



**UNIVERSIDAD TÉCNICA DEL NORTE**  
**FACULTAD DE INGENIERÍA EN CIENCIAS APLICADAS**  
**CARRERA DE INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**TEMA**

**“IMPLEMENTACION DEL SISTEMA DE CONTROL Y CALIFICACION DE PROVEEDORES  
PARA COOPERATIVAS DE AHORRO Y CRÉDITO”**

**Autor:** Daniel Fernando Cuarán Guerrero

**Director:** Ing. Mauricio Rea

**Ibarra – Ecuador**

**2014**

## CERTIFICACIÓN

Certifico que la Tesis “IMPLEMENTACION DEL SISTEMA DE CONTROL Y CALIFICACION DE PROVEEDORES PARA COOPERATIVAS DE AHORRO Y CRÉDITO” ha sido realizada en su totalidad por el señor: Daniel Fernando Cuarán Guerrero portador de la cédula de identidad número: 1002880662.

A handwritten signature in blue ink, appearing to read 'Mauricio Rea', enclosed within a circular scribble.

---

Ing. Mauricio Rea  
Director de la Tesis

## CERTIFICACIÓN

Ibarra, 02 de Marzo del 2015

Señores

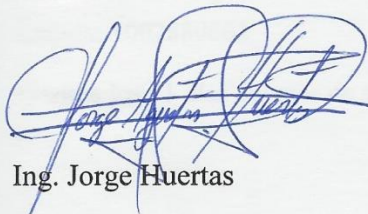
UNIVERSIDAD TÉCNICA DEL NORTE

Presente

De mis consideraciones.-

Siendo auspiciantes del proyecto de tesis del Egresado CUARÁN GUERRERO DANIEL FERNANDO con CI:1002880662 quien desarrollo su trabajo con el tema: **“IMPLEMENTACIÓN DEL SISTEMA DE CONTROL Y CALIFICACIÓN DE PROVEEDORES PARA COOPERATIVAS DE AHORRO Y CRÉDITO”** , me es grato informar que se han superado con satisfacción las pruebas técnicas y la revisión de cumplimiento de los requerimientos funcionales, por lo que se recibe el proyecto como culminado y realizado por parte del egresado CUARÁN GUERRERO DANIEL FERNANDO. Una vez que hemos recibido la capacitación y documentación respectiva, nos comprometemos a continuar utilizando el mencionado aplicativo en beneficio de nuestra empresa.

El egresado CUARÁN GUERRERO DANIEL FERNANDO puede hacer uso de este documento para los fines pertinentes en la Universidad Técnica del Norte.



Ing. Jorge Huertas

Gerente General

Cooperativa Financiera Atuntaqui Ltda.





**UNIVERSIDAD TÉCNICA DEL NORTE**  
**CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE**  
**INVESTIGACIÓN**  
**A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE**

Yo, DANIEL FERNANDO CUARAN GUERRERO, con cedula de identidad Nro. 1002880662, manifiesto mi voluntad de ceder a la Universidad Técnica del Norte los derechos patrimoniales consagrados en la ley de propiedad intelectual del Ecuador, articulo 4, 5 y 6, en calidad de autor del trabajo de grado denominado: **“IMPLEMENTACION DEL SISTEMA DE CONTROL Y CALIFICACION DE PROVEEDORES PARA COOPERATIVAS DE AHORRO Y CRÉDITO”**, que ha sido desarrollada para optar por el título de Ingeniería en Sistemas Computacionales, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En mi condición de autor me reservo los derechos morales de la obra antes mencionada, aclarando que el trabajo aquí descrito es de mi autoría y que no ha sido previamente presentado para ningún grado o calificación profesional.

En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Técnica del Norte

Firma

Nombre: DANIEL FERNANDO CUARAN GUERRERO

Cédula: 1002880662

Ibarra a los 11 días del mes de septiembre del 2014



## UNIVERSIDAD TÉCNICA DEL NORTE

### BIBLIOTECA UNIVERSITARIA

## AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE

### 1. IDENTIFICACIÓN DE LA OBRA

La UNIVERSIDAD TÉCNICA DEL NORTE dentro del proyecto Repositorio Digital institucional determina la necesidad de disponer los textos completos de forma digital con la finalidad de apoyar los procesos de investigación, docencia y extensión de la universidad.

Por medio del presente documento dejo sentada mi voluntad de participar en este proyecto, para lo cual ponemos a disposición la siguiente investigación:

#### DATOS DE CONTACTO

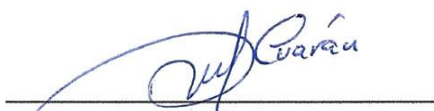
CEDULA DE IDENTIDAD	1002880662
APELLIDOS Y NOMBRES	CUARÁN GUERRERO DANIEL FERNANDO
DIRECCIÓN	Antonio Cordero 8-33 y Ricardo Sánchez
EMAIL	<a href="mailto:dcuaran@gmail.com">dcuaran@gmail.com</a>
TELÉFONO FIJO	062600195
TELÉFONO MOVIL	0982261220

#### DATOS DE LA OBRA

TITULO	“IMPLEMENTACION DEL SISTEMA DE CONTROL Y CALIFICACION DE PROVEEDORES PARA COOPERATIVAS DE AHORRO Y CRÉDITO”
AUTOR	CUARÁN GUERRERO DANIEL FERNANDO
FECHA	31 DE ENERO DEL 2013
PROGRAMA	PREGRADO
TITULO POR EL QUE	INGENIERÍA EN SISTEMAS COMPUTACIONALES
DIRECTOR	ING. MAURICIO REA

## 2. AUTORIZACIÓN DE USO A FAVOR DE LA UNIVERSIDAD

Yo, DANIEL FERNANDO CUARÁN GUERRERO, con cedula de identidad Nro. 1002880662, en calidad de autor y titular de los derechos patrimoniales de la obra o trabajo de grado descrito anteriormente, hago entrega del ejemplar respectivo en forma digital y autorizo a la Universidad Técnica del Norte, la publicación de la obra en el Repositorio Digital Institucional y el uso del archivo digital en la biblioteca de la universidad con fines académicos, para ampliar la disponibilidad del material y como apoyo a la educación, investigación y extensión, en concordancia con la Ley de Educación Superior Artículo 143.

A handwritten signature in blue ink, appearing to read 'D. Cuarán', is written over a horizontal line.

Firma

Nombre: DANIEL FERNANDO CUARÁN GUERRERO

Cédula: 1002880662

Ibarra a los 11 días del mes de septiembre del 2014

## **DEDICATORIA**

El siguiente trabajo se lo dedico a mis padres y hermanos por su apoyo incondicional.

## **AGRADECIMIENTO**

A la Cooperativa de Ahorro y Crédito “ATUNTAQUI” Ltda. por darme la oportunidad de realizar este trabajo, en especial al Departamento de Sistemas dirigido por el Ing. Roberto Peñafiel.

Al Ing. Mauricio Rea, por su ayuda y guía para hacer este proyecto.



## INDICE DE CONTENIDOS

CERTIFICACIÓN .....	i
CERTIFICACIÓN .....	ii
CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE INVESTIGACIÓN .....	iii
AUTORIZACIÓN DE USO Y PUBLICACIÓN A FAVOR DE LA UNIVERSIDAD TÉCNICA DEL NORTE.....	iv
DEDICATORIA.....	vi
AGRADECIMIENTO.....	vii
INDICE DE CONTENIDOS.....	viii
INDICE DE FIGURAS .....	xii
INDICE DE TABLAS.....	xiv
RESUMEN.....	xv
SUMMARY .....	xvi
1. Capítulo I: Introducción. ....	2
1.1. Tema.....	2
1.2. Problema.....	2
1.3. Objetivos .....	2
1.3.1. Objetivo General: .....	2
1.3.2. Objetivos Específicos: .....	2
1.4. Alcance.....	3
1.4.1. Calificación de Proveedores .....	3
1.4.2. Cotización y Adjudicación .....	3
1.4.3. Elaboración de Contrato.....	4
1.4.4. Devolución de Garantías .....	4
1.5. Justificación.....	6
1.5.1. Justificación Tecnológica: .....	6
1.5.2. Justificación Teórica: .....	6
1.6. Contexto .....	8
2. Capítulo II: Marco Teórico.....	10
2.1. Metodología combinación de RUP y MSF Agile.....	10
2.1.1. Participantes .....	10
2.1.2. Procesos.....	11

2.2.	Arquitectura 3 Capas .....	20
2.2.1.	Arquitectura multi capas .....	20
2.2.2.	Arquitectura de tres capas .....	21
2.3.	.NET Framework 3.5 .....	22
2.3.1.	Información general y conceptual sobre .NET Framework .....	22
2.3.2.	Arquitectura de .NET Framework 3.5 .....	26
2.4.	ASP.net con VB.net .....	41
2.4.1.	Marco de trabajo de páginas y controles .....	42
2.4.2.	Compilador de ASP.NET .....	43
2.5.	Librerías Developer Express .....	45
2.5.1.	ASPxperience .....	45
2.5.2.	ASPxEditors .....	45
2.5.3.	ASPxGauges .....	46
2.5.4.	ASPxGridView .....	46
2.5.5.	ASPxHtmlEditor .....	49
2.5.6.	ASPxPivotGrid .....	49
2.5.7.	ASPxScheduler .....	51
2.5.8.	ASPxSpellChecker .....	54
2.5.9.	ASPxTreeList .....	55
2.5.10.	XtraCharts .....	56
2.5.11.	XtraReports .....	57
2.6.	Sybase .....	58
2.6.1.	Características .....	58
2.6.2.	Creación de servidores mediante dataserver .....	58
2.6.3.	Bcp para transferir datos hacia y desde Adaptive Server .....	58
2.6.4.	Referencia de comandos de utilidades .....	59
2.6.5.	Programas de utilidades .....	59
2.7.	SQL dbx y Aqua Data Studio 9.0 .....	61
2.7.1.	SQL dbx .....	61
2.7.2.	Aqua Data Studio 9.0 .....	63
2.8.	Business Intelligence .....	65
2.8.1.	Introducción a BI .....	65
2.8.2.	Definiendo soluciones OLAP .....	67

2.8.3.	Diseñando una solución OLAP .....	71
Ejemplos de dimensión .....		74
Ejemplos de relaciones.....		74
Ejemplos de Dimensión Tiempo .....		79
2.8.4.	Construyendo una solución OLAP .....	83
2.8.5.	Implementando cubos OLAP .....	92
2.9.	SQL Server 2008 R2 BI .....	99
2.9.1.	BI con herramientas Microsoft.....	99
2.9.2.	SQL Server Integration Services .....	100
2.9.3.	SQL Server Analysis Services.....	101
2.9.4.	SQL Server Reporting Services.....	101
3.	Capítulo III: Proceso De Desarrollo.....	103
3.1.	Análisis.....	103
3.1.1.	Visión del Proyecto .....	103
3.1.2.	Requerimientos.....	103
3.2.	Diseño.....	104
3.2.1.	Diagramas de Actividades .....	105
3.3.	Construcción.....	109
3.3.1.	Diagrama Entidad Relación.....	109
3.3.2.	Diccionario de datos .....	110
3.3.3.	Clases.....	110
3.3.4.	Módulos.....	112
3.3.5.	Pruebas .....	117
3.3.6.	Creación Data mart.....	117
3.4.	Paso a producción.....	122
3.5.	Incidentes en Sistemas en Producción.....	122
3.6.	Respaldo Programas de Desarrollo .....	122
4.	Capítulo IV: Conclusiones y Recomendaciones.....	124
4.1.	Conclusiones .....	124
4.2.	Recomendaciones.....	125
5.	Capítulo V: Lista De Anexos.....	127
5.1.	Diccionario de Datos .....	127
5.2.	Manual técnico .....	127

5.3. Manual de usuario.....	127
GLOSARIO .....	128
BIBLIOGRAFIA .....	130
LINCOGRAFIA .....	131

## INDICE DE FIGURAS

Figura 1: Esquema del proyecto.....	5
Figura 2. .NET Framework en contexto.....	23
Figura 3. Clasificación de tipos.....	32
Figura 4. Etapa Inicial Data Mart.....	69
Figura 5. ETL.....	70
Figura 6. Segunda etapa Data Mart.....	71
Figura 7. Ejemplo esquema estrella.....	72
Figura 8. Dimensiones esquema estrella.....	75
Figura 9. Ejemplo dimensión.....	76
Figura 10. Ejemplo tabla normalizada y desnormalizada.....	77
Figura 11. Ejemplo dimensión tiempo.....	79
Figura 12. Tercera etapa data mart.....	83
Figura 13. MOLAP.....	84
Figura 14. ROLAP.....	85
Figura 15. MOLAP.....	86
Figura 16. Relación espacio almacenamiento – agregaciones.....	89
Figura 17. Cuarta etapa Data Mart.....	92
Figura 18. Permisos cubo.....	93
Figura 19. Permisos dimensión.....	93
Figura 20. Drill up – Drill down.....	94
Figura 21. Drill up-Drill down.....	94
Figura 22. Slice.....	95
Figura 23. Dice.....	95
Figura 24. Rotación.....	95
Figura 25. Consolidación.....	96
Figura 26. Áreas consultas, tabla pivotal.....	97
Figura 27. Ejemplo tabla pivotal.....	98
Figura 28. Calificación de proveedores.....	105
Figura 29. Cotizaciones y Adjudicaciones.....	106
Figura 30. Contratos.....	107
Figura 31. Garantías.....	108
Figura 32. Diagrama entidad - relación.....	109
Figura 33. Proyecto Conexión.....	110
Figura 34. Esquema del proyecto.....	112
Figura 35. Proyecto web services.....	113
Figura 36. Proyecto consola batch.....	114
Figura 37. Web applications.....	116
Figura 38. Creación SSIS.....	117
Figura 39. Creando OLE DB Source.....	118
Figura 40. Seleccionando proveedor en connection manager.....	118
Figura 41. Seleccionando tablas para extracción de datos.....	119
Figura 42. Modelo Dimensional para el data mart.....	119

Figura 43. Creando proyecto SSAS ..... 120  
Figura 44. Creando dimensiones del cubo..... 120  
Figura 45. Generando el data mart..... 121  
Figura 46. Data mart construido ..... 121

## INDICE DE TABLAS

Tabla 1. Indicadores de gestión.....	12
Tabla 2. Indicadores de gestión diseño.....	14
Tabla 3. Indicadores de gestión construcción.....	15
Tabla 4. Indicadores de gestión construcción.....	17
Tabla 5. Indicadores de gestión incidentes.....	18
Tabla 6. Indicadores de gestión respaldo programas.....	20
Tabla 7. Motor en tiempo de ejecución CLR.....	41
Tabla 8. OLTP – OLAP.....	66
Tabla 9. Ejemplo de dimensión.....	74
Tabla 10. Características esquemas.....	78
Tabla 11. Ejemplo comparaciones MOLAP ROLAP HOLAP.....	86
Tabla 12. Ejemplo comparaciones MOLAP ROLAP HOLAP.....	87
Tabla 13. Ejemplo comparaciones MOLAP ROLAP HOLAP.....	88

## RESUMEN

EL SISTEMA DE CONTROL Y CALIFICACION DE PROVEEDORES PARA COOPERATIVAS DE AHORRO Y CRÉDITO ha sido construido a partir de los requerimientos de los departamentos de Adquisiciones, Jurídico y Tesorería de la Cooperativa de Ahorro y Crédito “Atuntaqui” Ltda.

En éste documento se detalla la elaboración y desarrollo de los FRONT ENDS y BACK ENDS tanto para los usuarios internos y externos del sistema.

Se han realizado 2 FRONT ENDS:

- a) Front end dirigido a usuarios externos, en este caso llamados proveedores, los cuales podrán registrarse en el sistema de 3 formas diferentes de acuerdo a su estatus con la Institución, al registrarse en la página web ([www.atuntaqui.fin.ec](http://www.atuntaqui.fin.ec)) y ser aprobado su registro por el departamento de Adquisiciones si cumple con todos los requisitos, el proveedor será tomado en cuenta para participar en concursos de adjudicaciones de bienes y/o servicios que requiere la institución.
- b) Front end dirigido a usuarios internos, se accede a través de la página de la intranet, se ha dividido en 5 módulos:
  - Módulo de usuarios, aquí el usuario interno solicita al departamento de Servicios Generales los bienes y/o servicios que necesita, además de dar seguimiento a los procesos creados.
  - Módulo de Servicios Generales, aquí se tramitan los procesos de adquisición creados por los usuarios y aprobados por gerencia o ente encargado, se verifica que cumplan con los requisitos necesarios, se aprueba o niega, se envía las invitaciones a participar en el concurso a los proveedores de los bienes o servicios requeridos y se obtiene el resultado del concurso.
  - Módulo Administrativo, aquí se aprueban o niegan los procesos creados por los usuarios, también se realiza la adjudicación a la mejor propuesta (cotización).
  - Módulo Jurídico, aquí se realiza la elaboración de contratos del bien o servicio adjudicado en caso de ser necesario.
  - Módulo de Tesorería, aquí se puede visualizar los bienes que han cumplido con la garantía especificada en el momento de la adquisición.



## SUMMARY

The SYSTEM CONTROL AND QUALIFICATION OF SUPPLIERS FOR CREDIT UNIONS has been built from the requirements of the departments: Procurement, Legal and Treasury of the Cooperativa de Ahorro y Crédito “Atuntaqui” Ltda.

This document describes the design and development of front ends and back ends for both internal and external users of the system.

There have been 2 FRONT ENDS:

- a) Front end for external users, in this case called suppliers, which may be registered in the system in 3 different ways according to their status with the Institution, to register on the website ([www.atuntaqui.fin.ec](http://www.atuntaqui.fin.ec)) and approved registration by the Purchasing Department if meet all the requirements, the supplier will be considered for awards contests of goods and / or services required by the institution.
- b) Front end to internal users, is accessed through the intranet page. Has been divided into 5 modules:
  - Module of users, here the internal user requests the Department of General Services goods or services you need, in addition to monitor processes created.
  - Module of General Services, here are handled procurement processes created by users and approved by management or responsible entity, it verifies that comply with the requirements, is approved or denied, sent invitations to participate in the competition to suppliers of goods or services required and you get the results of the competition.
  - Administrative Module, here are approved or denied the processes created by users, also performed the allocation for the best proposal.
  - Legal Module, here is done the development of the contracts of the good or service adjudicated in if necessary.
  - Module of Treasury, here you can see the goods that they have complied with the warranty specified at the time of the acquisition.



*Daniel Cuarán*

2015





## **Capítulo I: Introducción.**

### **1.1. Tema**

IMPLEMENTACIÓN DEL SISTEMA DE CONTROL Y CALIFICACION DE PROVEEDORES PARA COOPERATIVAS DE AHORRO Y CRÉDITO

### **1.2. Problema**

El departamento de Servicios Generales de la Cooperativa de Ahorro y Crédito Atuntaqui Ltda. no posee ningún sistema para la adquisición y control de bienes.

Como la COAC Atuntaqui Ltda. cada día sigue creciendo, existe la necesidad de tener un sistema que permita dar un seguimiento correcto a todas las adquisiciones que se realizan para cada una de las sucursales y agencias de Imbabura y Pichincha, y que además siga los estándares, arquitectura y metodología establecidos para el desarrollo de sistemas de la Cooperativa.

Por este motivo se ha visto la necesidad de administrar los requerimientos de las oficinas y departamentos de la Cooperativa de Ahorro y Crédito Atuntaqui Ltda. en relación a la adquisición de bienes y servicios vía online para mejorar los procesos en cuanto a tiempos e incorporar el módulo de contratos y garantías y que seguirá los estándares, arquitectura y metodología establecidos en la COAC. Atuntaqui Ltda.

### **1.3. Objetivos**

#### **1.3.1. Objetivo General:**

IMPLEMENTAR EL SISTEMA DE CONTROL Y CALIFICACION DE PROVEEDORES PARA LA COOPERATIVA DE AHORRO Y CRÉDITO ATUNTAQUI LTDA

#### **1.3.2. Objetivos Específicos:**

Elaborar los modelos técnicos y la arquitectura del software en base a los requerimientos realizados por el Analista de Negocios.

1. Ejecutar la construcción y desarrollo del software respetando los estándares de programación de la Institución y guiándose por los requerimientos del usuario.
2. Proporcionar una herramienta web idónea y de fácil uso para los departamentos de Servicios Generales, Jurídico y Tesorería de acuerdo a los estándares establecidos en la Institución.
3. Agilizar los procedimientos administrativos y operativos entre los departamentos de Servicios Generales, Jurídico y Tesorería.
4. Creación de un Data mart<sup>1</sup>.
5. Instalar el sistema y todos sus componentes en ambiente de producción.

## **1.4. Alcance**

Se necesita crear y/o automatizar los siguientes procesos:

### **1.4.1. Calificación de Proveedores**

- Página Web solicitud de calificación
- Proveedor accede a los Requisitos
- Proveedor Adjuntar documentos y envía
- Bandeja de entrada de Servicios Generales recibe la documentación
- Aprueba o niega
- Bandeja de entrada del Proveedor Calificado recibe requerimientos
- Envía cotizaciones se recibe en bandeja de entrada de Servicios Generales.

### **1.4.2. Cotización y Adjudicación**

- Gerentes de oficina, Jefes de Agencia y Departamentales envían solicitud de adquisición
- Bandeja de entrada de Servicios Generales recibe la solicitud
- Remite a cada nivel de aprobación la solicitud para aprobación
- Si aprueba elabora las especificaciones o bases según el caso
- Sistema verifica a Proveedores Calificados y remite a los proveedores que consten con la actividad relacionada.
- Proveedores reciben solicitud de cotización en la bandeja de entrada
- Proveedores preparan información y envían las cotizaciones.
- Se receptan las cotizaciones

---

<sup>1</sup> **Data mart**: almacén de datos de un hecho en particular, es una base de datos departamental, especializada en el almacenamiento de los datos de un área de negocio específica.

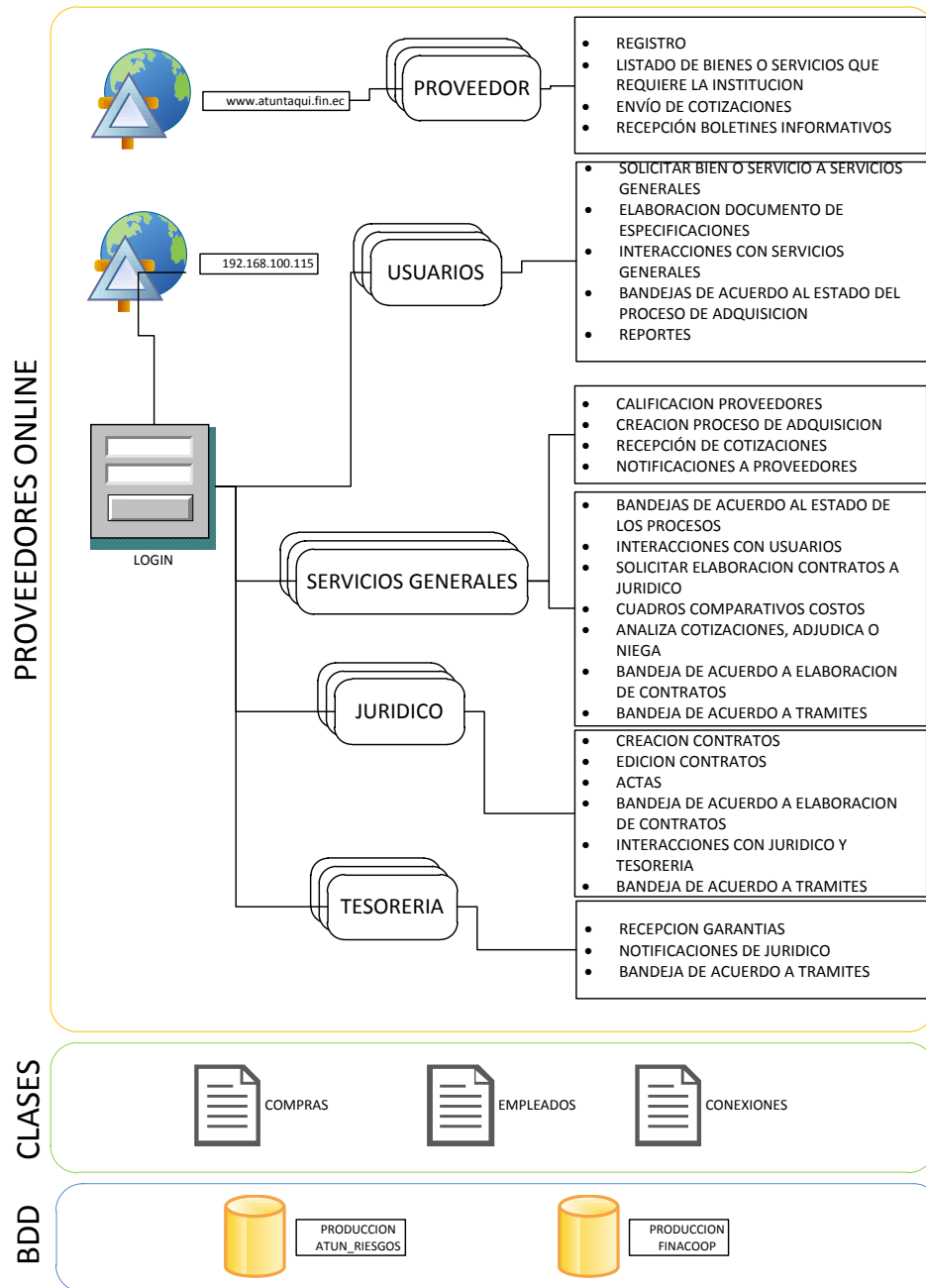
- Bandeja de entrada de Servicios Generales
- Servicios Generales elabora cuadro comparativo de costos
- Envía al nivel de adjudicación correspondiente
- Bandeja de entrada del nivel correspondiente recibe el cuadro comparativo con las cotizaciones
- Recibe, analiza y adjudica; o niega
- Si aprueba o niega se notifica a los proveedores la resolución tomada.
- Dependiendo del valor se envía para solicitar contrato.

#### **1.4.3. Elaboración de Contrato**

- Servicios Generales envía requerimiento de elaboración de contrato a Jurídico
- Jurídico elabora contrato
- Carga el contrato en el sistema para revisión
- Bandeja de entrada del área solicitante, aprueba u observa
- Si tiene cambios se realizan se imprimen para la sumilla
- Remite el físico: 4 ejemplares a Servicios Generales para legalización
- Notificación de entrega a Proveedor y usuario solicitante
- Retorna el contrato a Servicios Generales
- Bandeja de entrada solicitante notificación de retorno del contrato
- Servicios Generales entrega 1 ejemplar a Jurídico
- Jurídico remite la garantía y copia simple a Tesorería.
- Tesorería recibe y custodia
- Bandeja de entrada de Tesorería.

#### **1.4.4. Devolución de Garantías**

- Servicios Generales verifica la culminación del contrato
- Por el monto pide acta entrega recepción a Jurídico
- Bandeja de entrada de Jurídico elabora actas.
- Jurídico remite actas
- Se legalizan las actas
- Se realiza el pago a proveedores
- Se notifica a Tesorería la culminación del contrato
- Bandeja de entrada de Tesorería
- Devolución de Garantía



Fuente: Propia

Figura 1: Esquema del proyecto



## **1.5. Justificación**

El diseño y construcción del nuevo sistema de Proveedores Online de la institución es un requerimiento de alta prioridad debido a que es una recomendación de Auditoría y a que el sistema actual no cumple con las necesidades actuales.

### **1.5.1. Justificación Tecnológica:**

Un nuevo sistema de Proveedores Online contribuirá en gran medida con el área adquisiciones de la Cooperativa de Ahorro y Crédito Atuntaqui Ltda. ya que éste permitirá hacer un mejor seguimiento y mejorar la calidad en el servicio de los casos reportados por los usuarios internos de la Institución.

### **1.5.2. Justificación Teórica:**

#### **1.5.2.1. Metodología:**

La propuesta de la metodología es una combinación de dos tipos de metodologías muy usadas y para proyectos que abarcan ya sea de 3 a 4 personas, hasta proyectos más complejos que abarcan hasta 50 o más personas, estas metodologías son la RUP<sup>2</sup> y MSF Agile (Microsoft Solutions Framework).

Esta metodología debe acoplarse a las necesidades de la institución, así como también a las personas que intervienen en el desarrollo del mismo desde su inicio hasta la finalización del mismo con sus distintas etapas, tales como la Análisis, las incidencias o problemas, etc.

Los participantes que intervienen en esta metodología son los siguientes:

- Analista de Negocios
- Jefe de Sistemas
- Administrador de Desarrollo
- Equipo de Arquitectura
- Analista Programador
- Equipo de Pruebas
- Usuario

---

<sup>2</sup> **RUP** (*Rational Unified Process*): constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

**Analista de Negocios.-** Se entiende por analista de negocios al jefe de departamento que realizó el requerimiento inicial, a su vez esta persona se encargará del levantamiento de los requerimientos, de diseñar la visión o los objetivos del producto final, de designar a los participantes y el rol de cada uno de ellos, a su vez también será responsable de identificar y mitigar los riesgos de negocios que se puedan dar en la realización del proyecto.

**Jefe de Sistemas.-** Bajo esta metodología el jefe de sistemas recibirá el DVD con todo el desarrollo del proyecto, para que este se encargue del almacenamiento del mismo.

**Administrador de Desarrollo.-** Este se encargará de definir el entorno, el ambiente, la infraestructura y las herramientas que se usarán en el desarrollo del producto. Además de definir la planificación, el seguimiento de cada una de las tareas del desarrollo, revisión de los objetivos y los progresos del proyecto, la entrega a producción de la documentación y paquetes entregables, de la entrega a producción de nuevas versiones o actualizaciones, así como de la depuración de los directorios donde reside el software y la documentación de los productos desarrollados.

**Equipo de Arquitectura.-** Este se encargará de realizar una síntesis de la arquitectura, elaborar un modelo de arquitectura que brinde facilidad y flexibilidad en la construcción del producto, además tomando en cuenta la escalabilidad y el rendimiento del sistema, se encargará además de definir una arquitectura candidata, en base al análisis de requerimientos y casos de uso, además definirá los posibles riesgos técnicos y su mitigación.

**Analista Programador.-** El analista programador se encargará en la fase inicial de la elaboración de los prototipos de componentes, ya sean clases, interfaces, módulos, prototipos, bases de datos, también de particionar y separar los elementos en subsistemas cuando estos se puedan dividir, de la realización de pruebas, documentación de código fuente.

En la fase de construcción se encargará del desarrollo del software como tal incluyéndose componentes, clases, interfaces, módulos, optimización y refactoring <sup>3</sup> del código fuente, de la carga inicial de datos, resolución de conflictos, problemas o bugs y en cada una de las fases debe ir acompañado de la documentación, además en las fases finales deberá encargarse de copiar y sincronizar las últimas versiones del software en el servidor de desarrollo.

**Equipo de Pruebas.-** El equipo de pruebas se encargará de realizar las pruebas de aceptación en las fases de construcción y transición, previa su aceptación para realizar el paso del software a producción.

**Usuario.-** El usuario interviene en la fase post-producción y se encargará de reportar detalladamente sobre los incidentes, bugs <sup>4</sup> del software si es que los hubiere y proporcionar o reproducir el mismo incidente.

---

<sup>3</sup> **Refactoring:** técnica de la ingeniería de software para reestructurar un código fuente.

<sup>4</sup> **Bugs:** es un error o fallo en un programa de computador o sistema de software.

### **1.5.2.2. Procesos**

Los procesos son las distintas etapas que abarcan el proyecto desde la concepción o la necesidad de la realización del producto hasta su elaboración y puesta a producción.

La metodología propuesta contempla varias etapas:

- Análisis
- Diseño
- Construcción
- Paso a Producción
- Incidentes

### **1.6. Contexto**

En cuanto a proyectos similares encontrados en la lista de anteproyectos aprobados de la Escuela de Sistemas de la UTN existen los siguientes:

- Tema: Diseño e implementación de una aplicación WEB para la gestión documental de proveedores de materia prima en IANCEM.

De : Luis Francisco Nazate Maldonado

La diferencia que existe con este proyecto es que es un sistema personalizado de acuerdo a los procesos de los departamentos de Servicios Generales, Jurídico y Tesorería de la Cooperativa de Ahorro y Crédito “Atuntaqui Ltda.” en especial al departamento de Servicios Generales ya que permitirá realizar un adecuado seguimiento de los procesos de adquisiciones de bienes y servicios, tomando en cuenta que para la construcción del sistema de Proveedores Online se utilizará una Metodología, Arquitectura y Herramientas totalmente diferentes y el enfoque del sistema es muy distinto.

- Tema: Sistema de Gestión de Compras y Contratación Pública

De: Edison Israel Collaguazo Sinchiguano

La diferencia con este proyecto es el enfoque, éste proyecto se usa para la gestión de productos o servicios ya adquiridos mediante el sistema de compras públicas.

*Daniel Cuarán*

2015

*CAPITULO 2*



## Capítulo II: Marco Teórico.

### 2.1. Metodología combinación de RUP y MSF Agile.

La propuesta de la metodología actual es una combinación de dos tipos de metodologías muy usadas y para proyectos que abarcan ya sea de 3 a 4 personas, hasta proyectos más complejos que abarcan hasta 50 o más personas, estas metodologías son la RUP y MSF Agile.

Esta metodología debe acoplarse a las necesidades de la institución, así como también a las personas que intervienen en el desarrollo del mismo desde su inicio hasta la finalización del mismo con sus distintas etapas, tales como el Análisis, las incidencias o problemas, etc.

#### 2.1.1. Participantes

Los participantes que intervienen en esta metodología son los siguientes:

- Analista de Negocios
- Jefe de Sistemas
- Administrador de Desarrollo
- Equipo de Arquitectura
- Analista Programador
- Equipo de Pruebas
- Usuario

**Analista de Negocios.-** Se entiende por analista de negocios al jefe de departamento que realizó el requerimiento inicial, a su vez esta persona se encargará del levantamiento de los requerimientos, de diseñar la visión o los objetivos del producto final, de designar a los participantes y el rol de cada uno de ellos, a su vez también será responsable de identificar y mitigar los riesgos de negocios que se puedan dar en la realización del proyecto.

**Jefe de Sistemas.-** Bajo esta metodología el jefe de sistemas recibirá el DVD con todo el desarrollo del proyecto, para que este se encargue del almacenamiento del mismo.

**Administrador de Desarrollo.-** Este se encargará de definir el entorno, el ambiente, la infraestructura y las herramientas que se usarán en el desarrollo del producto. Además de definir la planificación, el seguimiento de cada una de las tareas del desarrollo, revisión de los objetivos y los progresos del proyecto, la entrega a producción de la documentación y paquetes entregables, de la entrega a producción de nuevas versiones o actualizaciones, así como de la depuración de los directorios donde reside el software y la documentación de los productos desarrollados.

**Equipo de Arquitectura.-** Este se encargará de realizar una síntesis de la arquitectura, elaborar un modelo de arquitectura que brinde facilidad y flexibilidad en la construcción del producto, además tomando en cuenta la escalabilidad y el rendimiento del sistema, se encargará además de definir una

arquitectura candidata, en base al análisis de requerimientos y casos de uso, además definirá los posibles riesgos técnicos y su mitigación.

**Analista Programador.-** El analista programador se encargará en la fase inicial de la elaboración de los prototipos de componentes, ya sean clases, interfaces, módulos, prototipos, bases de datos, también de particionar y separar los elementos en subsistemas cuando estos se puedan dividir, de la realización de pruebas, documentación de código fuente.

En la fase de construcción se encargará del desarrollo del software como tal incluyéndose componentes, clases, interfaces, módulos, optimización y refactoring del código fuente, de la carga inicial de datos, resolución de conflictos, problemas o bugs y en cada una de las fases debe ir acompañado de la documentación, además en las fases finales deberá encargarse de copiar y sincronizar las últimas versiones del software en el servidor de desarrollo.

**Equipo de Pruebas.-** El equipo de pruebas se encargará de realizar las pruebas de aceptación en las fases de construcción y transición, previa su aceptación para realizar el paso del software a producción.

**Usuario.-** El usuario interviene en la fase post-producción y se encargará de reportar detalladamente sobre los incidentes, bugs del software si es que los hubiere y proporcionar o reproducir el mismo incidente.

### **2.1.2. Procesos**

Los procesos son las distintas etapas que abarcan el proyecto desde la concepción o la necesidad de la realización del producto hasta su elaboración y puesta a producción.

La metodología propuesta contempla varias etapas:

- Análisis
- Diseño
- Construcción
- Paso a Producción
- Incidentes
- Respaldo Programas Desarrollo

#### **2.1.2.1. Análisis**

En este procedimiento se establece los requerimientos y alcances para la elaboración de proyectos de desarrollo de software.

#### **Objetivo General**

Establecer el alcance del proyecto de software, incluyendo una visión operacional y requisitos de calidad.

## Políticas Específicas.

Para la gestión de los proyectos de desarrollo se utilizará la herramienta Visual Studio Team Foundation Server, registrando la visión, requerimientos, riesgos en el portal creado por la herramienta para cada uno de los proyectos.

El equipo de arquitectura estará formado por el área de desarrollo (Administrador de Desarrollo, Analistas Programadores y Jefe de Sistemas).

Analista de negocios se entenderá como cualquier jefatura departamental que realice el requerimiento de desarrollo.

El usuario experto publicará la visión del proyecto en el formato de levantamiento de requerimientos.

El usuario experto y Administrador de Desarrollo ingresará las metas de evaluación o los requisitos de calidad en el formulario de levantamiento de requerimientos.

## Indicadores

INDICADOR DE GESTIÓN	
Nombre del indicador:	Requerimientos
Forma de cálculo:	Número de requerimientos
Valor inicial:	0
Meta:	Nuevos requerimientos
Unidad de medida:	Número de requerimientos
Frecuencia de evaluación:	Por proyecto
Fuente de información generado:	Formulario de requerimientos Formulario de pruebas de aceptación.

**Fuente:** Manuales de procesos cooperativa Atuntaqui.

**Tabla 1.** Indicadores de gestión.

## Descripción

Administrador de Desarrollo comunicará el Inicio del desarrollo del proyecto al Jefe del proyecto y solicita el desglose de los documentos para el levantamiento de los requerimientos, de acuerdo a los proyectos planificados en el plan de desarrollo anual.

Analista de negocios establecerá el alcance de nuevos proyectos de desarrollo o mejoras a sistemas ya existentes, en este punto se puede tomar la decisión de continuar o no con la ejecución del proyecto.

Analista de Negocios y Administradora de procesos establecerá la funcionalidad de los requerimientos y los roles de los involucrados a través de un flujograma<sup>5</sup> del procedimiento relacionado al uso del sistema.

