



# TESINA DE LICENCIATURA

**Título:** “Una herramienta para el Planeamiento Estratégico y la Evaluación del Rendimiento Empresarial”

**Autor:** Pablo Pesce

**Director:** Dra. Claudia Pons

**Carrera:** Licenciatura en Informática

## Resumen

*En los últimos años ha surgido un creciente interés por incorporar la tecnología de las ciencias de la computación aplicada a los ámbitos de estrategia empresarial. Sin embargo estas herramientas difícilmente llegan a implementarse tanto por la falta de clarificaciones en el área así como también debido a la falta de arquitecturas de integración que contemplen un cierto número de requerimientos imprescindibles a la hora de trabajar en estos contextos. En esta tesina se clarifican los conceptos del área de la Gestión de Rendimiento Empresarial, aportando la visión de su estado presente y necesidades. Se enumeran los principales problemas encontrados en las arquitecturas de software modernas que hacen difícil la implementación de estas técnicas, incluyendo un análisis del soporte actual existente de software. Además, se presentan las principales tecnologías basadas en la plataforma eclipse, capaces de ofrecer una sólida implementación para herramientas en esta área. Como resultado se propondrá los fundamentos de una nueva arquitectura denominada “Arquitectura Tecnológica Integrada Dinámica” (ATID) la cual se considera una sólida base para la rápida implementación de soluciones de software de código abierto que aporten el dinamismo y la extensibilidad necesaria para satisfacer las crecientes demandas y necesidades de los usuarios del área*

## Palabras Claves

*BI, CPM, CMI, Balance Scorecard, eclipse, jazz, arquitectura, SOA*

## Trabajos Realizados

*Entre los trabajos realizados se destacan:*

- Análisis del área de “Gestión de Rendimiento Empresarial” y en particular de la técnica de “Balanced Scorecard” exponiendo un marco conceptual.
- Análisis comparativo de herramientas existentes exponiendo sus características y falencias.
- Análisis de tecnologías y tendencias de evolución de soluciones basadas en la plataforma eclipse.
- Planteo y desarrollo de una nueva arquitectura de software basada en eclipse con la extensibilidad y dinamismo requerido.
- Implementación de las principales componentes.

## Conclusiones

*Para lograr el éxito de aplicación de técnicas avanzadas para la optimización del rendimiento empresarial se requieren de arquitecturas de software que soporten la compleja dinámica del área, sin descuidar aspectos importantes como la extensibilidad y el dinamismo. La arquitectura ATID creada e implementada en esta tesina abre el camino para el desarrollo de soluciones basadas en la plataforma de código abierto eclipse, permitiendo crear avanzadas configuraciones que soporten las necesidades particulares de optimización de cada empresa.*

## Trabajos Futuros

*Entre los trabajos futuros es posible mencionar:*

- Formalización del marco conceptual y metodologías presentadas para el área.
- Generalización de GMF mediante transformaciones que permitan la generación de editores gráficos que se ejecuten dentro del navegador web interactuando con los servicios de la ATID.
- Estudio y generalización de CDO3 para su uso en la implementación del repositorio de la ATID.
- Estudio de la generación automática de las distintas configuraciones permitidas por la ATID.
- Desarrollo de los componentes extendidos de la ATID



Universidad Nacional de La Plata  
Facultad de Informática  
Licenciatura en Informática

**”Una herramienta para el Planeamiento Estratégico y la Evaluación del Rendimiento Empresarial”**

**Trabajo de Grado.**

**Alumno:**

Pablo Luis Pesce

**Director de Tesis:**

Dra. Claudia Pons

La Plata

Argentina

Diciembre 2010

## Resumen

En los últimos años ha surgido un creciente interés por incorporar la tecnología de las ciencias de la computación aplicada a los ámbitos de estrategia empresarial. Sin embargo estas herramientas difícilmente llegan a implementarse tanto por la falta de clarificaciones en el área así como también debido a la falta de arquitecturas de integración que contemplen un cierto número de requerimientos imprescindibles a la hora de trabajar en estos contextos. En esta tesina se clarifican los conceptos del área de la Gestión de Rendimiento Empresarial, aportando la visión de su estado presente y necesidades. Se enumeran los principales problemas encontrados en las arquitecturas de software modernas que hacen difícil la implementación de estas técnicas, incluyendo un análisis del soporte actual existente de software. Además, se presentan las principales tecnologías basadas en la plataforma eclipse, capaces de ofrecer una sólida implementación para herramientas en esta área. Como resultado se propondrá los fundamentos de una nueva arquitectura denominada “Arquitectura Tecnológica Integrada Dinámica” (ATID) la cual se considera una sólida base para la rápida implementación de soluciones de software de código abierto que aporten el dinamismo y la extensibilidad necesaria para satisfacer las crecientes demandas y necesidades de los usuarios del área.

**Palabras Clave:** BI, CPM, CMI, Balance Scorecard, eclipse, jazz, arquitectura, SOA

## **Agradecimientos**

A mi familia, por el apoyo que desde siempre me brindaron a lo largo de la carrera.  
A Claudia Pons por su apoyo incondicional, predisposición y los esfuerzos realizados  
para que pudiese lograr terminar mi carrera profesional.  
A Valeria por acompañarme en todo este proceso, por su paciencia, comprensión  
y constante estímulo durante todo este proceso.  
Gracias a mis amigos y compañeros, por el apoyo que en su momento cada uno ha  
sabido brindarme a través de la carrera.  
A los profesores, por sus enseñanzas y la formación recibida.  
A mi familia postiza, por las muestras de afecto.  
En especial a la memoria de mi padre.  
A todos muchas gracias.

## Tabla de Contenidos

Resumen .....	2
Agradecimientos .....	3
Tabla de Contenidos .....	4
Lista de Figuras.....	7
1. Introducción.....	9
1.1 Motivación .....	9
1.2 Objetivos de la tesis .....	11
1.3 Organización del trabajo .....	12
2. Estado del arte de las técnicas .....	14
2.1 Inteligencia de Negocios (Business Intelligence) .....	14
2.2 Gestión del Rendimiento Corporativo (CPM) .....	17
2.2.1 BI vs CPM.....	18
2.2.2 Marco conceptual (Framework) para CPM. ....	19
2.3 Técnica de Balanced Scorecard (BSC) .....	22
2.3.1 Las 4 perspectivas originales del BSC .....	24
2.3.2 Tableros de Control.....	27
2.3.3 Mapas Estratégicos.....	30
2.3.4 Proceso de desarrollo de la BSC .....	36
2.4 Conclusiones del capítulo .....	39
3. Estado del arte de las herramientas.....	42
3.1 Herramientas para CPM.....	42
3.1.1 Clasificación de las Herramientas de CPM.....	47
3.2 Tecnologías basadas en eclipse.....	48
3.2.1 Rich Client Platform (RCP) .....	49
3.2.2 Eclipse Modeling Framework (EMF) .....	50
3.2.3 Frameworks para Editores Gráficos (GEF, GMF, Graphiti).....	51
3.2.4 Soporte de modelos distribuidos (CDO y Dawn) .....	54

3.2.5	Eclipse 4 .....	56
3.2.6	Jazz .....	58
3.3	Conclusiones del capitulo .....	61
4.	Solución propuesta.....	63
4.1	Consideraciones Iniciales.....	63
4.2	Requerimientos .....	67
4.3	Arquitectura de Integración Dinámica (ATID).....	73
4.3.1	Componentes Centrales.....	75
4.3.2	Componentes de Valor Agregado .....	81
4.3.3	Componentes de Análisis y Reportes.....	86
4.3.4	Componentes de Interoperabilidad y Otros Conceptos.....	88
4.3.5	Configuraciones de la arquitectura.....	89
4.4	Validación de la Arquitectura .....	91
4.5	Conclusiones del capitulo .....	96
5.	Resultados de Implementación .....	98
5.1	Componentes centrales.....	98
5.1.1	Mecanismo de extensibilidad de Jazz .....	98
5.1.2	Comunicación cliente-servidor .....	100
5.1.3	Gestión de Usuarios .....	102
5.1.4	Framework de aplicaciones.....	103
5.1.5	Repositorio .....	109
5.2	Componentes de Valor Agregado .....	111
5.2.1	Búsquedas .....	112
5.2.2	Gestión de metodologías .....	112
5.2.3	Extensiones para Colaboración.....	113
5.2.4	Alertas y notificaciones.....	114
5.3	Componentes de Análisis y Reportes.....	114
5.4	Componentes de Interoperabilidad y otros conceptos .....	115
5.5	Configuraciones de la Arquitectura .....	116

5.6 Conclusiones del Capitulo .....	117
6. Conclusiones y Trabajo Futuro.....	120
6.1 Conclusiones .....	120
6.2 Aportes .....	123
6.3 Trabajos Futuros .....	125
Referencias .....	128
Referencias web.....	131
Glosario.....	135

## Lista de Figuras

Figura 1 – Una Arquitectura típica de BI .....	15
Figura 2 – Marco Conceptual de CPM .....	20
Figura 3 – Componentes principales del framework de BSC.....	23
Figura 4 – Las cuatro perspectivas originales del BSC .....	25
Figura 5 – Ejemplos de Tableros de Control .....	28
Figura 6 – Principales elementos de un Mapa Estratégico .....	31
Figura 7 – Un ejemplo de relaciones causa-efecto .....	33
Figura 8 – Ejemplos de Métricas .....	34
Figura 9 – Ejemplo de un caso práctico en un Hospital Publico .....	35
Figura 10 – Estructura de una matriz de impactos entre iniciativas y objetivos .....	35
Figura 11 – V generación de BSC – The Execution Premiun Process (XPP).....	37
Figura 12 –Implementación de una estrategia de BSC.....	38
Figura 13 - IBM Cognos 8 Business Intelligence Scorecarding.....	42
Figura 14 - Oracle Scorecard and Strategy Management.....	43
Figura 15 - SAS SPM Strategic Performance Management.....	44
Figura 16 - SAP Corporate Performance Monitor.....	44
Figura 17 - Spider Strategies .....	45
Figura 18 - Watershed Strategy Execution.....	46
Figura 19 - Strategy Map Balanced Scorecard .....	46
Figura 20 - Caracterización de las Herramientas.....	47
Figura 21 – La Plataforma Eclipse .....	49
Figura 22 – Proceso de desarrollo de un editor gráfico usando GMF .....	52
Figura 23 – Ejemplo de aplicación de GEF 3D .....	54
Figura 24 – Arquitectura de tres capas de CDO .....	55
Figura 25 - Arquitectura de RAP .....	57
Figura 26 – Arquitectura de Integración de Jazz (JIA).....	60
Figura 27 -- Ejemplo de Suite de Rational basada en Jazz.....	60
Figura 28 – Ejemplo de Aplicación de CPM.....	64
Figura 29 – Categorización de los requerimientos más importantes.....	73
Figura 30 – Diagrama de alto nivel de la arquitectura ATID .....	75
Figura 31 – Mecanismos de extensibilidad de la ATID basado en Jazz .....	76



Figura 32 – Comunicación entre los clientes y los servidores.....	77
Figura 33 – Interface de administracion de usuarios de Jazz reusada por la ATID ...	78
Figura 34 – Framework de aplicaciones web de Jazz.....	79
Figura 35 – Capacidades de Auditoria y creación de Branchs en CDO .....	80
Figura 36 – Diagrama de estados de un objeto CDO .....	81
Figura 37 – Ejemplo de funcionalidades de asistencia en búsquedas .....	82
Figura 38 – Tablero de control que permite el seguimiento de las metricas. ....	85
Figura 39 – Ejemplo de utilizacion de la componente de analisis.....	87
Figura 40 – Escalabilidad y configuraciones del servidor basadas en Jazz.....	90
Figura 41 – Soporte para trabajar offline con CDO .....	91
Figura 42 - Mecanismo de extensibilidad de jazz .....	99
Figura 43 – Esquema simplificado de una extensión del cliente.....	99
Figura 44 – Ejemplo de implementación de una comunicación cliente-servidor.....	101
Figura 45 – Esquema de comunicación cliente-servidor de un componente.....	102
Figura 46 - Soporte para coloración de usuarios .....	102
Figura 47 - Editor de Mapas Estratégico prototipado.....	104
Figura 48 – Propiedades importantes para la generación de GMF a partir de Jazz..	107
Figura 49 - Posicionamiento de la ATID relativo al diagrama de la Fig.20.....	121

# 1. Introducción

## 1.1 Motivación

Desde que la era de la informática comenzó a intentar comprender las actitudes y necesidades de los seres humanos, una de las más importantes áreas en la tecnología de la información (IT) ha sido la de “soporte para decisiones”. Hoy en día, esta área es más importante que nunca. Trabajando en ambientes dinámicos y siempre cambiantes, los gerentes de la actualidad son responsables de una inmensa variedad de decisiones con múltiples alcances e implicaciones [2].

La aplicación de las ciencias informáticas para la evaluación y mejoramiento del rendimiento empresarial, de manera de dar soporte para la toma de decisiones, ha cobrado gran interés en los últimos años debido a los avances en diversas áreas, especialmente en aquella conocida comúnmente como *Inteligencia de Negocios* (Business Intelligence). En esta área se destacan la aplicación de técnicas informáticas como OLAP, ETL, data warehousing, data mining, etc.

