un detalle muy importante, es que este tipo de “*Frameworks*” ha sido previamente configurado para evitar acciones de tipo indebidas, y evita actividades como Ejecución de Código entre sitios, Inyección de Código, Fuga de Información sensible.

Aparte, desliga completamente la lógica de negocios y acceso a datos totalmente, pues utiliza objetos y colecciones genéricas de objetos para el manejo de datos a nivel de presentación, por lo que no es necesario utilizar sentencias de código incrustadas en los ficheros que contienen código meramente de presentación.

Otra facilidad, es que por medio de “*SiteMesh*”, es posible aplicar plantillas de código para toda la aplicación Web, o una parte de ella. Por lo que no es necesario repetir todo el código de presentación en cada uno de los ficheros destinados para esto, y posee otra ventaja que es extensibilidad.

Los ficheros orientados exclusivamente a presentación, mantienen el estándar de organización según

Capa de Negocios

Con respecto a la Capa de negocios, se utiliza código nativo de Java para las acciones de validaciones por expresiones regulares, validaciones más complejas, Casos especiales y Excepciones dentro de las reglas del negocio y otros.

Se maneja una capa de validación como abstracción intrínseca de cada acción o evento que se desea realizar.

Esta capa de Negocios también realiza control de Transacciones lógicas, comunicándose con la capa de acceso a datos utilizando así mismo “*Frameworks*” destinados para tal hecho, tal es el caso de “*Hibernate*” y por otro lado, la inyección de objetos entre distintos formatos (Objetos a/hacia XML, HTML a/hacia Objetos, etc.) es manejado por *Spring*.

Esta capa de Negocios también tiene el deber de manejar sesiones de persistencia y acceso a datos, de modo que las transacciones lógicas se realicen de la mejor forma, con la mejor administración de recursos, con la caducidad adecuada y con el manejo de excepciones o eventos correcto. Es por eso que se implementan servicios y Clases llamadas *Factories* para el manejo por sesión de los mecanismos de acceso a datos.

Los archivos orientados a capas de lógica de negocio, mantienen el estándar de organización de paquetes de Java, con implementación propia de Northek Software.

Capa de Acceso a Datos

Debido a la naturaleza de la información del proyecto es imposible dar una aproximación real al trabajo que se realiza en esta área. Pero el uso de e herramientas para acceso a datos por medio de Objetos, *Factores* y otros, facilita en gran medida el trabajo y paradigma de persistencia.

Dentro de los detalles que se pueden revelar, es que estas herramientas utilizan mapeos de Tablas simples o Interrelacionadas, en Objetos y Colecciones de Objetos a nivel lógico de aplicación. Por ejemplo. Una tabla Usuario, va a ser mapeada por un Objeto Usuario, y deberá tener sus equivalencias de tipo.

Para la implementación de estos Objetos a nivel de tablas, se siguen ciertas políticas y reglas.

1. Cada Objeto Mapeado deberá tener una clase homóloga en capa de Negocios.
2. Cada Objeto Mapeado deberá tener un archivo XML en el cual se especifiquen las equivalencias lógicas y físicas de cada tipo de dato, nombre alterno y propiedades o atributos.
3. Todas las consultas No almacenadas en Base de Datos a nivel de Procedimiento Almacenado, que se manejen en capa de Negocio, deberán existir dentro de este u otro archivo XML, y deberá proceder con el estándar adecuado del lenguaje en el que se encuentra escrito.
4. Los DAOS no tendrán lógica de Negocio de ningún tipo.

Los archivos orientados a capas de Acceso a Datos, mantienen el estándar de organización de paquetes de Java, con implementación propia de Northek Software

Capa de Servicios

Como parte de la robustez característica de una aplicación de esta familia, se mantiene un servicio completamente paralelo a todas las capas de cualquier proceso, el cual es invocado para almacenar en archivos de bitácora, alguna acción que resulte importante.

6.2.8.b Módulo Servicios Web

Debido a la naturaleza de una aplicación basada en Servicios, y la imperante necesidad de Seguridad, esta aplicación adjunta dos capas muy poco utilizadas: Capa de Transporte y Capa de Seguridad. Esta última suele ser inherente a la capa de negocios, pero para efectos de escalabilidad se decidió realizar como una capa más dentro de la aplicación.

Capa de Transporte

En esta capa, se atienden necesidades de seguridad meramente. Ya que esta capa de transporte es la encargada de la comunicación vía el medio Web, entre las entidades Recolectoras, y la Municipalidad de San Carlos.