Analista de Negocios realizará la definición de requerimientos funcionales del proyecto que serán utilizados para definir su alcance; y los detalles de las especificaciones de cada requerimiento se llenarán en el formato indicado, priorizando el orden en que serán desarrollados.

Analista de Negocios y Administrador de Desarrollo definen metas de evaluación o requisitos de calidad de servicio, que son requerimientos no funcionales, o restricciones sobre la funcionalidad del sistema.

El Equipo de Arquitectura analizará las arquitecturas candidatas que brinden facilidad y flexibilidad para su construcción, tomando en cuenta el rendimiento y escalabilidad del sistema.

Administrador de Desarrollo establecerá la planificación de actividades para el desarrollo del proyecto y a su vez realizará la creación del proyecto en el portal del Team Foundation Server (Proyectos Nuevos), así como también en el directorio de la documentación del proyecto correspondiente.

En el portal del Team Foundation Server, administrador de desarrollo publicará el levantamiento de requerimientos realizado por Analista de Negocios.

Finalmente Administrador de desarrollo manejará la planificación, realizando el seguimiento de cada una de las actividades.

### **2.1.2.2. Diseño**

Procedimiento que describe los modelos técnicos y la arquitectura que se va a utilizar previo el proceso de construcción del software.

#### **Objetivos**

Elaborar los modelos técnicos y la arquitectura del software en base a los requerimientos realizados por el Analista de Negocios.

#### **Políticas Específicas**

Para la gestión del proyecto se utilizará la herramienta Visual Studio Team Foundation Server.

El equipo de arquitectura estará formado por los siguientes perfiles: Administrador de Desarrollo, Analistas Programadores y Jefe de Sistemas.

---

<sup>5</sup> **Flujograma:** es una muestra visual de una línea de pasos de acciones que implican un proceso determinado.



Analista de negocios se entenderá como cualquier jefatura departamental que realice el requerimiento de desarrollo.

<b>INDICADOR DE GESTIÓN</b>	
Nombre del indicador:	Diseños de la Aplicación
Forma de cálculo:	Número de diseños realizados según la cantidad de requerimientos
Valor inicial:	0
Meta:	Ningún cambio en el diseño inicial.
Unidad de medida:	Número de diseños realizados
Frecuencia de Evaluación:	Anual
Fuente de información:	Formulario de Requerimientos
<b>RESPONSABLE</b>	<b>BACK UP</b>
Administrador de Desarrollo	Analista Programador

**Fuente:** Manuales de procesos cooperativa Atuntaqui.

**Tabla 2.** Indicadores de gestión diseño.

### **Descripción**

Administrador de desarrollo revisará el plan de desarrollo.

Por su parte el equipo de Arquitectura establecerá la arquitectura definitiva en base al flujograma del proceso del software, de los requerimientos funcionales y arquitecturas candidatas, considerando plataforma y herramientas instaladas.

Administrador de Desarrollo coordinará con Analistas Programadores la redefinición del sistema según la arquitectura seleccionada.

El/los analista/s programadores desarrollarán el diseño de la data de la aplicación.

Administrador de desarrollo revisará la planificación de desarrollo del proyecto, evaluando los objetivos y el progreso. De ser necesario se actualiza el cronograma de planificación del proyecto; finalmente Administrador de Desarrollo publicará en el portal del Team Foundation Server el diagrama entidad relación/Diagrama de Objetos.

### 2.1.2.3. Construcción

Procedimiento que describe las actividades para la construcción y desarrollo del proyecto de software mediante la construcción del diseño de la base de datos y del código fuente del programa.

#### Objetivo General.

Ejecutar la construcción y desarrollo del software respetando los estándares de programación y guiándose por los requerimientos del usuario.

#### Políticas Específicas.

Para la gestión del proyecto se utilizará la herramienta Visual Studio Team Foundation Server.

Formarán parte del equipo de pruebas auditoría informática, procesos, oficial de seguridad, y el personal designado por la jefatura que levanto el requerimiento (Usuario experto).

INDICADORES DE GESTIÓN	
Nombre	Errores o bugs en etapa de pruebas.
Forma de Cálculo	Número de errores encontrados.
Valor inicial	0
Meta	Menor índice de errores en etapa de pruebas.
Unidad de Medida	Número de errores encontrados.
Frecuencia de Evaluación	Por Proyecto
Fuente de información	Formulario de Requerimientos, Entrevistas, reuniones de trabajo.
RESPONSABLE	BACK UP
Administrador de Desarrollo	Analista Programador

**Fuente:** Manuales de procesos cooperativa Atuntaqui.

**Tabla 3.** Indicadores de gestión construcción.

#### Descripción

Analista Programador Implementará los objetos de la base de datos usando el modelo entidad relación, este también desarrollará los componentes, clases, interfaces, módulos, eventos que se han definido dentro del alcance del proyecto.

De la misma manera el analista programador realizará un Refactoring, optimizaciones en el código del sistema sin modificar la funcionalidad del mismo, a fin de corregir cualquier eventualidad encontrada en el desarrollo del proyecto.

Administrador de Desarrollo realizará la convocatoria al equipo de pruebas para la realización de las pruebas de aceptación y certificación del software desarrollado.

Una vez hecha la convocatoria a pruebas y terminado el desarrollo de la aplicación se realizarán las pruebas de aceptación conjuntamente con el equipo de pruebas.

Auditor informático por su parte revisará el sistema en términos de diseño, funcionalidad y controles, a su vez emitirá un informe de evaluación del diseño de bases de Datos, Metodología, Pistas de Auditoría y toda la verificación que el auditor informático estime necesario acerca del software desarrollado.

Analista de Negocios o el Jefe del Proyecto certificará las pruebas del software y autoriza el paso a producción.

El analista programador generará el script de conversión de datos, en caso que el sistema lo requiera para cargas iniciales de datos o una migración, así como también el material de soporte y la documentación respectiva.

Administrador de desarrollo entregará al equipo de producción la documentación, medios de instalación, scripts, etc. Y finalmente administrará la documentación del proyecto en el portal del Team Foundation Server.

#### **2.1.2.4. Paso a Producción**

Procedimiento que señala las actividades para ejecutar la implantación del sistema en producción previa las pruebas de aceptación y certificación del software.

##### **Objetivo General**

Instalar el sistema y todos sus componentes en ambiente de producción.

##### **Políticas Específicas.**

El listado de transacciones de seguridad que envíe Administrador de Desarrollo a Analista de Seguridades lo realizará en formato Excel y vía correo electrónico.

Administrador de desarrollo de software o su delegado, enviarán la convocatoria a los usuarios que van a manejar el software indicando la fecha, hora y lugar de la socialización vía correo electrónico.

El equipo de Soporte técnico lo conforma el empleado del departamento de producción que esté de turno en el momento de realizar el paso a producción.

<b>INDICADORES DE GESTIÓN</b>	
<b>Nombre del indicador:</b>	Paso a Producción
<b>Forma de cálculo:</b>	Cantidad de observaciones de los usuarios sobre el sistema en funcionamiento. (0 observaciones correcto)
<b>Valor inicial:</b>	No aplica
<b>Meta:</b>	Sistema funcionando al 100% en producción.
<b>Unidad de Medida:</b>	Número de observaciones de los usuarios relacionados a la instalación.
<b>Frecuencia de Evaluación:</b>	Por proyecto
<b>Fuente de información:</b>	Formulario de notas de actualizaciones Formulario de pruebas de aceptación.
<b>RESPONSABLE</b>	<b>BACK UP</b>
Administrador de Desarrollo	Analista Programador

**Fuente:** Manuales de procesos cooperativa Atuntaqui.

**Tabla 4. Indicadores de gestión construcción.**

### **Descripción**

El equipo de producción recibe el formulario de notas de actualización conjuntamente con el paquete de actualizaciones y scripts; además archiva la documentación recibida en el directorio de versiones en producción, organizado por módulo y a su interior por fecha del paso a producción.

De la misma manera el equipo de producción revisará el documento de notas de actualizaciones y descarga de medios e instaladores del software e instala de acuerdo a la información prevista en el documento.

Administrador de Desarrollo notificará a Analista de Seguridades las transacciones de seguridad que contiene el software y anexa el manual de usuario; a su vez registrará las transacciones nuevas en las tablas de transacciones de producción.

Analista de Negocios y Administrador de desarrollo conjuntamente planificarán la capacitación y socialización a los usuarios.

Analista de Negocios, Administrador de Desarrollo y/o Analistas programadores realizarán la socialización y capacitación del sistema desarrollado a los usuarios.

### 2.1.2.5. Incidentes

Proporcionar las actividades para solucionar algún incidente que se haya presentado al usuario en los sistemas que se encuentran ya en producción, y que ya han pasado las fases de desarrollo.

#### Objetivo general

Solucionar el incidente de forma que no se vuelva a presentar en ninguno de los sistemas que ya se encuentran en producción.

#### Políticas específicas

Para la gestión del proyecto se utilizará la herramienta Visual Studio Team Foundation Server.

Formarán parte del equipo de pruebas auditoría informática y oficial de seguridad.

INDICADORES DE GESTIÓN	
<b>Nombre del indicador:</b>	Solución al incidente o bug
<b>Forma de Cálculo:</b>	Cantidad de incidentes o bugs solucionados según los reportados
<b>Valor inicial:</b>	No Aplica
<b>Meta:</b>	Solución del incidente en el menor tiempo posible
<b>Unidad de Medida:</b>	Cantidad de incidentes o bugs solucionados con el visto bueno del usuario
<b>Frecuencia de evaluación:</b>	Por incidente
<b>Fuente de información:</b>	Detalles de error
<b>RESPONSABLE</b>	<b>BACK UP</b>
Administrador de Desarrollo	Analista Programador

**Fuente:** Manuales de procesos cooperativa Atuntaqui.

**Tabla 5. Indicadores de gestión incidentes.**

#### Descripción

El usuario final que usa el sistema definirá el escenario, con información detallada de la retroalimentación del usuario o los detalles del incidente que se le presentó, adjuntando pantallas, bosquejos a fin de reproducir correctamente el incidente.

Administrador de Desarrollo evaluará el incidente en base a la información provista por él usuario y asigna a analista programador para dar solución.

Analista Programador reproducirá el incidente con datos de prueba del servidor de desarrollo, y a su vez modificará los programas para solucionar el incidente; una vez corregidos los errores o bugs solucionados realizará un refactoring que consiste en realizar cambios en el código desarrollado para hacerlo más eficiente sin afectar la funcionalidad del usuario.

El equipo de pruebas realizará las pruebas tantas veces sean necesarias hasta que se solucione completamente el incidente.

Una vez realizadas las pruebas el analista programador desarrollará el material de soporte de los cambios realizados en los documentos correspondientes.

Administrador de Desarrollo entregará a producción una nueva versión de todos los cambios que se han realizado.

Administrador de Desarrollo y/o Analista programador notificará vía service desk <sup>6</sup> la solución del incidente con el estado completado.

Finalmente administrador de desarrollo administrará la documentación del proyecto en el portal del TFS.

#### **2.1.2.6. Respaldo de Programas de Desarrollo**

Describe las actividades para la realización de los respaldos de los sistemas desarrollados en la institución.

##### **Objetivo General**

Generar los respaldos de los sistemas desarrollados internamente.

<b>INDICADORES DE GESTIÓN</b>	
Nombre del indicador:	Registro de Respaldo
Forma de cálculo:	Cantidad de respaldos
Valor inicial:	No Aplica
Meta:	1 Respaldo Mensual
Unidad de medida:	Numérico
Frecuencia de evaluación:	Mensual
Fuente de información	Bitácora de Respaldos

<sup>6</sup> **Service desk:** Sistema informático de soporte técnico y punto único de contacto para los usuarios finales que necesitan ayuda.

<b>RESPONSABLE</b>	<b>BACK UP</b>
Administrador de Desarrollo	Analista Programador

**Fuente:** Manuales de procesos cooperativa Atuntaqui.

**Tabla 6. Indicadores de gestión respaldo programas.**

## **Descripción**

Administrador de desarrollo solicitará a Analistas Programadores confirmen las últimas versiones de los programas y documentación de desarrollo que se encuentran en el servidor de desarrollo.

Analista Programador confirmará si las últimas versiones de programas y documentación están en el servidor de desarrollo; en caso que no estén las últimas versiones de los programas, copiar (sincronizar) las copias locales del equipo al servidor de desarrollo.

Administrador de desarrollo depurará los directorios de los programas y documentación del servidor de desarrollo, y a su vez copiará los archivos de los programas del servidor de desarrollo a DVD y realizar la codificación del DVD usando el siguiente estándar:

RESP-DES-mm/dd/yyyy

Administrador de Desarrollo entregará DVD para almacenamiento a Jefe de Sistemas y este a su vez almacenará el DVD en un lugar seguro.

## **2.2. Arquitectura 3 Capas**

### **2.2.1. Arquitectura multi capas**

La arquitectura multicapas, que también se conoce como ‘arquitectura n capas’ es una arquitectura cliente servidor en la cual, el manejo de datos, el procesamiento de la aplicación y la presentación, son procesos lógicamente separados. La arquitectura n capas más difundida es la arquitectura en tres capas. La arquitectura en n capas provee un modelamiento a los desarrolladores para crear una aplicación flexible y reusable. Con una aplicación separada en capas, los desarrolladores solo tienen que modificar o adicionar una capa específica, en vez de tener que escribir toda la aplicación de nuevo. En una arquitectura a tres capas deberíamos de tener:

- Una capa de presentación
- Capa de lógica de negocios
- Capa de acceso a datos

## 2.2.2. Arquitectura de tres capas

Es una arquitectura cliente servidor en la donde el acceso a datos, la lógica de negocios y la interfaz de usuario están desarrollados y mantenidos como módulos independientes, frecuentemente en plataformas separadas. El modelo de tres capas es una arquitectura de software así como también un patrón de diseño de software. Además de todas las ventajas que tiene realizar software por módulos, la arquitectura de tres capas está hecha con el propósito de que cualquier de sus tres capas (datos, lógica y presentación) pueda ser actualizada o reemplazada independientemente de acuerdo a como vayan cambiando las necesidades con el avance tecnológico. Por ejemplo, cuando ocurre un cambio de sistema operativo solo afectaría a la capa de presentación ya que en esta capa se maneja el código para la interfaz con el usuario. Por lo general, la interfaz de usuario corre sobre un ordenador de escritorio y usa una interfaz estándar de gráficos. La lógica consiste en uno o más módulos separados que corren sobre una estación de trabajo o un servidor de aplicaciones. Y finalmente, un servidor de bases de datos es el que contiene los datos almacenados de la aplicación o servicio.

### 2.2.2.1. Capa de presentación

Esta capa se encuentra en el nivel más externo de la aplicación, en la capa de presentación se despliega información relacionada a ciertos servicios como por ejemplo, compras, mercadeo, contenidos de ventas, etc.

Por lo general está relaciona mayormente con la capa de lógica de negocios.

### 2.2.2.2. Capa de lógica de negocios

Se la conoce también como “capa de negocios”, “capa lógica”, “capa de aplicación” o “capa media”; Esta capa controla la funcionalidad de la aplicación mediante procesamiento detallado de acuerdo a los requisitos otorgados por la capa de presentación.

### 2.2.2.3. Capa de acceso a datos

Esta capa está formada de servidores de bases de datos, la información es almacenada y recibida, mantiene a los datos de una manera neutral e independiente a la capa de presentación y/o la capa de negocios.

### Uso en el desarrollo web

En desarrollo web, el modelo de tres capas es comúnmente usado al referirnos a páginas web, generalmente con páginas que alberguen cosas tales como comercio electrónico. Las tres capas se repartirían así:

- Un front-end <sup>7</sup> o contenido desplegado en el navegador
- Un servidor de aplicaciones que albergue contenido medio de procesamiento y generación, por ejemplo ASP.NET, PHP, Java EE, etc.

---

<sup>7</sup> **Front End:** es la parte del software que interactúa con el o los usuarios.



- Una RDBMS (Sistema de manejo de bases de datos relacionales, por sus siglas en inglés), que maneje y provea el acceso a los datos en una base de datos.

## 2.3. .NET Framework 3.5

### 2.3.1. Información general y conceptual sobre .NET Framework

.NET Framework<sup>8</sup> es un componente integral de Windows que admite la creación y la ejecución de la siguiente generación de aplicaciones y servicios Web XML<sup>9</sup>. El diseño de .NET Framework está enfocado a cumplir los objetivos siguientes: (MICROSOFT, 2008)

- Proporcionar un entorno coherente de programación orientada a objetos, en el que el código de los objetos se pueda almacenar y ejecutar de forma local, ejecutar de forma local pero distribuida en Internet o ejecutar de forma remota.
- Proporcionar un entorno de ejecución de código que reduzca lo máximo posible la implementación de software y los conflictos de versiones.
- Ofrecer un entorno de ejecución de código que fomente la ejecución segura del mismo, incluso del creado por terceras personas desconocidas o que no son de plena confianza.
- Proporcionar un entorno de ejecución de código que elimine los problemas de rendimiento de los entornos en los que se utilizan secuencias de comandos o intérpretes de comandos.
- Ofrecer al programador una experiencia coherente entre tipos de aplicaciones muy diferentes, como las basadas en Windows o en el Web.
- Basar toda la comunicación en estándares del sector para asegurar que el código de .NET Framework se puede integrar con otros tipos de código.

.NET Framework contiene dos componentes principales: Common Language Runtime<sup>10</sup> y la biblioteca de clases de .NET Framework. Common Language Runtime es el fundamento de .NET Framework. El motor en tiempo de ejecución se puede considerar como un agente que administra el código en tiempo de ejecución y proporciona servicios centrales, como la administración de memoria, la administración de subprocesos y la interacción remota, al tiempo que aplica una seguridad estricta a los tipos y otras formas de especificación del código que fomentan su seguridad y solidez. De hecho, el concepto de administración de código es un principio básico del motor en tiempo de ejecución. El código destinado al motor en tiempo de ejecución se denomina código administrado, a diferencia del resto de código, que se conoce como código no administrado. La biblioteca de clases, el otro componente principal de .NET Framework, es una completa colección orientada a objetos de tipos reutilizables que se pueden emplear

---

<sup>8</sup> **Framework:** es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

<sup>9</sup> **XML:** eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium(W3C) utilizado para almacenar datos en forma legible.

<sup>10</sup> **Common Language Runtime:** o CLR es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma Microsoft .NET. El CLR es el encargado de compilar una forma de código intermedio llamada Common Intermediate Language al código de maquina nativo.

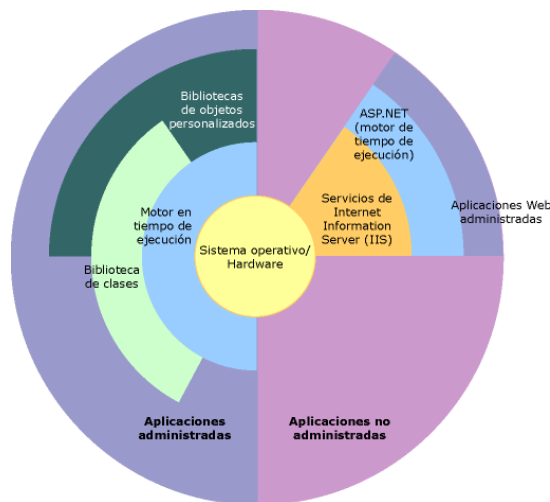
para desarrollar aplicaciones que abarcan desde las tradicionales herramientas de interfaz gráfica de usuario (GUI) o de línea de comandos hasta las aplicaciones basadas en las innovaciones más recientes proporcionadas por ASP.NET, como los formularios Web Forms y los servicios Web XML.

.NET Framework puede alojarse en componentes no administrados que cargan Common Language Runtime en sus procesos e inician la ejecución de código administrado, con lo que se crea un entorno de software en el que se pueden utilizar características administradas y no administradas. En .NET Framework no sólo se ofrecen varios hosts de motor en tiempo de ejecución, sino que también se admite el desarrollo de estos hosts por parte de terceros.

Internet Explorer es un ejemplo de aplicación no administrada que aloja el motor en tiempo de ejecución (en forma de una extensión de tipo MIME). Al usar Internet Explorer para alojar el motor en tiempo de ejecución, puede incrustar componentes administrados o controles de Windows Forms en documentos HTML. Al alojar el motor en tiempo de ejecución de esta manera se hace posible el uso de código móvil administrado, pero con mejoras significativas que sólo el código administrado puede ofrecer, como la ejecución con confianza parcial y el almacenamiento aislado de archivos.

En la ilustración siguiente se muestra la relación de Common Language Runtime y la biblioteca de clases con las aplicaciones y el sistema en su conjunto. En la ilustración se representa igualmente cómo funciona el código administrado dentro de una arquitectura mayor.

### **.NET Framework en contexto**



**Fuente:** (MICROSOFT, 2008)

**Figura 2. .NET Framework en contexto**

En las secciones siguientes se describen con más detalle los componentes y características principales de .NET Framework.

### 2.3.1.1. Características de Common Language Runtime

Common Language Runtime administra la memoria, ejecución de subprocesos, ejecución de código, comprobación de la seguridad del código, compilación y demás servicios del sistema. Estas características son intrínsecas del código administrado que se ejecuta en Common Language Runtime. (MICROSOFT, 2008)

Con respecto a la seguridad, los componentes administrados reciben grados de confianza diferentes, en función de una serie de factores entre los que se incluye su origen (como Internet, red empresarial o equipo local). Esto significa que un componente administrado puede ser capaz o no de realizar operaciones de acceso a archivos, operaciones de acceso al Registro y otras funciones delicadas, incluso si se está utilizando en la misma aplicación activa.

El motor en tiempo de ejecución impone seguridad en el acceso al código. Por ejemplo, los usuarios pueden confiar en que un archivo ejecutable incrustado en una página Web puede reproducir una animación en la pantalla o entonar una canción, pero no puede tener acceso a sus datos personales, sistema de archivos o red. Por ello, las características de seguridad del motor en tiempo de ejecución permiten que el software legítimo implementado en Internet sea excepcionalmente variado.

Además, el motor en tiempo de ejecución impone la solidez del código mediante la implementación de una infraestructura estricta de comprobación de tipos y código denominado CTS<sup>11</sup>. CTS garantiza que todo el código administrado es auto descriptivo. Los diferentes compiladores de lenguajes de Microsoft y de terceros generan código administrado que se ajusta a CTS. Esto significa que el código administrado puede usar otros tipos e instancias administrados, al tiempo que se aplica inflexiblemente la fidelidad y seguridad de los tipos.

Además, el entorno administrado del motor en tiempo de ejecución elimina muchos problemas de software comunes. Por ejemplo, el motor en tiempo de ejecución controla automáticamente la disposición de los objetos, administra las referencias a éstos y los libera cuando ya no se utilizan. Esta administración automática de la memoria soluciona los dos errores más comunes de las aplicaciones: la pérdida de memoria y las referencias no válidas a la memoria.

Además, el motor en tiempo de ejecución aumenta la productividad del programador. Por ejemplo, los desarrolladores pueden crear aplicaciones en el lenguaje que prefieran y seguir sacando todo el provecho del motor en tiempo de ejecución, la biblioteca de clases y los componentes escritos en otros lenguajes por otros colegas. El proveedor de un compilador puede elegir destinarlo al motor en tiempo de ejecución. Los compiladores de lenguajes que se destinan a .NET Framework hacen que las características de .NET Framework estén disponibles para el código existente escrito en dicho lenguaje, lo que facilita enormemente el proceso de migración de las aplicaciones existentes.

Aunque el motor en tiempo de ejecución está diseñado para el software del futuro, también es compatible con el software actual y el software antiguo. La interoperabilidad entre el código administrado y no administrado permite que los desarrolladores continúen utilizando los componentes COM y las DLL que necesiten.

---

<sup>11</sup> CTS: Common Type System, Sistema de tipos común.

El motor en tiempo de ejecución está diseñado para mejorar el rendimiento. Aunque Common Language Runtime proporciona muchos servicios estándar de motor en tiempo de ejecución, el código administrado nunca se interpreta. Una característica denominada compilación JIT (Just-In-Time) permite ejecutar todo el código administrado en el lenguaje máquina nativo del sistema en el que se ejecuta. Mientras tanto, el administrador de memoria evita que la memoria se pueda fragmentar y aumenta la zona de referencia de la memoria para mejorar aún más el rendimiento.

Por último, el motor en tiempo de ejecución se puede alojar en aplicaciones de servidor de gran rendimiento, como Microsoft® SQL Server™ e Internet Information Services (IIS). Esta infraestructura permite utilizar código administrado para escribir lógica empresarial, al tiempo que se disfruta del superior rendimiento de los mejores servidores empresariales del sector que puedan alojar el motor en tiempo de ejecución.

### **2.3.1.2. Biblioteca de clases de .NET Framework**

La biblioteca de clases de .NET Framework es una colección de tipos reutilizables que se integran estrechamente con Common Language Runtime. La biblioteca de clases está orientada a objetos, lo que proporciona tipos de los que su propio código administrado puede derivar funciones. Esto ocasiona que los tipos de .NET Framework sean sencillos de utilizar y reduce el tiempo asociado con el aprendizaje de las nuevas características de .NET Framework. Además, los componentes de terceros se pueden integrar sin dificultades con las clases de .NET Framework.

Por ejemplo, las clases de colección de .NET Framework implementan un conjunto de interfaces que puede usar para desarrollar sus propias clases de colección. Éstas se combinarán fácilmente con las clases de .NET Framework.

Como en cualquier biblioteca de clases orientada a objetos, los tipos de .NET Framework permiten realizar diversas tareas de programación comunes, como son la administración de cadenas, recolección de datos, conectividad de bases de datos y acceso a archivos. Además de estas tareas habituales, la biblioteca de clases incluye tipos adecuados para diversos escenarios de desarrollo especializados. Por ejemplo, puede utilizar .NET Framework para desarrollar los siguientes tipos de aplicaciones y servicios:

- Aplicaciones de consola.
- Aplicaciones GUI de Windows (Windows Forms).
- Aplicaciones de Windows Presentation Foundation (WPF).
- Aplicaciones de ASP.NET.
- Servicios web.
- Servicios de Windows.
- Aplicaciones orientadas a servicios utilizando Windows Communication Foundation (WCF).
- Aplicaciones habilitadas para el flujo de trabajo utilizando Windows Workflow Foundation (WF).

Por ejemplo, las clases de Windows Forms son un conjunto completo de tipos reutilizables que simplifican enormemente el desarrollo de interfaces GUI para Windows. Si escribe una aplicación Web Form de ASP.NET, puede utilizar las clases de formularios Web Forms.

### **2.3.2. Arquitectura de .NET Framework 3.5**

.NET Framework versión 3.5 se basa en las versiones 2.0 y 3.0 y sus Service Pack correspondientes. .NET Framework versión 3.5 Service Pack 1 actualiza los ensamblados de la versión 3.5 e incluye nuevos Service Pack para las versiones 2.0 y 3.0. En este tema se resume brevemente la relación de las versiones 2.0, 3.0 y 3.5 de .NET Framework y sus Service Packs.

#### **2.3.2.1. Relación entre las versiones 2.0, 3.0 y 3.5 de .NET Framework**

Los componentes que se enumeran a continuación se consideran parte de .NET Framework 3.5 SP1:

- .NET Framework 2.0
- Service Pack 1 y 2 de .NET Framework 2.0, que actualizan los ensamblados incluidos en .NET Framework 2.0.
- .NET Framework 3.0, que utiliza los ensamblados de .NET Framework 2.0 y sus Service Pack, e incluye los ensamblados necesarios para las tecnologías introducidas en .NET Framework 3.0. Por ejemplo, PresentationFramework.dll y PresentationCore.dll, que son necesarios para Windows Presentation Foundation (WPF), se instalan con .NET Framework 3.0.
- Service Pack 1 y 2 de .NET Framework 3.0, que actualizan los ensamblados que se introducen en .NET Framework 3.0.
- .NET Framework 3.5, que incluye nuevos ensamblados que proporcionan una funcionalidad adicional a .NET Framework 2.0 y 3.0.
- .NET Framework 3.5 Service Pack 1, que actualiza los ensamblados que se incluyen en .NET Framework 3.5.

Cada versión de .NET Framework se puede instalar independiente de las versiones posteriores y cada versión instalará automáticamente las versiones anteriores si aún no están instaladas. Además, .NET Framework 3.5 SP1 introduce el paquete de instalación Client Profile, que únicamente contiene los ensamblados que se utilizan en las aplicaciones cliente.

Una aplicación utiliza los mismos ensamblados sin tener en cuenta si tiene como destino la versión 2.0, 3.0 o 3.5 de .NET Framework o Client Profile, y sin tener en cuenta si los ensamblados se han actualizado en el equipo del usuario. Por ejemplo, una aplicación que utiliza WPF <sup>12</sup>y tiene como destino .NET Framework 3.0 utiliza la misma instancia del ensamblado mscorlib que una aplicación que utiliza formularios Windows Forms y tiene como destino .NET Framework 2.0. Si un usuario ha instalado una versión posterior de .NET Framework o un Service Pack que actualiza su copia de mscorlib.dll, las dos aplicaciones utilizarán la versión actualizada del ensamblado. (MICROSOFT, 2008)

#### **2.3.2.2. Características incluidas en .NET Framework 3.5 SP1**

En esta sección se resumen las tecnologías de .NET Framework versión 3.5 SP1 y cada una de las versiones de .NET Framework que contiene. Esta lista no es exhaustiva, solo incluye algunas de las principales tecnologías que se incluyen en .NET Framework. (MICROSOFT, 2008)

---

<sup>12</sup> **WPF:** Windows Presentation Fundation, permite el desarrollo de interfaces de interacción en Windows tomando características de aplicaciones Windows y de aplicaciones web.

## **.NET Framework 2.0**

Las tecnologías siguientes se incluyen en .NET Framework 2.0.

- Common Language Runtime (CLR) y bibliotecas de clases base.
- Compatibilidad con los tipos y métodos genéricos.
- Compiladores para C#, Visual Basic, C++ y J#.
- ADO.NET.
- ASP.NET.
- Formularios Windows Forms.
- Servicios web.

## **.NET Framework 2.0 SP 1 y SP 2**

.NET Framework 2.0 Service Pack 1 y Service Pack 2 actualizan CLR y varios ensamblados que se suministran con .NET Framework 2.0. Se puede instalar con independencia de .NET Framework 3.5 SP1. La mayoría de las actualizaciones a .NET Framework 2.0 no constituyen cambios trascendentales, aunque hay algunos pocos casos en los que se han agregado nuevos elementos de API o su comportamiento ha cambiado. Si la aplicación se basa en los cambios introducidos con .NET Framework 2.0 SP 1 o SP 2, puede utilizar .NET Framework 2.0 como destino de la aplicación y solicitar a los clientes que descarguen .NET Framework 2.0 SP 1 o SP 2. Sin embargo, si su aplicación se basa en una funcionalidad nueva o modificada, es recomendable que utilice .NET Framework 3.5 o .NET Framework Client Profile como destino de la aplicación.

## **.NET Framework 3.0**

.NET Framework 3.0 necesita que .NET Framework 2.0 esté instalado en el equipo. Si un usuario instala .NET Framework 3.0 en un equipo que no tiene .NET Framework 2.0, se instalará automáticamente. (MICROSOFT, 2008)

Las tecnologías siguientes se introducen en .NET Framework 3.0:

- Windows Presentation Foundation (WPF).
- Windows Communications Foundation (WCF).
- Windows Workflow Foundation (WF).

## **.NET Framework 3.0 SP 1 y SP 2**

.NET Framework 3.0 Service Pack 1 y Service Pack 2 actualizan algunos ensamblados que se suministran con .NET Framework 3.0 y se puede instalar con independencia de .NET Framework 3.5. Las actualizaciones incorporan cambios intrascendentes, nuevos elementos de API <sup>13</sup>y una funcionalidad adicional para las tecnologías que se incluyen en .NET Framework 3.0. Si la aplicación se basa en los cambios incorporados en .NET Framework 3.0 SP 1 o SP 2, puede utilizar .NET Framework 3.0 como destino de la aplicación y solicitar a sus clientes que descarguen .NET Framework 3.0 SP 1 o

---

<sup>13</sup> **API:** Interfaz de programación de aplicaciones.

SP 2. Sin embargo, si la aplicación se basa en una funcionalidad nueva o modificada, es conveniente que utilice .NET Framework 3.5 o .NET Framework Client Profile como destino de la aplicación.

### **.NET Framework 3.5**

.NET Framework 3.5 introduce nuevas características para las tecnologías de las versiones 2.0 y 3.0 e incorpora tecnologías adicionales en forma de nuevos ensamblados. Las tecnologías siguientes se introducen en .NET Framework 3.5:

- Language Integrated Query (LINQ).<sup>14</sup>
- Nuevos compiladores para C#, Visual Basic y C++.
- ASP.NET AJAX.

### **.NET Framework 3.5 SP1**

.NET Framework 3.5 Service Pack 1 actualiza varios ensamblados que se suministran con .NET Framework 3.5. Las actualizaciones incorporan cambios intrascendentes, nuevos elementos de API y una funcionalidad adicional para las tecnologías que se incluyen en .NET Framework 3.5. Las tecnologías siguientes se suministran con .NET Framework 3.5 SP1:

- Datos dinámicos de ASP.NET.
- ADO.NET Entity Framework.
- Compatibilidad con el proveedor de datos de SQL Server 2008.
- Compatibilidad con .NET Framework Client Profile.

#### **2.3.2.3. Common Language Runtime**

Los compiladores y las herramientas exponen la funcionalidad en tiempo de ejecución y permiten escribir código con las ventajas que proporciona este entorno de ejecución administrado. El código desarrollado con un compilador de lenguaje orientado al tiempo de ejecución se denomina código administrado. Este código se beneficia de características como: la integración entre lenguajes, el control de excepciones entre lenguajes, la seguridad mejorada, la compatibilidad con la implementación y las versiones, un modelo simplificado de interacción y servicios de creación de perfiles y depuración.

Para permitir al motor en tiempo de ejecución proporcionar servicios al código administrado, los compiladores de lenguajes deben emitir metadatos que describen los tipos, los miembros y las referencias del código. Los metadatos se almacenan con el código; cada archivo ejecutable portable de Common Language Runtime cargable contiene metadatos. El motor en tiempo de ejecución utiliza los metadatos para localizar y cargar clases, colocar instancias en memoria, resolver invocaciones a métodos, generar código nativo, exigir mecanismos de seguridad y establecer los límites del contexto en tiempo de ejecución. (MICROSOFT, 2008)

---

<sup>14</sup> **LINQ:** Language Integrated Query (LINQ, pronunciado “link”) es un componente de la plataforma Microsoft .NET que agrega capacidades de consulta a datos de manera nativa a los lenguajes .NET, si bien existen implementaciones para Java, PHP, JavaScript y ActionScript.

El tiempo de ejecución controla automáticamente la disposición de los objetos y administra las referencias a éstos, liberándolos cuando ya no se utilizan. Los objetos cuya duración se administra de esta forma se denominan datos administrados. La recolección de elementos no utilizados elimina pérdidas de memoria así como otros errores habituales de programación. Con un código administrado se pueden utilizar datos administrados, datos no administrados o estos dos tipos de datos en una aplicación .NET. Framework. Como los compiladores de lenguajes proporcionan sus propios tipos, como tipos primitivos, no siempre se sabe (o no es necesario saber) si los datos se están administrando.

Common Language Runtime facilita el diseño de los componentes y de las aplicaciones cuyos objetos interactúan entre lenguajes distintos. Los objetos escritos en lenguajes diferentes pueden comunicarse entre sí, lo que permite integrar sus comportamientos de forma precisa. Por ejemplo, puede definir una clase y, a continuación, utilizar un lenguaje diferente para derivar una clase de la clase original o llamar a un método de la clase original. También se puede pasar al método de una clase una instancia de una clase escrita en un lenguaje diferente. Esta integración entre lenguajes diferentes es posible porque los compiladores y las herramientas de lenguajes orientados al motor en tiempo de ejecución utilizan un sistema de tipos común definido por el motor en tiempo de ejecución, y los lenguajes siguen las reglas en tiempo de ejecución para definir nuevos tipos, así como para crear, utilizar, almacenar y enlazar tipos.

Como parte de los metadatos, todos los componentes administrados contienen información sobre los componentes y los recursos utilizados en su creación. El motor en tiempo de ejecución utiliza esta información para garantizar que el componente o la aplicación contiene las versiones especificadas de todo lo necesario, por lo que hay menos posibilidades de que la ejecución del código se interrumpa debido a una dependencia inadecuada. La información de registro y los datos de estado ya no se almacenan en el Registro, donde puede ser difícil establecer y mantener datos. En su lugar, la información sobre tipos definidos por el usuario (y sus dependencias) se almacena con el código como metadatos y, de este modo, las tareas de réplica y eliminación de componentes es mucho menos complicada.

Las herramientas y los compiladores de lenguajes exponen la funcionalidad del motor en tiempo de ejecución de forma que resulte útil e intuitiva para los programadores. Esto significa que algunas características en tiempo de ejecución pueden ser más evidentes en un entorno que en otro. El funcionamiento del motor en tiempo de ejecución depende de las herramientas y los compiladores utilizados. Por ejemplo, un programador de Visual Basic observará que con Common Language Runtime, el lenguaje Visual Basic contiene más características orientadas a objetos que antes. Algunas de las ventajas del motor en tiempo de ejecución son:

- Mejoras en el rendimiento.
- Capacidad para utilizar fácilmente componentes desarrollados en otros lenguajes.
- Tipos extensibles que proporciona una biblioteca de clases
- Nuevas características del lenguaje como herencia, interfaces y sobrecarga para la programación orientada a objetos; compatibilidad con el uso de subprocesos libres que permite la creación de multiprocesos; aplicaciones escalables; compatibilidad con los atributos personalizados y el control de excepciones estructurado.



Si utiliza Microsoft® Visual C++® .NET, puede escribir código administrado utilizando Visual C++, que proporcionan las ventajas de un entorno de ejecución administrado, así como el acceso a características eficaces y a tipos de datos informativos que ya le resultan familiares. Otras características del motor en tiempo de ejecución son: (MICROSOFT, 2008)

- Integración entre lenguajes diferentes y, en especial, herencia entre lenguajes.
- Recolección de elementos no utilizados, que administra la duración de los objetos de modo que no es necesario el recuento de referencias.
- Objetos auto descriptivos que hacen innecesario el Lenguaje de definición de interfaces (IDL).
- Capacidad para compilar una vez y ejecutar código en cualquier CPU y sistema operativo que sea compatible con el motor en tiempo de ejecución.