Sin embargo, las iniciativas relacionadas con el área de *Gestión de Rendimiento de Negocios* o BPM (Business Performance Management) están aún en su infancia y cambiando rápidamente. Por ejemplo, es común que se utilicen una gran variedad de términos para describir esta área de interés y no siempre las definiciones son claras o consistentes debido a que a pesar de que muchos términos suenen de manera similar, se utilizan con significados diferentes, lo que genera gran confusión [9]

El área de *Gestión de Rendimiento de Negocios* (BPM) también se conoce como *Gestión de Rendimiento Corporativo* (CPM - Corporate Performance Management) o *Gestión del Rendimiento Empresarial* (EPM – Enterprise Performance Management). En particular en la tesina se utilizara de aquí en adelante las siglas CPM para describir esta área y de manera de evitar confusión con el área de *Gestión de Procesos de Negocios* (BPM - Business Process Management)

CPM se puede definir como el framework que permite organizar, automatizar y analizar las metodologías, métricas, procesos y sistemas que dirigen el rendimiento de las empresas. Suele ser visto como la próxima generación de la Inteligencia de Negocios (BI) y ayuda a las empresas a hacer un uso eficiente de los recursos tanto financieros, como humanos, materiales, etc. Se puede ver como el conjunto de procesos analíticos y de gerencia, soportados por la tecnología que permite a las empresas definir objetivos estratégicos y luego medirlos de manera de controlar y

administrar el rendimiento respecto de esos objetivos. Los principales procesos de CPM incluyen el planeamiento operacional y financiero, la generación de reportes, modelación de negocios, análisis y monitoreo de *Indicadores Claves de Rendimiento* (KPI – Key Performance Indicators) relacionados con la estrategia. CPM también involucra la consolidación de datos desde varias fuentes, consultas y análisis de los datos, y la puesta en práctica de los resultados.

Los gerentes saben que la respuesta a muchas de sus dudas a la hora de tener que tomar decisiones, esta oculta en la información con la que cuentan. Sin embargo se hace muy difícil la búsqueda de dichas respuestas si no se dispone de las herramientas o técnicas adecuadas. Muchos gerentes dan prioridad al almacenamiento de grandes cantidades de datos en crudo creyendo que ellos contienen algún valor. Sin embargo, suelen eventualmente descubrir que los datos en crudo raramente disponen de algún beneficio, y que el valor real depende de la habilidad de una organización para poder analizar la información disponible. Por lo tanto, emerge la necesidad de sistemas de software capaces de recuperar, resumir, e interpretar los datos para los usuarios finales y relacionarlos con los objetivos y visiones de la empresa de manera de lograr un mejor seguimiento de los mismos.

En las empresas modernas se utilizan tecnologías de repositorios Data Warehouse para almacenar sus datos. Mejoras como ETL y otras herramientas han incrementado la velocidad con que se colecta la información. OLAP ha permitido la generación más rápida de reportes que analizan los datos. Inteligencia de Negocios se ha transformado en el arte de tomar grandes cantidades de datos para extraer la información útil y transformarla en conocimiento que permita ser usado para otras aplicaciones. Sin embargo, aún existe un GAP entre el análisis y la modelación de los objetivos empresariales (la estrategia) y la utilización del conocimiento recolectado con las técnicas de BI para el seguimiento de los mismos.

De este modo, es claro que uno de los factores de éxito necesarios para implementar un correcto sistema de manejo de rendimiento y que permita un incremento dramático de la tasa de éxito, es contar con software que sea capaz de realizar análisis adecuados que logren determinar causas y efectos, pronósticos y permitan simular estrategias antes de que estas se implementen.

Como se detallara más adelante, en esta tesis se estudian las principales metodologías y técnicas del área de CPM y se desarrolla una arquitectura de software que soporta el tipo de actividades con las que se enfrentan los usuarios en esta área.

Además, se demuestran implementaciones basadas en la plataforma eclipse para el desarrollo de las principales componentes de la arquitectura de manera de exponer las ventajas y problemas actuales de este tipo de tecnologías.

## **1.2 Objetivos de la tesis**

Se mencionan en esta sección de manera resumida los principales objetivos buscados por el presente trabajo. En la sección 6.2 se enumeran los resultados y principales aportes logrados. Entre los objetivos puntuales considerados y que persigue este trabajo se pueden mencionar:

- Clarificación del área de “Gestión de Rendimiento de Negocios” investigando el estado del arte de definición del ambito, estableciendo su relación con la “Inteligencia de Negocios” y definiendo un marco conceptual para el área. En particular analizar la técnica de “Balanced Scorecard” estableciendo su formalización dentro del marco conceptual previamente definido. Estudiar la semántica de los indicadores claves de rendimiento (KPI), los tipos de relaciones entre ellos y las ventajas de estandarización como medio de comparación y establecimiento de líneas base. Documentar cualquier inconsistencia, GAPS, o problema descubierto durante la formalización del área de dominio y el planteo de su solución.
- Estudiar y analizar el mercado de herramientas que ofrecen soluciones para el área anteriormente descrita analizando sus características y arquitecturas, exponiendo sus ventajas o falencias, estableciendo en lo posible criterios de clasificación generales para este tipo de herramientas, en función de las metodologías descritas y teniendo en cuenta las necesidades planteadas.
- Analizar las principales tecnologías disponibles basadas en la plataforma eclipse, orientadas especialmente a la construcción de herramientas y soporte de arquitecturas empresariales. Tener presente las evoluciones y tendencias de los principales desarrollos, exponiendo sus ventajas y debilidades, viendo las posibilidades de su utilización como base para la implementación. Incluir un análisis del estado de arte de tecnologías como jazz, GMF, BIRT, etc. y dentro de lo posible, utilizarl dichas componentes para acelerar la implementación de la arquitectura propuesta.
- En base al estudio anterior, proponer una arquitectura para una herramienta que permita implementar el área de dominio antes mencionada y que soporte

las técnicas más importantes. Esta arquitectura debe contemplar aspectos que hoy en día no han sido aún resueltos por la mayoría de los vendedores de software, como la fácil extensibilidad y la interoperabilidad con otras herramientas. Se debe intentar describir un diseño moderno que haga uso de las ventajas de los estándares emergentes tales como servicios web.

- Validar el uso de la arquitectura exponiendo su utilización en los principales casos de uso en el área.

### **1.3 Organización del trabajo**

En esta sección se describe brevemente la estructura de la tesina.

En el capítulo 2 se detallan los principales conceptos teóricos relacionados con el área de estudio. Primero se define el concepto de Inteligencia de Negocios (BI) (2.1) y sus alcances. Luego se define CPM (2.2) marcando su diferencia con BI. Además se enuncia un marco conceptual para el área de CPM. Entre todas las técnicas de CPM, luego se detalla en especial la técnica de Balanced Scorecard (BSC) (2.3), la que se utilizó en la tesina para prototipar y ejemplificar la clase de problemáticas con la que se enfrentan los usuarios. En particular se comenta la evolución de la técnica desde sus orígenes hasta las más recientes metodologías y adaptaciones de la misma.

En el capítulo 3 se detallan las principales herramientas que implementan actualmente el área de CPM (3.1) clasificándolas según sus características y se describe la evolución de la plataforma eclipse (3.2) desde sus inicios hasta los más recientes desarrollos y tendencias futuras para la plataforma, en especial enumerando las extensiones más importantes necesarias para el desarrollo de una arquitectura como la descrita en el capítulo 4.

En el capítulo 4 se describe la solución propuesta para el tipo de problemática descrita en el capítulo 2. Primero se describe la dinámica del área (4.1) en base a la cual se enumeran los principales requerimientos (4.2). Luego se presenta como solución la arquitectura dinámica ATID (4.3) detallando sus fundamentos, componentes principales y tecnologías de implementación de cada uno de ellos. Por último se provee un ejemplo de instanciación de la arquitectura ATID (4.4) a manera de validación de la misma.

En el capítulo 5 se describen los resultados de implementación de las principales componentes de la arquitectura ATID. Por cada componente de la arquitectura se describen las técnicas de implementación, los prototipos implementados, los

resultados y las ventajas o desventajas encontradas. Por último, en el capítulo 6 se enumeran las conclusiones finales (6.1), los principales aportes de la tesina (6.2) y las posibles extensiones o trabajos futuros (6.3).

Se finaliza con las referencias tanto bibliográficas como de sitios web y glosario de la terminología empleada.

## **2. Estado del arte de las técnicas**

En este capítulo se describe el concepto de la *Gestión de Rendimiento Corporativo* (CPM), y se detalla su diferencia y relación con la *Inteligencia de Negocios* (BI). Además se describe la técnica de Balanced Scorecard (BS) como representativa del método de CPM. Esta información se usará como base para la creación de los casos de uso que sustentan la arquitectura de software presentada en esta tesis. De esta forma se justificará el principal aporte de este trabajo que es el desarrollo y prototipado de la arquitectura de software que permite el soporte para el tipo de desafíos encontrados. Cabe destacar que muchas de las técnicas aquí descritas aún se encuentran en su infancia, y con grandes inconsistencias entre la terminología usada. Se intenta dar en esta sección una introducción a las técnicas, el detalle en profundidad de las mismas escapa al alcance de esta tesis.

### **2.1 Inteligencia de Negocios (Business Intelligence)**

Antes de definir el área de Gestión de Rendimiento Corporativo (CPM) se comenzará definiendo la Inteligencia de Negocios (BI) que muchas veces se confunde equivocadamente con CPM.

La adopción de tecnologías de almacenamiento masivo y la gran disponibilidad de las conexiones de internet ha puesto a disposición enormes cantidades de datos que han sido colectados y acumulados por las empresas en los últimos años. Por ejemplo, se estima que el tamaño total de las bases de datos de la cadena Walmart excedía 1 Petabyte (1K Terabyte) hacia el año 2005. Las empresas que son capaces de transformar esos datos en información y conocimiento, pueden usarlos para tomar decisiones más rápidamente y de manera más eficaz, dándoles esto una ventaja competitiva respecto de otras empresas. Por lo tanto es crítica la necesidad de sistemas de análisis que puedan analizar datos automáticamente, para resumir y predecir tendencias futuras.

Por ejemplo, Twentieth Century Fox utiliza BI para predecir qué actores y filmes serán populares en distintas zonas. La compañía asegura lograr ahorros de aproximadamente \$100 Millones de dólares alrededor del mundo cada año, gracias a poder evitar la presentación de ciertos filmes en cines específicos. Esa misma tecnología se utiliza para seleccionar los "trailers" (cortos previos a la presentación de una película) en cada cine y así maximizar las ventas. [w1]

Inteligencia de Negocios puede definirse como el conjunto de modelos matemáticos y metodologías de análisis que sistemáticamente explotan los datos disponibles para recuperar información y conocimiento útiles para soportar procesos de toma de decisiones complejos [32]. Aquí es donde surgen las primeras dudas sobre el alcance de BI ya que si lo miramos desde el punto de vista de análisis de datos para soportar toma de decisiones, esa descripción encajaría con el área de CPM que veremos más adelante.

Un ambiente de inteligencia de negocios intenta ofrecer al personal encargado de la toma de decisiones, información y conocimiento derivados a partir de los datos procesados, mediante la aplicación en general de modelos matemáticos y algoritmos. En algunos casos, estos pueden consistir únicamente de cálculo de totales y porcentajes, mientras que otros análisis más desarrollados podrían usar modelos avanzados de optimización, aprendizaje inductivo y predicción [32]. Estas definiciones van más allá de las que muchos autores y empresas que ofrecen sistemas de BI publican, en donde se intenta ver a la BI solo como el conjunto de herramientas electrónicas para hacer complejas consultas, visualizaciones y reportes.

En general las arquitecturas de BI tienen 3 componentes principales (Figura 1):

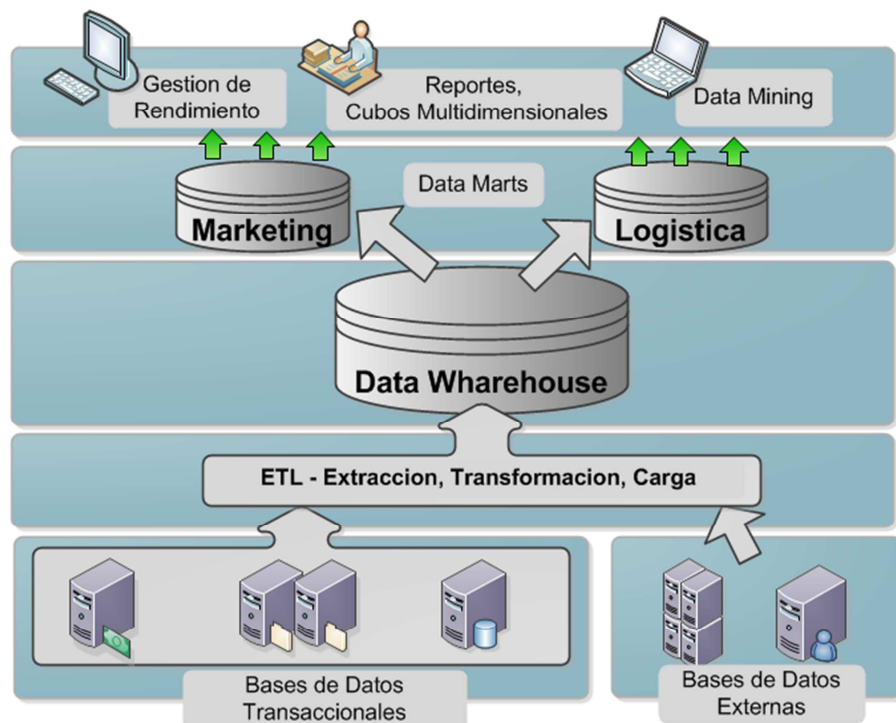


Figura 1 – Una Arquitectura típica de BI

- 1- **Fuentes de datos:** En primer lugar es necesario recolectar e integrar los datos almacenados en varias fuentes tanto primarias como secundarias,



que por lo general son heterogéneas en tipo y origen. Las fuentes consisten en su mayoría de datos de sistemas operacionales, aunque podrían incluir documentos sin estructura como emails o documentos recibidos de proveedores externos. Generalmente se requiere un esfuerzo relativamente grande para unificar e integrar las diferentes fuentes de datos.