En esta capa, se aseguran detalles como la seguridad y encriptación de los datos mediante Certificados de Autenticidad Internacionalmente aceptados basados en SSL, llaves públicas y privadas generadas utilizando algoritmos como AES, SHA y otros.

Capa de Presentación

En este punto, es necesario aclarar que un Servicio Web puede ser visto como un hilo que “escucha” peticiones. Para que este hilo sea correctamente “disparado”, la llamada debe darse con un formato específico. Este formato utilizado es usualmente un formato basado en XML, y en esta ocasión, se ha elegido SOAP como lenguaje de intercambio.

Para que las entidades Recaudadoras puedan ejecutar bien sus transacciones, deben poseer el formato de estos mensajes SOAP que son interpretados, y para esto se generan dos ficheros:

- Uno con extensión WSDL
- Otro con extensión DTD

La estructura de ambos archivos no viene al caso, pero lo importante de ellos es que son la única forma como pueden ser llamados los Servicios Web de forma correcta.

Capa de Seguridad

Debido a la naturaleza de las transacciones que se realizan en estos Servicios Web, es muy importante la protección de la información que se envía y recibe, tanto como la veracidad de la información que se transfiere. Es por eso que se utilizan varias políticas de encriptación de datos, validación de sesiones, certificados generados en un tiempo / fecha específicos, políticas para la codificación y de-codificación de los datos

de forma correcta, validación de datos de entrada, y protección de los datos a través de canales no seguros.

Es por eso que para todas estas políticas y restricciones, se ha realizado una capa exclusiva que se encarga de validar, decodificar, certificar, etc., todos los datos que se envía y reciben.

Capa de Negocios

En esta capa, es donde se realizan los procesos propios del negocio, algunas validaciones propias de cada entrada o dato que se recibe, y es donde se ejecutan las transacciones atómicas por petición.

Capa de Acceso a Datos

Esta capa es similar al módulo de Sitio Web, debido a que se utilizan las mismas herramientas y “*Frameworks*”, además que se rigen mediante las mismas reglas.

Capa de Servicios

Como parte de la robustez característica de una aplicación de esta familia, se mantiene un servicio completamente paralelo a todas las capas de cualquier proceso, el cual es invocado para almacenar en archivos de bitácora, alguna acción que resulte importante.

6.2.9 Resultados:

Los resultados palpables de este proceso de desarrollo de estos módulos, son varios:

1. Un Sitio Web, encargado de realizar consultas en línea sobre información del contribuyente. (Ver anexos adjuntos)
 - Sitio Web de Ingreso para consultas. (sitio.jpg)
 - Consulta de los Tributos Pendientes (pendiente.jpg)
 - Consulta de Tributos Al Cobro. (tributos.jpg)
 - Consulta sobre Histórico de Pagos (historico.jpg)

2. Un Sistema Integrado de Transacciones por medio de Servicios Web, totalmente seguro.

Debido a que en un sistema de esta naturaleza no interviene u usuario final, no posee una representación gráfica. Es por esto que se ilustra por medio de un diagrama de flujo de la información y mensajes que se envían desde las Entidades Recaudadoras, hacia la Municipalidad y Viceversa.