También se puede escribir código administrado en lenguaje C#, que proporciona las siguientes ventajas:

- Diseño completo orientado a objetos
- Seguridad de tipos muy sólida
- Buena combinación entre la simplicidad de Visual Basic y la eficacia de C++.
- Recolección de elementos no utilizados.
- Sintaxis y palabras clave similares en C y en C++.
- Utilice delegados para una mayor seguridad y protección de tipos, en vez de punteros a funciones. Los punteros a funciones están disponibles mediante la utilización de la palabra clave **unsafe** de C# y de la opción **/unsafe** del compilador de C# (Csc.exe) para datos y código no administrados.

#### **2.3.2.4. Sistema de tipos comunes**

##### **Clasificación de tipos**

El sistema de tipos común es compatible con dos categorías generales de tipos, que a su vez se dividen en subcategorías:

- Tipos de valor

Los tipos de valor contienen directamente sus datos y las instancias de los tipos de valor se asignan en la pila o se asignan en línea en una estructura. Los tipos de valor pueden ser integrados (implementados por el motor en tiempo de ejecución), definidos por el usuario o enumeraciones.

- Tipos de referencia

Los tipos de referencia guardan una referencia a la dirección en memoria del valor y se asignan en el montón. Los tipos de referencia pueden ser tipos auto descriptivos, de puntero o de interfaz. El tipo de un tipo de referencia se puede determinar a partir de los valores de los tipos auto descriptivos. Los tipos auto descriptivos se dividen en matrices y tipos de clase. Los tipos de clase son clases definidas por el usuario, tipos de valor a los que se ha aplicado la conversión boxing y delegados.

Las variables que son tipos de valor tienen, cada una, su propia copia de los datos y, por lo tanto, las operaciones en una variable no afectan a las demás. Las variables que son tipos de referencia pueden hacer referencia al mismo objeto y, por lo tanto, las operaciones en una variable pueden afectar al mismo objeto al que hace referencia otra variable.

Todos los tipos derivan del tipo base System.Object.

En el siguiente ejemplo se muestra la diferencia entre los tipos de referencia y los tipos de valor.

C#

```
Class Class1
```

```
    Public Value As Integer = 0
```

```
End Class 'Class1
```

```
Class Test
```

```
    Shared Sub Main()
```

```
        Dim val1 As Integer = 0
```

```
        Dim val2 As Integer = val1
```

```
        val2 = 123
```

```
        Dim ref1 As New Class1()
```

```
        Dim ref2 As Class1 = ref1
```

```
        ref2.Value = 123
```

```
        Console.WriteLine("Values: {0}, {1}", val1, val2)
```

```
        Console.WriteLine("Refs: {0}, {1}", ref1.Value, ref2.Value)
```

```
    End Sub 'Main
```

```
End Class 'Test
```

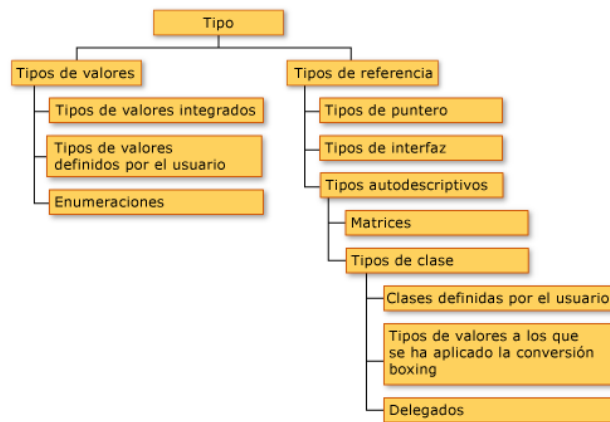
Los resultados de este programa son los siguientes:

```
Values: 0, 123
```

```
Refs: 123, 123
```

El diagrama siguiente ilustra cómo se relacionan estos distintos tipos. Tenga en cuenta que las instancias de los tipos pueden ser simplemente tipos de valor o tipos auto descriptivos, aunque haya subcategorías de estos tipos.

### 2.3.2.5. Clasificación de tipos



Fuente: (MICROSOFT, 2008)

Figura 3. Clasificación de tipos

### 2.3.2.6. Valores y objetos

Los valores son representaciones binarias de datos y los tipos proporcionan una forma de interpretar estos datos. Un tipo de valor se almacena directamente como una representación binaria de los datos del tipo. El valor de un tipo de referencia es la ubicación de la secuencia de bits que representa los datos del tipo. (MICROSOFT, 2008)

Cada valor tiene un tipo exacto que define por completo la representación del valor y las operaciones definidas en el valor. Los valores de los tipos auto descriptivos se denominan objetos. Si bien siempre se puede determinar el tipo exacto de un objeto examinando su valor, ello no se puede hacer con tipo de valor o un tipo de puntero. Un valor puede tener más de un tipo. El valor de un tipo que implementa una interfaz es también un valor de ese tipo de interfaz. De la misma manera, el valor de un tipo derivado de un tipo base es también un valor de ese tipo base.

### 2.3.2.7. Tipos y ensamblados

El motor de tiempo en ejecución utiliza ensamblados para ubicar y cargar tipos. El manifiesto del ensamblado contiene la información que el motor en tiempo de ejecución utiliza para resolver todas las referencias a tipos hechas dentro del ámbito del ensamblado.

Un nombre de tipo del motor en tiempo de ejecución tiene dos partes lógicas: el nombre del ensamblado y el nombre del tipo que se encuentra en el ensamblado. Dos tipos que tengan el mismo nombre pero estén en ensamblados distintos se definen como dos tipos diferentes.

Los ensamblados proporcionan coherencia entre el ámbito de los nombres que ve el programador y el que ve el sistema del motor en tiempo de ejecución. Los programadores escriben tipos en el contexto de

un ensamblado. El contenido del ensamblado que está creando un programador establece el ámbito de los nombres que estarán disponibles en tiempo de ejecución.

### 2.3.2.8. Tipos y espacios de nombres

Desde el punto de vista del motor en tiempo de ejecución, un espacio de nombres no es más que una colección de nombres de tipos. Algunos lenguajes pueden tener construcciones y la sintaxis correspondiente que ayudan a los programadores a formar grupos lógicos de tipos, pero el motor en tiempo de ejecución no utiliza estas construcciones al enlazar tipos. Así, las clases **Object** y **String** forman parte del espacio de nombres **System**, pero el motor en tiempo de ejecución sólo reconoce los nombres completos de cada tipo, que son `System.Object` y `System.String` respectivamente.

Se puede generar un único ensamblado que exponga tipos que parezcan proceder de dos espacios de nombres jerárquicos distintos, como `System.Collections` y `System.Windows.Forms`. También se pueden crear dos ensamblados que exporten tipos cuyos nombres contengan `MyDll.MyClass`.

Si se crea una herramienta para representar los tipos de un ensamblado como pertenecientes a un espacio de nombres jerárquico, la herramienta debe enumerar los tipos de un ensamblado o grupo de ensamblados y analizar los nombres de tipo para generar una relación jerárquica.

### 2.3.2.9. Metadatos y componentes autodescriptivos

Los metadatos son información binaria que describe un programa, almacenada en un archivo ejecutable portable (PE) de Common Language Runtime o en memoria. Cuando se compila el código en un archivo PE, los metadatos se insertan en una parte del archivo, mientras que el código se convierte en lenguaje intermedio de Microsoft (MSIL<sup>15</sup>) y se inserta en otra parte del archivo. Cada tipo y miembro definido, o al que se hace referencia, en un módulo o ensamblado se describe en los metadatos. Cuando se ejecuta código, el motor en tiempo de ejecución carga los metadatos en la memoria y hace referencia a ellos para obtener información acerca de las clases, miembros, herencia, etc., del código.

Los metadatos describen todos los tipos y miembros definidos en el código mediante un lenguaje neutro. Los metadatos almacenan la siguiente información:

- Descripción del ensamblado
  - Identidad (nombre, versión, referencia cultural, clave pública).
  - Los tipos que se exportan.

---

<sup>15</sup> **MSIL**: es el lenguaje de programación legible por humanos de más bajo nivel en el Common Language Infrastructure y en el .NET Framework. Los lenguajes del .NET Framework compilan a CIL. CIL es un lenguaje ensamblador orientado a objetos, y está basado en pilas. Es ejecutado por una máquina virtual. Los lenguajes .NET principales son C#, Visual Basic .NET, C++/CLI, y J#.

- Otros ensamblados de los que depende éste.
- Permisos de seguridad que hay que ejecutar.
- Descripción de los tipos.
- Nombre, visibilidad, clase base e interfaces implementadas.
- Miembros (métodos, campos, propiedades, eventos, tipos anidados).
- Atributos.
- Elementos descriptivos adicionales que modifiquen los tipos y los miembros.

#### Ventajas de los metadatos

Los metadatos son la clave para un modelo de programación más sencillo, eliminando la necesidad de tener archivos de Lenguaje de definición de interfaz (IDL), archivos de encabezado o cualquier método externo de referencia a componentes. Los metadatos permiten que los lenguajes .NET se describan a sí mismos automáticamente usando un lenguaje de forma neutral, que no ven ni el programador ni el usuario. Además, los metadatos se pueden extender mediante el uso de atributos. Los metadatos proporcionan las siguientes ventajas principales:

- Archivos auto descriptivos

Los módulos y ensamblados de Common Language Runtime son autodescriptivos. Los metadatos de un módulo contienen todo lo necesario para interactuar con otro módulo. Los metadatos proporcionan automáticamente la funcionalidad del IDL en COM, lo que permite usar un archivo para la definición y la implementación. Los módulos y ensamblados de Common Language Runtime no necesitan ni registrarse en el sistema operativo. En consecuencia, las descripciones que usa el motor en tiempo de ejecución reflejan siempre el código actual del archivo compilado, lo que aumenta la confiabilidad de la aplicación.

- Interoperabilidad de lenguajes y diseño más sencillo, basado en el componente.

Los metadatos proporcionan toda la información necesaria sobre el código compilado para derivar clases de archivos PE escritos en otro lenguaje. Se puede crear una instancia de cualquier clase escrita en cualquier lenguaje administrado (cualquier lenguaje dirigido a Common Language Runtime) sin tener que preocuparse por el cálculo de referencias explícito ni por usar código de interoperabilidad personalizado.

- Atributos.

.NET Framework permite declarar tipos específicos de metadatos, denominados atributos, en el archivo compilado. Los atributos se encuentran en todo .NET Framework y se usan para controlar más minuciosamente el comportamiento del programa en tiempo de ejecución. Además, se pueden

emitir metadatos personalizados propios en los archivos .NET Framework mediante atributos personalizados definidos por el usuario.

### **2.3.2.10. Interoperabilidad entre lenguajes**

La interoperabilidad entre lenguajes es la posibilidad de que el código interactúe con código escrito en un lenguaje de programación diferente. La interoperabilidad entre lenguajes puede ayudar a maximizar la reutilización de código y, por tanto, puede mejorar la eficacia del proceso de programación. (MICROSOFT, 2008)

Dado que los desarrolladores utilizan una gran variedad de herramientas y tecnologías, cada una de las cuales podría admitir distintos tipos y características, desde tiempo atrás ha sido complicado garantizar la interoperabilidad entre lenguajes. No obstante, los compiladores y las herramientas de lenguaje dirigidos a Common Language Runtime se benefician de la compatibilidad que integra el motor en tiempo de ejecución para la interoperabilidad entre lenguajes.

Common Language Runtime ofrece la base necesaria para la interoperabilidad entre lenguajes al especificar e imponer un sistema de tipos común, y al suministrar metadatos. Debido a que todos los lenguajes dirigidos a Common Language Runtime siguen las reglas del sistema de tipos común para definir y utilizar los tipos, la utilización de tipos es coherente entre todos los lenguajes. Los metadatos habilitan la interoperabilidad entre lenguajes mediante la definición de un mecanismo uniforme para almacenar y recuperar la información sobre tipos. Los compiladores almacenan la información sobre tipos como metadatos y Common Language Runtime usa esta información para proporcionar servicios durante la ejecución; el motor en tiempo de ejecución puede administrar la ejecución de aplicaciones de varios lenguajes porque toda la información de tipos se almacena y recupera de la misma forma, independientemente del lenguaje en que se haya escrito el código.

El código administrado se beneficia de que el motor en tiempo de ejecución admita la interoperabilidad entre lenguajes de las maneras siguientes:

- Los tipos pueden heredar la implementación de otros tipos, pasar objetos a los métodos de otro tipo y llamar a métodos definidos para otros tipos, sea cual sea el lenguaje en que se implementen los tipos.
- Los depuradores, generadores de perfiles u otras herramientas deben reconocer un solo entorno, es decir, MSIL (Microsoft intermediate language, Lenguaje intermedio de Microsoft) y los metadatos de Common Language Runtime, para poder ser compatibles con cualquier lenguaje de programación dirigido al motor en tiempo de ejecución.
- El control de excepciones es coherente entre todos los lenguajes. El código puede producir una excepción en un lenguaje y esa excepción puede ser recibida y reconocida por un objeto escrito en otro lenguaje.

Aunque el motor en tiempo de ejecución permite que todo el código administrado se ejecute en un entorno de múltiples lenguajes, no existe ninguna garantía de que la funcionalidad de los tipos creados pueda ser utilizada completamente por los lenguajes de programación que usen otros desarrolladores. La razón principal es que el compilador de un lenguaje dirigido a Common Language Runtime usa el sistema de tipos y los metadatos que son compatibles con un conjunto de características de lenguaje propio y exclusivo. Si no se sabe en qué lenguaje se va a escribir el código que realiza la llamada, existen pocas posibilidades de saber si las características que expone un componente van a estar accesibles para el llamador. Por ejemplo, si el lenguaje que ha elegido proporciona compatibilidad para enteros sin signo, podría diseñar un método con un parámetro de tipo UInt32; sin embargo, ese método sería inútil para un lenguaje que no reconociese los enteros sin signo.

Para garantizar que el código administrado será accesible para los desarrolladores que utilicen cualquier lenguaje de programación, .NET Framework proporciona Common Language Specification (CLS), que describe un conjunto fundamental de características de lenguaje y define reglas sobre cómo utilizar dichas características.

### **2.3.2.11. Ensamblados en Common Language Runtime**

Los ensamblados son una parte fundamental de la programación con .NET Framework. Un ensamblado realiza las funciones siguientes: (MICROSOFT, 2008)

- Contiene el código que ejecuta Common Language Runtime. El código del lenguaje intermedio de Microsoft (MSIL) de un archivo ejecutable portable (PE) no se ejecuta si no tiene asociado un manifiesto de ensamblado. Hay que tener en cuenta que cada ensamblado sólo puede tener un punto de entrada (es decir, DllMain, WinMain o Main).
- Crea un límite de seguridad. Un ensamblado es la unidad en la que se solicitan y conceden los permisos.
- Crea un límite de tipos. La identidad de cada tipo incluye el nombre del ensamblado en que reside. Por ello, un tipo MyType cargado en el ámbito de un ensamblado no es igual que un tipo denominado MyType cargado en el ámbito de otro ensamblado.
- Crea un límite de ámbito de referencia. El manifiesto del ensamblado contiene los metadatos del ensamblado que se utilizan para resolver tipos y satisfacer las solicitudes de recursos. Especifica los tipos y recursos que se exponen fuera del ensamblado. El manifiesto también enumera otros ensamblados de los que depende.
- Forma un límite de versión. El ensamblado es la unidad versionable más pequeña de Common Language Runtime; todos los tipos y recursos del mismo ensamblado pertenecen a la misma

versión. El manifiesto del ensamblado describe las dependencias de versión que se especifiquen para los ensamblados dependientes.

- Crea una unidad de implementación. Cuando se inicia una aplicación, sólo deben estar presentes los ensamblados a los que llama la aplicación inicialmente. Los demás ensamblados, como los recursos de localización o los ensamblados que contengan clases de utilidad, se pueden recuperar a petición. De este modo, se puede mantener la simplicidad y transparencia de las aplicaciones la primera vez que se descargan.
- Es la unidad que permite la ejecución simultánea

Los ensamblados pueden ser estáticos o dinámicos. Los ensamblados estáticos pueden incluir tipos de .NET Framework (interfaces y clases), así como recursos para el ensamblado (mapas de bits, archivos JPEG, archivos de recursos, etc.). Los ensamblados estáticos se almacenan en el disco, en archivos ejecutables portables PE. También se puede utilizar .NET Framework para crear ensamblados dinámicos, que se ejecutan directamente desde la memoria y no se guardan en el disco antes de su ejecución. Los ensamblados dinámicos se pueden guardar en el disco una vez que se hayan ejecutado.

Existen varias formas de crear ensamblados. Puede utilizar herramientas de desarrollo, como Visual Studio 2005, que haya empleado anteriormente para crear archivos .dll o .exe. Puede utilizar las herramientas suministradas en Kit de desarrollo de software de Windows (SDK) para generar ensamblados con módulos creados en otros entornos de programación. También puede utilizar las API de Common Language Runtime, como Reflection.Emit, para crear ensamblados dinámicos.

### **2.3.2.12. Dominios de aplicación**

Tradicionalmente se han utilizado límites de proceso para aislar las aplicaciones que se ejecutan en un mismo equipo. Cada aplicación se carga en un proceso independiente que aísla la aplicación de las demás que se estén ejecutando en el mismo equipo. (MICROSOFT, 2008)

Las aplicaciones se aíslan porque las direcciones de memoria son específicas de cada proceso; un puntero de memoria pasado de un proceso a otro no se puede utilizar de ninguna manera coherente en el proceso de destino. Tampoco se pueden realizar llamadas directas entre dos procesos. En su lugar, se deben utilizar servidores proxy, que proporcionan un nivel de direccionamiento indirecto.

El código administrado debe pasar por un proceso de verificación para poder ejecutarse (a menos que el administrador haya concedido permiso para omitir la comprobación). El proceso de comprobación determina si el código puede intentar el acceso a direcciones de memoria no válidas o realizar alguna otra acción que pudiera hacer que el proceso en el que se ejecuta deje de funcionar correctamente. Cuando el código pasa la prueba de comprobación, se dice que tiene seguridad de tipos. La posibilidad de comprobar la seguridad de tipos del código permite que Common Language Runtime proporcione un gran nivel de seguridad respecto a los límites del proceso, con un costo de rendimiento mucho menor.



Los dominios de aplicación constituyen una unidad de procesamiento más segura y versátil que puede utilizar Common Language Runtime para proporcionar el aislamiento entre las aplicaciones. En un solo proceso se pueden ejecutar varios dominios de aplicación con el mismo nivel de aislamiento que existiría en procesos independientes, sin incurrir en la sobrecarga adicional que supone realizar llamadas entre procesos o cambiar de un proceso a otro. La posibilidad de ejecutar múltiples aplicaciones en un solo proceso aumenta la escalabilidad del servidor de manera importante.

Aislar las aplicaciones es también importante para la seguridad de las mismas. Por ejemplo, en un solo proceso de explorador se pueden ejecutar controles de varias aplicaciones Web de tal forma que no puedan tener acceso a los datos y recursos de los demás controles.

Éstas son las ventajas del aislamiento que ofrecen los dominios de aplicación:

- Los errores de una aplicación no pueden afectar a otras aplicaciones. Debido a que el código seguro no puede generar problemas de memoria, el uso de dominios de aplicación garantiza que el código que se ejecute en un dominio no afectará a las demás aplicaciones del proceso.
- Es posible detener aplicaciones concretas sin detener todo el proceso. El uso de dominios de aplicación permite descargar el código que se ejecuta en una sola aplicación.
- El código que se ejecuta en una aplicación no puede tener acceso directo al código o a los recursos de otra aplicación. Common Language Runtime impone este aislamiento al impedir que se realicen llamadas directas entre objetos de dominios de aplicación diferentes. Los objetos que se pasan entre dominios se copian o se obtiene acceso a ellos mediante proxy. Si el objeto se copia, la llamada al objeto es local. En otras palabras, el llamador y el objeto al que se hace referencia se encuentran en el mismo dominio de aplicación. Si se tiene acceso al objeto a través de un proxy, la llamada al objeto es remota. En este caso, el llamador y el objeto al que se hace referencia se encuentran en dominios de aplicación diferentes. En las llamadas entre dominios se utiliza la misma infraestructura de llamada remota que en las llamadas entre dos procesos o entre dos equipos. En consecuencia, los metadatos del objeto al que se hace referencia deben estar disponibles para ambos dominios de aplicación a fin de que la llamada al método no provoque un error en la compilación JIT. Si el dominio que llama no tiene acceso a los metadatos del objeto al que se está llamando, se podría producir un error de compilación con una excepción del tipo **System.IO.FileNotFoundException**. El objeto es quien decide el mecanismo para determinar cómo se puede obtener acceso a los objetos entre dominios. (MICROSOFT, 2008)
- La aplicación en la que se ejecuta el código establece el comportamiento del mismo. En otras palabras, el dominio de aplicación proporciona valores de configuración tales como las directivas de versión de la aplicación, la ubicación de los ensamblados remotos a los que tiene acceso e información sobre dónde encontrar los ensamblados que se cargan en el dominio.
- El dominio de aplicación en el que se ejecuta el código puede controlar los permisos que se conceden al código.

### 2.3.2.13. Información general de la biblioteca de clases de .NET Framework

.NET Framework incluye clases, interfaces y tipos de valores que agilizan y optimizan el proceso de desarrollo y proporcionan acceso a las funciones del sistema. Para facilitar la interoperabilidad entre lenguajes, la mayoría de los tipos de .NET Framework cumplen la especificación de lenguaje común (CLS) y, por tanto, se pueden utilizar en todos los lenguajes de programación cuyo compilador satisfaga los requisitos de CLS.

Los tipos de .NET Framework son la base sobre la que se crean aplicaciones, componentes y controles de .NET. .NET Framework incluye tipos que realizan las funciones siguientes:

- Representar tipos de datos base y excepciones.
- Encapsular estructuras de datos.
- Realizar E/S.
- Obtener acceso a información sobre tipos cargados.
- Invocar las comprobaciones de seguridad de .NET Framework.
- Proporcionar: acceso a datos, interfaz gráfica para el usuario (GUI) independiente de cliente e interfaz GUI de cliente controlada por el servidor.

.NET Framework proporciona un conjunto completo de interfaces, así como clases abstractas y concretas (no abstractas). Se pueden utilizar las clases concretas tal como están o, en muchos casos, derivar las clases propias de ellas. Para utilizar la funcionalidad de una interfaz se puede crear una clase que implemente la interfaz o derivar una clase de una de las clases de .NET Framework que implementa la interfaz.

#### Convenciones de nomenclatura

Los tipos de .NET Framework utilizan un esquema de nomenclatura con sintaxis de punto lo que indica la existencia de una jerarquía. Esta técnica agrupa tipos relacionados en espacios de nombres para que se pueda buscar y hacer referencia a ellos más fácilmente. La primera parte del nombre completo, hasta el punto situado más a la derecha, es el nombre del espacio de nombres. La última parte es el nombre de tipo. Por ejemplo, **System.Collections.ArrayList** representa el tipo **ArrayList** que pertenece al espacio de nombres **System.Collections**. Los tipos de **System.Collections** se pueden utilizar para manipular colecciones de objetos.

Este esquema de nomenclatura facilita a los programadores de bibliotecas la tarea de extender .NET Framework para poder crear grupos jerárquicos de tipos y asignarles nombre de forma coherente e ilustrativa. También permite identificar de forma inequívoca los tipos mediante su nombre completo (es decir, por su espacio de nombres y nombre de tipo), lo que evita que se produzcan conflictos entre los nombres de tipo. Se supone que los programadores de bibliotecas utilizarán la siguiente directriz cuando creen nombres para sus propios espacios de nombres:

*NombreCompañía.NombreTecnología*

Por ejemplo, el espacio de nombres Microsoft.Word cumple esta directriz.

El uso de modelos de nomenclatura para agrupar tipos relacionados en espacios de nombres es una forma muy útil de crear y documentar bibliotecas de clases. Sin embargo, este esquema de nomenclatura no influye en la visibilidad, el acceso a miembros, la herencia, la seguridad o el enlace. Se puede hacer la partición de un espacio de nombres en varios ensamblados y un ensamblado individual puede contener tipos de varios espacios de nombres. El ensamblado proporciona la estructura formal para el control de versiones, la implementación, la seguridad, la carga y la visibilidad en Common Language Runtime.

### **System (Espacio de nombres)**

El espacio de nombres System es el espacio de nombres de la raíz de los tipos fundamentales de .NET Framework. Este espacio de nombres contiene clases que representan los tipos de datos base que se utilizan en todas las aplicaciones: Object (raíz de la jerarquía de herencia), Byte, Char, Array, Int32, String, etc. Muchos de estos tipos se corresponden con los tipos de datos primitivos que utiliza el lenguaje de programación. Cuando se escribe código utilizando tipos de .NET Framework se puede utilizar la palabra clave correspondiente del lenguaje cuando se espera un tipo de datos base de .NET Framework.

Además de los tipos de datos base, el espacio de nombres System contiene más de 100 clases, que comprenden desde las clases que controlan excepciones hasta las clases que tratan conceptos básicos en tiempo de ejecución, como los dominios de aplicación y el recolector de elementos no utilizados<sup>16</sup>. El espacio de nombres System también contiene muchos espacios de nombres de segundo nivel.

#### **2.3.2.14. Hosts del motor en tiempo de ejecución**

Common Language Runtime se ha diseñado para admitir distintos tipos de aplicaciones, desde aplicaciones de servidor Web hasta aplicaciones con una interfaz de usuario eficaz y tradicional de Windows. Cada tipo de aplicación requiere un host de motor en tiempo de ejecución que la inicie. El host de motor en tiempo de ejecución carga el motor en tiempo de ejecución en un proceso, crea los dominios de aplicación en el proceso y carga el código de usuario en los dominios de aplicación.

.NET Framework incluye varios hosts de motor en tiempo de ejecución, incluidos los de la tabla siguiente.

<b>Host del motor en tiempo de ejecución</b>	<b>Descripción</b>
--	--------------------

<sup>16</sup> **Recolector de elementos no utilizados:** administra la asignación y liberación de la memoria de la aplicación.

ASP.NET	Carga el motor en tiempo de ejecución en el proceso que va a controlar la solicitud Web. ASP.NET crea también un dominio de aplicación para cada aplicación Web que se vaya a ejecutar en un servidor Web.
Microsoft Internet Explorer	Crea dominios de aplicación en los que ejecutar controles administrados. .NET Framework admite la descarga y ejecución de controles de explorador. El motor en tiempo de ejecución interactúa con el mecanismo de extensibilidad de Microsoft Internet Explorer a través de un filtro MIME para crear los dominios de aplicación donde se van a ejecutar los controles administrados. De manera predeterminada, se crea un dominio de aplicación para cada sitio Web.
Ejecutable del shell <sup>17</sup>	Invoca el código que aloja el motor en tiempo de ejecución para transferir el control al motor en tiempo de ejecución cada vez que se inicia la ejecución de un archivo desde el shell.

Fuente: (MICROSOFT, 2008)

Tabla 7. Motor en tiempo de ejecución CLR.

## 2.4. ASP.net con VB.net

ASP.NET es un modelo de desarrollo Web unificado que incluye los servicios necesarios para crear aplicaciones Web empresariales con el código mínimo. ASP.NET forma parte de .NET Framework y al codificar las aplicaciones ASP.NET tiene acceso a las clases en .NET Framework. El código de las aplicaciones puede escribirse en cualquier lenguaje compatible con el Common Language Runtime (CLR), entre ellos Microsoft Visual Basic, C#, JScript .NET y J#. Estos lenguajes permiten desarrollar aplicaciones ASP.NET que se benefician del Common Language Runtime, seguridad de tipos, herencia<sup>18</sup>, etc.

<sup>17</sup> **Shell:** programas que proveen una interfaz de usuario para acceder a los servicios del sistema operativo.

<sup>18</sup> **Herencia.** el mecanismo más utilizado para alcanzar algunos de los objetivos más preciados en el desarrollo de software como lo son la reutilización y la extensibilidad.

ASP.NET incluye:

- Marco de trabajo de página y controles
- Compilador de ASP.NET
- Infraestructura de seguridad
- Funciones de administración de estado
- Configuración de la aplicación
- Supervisión de estado y características de rendimiento
- Capacidad de depuración
- Marco de trabajo de servicios Web XML
- Entorno de host extensible y administración del ciclo de vida de las aplicaciones
- Entorno de diseñador extensible

#### **2.4.1. Marco de trabajo de páginas y controles**

El marco de trabajo de páginas y controles ASP.NET es un marco de trabajo de programación que se ejecuta en un servidor Web para generar y representar de forma dinámica páginas Web ASP.NET. Las páginas Web ASP.NET se pueden solicitar a cualquier explorador o dispositivo del cliente y ASP.NET representa el marcado (como HTML) al explorador que realizó la solicitud. Como norma, puede utilizar la misma página para varios exploradores, porque ASP.NET representa el marcado adecuado para el explorador que realiza la solicitud. Sin embargo, puede diseñar una página Web ASP.NET para ejecutarse en un explorador determinado, como Microsoft Internet Explorer 6, y aprovechar así todas las características de ese explorador. ASP.NET es compatible con los controles móviles de los dispositivos preparados para trabajar en Web como teléfonos celulares, PC portátiles y asistentes digitales personales (PDA).

Las páginas Web ASP.NET están completamente orientadas a objetos. En las páginas Web ASP.NET se puede trabajar con elementos HTML que usen propiedades, métodos y eventos. El marco de trabajo de páginas ASP.NET quita los detalles de implementación relacionados con la separación de cliente y servidor inherente a las aplicaciones Web presentando un modelo unificado que responde a los eventos de los clientes en el código que se ejecuta en el servidor. El marco de trabajo también mantiene automáticamente el estado de la página y de los controles que contenga durante el ciclo vital de procesamiento de la página.

El marco de trabajo de páginas y controles ASP.NET también permite encapsular la funcionalidad común de la interfaz de usuario en controles fáciles de usar y reutilizables. Los controles se escriben una vez, se pueden utilizar en varias páginas y se integran en la página Web ASP.NET en la que se colocan durante la representación.

El marco de trabajo de páginas y controles ASP.NET también proporciona funciones para controlar la apariencia y el funcionamiento general de los sitios Web a través de temas y máscaras. Se pueden definir temas y máscaras y, a continuación, aplicarlos en las páginas o controles.

Además de los temas, es posible definir páginas principales que se crean para conseguir un diseño coherente en las páginas de la aplicación. Una página principal única define el diseño y el comportamiento estándar deseados para todas las páginas (o un grupo de páginas) de la aplicación. A continuación, se pueden crear páginas de contenido individuales con el contenido específico de la página que se desee mostrar. Cuando los usuarios solicitan las páginas de contenido, las combinan con la página principal con el fin de generar un resultado que combine el diseño de la página principal con el de la página de contenido.

## **2.4.2. Compilador de ASP.NET**

Compila todo el código de ASP.NET, lo que permite el establecimiento inflexible de tipos, las optimizaciones de rendimiento y el enlace en tiempo de compilación, entre otras ventajas. Una vez que se ha compilado el código, el Common Language Runtime compila una vez más código de ASP.NET en código nativo, lo que permite un mayor rendimiento.

ASP.NET incluye un compilador que compilará todos los componentes de la aplicación, incluidas las páginas y los controles, en un ensamblado que el entorno de host de ASP.NET puede utilizar a continuación para atender las solicitudes del usuario. (MICROSOFT, 2008)

### **2.4.2.1. Infraestructura de seguridad**

Además de las características de seguridad de .NET, ASP.NET proporciona una infraestructura de seguridad avanzada para autenticar y autorizar el acceso de los usuarios y realizar otras tareas relacionadas con la seguridad. Puede autenticar usuarios con la autenticación de Windows suministrada por IIS o puede administrar la autenticación con su propia base de datos de usuario utilizando la autenticación mediante formularios ASP.NET y la suscripción ASP.NET. Además, puede administrar la autorización a las capacidades e información de su aplicación Web mediante los grupos de Windows o su propia base de datos de funciones personalizada utilizando las funciones de ASP.NET. Resulta fácil quitar, agregar o reemplazar estos esquemas dependiendo de las necesidades de la aplicación.

ASP.NET siempre se ejecuta con una identidad particular de Windows de modo que puede asegurar su aplicación utilizando las capacidades de Windows como, por ejemplo, las listas de control de acceso (ACL) de NTFS, permisos de la base de datos, etc.

### **2.4.2.2. Funciones de administración de estado**

ASP.NET proporciona funcionalidad de administración de estado intrínseca que permite almacenar información entre las solicitudes de página, como la información de clientes o el contenido del carro de la compra. Puede guardar y administrar información específica de la aplicación, específica de la sesión, específica de la página, específica del usuario y definida por el desarrollador. Esta información puede ser independiente de cualquier control de la página,

ASP.NET ofrece funciones de estado distribuidas, lo que le permite administrar información de estado en múltiples instancias de la misma aplicación en un equipo o en varios.

### **2.4.2.3. Configuración de ASP.NET**

Las aplicaciones ASP.NET utilizan un sistema de configuración que le permite definir valores de configuración para su servidor Web, para un sitio Web o para aplicaciones individuales. Puede crear valores de configuración cuando se implementan las aplicaciones ASP.NET y puede agregar o revisar los valores de configuración en cualquier momento con un impacto mínimo en aplicaciones y servidores Web de operaciones. Los valores de configuración de ASP.NET se almacenan en archivos basados en la tecnología XML. Dado que estos archivos XML son archivos de texto ASCII, es fácil realizar cambios de configuración a sus aplicaciones Web. Puede extender el esquema de configuración para satisfacer sus requisitos. (MICROSOFT, 2008)

### **2.4.2.4. Supervisión de estado y características de rendimiento**

ASP.NET incluye características que le permiten supervisar el estado y el rendimiento de su aplicación ASP.NET. La supervisión del estado de ASP.NET permite proporcionar información sobre eventos clave que proporcionan información sobre el estado de una aplicación y sobre las condiciones de error. Estos eventos muestran una combinación de diagnósticos y características de supervisión, a la vez que proporcionan un elevado grado de flexibilidad en lo que respecta a lo que se registra y cómo.

ASP.NET admite dos grupos de contadores de rendimiento a los que pueden obtener acceso las aplicaciones:

- El grupo de contadores de rendimiento del sistema ASP.NET
- El grupo de contadores de rendimiento de la aplicación ASP.NET

### **2.4.2.5. Capacidad de depuración**

ASP.NET aprovecha la infraestructura de depuración en tiempo de ejecución para permitir la depuración entre lenguajes y equipos. Se pueden depurar tanto objetos administrados como no administrados, así como todos los lenguajes compatibles con el Common Language Runtime y los lenguajes de script.

Además, el marco de trabajo de páginas ASP.NET proporciona un modo de seguimiento que permite insertar mensajes de instrumentalización en las páginas Web ASP.NET.

### **2.4.2.6. Marco de trabajo de servicios Web XML**

ASP.NET es compatible con los servicios Web XML. Un servicio Web XML es un componente que incluye funcionalidad de empresa que permite a las aplicaciones intercambiar información entre firewalls utilizando estándares como los servicios de mensajería HTTP y XML. Los servicios Web XML no están relacionados con ninguna tecnología de componentes ni con ninguna convención de llamada a objetos en concreto. Como resultado, pueden obtener acceso a los servicios Web XML los programas escritos en cualquier lenguaje, que usen cualquier modelo de componentes y se ejecuten en cualquier sistema operativo.

### **2.4.2.7. Entorno de host extensible y administración del ciclo de vida de las aplicaciones**

ASP.NET incluye un entorno de host extensible que controla el ciclo de vida de una aplicación desde el momento en que un usuario cualquiera tiene acceso a un recurso (como una página) en la aplicación hasta el momento en que se cierra la aplicación. Aunque ASP.NET se basa en un servidor Web (IIS) como un host de la aplicación, ASP.NET proporciona gran parte de la propia funcionalidad de host. La arquitectura de ASP.NET permite responder a los eventos de aplicación y crear controladores y módulos HTTP personalizados.

#### **Entorno de diseñador extensible**

ASP.NET incluye la compatibilidad mejorada para crear diseñadores de controles de servidor Web para utilizarlos con una herramienta de diseño visual como Visual Studio. Los diseñadores permiten crear una interfaz de usuario en tiempo de diseño para un control; de este modo, los desarrolladores pueden configurar las propiedades y el contenido del control en una herramienta de diseño visual.

## **2.5. Librerías Developer Express**

Los siguientes productos independientes se incluyen en la suite para DXperience ASP.NET: (DevExpress, 2009)

- ASPxperience
- ASPxEditors
- ASPxGauges
- ASPxGridView
- ASPxHtmlEditor
- ASPxPivotGrid
- ASPxScheduler
- ASPxSpellChecker
- ASPxTreeList
- XtraCharts
- XtraReports

### **2.5.1. ASPxperience**

Esta suite incluye controles ligeros que generan el código mínimo posible, guías de carga sólo cuando son necesarios, optimiza el uso del estilo, y ofrece un completo API del lado del cliente.

### **2.5.2. ASPxEditors**

- **Una amplia gama de controles.**

Hay más de 20 controles incluidos en la suite. El gran número de controles a disposición asegura que se puede encontrar el que ofrece la mejor experiencia para los usuarios finales.



- **Adaptabilidad del lado del cliente.**

Para reducir el riesgo de error, y hacer que sea más fácil para los usuarios finales para introducir datos, es posible que deba cambiar dinámicamente el contenido en los editores de datos. Puede que tenga que filtrar los valores en los cuadros de lista o cuadros combinados y rellenar con diferentes valores dependiendo de otros controles. Para proporcionar una interfaz de usuario superior, estas operaciones deben realizarse en el cliente.

- **Incorporado en el mecanismo de validación.**

A pesar de que los controles son totalmente compatibles con el estándar de controles de validación, también tienen integrada una función de validación. Se recomienda utilizar la última opción.

- **SharePoint Support.**

Se puede mejorar las soluciones de SharePoint <sup>19</sup> mediante el uso de la potente funcionalidad de la **suite ASPxGridView y Editores de controles**

### 2.5.3. ASPxGauges

El ASPxGaugeControl le permite mostrar indicadores y cuadros de mando digitales en las aplicaciones basadas en web.

Sus principales características son las siguientes:

- Un poderoso paquete de medidores digitales le permiten imitar cuadros de mando integrales, diales, termómetros y pantallas LED.
- Simular indicadores simples utilizando indicadores del estado de los indicadores.
- Escalas múltiples dentro de un solo medidor.
- Más de 180 listas para el uso de pre ajustes de calibración.
- Insertar varios medidores de los mismos o diferentes tipos de control de modelo en uno, y consistentemente asegurar sin solapamiento.
- Manualmente construir indicadores a partir de unidades básicas.
- Zoom suave de manómetros. Los calibradores están todos basados en gráficos vectoriales escalables, por lo que puede ser estirado o exprimido de forma segura sin ninguna pérdida de calidad
- Apariencia personalización de casi todos los elementos de calibre.
- Fácil de usar y el diseño le permite construir indicadores de una manera intuitiva.
- Acceso rápido a las herramientas en tiempo de diseño a través de las etiquetas inteligentes.
- Impresión (a través de la Biblioteca XtraPrinting).
- Exportar a PDF, XLS, MHT, HTML y formatos de imagen.