- 2- **Data warehouses y data marts:** usando herramientas de extracción y transformación conocidas como ETL (Extracción, Transformación y Carga) los datos originados de las distintas fuentes de datos se almacenan en bases de datos diseñadas para soportar los análisis de BI. Estas bases de datos se conocen frecuentemente como Data warehouses y Datamarts. Durante la extracción, los datos se extraen desde las fuentes internas y externas. La selección de los datos a ser importados se basa en función del diseño del Data-Warehouse que a su vez, depende de la información que se requiere para los análisis de BI y los sistemas de soporte de decisiones operando sobre un dominio específico. El objetivo de la transformación y limpieza (cleanning) es mejorar la calidad de los datos extraídos desde las diferentes fuentes, mediante la corrección de inconsistencias, imprecisiones y valores faltantes. Muchas de las inconsistencias se refieren a valores en diferentes atributos que tienen el mismo significado, duplicaciones de datos, datos faltantes o fuera de rango. Los Data Warehouses abarcan tanto algoritmos como herramientas que permiten juntar repositorios de información en un único repositorio optimizado para realizar análisis de datos. Durante la fase de limpieza, reglas automáticas predefinidas se aplican para corregir la mayoría de los errores. Finalmente, los datos se cargan en las tablas del Data Warehouse para que estén disponibles para las aplicaciones.
- 3- **Metodologías de BI:** los datos son finalmente extraídos y usados para alimentar modelos matemáticos y metodologías de análisis cuya intención general es la de soportar la toma de decisiones. En un sistema de BI, pueden implementarse una gran variedad de aplicaciones para el soporte de toma de decisiones, entre las que se pueden mencionar: Análisis de Cubo Multidimensional (OLAP), Modelos de análisis estadístico y visualización, Modelos de datos de aprendizaje inductivo

para Data Mining, modelos de optimización (usados en marketing y logística). OLAP utiliza estructuras multidimensionales (o Cubos OLAP) que contienen datos resumidos de grandes bases de datos con una disposición en vectores para permitir un análisis rápido. Las actividades de data-mining constituyen un proceso iterativo destinado a analizar grandes bases de datos, con el propósito de extraer información y conocimiento que puede potencialmente ser útil para las organizaciones. En general estas técnicas poseen distintas fases las cuales pueden requerir asistencia por parte de los usuarios en etapas intermedias lo que las hace difíciles de automatizar. El propósito del data mining es deducir conclusiones a partir de observaciones o muestras del pasado, de manera de generalizarlas y aplicarlas para predecir el comportamiento de la población entera, de la manera más precisa posible. Existen muchas metodologías de data mining entre las que se destacan CRISP (CRoss-Industry Standard Process for Data Mining) y SEMMA (Sample, Explore, Modify, Model and Assess) entre otras.[w2]

## **2.2 Gestión del Rendimiento Corporativo (CPM)**

Desde principios de los noventa, las empresas han venido utilizando software de Inteligencia de Negocios (BI) para acceder, analizar y compartir información con el objetivo de entender su desempeño. Como vimos anteriormente, la Inteligencia de Negocios (BI) entrega a cada usuario de la organización acceso a información histórica almacenada en diversas fuentes de datos así como herramientas de análisis y predicciones estadísticas. El resultado es la transformación de grandes volúmenes de datos en información que mejore substancialmente el proceso de toma de decisiones en los niveles estratégicos, tácticos y operativos de la organización.

Sin embargo, a pesar de que la Inteligencia de Negocios convencional entrega información muy importante, no otorga ninguna referencia en cómo actuar o en cómo alinear a los miembros de la organización para corregir el rumbo o materializar oportunidades de negocio.

A partir del año 2002, se ha visto la aparición del concepto de *Gestión del Rendimiento Corporativo* (CPM - Corporate Performance Management), la cual representa una de las áreas de software empresarial de mayor crecimiento en la actualidad. CPM, también conocido como *Gestión del Rendimiento de Negocio*

(BPM) es un concepto cuyos orígenes pueden remontarse al año 2001 cuando los analistas de la industria Gartner Research presentaron una definición que se ha mantenido relativamente consistente durante los años siguientes. Según la definición de Gartner, el CPM describe un *conjunto de procesos, metodologías, sistemas y métricas cuyo objetivo es evaluar y dirigir el rendimiento de una empresa*. Es decir, que es un conjunto de tecnologías y procesos que permiten a las organizaciones traducir estrategias corporativas en planes de acción, monitorear la ejecución de estos y proveer información que derive en importantes mejoras financieras y operativas, de manera poder alcanzar los objetivos prefijados. En el año 2003, con el apoyo de las principales empresas, se crea el “BPM Standards Group” para tratar de clarificar las terminologías [34].

CPM es una iniciativa clave que permite a las compañías alinear objetivos estratégicos y operacionales con las actividades de la organización. CPM ha generado un gran nivel de interés entre la comunidad de IT y de negocios debido a la capacidad que ha demostrado para permitir que se alcancen las metas y objetivos planeados [9]. CPM involucra un conjunto de procesos operativos y analíticos que buscan lograr dos tareas. Primero, facilita la creación de los objetivos estratégicos permitiendo encontrar además sus indicadores claves de rendimiento (KPI), es decir, aquellos objetivos y métricas que son de valor para la organización. En segundo lugar, soporta el seguimiento de esas métricas, mediante el establecimiento de metas y la definición de iniciativas. Los objetivos e indicadores son luego asociados con las métricas operacionales y ligadas a las iniciativas concretas lo que llevan a la efectiva ejecución de estrategia en toda la organización. [34]

### **2.2.1 BI vs CPM**

Como se describió anteriormente la Inteligencia de Negocios aplica herramientas analíticas y metodológicas a grandes volúmenes de información para ser entregada a los diferentes usuarios de la organización en forma de reportes, tableros de control (Dashboard), y otras formas de resumir la información de manera de servir de soporte para la toma de decisiones. CPM intenta modelar las estrategias financieras y operativas de la empresa y relacionarlas de ser posible con la información provista por la Inteligencia de Negocios, con el objetivo de tomar acciones concretas y, por lo tanto, de generar valor a la empresa. Por ejemplo, en la Inteligencia de Negocios en general es difícil encontrar un vínculo directo entre el indicador de un tablero de

control que representa un desfase y la acción que realizará un gerente para intentar corregir de manera efectiva esa problemática. Ese tipo importante de información es la que trata de representar CPM, es decir, que CPM va un paso más allá de BI, enfocándose en la definición de los objetivos estratégicos y utilizando las mediciones de los indicadores para monitorear el rendimiento contra los objetivos prefijados.

Suele decirse que las soluciones de CPM son *proactivas* ya que las acciones e iniciativas se toman en función de los objetivos a alcanzar mientras que las soluciones de BI se dice que son *reactivas* ya que están más enfocadas en brindar información respecto de las operaciones diarias e históricas, de tal manera que los gerentes puedan “reaccionar” y tomar decisiones basados en una oportunidad específica

Por lo tanto, CPM es un área que puede desarrollarse independientemente de BI, aunque para lograr maximizar las chances de éxito se requieren de mecanismos de integración de manera de tener en cuenta en CPM la información obtenida a través de las técnicas de BI. Esta separación de alcances entre una y otra área más el hecho de tener en cuenta que CPM se pueda nutrir de los resultados de BI es lo que hace que surja la necesidad de software con un nivel de integración no antes encontrado [43].

Las tecnologías de CPM y BI obtienen su mayor impacto y beneficio en su convergencia, pero es importante como se mencionó anteriormente que esta convergencia se dé bajo un marco conceptual bien definido y bajo una arquitectura tecnológica integrada como se detallara en las secciones subsiguientes.

### **2.2.2 Marco conceptual (Framework) para CPM.**

El marco conceptual permite explicar el conjunto de ideas y condiciones que determina como se aplican las actividades de CPM. Una versión simplificada de CPM puede verse como compuesta de 4 procesos principales (Figura 2).

La importancia del marco conceptual es que proporcionan la base para el diseño, implementación y administración de un sistema de CPM particular. Los dos primeros pasos representan la formulación de la estrategia, y los dos últimos pasos definen como monitorear, ejecutar y modificar la estrategia. Los cuatro procesos se describen a continuación:

**Estrategia:** En este paso se identifica que es lo que la organización quiere lograr. Describe el curso de acción empleado para identificar la estrategia, y para generar las métricas que miden el rendimiento en el tiempo. Por ejemplo, el

establecimiento de una estrategia centrada en el cliente, puede llevar a identificar como un indicador clave del valor del negocio, un alto grado de satisfacción del mismo. Por lo tanto, un índice de satisfacción al cliente es una métrica básica que permite una evaluación del rendimiento. Este paso consiste en crear métricas de rendimiento ligadas con la estrategia y puede ser uno de los más desafiantes. Las organizaciones a menudo tienen problemas para identificar métricas que capturen con precisión el progreso respecto de los objetivos. Las mediciones usadas para capturar los elementos que favorecen la creación de valor (value drivers) se conocen comúnmente como Indicadores de Rendimiento Clave (KPI). Como se explicara más adelante los Mapas Estratégicos (usados en Balanced Scorecard) son uno de los métodos más populares para identificar los value drivers, que a su vez lleva al descubrimiento de los KPI. Este es uno de los pasos más importantes dentro de CPM.



**Figura 2 – Marco Conceptual de CPM**

**Planificación:** En el paso de planificación se desarrolla un plan de acción sobre cómo implementar la estrategia. Este paso permite a los gerentes dentro de distintas unidades del negocio a establecer los objetivos, diseñar los proyectos y desarrollar los presupuestos que soportan la estrategia corporativa. Un resultado primario de la planificación es un plan detallado que especifica cómo se asignaran los recursos para cumplir los objetivos. Los planes describen cómo cada unidad contribuirá o influenciará para alcanzar los objetivos. Las unidades de la empresa con aspectos en común deben trabajar conjuntamente para asegurar que los planes no se contradicen

unos con otros. Las KPI permiten a las unidades funcionales a mantener el foco en las iniciativas.

**Monitorear y Analizar:** este es el tercer paso del framework de CPM. La influencia y beneficios de la implementación CPM se hacen aquí más visibles. Este paso facilita el constante monitoreo de los resultados de rendimiento respecto de las métricas. Al proveer un balanceado equilibrio de reportes operacionales y analíticos en todos los niveles de la organización, se ayuda a evaluar el rendimiento de las unidades individuales. Aquí es donde puede resultar beneficiosa la utilización de BI para soportar el tipo de actividades de este paso. Datos actuales e históricos son consolidados en data warehouses con tal propósito. Las aplicaciones que permiten realizar consultas multidimensionales y análisis de datos, proveen los mecanismos para el análisis del rendimiento de manera de poder compararlos contra las metas preestablecidas.

**Tomar acciones correctivas:** este es el cuarto proceso central dentro del framework CPM. Este paso contribuye a tomar en el momento adecuado, las acciones apropiadas para cambiar un desfase de rendimiento descubierto durante el monitoreo y análisis. Además de alertar a los usuarios de problemas potenciales, las actividades en este paso proveen guías y sugerencias de cómo tratar las situaciones problemáticas que surgen. Consecuentemente, al tomar acciones correctivas lo antes posible, se permite a los usuarios el evitar problemas que podrían salir fuera de control de otro modo.

Como se mencionó anteriormente, los primeros dos pasos representan la estrategia y el planeamiento, mientras que los dos últimos representan la estrategia de ejecución y acción. Por ello se dice comúnmente que CPM permite además llenar el gap que suele existir en las organizaciones entre la estrategia y la ejecución. Con estos cuatro pasos además el framework intenta alinear a toda la empresa en la misma dirección. De acuerdo con una encuesta realizada por la “American Management Association”, de 203 compañías, aquellas gerenciadas con este framework lograban un mejor rendimiento que sus competidoras.

Existen varias metodologías para implementar CPM y cada empresa suele elegir distintas disciplinas creando un framework propio mediante el cual intentan alinear la planificación y ejecución, con las estrategias, tácticas y los objetivos tanto empresariales como de las distintas unidades de negocio. Entre las metodologías más conocidas podemos mencionar *Six Sigma*, *Balanced Scorecard*, *Actividades basadas*

*en costo (ABC), Manejo de Calidad Total (TQM), Valor Económico Agregado (EVA), Teoría de restricciones (TOC), entre otras.*

El área de CPM también produce otras ramificaciones importantes como por ejemplo, la definición del “*Strategic Management Maturity Model*” (SMMM) desarrollado por el Balanced Scorecard Institute, modelo de prácticas y capacidades similar en estructura al “*Capability Maturity Model*” (CMM),

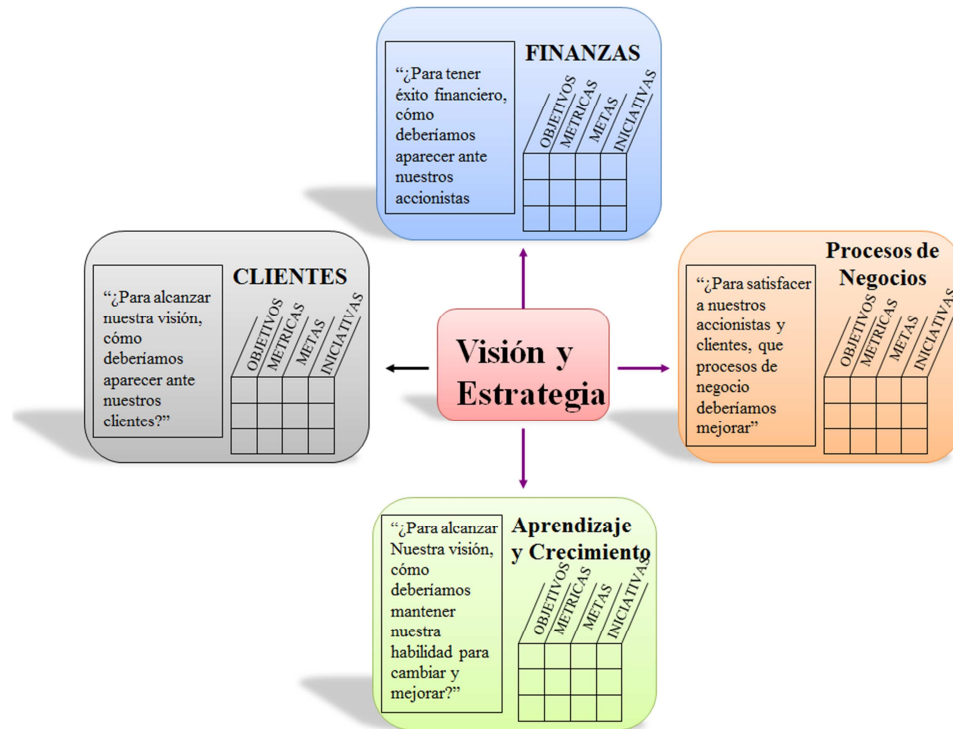
Uno de los desafíos más grandes para implementar CPM es la selección de las mediciones que sirvan como indicadores claves de rendimiento. Una de las metodologías que simplifica esta búsqueda y una de las más ampliamente adoptadas es Balanced Scorecard y es la que se desarrolló en esta tesina para demostrar las necesidades del área. En la siguiente sección se da una descripción simplificada de la técnica.