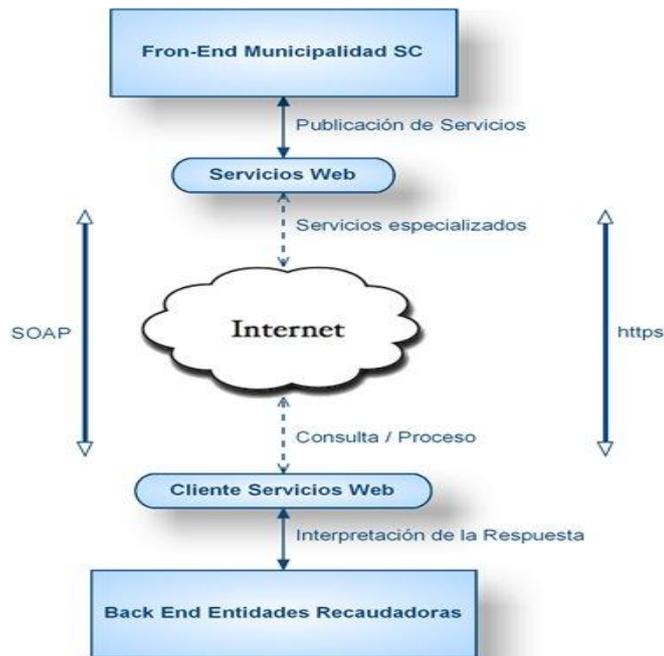


Ilustración 42 Diagrama de Servicios Web

7 Conclusiones

En referencia la aplicación de la metodología expuesta y analizada en este documento, se puede rescatar a un nivel general, el hecho de que estos sistemas completamente distintos, han puesto en perspectiva y en análisis la forma de utilizar la metodología, lo compleja que se puede tornar para proyectos sencillos, pero también destaca la flexibilidad existente para acoplarse a necesidades específicas de un proyecto determinado.

Como se puede observar, la metodología teórica, se convierte en una guía para la realización de las distintas medidas en pro del desarrollo armonioso de cualquier proyecto, y no encapsula al ingeniero, en un paradigma rígido de requerimientos, procesos e inclusive formas de trabajo. Sino que le brinda la libertad de alimentar el proceso, ajustar a lo necesario, rectificar su trabajo, y ver entregables a corto plazo. Lo que significa en procesos cortos de estimación y planeamiento, entregables rápidos y una mayor satisfacción del cliente final al poder ver un avance real integrado de todo el (los) módulo (s).

En cuestiones meramente técnicas, ambos proyectos han aportado mucho conocimiento desde el punto de vista investigativo y productivo. Esto se debe a que se han investigado productos, librerías y otros que agilizan el trabajo del Ingeniero de Software, de manera espectacular, de modo que no se debe invertir más tiempo del necesario, en dejar algo completamente funcional, y se aprovecha todo el trabajo ya hecho y probado por terceros, dentro de un esquema propio y regulado.

Por otro lado, la experiencia aportada por los proyectos, es impresionante. Por un lado el manejo de recursos que requiere una aplicación de escritorio nunca será el mismo que una aplicación sobre un dispositivo móvil, la sincronización, acceso a datos, tiempos de respuesta, etc., son factores de gran importancia por considerar.

El manejo de soluciones web, posee otros comportamientos y generalidades especiales que se deben abordar de una manera completamente distinta. En esta se deben tomar en cuenta muchos aspectos de estética, rendimiento, tiempos de espera, facilidad, simplicidad, seguridad, controles de acceso; que suelen ser dejados de lado en aplicaciones de escritorio o similares.

La aplicación de soluciones orientadas al servicio, también tiene sus enseñanzas, ya que sus aspectos de seguridad son de vital importancia, la concurrencia y completitud de transacciones es otro factor indispensable. En este tipo de soluciones, no existe una

interfaz directa orientada al servicio, se da una comunicación entre otros sistemas, más sin embargo, este tipo de condiciones lo deben hacer robusto, confiable y transparente.

Para finalizar, la ejecución, planeación y desenvolvimiento de estos proyectos, ha dejado una marca muy importante a nivel tecnológico de experiencia laboral; pero también deja enseñanzas de vida, experiencias que solamente estando en situaciones reales, donde intervienen muchas características personales no transmisibles, modos, acciones y en fin, un sinnúmero de elementos. Aquí es cuando se forma el verdadero profesional de forma integral. Es en este punto donde la parte tecnológica se aplica, para convertirse en una sola, inherente al pensamiento y actual de la persona, y donde el programador o estudiante, se convierte en un ingeniero integral con su ambiente, en donde sabe manejar de una u otra forma las oportunidades y problemas que se le presentan, y los resuelve de la forma más eficaz posible. Este tipo de enseñanzas solamente se aprenden con el paso del tiempo y estando en el medio.

8 Bibliografía

[Ficher, RB. 2004]. Fisher, RB. 2005, *Desarrollo y Gestión de Proyectos Informáticos*, Gestión de Riesgos. Barcelona.

[Robert C 2005]. Robert C. Martin Series. 2005, *Agile Java(TM): Crafting Code with Test-Driven Development*,

[James, 2004]. Highsmith James A. 2004, *Agile Project Management: Creating Innovative Products*. [s.I], Boston, Addison-Wesley.

[IEEE 1998]. IEEE Project Team. 1998, *IEEE-SA Standards Board Operations Manual*, [Especificación en Línea]. capítulo 8. Sección 8.2. Sitio Web: <http://standards.ieee.org/guides/opman/sect8.html#8.2>