### 2.5.4. ASPxGridView

#### Datos Específicos

---

<sup>19</sup> **SharePoint:** es una plataforma de colaboración empresarial, formada por productos y elementos de software.

- Unbound Columns: Sólo tiene que añadir columnas sin enlazar y suministrar datos a través de un evento. Los usuarios finales pueden editar los valores en las columnas sin enlazar y los cambios se pueden guardar en un origen de datos personalizado. (DevExpress, 2009)
- Server-side Data Management: ha sido diseñado para trabajar con conjuntos de datos de gran tamaño (que consiste en 50.000 registros e incluso más). Bound data no se carga en memoria en su totalidad. En su lugar, ASPxGridView carga los datos en porciones pequeñas en la demanda. Cuando la red se inicia, carga y muestra la primera parte de datos de un almacén de datos. Entonces, como un usuario se desplaza al final de la red, cuando el último registro del subconjunto cargado se alcanza, las solicitudes de la red de datos de la tienda para devolver una porción posterior de los datos.
- Maestro-Detalle: ASPxGridView le permite presentar en maestro-detalle la información de cualquier complejidad.

### La edición de datos

- Two Edit Modes: ASPxGridView ofrece dos modos integrados de edición: a través del formulario de edición y edición de In-Line.
- Data Editors (text box, check editor, date editor, etc.): se puede utilizar como independiente o en su lugar dentro de ASPxGridView
- Row Validation and Error Indication: ASPxGridView le permite validar manualmente una fila editada y mostrar los iconos de error para campos no válidos.

### De forma de datos y manipulación

- Automatic Data Grouping for Any Datasource: permite a los usuarios agrupar los datos contra un número ilimitado de columnas, dándoles una capacidad sin precedentes para analizar y editar información.
- Full Data Summaries: le permiten presentar la información estadística como MIN, MAX, AVG, SUM y COUNT directamente en la propia red, sin tener que recurrir a la solución fea vieja de utilizar un control individual de edición. Lo que es aún mejor, es que se puede controlar los eventos de sumarios específicos para implementar funciones complejas con facilidad. Desviación estándar y todo tipo de otras fórmulas estadísticas ya están disponibles.
- Custom Text for Summary Values: puede proporcionar un texto de presentación personalizado para cualquier valor de resumen (grupo o resumen del total), lo que se utilizará al exportar una cuadrícula.
- Filtering by Values and Display Text: se puede especificar para cada columna de la cantidad de datos que la columna debería ser resuelto - por valores o editar los valores mostrados.
- Preview Rows: con filas de vista previa, puedes dejar que tus usuarios finales ver texto grande o directamente en los campos memo ASPxGridView, sin la necesidad de abrir un editor externo o una ventana desplegable.
- Auto Filter Row: esto permite a los usuarios filtrar los datos en el momento, al escribir valores directamente en esta fila.
- Runtime Column Customization: con ASPxGridView, los usuarios finales pueden controlar la apariencia de la red mediante la adición y eliminación de columnas mediante drag-and-drop<sup>20</sup>.
- Band Columns: multi-fila ASPxGridView cuentan con cabeceras que permiten organizar columnas de la cuadrícula en grupos lógicos (bandas).

---

<sup>20</sup> **drag-and-drop:** poder arrastrar y mover los controles.

- Multiple Records Selection: ASPxGridView compatible con los modos de fila única y múltiple selección.

### **El enfoque y la navegación**

- Built-in Pager: ASPxGridView automáticamente divide el contenido en varias páginas y proporciona el paginador incorporado que permite a los usuarios navegar a través de los datos.
- Vertical Scrolling Mode: para reducir la altura de una cuadrícula, puede mostrar su barra de desplazamiento vertical y especificar la altura del área de desplazamiento. (DevExpress, 2009)

### **Apariencia de personalización**

- Appearances: es un mecanismo poderoso para controlar todo el look-and-feel<sup>21</sup> de una cuadrícula. Se puede personalizar la apariencia de casi cualquier elemento visual de ASPxGridView, e incluso aplicar condicionalmente para reflejar ciertos estados, tales como errores. Además, ASPxGridView cuenta con un número predefinido de temas que le permiten especificar su apariencia y comportamiento.
- Templates Technology: las plantillas son totalmente compatibles para cada elemento dentro de ASPxGridView, se puede personalizar por completo cualquier aspecto o diseño utilizando HTML o cualquier control de servidor.
- Conditional Formatting: ASPxGridView tiene un mecanismo flexible con el que controla celdas condicionales y el formato de fila.
- Full CSS Support: en ASPxGridView la apariencia se puede personalizar mediante clases CSS. Este enfoque es la mejor opción si necesita un control centralizado sobre la aparición de varios controles o un sitio entero.

### **Misceláneo**

- Compatibilidad entre navegadores: Internet Explorer, Mozilla Firefox, Google Chrome, Opera, Apple Safari, etc El control ASPxGridView es cross-browser compatible.
- Funcionalidad Rich Client-Side: una base general del lado del cliente API está disponible para ASPxGridView. Esta funcionalidad, que se acopla con la capacidad de manejar los eventos en el lado del cliente, le permite tener un control total sobre el comportamiento de un control en el lado del cliente.
- Los eventos del cliente: ASPxGridView expone un amplio conjunto de eventos del cliente. Estos eventos se ejecutan en el lado del cliente y han sido diseñados para maximizar el rendimiento.
- Cookies Support: el control ASPxGridView es totalmente compatible con las cookies que permiten a los visitantes de su sitio web personalizar páginas. Si las cookies están habilitadas, cada posible usuario final tiene la opción de personalización, se guarda y se restaura en sesiones futuras.
- Guardar y restaurar Layout: ASPxGridView le permite guardar la información en su diseño a una base de datos y luego restaurarlo.
- Exportar: ASPxGridView le permite exportar sus datos como un archivo PDF, XLS y RTF.
- Soporte SEO: ASPxGridView le permite hacer su busca personas SEO-amigoso.
- AJAX como apoyo tecnológico.

---

<sup>21</sup> **look-and-feel:** interfaz gráfico y la presentación.

### 2.5.5. ASPxHtmlEditor

La suite ASPxHtmlEditor le permite habilitar la edición de texto rico en sitios web. Para lo que sea que la necesite por ejemplo escribir entradas de blog, componer mensajes de correo electrónico o simplemente el intercambio de documentos con formato, simplemente se coloca un control en un formulario y obtiene un editor de texto con una interfaz de usuario intuitiva que se parece mucho a Microsoft® Word. El editor ofrece las mismas barras de herramientas y los mismos atajos de teclado, no hay necesidad de aprender algo nuevo.

Además de ser un editor WYSIWYG<sup>22</sup>, el ASPxHtmlEditor control admite entrada HTML. Usted no tiene que elegir el método de entrada que desea utilizar, porque no hay una sincronización bidireccional entre el editor visual y el código HTML. Los usuarios finales pueden cambiar a la vista de código y editar documentos, tener control manual de cada etiqueta y atributo. Esto les permite utilizar incluso las etiquetas que no están actualmente soportadas por el editor visual. En cualquier momento, puede cambiar de un lado a otro, para utilizar la vista que mejor se adapte a sus necesidades de edición actual. Una vez que se haya completado la edición, inmediatamente obtendrá el código HTML para ser publicado en Internet.

Para asegurar un rendimiento máximo, el editor inicialmente genera el código mínimo necesario para que funcione en el cliente. Cuando los usuarios finales necesitan elementos adicionales, tales como ventanas emergentes para la gestión de imagen o hipervínculo, estos elementos se cargan a través de devoluciones de llamada AJAX. Devoluciones de llamada también se utilizan, por defecto, para todas las demás acciones que requieren la comunicación con el servidor.

### 2.5.6. ASPxPivotGrid

La suite ASPxPivotGrid es la biblioteca de componentes primero de su tipo para ASP.NET que permite a los usuarios finales examinar minuciosamente los datos de una manera similar a la encontrada en MS Excel®. Tiene el poder de convertir los reportes que habitualmente eran difíciles de leer y difíciles de diseccionar la información en una base de datos compacta y que resume los informes visuales, hace sus aplicaciones mucho más útiles y más eficaces para indicar a sus clientes en tiempo real las necesidades de análisis de negocio. Y lo más importante, con ASPxPivotGrid, los usuarios pueden cambiar el diseño de un informe visual con el fin de examinar los datos desde diferentes perspectivas / múltiple.

#### Controlar el nivel de detalle

Por defecto, los ejes de fila y columna contienen cada valor único en la columna de origen. Con ASPxPivotGrid, usted puede simplificar un informe mediante la fusión de los valores. Por ejemplo, usted podría tener un informe con datos de la encuesta en la que la edad del encuestado representa una característica importante. Los usuarios finales pueden no querer navegar a través de todas y cada una edad única y en su lugar, quieren ver las edades de los encuestados en intervalos de 31-40 como 21-30, etc. En lugar de ver las ventas por cada fecha única, es posible que quieran para ver las ventas sólo para

---

<sup>22</sup> **WYSIWYG** es el acrónimo de *What You See Is What You Get* (en español, "lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final.

los primeros meses. El eje puede personalizar en casi cualquier moda y las personalizaciones más comunes sólo necesitan una única propiedad que modificarse para la aplicación. (DevExpress, 2009)

### **Organizar jerárquicamente Valores - Ver datos brutos o se resumen en un único informe**

Los usuarios finales pueden querer analizar las ventas para cada producto o categoría de productos. Es posible que desee ver los datos anuales, o verlo con más detalle mediante la visualización de los datos para cada mes del año. Con los informes tradicionales / herramientas de bases de datos, a menudo se ven obligados a crear una gran variedad de diferentes informes con el fin de ofrecer la solución adecuada a sus clientes, pero eso ya no es el caso con ASPxPivotGrid. Puesto que el control visual ofrece a sus usuarios finales una flexibilidad total personalización en tiempo de ejecución, no un diseño de informe único que debe ser modificado para ver los datos en diferentes niveles de detalle. Sólo tiene que colocar los campos en los niveles de detalle inferiores en el área de la columna o fila. Los valores de campo se organizan automáticamente en una estructura de árbol con los padres expandibles. Los usuarios finales son libres de expandir y contraer los grupos secundarios para ver.

### **Totales automáticos y especificar manualmente**

Para mejorar la experiencia de desarrollo, ASPxPivotGrid tiene la capacidad de ofrecer contenidos superiores con el mínimo de código. Entre las muchas características disponibles en la biblioteca de componentes es la capacidad de mostrar los totales generales de cada fila y columna de forma automática. Puede mostrar cualquier número de los totales para cualquier grupo de valores.

### **Ordenar datos y mostrar filas principales - Fácil acceso a la información más importante**

ASPxPivotGrid ordena automáticamente los valores de fila y columna de esta manera puede localizar los datos requeridos y realizar un análisis integral de datos, sin embargo, a menudo se requiere un enfoque de dos vías para los datos de navegación en el que usted puede necesitar encontrar los datos por un valor específico o encontrar un valor por sus datos. Para tener en cuenta este último enfoque, ASPxPivotGrid presenta una característica única, los valores de clasificación por parte de sus grandes totales correspondientes. Esto significa que el producto que va a ver primero no es el que se inicia con una A o Z, sino más bien el que tiene al menos la mayoría de las ventas, por supuesto, esto es sólo el principio para el análisis de datos más eficaz y eficiente. También puede especificar el número de filas que se muestran arriba. Así, se puede ver un número determinado de productos vendidos o el peor de venta para cada año, el mes, cliente, etc. Usted puede concentrarse únicamente en la información más importante y evitar ser sobrecargado con una gran cantidad de datos excedentes.

### **Filtrar datos - Filtrar los valores utilizados para calcular resúmenes sobre la marcha**

ASPxPivotGrid no tiene rival en su conjunto de características y capacidades para .NET. El objetivo era simplificar radicalmente el análisis de datos en tiempo de ejecución y proporcionar al consumidor final con opciones de personalización superiores. Encontrará que con su opción de construcción visual de filtrado, los usuarios finales pueden manipular los informes sobre la marcha.

### **Ocultar valores innecesarios en los ejes.**

Otro mecanismo que permite a los usuarios centrarse únicamente en los datos más importantes.

## **Filtrar los datos.**

Para permitir a los usuarios finales gestionar esta capacidad visual, ASPxPivotGrid tiene una Región Panel Filter. Este panel fue diseñado específicamente para contener campos que no se utilizan actualmente en el informe, y cuando sea necesario, se pueden soltar sobre el informe de modificar automáticamente los datos de base utilizados en los cálculos.

### **2.5.7. ASPxScheduler**

Construido y optimizado para ASP.NET 2. La suite incluye todas las herramientas que necesitas para crear impresionantes aplicaciones web que imitan las capacidades que se encuentran en Microsoft Outlook.

La suite ASPxScheduler le ofrece cinco tipos de vista diferentes, pantallas de múltiples recursos, el apoyo de recurrencia, una navegante fecha avanzada, una función de editor de cuadros de diálogo y menús desplegados. Todas estas características se han construido en la cima de nuestro núcleo de biblioteca ASP.NET, y proporcionar la representación ligera y de alta capacidad de respuesta a través de devoluciones de llamada AJAX<sup>23</sup>.

La suite ASPxScheduler está compuesta por varios componentes, incluyendo el Planificador y el Explorador de fechas, junto con los editores de datos más pequeños utilizados en los cuadros de diálogo integrados.

#### **Built-In Date Views (5 en 1)**

Vista del día: da la visión más detallada de las citas para un día en particular.

Vista Semana laboral: Muestra las citas para los días de trabajo en una semana determinada.

Vista de la semana: Muestra las citas de una semana determinada.

Vista del mes (Multi-Week view): La vista de mes es el menos detallado de los puntos de vista, y está diseñado para permitir a los usuarios finales para buscar y analizar planes a largo plazo.

Cronología vista: Muestra citas como las barras horizontales a lo largo de la escala de tiempo, y proporciona a los usuarios finales con una visión más clara para fines de programación.

#### **Date Navigator control**

Botón Hoy: El botón Today (que puede ser opcionalmente oculta) salta hasta hoy en el planificador. (DevExpress, 2009)

Números de semana: Ocultar o mostrar los números de la semana.

Negrita fechas con citas: Opcional audaz margen para las fechas que tienen citas asignadas.

---

<sup>23</sup> **AJAX**: acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente.

Selección múltiple: Posibilidad de seleccionar varias fechas (días, semanas, etc) para mostrarlos juntos en el planificador.

Destacar vacaciones: Todos los días de fiesta están automáticamente pintado en rojo.

### **Capacidades de citas**

Representación de estado: Cuatro diferentes opciones de visualización de tiempo para la vista del día (incluso para citas que se repitan durante varios días).

Representación Label: Once tipos de citas para colorear para indicar el tipo de una cita / importancia.

Botones para citas ocultas: Botones que indican que los nombramientos adicionales existen, pero no encajan en la región de la pantalla. Al hacer clic en este botón se cambia la vista para la fecha correspondiente. Estos botones se denominan botones "más" y se puede hacer clic para ir a la cita siguiente / anterior.

Recurrencia: Las citas periódicas son una parte importante de cualquier aplicación de programación, y muchas citas periódicas se pueden crear a partir de un patrón de frecuencia única.

Recordatorios: Una cita puede contener un recordatorio de que es responsable de enviar alertas en períodos específicos de tiempo antes de la hora de inicio de una cita.

Mostrar la hora como reloj: Tiempos Mostrar inicio de la cita y al final como relojes digitales o analógicos.

De inicio y finalización de supresión tiempo: supresión opcional de las horas de inicio y fin de las citas.

Citas y con los recursos no asignados: Nombramientos pueden ser asignados a los recursos o no.

Comprobación de Conflictos: El planificador puede resolver conflictos de citas ya sea prohibiendo o permitiendo que ellos.

### **Enlace de datos**

Enlazar a una base de datos: El programador puede almacenar sus datos en una base de datos, de modo que otros controles data-aware<sup>24</sup> puede acceder a sus datos.

Enlace de campos personalizados a una cita: Los campos de datos por omisión utilizados para gestionar las citas no son todos los datos de una cita puede contener. Puede especificar campos tan diversos como necesite. Si especifica los campos personalizados, puede permitir que los usuarios finales puedan modificar los valores de campo a través de cuadros de diálogo personalizados, o inicializar estos campos en el código. Con este enfoque, usted no tiene que crear el almacenamiento de datos adicional para mantener los datos ampliados para citas individuales. (DevExpress, 2009)

### **Multiple Resources Display**

---

<sup>24</sup> **Controles data-aware:** Para navegación y manipulación de la bdd.

Agrupar por fecha: Las citas pueden ser agrupados por sus fechas.

Grupo de recursos: Las citas pueden ser agrupados por sus recursos.

Recursos imágenes: Es capaz de unirse en cualquier imagen para un recurso, por lo que se muestra en su cabecera.

Recursos de filtrado: Es posible filtrar los datos por los recursos.

Desplazamiento de Recursos: Si el número de recursos que aparece en pantalla al mismo tiempo dentro del control del planificador hace que sea difícil de ver todos los datos programados, se puede limitar el número de recursos y, a continuación, desplácese entre ellos según sea necesario. Para permitir que los usuarios finales puedan desplazarse entre los recursos, hay un control Resource Navigator que está incrustado en el Scheduler Control, y se muestra si es necesario.

Compartir recursos: La misma cita puede ser asignado a varios recursos. Esta cita se mostrará en todos los plazos correspondientes a los recursos vinculados.

### **End-User Capabilities**

Cambiar el tamaño y citas en movimiento: El programador proporciona una herramienta extremadamente fácil de usar mecanismo de nombramientos arreglo, con soporte completo para redimensionar y citas móviles.

Construido en los menús emergentes: El programador implementa una serie de menús integrados contexto similar a los encontrados en Microsoft Outlook. Todos estos menús están controlados por las propiedades del planificador, que le permiten deshabilitar elementos particulares o sustituir un menú con los tuyos. Tenga en cuenta que también puede personalizar estos menús según sea necesario.

Construido en los cuadros de diálogo: Nuestro planificador de control ofrece una interfaz de usuario completa en su aplicación, ya que hemos integrado todos los cuadros de diálogo que necesitan los usuarios finales para navegar y editar sus planes.

### **Personalización de la interfaz de usuario**

Localización: El programador muestra dos tipos de cadenas en sus elementos, cuadros de diálogo y menús, algunas cadenas se leen de la configuración regional, por lo que nunca tendrá que preocuparse acerca de su localización, y el resto puede ser fácilmente localizado.

Vacaciones del Estado: ASPxScheduler es capaz de mostrar los días festivos estatales. Puede exportar listas vacaciones ya sea desde Microsoft Outlook o desde un archivo XML personalizado con una sintaxis muy simple. Vacaciones especificadas se indican visualmente tanto en el navegador de fechas y el calendario. El navegador de fechas muestra las fechas de vacaciones con un color diferente. En un calendario, las vacaciones no son una indicación de la hora de trabajo (al igual que los fines de semana).

Restricciones de usuario final: Puede prohibir que los usuarios finales crear, mover, cambiar el tamaño, copiar, etc., las citas programadas.



Limite el intervalo de tiempo accesible: El planificador le permite establecer el intervalo de tiempo permitido. Si se configura, los usuarios finales no pueden acceder a las fechas fuera del intervalo especificado.

Las imágenes personalizadas cita: en lugar de la norma "clásicos" de las imágenes usadas para las citas, también es posible integrar los iconos personalizados que se muestran en diferentes citas.

Personalizar el editor in situ: Se puede aplicar una forma editor personalizado y utilizar este formulario como un editor local para las citas. Esto significa que los usuarios finales verán su formulario personalizado en lugar de un editor tema simple. Ni que decir, el formulario personalizado se redimensiona automáticamente para adaptarse a la cita que se está editando.

Conjunto de controles nativos: Existe también un conjunto de editores adicionales suministrados con el paquete de ASPxScheduler, ASPxViewSelector, ASPxViewNavigator, etc. Así que usted puede crear fácilmente formularios personalizados utilizando las nativas de ASPxScheduler controles de edición.

Información sobre herramientas personalizada: Se ha añadido la capacidad de crear información sobre herramientas personalizadas, que se muestran cuando se pasa el cursor del ratón sobre una cita o una selección de tiempo celular. También puede personalizar la apariencia de la información sobre herramientas y aparece cuando se arrastra una cita. Una descripción está representada por un control de usuario web, que permite a un desarrollador ajustar el diseño y el comportamiento según se requiera. (DevExpress, 2009)

## **Informes (impresión a través de XtraReports)**

Proporciona una gran flexibilidad en la construcción de estilos de impresión para los datos XtraScheduler. Utiliza la suite XtraReports para crear un informe que contenga controles especialmente diseñados para representar entidades Scheduler. Varios controles del programador de informes y el conjunto de todas las funciones de los controles XtraReports se pueden combinar en un informe para realizar una tarea compleja. La presentación de informes para el usuario final en formularios Windows Forms proporciona a los usuarios finales la capacidad de editar informes, cargar y guardar plantillas de informes, y la vista previa y la exportación a varios formatos (incluyendo PDF y BMP).

### **2.5.8. ASPxSpellChecker**

El **ASPxSpellChecker** aprovecha al máximo el marco ASP.NET y viene con código fuente completo. Sus características incluyen:

- El soporte integrado para ISpell <sup>25</sup>, OpenOffice y Hunspell <sup>26</sup> diccionarios.

---

<sup>25</sup> **ISpell**: correctores de ortografía en paquetes ofimáticos y en programas de correo.

<sup>26</sup> **Hunspell**: es un corrector ortográfico y un analizador morfológico diseñado para idiomas con una morfología rica, compleja formación de palabras compuestas o con una codificación de caracteres distinta del ASCII de 8 bits, diseñado en un principio para el idioma húngaro.

- Callbacks AJAX se utiliza para las operaciones que requieren la comunicación con el servidor. Estas operaciones incluyen mostrar un cuadro de diálogo con sugerencias de ortografía, y actualizar un diccionario con palabras personalizadas.
- Formato de texto plano diccionario personalizado de apoyo.
- Diccionario de diálogos permiten a los usuarios agregar palabras desconocidas en el diccionario. Como era de esperar de este tipo de control, los usuarios finales pueden crear una lista de palabras personalizada según sus necesidades lo exijan.
- Pre-built Opciones editor.
- Opciones del corrector le permiten ignorar correos electrónicos, direcciones URL, texto dentro de las etiquetas (entre paréntesis angulares y cuadrados), mezcla de casos / palabras en mayúscula y las palabras con números en su interior.
- El **ASPxSpellChecker** 's API incluye métodos para comprobar los contenidos del elemento o el contenido de texto de cada editor ubicado en un elemento contenedor.
- Usted puede controlar los eventos especialmente diseñados para gestionar por completo el proceso de corrección.

### 2.5.9. ASPxTreeList

#### Datos Específicos

Enlazar los datos: El ASPxTreeList puede ser fácilmente llenado con datos por la unión a cualquier tipo de fuente de datos estándar, incluyendo: SqlDataSource, ObjectDataSource, XmlDataSource, AccessDataSource, SiteMapDataSource. También puede enlazar el ASPxTreeList a cualquier objeto que representa una estructura de árbol, o crear manualmente un árbol en el código.

Virtual Mode: La ASPxTreeList puede funcionar en modo virtual. Este modo puede ser útil al enlazar con grandes conjuntos de datos. En el modo virtual, un árbol se crea en la demanda. En este caso, los nodos secundarios son creados e inicializados cuando su nodo padre está expandido.

#### Formateo de datos y manipulación

Resúmenes de datos completos: El ASPxTreeList permite calcular valores de resumen de todos los valores en una columna o en un grupo de nodos, y mostrar el resultado en el interior de las células pie de página. El ASPxTreeList proporciona cinco funciones integradas agregados (count, max, min, etc) y los eventos personalizados para implementar manualmente funciones de agregado. (DevExpress, 2009)

Previa: Con filas de vista previa, puedes dejar que tus usuarios finales ver texto grande o campos Memo directamente en la ASPxTreeList, sin la necesidad de abrir un editor externo o una ventana desplegable.

Columna de personalización en tiempo de ejecución: Los usuarios finales pueden gestionar el diseño de columnas de la lista de árboles por reorganizar, agregar y quitar columnas, usando arrastrar y soltar.

Selección de varios nodos: El ASPxTreeList permite la selección de nodo único y múltiple.

#### El enfoque y la navegación

Built-in Pager: El ASPxTreeList automáticamente puede dividir el contenido en varias páginas, y proporcionar el paginador incorporado que permite a los usuarios navegar a través de los datos.

Iterator Nodes: Le permite evitar escribir código recursivo para recorrer todos los nodos contenidos en el ASPxTreeList. Se hace atravesar a través de los nodos de una tarea trivial.

### **Apariencia personalizada**

Apariciones: un mecanismo poderoso para controlar el aspecto y la sensación de la lista de árbol. Se puede personalizar la apariencia de casi cualquier elemento visual de la ASPxTreeList, e incluso aplicar condicionalmente para reflejar ciertos estados, tales como errores. Además, los barcos ASPxTreeList con varios temas predefinidos.

Plantillas Tecnología: Plantillas son totalmente compatibles. Para cada elemento dentro de la ASPxTreeList, usted puede personalizar completamente la apariencia o diseño utilizando HTML o cualquier control de servidor.

Soporte completo de CSS: La aparición del ASPxTreeList se puede personalizar mediante clases CSS.

### **Misceláneo**

Actualización parcial: Esta característica reduce el tiempo de carga de página. Sólo una parte del control ASPxTreeList se actualiza, sin una actualización de cliente de pantalla completa.

Compatibilidad entre navegadores: Internet Explorer, Mozilla Firefox, Google Chrome, Opera, Apple Safari, etc. El control ASPxTreeList es cross-browser compatible.

Rich funcionalidad del lado cliente: API integral del lado del cliente está disponible para la ASPxTreeList.

Los eventos del cliente: La ASPxTreeList expone un conjunto rico de eventos del cliente. Estos eventos se ejecutan en el lado del cliente, y se han diseñado para maximizar el rendimiento.

Cookies de Apoyo: El control ASPxTreeList es totalmente compatible con las cookies, permitiendo a los visitantes de su sitio web para personalizar páginas según sea necesario. Si las cookies están habilitadas, cada posible usuario final guarda opción de personalización, y luego restaura en futuras sesiones.

Guardar y restaurar Layout: El ASPxTreeList le permite guardar la información en su diseño a una base de datos, y luego restaurarlo.

AJAX apoyo tecnológico.

### **2.5.10.XtraCharts**

- Soporte para múltiples plataformas de destino: XtraCharts Suite incluye el ChartControl (para Windows Forms) y WebChartControl (para ASP.NET). Ambos ofrecen una funcionalidad similar, mientras que ofrece todas las ventajas de sus plataformas específicas. (DevExpress, 2009)
- Profundo apoyo para los productos DevExpress: PivotGridControl ( ASPxPivotGrid ) y XtraReports .

- Soporte para múltiples tipos de gráficos diferentes: varios tipos de gráficos 2D y 3D desde lo básico (por ejemplo, barras y columnas, Punto y línea, Pie y re-direccionamiento, o Radar y Polar) a lo específico (por ejemplo, Burbuja , Gantt o Parcela Swift que es especialmente diseñado para su procesamiento inmediato de cantidades muy grandes de puntos de datos).
- Gráficos Financieros: como los indicadores técnicos (como las líneas de regresión y Trend, indicadores de Fibonacci y media móvil).
- Las funciones inteligentes y automatización: etiquetas inteligentes de los puntos de la serie, auto-ajustes a orígenes de datos conocidos (como Pivot Grid), saltos automáticos escala, la escala del eje de desplazamiento automático y ajustes de gama, así como fecha y hora automáticas modos para el eje de argumentos.
- Representación de varias series de diversos tipos: compatibilidad de las diferentes series en el mismo diagrama, soporte para un número ilimitado de paneles y ejes secundarios, con quiebres de escala logarítmica y soporte escala.
- Ampliado las capacidades explicativas: acompañan con una leyenda flexible y soporte para un número ilimitado de títulos de gráfico, así como las dos anotaciones de texto e imagen, con múltiples opciones para la mejor adaptación de su diseño.

### 2.5.11.XtraReports

La suite XtraReports es de los componentes más poderosos para la plataforma .NET para crear un sistema de información integrado, potente y flexible. Brinda una facilidad con la que incluso las aplicaciones más complejas de informes se pueden crear sin dificultad.

A continuación se muestra una breve lista de las principales características y beneficios disponibles en la Suite XtraReports.

- Multi-Platform Solution
- Arquitectura Flexible
- Amplio conjunto de controles
- Amplio conjunto de proveedores de datos
- Datos Extendidos Shaping<sup>27</sup>
- Tiempo de diseño amistoso
- Amplio conjunto de formatos de exportación
- Vista previa de impresión versátil
- Información para el usuario final
- Soporte profunda para ASP.NET
- En profundidad Apariencia y Look-and-Feel personalización
- La localización completa
- Fácil migración

---

<sup>27</sup> **Shaping:** Intenta controlar el tráfico en redes de ordenadores para así lograr optimizar o garantizar el rendimiento, baja latencia, y/o un ancho de banda determinado retrasando paquetes.

## 2.6. Sybase

### 2.6.1. Características

Principales características de Ase:

- Creación de servidores mediante dataserver
- La utilidad isql
- bcp para transferir datos hacia y desde Adaptive Server
- dsedit
- dscp
- Referencia de comandos de utilidades

### 2.6.2. Creación de servidores mediante dataserver

El comando dataserver permite crear dispositivos master y bases de datos con páginas lógicas de un tamaño de 2 K, 4 K, 8 K ó 16 K. Cuanto mayores sean las páginas lógicas, mayores filas podrá crear, mejorando así el rendimiento, ya que Adaptive Server accede a más datos cada vez que lee una página. Por ejemplo, una página de 16 K puede albergar ocho veces más datos que una página de 2 K, una página de 8 K alberga cuatro veces más datos que una página de 2 K, y así sucesivamente con todos los tamaños de página lógica.

Dado que el tamaño de página lógica es un parámetro de configuración para todo el servidor, no podrá tener en el mismo servidor bases de datos con páginas lógicas de diferentes tamaños. Todas las tablas se ajustan adecuadamente para que el tamaño de la fila no exceda el tamaño actual de página del servidor. Es decir, las filas no pueden ocupar más de una página.

#### La utilidad isql

isql envía comandos Transact-SQL a Adaptive Server, da formato a los resultados y los imprime por la salida estándar. No existe un tamaño máximo para las instrucciones de isql. (Sybase)

Para utilizar Transact-SQL directamente desde el sistema operativo con el programa de utilidades isql, será necesario poseer una cuenta o un identificador de inicio de sesión en Adaptive Server

### 2.6.3. Bcp para transferir datos hacia y desde Adaptive Server

bcp es la utilidad de copia masiva para transferir datos entre Adaptive Server y un archivo del sistema operativo.

bcp puede funcionar en dos modos:

- Versión rápida de bcp: registra cada inserción de fila que realiza, y se utiliza para tablas que tienen uno o más índices o triggers.
- Versión lenta de bcp: sólo registra asignaciones de página, y copia datos en tablas sin índices ni triggers a la mayor velocidad posible.

Para determinar el mejor modo bcp para realizar el proceso de copia hay que tener en cuenta lo siguiente:

- Tamaño de la tabla en la que desea copiar datos
- Cantidad de datos que desea copiar
- Número de índices de la tabla
- Cantidad de espacio disponible en el dispositivo de base de datos para volver a crear índices

### **dsedit**

dsedit es una utilidad gráfica que permite ver y editar las entradas de servidor del archivo de interfaces (sql.ini en Windows 98 y Windows NT).

### **dscp**

dscp es una utilidad de UNIX que sirve para ver y editar entradas de servidor en el archivo de interfaz.

## **2.6.4. Referencia de comandos de utilidades**

En UNIX, los comandos de los programas de las utilidades se introducen en el indicador del sistema en el núcleo de UNIX.

En Windows NT:

- Si la utilidad tiene un icono en el grupo de programas de Sybase para Windows o de Sybase para Windows NT, haga doble-clic en el icono para ejecutar el programa.
- Si la utilidad no tiene un icono en el grupo de programas, introduzca el comando del programa de la utilidad en el indicador del sistema de Windows o Windows NT para ejecutar el programa.

Coloque entre comillas los caracteres con especial significado para el núcleo (el indicador del sistema en Windows NT), como la barra invertida (\), el asterisco (\*), la barra (/) y los espacios. Se puede colocar una barra invertida delante de algunos caracteres especiales (\) para anularlos. Esto evita que el núcleo (el indicador del sistema) interprete los caracteres especiales.

## **2.6.5. Programas de utilidades**

Backupserver: Forma ejecutable del programa Backup Server™. (Sybase)

bcp: Copia las filas de una tabla de base de datos a o desde un archivo del sistema operativo con el formato-especificado por el usuario.

certauth: Convierte una solicitud de certificado-servidor en un certificado-firmado de CA (certificateauthority).

certpk12: Exporta o importa un archivo PKCS#12<sup>28</sup>.

certreq: Crea una solicitud de certificado y la correspondiente clave privada de dos formas:

charset: Carga los archivos del juego de caracteres y del criterio de ordenación.

cobpre: Precompilador de COBOL<sup>29</sup>.

cpre: Precompilador de C.

dataserver: Forma ejecutable del programa Adaptive Server.

dataxtr: Herramienta de migración de datos.

ddlgen: Genera un lenguaje de definición de datos para los objetos a nivel-servidor y a nivel-base de datos en ASE.

defncopy: Copia las definiciones de las vistas, reglas, valores predeterminados, triggers, procedimientos o informes especificados desde una base de datos a un archivo del sistema operativo o viceversa.

dscp: Permite ver y editar las entradas del servidor en el archivo de interfaces en modo de línea de comandos.

dsedit: Permite ver y editar las entradas del servidor en el archivo de interfaces utilizando una interfaz gráfica de usuario basada en X11/Motif.<sup>30</sup>

extractjava: Copia un JAR<sup>31</sup> retenido desde Adaptive Server a un archivo de cliente.

installjava: Instala un JAR desde un archivo de cliente a Adaptive Server.

isql: Analizador SQL interactivo de Adaptive Server.

langinstall: Instala un nuevo idioma en Adaptive Server.

optdiag: Muestra las estadísticas del optimizador o carga estadísticas actualizadas en las tablas del sistema.

pwdcrypt: Crea e imprime una contraseña LDAP encriptada en el archivo libtcl.cfg.

showserver Muestra los Adaptive Servers y Backup Servers actualmente en ejecución en el equipo local.

sqldbg: Depura procedimientos y triggers almacenados.

---

<sup>28</sup> **PKCS:** En criptografía, PKCS (Public-Key Cryptography Standards) se refiere a un grupo de estándares de criptografía de clave pública concebidos y publicados por los laboratorios de RSA.

<sup>29</sup> **COBOL:** El lenguaje COBOL (acrónimo de COmmon Business-Oriented Language, Lenguaje Común Orientado a Negocios) fue creado en el año 1959 con el objetivo de crear un lenguaje de programación universal que pudiera ser usado en cualquier ordenador.

<sup>30</sup> **Motif:** es una biblioteca para la creación de entornos gráficos bajo X Window System en sistemas Unix.

<sup>31</sup> **JAR:** el comando jar permite generar, observar y descomprimir archivos JAR.

sqlloc: Instala y modifica los idiomas, juegos de caracteres y criterios de ordenación predeterminados de Adaptive Server en modo GUI.

sqllocres: Instala y modifica los idiomas, juegos de caracteres y criterios de ordenación predeterminados de Adaptive Server en modo línea-comandos.

sqlsrvr: Forma ejecutable del programa Adaptive Server.

sqlupgrade: Actualiza la versión de Adaptive Server instalada actualmente con la versión más reciente en modo GUI.

sqlupgraders: Actualiza la versión de Adaptive Server instalada actualmente con la versión más reciente en modo línea-comandos.

srvbuild: Crea un nuevo Adaptive Server, Backup Server, Monitor Server o XP Server en modo GUI con los valores predeterminados o especificados por el-usuario para los atributos de configuración de clave.

srvbuildres: Crea un nuevo Adaptive Server, Backup Server, Monitor Server o XP Server en modo línea-comandos con los valores predeterminados o especificados por el-usuario para los atributos de configuración de la clave. (Sybase)

sqlsrvr: Forma ejecutable del programa Adaptive Server.

startserver: Inicia un Adaptive Server o un Backup Server.

srvmgr: Inicia Adaptive Server y Backup Server como servicios de Windows NT.

sybload: Carga los productos Sybase desde un soporte de distribución y crea el directorio de instalación de Sybase desde la línea de comandos.

sybsetup: Instala y configura Adaptive Server desde una única ubicación mediante una interfaz GUI.

xpsrvr: Inicia XP Server manualmente.

## **2.7. SQL dbx y Aqua Data Studio 9.0**

Son editores de bases de datos SQL, se ha escogido estas herramientas porque proporcionan las utilidades necesarias para trabajar con sybase.

### **2.7.1. SQL dbx**

SqlDbx es un editor de bases de datos SQL que permite explorar estructuras, realizar consultas y guardar cambios de un modo remoto.

Es muy sencillo de utilizar. Basta con conectar con un servidor SQL para que, al instante, el programa muestre en pantalla la estructura de la base de datos. Sus funciones permiten hacer consultas, cambiar estructuras y modificar registros.



SqlDbx es compatible con la mayoría de servidores SQL disponibles en la red: Microsoft SQL Server, Sybase ASE Server, Oracle Server, IBM DB2/UDB Server y ODBC.

La edición personal de SqlDbx es una versión útil y libre de la edición profesional para el uso no comercial personal. La diferencia de la edición profesional es permitir conectar con solamente un servidor de base de datos a la vez, si no ambas son versiones idénticas.

SqlDbx se construyó alrededor de un explorador avanzado del objeto del redactor y de base de datos del SQL. SqlDbx proporciona un interfaz utilizador constante entre diversos sistemas del DBMS. El interfaz intuitivo y directo permite que los reveladores mejoren su productividad teniendo de fácil acceso a las características más de uso general.

Funcionar con las preguntas, ejecutar las escrituras y hojear los objetos de base de datos sin dejar la ventana del redactor. SqlDbx es un fichero ejecutable independiente, así que no hay instalación necesaria. Además, SqlDbx no instala ni modifica cualquier cosa en la computadora del usuario.