### **2.3 Técnica de Balanced Scorecard (BSC)**

Desde 1992, la gestión del rendimiento corporativo CPM ha sido fuertemente influenciada por el surgimiento del framework de Balanced Scorecard (conocido a veces en castellano como cuadro de mando integral (CMI) o Indicadores Balanceados de Desempeño). Es común que los gerentes usen este framework para clarificar los objetivos de la organización, para identificar como llevar registro de ellos y para estructurar los mecanismos por los cuales se ejecutaran las acciones correctivas necesarias. Estos pasos son los mismos que se describieron en el marco general de CPM, y debido a ello a menudo BSC es usado como la base para las actividades de CPM dentro de la organización.

El concepto de BSC fue presentado en el número de Enero/Febrero de 1992 de la revista Harvard Business Review. Sus autores, Robert Kaplan y David Norton, plantearon BSC como un sistema de administración que iba más allá de la perspectiva financiera con la que los gerentes acostumbran evaluar la marcha de una empresa. Se solía ejemplificar que así como no es posible comandar un avión controlando apenas la velocidad, los indicadores financieros por si solos no eran suficientes para garantizar que una empresa se dirija en la dirección correcta. Los directores de importantes empresas se daban cuenta de que indicadores como el ROI o el de “Ganancias por Acción” emitían señales erradas cuando se pretendía incorporar en el análisis actividades como la innovación o las mejoras continuas. Por ello se plantearon una serie de perspectivas que se describirán en la siguiente sección.

Por lo tanto se puede ver a BSC como un método que ayuda a las empresas a expresar los objetivos e iniciativas necesarias para cumplir con la visión y estrategia. Es una herramienta que provee técnicas para descubrir métricas claves que permiten monitorear continuamente si la empresa y sus empleados están alcanzando las metas definidas en el plan estratégico.



**Figura 3 – Componentes principales del framework de BSC**

BSC intenta dar un puente entre la visión y estrategia de una unidad del negocio de manera de relacionarlos con objetivos y mediciones tangibles. El desarrollo de BSC comienza con la enunciación de la visión y estrategias de la empresa. A partir de allí, se definen los objetivos que se agrupan en 4 perspectivas que están muy relacionadas entre si. En la perspectiva de finanzas se enuncian los objetivos financieros requeridos para alcanzar la visión. Estos a su vez serán el resultado de los mecanismos y estrategias que rijan las relaciones con los clientes. Los procesos internos se planifican para satisfacer los requerimientos financieros y los de los clientes. Finalmente, la metodología reconoce que el aprendizaje y crecimiento es la plataforma donde reposa todo el sistema.

Recientes encuestas indican que cerca del 50% de las empresas listadas en el ranking “Fortune 1000” utilizan este modelo en los EEUU, mientras que en Europa este porcentaje alcanza el 45%. Es por este motivo que BSC fue escogido por la prestigiosa revista Harvard Business Review como una de las prácticas de gestión



más importantes y revolucionarias de los últimos 75 años. Asimismo, en el año 2001, el primer comité Temático de Premio Nacional de la Calidad (PNQ) eligió al BSC como una de las herramientas de gestión para la excelencia empresarial. [35]

### 2.3.1 Las 4 perspectivas originales del BSC

La técnica inicial de BSC sugería que se vea a la organización desde cuatro perspectivas, cada una de las cuales ayuda a dar respuestas a una serie de preguntas determinadas como se detalla a continuación:

- **Financiera** (Financial): ¿Cómo nos vemos a los ojos de los accionistas? ¿Qué Objetivos Financieros debemos lograr para ser exitosos? ¿Cómo puede la empresa demostrar que su estrategia está siendo exitosa financieramente?
- **Del cliente** (Customer): ¿Cómo nos ven los clientes? ¿Qué necesidades de los Clientes debemos atender para tener éxito? ¿Cómo puede la empresa demostrar que está entregando el valor que los clientes esperan?
- **Interna del Negocio** (Internal Business): ¿En qué debemos sobresalir? ¿En qué Procesos debemos ser excelentes? ¿En cuáles procesos debe la empresa ser excelente para entregarles valor a sus clientes?
- **Desarrollo y Aprendizaje** (Learning and Growth): ¿Podemos continuar mejorando y creando valor? ¿Cómo debe nuestra organización aprender e innovar para alcanzar sus objetivos? ¿Qué acciones debe emprender la empresa para preparar la gente y la organización para el futuro?

A continuación se describen detalladamente cada una de las perspectivas dando ejemplos de métricas típicas para cada una de ellas.

**Perspectiva financiera.** En general, los indicadores financieros están basados en la contabilidad de la Compañía, y muestran el pasado de la misma. El BSC continúa usando los índices financieros tradicionales (ROI, EVA, etc) que representan una síntesis clave del desempeño comercial y gerencial. Los accionistas de la empresa, por ejemplo, esperan que el dinero que invirtieron en ella produzca un rendimiento adecuado. Si no lo logran, es probable que trasladen su dinero a otra empresa o a una inversión más rentable. Con el BSC no estamos dejando a un lado la información financiera que tradicionalmente necesita una empresa y que siempre será prioritaria. Algunos indicadores frecuentemente utilizados son: Índice de liquidez. Índice de

endeudamiento, índice de rendimiento del capital invertido, Cuota de Mercado, Flujo de Dinero, etc



Figura 4 – Las cuatro perspectivas originales del BSC

**Perspectiva del cliente.** Para lograr el desempeño financiero que una empresa desea, es fundamental que posea clientes leales y satisfechos, con ese objetivo en esta perspectiva se miden las relaciones con los clientes y las expectativas que los mismos tienen sobre los negocios. Además, en esta perspectiva se toman en cuenta los principales elementos que generan valor para los clientes integrándolos en una propuesta de valor, para poder así centrarse en los procesos que para ellos son más importantes y que más los satisfacen. El conocimiento de los clientes y de los procesos que más valor generan es muy importante. Las mediciones incluyen la satisfacción del cliente, retención de clientes, adquisición de nuevos clientes, tiempos de entrega reducidos, entrega en tiempo convenido. Las relaciones a largo plazo con el cliente son muy importantes, son la base para poder permanecer en un mercado competitivo y deberían formar parte destacada de la misión de la empresa. Los clientes esperan recibir durante su relación con la empresa, productos de óptima calidad, con un costo adecuado, que se entreguen a tiempo y que su utilidad corresponda a lo verdaderamente ofrecido.

**Perspectiva de Procesos Internos de Negocios.** Analiza la adecuación de los procesos internos de la empresa para la obtención de la satisfacción del cliente y

logro de altos niveles de rendimiento financiero. Los procesos de la empresa deben analizarse y evaluarse para conseguir entregarles valor a los clientes. En esta perspectiva se enfatiza una diferencia fundamental entre BSC y los sistemas tradicionales de medición del rendimiento. En los sistemas tradicionales lo que se busca es hacer seguimiento a los procesos actuales de la empresa y tratar de mejorarlos. Con el BSC lo que surge además es la identificación de procesos enteramente nuevos donde la empresa debe destacarse para lograr sus objetivos financieros y los del cliente. Se distinguen cuatro tipos de procesos:

- *Procesos de Operaciones*. Desarrollados a través de los análisis de calidad y reingeniería. Los indicadores son los relativos a costos, calidad, tiempos o flexibilidad de los procesos.
- *Procesos de Gestión de Clientes*. Ejemplos de Indicadores: Selección de clientes, captación de clientes, retención y crecimiento de clientes.
- *Procesos de Innovación (difícil de medir)*. Ejemplo de indicadores: % de productos nuevos, % productos patentados, introducción de nuevos productos en relación a la competencia.
- *Procesos relacionados con el Medio Ambiente y la Comunidad*. Ejemplos de Indicadores: de Gestión Ambiental, Seguridad e Higiene y Responsabilidad Social Corporativa

**Perspectiva del desarrollo de las personas y el aprendizaje.** El modelo plantea los valores de este segmento como el conjunto de guías para el resto de las perspectivas. La perspectiva de la innovación y del aprendizaje es el cimiento de toda estrategia y se enfoca en el capital humano y en los activos intangibles de la empresa. Esta perspectiva busca definir las necesidades de talento humano y desarrollo que necesita una empresa para alcanzar los objetivos propuestos en las otras perspectivas del BSC. Los productos y servicios cumplen su ciclo de vida y es necesario rediseñarlos o disponer de unos nuevos, con mejores y más atractivas características. La capacidad de innovar y mejorar exige el entrenamiento de los empleados y la revisión de las actitudes culturales de la empresa relacionadas tanto con el individuo como con el propio mejoramiento de la organización. Se critica la visión de la contabilidad tradicional, que considera la formación como un gasto, no como una inversión. Se clasifica los activos relativos al aprendizaje y mejora en:

- *Capacidad y competencia de las personas* (gestión de los empleados). Incluye indicadores de satisfacción de los empleados, productividad, necesidad de formación, entre otros.
- *Sistemas de información* (sistemas que proveen información útil para el trabajo). Indicadores: bases de datos estratégicos, software propio, las patentes y copyrights, entre otros.
- *Cultura-clima-motivación para el aprendizaje y la acción*. Indicadores: iniciativa de las personas y equipos, la capacidad de trabajar en equipo, el alineamiento con la visión de la empresa, entre otros.

Si bien estas 4 perspectivas han sido exitosas en muchas industrias, puede ser que dependiendo del tipo de empresa o unidad de negocio específica sea más conveniente otro conjunto de perspectivas. Esto es permitido por BSC, es decir, que las perspectivas no son fijas sino que es posible adaptarlas a las necesidades [16].

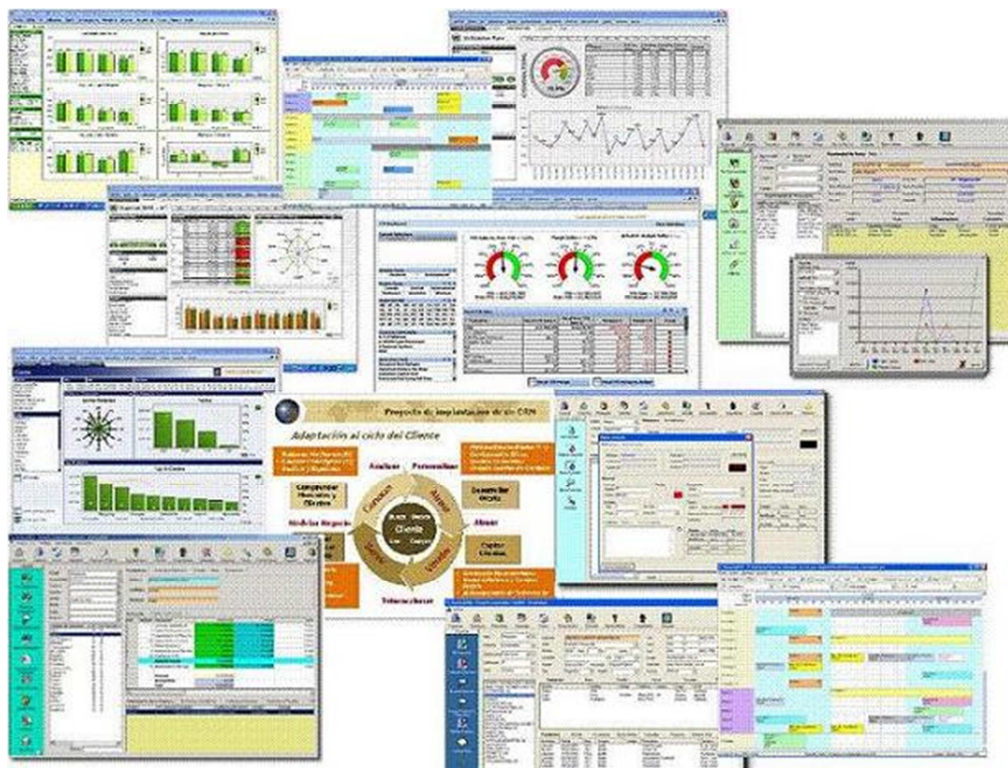
Veremos que uno de los primeros objetivos de BSC fue la creación de los *Tableros de Control*, por lo que se comenzará caracterizando los mismos en la siguiente sección.

### **2.3.2 Tableros de Control**

El concepto de Tablero de Control (Dashboard) deriva del concepto denominado "Tableau de bord" en Francia, que significa "Tablero de Instrumentos". También denominados Tableros de Comandos o Cuadros de Mando. De forma genérica, engloba a varias herramientas que muestran información relevante para la empresa a través de una serie de indicadores claves de rendimiento (KPI). Los primeros Tableros de Control eran desarrollados por los propios empleados de las empresas basados en su experiencia y como método de recopilar o resumir datos. Sin embargo, se descubrió que el armado de un Tablero de Control que resulte de valor para la empresa es una tarea compleja y para lo que se deben tener en cuenta una gran cantidad de factores. Por ejemplo, por un lado se debe evitar el exceso de información seleccionando solo los indicadores relevantes, pero por otro lado se deben encontrar todos los indicadores que son realmente importantes, dado que la carencia de alguno de ellos puede dar falsas indicaciones sobre la evolución de la estrategia. Además, la información de los distintos tableros debe estar generada de manera de evitar situaciones como las islas sectoriales dentro de las propias empresas

o contradicciones. También se deben tener en cuenta otros aspectos como por ejemplo, que los indicadores sean presentados de una manera clara, útil y fácil de leer.

Por lo tanto se vio la importancia de establecer un sistema de señales en forma de Tablero de Control que indique la variación de las magnitudes verdaderamente importantes que se deben vigilar para someter a control la gestión. De esta forma surgió la pregunta de cómo encontrar un conjunto reducido de aquellas métricas que resultasen claves para medir el rendimiento de las empresas, es decir, se requerían de técnicas para la selección de los indicadores necesarios para la toma de decisiones, sobre todo en el menor número posible.



**Figura 5 – Ejemplos de Tableros de Control**

Uno de los primeros objetivos del framework de BSC fue producir como resultado un conjunto de indicadores que guiaban la construcción de los tableros de control. Estos se conocen como también *Cuadros de Mando Integral* (CMI) y están basados en los principios de Kaplan y Norton. En algunos casos, debido a cuestiones de costos de puesta en funcionamiento y por simplicidad, se utilizan otros tipos de tableros, como por ejemplo el Tablero de Control Operativo, que es una utilidad de control enfocada a la monitorización de variables operativas, es decir, indicadores rutinarios ligados a áreas o departamentos específicos de la empresa.

La que es considerada como la primera generación de BSC deriva del primer libro de Kaplan y Norton ("The Balanced Scorecard") [16] publicado en el año 1996, el cual aportaba un conjunto de mediciones que sirvieron fundamentalmente para establecer el desempeño de la empresa, las cuales permitían construir los Cuadros de Mando Integrales. Si bien fue cierto que los objetivos estratégicos relacionados con estas mediciones estaban ligados a un nivel más alto dentro de la organización, enmarcados dentro de las perspectivas denominadas Financiera, Clientes, Proceso, Aprendizaje y Crecimiento, la primera generación de BSC no le exigía a las empresas ciertos aspectos, como por ejemplo que se hicieran análisis estratégicos ni que examinaran su posición competitiva. En estos casos BSC se adaptaba mejor a situaciones donde lo que quería la empresa era comunicarle a sus distintas unidades de negocio y a diferentes niveles, las mediciones que iban a ser importantes para avanzar hacia el futuro. Pronto la experiencia señaló que las mediciones por sí solas, sin estar claramente articuladas a una estrategia y sin aporte de liderazgo por parte de la alta gerencia, no eran los catalizadores para crear una cultura orientada hacia un alto rendimiento.

En una segunda generación del BSC se buscó incluir la visión estratégica dentro del conjunto de indicadores de gestión. La primera generación de BSC afrontó serias dificultades en su implementación debido a su vaguedad en cuanto al proceso y criterio de selección de los indicadores de gestión y a su agrupamiento dentro de las cuatro perspectivas. Kaplan y Norton pronto reconocieron esta situación e introdujeron el concepto de los "objetivos estratégicos" para esclarecer los objetivos de las metas y se enfocaron también en buscar causalidad entre las mediciones. Ya a mediados de la década de los 1990s se había abierto paso la idea de que establecer nexos estratégicos era un elemento clave en la metodología de diseño del BSC y empezaron a aparecer representaciones gráficas que mostraban los nexos entre los objetivos estratégicos como también las relaciones de causalidad entre objetivos clave dentro de las perspectivas y relacionados con los indicadores financieros. Estos gráficos que inicialmente se denominaron "modelos de conectividad estratégica", más tarde recibieron el nombre de "mapas estratégicos", innovación que Norton y Kaplan consignaron en su segundo libro, "The Strategy Focused Organization" (2001) [36]. El impacto de estos cambios, logró que el BSC pasara de ser un sistema mejorado de medición a un sistema de administración estratégica.

### 2.3.3 Mapas Estratégicos

En lo que se suele denominar como la tercera generación de BSC se unificaron los objetivos con los indicadores de gestión, a partir de unas relaciones causa-efecto y se establecieron los mapas estratégicos. A finales de la década de los 1990s se le agregó al BSC otro elemento avanzado de diseño: la declaración de la misión y visión. Inicialmente esta declaración se hacía al final del proceso de diseño y se les pedía a los gerentes involucrados que se imaginaran cuál sería el impacto en la empresa si se lograban los objetivos determinados al inicio del proceso. Sin embargo, los autores de dieron cuenta de que esta declaración es más útil si se hace desde el principio del proceso ya que es más fácil conceptualizar la estrategia, seleccionar los objetivos estratégicos y elaborar las hipótesis de causalidad. En 2004 Norton y Kaplan dedicaron todo un libro a los Mapas Estratégicos, *Strategy Maps*, (2004) [37], donde ya con mucha claridad y precisión se indicaba a las empresas como describir los nexos entre los activos intangibles y la creación de valor. En esta sección se describen las características de los mapas estratégicos.

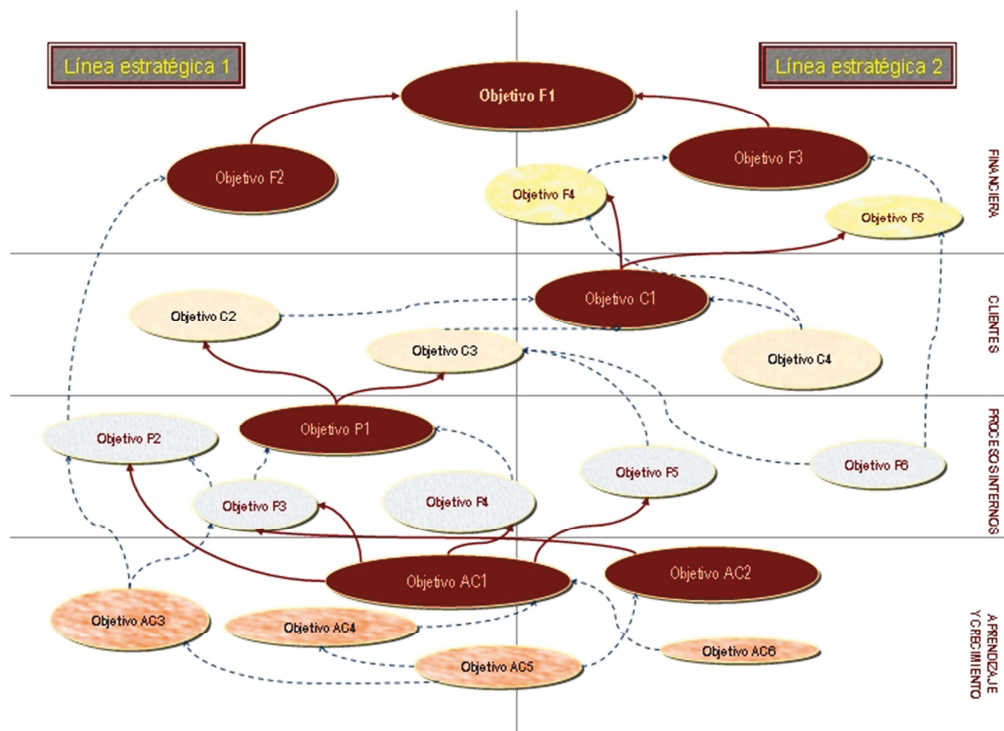
Las organizaciones, mientras elaboraban sus CMI, fueron forzadas a repensar sus prioridades estratégicas y describir sus estrategias. Esto llevó a Kaplan y Norton a toparse con un principio más profundo: *no se puede medir lo que no se puede describir*. Los mapas estratégicos, que originalmente habían sido una parte del proceso de construcción del CMI, ahora se convirtieron en el tema central.

Los mapas estratégicos (strategic maps) son una manera de proporcionar una visión macro de la estrategia de una organización, y proveen un lenguaje para describir la estrategia, antes de elegir las métricas para evaluar su desempeño.

En todo proyecto de BSC debe elaborarse uno o mas Mapas Estratégicos que describen, en forma clara, la estrategia de la organización. Esto se presenta a través de objetivos vinculados entre sí y distribuidos en las 4 perspectivas. Cada objetivo tendrá, a su vez, 1 o 2 indicadores asociados con metas y planes de acción. Esquemáticamente un Mapa Estratégico tiene el aspecto mostrado en la Figura 6.

De este modo por ejemplo, en los mapas estratégicos se plantea como objetivo “ser líderes indiscutidos del mercado”, que luego durante la implantación del CMI se transformará en una métrica del tipo “alcanzar el 28% del mercado en 12 meses”.

A continuación se detallan cada uno de los componentes de BSC y de los mapas estratégicos:



**Figura 6 – Principales elementos de un Mapa Estratégico**

**Misión y Visión.** La *misión* está vinculada con los valores centrales, es una frase concisa, con foco “interno”, de la razón de la existencia de la organización, el propósito básico hacia el que apuntan sus actividades, y los valores que guían las actividades de sus empleados. También describe cómo competir y generar valor al cliente. La *visión* de una organización es también una frase concisa que describe las metas de mediano y largo plazo. La visión es “externa”, orientada al mercado, y debería expresar de una manera visionaria cómo quiere la organización ser percibida por el mundo.

Las principales diferencias entre Misión y Visión son:

- La *Misión* mira hacia “adentro” de la organización, es el que da sentido a la organización; mientras que la *Visión* lo que hace es mirar hacia “afuera”.
- La *Misión* tiene su horizonte en el mediano plazo, acentuando los aspectos que se deben cambiar; la *Visión* se orienta al muy largo plazo, en el sentido de rescatar la identidad, los aspectos inmutables esenciales.

Una *Estrategia* implica elegir un conjunto de actividades en las que la organización sobresaldrá para crear una diferencia sostenida. El *mapa estratégico* está relacionado con una visión macro de las estrategias seguidas por la organización.



La visión, como meta a alcanzar a largo plazo, debe ser el eje de impulso de la propia estrategia

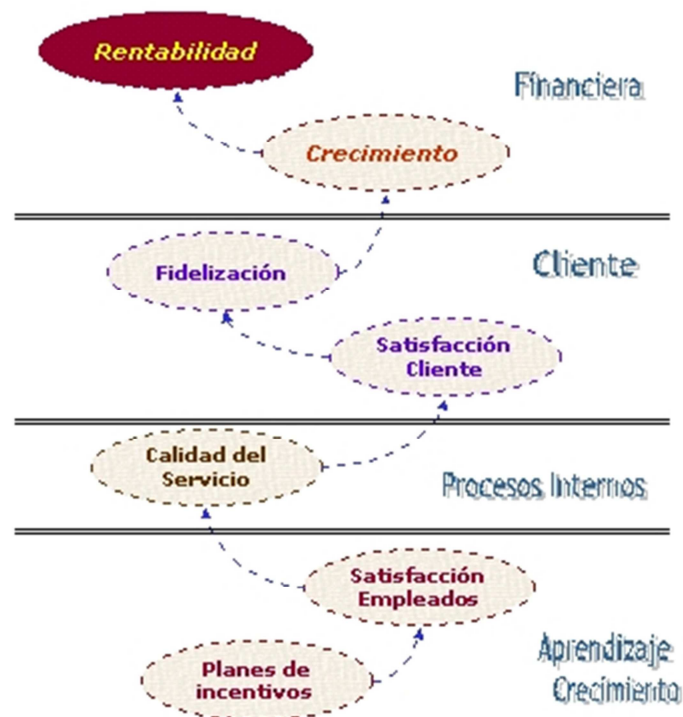
**Perspectivas.** Como ya hemos visto en la sección 2.3.1 se trata de un grupo equilibrado de objetivos estratégicos. Generalmente, aparecen ordenadas tal como hemos dispuesto en el mapa estratégico de la figura anterior. Aunque podremos disponer de ellas en el orden que consideremos oportuno en función de aspectos como la naturaleza de la organización, su estructura jerárquica, sus relaciones con el entorno, la importancia de alguno de sus procesos, el interés y funcionalidad de algunos de sus clientes. En el sentido de visualización "horizontal", los objetivos están agrupados por Perspectivas . En el sentido "vertical" del mapa suelen estar agrupados por *Líneas estratégicas* o *Temas*.

**Objetivos Estratégicos.** Las perspectivas, albergan un conjunto de *Objetivos Estratégicos*, aunque en ocasiones y en algunas empresas se ha podido constatar el uso de *Iniciativas Estratégicas* para su confección. Lo que no debería suceder es el hecho de que los mapas estratégicos contengan *Indicadores*, pues se verá que las relaciones entre objetivos estratégicos son de causalidad, no matemáticas como ocurre en el caso de los indicadores. Por lo cual no es aconsejable mezclar objetivos e indicadores en el mapa estratégico.

Los objetivos son una simple apuesta por algo que consideramos estratégico en la empresa. El conjunto de objetivos estratégicos, es decir, el *Mapa estratégico*, representa una situación óptima deseada en la organización. A través de los mismos se facilita la posterior medición y valoración de la gestión. Los objetivos tienen que ser congruentes con la Misión-Visión propuestas y consistentes con el resto de objetivos.