Aquí están algunas características dominantes de “SqlDbx personal”:

- Ayuda nativa del alto rendimiento para los servidores de base de datos importantes incluyendo Microsoft, Sybase ASE, Oracle e IBM DB2/UDB. Hojear/explorar los servidores
- Ayuda inteligente, extenso y alto adaptable Intellisense<sup>32</sup>
- El editor del guión avanzado con el sintaxis que destaca, auto completar, velocidad mecanografiando, variables de la escritura, fricción/gota, deshace/hace de nuevo, modo de la columna (bloque). Datos visuales y archivo Diff.
- Plantillas del SQL con la ayuda para los parámetros formales
- Importación/exportación a CSV, XML, Excel. Hojeador del objeto del esquema
- Historia de la declaración de SQL
- Objetos preferidos
- Objeto del hallazgo
- Búsqueda en base de datos y en rejilla del resultado
- Corregir directo los datos de la tabla en rejilla del resultado
- La clase, filtro, formato, exportación y transporta datos en rejilla del resultado
- Ejecutar los procedimientos y las funciones con la especificación automática de la discusión
- Exhibir el plan de la pregunta y las estadísticas de la pregunta
- Visiones administrativas
- Despliegue y retiro simples
- Ningunos DLL u otras dependencias
- Apenas uno ejecutable (MB 1.5)
- Ayudas Oracle, IBM DB2 UDB/LUW/z/OS/iSeries/AS/400, Sybase ASE/índice de inteligencia/dondequiera, servidor del SQL, ODBC
- Versión nativa x64 de SqlDbx
- La columna multi agregada resulta clase de la rejilla (el tecleo del Ctrl + del doble)
- Ayuda mejorada para los hogares múltiples de Oracle

---

<sup>32</sup> **Intellisense:** Es una aplicación de autocompletar.

- Exportación mejorada de la tabla al archivo
- Nueva característica mejorada de la fila del parte movable. Una mejor dirección de la identidad y de las columnas del reloj fechador
- El scripting fijo de defectos y de apremios del cheque para Sybase y el servidor de Microsoft SQL
- Ediciones fijas de la pintura en Windows 7
- DB2 contrarios fijos conectan comportamiento
- Pequeños mejoras y realces en diversas áreas de los pares

### **2.7.2. Aqua Data Studio 9.0**

Ésta utilidad fue selecciona porque tiene la herramienta para depurar las funciones o procedimientos almacenados, a continuación se describen las principales características.

#### **Ventana de Analizador de Búsqueda**

Aqua Data Studio es un sofisticado analizador de búsqueda que permite a los usuarios trabajar con consultas de base de datos con RDBMS específico, sintaxis enfocada y auto finalización para desarrollar y probar consultas SQL. Otras características que pueden darle rapidez al proceso de desarrollo incluyen SQL automatización, auto-describir y resultados clasificados por rejilla-múltiple. El analizador de búsqueda también proporciona al lado del cliente la habilidad de hacer encuadernación variable, dejando a un usuario Cumplir con procedimientos o funciones guardadas con variables locales y encuadernados a recibir parámetros afuera. La ventana de búsqueda también proporciona dos módulos de operación, usando una vista de ventana partida o de pestaña múltiple para el redactor y los resultados de búsqueda. (Wikipedia)

#### **Buscador de Esquema Y Redactor Visual**

Su capacidad de examinar visualmente permite entender la estructura y dependencias del esquema de la base de datos con un toque del ratón. El examinador de esquema también deja editar visualmente cualquier objeto de esquema con una forma de diseño gráfico a CREAR, ALTERAR o BOTAR al objeto. Editando visualmente sostiene Tablas, Índices, Procedimientos, Tipos de Data y otros objetos del esquema. El editor visual también proporciona una vista preliminar de SQL de todas las órdenes ejecutables para cometer la operación.

#### **Extracción de Esquema y Escritura de DDL**

Su habilidad de examinar gráficamente te deja extraer la definición de la estructura de todos los objetos del esquema y a escribir el SQL DDL y el DML (ejemplo: CREAR, ALTERAR, ELIMINAR, SELECCIONAR, INSERTAR, ACTUALIZAR O BORRAR) sintaxis para los objetos de los esquemas de las bases de datos (incluyendo Rejillas, Vistas, Gatillos, Procedimientos Guardados y Funciones).

#### **Redactor de Tabla de Data**

Un poderoso Redactor de Tabla de Datos te da la habilidad de modificar y guardar los resultados gráficamente. Simplemente escribiendo una declaración de tabla SELECT en el Analizador de Búsqueda y oprimiendo el botón de "Ejecutar Corrección," se abre una ventana separada con el resultado, que

puedes revisar y guardar. También puedes examinar en una tabla del buscador de esquema y seleccionar la Tabla de Editar en la barra de menú para editar los primeros mil archivos.

### **Plano de Explicación Visual**

El analizador de búsqueda deja a los usuarios generar planos de explicación visuales con o sin ejecutar las declaraciones de SQL. Estos planos pueden ser visualizados en tabla de árbol o en diagrama. También se le puede activar la habilidad de ejecutar cada vez que declaraciones son ejecutadas, o se pueden ejecutar individualmente con el mando "Ejecutar Explicación" en la ventana de búsqueda. Corrientemente apoya a Oracle, DB2 y Microsoft SQL Server, proporcionando una interfaz común y fácil de usar para las tres bases de datos.

### **Editando Visualmente para Almacenar**

Permite CREAR, ALTERAR, CAMBIAR TAMAÑO y ELIMINAR aparatos de almacenar para cada base de datos. El buscador de esquema te deja navegar por la estructura del almacén, alterar visualmente cualquiera de los componentes de los objetos del almacén y cometer los cambios. El editor visual también te proporciona una vista de antemano del SQL que se va a usar para cometer los cambios. De los objetos que se apoyan se incluyen Bases de datos, Espacios de Tabla, Archivo de Data, Segmentos de Regresión, Registro de Rehacer, Contenedores y Aparatos de Descarga. (Wikipedia)

### **Editando Visualmente Objetos de Seguridad**

Su habilidad para Editar Visualmente te deja CREAR, PERMITIR, REVOCAR Y ELIMINAR objetos de seguridad y permisos para cada base de datos. El buscador de esquema te deja navegar la estructura del almacén, alterar visualmente cualquiera de los componentes de los objetos de seguridad y cometer los cambios. El editor visual también te proporciona una vista de antemano del SQL que se va a usar para ejecutar los cambios. De los objetos que se apoyan se incluyen Usuarios, Papeles, Retrato, Iniciación de sesión, Papeles de Servidor y Permisos de Objetos/Columnas.

### **Extracción de Objeto y Escritura de DBA**

Su habilidad de examinar gráficamente te deja examinar entre todos los objetos de administración de bases de datos y sacar las definiciones para escribir el SQL DDL sintaxis para creando, alterando y botando los objetos de almacenar y seguridad (incluyendo bases de datos, espacios de tabla, archivo de data, contenedores, usuarios, grupos/papeles y retrato).

### **Herramienta de Importar y Exportar**

Su herramienta de importación deja importar cualquier archivo de texto demarcado a un archivo existente o nuevo. El ayudante de importación te proporciona una interfaz fácil de usar que guía a través del proceso de importación y te proporciona una vista preliminar de la información que va ser importada antes de llevarla a cabo. La herramienta de exportación deja que se exporte el contenido de cualquier tabla o el resultado de cualquier declaración de SELECT. La información exportada puede ser salvada en un archivo de texto demarcado o un archivo con declaraciones de INSERTAR. Aqua Data Studio también tiene una barra de menú de "Herramientas" que es sensitiva al contexto en el buscador de

esquema. Esto deja a usuarios examinar el esquema de la Tabla que va ser exportada y después comenzar la herramienta. La Herramienta va a usar la conexión del servidor corriente y la selección actual de base de datos/esquema/tabla.

## **2.8. Business Intelligence**

Bajo el nombre de Business Intelligence (Inteligencia de Negocios) se cobijan diferentes acrónimos, herramientas y disciplinas que apuntan a dar soporte a la tarea de toma de decisiones.

**Business Intelligence (BI)** es una disciplina que, junto con sus correspondientes herramientas, hacen centro en el análisis de la información para la correcta toma de decisiones que le permita a la organización cumplir con los objetivos de negocio.

Fundamentalmente podemos nombrar a:

### **2.8.1. Introducción a BI**

#### **2.8.1.1. Data Warehousing.**

Los Data Warehouses o Almacenes de Datos se basan en estructuras multidimensionales (cubos) en las que se almacena la información calculando previamente todas las combinaciones de todos los niveles de todas las aperturas de análisis. Es, por decirlo llanamente, un producto cartesiano que almacena todas las combinaciones. Se puede decir que este método es exagerado o salvaje y en parte esta afirmación es real. Lo que no debe perderse de vista es que esta “salvajada” es el precio que paga la organización para poder tomar decisiones correctas. Siempre va a ser más barato el gasto que conlleva la adquisición de SW o HW que el costo que representa una decisión tomada a destiempo. (Microsoft, 2009)

#### **2.8.1.2. Data Mart.**

El almacén de datos de un hecho en particular se denomina Data Mart (DM).

#### **2.8.1.3. Data Mining.**

Está asociado al escalón más alto de la pirámide (Nivel Estratégico) y tiene por objeto eliminar los errores cometidos por las personas al analizar los datos debido a prejuicios y dejar que sean los datos los que muestren los modelos subyacentes en ellos. La Minería de Datos ayuda a crear nuevos modelos no percibidos por el analista hasta ese momento pero que realmente existen en los datos.

Todas las herramientas o disciplinas que pueden contenerse en la definición de BI tienen tres características comunes:

**Primera:** Proveen información para el control del proceso de negocio, independientemente de la fuente en la que los datos se encuentran almacenados.

**Segunda:** Dan soporte a la toma de decisiones, siendo esta la característica más importante.

**Tercera:** La capa semántica. No se pueden tomar decisiones de negocio si no se habla el lenguaje propio del negocio. Independientemente del origen de los datos o de la forma de extracción, transformación y agregación, lo verdaderamente importante es que la información le debe “servir” a los usuarios finales en un lenguaje de negocios comprensible por ellos sin la necesidad de intérpretes. La idea es que el analista se concentre en la toma de decisiones, las tome con rapidez y seguridad, lo que le ofrece una ventaja competitiva a la empresa y la acerca al cumplimiento de los objetivos.

En la siguiente tabla se muestran las diferencias que son clave entre un sistema OLTP y un DW.

	<b>OLTP</b>	<b>DW</b>
Objetivos	Operacionales	Información para la toma de decisiones
Orientación	A la aplicación	Al sujeto
Vigencia de los datos	Actual	Actual + histórico
Granularidad de los datos	Detallada	Detallada + resumida
Organización	Organización normalizada	Organización estructurada en función del análisis a realizar
Cambios en los datos	Continuos	Estable

**Fuente:** (Microsoft, 2009)

**Tabla 8.** OLTP – OLAP.

A continuación se comentan cada una de las diferencias mencionadas para comprender con mayor detalle el concepto de **DW**.

**Objetivos:** Un sistema OLTP debe garantizar la consistencia de los datos, mientras que un OLAP consolida datos ya validados y los adecua a las necesidades propias de la toma de decisiones.

**Orientación:** Un sistema OLTP está orientado a la Aplicación, debe hacer cumplir las Reglas de Negocio. En cambio un sistema OLAP está orientado al Sujeto, se define en base a lo que el analista necesita ver.

**Vigencia de los Datos:** En un sistema OLTP los datos se usan en la medida que se van produciendo y dejan de ser importantes en el corto plazo (un diario de ventas se lista para el mes que finaliza y, en el mismo momento comienzan a ser importantes los datos del mes actual). En un sistema OLAP se guardan los datos actuales y los históricos para poder realizar análisis comparativos, de tendencias, etcétera. La cantidad de períodos almacenados dependerá exclusivamente de la necesidad de análisis de la empresa y de la capacidad de almacenamiento.

**Granularidad de los Datos:** En un sistema OLTP la granularidad está dada por los controles que deban realizarse, ya sea controles definidos por la organización como por las normas legales vigentes. En un OLAP estará dada por el tipo de análisis que se quieran realizar. Si el análisis del tráfico se realiza analizando el número de llamadas en el mes, no tiene sentido guardar el detalle diario en el OLAP, mientras que en el OLTP tal vez no tenga la libertad de decidir el nivel de granularidad.

**Organización:** Un sistema OLTP es normalizado, mientras que un sistema OLAP se basa en estructuras jerárquicas desnormalizadas modeladas de acuerdo a la forma en que se analizarán los datos.

**Cambios en los datos:** Un sistema OLTP modifica sus datos en forma constante porque maneja las transacciones de la empresa. Un sistema OLAP no tiene como objetivo la presentación de los datos en línea y, menos aún, pretende modificar los datos originales, solo consultarlos. La frecuencia de actualización de los datos en un sistema OLAP está definida por la granularidad.

## **2.8.2. Definiendo soluciones OLAP**

### **2.8.2.1. Sistema Transaccional (OLTP)**

#### **Características**

Los sistemas de OLTP (On-Line Transaction Processing) son los sistemas operacionales que capturan las transacciones de un negocio y las persisten en estructuras relacionales llamadas Base de Datos. (Microsoft, 2009)

Las características principales de los sistemas OLTP son:

Realizan transacciones en tiempo real del proceso de un negocio, con lo cual los datos almacenados cambian continuamente. Los sistemas OLTP en sus transacciones conducen procesos esenciales del negocio.

Los sistemas OLTP son los responsables del mantenimiento de los datos, ya sea agregando datos, realizando actualizaciones o bien eliminándolos.

Las estructuras de datos deben estar optimizadas para validar la entrada de los mismos, y rechazarlos si no cumplen con determinadas reglas de negocio.

Para la toma de decisiones, proporciona capacidades limitadas ya que no es su objetivo, por lo tanto no es prioridad en su diseño. Si se quisiera obtener determinada información histórica relativa al negocio consultando un sistema OLTP, se produciría un impacto negativo en el funcionamiento del sistema.

Normalmente, para el diseño de un sistema OLTP se define un modelo de Diagrama Entidad Relación (DER). Un DER es una representación de la realidad a través de un esquema gráfico que contiene los siguientes elementos:

**Entidades:** Una Entidad es un tipo de objeto que puede identificarse de manera única por algún medio. Este objeto es traducido a la estructura física de una base de datos como una tabla.

**Atributos:** Las características particulares que distinguen a las Entidades se denominan Atributos.

**Relaciones:** vínculos existentes entre las tablas que sirven para asegurar la integridad referencial.

Para llegar a esquematizar un DER, se debe realizar un proceso de normalización basado en las Formas Normales, lo que además garantiza una optimización del espacio de disco a utilizar.

### **Usos comunes de OLTP**

Toda organización o empresa, lleva adelante sus objetivos diarios realizando un conjunto de tareas que se encuentran cuidadosamente agrupadas dentro de procesos, estos últimos estrechamente relacionados entre sí. Los procesos pueden pertenecer al área Industrial, al departamento de Marketing, al departamento de Ventas o al sector Administrativo, mencionando solo algunos.

Decimos entonces, que en la definición de OLTP se pueden encuadrar a todos los sistemas tradicionales dedicados a la captura, validación y almacenamiento de datos de manera estructurada y que corresponden a los procedimientos. (Microsoft, 2009)

### **2.8.2.2. Sistemas OLAP**

Los sistemas OLAP (On-Line Analytical Processing) proporcionan una alternativa a los sistemas transaccionales, ofreciendo una visión de los datos orientada hacia el análisis y una rápida y flexible navegación por estos.

Las siguientes son características que la tecnología OLAP posee:

Las bases de datos de OLAP tienen un esquema que está optimizado para que las preguntas realizadas por los usuarios sean respondidas rápidamente.

Las preguntas que se le hacen a un OLAP, deben permitir un uso interactivo con los usuarios.

Los cubos de OLAP almacenan varios niveles de datos conformados por estructuras altamente optimizadas que responden a las expectativas de negocio de la empresa.

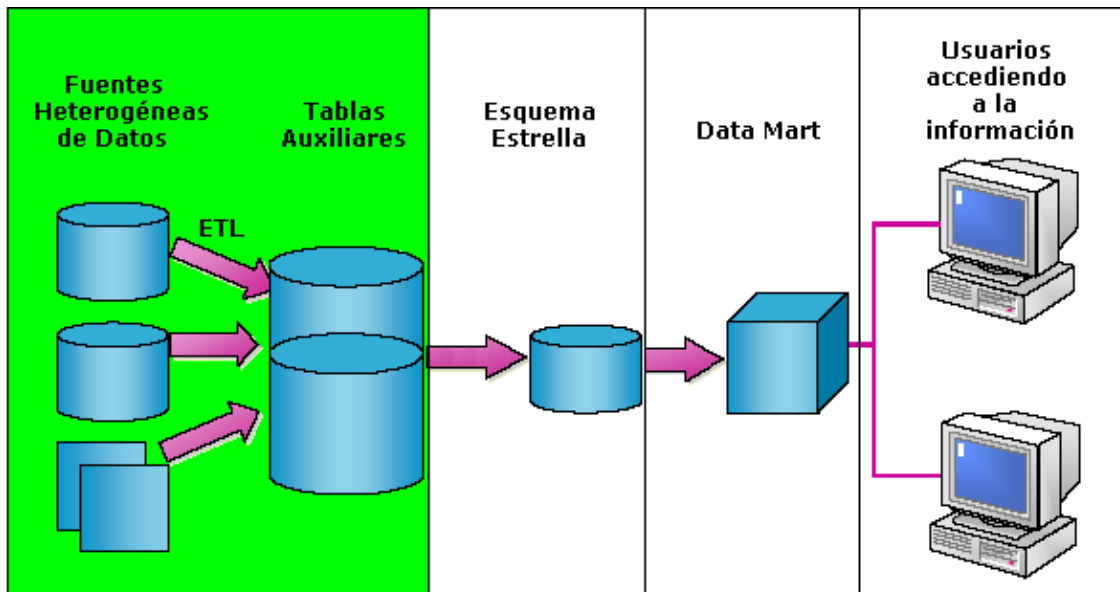
Un sistema OLAP está preparado para realizar informes complejos de una manera simple.

OLAP proporciona una vista de datos multidimensional. Los cubos proporcionan una vista de los datos multidimensional que se extiende más allá del análisis de dos dimensiones que puede proporcionar una simple planilla de cálculo utilizada como tal.

Los usuarios pueden cambiar fácilmente las filas, las columnas, y las páginas en informes de OLAP, pudiendo leer la información de la manera que se crea más conveniente para el análisis.

### **Datos de origen vs. Información del negocio**

El presente esquema representa las distintas etapas que se deben ejecutar para la construcción de un Data Mart, desde que se identifican los datos originales en los sistemas transaccionales hasta que los Usuarios pueden disponer la información. A modo de guía, se irá indicando qué parte de estos procesos cubre cada Unidad. (Microsoft, 2009)



Fuente: (Microsoft, 2009)

Figura 4. Etapa Inicial Data Mart

Las etapas que deben cubrirse durante el proceso de construcción de un DW cumplen con lo siguiente:

Identificación de las necesidades y requerimientos.

Reconocimiento de las fuentes de datos originales y sus estructuras.

En base a los requerimientos, definir las tablas auxiliares y los procesos de selección, transformación e importación de datos.

Construir el esquema multidimensional. Debe controlarse que este esquema concuerde con los requerimientos y las tablas auxiliares, como primera forma de testeo.

Acceso al sistema desde las estaciones de trabajo de los analistas obteniendo la información identificada en la etapa de requerimientos.

### 2.8.2.3. Convirtiendo datos en información

Para convertir los datos en información, se debe entender de qué manera pueden interpretarse los datos almacenados en sistemas operacionales, determinando:

Como los hechos que deseamos medir se relacionan con los datos que podemos obtener.

Entendiendo cómo estos datos reflejan metas y objetivos que abarcan el negocio involucrado.

Un DW, clasifica la información en base a los aspectos que son de interés para la empresa.

En el ambiente operacional se diseña alrededor de las aplicaciones y funciones (ventas, facturación, stock, etc.). La base de datos combina los procesos en una estructura que responde a las necesidades de las reglas del negocio.



En cambio en un DW, estos elementos se organizan alrededor de sujetos clientes (vendedores, productos, sucursales, etc.).

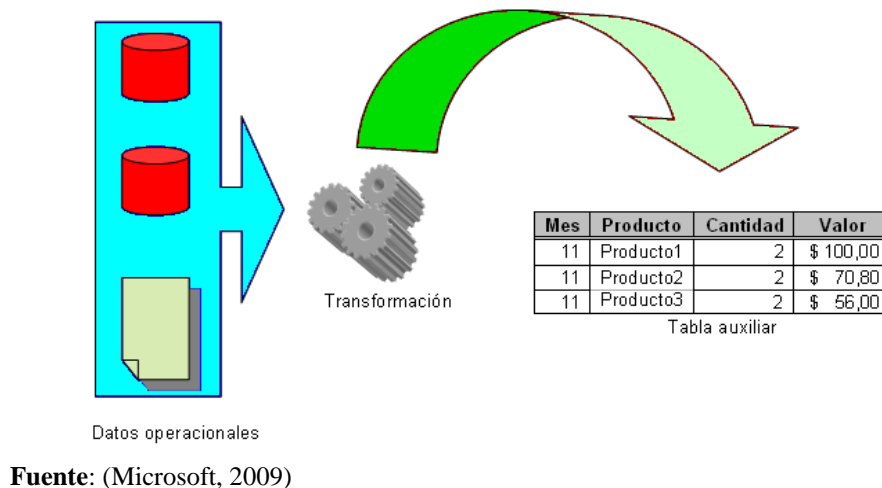
Una vez que el análisis del negocio se reconoce como un valor significativo para una organización, las peticiones de los datos y de la información llegan a ser numerosas y frecuentes.

Satisfacer estas peticiones puede ser una tarea muy compleja en un sistema OLTP, se debe bucear por grandes cantidades de datos obtenidos de distintas fuentes, procurando seleccionar, adecuar y consolidar la información. En un sistema OLAP, estos temas se resuelven por única vez, en la etapa de diseño.

#### 2.8.2.4. Transformación y agrupación de datos ETL

Los datos que alimentan a un sistema DW provienen de diferentes fuentes, estas fuentes son los distintos sistemas operacionales que la empresa posee, generalmente ni son homogéneos entre sí ni concuerdan exactamente con lo que se necesita, por lo que será necesario realizar todas las adaptaciones pertinentes.

Los diferentes procesos que se concentran en el concepto de toma, transformación y carga de datos en un DW se denominan ETL, sus siglas en inglés significan Extract – Transform – Load.



Fuente: (Microsoft, 2009)

Figura 5. ETL

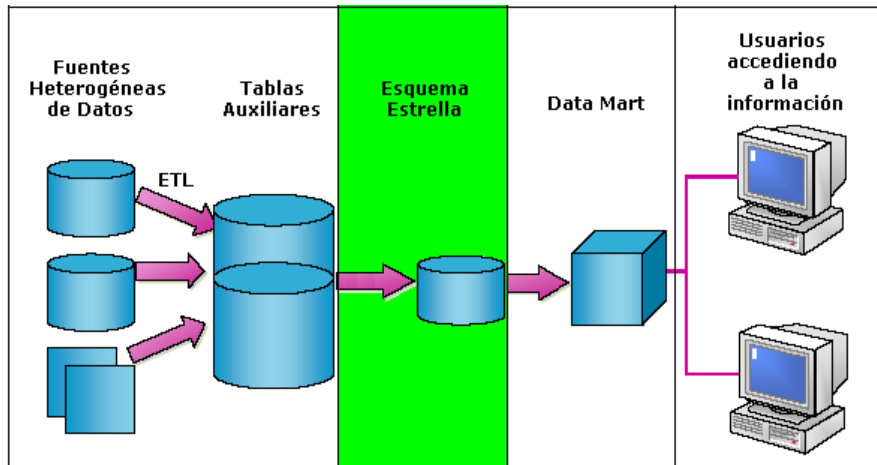
Es común que los sistemas operacionales que se encuentran en las organizaciones hayan sido desarrollados por diferentes equipos de programadores o empresas de software, y en su desarrollo, hayan adoptado diferentes convenciones en la codificación de variables, nombres de los atributos de las tablas, diferentes tipos de datos o formatos de fechas. (Microsoft, 2009)

Al reunir datos de los diferentes sistemas, se debe definir una norma única para el DW y realizar las transformaciones que sean necesarias en cada caso. Básicamente deben realizarse las siguientes tareas:

- Establecer las reglas que serán utilizadas para realizar la transformación.
- Detectar las inconsistencias que pueden originarse al tomar los datos desde distintas fuentes.

- Planificar cuidadosamente y con detalles la transformación de los datos que den como resultado final conjuntos de datos consistentes.

### 2.8.3. Diseñando una solución OLAP



Fuente: (Microsoft, 2009)

Figura 6. Segunda etapa Data Mart

#### 2.8.3.1. Construyendo el data mart

Hasta ahora hemos analizado los requerimientos del usuario, y depuramos sus datos para la formación del data warehousing, en esta unidad comenzaremos a diseñar el modelo del data mart. Este modelo, será el paso previo al armado de nuestra base de datos OLAP.

En esta etapa vamos a modelar las tablas relacionales en una gran estructura des normalizada, compuesta por tabla de hechos, y tablas más pequeñas que definirán las n-dimensiones o aperturas de nuestro cubo, llamadas tablas de dimensiones.

Para ello, primero debemos conocer algunos conceptos que tendremos en cuenta en la construcción del modelo.

#### 2.8.3.2. Esquema Estrella

Para facilitar el análisis, el data mart organiza los datos en una estructura llamada esquema de estrella.

Esta estructura está compuesta por una tabla central - tabla de hechos - y un conjunto de tablas organizadas alrededor de ésta - tablas de dimensiones.

En las puntas de la estrella se encuentran las tablas de dimensión que contienen los atributos de las aperturas que interesan al negocio que se pueden utilizar como criterios de filtro y son relativamente pequeñas. Cada tabla de dimensión se vincula con la tabla de hechos por un identificador.

Las características de un esquema de estrella son:

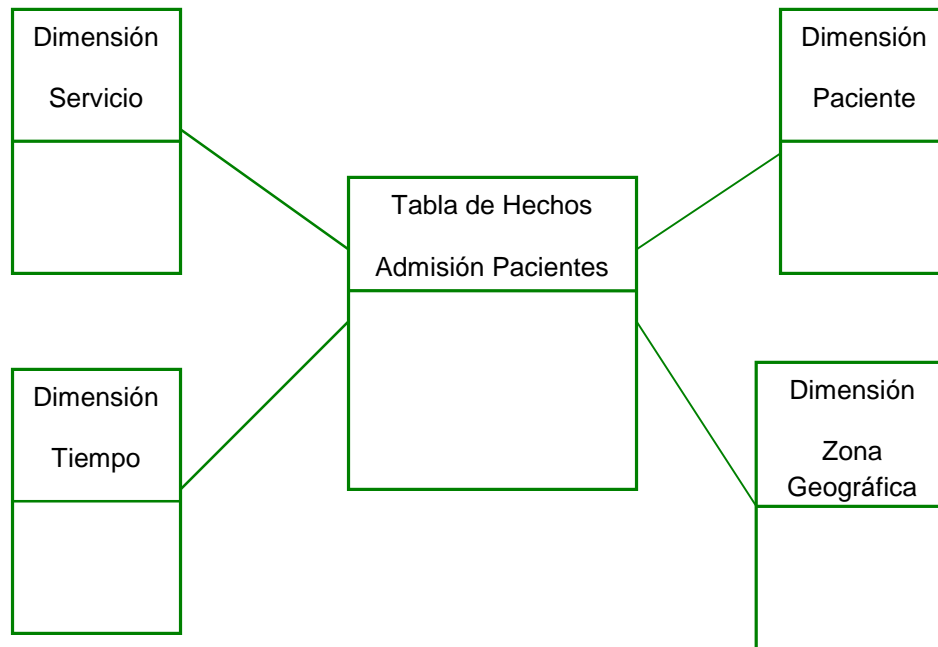
El centro de la estrella es la tabla de hecho.

Los puntos de la estrella son las tablas de dimensiones.

Cada esquema está compuesto por una sola tabla de hechos

Generalmente es un esquema totalmente des normalizado, pudiendo estar parcialmente normalizado en las tablas de dimensiones.

En el ejemplo construimos un esquema estrella considerando que se necesita analizar cómo evoluciona la Admisión de Pacientes (Hecho) por servicio, pacientes y zona geográfica a lo largo del tiempo.



**Fuente:** (Microsoft, 2009)

**Figura 7.** Ejemplo esquema estrella

## **Tabla de Hechos**

El modelo dimensional divide el mundo de los datos en dos grandes tipos: las medidas y las dimensiones de estas medidas. (Microsoft, 2009)

Las medidas, siempre son numéricas, se almacenan en las tablas de hechos y las dimensiones que son textuales se almacenan en las tablas de dimensiones.

La tabla de hechos es la tabla primaria del modelo dimensional, y contiene los valores del negocio que se desea analizar.

Cada tabla de hechos contiene las claves externas, que se relacionan con sus respectivas tablas de dimensiones, y las columnas con los valores que serán analizados.

Ejemplos de hechos:

- En un hospital: admisión de pacientes
- En un operador telefónico: Tráfico telefónico

Un hecho es un concepto de interés primario para el proceso de toma de decisiones, corresponde a eventos que ocurren dinámicamente en el negocio de la empresa.

## Dimensiones

Diseñaremos y construiremos cada dimensión basados en los procesos de negocio definidos por el cliente.

Las dimensiones organizan los datos en función de un área de interés para los usuarios.

Cada dimensión describe un aspecto del negocio y proporciona el acceso intuitivo y simple a datos.

Una dimensión provee al usuario de un gran número de combinaciones e intersecciones para analizar datos.

Las tablas de dimensiones son las compañeras de las tablas de hechos. Cada dimensión se define por su clave primaria que sirve para mantener la integridad referencial en la tabla de hechos a la que se relaciona.

Un cubo requiere que se defina al menos una dimensión en su esquema.

## Relaciones y Estructura de una dimensión

Cada nivel de una dimensión debe corresponderse con una columna en la tabla de la dimensión. Los niveles se ordenan por grado de detalle y se organizan en una estructura jerárquica. Cada nivel contiene miembros, los miembros son los valores de la columna que define el nivel.

Entre los miembros y entre los niveles de una dimensión existen relaciones, estas se pueden comprender como las relaciones que existen en un árbol genealógico donde los términos padre, hijo, hermano, primo, etc. indican una correspondencia entre elementos del árbol; y los miembros de la dimensión se comportan como familiares dentro del árbol genealógico. (Microsoft, 2009)

- **Padre:** Es el miembro del nivel inmediatamente superior que se relaciona con el miembro seleccionado. Cada elemento tiene un solo padre.
- **Hijo:** Son los elementos del siguiente nivel inferior que se relacionan con el miembro seleccionado. Pueden existir varios hijos para un mismo miembro.
- **Hermano:** Son los miembros que se encuentran en el mismo nivel que el miembro seleccionado y poseen el mismo padre.
- **Primo:** Son los miembros que se encuentran en el mismo nivel que el miembro seleccionado, pero que tienen diferentes padres. Los primos tienen padres que son hermanos.
- **Descendientes:** Son todos los miembros que se encuentran debajo del nivel del miembro seleccionado. independientemente de la cantidad de niveles que los separen.

### Ejemplos de dimensión

✓ Dimensión zona geográfica

* PAIS	ARGENTINA		BRASIL	URUGUAY
** PROVINCIA	BUENOS AIRES	CORDOBA	SAN PABLO	MONTEVIDEO
*** CIUDAD	MAR del PLATA	LA PLATA	VILLA GRAL. BELGRANO	....

### Ejemplos de relaciones

En una dimensión zona geográfica tendríamos las siguientes relaciones entre niveles y entre miembros:

✓ Padre:

Argentina es padre de Buenos Aires y de Córdoba

✓ Hijo:

Buenos Aires y Córdoba son hijos de Argentina

✓ Hermano:

Buenos Aires y Córdoba son hermanos el uno al otro, también son hermanos Argentina, Brasil y Uruguay.

✓ Primo:

Mar del Plata es primo de Villa General Belgrano.

✓ Descendiente:

Todos los miembros que estén por debajo de Argentina son sus descendientes, por ejemplo Buenos Aires, Mar del Plata y Villa General Belgrano son alguno de sus descendientes.

✓ Ancestro:

Mar del Plata tiene dos antepasados Buenos Aires y Argentina.

- **Ancistros:** Son todos los miembros que se encuentran por encima del nivel del miembro seleccionado.
- **Un miembro es independiente de las relaciones.** Cada integrante de la dimensión es miembro de ella.

**Fuente:** (Microsoft, 2009)

**Tabla 9. Ejemplo de dimensión.**

Las dimensiones pueden ser:

- Locales
- Compartidas

Las **dimensiones locales** son las que se definen y se utilizan dentro de un mismo cubo.

Las **dimensiones compartidas** son aquellas dimensiones que se definen independientes de los cubos y pueden ser utilizadas por varios de ellos.

Ventajas de las dimensiones compartidas

- Evitamos duplicar dimensiones locales
- Aseguramos que los datos analizados estén organizados de la misma forma en todos los cubos, lo que implica un menor costo de mantenimiento.

Desventajas de las dimensiones compartidas

- Deben emplearse del mismo modo en los cubos que las usen.
- Un cambio implica que la dimensión deberá ser modificada en todos los cubos

### Ejemplos de Dimensión Compartida

La dimensión Producto puede utilizarse para el Data Mart Ventas y para el Data Mart Producción.

Así, la dimensión producto es una dimensión compartida por los dos Data Mart.

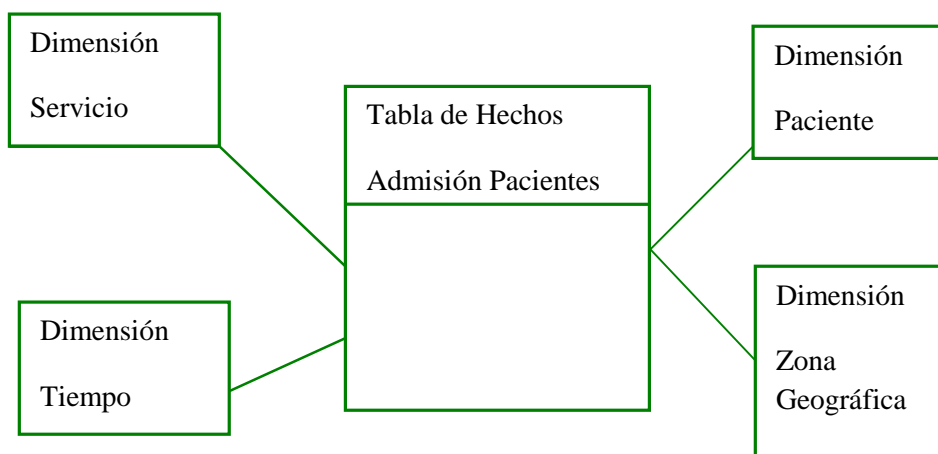
### Dimensiones: Esquema Estrella

En el **esquema estrella** cada dimensión está compuesta por una sola tabla, esta tabla esta des normalizada.

El esquema se denomina así debido a que el diagrama se parece a una estrella.

Debido a que las tablas de dimensión están des normalizadas lograremos en el modelo del data mart, una menor cantidad de tablas.

Este es un esquema donde las dimensiones tienen un esquema estrella.



**Fuente:** (Microsoft, 2009)

**Figura 8.** Dimensiones esquema estrella

## Dimensiones: Esquema Copo de Nieve

El esquema copo de nieve es una variación del esquema estrella donde alguna punta de la estrella se explota en más tablas. (Microsoft, 2009)

El nombre del esquema se debe a que el diagrama se asemeja a un copo de nieve.

En este esquema, las tablas de dimensión copo de nieve se encuentran normalizadas para eliminar redundancia de datos.

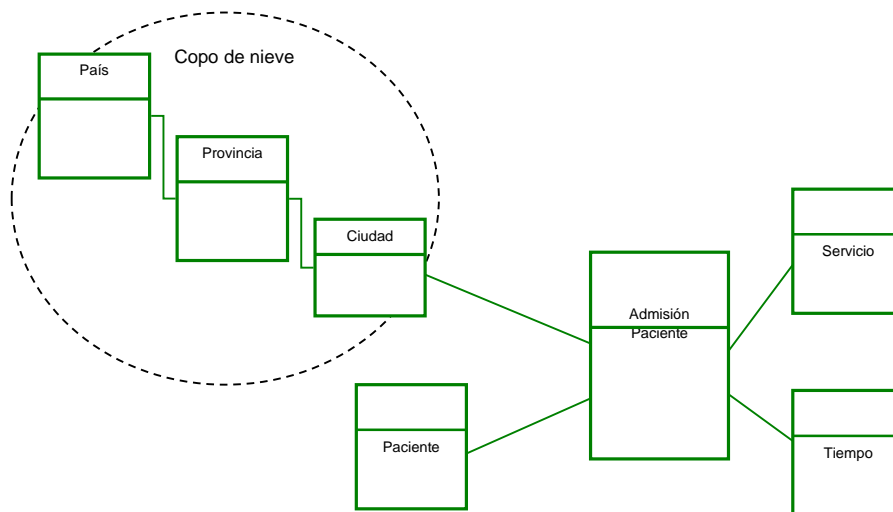
A diferencia del esquema estrella, los datos de las dimensiones se reparten en múltiples tablas.

Como ventaja del esquema destacamos el ahorro de espacio de almacenamiento en disco, pero en perjuicio de un aumento en la cantidad de tablas.

Los siguientes son las características de un copo de nieve:

- La dimensión está normalizada
- Los distintos niveles se encuentran almacenados en tablas separadas
- Se argumenta ahorro de espacio

Se muestra un esquema donde la dimensión zona geográfica presenta un esquema copo de nieve.