**Relaciones causa-efecto.** Ciertos objetivos estratégicos guardan relación unos con otros. Los mapas estratégicos permiten documentar y visualizar estas dependencias, mediante la conexión de los objetivos con flechas cuya interpretación es la de relaciones de causa-efecto. La punta de la flecha señala hacia el efecto, mientras que la base indica la causa. En general las relaciones pueden leerse desde el origen hacia el destino intercalando las palabras SI y ENTONCES. En el caso de la distribución más frecuente de las perspectivas como la mostrada en la figura 7, las relaciones suceden desde abajo hacia arriba. Esto se debe a que existe una relación inherente entre las perspectivas. Comenzando desde arriba, encontramos la hipótesis de que los resultados financieros sólo pueden conseguirse si los objetivos de los

clientes están satisfechos. La propuesta de valor para el cliente describe como generar ventas y fidelidad de los clientes. Los procesos internos crean y aportan la propuesta de valor para el cliente, mientras que los activos intangibles que respaldan los procesos internos proporcionan los fundamentos de la estrategia. Alinear los objetivos de estas cuatro perspectivas es la clave de la creación de valor y, por lo tanto, de una estrategia focalizada e internamente consistente. Esta arquitectura de causa efecto, que vincula las cuatro perspectivas, es la estructura alrededor de la cual se desarrolla un mapa estratégico



**Figura 7 – Un ejemplo de relaciones causa-efecto**

Un ejemplo de una cadena de relaciones causa-efecto es la mostrada en la figura 7. Si se observa desde abajo la figura, en el área de Aprendizaje-Crecimiento la empresa podría establecer inicialmente como objetivo interesante una serie de “Planes de incentivos”, de forma que los empleados estén más satisfechos con su trabajo y sean más eficientes. De ese modo, es de esperarse que la “Calidad del servicio” mejoraría, siendo este un objetivo clave del área de Procesos internos de la organización. Todo ello debería derivar en una “satisfacción del cliente” mayor, lo que consecuentemente puede desembocar en una “fidelización” notable de la clientela, aspecto muy cuidado en el área de Clientes. Finalmente, ello genera mayores ventas lo que hace aumentar los beneficios y en cierto modo, estamos

incidiendo en la Estrategia de “Crecimiento” de la empresa, lo que deriva en una mayor “rentabilidad” y una creación de valor importante, aspectos significativos del área financiera de la empresa.

**Indicadores.** Son los que permiten *medir y valorar* el logro y avance de los objetivos definidos y propuestos. Los indicadores son métricas financieras y no financieras que son relevantes para la obtención del objetivo planeado. En general no todas las métricas que miden rendimiento son útiles. Los indicadores que se derivan de BSC también se conocen como indicadores claves de rendimiento (KPI) ya que son los más importantes para la organización. En general BSC ayuda a obtener un número acotado (15 a 20) de KPIs útiles para la organización. Es en este momento cuando estaríamos abordando el concepto de Cuadro de Mando Integral propiamente dicho.

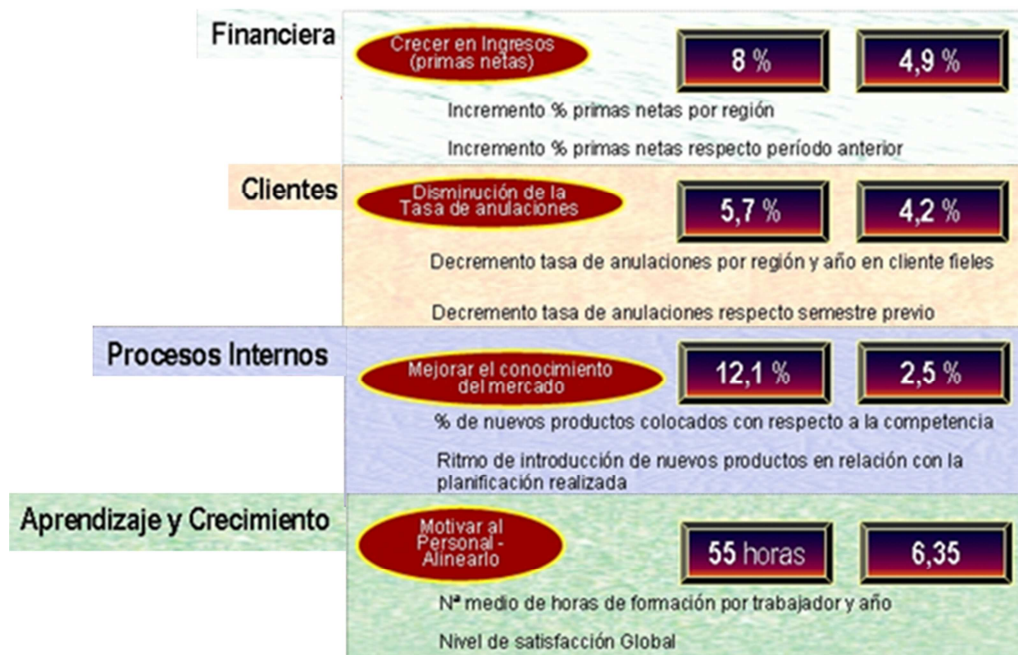


Figura 8 – Ejemplos de Métricas

Además, el hecho de tener indicadores, permite el establecimiento de Metas como se definirá en el siguiente apartado. Dichas metas condicionan el ritmo de evolución estratégica deseado en relación con la situación inicial y la cadencia de cambio que es posible asumir.

**Metas.** Las metas son los valores esperados o deseados en un marco temporal específico para un indicador estratégico. Las metas comúnmente se definen de acuerdo a la frecuencia de reporte del indicador, esto es, si el valor del indicador se obtiene semestralmente la meta se plantea de manera acorde.

**Iniciativas.** Una iniciativa es la acción o conjunto de acciones que contribuyen a alcanzar las metas que nos hemos planteado para los distintos objetivos y, por ende, son las que efectivamente van a permitir la creación de valor de la organización.

Objetivos	Indicadores	Metas	Iniciativas
Mejorar la accesibilidad y oportunidad de la atención	Horarios flexibles de consultas e intervenciones	100% de atención las 24 horas	Intensificación de horarios nocturnos
	Disponibilidad de recursos humanos	Tasa de suspensión de consultas o intervenciones por causa de falta de personal del 0%	Esudio del personal optimo por departamento
Aumentar la satisfacción de los clientes	Traslados seguros y adecuados	Cero quejas recibidas en procedimientos de traslados	Seguimiento del protocolo de traslados
	Bajo porcentaje de suspensión de operaciones	Reducir al 0.03% la suspensión de operaciones	Informe de las causas de la suspensión de operaciones
	Trato digno y amable	4/5 en encuestas de satisfacción del cliente	Ofrecer formación en liderazgo efectivo al personal

Figura 9 – Ejemplo de un caso práctico en un Hospital Publico

El nivel de impacto de las iniciativas sobre los distintos objetivos ha de ser evaluado. A través de una matriz de impactos y su análisis, podemos supervisar y coordinar los posibles desequilibrios existentes entre objetivos

		Iniciativa 1	Iniciiativa 2	Iniciiativa 3	Iniciiativa 4	Iniciiativa 5	Iniciiativa 6	Iniciiativa 7	Iniciiativa 8	Iniciiativa 9	Iniciiativa 10	Iniciiativa 11	Iniciiativa 12	Iniciiativa 13	Iniciiativa 14	Iniciiativa 15	Iniciiativa 16	Iniciiativa 17	Iniciiativa 18	Iniciiativa 19	Iniciiativa 20	Iniciiativa 21	Iniciiativa 22	
<b>F</b>	Objetivo F1																							
	Objetivo F2																							
	Objetivo F3																							
	Objetivo F4																							
<b>C</b>	Objetivo C1																							
	Objetivo C2																							
	Objetivo C3																							
	Objetivo C4																							
<b>P</b>	Objetivo P1																							
	Objetivo P2																							
	Objetivo P3																							
	Objetivo P4																							
<b>AC</b>	Objetivo AC1																							
	Objetivo AC2																							
	Objetivo AC3																							
	Objetivo AC4																							

<span style="color: green;">■</span>	Impacto positivo ALTO
<span style="color: yellow;">■</span>	Impacto positivo NORMAL
<span style="color: white;">■</span>	Sin Impacto
<span style="color: red;">■</span>	Impacto negativo NORMAL
<span style="color: darkred;">■</span>	Impacto negativo ALTO

Figura 10 – Estructura de una matriz de impactos entre iniciativas y objetivos

#### 2.3.4 Proceso de desarrollo de la BSC

En esta sección se presentan las etapas y actividades involucradas en el desarrollo del framework de BSC. El mismo ha sufrido cambios desde sus inicios. El método aquí descrito se basa en las últimas publicaciones de Norton y Kaplan. Los cambios introducidos se describen a continuación.

En 2006, Norton y Kaplan publicaron "*Alignment: Using the Balanced Scorecard to Create Corporate Synergies*" [38], donde explicaron la aplicación del BSC a nivel de la estrategia de toda la corporación y mostraron como las empresas exitosas lograron desarrollar sinergias poderosas entre sus unidades de negocios, donde antes se presentaba la falta de coordinación, conflictos, pérdida de oportunidades y desempeños medianos, definiendo explícitamente el papel de la casa matriz en el establecimiento, coordinación y supervisión de la estrategia corporativa. Esta fue considerada por algunos autores como la cuarta generación de BSC y planteaba el hecho de que contar con un mapa estratégico no era suficiente si el resto de la organización no estaba alineado con esa estrategia, principalmente el nivel operativo.

Recientemente, en 2008, en su libro "*The Execution Premium: Linking Strategy to Operations for Competitive Advantage*" [39], Norton y Kaplan describen un sistema que se desarrolla en diferentes etapas y que permite a las empresas obtener beneficios mensurables mediante una cuidadosa y bien formulada estrategia empresarial. Una de esas etapas consiste en el planeamiento de la ejecución de la estrategia mediante el uso de los mapas estratégicos y los tableros de control para la identificación y descripción de los nexos de causalidad entre el conjunto de iniciativas estratégicas. Este libro sienta las bases para lo que algunos consideran como la V generación de BSC [w3]. Los autores presentan el modelo denominado "Execution Premium Process (XPP)" que se ilustra en la figura 11 el cual consta de seis fases que se describen a continuación:

1. **Desarrollar la estrategia** – plantea que las organizaciones deben ser capaces de establecer exactamente en qué negocio están, identificar los aspectos claves que deben enfrentar y determinar cómo poder competir mejor. El desarrollo de la estrategia utiliza herramientas estratégicas tales como la misión, los valores y la visión; el análisis de la competencia, de la economía y el entorno en general, así como otras metodologías.



Figura 11 – V generación de BSC – The Execution Premium Process (XPP)

2. **Traducir la Estrategia** – sugiere que las organizaciones se concentren en desarrollar objetivos estratégicos, indicadores, metas, iniciativas ligadas a presupuestos que guíen la acción y la asignación de recursos. Se requiere no solo describir la estrategia en un mapa, sino que hay que evaluar el plan, identificar planes de acción y plantearse como llevar a cabo las iniciativas. Recomienda identificar a los responsables de liderar el proceso de ejecución de la estrategia. En la etapa de planeamiento de la estrategia recomiendan la utilización de los mapas estratégicos y el mismo Balanced Scorecard, unido a las metas e iniciativas estratégicas.

En la figura 12 se representa el desarrollo de una BSC como se explicó en el capítulo 2.3.3. Las flechas en el diagrama indican que es un proceso tanto top-down como bottom-up. Usualmente se construye la BSC comenzando desde el tope, pero en muchas ocasiones es igualmente útil comenzar desde las mediciones.

3. **Alinear la Organización** – se requiere sincronizar la estrategia de la compañía con las estrategias de las distintas unidades de negocio, a la vez que se alinea y motiva al personal para optimizar la estrategia de ejecución. La alineación de la organización con la estrategia se logra utilizando el efecto cascada de mapas y BSC en las distintas unidades organizacionales. También se logra alinear a todos los

empleados por medio de procesos de comunicación formales y ligando los objetivos personales e incentivos con los objetivos estratégicos. En lugar de realizar un único mapa estratégico para toda la organización, es común en estos casos que se realicen más de un mapa estratégico, generalmente un mapa por unidad de negocio [16].



**Figura 12 –Implementación de una estrategia de BSC**

El proceso de desarrollo comienza con el equipo de nivel de gerencia ejecutivo (senior executive management) trabajando juntos para trasladar la estrategia de la unidad de negocio en objetivos estratégicos. Una vez que se entienden los objetivos y las mediciones de más alto nivel es posible que los mismos sean refinados en otros niveles de operación. Pudiendo clarificar los objetivos de las distintas áreas funcionales. Por ejemplo, el objetivo denominado “Entrega a tiempo” (Delivery on Time - DOT) podría ser trasladado por una unidad de negocio operacional a “Reducir el tiempo de instalación de las maquinas”.

En esta etapa es donde se observa la importancia de BS como herramienta de comunicación entre las diferentes unidades de negocio y para estar seguro de que todas las acciones de la empresa están alineadas con los objetivos de largo plazo.

4. **Planificar las Operaciones** – la organización debe relacionar la estrategia de largo plazo con las operaciones del día a día, alinear la estrategia con los planes operacionales y los presupuestos a la vez que se debe enfocar en la mejora de aquellos procesos críticos para la estrategia. Requiere planear las operaciones

utilizando herramientas tales como Calidad Total, Gestión por Procesos, la reingeniería, el ABC (costeo basado en actividades), etc.

5. **Monitorear y aprender** – la organización se debe comprometer a monitorear los resultados del desempeño una vez que la estrategia ha sido desarrollada, planeada e implementada, permitiendo validar si esta ha sido correctamente ejecutada. Esta etapa requiere monitoreo y aprendizaje sobre los problemas, obstáculos y retos a que se enfrenta la organización. Este proceso integra información acerca de las operaciones y la estrategia en una estructura cuidadosamente diseñada para las reuniones de revisión gerencial. En esta fase interviene la utilización de los programas informáticos, los cuales deben tener una clara orientación hacia el seguimiento y la validación. La retroalimentación de la información de la BS hacia el nivel ejecutivo es muy importante ya que el feedback de monitoreo recibido sobre las mediciones permite ajustar la estrategia y de ser necesario, efectuar los cambios fundamentales necesarios a la estrategia original.