**Fuente:** (Microsoft, 2009)

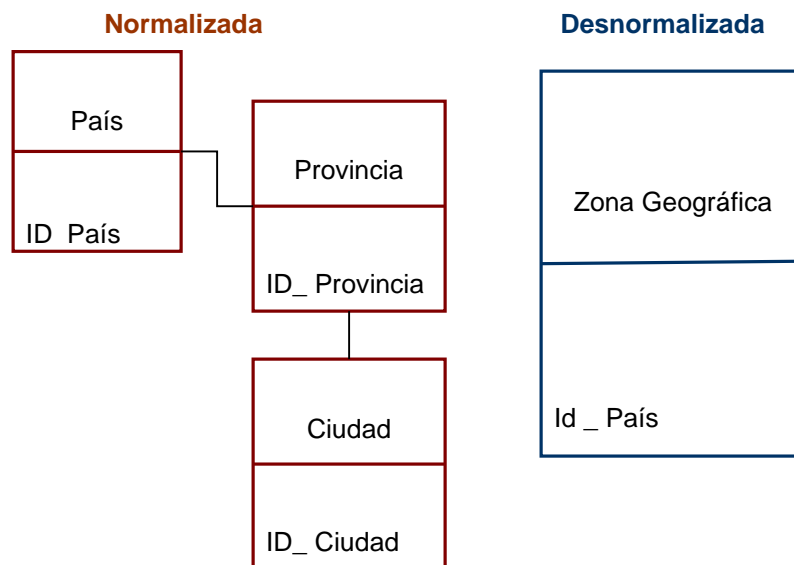
**Figura 9.** Ejemplo dimensión

	<b>Estrella</b>	<b>Copo de nieve</b>
<b>Cantidad de tablas</b>	Menor	Mayor
<b>Consultas</b>	Mejora la performance	Aumenta la cantidad de uniones entre tablas provocando baja en la performance
<b>Almacenamiento</b>	Aumenta el espacio	Ahorra espacio

### Ejemplo de Tabla Normalizada y Tabla Desnormalizada

En la imagen vemos en la tabla normalizada los datos nombre de país y nombre de provincia aparecerán una sola vez en las tablas País y Provincia respectivamente.

Si en cambio, la tabla esta desnormalizada tendremos redundancia de datos, ya que se repetirán los datos del País y de la Provincia por cada Ciudad.



**Fuente:** (Microsoft, 2009)

**Figura 10.** Ejemplo tabla normalizada y desnormalizada



**Fuente:** (Microsoft, 2009)

**Tabla 10.** Características esquemas.

### **Dimensiones: Padre – Hijo (Parent – Child)**

Una dimensión padre-hijo es una dimensión donde el dato del Padre se relaciona con el Hijo y ambos se encuentran en la misma tabla de dimensión, es decir, la dimensión se relacionan consigo misma. (Microsoft, 2009)

#### **Ejemplos de Dimensión Padre - Hijo**

La dimensión Cuenta Contable donde una cuenta imputable forma parte de un Sub Rubro y el Sub Rubro a su vez forma parte de un Rubro. Estos datos se encuentran en un solo Plan de Cuentas.

La cuenta Activo, contiene los rubros Inversiones, Créditos y Caja, y el rubro Caja a su vez contiene Caja y Fondo Fijo.

### **Dimensiones Virtuales**

Las dimensiones virtuales, no requieren un almacenamiento físico en el cubo, se evalúan en el momento de la consulta.

Funcionan de manera similar a las dimensiones reales y son transparentes para el usuario.

#### **Ejemplos de Dimensión Virtual**

Podemos tener una dimensión Producto organizada de la siguiente manera:

##### **Producto (Dimensión real)**

.....

Fabricante

Marca

Calibre

Producto

Si el usuario requiere que sus análisis de información se realicen por Marca, utilizando la dimensión Producto requerirá seleccionar a cada fabricante para obtener la información de la marca.

Para evitar esto, podemos crear una dimensión virtual donde el orden de los niveles Fabricante - Marca están invertidos, que le permita ver sus datos por Marca sin necesidad de seleccionar a todos los fabricantes. Esta dimensión la construiremos de la siguiente manera:

### Producto\_Marca (Dimensión virtual 1)

.....

Marca

Fabricante

Calibre

Producto

Otra necesidad del usuario podría ser obtener los totales o filtros de calibre sin importar la marca o el fabricante, entonces construiríamos una dimensión virtual que contenga solo la columna calibre.

Calibre (Dimensión virtual 2)

Calibre

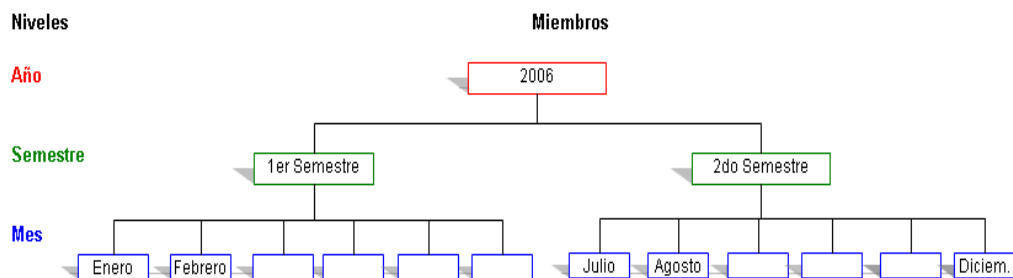
### La dimensión Tiempo

Mencionaremos esta dimensión ya que ocupa un lugar especial en cada data mart. Recordemos que el tiempo es parte implícita de la información que contiene el data mart.

Esta dimensión la podemos definir separándola en distintas jerarquías de tiempo:

- Año
- Semestre
- Mes

### Ejemplos de Dimensión Tiempo



Fuente: (Microsoft, 2009)

Figura 11. Ejemplo dimensión tiempo

La definición de la jerarquía la haremos teniendo en cuenta las necesidades que tiene la organización. Debemos contemplar los periodos de tiempo por los cuales la información necesita ser analizada y la regularidad con la que se cargaran los datos en el cubo.

Consideraciones para esta dimensión:

**Nombres de los miembros:** Cuando construyamos la dimensión tiempo es conveniente que los nombres de los miembros sean únicos. Así, si utilizamos una nomenclatura para la jerarquía MES que sea “Mes – Año” cuando busquemos un periodo debemos identificarlo como Julio – 2006. De esta manera nos ahorramos de utilizar dos niveles de la dimensión logrando una mayor calidad en los informes.

Si en cambio, el nombre de la jerarquía MES se compone solo del nombre del mes, para identificar el periodo Julio del 2006 primero debemos seleccionar sobre el nivel Año y luego sobre el nivel Mes.

**Puede existir más de una:** Cabe aclarar que no necesariamente esta dimensión es única dentro del cubo, podríamos tener que armar **más de una dimensión Tiempo**. Si necesitáramos analizar la información de la empresa en base al año calendario y realizar otro análisis basándonos en el año fiscal, deberíamos construir dos dimensiones de tiempo para el mismo data mart.

### 2.8.3.3. Medidas

Las medidas son los valores de datos que se analizan.

Una medida es una columna cuantitativa, numérica, en la tabla de hechos. Las medidas representan los valores que son analizados, como cantidad de pacientes admitidos o llamadas efectuadas.

Las medidas son:

- Valores que permiten analizar los hechos
- Valores numéricos porque estos valores son las bases de las cuales el usuario puede realizar cálculos.

Si la medida fuera un valor no numérico debemos codificarla a un valor numérico en el proceso de obtención de datos, y luego cuando tengamos que exponer sus valores decodificarla para mostrarla con el valor original.

Las siguientes son algunas de las características de las medidas:

- Deben ser numéricas.
- Cruzan todas las dimensiones en todos los niveles.

Las medidas pueden clasificarse en:

- Naturales
- Calculadas

### Ejemplos de Medidas

En un hospital, donde el hecho es Admisión de Pacientes las medidas pueden ser:

- Pacientes Admitidos
- Pacientes Atendidos

En un operador telefónico, donde el hecho es Tráfico Telefónico, las medidas pueden ser:

- Llamadas Cantidad
- Llamadas Duración

Ejemplos de Medidas no numéricas

Supongamos el hecho Recursos Humanos, donde podemos tener la medida Sexo que toma los valores "F" o "M".

Estos valores debemos codificarlos en valores numéricos durante el proceso de transformación de datos (ETL). Así, por ejemplo tendremos 0="F" y 1="M".

Cuando el usuario visualice esta medida, debemos volver los datos a sus valores originales (decodificarlos) para mostrar "F" o "M".

## Medidas Naturales

Son las columnas numéricas que queremos analizar que provienen directamente de los sistemas OLTP.

Cuando definimos una medida debemos tener en cuenta cual será la forma de **agregación** (agrupación de la misma) al subir por la estructura dimensional.

Estas formas de agregación pueden ser:

- Suma: es la operación que suma los valores de las columnas
- Cuenta: realiza un conteo de los valores
- Mínima: devuelve un valor mínimo
- Máxima: proporciona el mayor de los valores
- Cuenta de Distintos: cuenta los valores diferentes

Las agregaciones son resúmenes de datos pre calculados que mejoran el tiempo de respuesta por el simple hecho de tener preparadas las respuestas antes de que se planteen las preguntas.

## Medidas Calculadas

Son las medidas que se calculan en el cubo en base a los valores de las medidas naturales.

El sentido de la expresión medidas calculadas es muy amplio y engloba a cualquier manipulación de las medidas naturales que nos faciliten el análisis de los hechos.

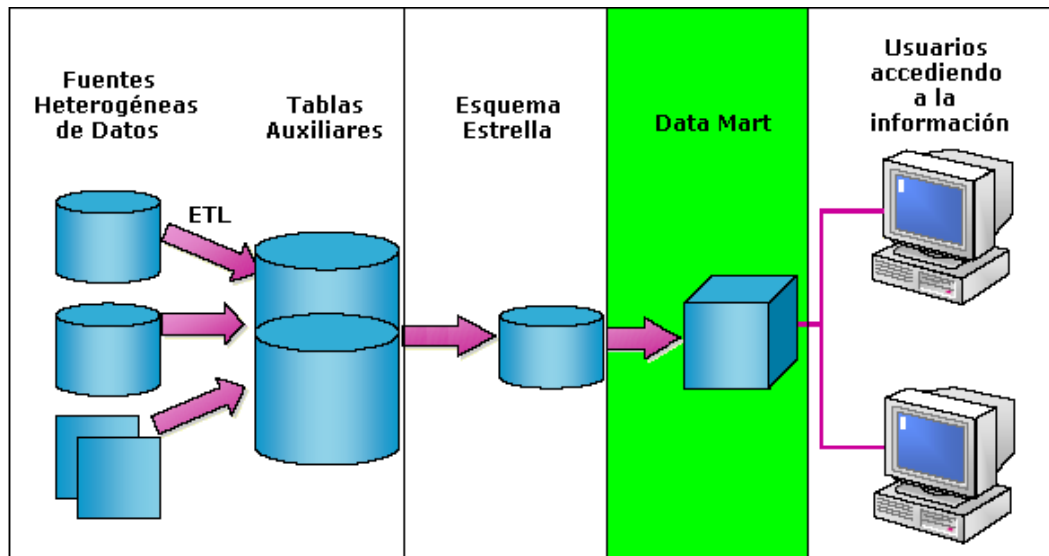
En una medida calculada puede haber:

- Cálculos Matemáticos

- Expresiones condicionales
- Alertas

Estos tres tipos (cálculos, condiciones y alertas) usualmente pueden existir juntos dentro de la misma medida calculada.

## 2.8.4. Construyendo una solución OLAP



Fuente: (Microsoft, 2009)

Figura 12. Tercera etapa data mart

En esta unidad abordaremos los conceptos a tener en cuenta para la implementación del data mart. Describiremos los distintos tipos de almacenamiento y las consideraciones que debemos analizar para mejorar la performance del sistema. (Microsoft, 2009)

Además, veremos con qué frecuencia es conveniente procesar nuestros cubos y explicaremos el uso de los cubos virtuales y particiones.

Al final de este módulo, el lector conocerá qué modo de almacenamiento será el más adecuado para los requerimientos de la organización, como balancear los distintos factores que intervienen al implementar un cubo.

### 2.8.4.1. Tipos de Almacenamiento

Haciendo un pequeño balance de las unidades anteriores, vemos que ya tenemos: un diseño de requerimientos, sabemos de dónde y cómo obtener los datos y ya contamos con la definición de la estructura multidimensional. Ahora que vamos a armar físicamente el cubo debemos elegir entre los distintos modos de almacenamiento que podemos utilizar. Para facilitar esta elección, desarrollaremos los conceptos de **MOLAP**, **ROLAP** y **HOLAP** y luego haremos una comparación de los mismos.

#### **MOLAP**

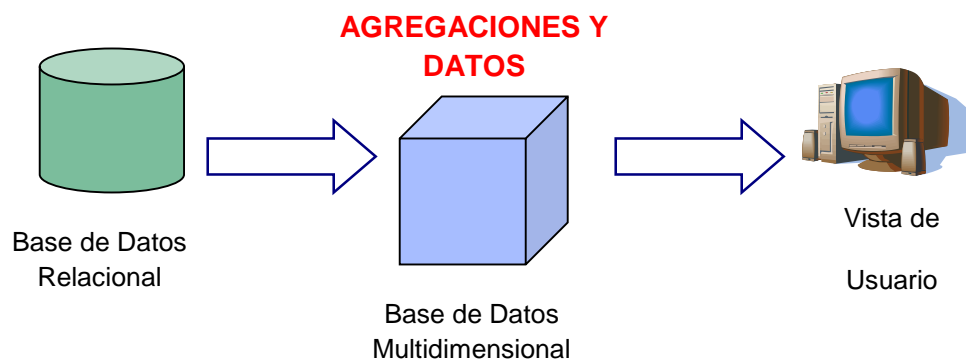
En el modo de almacenamiento MOLAP (OLAP Multidimensional) una copia de los datos de origen del cubo, junto con sus agregaciones, es almacenada en una estructura multidimensional.

Debemos tener en cuenta que mientras los datos de origen cambian directamente con las operaciones, los objetos con almacenamiento MOLAP deben ser procesados para incorporar estos cambios.

El tiempo comprendido entre un procesamiento y el siguiente, crea un periodo de latencia durante el que puede que la información OLAP no coincida con los datos de origen actuales.

Como característica del almacenamiento MOLAP podemos destacar:

- Provee excelente rendimiento y compresión de datos.
- Tiene mejor tiempo de respuesta, dependiendo solo del porcentaje de las agregaciones del cubo.
- La estructura está muy optimizada para maximizar el rendimiento de las consultas.
- En general este método, es muy apropiado para cubos con uso frecuente por su rápida respuesta.



**Fuente:** (Microsoft, 2009)

**Figura 13.** MOLAP

## ROLAP

En un modelo ROLAP (**OLAP Relacional**) toda la información del cubo, sus datos, su agregación, sumas, etc., son almacenados en una base de datos relacional.

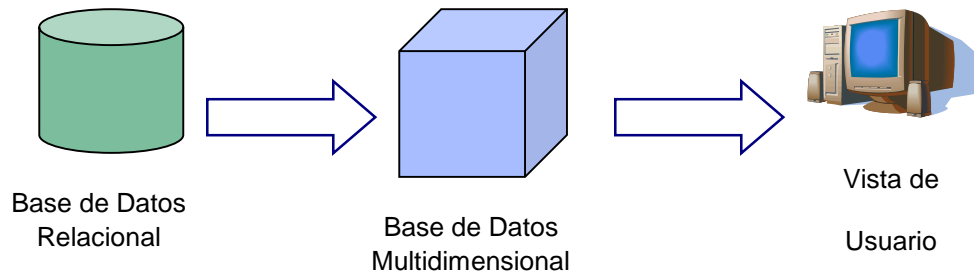
A diferencia del modo de almacenamiento MOLAP, ROLAP no almacena copia de la base de datos, accede a las tablas originales cuando necesita responder a las consultas, generalmente es mucho más lenta que las otras estrategias de almacenamiento (MOLAP o HOLAP).

ROLAP se utiliza para ahorrar espacio de almacenamiento cuando se trabaja con grandes conjuntos de datos que se consultan con poca frecuencia; por ejemplo, datos exclusivamente históricos.

Los usos comunes de este esquema son:

- Cuando los clientes desean ver los cambios inmediatamente.
- Cuando contamos con grandes conjuntos de datos que no son frecuentemente buscados

## AGREGACIONES Y DATOS



Fuente: (Microsoft, 2009)

Figura 14. ROLAP.

## HOLAP

HOLAP (**OLAP híbrido**) combina atributos de MOLAP y ROLAP.

Al igual que MOLAP, HOLAP hace que las agregaciones se almacenen en una estructura multidimensional, y los datos a nivel de detalle, en una base de datos relacional como lo hace el almacenamiento ROLAP. (Microsoft, 2009)

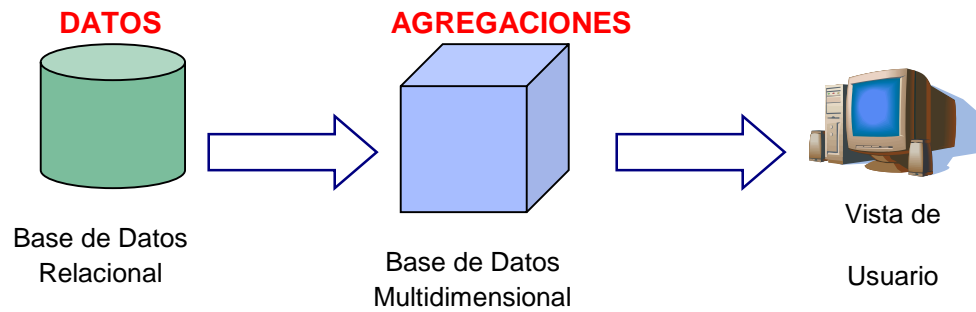
Para procedimientos de búsqueda que accedan datos sumariados, HOLAP es equivalente a MOLAP. Por el contrario, si los procesos de consultas accedieran a los máximos niveles de detalle, deberían recuperar los datos de la base de datos relacional y esto no sería tan rápido comparado con una estructura MOLAP.

Los cubos almacenados como HOLAP, son más pequeños que los MOLAP y responden más rápidos que los ROLAP.

Usos comunes de HOLAP

- Cubos que requieren rápida respuesta
- Cuando existen sumalizaciones basadas en una gran cantidad de datos de origen.
- Solución de compromiso para bajar el espacio ocupado sin perjudicar totalmente el rendimiento de las consultas.





Fuente: (Microsoft, 2009)

Figura 15. MOLAP.

Ejemplo del operador telefónico. Supongamos que:

- ✓ Se miden las llamadas realizadas x Día y x Cliente.
- ✓ El tiempo se estructura en Día – Mes – Año.
- ✓ Los Clientes se estructuran en Cliente – Ciudad – País.

	Definición	MOLAP	ROLAP	HOLAP
<b>Datos detallados</b>	Llamadas para un Día y Cliente	EM	BR	BR
<b>Datos sumarios</b>	Suma de llamadas para algún cruce de Cliente – Tiempo donde al menos uno de las dos dimensiones no esté en el mínimo nivel. (Cliente y Mes ó Año, Día y Ciudad o País, etcétera)	EM	BR	EM
EM = Estructura Multidimensional				
BR = Base de Datos Relacional				

Fuente: (Microsoft, 2009)

Tabla 11. Ejemplo comparaciones MOLAP ROLAP HOLAP.

	MOLAP	ROLAP	HOLAP
<b>Almacenamiento de las Agregaciones</b>	Modelo Multidimensional	Base de datos relacional	Modelo Multidimensional
<b>Almacenamiento de los datos</b>	<b>Modelo Multidimensional</b>	<b>Base de datos relacional</b>	<b>Base de datos relacional</b>
<b>Facilidad de Creación</b>	Sencillo	Muy Sencillo	Sencillo
<b>Velocidad de respuesta</b>	Buena	Regular o Baja	Buena para consultas que posean agregaciones, Regular para datos de bajo nivel
<b>Escalabilidad</b>	Problemas de escalabilidad	Son más escalables	
<b>Recomendados para</b>	Cubos con uso frecuente	Datos que no son frecuentemente usados	Si el cubo requiere una rápida respuesta

**Fuente:** (Microsoft, 2009)

**Tabla 12.** Ejemplo comparaciones MOLAP ROLAP HOLAP.

	<b>Ventajas</b>	<b>Desventajas</b>
<b>MOLAP</b>	Mejor performance en los tiempos de respuesta	Duplica el almacenamiento de datos (ocupa más espacio)  Tiempo de Latencia

<b>ROLAP</b>	Ahorra espacio de almacenamiento. Útil cuando se trabaja con muy grandes conjuntos de datos.	El tiempo de respuesta a consultas es mayor.
<b>HOLAP</b>	Buen tiempo de respuesta sólo para información sumariada	Volúmenes de datos más grandes en la base de datos relacional

**Fuente:** (Microsoft, 2009)

**Tabla 13.** Ejemplo comparaciones MOLAP ROLAP HOLAP.

- MOLAP es un OLAP basado en el acceso a una base de datos multidimensional
- ROLAP es un OLAP basado en el acceso a una base de datos relacional
- HOLAP es un OLAP situado entre ROLAP y MOLAP, accede a la Multidimensional y a la Relacional.

#### **2.8.4.2. Definición de Agregaciones**

Otro factor para considerar en la implementación del modelo OLAP, además del modo de almacenamiento, es la definición del porcentaje de agregaciones. (Microsoft, 2009)

Se denomina agregación al proceso de pre calcular el cálculo de los datos a través de los niveles, para disminuir los tiempos de respuestas en los procesos de búsquedas de información. El porcentaje de agregación da idea de la proporción o profundidad hasta la que se realizarán los pre-cálculos.

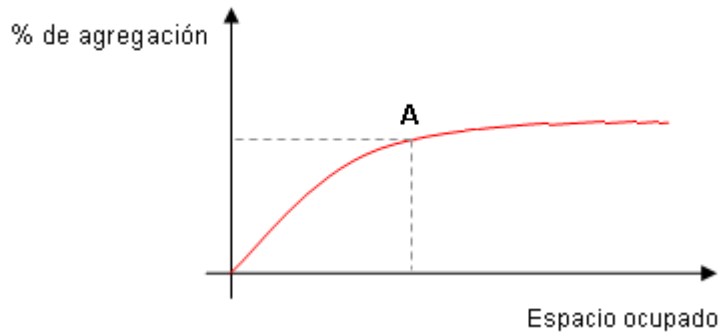
Las agregaciones se almacenan en la estructura multidimensional (según el modo de almacenamiento que escogimos).

Cuando definamos agregaciones debemos tener en cuenta de especificar las restricciones de almacenamiento y de porcentaje de agregación, a fin de lograr una buena solución de compromiso entre el tiempo de respuesta a las consultas y los requisitos de almacenamiento.

Si calculáramos todas las agregaciones posibles necesitaremos gran cantidad de tiempo de procesamiento y espacio de almacenamiento. Si por el contrario, no se pre calculan agregaciones (0%), la cantidad de espacio de almacenamiento que se necesita se reduce al mínimo, pero el tiempo de respuesta aumenta.

Por lo tanto, suele existir un equilibrio entre el espacio de almacenamiento, el porcentaje de posibles agregaciones que se pre calculan y la performance requerida.

En la figura mostramos la gráfica de esta relación:



**Fuente:** (Microsoft, 2009)

**Figura 16.** Relación espacio almacenamiento – agregaciones.

En el gráfico podemos observar que llega un punto en el cual ya no se consigue un aumento significativo en las agregaciones (debemos recordar que, en este contexto, aumentar las agregaciones es el sinónimo de mejorar la performance en las consultas), a pesar de aumentar la cantidad de espacio de almacenamiento. Debemos escoger un porcentaje situado en la zona del punto **A**, donde logramos el máximo porcentaje de agregación con la menor cantidad de espacio posible.

Características de las agregaciones:

Las agregaciones permiten mejorar los tiempos de respuesta

Requieren de almacenamiento adicional

Si no son controladas pueden provocar una explosión en los requisitos de almacenamiento

A mayor número de agregaciones más tiempo de procesamiento y más requerimiento de espacio

A menor número de pre agregaciones peor tiempo de respuesta de las consultas

### 2.8.4.3. Procesamiento de cubos

En esta etapa debemos definir cuándo y con qué frecuencia procesar los cubos.

Cuando se procesan Dimensiones o cubos se están actualizando los datos, las estructuras multidimensionales o ambas cosas.

Esta definición debe considerar los siguientes factores

- Modo de almacenamiento que escogimos (MOLAP-ROLAP-HOLAP),
- Tamaño de la tabla de hechos (cantidad de registros)
- Numero de dimensiones del modelo
- Porcentaje de agregaciones

Para determinar la frecuencia con que procesaremos el cubo debemos tener en cuenta lo analizado con el cliente respecto de la granularidad de los datos para el tiempo. EL nivel de detalle (día, mes, etcétera) nos fijará la periodicidad de actualización de los datos.

A diferencia de los sistemas OLTP en los que la actualización de los datos se realiza en línea con las transacciones y la agregación de los datos se realiza en el momento en que el usuario realiza una consulta, en OLAP el procesamiento de los cubos se realiza a contra turno, en los horarios en que no se afecta la tarea de los usuarios.

#### 2.8.4.4. Cubos Virtuales

Los cubos virtuales son vistas de cubos reales. Los cubos virtuales pueden ser utilizados:

- Cuando el usuario desee ver información conjunta de dos cubos diferentes.
- Cuando se quiere tener una visión parcial de un cubo. Es una forma de simplificar el manejo de la seguridad.

#### 2.8.4.5. Particiones

Los cubos están **compuestos por particiones**. Como su nombre lo sugiere, una partición es una división o fraccionamiento de la información que conforma a un cubo. Cada cubo contiene al menos una partición, pero puede estar compuesto por múltiples particiones,

Las particiones de un cubo son **invisibles** para el usuario, pero su uso aumenta la carga de trabajo del administrador del modelo multidimensional.

Para cada partición podemos definir la fuente de datos, el modo de almacenamiento y el porcentaje de agregación de manera independiente de las demás particiones.

Además, una partición de datos puede ser actualizada independientemente de las otras. Esta propiedad es muy importante ya que nos brinda la ventaja de mejorar los tiempos de procesamiento si dividimos correctamente las particiones y las procesamos adecuadamente.

Así, si dividimos nuestro cubo en particiones definiremos cada uno de estos parámetros de la manera más indicada.

- Partición **más utilizada** (Tiempo Actual):
  - Modo de Almacenamiento MOLAP,
  - % de Agregación: alto
  - Frecuencia de procesamiento: alta
- Partición **medianamente consultada** (Tiempos intermedios):
  - Modo de Almacenamiento: HOLAP
  - % de Agregación: bajo
  - Frecuencia de procesamiento: ocasional
- Partición **poco accedida** (Períodos viejos):
  - Modo de Almacenamiento ROLAP,

% de Agregación: nulo

Frecuencia de procesamiento: muy baja (normalmente sólo al crear la partición)

Desde el punto de vista de la administración, se puede manejar cada partición como si fuera un cubo independiente. Puede tener fuente de datos, modo de almacenamiento, porcentaje de agregación y frecuencia de procesamiento propios.

Podemos crear una partición por cada año que contenga el cubo, (por ejemplo 2004, 2005 y 2006), y almacenar las particiones de la siguiente manera:

**Año 2006:** En una estructura MOLAP, con un alto porcentaje de agregaciones, para obtener una respuesta rápida a las consultas.

**Año 2005:** En una estructura HOLAP, con un bajo porcentaje de agregaciones, lo que permitirá buenos tiempos de respuesta para consultas de resumen, con un espacio de almacenamiento mínimo.

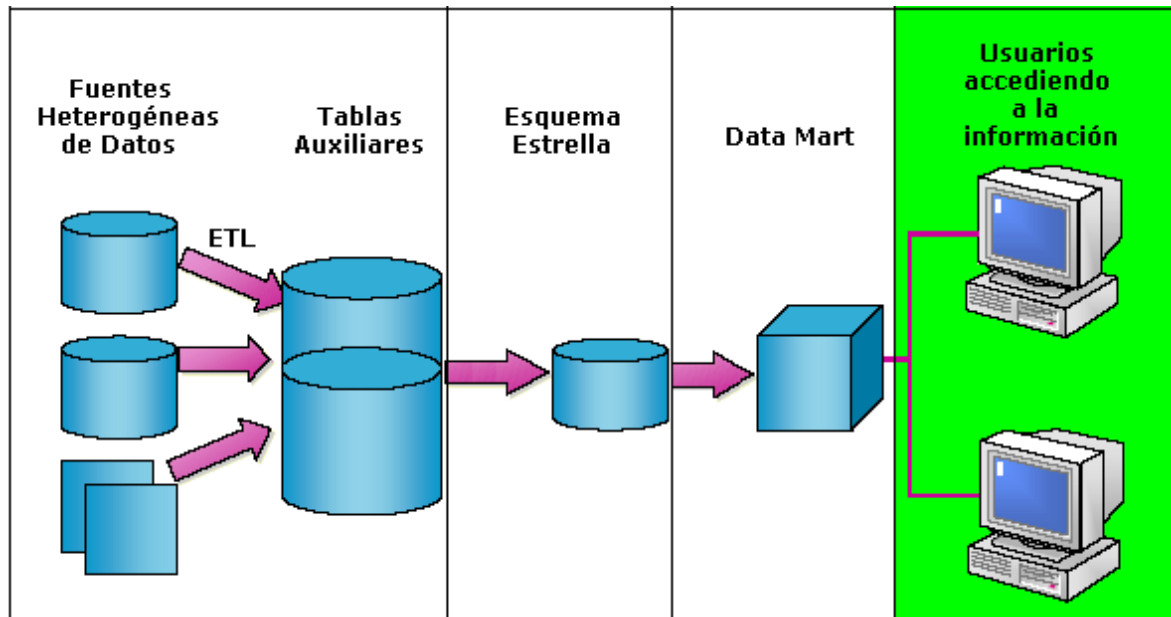
**Años anteriores:** En una estructura ROLAP, con porcentaje de agregaciones cero, lo que nos ahorrará espacio de almacenamiento. Este ahorro se paga con el aumento del tiempo de respuesta, pero no es caro, porque las consultas son ocasionales.

#### **2.8.4.6. La difícil búsqueda del equilibrio**

En el momento de implementar el cubo, debemos analizar en conjunto los siguientes factores, tratando de llegar a un punto de equilibrio.

- % de Pre agregación.
- Tiempo de Procesamiento.
- Requerimientos de tiempos de respuesta
- Tipo de almacenamiento
- Tipificación de las consultas (Base para decidir si se manejan particiones)
- Uso de Particiones

## 2.8.5. Implementando cubos OLAP



Fuente: (Microsoft, 2009)

Figura 17. Cuarta etapa Data Mart.

### 2.8.5.1. Seguridad

A la hora de diseñar el modelo multidimensional, es fundamental definir la seguridad adecuada sobre los diferentes componentes y niveles de la solución, debido a lo sensible que puede ser para la organización la información que suelen manejar este tipo de aplicaciones. (Microsoft, 2009)

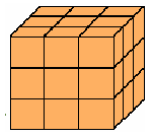
Al igual que ocurre con las bases de datos de los sistemas transaccionales, en OLAP pueden manejarse distintos niveles de seguridad.

La seguridad en OLAP tiene una arquitectura jerárquica, partiendo del cubo y llegando al nivel de celda dentro del cubo.

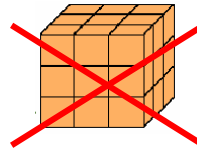
De este modo, podemos definir los permisos a nivel de:

- Cubo
- Dimensión
- Celda (Medida)

**Cubo:** esta restricción de seguridad se realiza sobre todo el cubo, se puede permitir o denegar el acceso al cubo.



**Permitido**

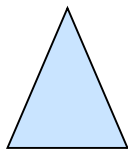


**Denegado**

Fuente: (Microsoft, 2009)

**Figura 18.** Permisos cubo.

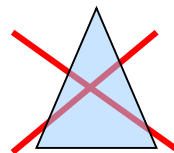
**Dimensión:** Podemos permitir que el usuario vea la dimensión, que acceda solo a una parte de ella, o que no tenga permiso de visualizarla.



**Permitido**



**Solo una parte**



**Denegado**

Fuente: (Microsoft, 2009)

**Figura 19.** Permisos dimensión

**Celda:** En una celda o medida podemos permitir el acceso, o bien personalizarlo utilizando expresiones que verifiquen alguna condición para acceder a los datos.

Otra opción para limitar los accesos puede ser el uso de  **cubos virtuales** . Podríamos crear un cubo virtual solo con las medidas que deseamos que tenga acceso el usuario y luego otorgar los permisos sobre el cubo virtual, y denegar o no otorgar permiso sobre el cubo original.

Por ejemplo, si solo un grupo de usuarios puede visualizar el importe de los sueldos de los empleados, entonces podríamos definir una restricción de acceso a nivel celda, sobre la medida Sueldo o crear un cubo virtual que no muestre esta medida.

### **2.8.5.2. Consultas**

Una vez que tenemos armado el cubo, los usuarios pueden realizar diferentes operaciones para poder visualizar y analizar sus datos.

Las operaciones que se pueden realizar son:

Drill - Down

Drill - Up

Slice y Dice. (Filtrado)

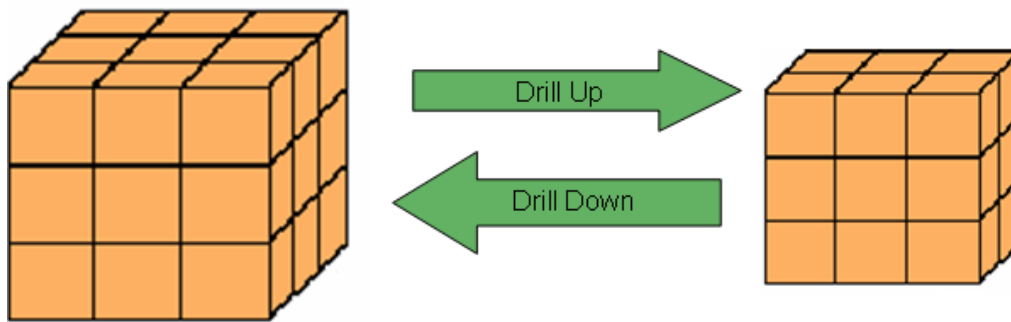


Rotación

Consolidación

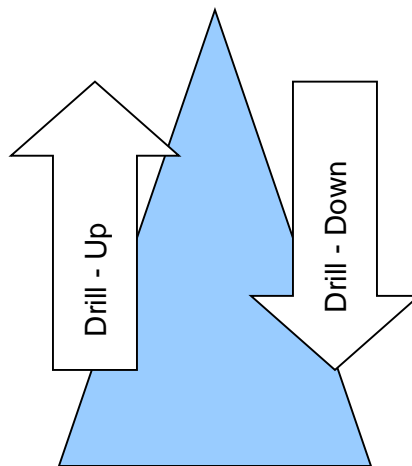
**Drill Down – Drill Up:** Es una técnica por la que el usuario puede navegar entre las jerarquías de una dimensión agrupando (Drill-up) o desagrupando (Drill-down) los datos. (Microsoft, 2009)

El drill down y el drill up sirven para navegar el cubo sobre sus dimensiones, con el drill up se pasa desde el detalle a la generalización, y con el drill down se pasa desde un nivel general al detalle.



Fuente: (Microsoft, 2009)

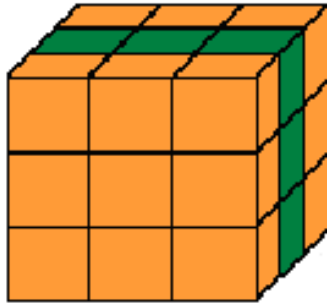
Figura 20. Drill up – Drill down.



Fuente: (Microsoft, 2009)

Figura 21. Drill up-Drill down.

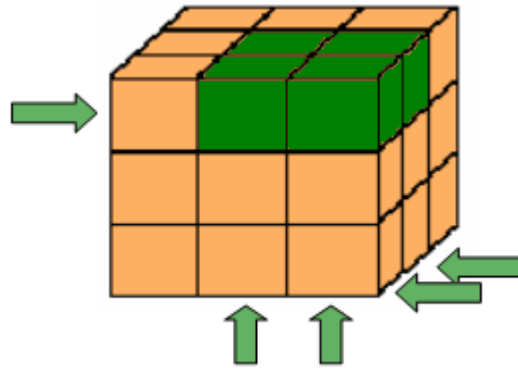
**Slice:** Al seleccionar un miembro en particular de una dimensión se forma una especie de rebanada (slice) del cubo original.



Fuente: (Microsoft, 2009)

Figura 22. Slice.

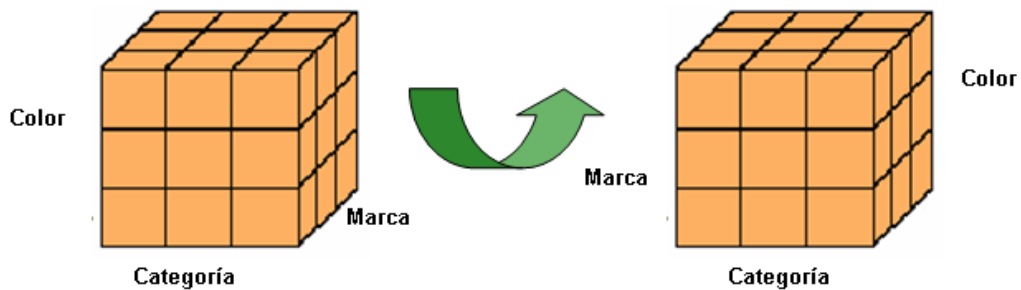
**Dice:** Al seleccionar varios miembros de varias dimensiones se forma sub-cubo, cubo más pequeño o dado (dice). Tanto Slice como Dice son formas particulares de **Filtrado**.



Fuente: (Microsoft, 2009)

Figura 23. Dice.

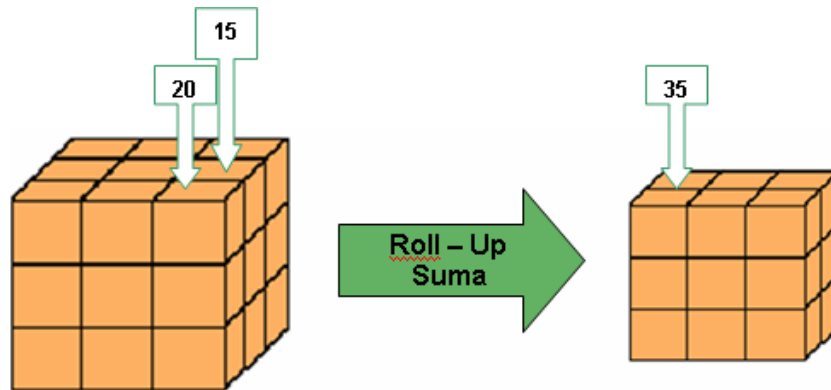
**Rotación:** Selecciona el orden de visualización de las dimensiones, rota o gira el cubo según sus dimensiones.



Fuente: Academia BI Microsoft.

Figura 24. Rotación.

**Consolidación (Roll-Up):** Calcula las medidas en función de agrupamientos, realiza el re-cálculo de la medida de acuerdo a los ajustes de escala.



Fuente: (Microsoft, 2009)

Figura 25. Consolidación.