6. **Validar y Adaptar** – la organización debe también validar los supuestos fundamentales para determinar si tiene la estrategia correcta. Esto involucra validar y adaptar la estrategia, utilizando datos operacionales internos y datos de los cambios del entorno y la competencia, implica desarrollar un nuevo ciclo para integrar la planeación de la estrategia y la ejecución operacional. Los indicadores de los distintos objetivos permiten hacer los análisis de correlación que nos indicaran la relación causa efecto entre los distintos temas. Por ejemplo, reducir dramáticamente el tiempo de procesamiento de las órdenes, reducir el tiempo de desarrollo de los productos y mejorar las capacidades de los empleados, son objetivos que si bien no son los objetivos finales, se relacionan con los mismos a través de una serie de relaciones causa-efecto

Esta metodología compuesta por seis fases, ha tenido menor repercusión que el Cuadro de Mando Integral (CMI), en parte porque siempre es más complejo adoptar o cambiar las formas de trabajo ya que implican una transformación de la manera de hacer las cosas (cambio de los procesos).

## 2.4 Conclusiones del capítulo

La inteligencia de Negocios (BI) se basa en información diaria o histórica para obtener los resultados. No siempre una tendencia o comportamiento puede deducirse de la información histórica. Por ello BI por sí misma suele no ser suficiente para



desarrollar el área estratégica que desarrolla CPM. Sin embargo, las tecnologías de CPM y BI obtienen su mayor impacto y beneficio en su convergencia, pero es importante que esta convergencia se dé bajo un marco conceptual bien definido y bajo una arquitectura tecnológica integrada. Como se verá más adelante la arquitectura denominada ATID propuesta en esta tesina busca dar soporte para las decisiones estratégicas de manera independiente de BI teniendo en cuenta el marco conceptual descrito y ofreciendo una integración y uso de los sistemas de BI en los casos en que sea necesario.

CPM se enfoca en proveer la información adecuada a las personas adecuadas y en el instante oportuno para traducir las estrategias en planes, monitorear su ejecución y proveer un entendimiento profundo que mejore el rendimiento financiero y operacional. CPM es un área emergente que no cuenta con la estandarización de terminologías comúnmente asociadas a otras áreas. Por ejemplo, empresas como IBM, MetaGroup y Aberdeen definen el área como BPM, Gartner lo denomina CPM y compañías como PeopleSoft y Business Objects usaban el término EPM para referirse al área.

El BSC es un sistema de gestión estratégica de la empresa, que consiste en:

- Formular una estrategia consistente y transparente.
- Comunicar la estrategia a través de la organización.
- Coordinar los objetivos de las diversas unidades organizativas.
- Conectar los objetivos con la planificación financiera y presupuestaria.
- Identificar y coordinar las iniciativas estratégicas.
- Medir de un modo sistemático la realización, proponiendo acciones correctivas oportunas.

El Mapa Estratégico es un sistema de comunicación, información y aprendizaje. Son una representación visual de la estrategia de una organización y demuestran claramente por qué una imagen es más poderosa que mil palabras.

La elaboración de un Mapa Estratégico debe originarse en la visión y estrategia de la empresa, para luego entrar a definir los factores críticos necesarios para poder alcanzar el éxito empresarial. Los indicadores de gestión que utiliza el Mapa Estratégico le ayudarán a la empresa a medir los objetivos y a definir las áreas críticas en la implementación de la estrategia. El concepto de Mapa Estratégico sirve

de soporte a la planificación estratégica, pues busca ajustar las acciones de todos los miembros de la organización a los objetivos y a facilitar la ejecución de la estrategia.

En resumen la evolución del BSC nace con el desarrollo de un conjunto de indicadores. Muchos consideraron que era suficiente con tener objetivos en las cuatro perspectivas. La segunda generación planteaba que eso era importante pero no suficiente si no se contaba con una organización orientada por una estrategia. Posteriormente evoluciona el esquema a una tercera generación en donde se desarrollaron una serie de relaciones causa efecto descritas en los mapas estratégicos que cuentan la estrategia sobre el que y el cómo. La cuarta generación nace con la necesidad de que toda la organización esté alineada con la estrategia y por último nos llega la necesidad de alinear la estrategia con los planes y procesos operativos, esta es la que se interpreta como la V generación, la cual de seguro no será la última.

### 3. Estado del arte de las herramientas

En esta sección se describen tanto las herramientas existentes actualmente para resolver los problemas propuestos del área de CPM así como también el estado del arte de las tecnologías basadas en la plataforma eclipse y que serán utilizadas para ilustrar la implementación de los componentes de la arquitectura de software ATID presentada más adelante en esta tesina.

#### 3.1 Herramientas para CPM

Se comenzará analizando las características dadas a conocer por las empresas que ofrecen alguna herramienta de software que soporte el área de CPM. En particular se listan solamente aquellas que muestran mecanismos gráficos de generación, edición y visualización de los mapas estratégicos en representación del uso de la técnica de BSC. Este análisis se usa al final de esta sección para clasificar las herramientas según sus características en tres grandes grupos. Esta clasificación será importante para entender el posicionamiento de la arquitectura ATID presentada en el capítulo 4.

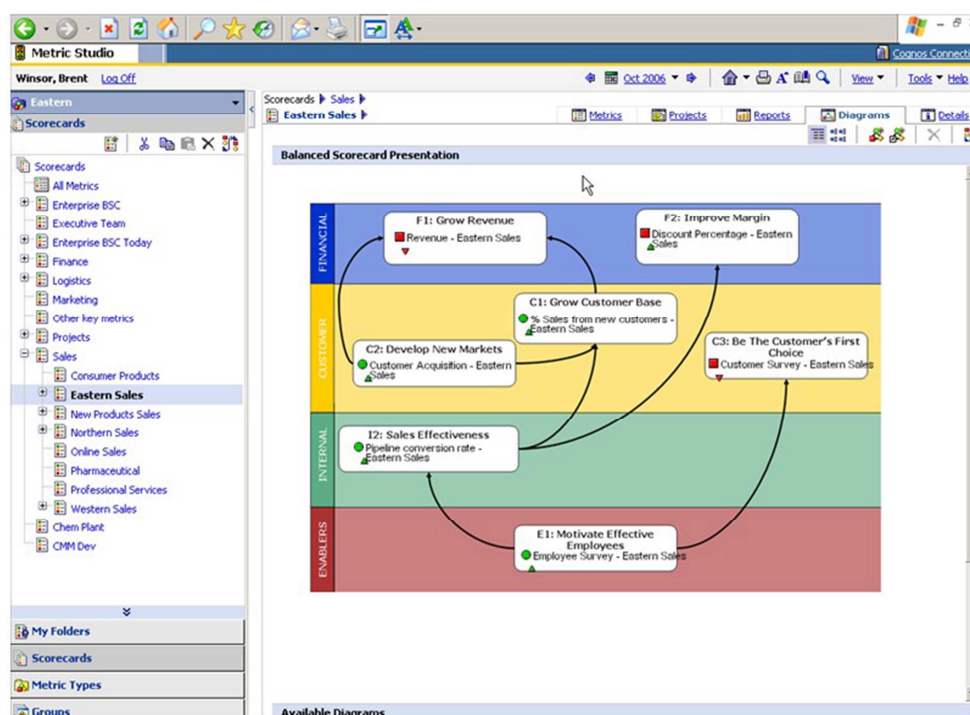


Figura 13 - IBM Cognos 8 Business Intelligence Scorecarding

**IBM Cognos 8 Business Intelligence Scorecarding** – El enfoque que le da IBM a la técnica de BSC mediante la suite de Cognos es como complemento de su propuesta de BI. Ofrece interfaces que se ejecutan en el navegador web. Permite crear mapas estratégicos y asociar métricas claves a los objetivos. Las métricas

pueden tener distintas fuentes de datos ofreciendo principalmente facilidades para conectarlas con los sistemas de BI. Todo está integrado en una única arquitectura propietaria. La extensibilidad se basa en las personalizaciones de las bases de datos y codificación de bajo nivel. No se encontraron mecanismos de modelación asistida por modelos (Model Driven Development – MDD). La instalación es tan compleja como un típico sistema de BI. Los requerimientos de hardware son bastante importantes así como los conocimientos requeridos para poder instalar el software. [18][44].

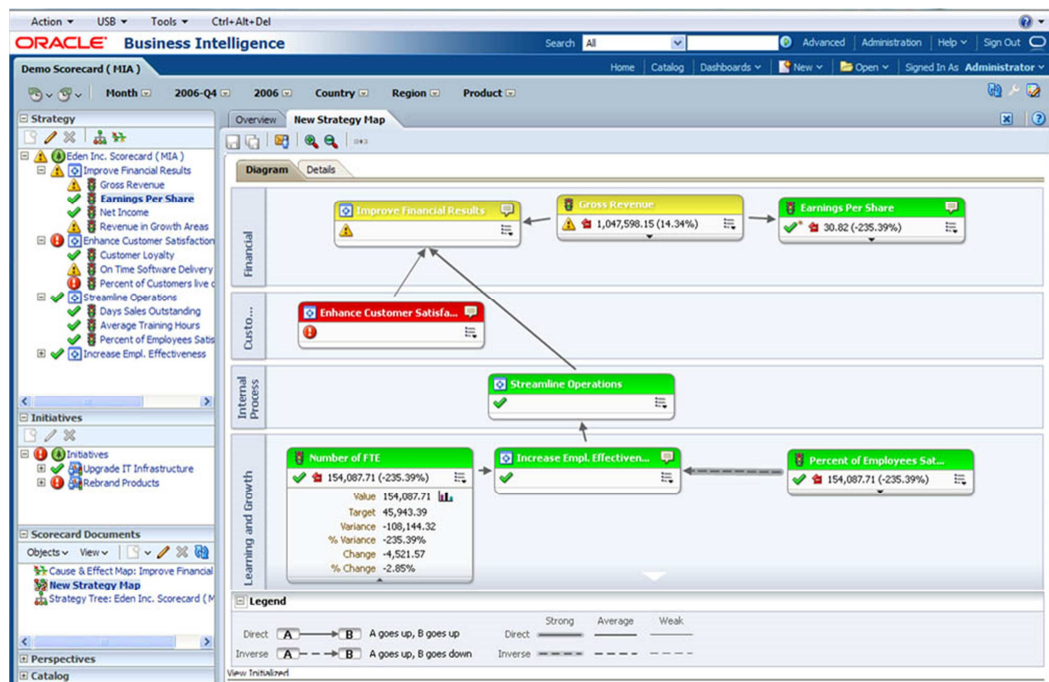


Figura 14 - Oracle Scorecard and Strategy Management

**Oracle Scorecard and Strategy Management.** Este módulo extiende Oracle BI Enterprise Edition (OBIEE). Se integra con Oracle Business Intelligence Suite 11g. Similarmente a la solución de IBM tiene la particularidad de que no es posible ejecutar independientemente la parte de estrategia de la de BI. También es una solución principalmente web. La instalación requiere de conocimientos avanzados. La arquitectura es propietaria. Posee mecanismos limitados de extensibilidad dado principalmente a que posee librerías (dll) compiladas con APIs no muy bien definidas. Solo posee algunos mecanismos de personalización de funcionalidades para adaptarlas a casos particulares. La interacción con otras herramientas se logra a través de servicios web y es compleja [w4][w5]

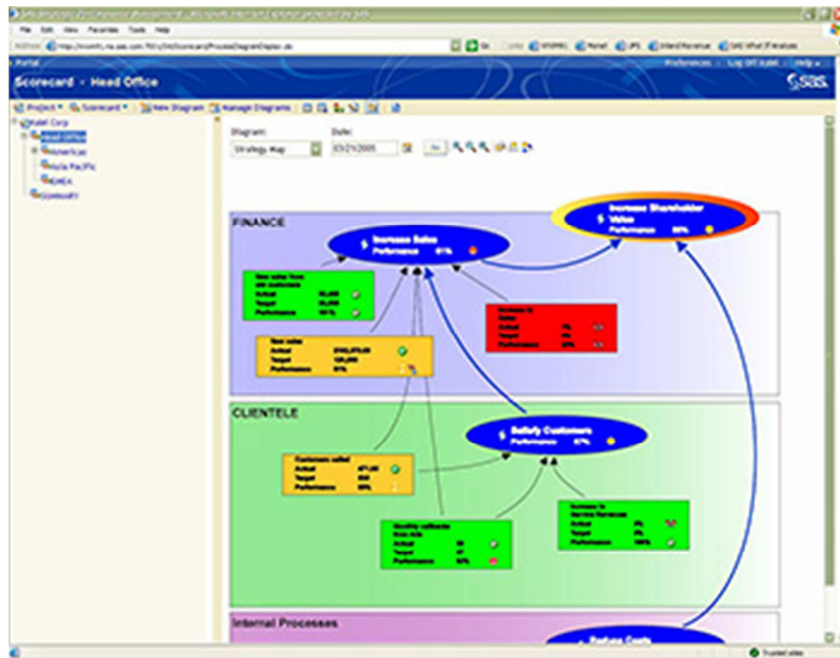


Figura 15 - SAS SPM Strategic Performance Management

**SAS SPM Strategic Performance Management.** Se integra con SAS Business Analytics, el equivalente de SAS para el área de BI. El acceso a las herramientas es a través del navegador web. Los mapas estratégicos son creados por diseñadores con herramientas especializadas de escritorio para luego ser publicados como reportes con valores obtenidos en tiempo real al que acceden los usuarios a través del navegador web. Soluciones personalizadas son difíciles de obtener sin soporte de SAS. Provee su propio lenguaje de macros para modelación. [33][w6][w7][w8]