### 2.8.5.3. Herramientas de visualización

La **navegación** es un término que usamos para describir la posibilidad que tienen los usuarios de recorrer las distintas dimensiones y sus cruces, visualizando para cada caso los valores resultantes de las medidas.

Estas son algunos tipos de herramientas que se pueden utilizar para navegar el cubo:

- **Planillas de Cálculo:** Las planillas de cálculo pueden conectarse a la estructura dimensional y alimentar una **tabla pivot** con la información que extraen de los cubos.
- **Tablero de Control:** Los tableros de control se conectan a la estructura dimensional y generan **indicadores** que permiten una rápida visión del estado actual de las variables básicas y su relación con los objetivos de la empresa.
- **Desarrollos propios:** Soluciones o aplicaciones desarrolladas a medida, especialmente para la organización. Estas soluciones puede desarrollarlas el área de Sistemas de la empresa o un Proveedor externo, pero siempre en base a los requerimientos propios de la organización.
- **Software especializado:** Soluciones o aplicaciones creadas por empresas dedicadas principalmente al desarrollo de visualizadores de información orientada al análisis. Existe una gran variedad de herramientas con diversidad de prestaciones y costos, pudiendo ser tanto genéricas como orientadas a algún mercado en particular.
- **Reporteadores:** Herramientas especializadas en la construcción de informes que pueden conectarse a la estructura dimensional y generar reportes con la información que extraen de los cubos.

Existe una gran variedad de Herramientas de visualización de la información almacenada en una estructura multidimensional. Se debe estudiar cada conjunto necesidad – recurso para decidir cuál de ellas usar.

En general, los factores que influyen en la elección de una herramienta son:

- ✓ Tipo de consultas o análisis.
- ✓ Presupuesto.
- ✓ Valor del desarrollo o de las licencias.
- ✓ Usuario al que va destinada la herramienta.

- ✓ Otras herramientas existentes en la empresa.
- ✓ Capacidad de desarrollo de aplicaciones propias.

#### 2.8.5.4. La Tabla Pivotal

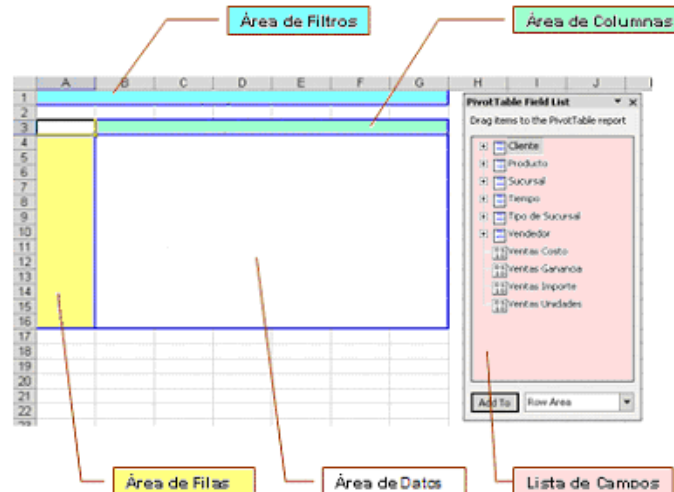
La tabla pivotal es una herramienta gráfica que permite a los usuarios explorar fácilmente las dimensiones y medidas del cubo. De esta manera el usuario puede construir sus propios informes.

La tabla pivotal se utiliza a través de una planilla de cálculo que se conecta al modelo multidimensional. Con ella se pueden realizar todas las operaciones que vimos en el punto

#### 2.8.5.5. Consultas

Una Tabla Pivotal consta de las siguientes áreas:

- **Área de Filtros:** En la parte superior de la tabla. Se puede incluir una o más dimensiones. Se puede filtrar la información seleccionando niveles en general o miembros en particular. Cuando se realizan selecciones múltiples dentro de una dimensión, éstas se relacionan mediante el operador OR. Si las selecciones se realizaron en varias dimensiones, se vinculan con el operador AND. En esta área se implementa exclusivamente la operación Filtrado; en base a las selecciones realizadas se forma un conjunto reducido de datos que cumplen con los valores elegidos.
- **Área de Filas:** A la izquierda, define qué dimensiones cruzan la tabla como filas. En esta zona se pueden arrastrar las dimensiones, se puede navegar por ellas y decidir qué niveles mostrar y el grado de apertura de la información. También se puede seleccionar qué información se muestra. En esta área se implementan las operaciones Drill-Up, Drill-Down y Filtrado.



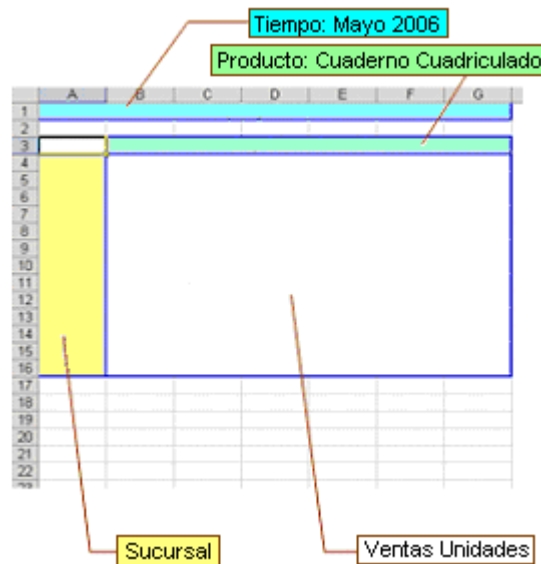
Fuente: (Microsoft, 2009)

Figura 26. Áreas consultas, tabla pivotal

- **Área de Columnas:** En la parte superior de la tabla debajo del área de filtros. Se define qué dimensiones cruzan la tabla como columnas. En esta zona se pueden arrastrar las dimensiones, se puede navegar por ellas y decidir qué niveles mostrar y el grado de apertura de la información. También se puede seleccionar qué información se muestra. En esta área se implementan las operaciones Drill-Up, Drill-Down y Filtrado.

- **Área de Datos:** En el centro de la tabla, se pueden incluir sólo medidas. Cuando arrastramos una medida tendremos el resultado de la intersección con las dimensiones que escogimos como filas y columnas, para el subconjunto que define el Filtrado.
- **Lista de Campos:** Contiene la lista de las dimensiones y las medidas del cubo.
- **Notas:**
  - Una dimensión puede estar en Filtros, Filas o Columnas, pero sólo en un área a la vez.
  - En general, en el área de Filtros no se muestra la selección realizada. Si se desea filtrar por un nivel o miembro y, a la vez mostrarlo, se lo debe incluir en Filas o Columnas.

En este ejemplo se quieren ver las unidades vendidas (Ventas Unidades), para el mes de Mayo de 2006, de Cuadernos cuadriculados, detalladas por Sucursal.



Fuente: (Microsoft, 2009)

Figura 27. Ejemplo tabla pivotada

### 2.8.5.6. El Tablero de Control

El Tablero de Control es una herramienta gráfica que le permite a los directivos concentrarse en indicadores fundamentales que tienen relación directa con los objetivos de negocio de la empresa. El Tablero de Control no es un repositorio de datos, básicamente es una herramienta que muestra indicadores relacionando los resultados esperados con los reales, es una manera de analizar la evolución del negocio.

Un Tablero de Control muestra, en pocos indicadores, datos trascendentes que extractan la naturaleza de la empresa y su porvenir. Estos indicadores deben mostrar la información en forma oportuna, sencilla e integrada, y ser claros y confiables.

Un Tablero de Control no garantiza el éxito de una empresa, debe comprometerse el esfuerzo necesario para su efectiva utilización y generar una transformación en la cultura de trabajo empresarial.

Finalmente, se debe tener perfectamente en claro que un Tablero de Control no gerencia ni gestiona; los indicadores le muestran los problemas a los directivos, pero el análisis de las causas y la forma de solucionarlos depende de las decisiones que ellos tomen. El Tablero de Control le indica a los directivos si la organización está cumpliendo con los objetivos o no, pero en ningún momento genera una solución automática. ¿Para qué sirve, entonces, el Tablero de Control? Básicamente, el Tablero de Control permite una rápida lectura del estado actual de las variables básicas y su relación con los objetivos de la empresa, alerta sobre la existencia de problemas actuales y facilita la visión de la evolución esperada, con lo cual ayuda a detectar los desvíos en los objetivos y tomar decisiones oportunas para corregirlos a tiempo.

## **2.9. SQL Server 2008 R2 BI**

### **2.9.1. BI con herramientas Microsoft**

#### **Server Management Studio**

SQL Server Management Studio es un entorno integrado para obtener acceso, configurar y administrar todos los componentes de SQL Server. Management Studio combina un amplio grupo de herramientas gráficas con cinco editores de script enriquecidos para ofrecer a desarrolladores y administradores con cualquier grado de experiencia acceso a SQL Server. Management Studio está diseñado para desarrollar y administrar objetos de base de datos y para administrar y configurar los objetos existentes de Analysis Services. Si está implementando una solución que utiliza servicios de bases de datos de SQL Server o si está administrando una solución existente que utiliza SQL Server, Analysis Services, Integration Services o Reporting Services, debe utilizar Management Studio. (Microsoft, 2009)

#### **Business Intelligence Development Studio**

Business Intelligence Development Studio es un entorno integrado para desarrollar construcciones de inteligencia empresarial, como cubos, orígenes de datos, informes y paquetes de Integration Services. BI Development Studio incluye plantillas de proyecto que proporcionan un contexto para desarrollar construcciones específicas. Por ejemplo, se puede optar por un proyecto de Analysis Services si el objetivo es crear una base de datos de Analysis Services que contenga cubos, dimensiones o modelos de minería de datos.

En BI Development Studio, es posible desarrollar proyectos que formen parte de una solución independiente de un servidor concreto. Por ejemplo, puede incluir un proyecto de Analysis Services, de Integration Services y de Reporting Services en la misma solución. Puede implementar los objetos en un

servidor de prueba para probarlos durante el desarrollo y, posteriormente, implementar el resultado de los proyectos en uno o más servidores de ensayo o de producción.

### **Componentes y herramientas de Reporting Services**

SQL Server 2008 Reporting Services proporciona componentes de proceso, herramientas e interfaces de programación para desarrollar, administrar y utilizar los informes enriquecidos.

### **Información general sobre herramientas y utilidades (Integration Services)**

SQL Server 2008 Integration Services (SSIS) proporciona herramientas de creación y administración completas para paquetes SSIS, como el Diseñador SSIS, el Generador de consultas, el Generador de expresiones y varias utilidades del símbolo del sistema.

### **Administrador de configuración de SQL Server**

El Administrador de configuración de SQL Server es una herramienta para administrar los servicios asociados a SQL Server, para configurar los protocolos de red utilizados por SQL Server y para administrar la configuración de conectividad de red de los equipos cliente de SQL Server.

### **Referencia del Analizador de SQL Server**

SQL Server Profiler es una herramienta que captura los eventos SQL Server de una instancia del motor de base de datos. Los eventos se guardan en un archivo de seguimiento que posteriormente se puede analizar o utilizar para reproducir una serie de pasos específicos cuando se intenta diagnosticar un problema.

### **Introducción al Asistente para la optimización de motor de base de datos**

El Asistente para la optimización de motor de base de datos de SQL Server ayuda a seleccionar y crear un conjunto óptimo de índices, vistas indizadas y particiones sin necesidad de conocer detalladamente la estructura de la base de datos ni el funcionamiento interno de SQL Server.

### **Utilidades del símbolo del sistema**

El motor de base de datos de SQL Server proporciona herramientas adicionales que se pueden ejecutar desde el símbolo del sistema.

## **2.9.2. SQL Server Integration Services**

Business Intelligence Development Studio incluye el proyecto Integration Services para desarrollar soluciones ETL. Este tipo de proyecto incluye las plantillas para paquetes, orígenes de datos y vistas de orígenes de datos, y proporciona las herramientas para trabajar con estos objetos. (Microsoft, 2009)

Es una plataforma para la creación de soluciones empresariales de transformaciones de datos e integración de datos. Integration Services sirve para resolver complejos problemas empresariales mediante la copia o descarga de archivos, el envío de mensajes de correo electrónico como respuesta a

eventos, la actualización de almacenamientos de datos, la limpieza y minería de datos, y la administración de objetos y datos de SQL Server. Los paquetes pueden funcionar en solitario o junto con otros paquetes para hacer frente a las complejas necesidades de la empresa. Integration Services puede extraer y transformar datos de diversos orígenes como archivos de datos XML, archivos planos y orígenes de datos relacionales y, después, cargar los datos en uno o varios destinos.

Integration Services contiene un variado conjunto de tareas y transformaciones integradas, herramientas para la creación de paquetes y el servicio Integration Services para ejecutar y administrar los paquetes. Las herramientas gráficas de Integration Services se pueden usar para crear soluciones sin escribir una sola línea de código. También se puede programar el amplio modelo de objetos de Integration Services para crear paquetes mediante programación y codificar tareas personalizadas y otros objetos de paquete.

### **2.9.3. SQL Server Analysis Services**

Business Intelligence Development Studio incluye el proyecto Analysis Services para desarrollar la funcionalidad de proceso analítico en línea (OLAP) y minería de datos para aplicaciones de Business Intelligence. Este tipo de proyecto incluye las plantillas para cubos, dimensiones, estructuras de minería de datos, orígenes de datos, vistas de orígenes de datos y roles, y proporciona las herramientas para trabajar con estos objetos. (Microsoft, 2009)

Proporciona un intervalo de soluciones para crear e implementar las bases de datos analíticas que se usan como respaldo para la toma de decisiones en las aplicaciones Excel, PerformancePoint, Reporting Services y otras de Business Intelligence. La base de una solución de Analysis Services es un modelo de datos semántico de Business Intelligence y una instancia de servidor que crea instancias, procesos, consultas y administra los objetos de ese modelo.

Los modelos se basan en los datos históricos que ya está recopilando en bases de datos transaccionales y otros almacenes de datos y se anotan con metadatos que permiten medir, tratar y comparar datos empresariales en consultas ad hoc o en informes personalizados. Una vez diseñado un modelo, se implementa en un servidor de Analysis Services como una base de datos, donde se pone a disposición de los usuarios autorizados que se conectan a él a través de Excel y de otras herramientas.

### **2.9.4. SQL Server Reporting Services**

Business Intelligence Development Studio incluye los proyectos de modelos de informe e Informes para desarrollar soluciones de informes. El tipo de proyecto Modelo de informe incluye las plantillas para modelos de informes, orígenes de datos y vistas de orígenes de datos, y proporciona las herramientas para trabajar con estos objetos. El proyecto Informe incluye las plantillas para trabajar con informes y orígenes de datos compartidos.



*Daniel Cuarán*

*2015*

*CAPITULO 3*



## **Capítulo III: Proceso De Desarrollo.**

### **3.1. Análisis**

Aquí se establece el alcance del proyecto de software, incluyendo una visión operacional y requisitos de calidad.

#### **3.1.1. Visión del Proyecto**

Administrar los requerimientos de las oficinas y departamentos de la Cooperativa de Ahorro y Crédito Atuntaqui Ltda. en relación a la adquisición de bienes y servicios vía online para mejorar los procesos en el tema de tiempos e incorporar el módulo de contratos y garantías.

#### **3.1.2. Requerimientos**

Se necesita crear y/o automatizar los siguientes procesos:

##### **Calificación de Proveedores**

- Página Web solicitud de calificación
- Proveedor accede a los Requisitos
- Proveedor Adjuntar documentos y envía
- Bandeja de entrada de Servicios Generales recibe la documentación
- Aprueba o niega
- Bandeja de entrada del Proveedor Calificado recibe requerimientos
- Envía cotizaciones se recibe en bandeja de entrada de Servicios Generales.

##### **Cotización y Adjudicación**

- Gerentes de oficina, Jefes de Agencia y Departamentales envían solicitud de adquisición
- Bandeja de entrada de Servicios Generales recibe la solicitud
- Remite a cada nivel de aprobación la solicitud para aprobación
- Si aprueba elabora las especificaciones o bases según el caso
- Sistema verifica a Proveedores Calificados y remite a los proveedores que consten con la actividad relacionada.
- Proveedores reciben solicitud de cotización en la bandeja de entrada
- Proveedores preparan información y envían las cotizaciones.
- Se receptan las cotizaciones
- Bandeja de entrada de Servicios Generales
- Servicios Generales elabora cuadro comparativo de costos
- Envía al nivel de adjudicación correspondiente
- Bandeja de entrada del nivel correspondiente recibe el cuadro comparativo con las cotizaciones
- Recibe, analiza y adjudica; o niega
- Si aprueba o niega se notifica a los proveedores la resolución tomada.
- Dependiendo del valor se envía para solicitar contrato.

### **Elaboración de Contrato**

- Servicios Generales envía requerimiento de elaboración de contrato a Jurídico
- Jurídico elabora contrato
- Carga el contrato en el sistema para revisión
- Bandeja de entrada del área solicitante, aprueba u observa
- Si tiene cambios se realizan se imprimen para la sumilla
- Remite el físico: 4 ejemplares a Servicios Generales para legalización
- Notificación de entrega a Proveedor y usuario solicitante
- Retorna el contrato a Servicios Generales
- Bandeja de entrada solicitante notificación de retorno del contrato
- Servicios Generales entrega 1 ejemplar a Jurídico
- Jurídico remite la garantía y copia simple a Tesorería.
- Tesorería recibe y custodia
- Bandeja de entrada de Tesorería.

### **Devolución de Garantías**

- Servicios Generales verifica la culminación del contrato
- Por el monto pide acta entrega recepción a Jurídico
- Bandeja de entrada de Jurídico elabora actas.
- Jurídico remite actas
- Se legalizan las actas
- Se realiza el pago a proveedores
- Se notifica a Tesorería la culminación del contrato
- Bandeja de entrada de Tesorería
- Devolución de Garantía

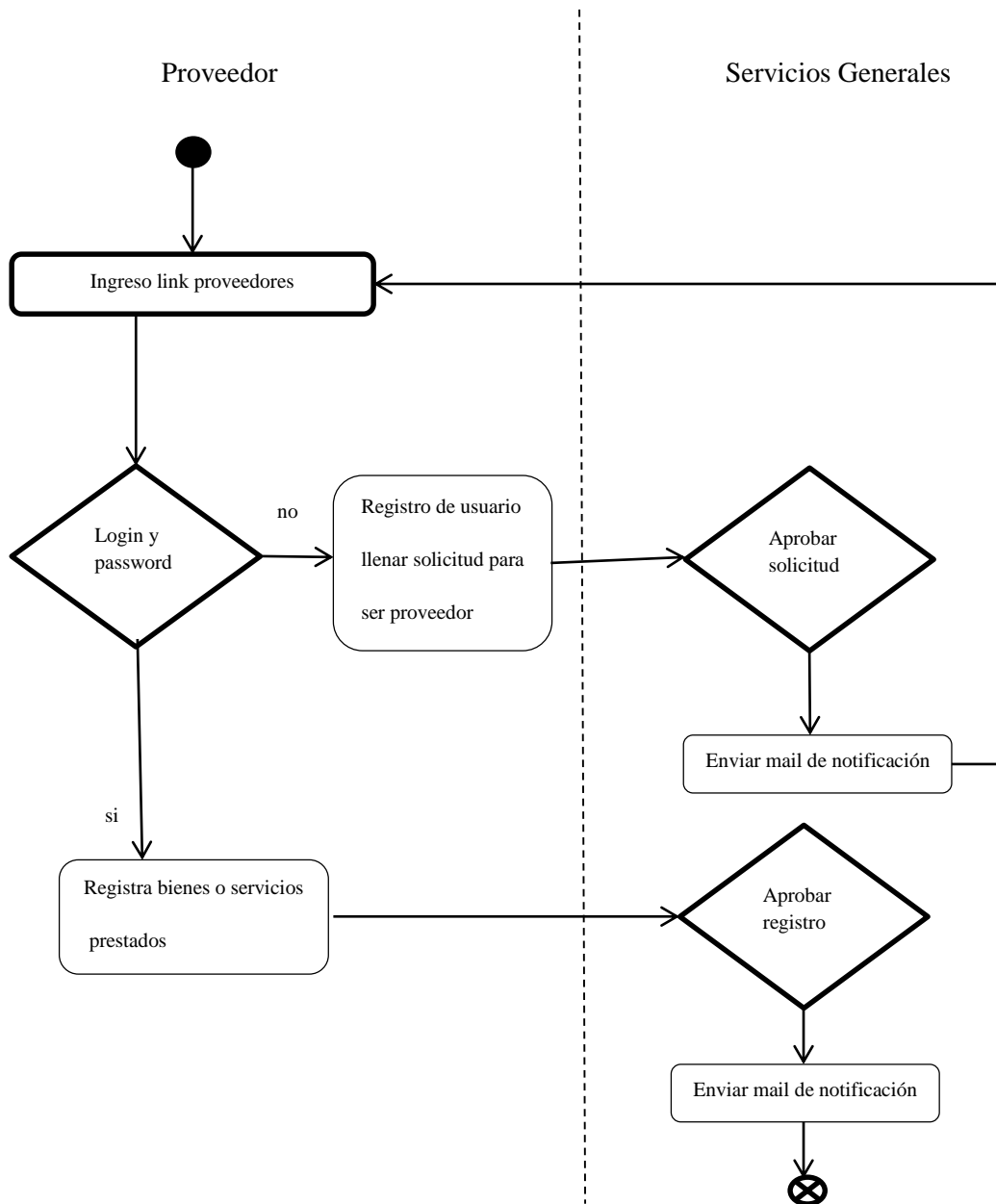
## **3.2. Diseño**

Elaborar los modelos técnicos y la arquitectura del software en base a los requerimientos realizados.

### 3.2.1. Diagramas de Actividades

En los diagramas de actividades se muestra un proceso de negocio o un proceso de software como un flujo de trabajo a través de una serie de acciones. Estas acciones las pueden llevar a cabo personas, componentes de software o equipos.

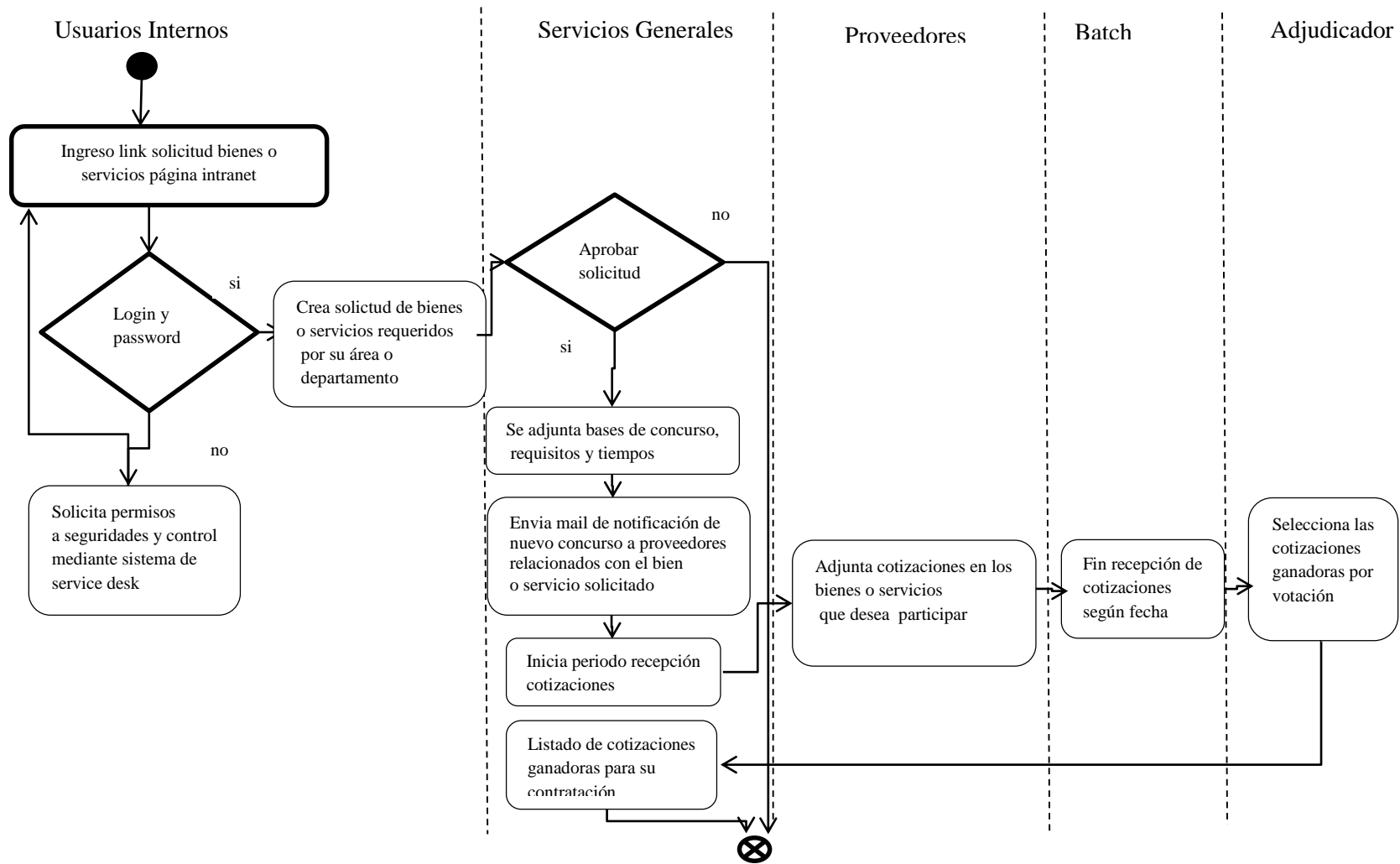
#### 3.2.1.1. Acciones de Calificación de Proveedores



Fuente: Propia

Figura 28. Calificación de proveedores.

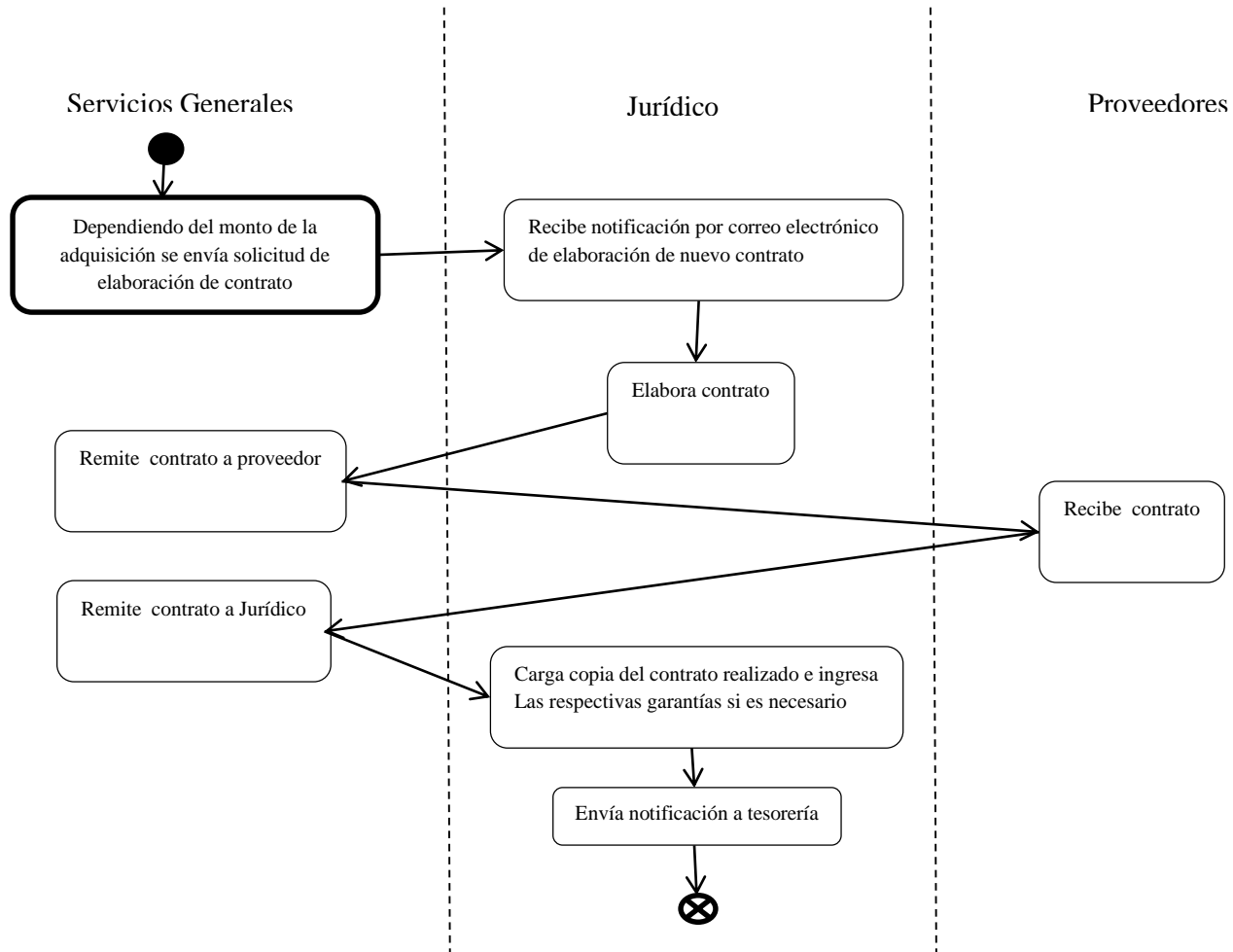
### 3.2.1.2. Acciones de Cotizaciones y adjudicaciones



Fuente: Propia

Figura 29. Cotizaciones y Adjudicaciones.

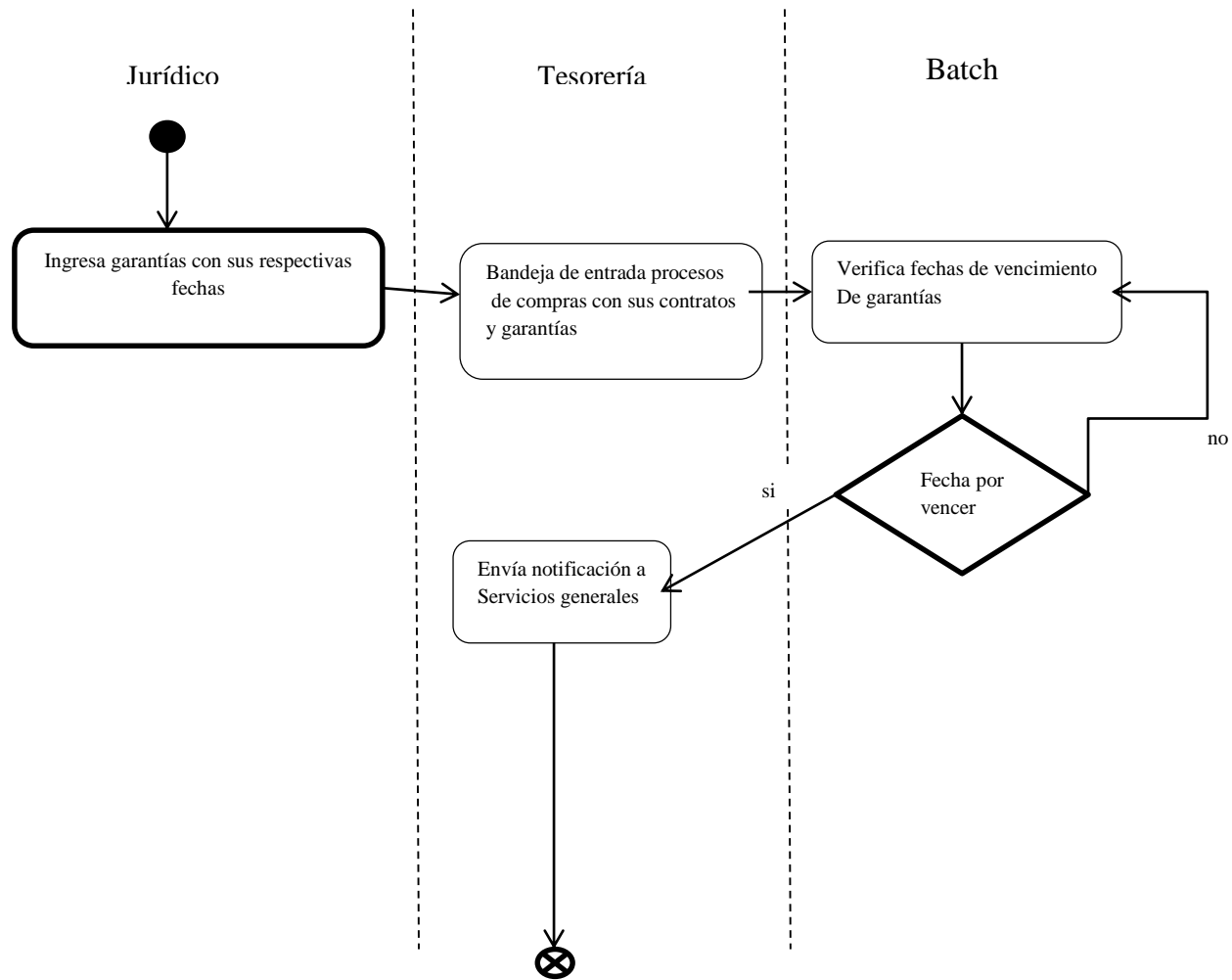
### 3.2.1.3. Acciones de Contratos



Fuente: Propia

Figura 30. Contratos.

### 3.2.1.4. Acciones deGarantias



Fuente: Propia

Figura 31. Garantías.





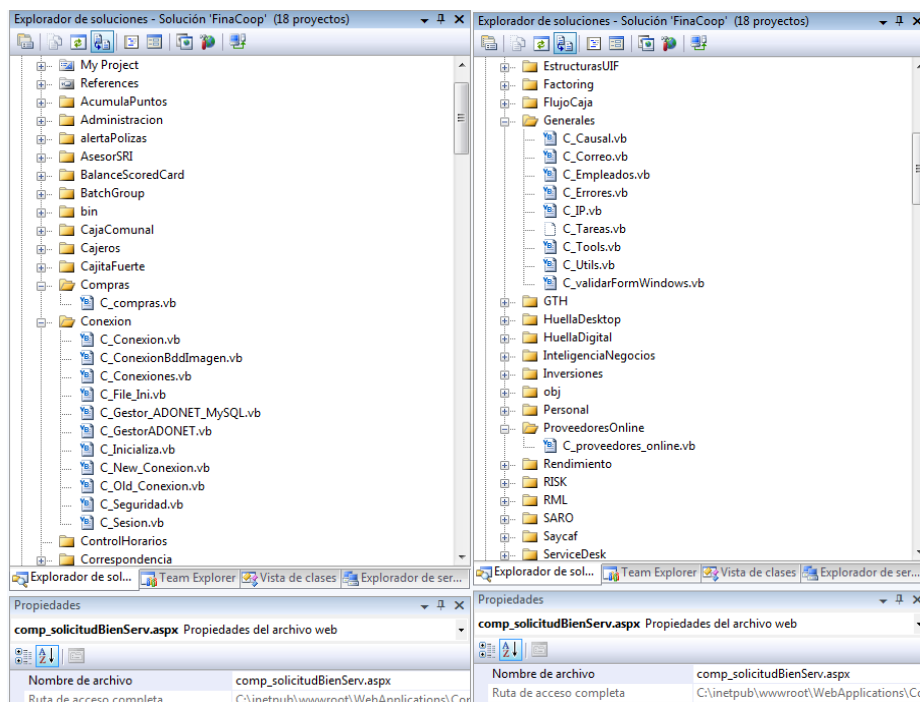
### 3.3.2. Diccionario de datos

Se adjuntan en la parte de anexos.

### 3.3.3. Clases

#### Proyecto Conexion\_Sybase

Este proyecto se diseñó como una biblioteca de clases (dll) mismo que contiene la lógica del negocio, así como también administra la conexión a la base de datos sybase.



Fuente: Propia

Figura 33. Proyecto Conexión.

#### Estructura de directorios

Directorios Principales:

- **Conexión**

En este directorio se mantendrán todas las clases que son utilizadas para manejar la conexión a la base de datos.

- **Compras**  
En este directorio se encuentra la clase de la aplicación web para nuestra intranet, aquí se encuentran los objetos que interactúan con nuestra base de datos.
- **Proveedores Online**  
En este directorio se encuentra la clase de la aplicación web que funcionará en internet, aquí se encuentran los objetos que interactúan con nuestra base de datos.
- **Generales**  
En este directorio se mantendrán todas las clases que representen a objetos generales que podrán ser utilizados por todos los módulos.
- **Directorios de módulo**  
Se deberá crear un directorio por cada módulo que sea desarrollado por ejemplo:  
*Administración* (módulo de administración del sistema)  
*Personal* (módulo de personal del sistema)  
Se crearán tantos directorios como módulos tenga el sistema. Dentro de este directorio se almacenarán directamente las clases que representen a beans, es decir clases que representen a los objetos directamente.
- **Persistencia**  
Este directorio se encontrará dentro de cada uno de los directorios que correspondan a los módulos del sistema, este directorio contendrá los objetos que administren la persistencia. Entendiendo como persistencia la acción de preservar un objeto de forma permanente (guardar), y a su vez recuperar información del objeto (leer) para que pueda ser utilizada nuevamente, es decir todas las acciones en las que intervengan los objetos con la base de datos.

## Restricciones

Únicamente las clases que representen a persistencias podrán llamar a la clase C\_Conexion.

Las clases que representen a Beans podrán ser utilizadas tanto por las clases de persistencia como por otros proyectos.

Las clases de persistencia podrán ser usadas por otras persistencias o por otros proyectos.

Las clases utilizadas en el sistema son:

**C\_compras():** Ésta es la clase principal del sistema que funciona en la intranet, aquí se encuentran todas las funciones que interactúan con los procedimientos almacenados.

**C\_proveedores\_online():** Ésta es la clase principal del sistema que funciona en internet, aquí se encuentran todas las funciones que interactúan con los procedimientos almacenados.

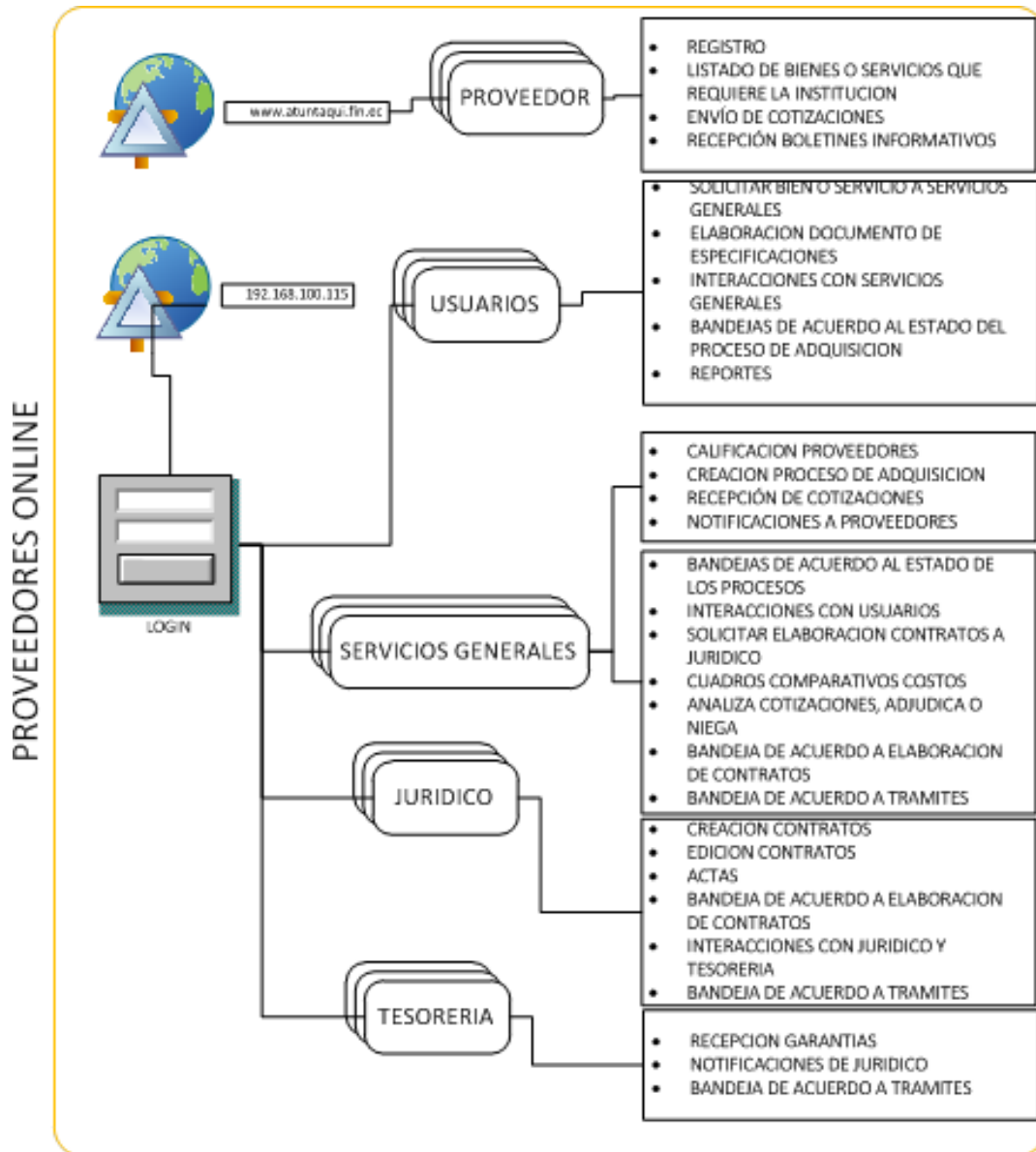
**C\_Correo():** En esta clase se encuentran las funciones para el envío de correos electrónicos

**C\_Empleados():** Ésta clase tiene funciones para obtener la información relacionada con los empleados que en este caso son los usuarios internos.

**C\_Seguridad():** En esta clase se tiene las funciones de seguridad, por ejemplo la función para encriptar o desencriptar contraseñas.

**C\_zeroPapeles():** Esta clase tiene las funciones para el sistema de cero papeles que en nuestro caso se las utiliza par verificar que el socio proveedor tenga los documentos actualizados.

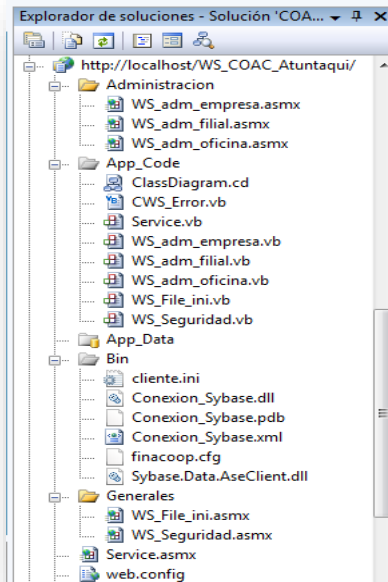
### 3.3.4. Módulos



Fuente: Propia

Figura 34. Esquema del proyecto.

## Proyecto WS\_COAC\_Atuntaqui



Fuente: Propia

Figura 35. Proyecto web services.

Este proyecto es una aplicación Web, que se encarga de proveer los servicios Web que devolverá datos y recibirá datos para que sean procesados por la capa de lógica del negocio (Conexion\_Sybase).

### Referencias

Este proyecto utiliza como referencia el proyecto **Conexion\_Sybase**, para acceder a la capa de la lógica del negocio.

### Estructura de directorios

Directorios Principales:

- **App\_Code**  
En este directorio se guardan los archivos del código fuente de los WebServices.
- **Bin**  
En este directorio se encuentran los archivos de configuración para la conexión a la base de datos así como también el archivo de contraseña de la base de datos.
- **Directorios de módulo**  
Se deberá crear un directorio por cada módulo que sea desarrollado, manteniendo la misma estructura que en el proyecto Conexion\_Sybase, en donde están las definiciones de los servicios web.  
*Administración* (módulo de administración del sistema)
- **General**

Se mantienen los servicios web que tienen funciones generales.

### Archivo de configuración

**web.config:** en este archivo en formato XML están seteadas las siguientes variables:

```
<appSettings>
  <add key="archivo_ini" value="cliente.ini" />
  <add key="archivo_cfg" value="finacoop.cfg" />
</appSettings>
```

La variable **archivo\_ini** debe tener en valor el nombre del archivo con las configuraciones generales de conexión a la base de datos.

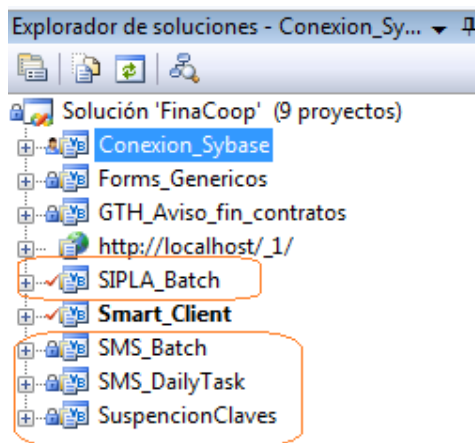
La variable **archivo\_cfg** debe tener en valor el nombre del archivo que contiene la clave encriptada del servidor de base de datos.

Estas variables son utilizadas por el proyecto **Conexión\_Sybase**, para la conexión a la base de datos.

Los web services creados para este proyecto son:

**WS\_Compras.asmx:** Se creó éste servicio web para poder implementar el batch que envía invitaciones a los proveedores para participar en los concursos.

### Proyectos en Consola BATCH.NET



Fuente: Propia

Figura 36. Proyecto consola batch.

Este tipo de proyectos son programas sencillos y muy livianos realizados como proyecto de consola en VB.NET, bajo todo el esquema de la infraestructura actual que realizan a determinada hora programada mediante el programador de tareas de Windows Server, procesos Batch asíncronos tales como el envío de SMS diarios a los socios a una hora determinada, como por ejemplo el envío de alertas de forma

automática en el sistema de Prevención de Lavado de Activos, procesos como el de bloqueo/desbloqueo de claves, etc.

Este proyecto es una aplicación tipo Consola, el cual consume los servicios web que provee el proyecto **WS\_COAC\_Atuntaqui**.

## Referencias

Este proyecto hace referencias a los servicios web que provee el proyecto **WS\_COAC\_Atuntaqui**.

## Estructura de directorios

Directorios principales:

- **Web\_References**

En este directorio se guardarán todas las referencias web tomadas desde **WS\_...** que son utilizadas por el proyecto.

## Archivo de configuración

**app.config** Archivo de configuración en formato XML, en el cual se setean las siguientes variables:

```
<applicationSettings>
  <Smart_Client.My.MySettings>
    <setting name="Smart_Client_WS_adm_filial_WS_adm_filial" serializeAs="String">
      <value>http://localhost/WS_COAC_Atuntaqui/Administracion/WS_adm_filial.asmx</value>
    </setting>
    <setting name="Smart_Client_WS_adm_oficina_WS_adm_oficina" serializeAs="String">
      <value>http://localhost/WS_COAC_Atuntaqui/Administracion/WS_adm_oficina.asmx</value>
    </setting>
    <setting name="Smart_Client_WS_File_ini_WS_File_ini" serializeAs="String">
      <value>http://localhost/WS_COAC_Atuntaqui/Generales/WS_File_ini.asmx</value>
    </setting>
    <setting name="Smart_Client_WS_Seguridad_WS_Seguridad" serializeAs="String">
      <value>http://localhost/WS_COAC_Atuntaqui/Generales/WS_Seguridad.asmx</value>
    </setting>
    <setting name="Smart_Client_WsAdm_empresa_WS_adm_empresa" serializeAs="String">
      <value>http://localhost/WS_COAC_Atuntaqui/Administracion/WS_adm_empresa.asmx</value>
    </setting>
  </Smart_Client.My.MySettings>
</applicationSettings>
<appSettings>
  <add key="url_ws" value="http://localhost/WS_COAC_Atuntaqui/" />
</appSettings>
```

Dentro del siguiente bloque, se registran todas las referencias Web de los servicios web que consume la aplicación.

```
<applicationSettings>
  <Smart_Client.My.MySettings>
    .....
    .....
    .....
  </Smart_Client.My.MySettings>
</applicationSettings>
```

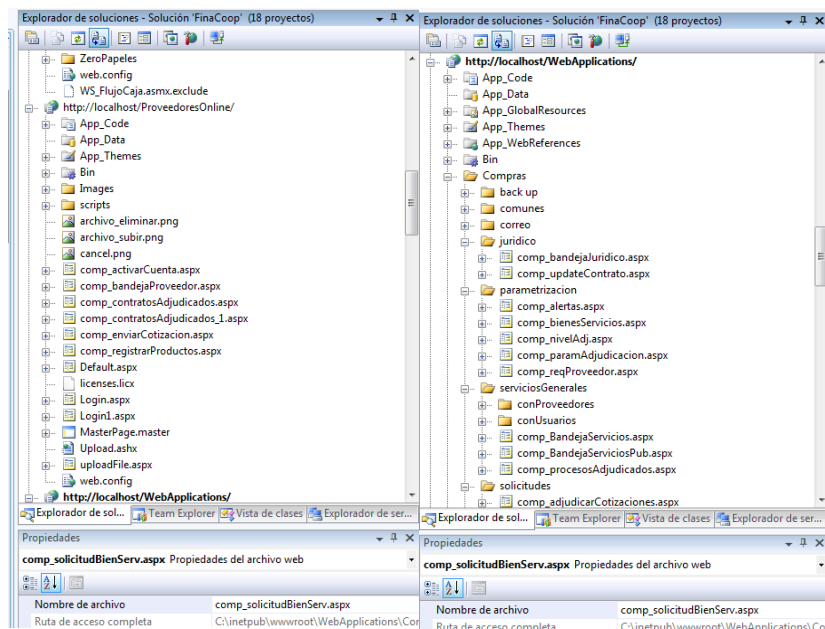
En la variable url\_ws, se guarda la dirección URL, del servidor del cual se consumirán los servicios web.

Estás variables se crearon para que sea posible parametrizar el servidor del cual las aplicaciones consuman los servicios web.

Los proyectos Batch para éste sistema son:

**GroupingBatch:** Se encarga de obtener el listado de productos que están en proceso de recibir cotizaciones y obtiene el listado de proveedores que cumplen los requisitos y procede a enviar una invitación al proveedor para participar en el concurso.

## Proyectos en ASP.NET



Fuente: Propia

Figura 37. Web applications.

En los proyectos de aplicación web vamos a tener 2:

## ProveedoresOnline

En este proyecto se tiene toda la aplicación que va a estar funcionando en internet, en estas aplicación vamos tener el módulo de proveedores, aquí está el registro del proveedor, bandejas del proveedor, listado de bienes y servicios que brinda el proveedor, etc.

## WebApplications/Compras

En este proyecto se tiene toda la aplicación que va a estar funcionando en la intranet, en estas páginas vamos tener todos los módulos de usuarios, servicios generales, jurídico y tesorería.

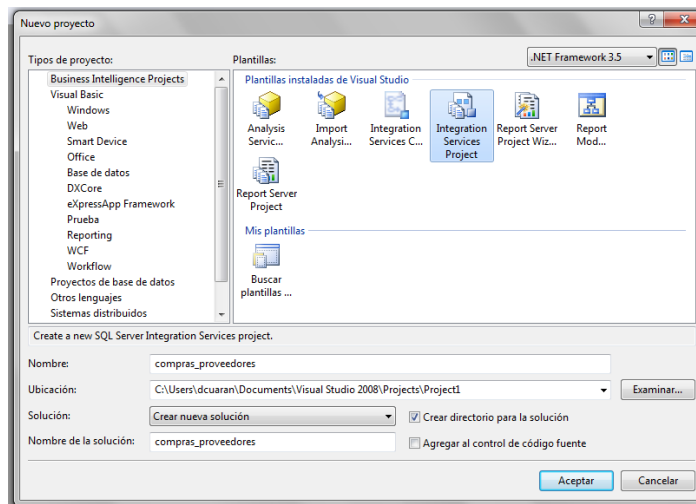
### 3.3.5. Pruebas

Se llena el formulario GTI-DES-SOF-03-FR-02 con los participantes, éste formulario se encuentra en anexos.

### 3.3.6. Creación Data mart

#### 3.3.6.1. Procesos de extracción, transformación y carga (ETL).

Lo primero que debemos hacer es la extracción de datos, para la creación de nuestro Data Mart se utiliza SQL Server Business Intelligence Development Studio, como el origen de datos es Sybase 15.7 primero vamos a crear un proyecto de BI para extracción de datos Integration Services(SSIS) como se muestra en la figura siguiente:

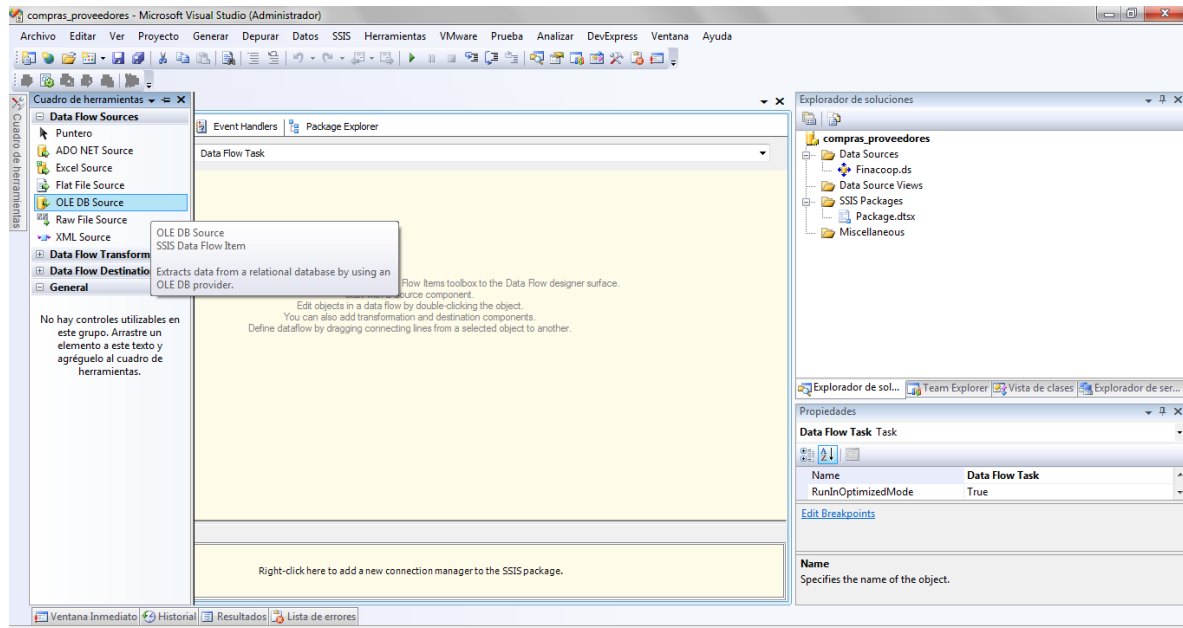


Fuente: Propia

Figura 38. Creación SSIS

Escogemos OLE DB Source para conectarnos a Sybase



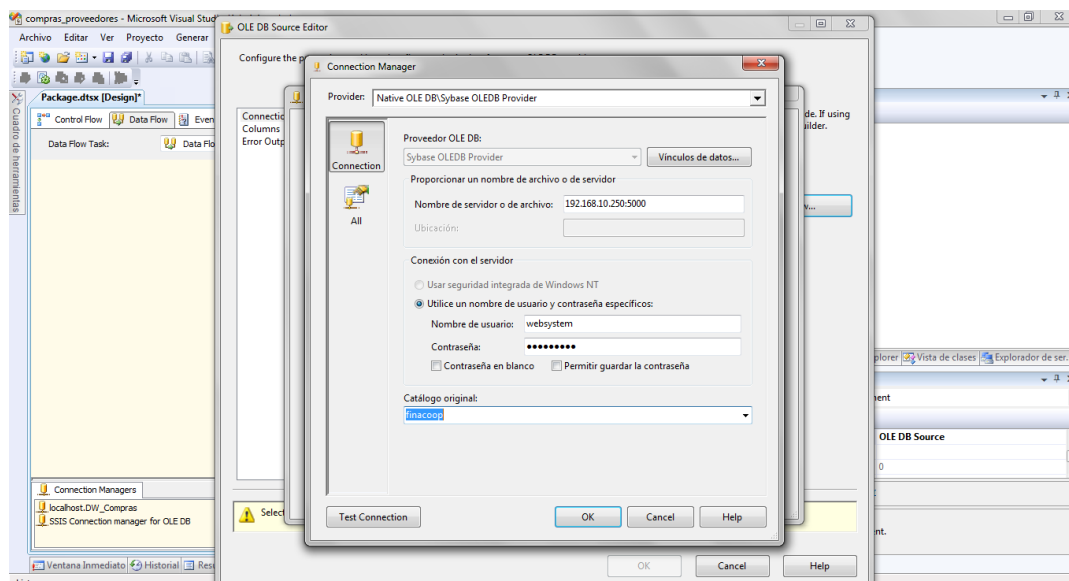


Fuente: Propia

Figura 39. Creando OLE DB Source

Desde la versión 15.7 de Sybase ya se puede utilizar las herramientas de B.I. de Microsoft.

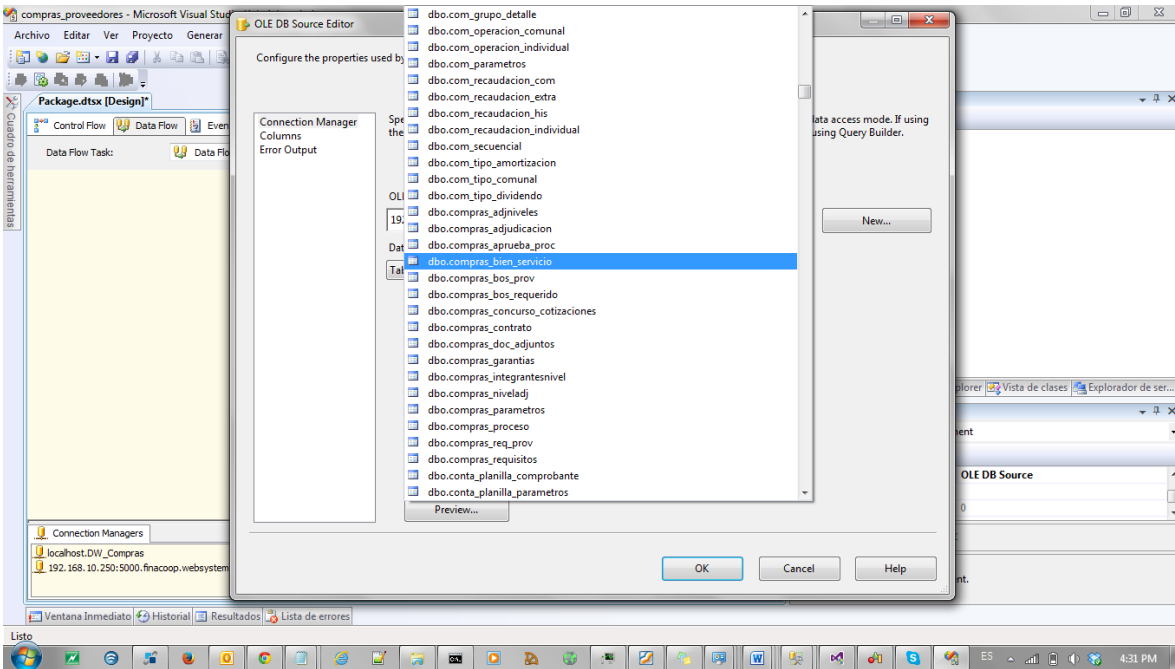
El siguiente paso es escoger nuestro vínculo de datos OLE DB, ponemos el nombre del servidor o la ip más el puerto, nombre de usuario, contraseña y escogemos la base de datos de donde queremos hacer la extracción.



Fuente: Propia

Figura 40. Seleccionando proveedor en connection manager

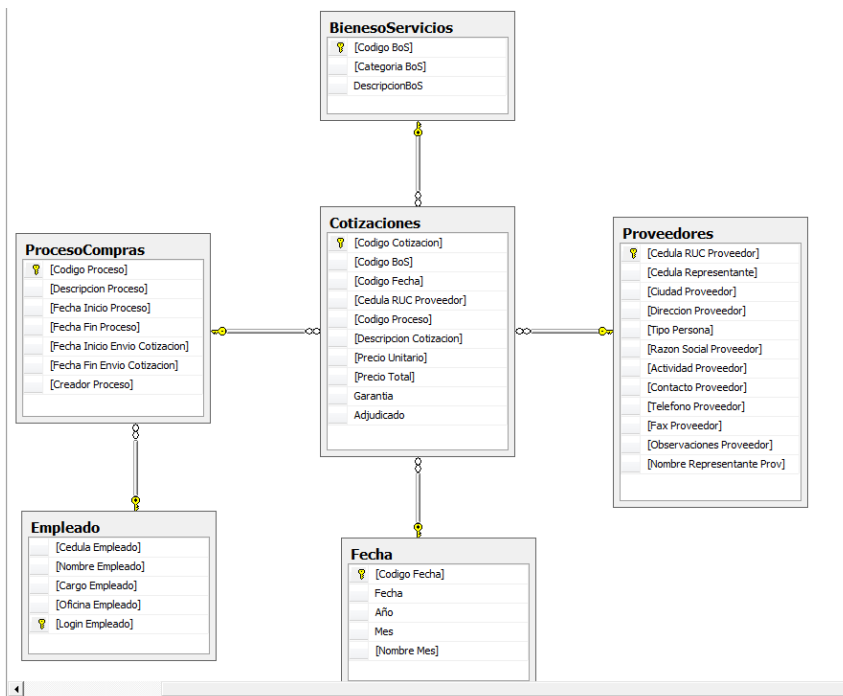
Seleccionamos la tabla u otro objeto o sql de donde obtendremos nuestros datos y construimos nuestro SSIS.



Fuente: Propia

Figura 41. Seleccionando tablas para extracción de datos

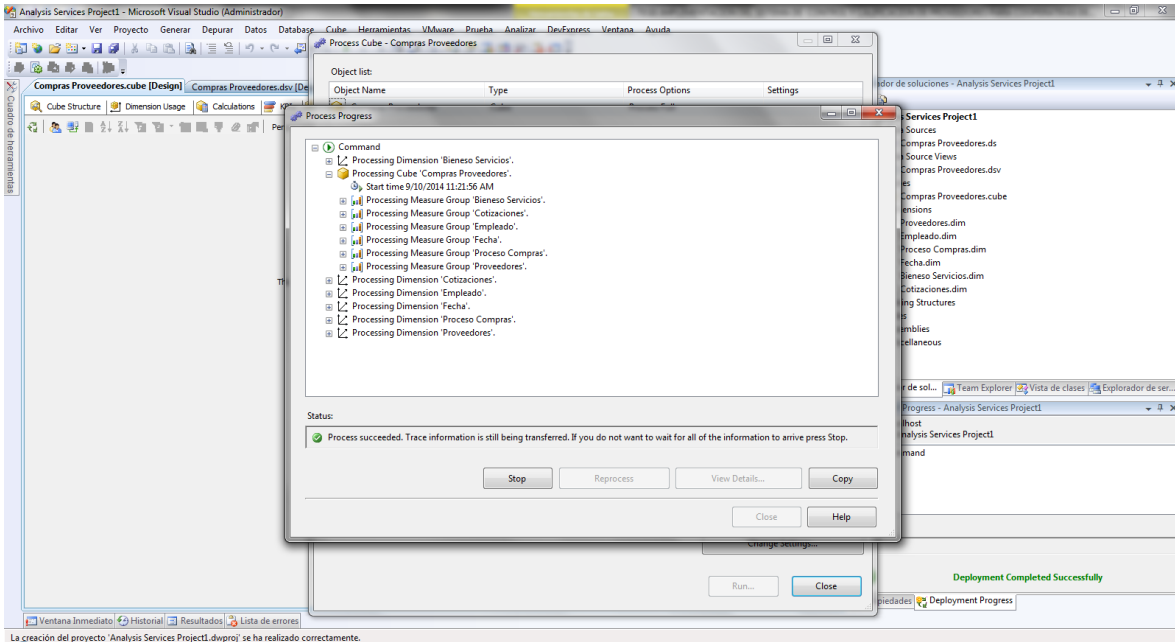
### 3.3.6.2. Modelo Dimensional



Fuente: Propia

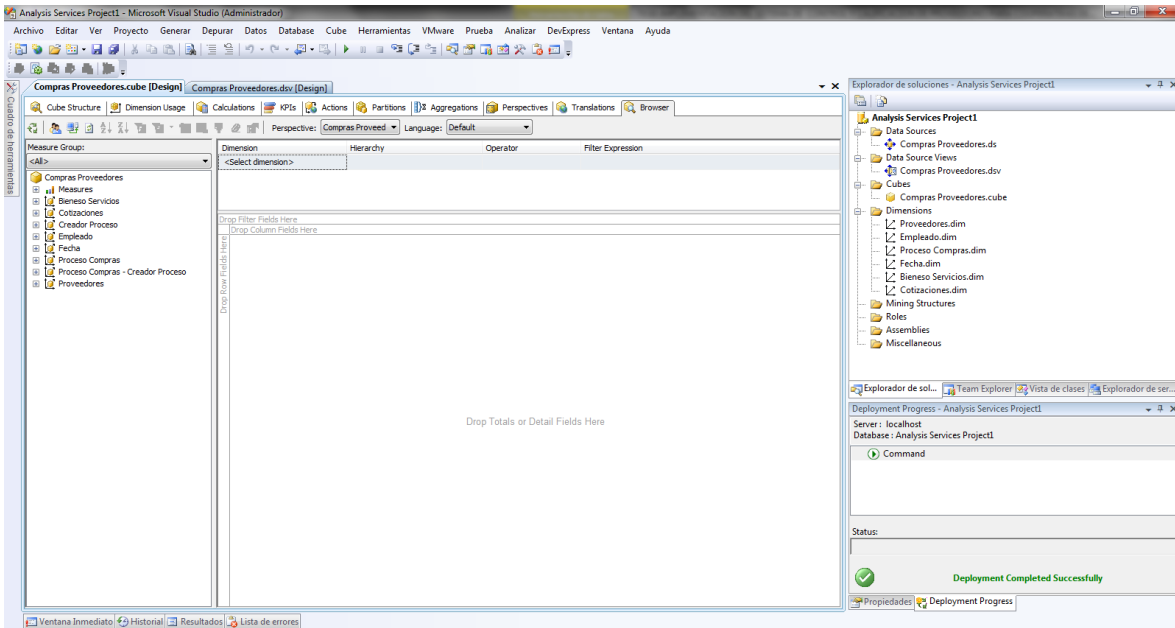
Figura 42. Modelo Dimensional para el data mart.





Fuente: Propia

Figura 45. Generando el data mart.



Fuente: Propia

Figura 46. Data mart construido

### **3.4. Paso a producción**

Instalar el sistema y todos sus componentes en ambiente de producción.

En la parte de anexos se adjunta lo siguiente:

- Formulario de Notas de Actualizaciones
- Formulario de pruebas de aceptación GTI-DES-SOF-03-FR-05.

### **3.5. Incidentes en Sistemas en Producción**

Solucionar el incidente de forma que no se vuelva a presentar en ninguno de los sistemas que ya se encuentran en producción

Hasta el momento no se han encontrado incidentes, en caso de haberlos se debe llenar el formulario.

### **3.6. Respaldo Programas de Desarrollo**

El Analista Programador confirmará si las últimas versiones de programas y documentación están en el servidor de desarrollo; en caso que no estén las últimas versiones de los programas, copiar (sincronizar) las copias locales del equipo al servidor de desarrollo.

Administrador de desarrollo depurará los directorios de los programas y documentación del servidor de desarrollo, y a su vez copiará los archivos de los programas del servidor de desarrollo a DVD y realizar la codificación del DVD usando el siguiente estándar:

RESP-DES-mm/dd/yyyy

Administrador de Desarrollo entregará DVD para almacenamiento a Jefe de Sistemas y este a su vez almacenará el DVD en un lugar seguro.

Esto se realizará con el formulario GTI-DES-SOF-03-FR-05

*Daniel Cuarán*

2015

*CAPITULO 4*

The background of the page is a complex, abstract composition of overlapping geometric shapes and lines. The color palette is vibrant, featuring shades of red, blue, purple, and orange. The shapes include triangles, polygons, and circles, some of which are semi-transparent, creating a layered effect. Thin white lines connect various points across the composition, suggesting a network or a path. The overall aesthetic is modern and dynamic, with a sense of movement and depth.

## Capítulo IV: Conclusiones y Recomendaciones.

### 4.1. Conclusiones

- ✓ La metodología utilizada permite a los participantes llegar a desarrollar el software de acuerdo a sus necesidades debido a la participación continua en cada fase de desarrollo.
- ✓ La arquitectura utilizada permite generar funciones que pueden ser utilizadas nuevamente en algún otro software relacionado.
- ✓ El uso de las herramientas Developer Express facilita el desarrollo de cada uno de los formularios debido a su entorno amigable y a las mejoras que tiene en cuanto a las grillas para manipular los datos.
- ✓ El uso de la herramienta Aqua Data Studio ayuda bastante en cuanto al desarrollo de procedimientos almacenados complejos gracias a su depurador.
- ✓ El software construido con esta arquitectura y herramientas tiene un buen rendimiento ya que accede a la base de datos sólo cuando es necesario y todas las operaciones de consulta, inserción o modificación se ejecutan en el motor de la base de datos más no en el frontal.
- ✓ En una entidad financiera, sistemas de BI son una gran herramienta para la evolución y el crecimiento del negocio y se deben diseñar de tal forma que puedan seguir esa evolución y crecimiento.
- ✓ Los sistemas de BI influyen en gran magnitud en la institución, no son una facilidad de un sector, los sistemas de BI son válidos para cualquier proceso en el que deban tomarse decisiones, no son una herramienta del área de Sistemas para mantener cautivos a los usuarios, por el contrario, con un sistema de éstos los usuarios consiguen más independencia al poder realizar las consultas en forma intuitiva y flexible.

## 4.2. Recomendaciones

- ✓ La metodología utilizada por la institución genera un proceso muy burocrático y causa demora en poner los sistemas en producción, por lo que sería bueno optimizar la etapa de paso a producción.
- ✓ En cuanto al proceso de desarrollo, se debe mejorar la rapidez en cuanto a la entrega de los requerimientos por parte de los usuarios.
- ✓ La arquitectura está acorde a las necesidades de la institución pero en desarrollo de sistemas la tecnología avanza a pasos agigantados y se debería actualizar a una más moderna como wpf.
- ✓ Debido a la poca documentación del conjunto de herramientas de Developer Express y complejidad de éstas, se debe realizar una buena documentación sobre el uso de cada componente por parte de los Analistas Programadores o Administrador de Desarrollo.
- ✓ Usar procedimientos almacenados es una buena práctica para tener un buen rendimiento de las aplicaciones y ayuda al uso de las grillas del developer express.
- ✓ Se debería planificar la creación de un data mart para las diferentes áreas de la Institución.



*Daniel Cuarán*

*2015*

*CAPITULO 5*



## **Capítulo V: Lista De Anexos.**

**5.1. Diccionario de Datos**

**5.2. Manual técnico**

**5.3. Manual de usuario**

## GLOSARIO

**Data mart:** almacén de datos de un hecho en particular, es una base de datos departamental, especializada en el almacenamiento de los datos de un área de negocio específica.

**RUP (Rational Unified Process):** constituye la metodología estándar más utilizada para el análisis, diseño, implementación y documentación de sistemas orientados a objetos.

**Refactoring:** técnica de la ingeniería de software para reestructurar un código fuente.

**Bugs:** es un error o fallo en un programa de computador o sistema de software.

**Flujograma:** es una muestra visual de una línea de pasos de acciones que implican un proceso determinado.

**Service desk:** Sistema informático de soporte técnico y punto único de contacto para los usuarios finales que necesitan ayuda.

**Front End:** es la parte del software que interactúa con el o los usuarios.

**Framework:** es una estructura conceptual y tecnológica de soporte definido, normalmente con artefactos o módulos de software concretos, que puede servir de base para la organización y desarrollo de software.

**XML: eXtensible Markup Language ('lenguaje de marcas extensible'),** es un lenguaje de marcas desarrollado por el World Wide Web Consortium(W3C) utilizado para almacenar datos en forma legible

**Common Language Runtime:** o CLR es un entorno de ejecución para los códigos de los programas que corren sobre la plataforma Microsoft .NET. El CLR es el encargado de compilar una forma de código intermedio llamada Common Intermediate Language (CIL, anteriormente conocido como MSIL, por Microsoft Intermediate Language), al código de máquina nativo.

**API:** Interfaz de programación de aplicaciones.

**LINQ: Language Integrated Query (LINQ,** pronunciado “link”) es un componente de la plataforma Microsoft .NET que agrega capacidades de consulta a datos de manera nativa a los lenguajes .NET, si bien existen implementaciones para Java, PHP, JavaScript y ActionScript.

**MSIL:** es el lenguaje de programación legible por humanos de más bajo nivel en el Common Language Infrastructure y en el .NET Framework. Los lenguajes del .NET Framework compilan a CIL. CIL es un lenguaje ensamblador orientado a objetos, y está basado en pilas. Es ejecutado por una máquina virtual. Los lenguajes .NET principales son C#, Visual Basic .NET, C++/CLI, y J#.

**Recolector de elementos no utilizados:** administra la asignación y liberación de la memoria de la aplicación

**WPF: Windows Presentation Foundation,** permite el desarrollo de interfaces de interacción en Windows tomando características de aplicaciones Windows y de aplicaciones web.

**Shell:** programas que proveen una interfaz de usuario para acceder a los servicios del sistema operativo.

**Herencia:** el mecanismo más utilizado para alcanzar algunos de los objetivos más preciados en el desarrollo de software como lo son la reutilización y la extensibilidad.

**API:** Interfaz de programación de aplicaciones, es el conjunto de funciones y procedimientos que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

**SharePoint:** es una plataforma de colaboración empresarial, formada por productos y elementos de software.

**WYSIWYG** es el acrónimo de *What You See Is What You Get* (en español, "lo que ves es lo que obtienes"). Se aplica a los procesadores de texto y otros editores de texto con formato (como los editores de HTML) que permiten escribir un documento viendo directamente el resultado final.

**AJAX:** acrónimo de Asynchronous JavaScript And XML (JavaScript asíncrono y XML), es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (Rich Internet Applications). Estas aplicaciones se ejecutan en el cliente.

**Controles data-aware:** Para navegación y manipulación de la bdd.

**Hunspell:** es un corrector ortográfico y un analizador morfológico diseñado para idiomas con una morfología rica, compleja formación de palabras compuestas o con una codificación de caracteres distinta del ASCII de 8 bits, diseñado en un principio para el idioma húngaro.

**ISpell:** correctores de ortografía en paquetes ofimáticos y en programas de correo.

**Shaping:** Intenta controlar el tráfico en redes de ordenadores para así lograr optimizar o garantizar el rendimiento, baja latencia, y/o un ancho de banda determinado retrasando paquetes.

**PKCS:** En criptografía, PKCS (Public-Key Cryptography Standards) se refiere a un grupo de estándares de criptografía de clave pública concebidos y publicados por los laboratorios de RSA.

**COBOL:** El lenguaje COBOL (acrónimo de COmmon Business-Oriented Language, Lenguaje Común Orientado a Negocios) fue creado en el año 1959 con el objetivo de crear un lenguaje de programación universal que pudiera ser usado en cualquier ordenador.

**Motif:** es una biblioteca para la creación de entornos gráficos bajo X Window System en sistemas Unix.

**JAR:** el comando jar permite generar, observar y descomprimir archivos JAR.

## BIBLIOGRAFIA

- Fouche, G. (April 1, 2011). *Foundations of SQL Server 2008 R2 Business Intelligence*. Apress.
  
- Haidar, B. (November 10, 2008). *Professional ASP.NET 3.5 Security, Membership, and Role Management with C# and VB*. Wrox.
  
- Kimmel, P. T. (November 16, 2009). *Professional DevExpress ASP.NET Controls*. Wrox.
  
- Varallo, V. (February 3, 2009). *ASP.NET 3.5 Enterprise Application Development with Visual Studio 2008: Problem Design Solution*. Wrox .

## LINCOGRAFIA

- DevExpress. (2009). *DevExpress*. Obtenido de <https://documentation.devexpress.com/>
  
- MICROSOFT. (2008). *Información general acerca del framework*. Obtenido de [http://msdn.microsoft.com/es-es/library/zw4w595w\(v=vs.110\).aspx](http://msdn.microsoft.com/es-es/library/zw4w595w(v=vs.110).aspx)
  
- Microsoft. (2009). *Academia BI*. Obtenido de <http://www.microsoft.com/latam/technet/bi/default.msp>
  
- Sybase. (s.f.). *Manual utilidades*. Obtenido de [download.sybase.com/pdfdocs/asg1250s/util.pdf](http://download.sybase.com/pdfdocs/asg1250s/util.pdf)
  
- Wikipedia. (s.f.). *Wikipedia*. Obtenido de [http://es.wikipedia.org/wiki/Aqua\\_Data\\_Studio](http://es.wikipedia.org/wiki/Aqua_Data_Studio)