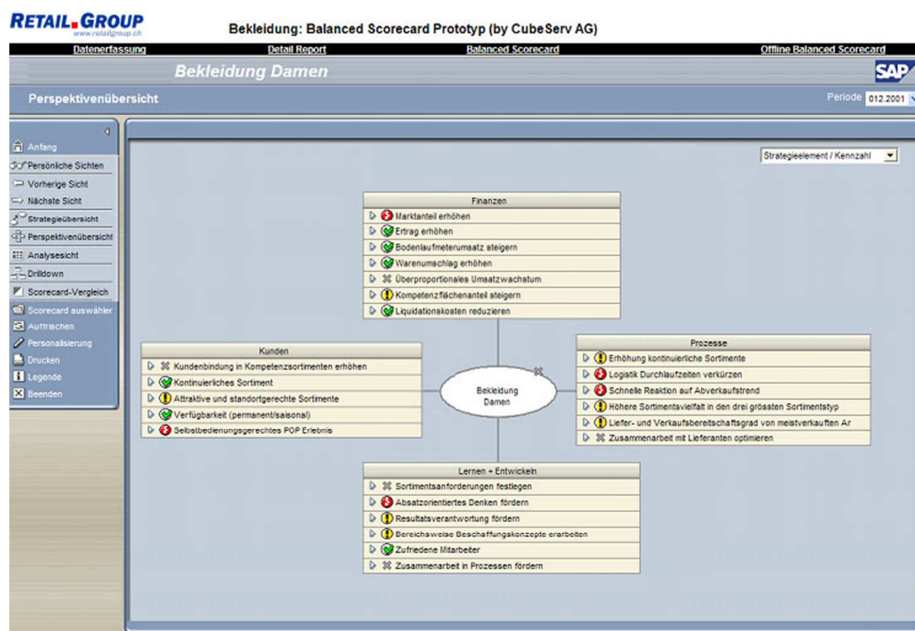
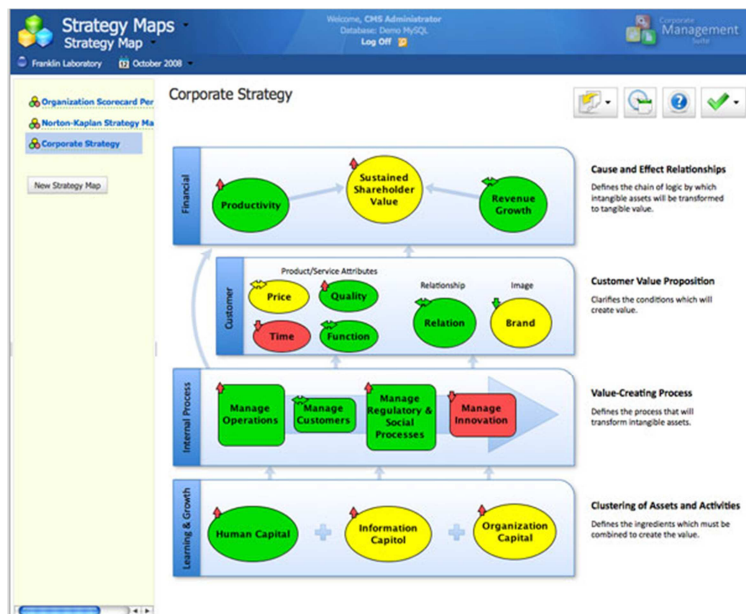


Figura 16 - SAP Corporate Performance Monitor

**SAP Corporate Performance Monitor (CPM)** Es otro de los líderes en el área. En este caso se implementa como un módulo que utiliza aplicaciones web y que corren sobre un servidor denominado SAP Web Application Server. Puede ser integrado con cualquier otro módulo de SAP, incluyendo BI, Finanzas (FI), Control (CO), Manejo de Materiales (MM), Manejo de Recursos Humanos (HRI), Supply Chain Management (SCM), Supplier Relationship Management (SRM) y otros. Permite la creación de perspectivas, objetivos, iniciativas y métricas. Los requerimientos de hardware al igual que las herramientas vista hasta ahora son bastante importes. SAP propone la plataforma NetWeaver como plataforma de integración. Es conocida como una aplicación orientada a servicios sobre la que se construyen el resto de las aplicaciones. [45]



**Figura 17 - Spider Strategies**

**Spider Strategies** - Provee un conjunto de herramientas que corren en navegadores web y permiten la creación de mapas estratégicos y tableros de control así como también provee interfaces web para poder importar data de otros sistemas (a través de Excel, o por servicios web). Los mapas estratégicos se acceden como imágenes no posee editores para los mismos. Además uno de los problemas de sistemas como este es la pérdida de sincronismo entre la herramienta y las fuentes. No brindan una descripción de la arquitectura o extensibilidad subyacente. No se comentan mecanismo de colaboración o comunicación. [w9]

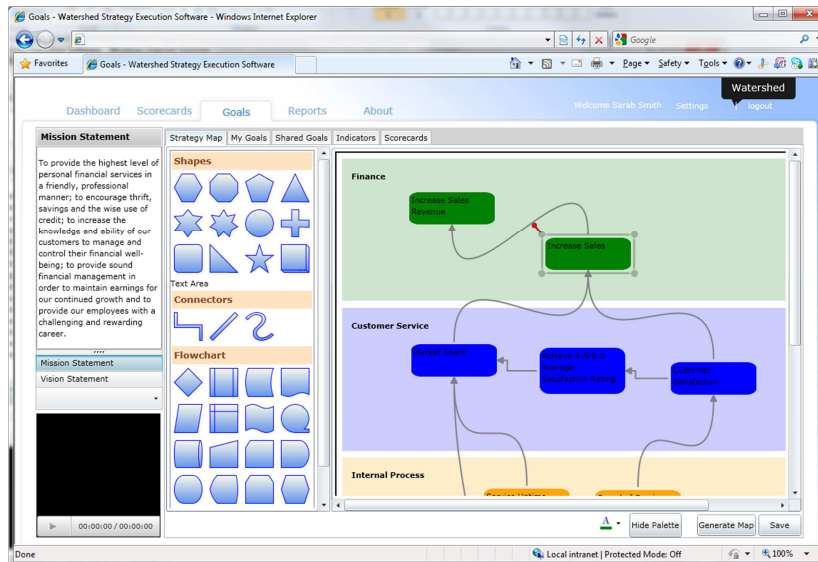


Figura 18 - Watershed Strategy Execution

**Watershed Strategy Execution** - Similar a la herramienta anterior pero provee un editor primitivo de mapas estratégicos que corre en el navegador. Menciona soporte para múltiples usuarios. No provee una descripción de la arquitectura ni se detalla ningún mecanismo de conectividad con fuentes de datos. Funcionalidad bastante limitada y acotada. [w10]

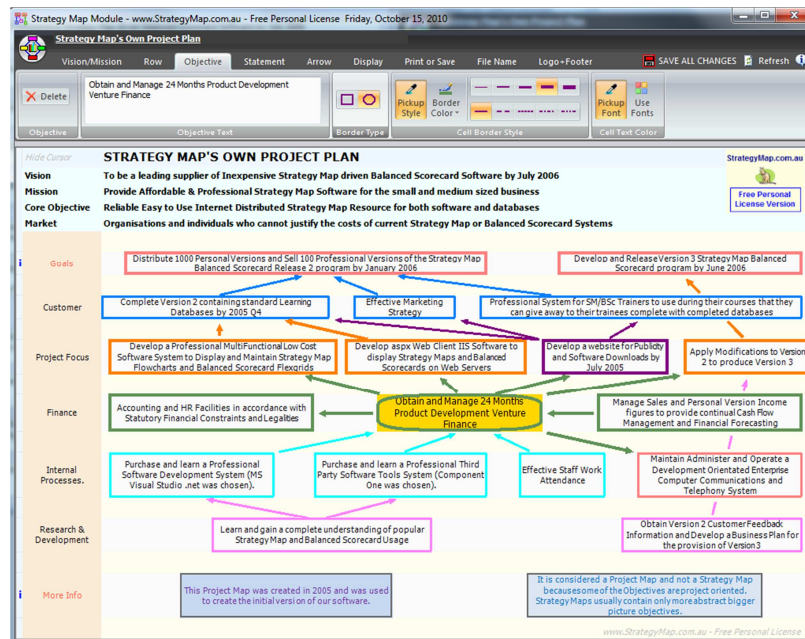


Figura 19 - Strategy Map Balanced Scorecard

**Strategy Map Balanced Scorecard** – Es una herramienta de escritorio creada con el apoyo del Balanced Scorecard Institute. Es una de las pocas que es fácil y rápida para instalar aunque esto es principalmente debido a su simplicidad. Está

fuertemente orientada a la creación de los mapas estratégicos pero posee muchos detalles de usabilidad (como la dificultad para encontrar las acciones para realizar las tareas) y no provee capacidades de extensibilidad. Tampoco presenta una arquitectura extensible o integrable con otras herramientas y la funcionalidad es bastante limitada. Permite crear los mapas estratégicos pero con un editor semi-gráfico.

### 3.1.1 Clasificación de las Herramientas de CPM

En base a las herramientas analizadas, se creó a manera de resumen una clasificación de las mismas en base a ciertas características en común. Esta clasificación será de utilidad en el capítulo 4 para posicionar a la arquitectura ATID allí propuesta. Del análisis anterior se puede deducir que en general hay tres grandes tipos de herramientas (Figura 20).

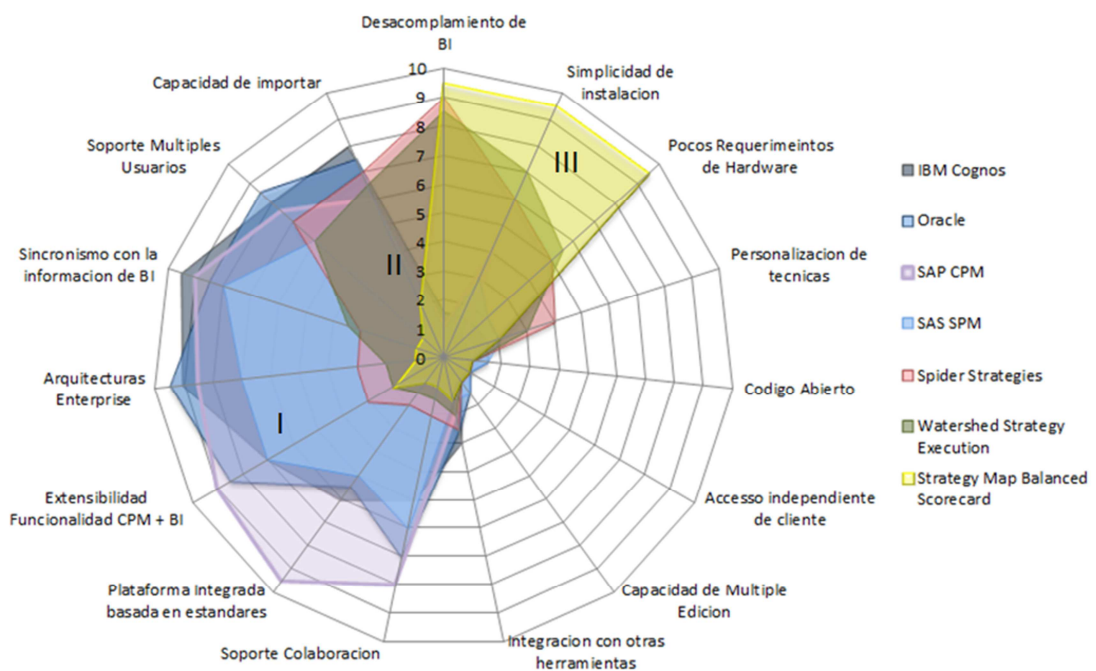


Figura 20 - Caracterización de las Herramientas

**Tipo I** – Son las que ofrecen CPM como una extensión de su propuesta de BI. Este grupo está representado por las grandes empresas (IBM, SAP, ORACLE, SAS). En general son herramientas con procesos complejos de instalación, no se integran bien con otras tecnologías no propietarias, mecanismos propios y cerrados de extensibilidad. Con requerimientos altos de hardware para instalación aunque accedidas a través del navegador. Capacidades de múltiple edición limitadas. En general, o tienen acceso web o tienen acceso a través de herramientas de escritorio



pero no ambas al mismo tiempo. Otra característica de este grupo es que las empresas ofrecen un paquete, no es posible simplemente conseguir solo la solución de BSC. SAP si bien sigue dentro de este grupo, es la única que menciona o propone una plataforma integrada basada en estándares del tipo SOA (orientados a servicio) lo cual lo hace poseer una arquitectura más abierta a la integración aunque aún es propietaria.

**Tipo II** – Son soluciones más pequeñas que solo ofrecen BSC con capacidades de importar datos de métricas de otras fuentes pero sin sincronismo. No funcionan tan bien en tiempo real. El sincronismo puede ser un problema. Poseen arquitecturas propietarias y no comentan mecanismos de extensibilidad o personalización. Permiten el acceso de múltiples usuarios pero no presentan mecanismos de colaboración o comunicación claros.

**Tipo III** - Las que ofrecen solo CPM (experimentales). Son herramientas de escritorio generalmente monousuario. No tienen una arquitectura que sustente un modelo distribuido. Problemas de usabilidad importantes. Dentro de este grupo clasifica la herramienta Strategy Map Balanced Scorecard y otras que no se incluyeron en el análisis debido a sus escasas prestaciones

### **3.2 Tecnologías basadas en eclipse**

En esta sección se describe brevemente las principales características de las distintas tecnologías basadas en la plataforma eclipse y que serán relevantes para la implementación de los componentes de la arquitectura presentada en el capítulo 4.

Eclipse surgió como un ambiente de desarrollo pero gracias a sus principios de extensibilidad rápidamente se convirtió en una plataforma de desarrollo de software. Esto significa que su arquitectura me permite a través de mecanismos de extensión denominados *plugins*, armar personalizaciones de herramientas de todo tipo. Esto permitió la creación de numerosos proyectos diferentes construidos en base a esta extensibilidad que cubren las más diversas áreas. Hoy en día es posible encontrar más de 1000 proyectos diferentes que proporcionan funcionalidades en áreas tan diversas como bases de datos, dispositivos móviles, modelación, arquitecturas orientadas a servicios (SOA), herramientas web y muchas más. Los principales conceptos relacionados con la arquitectura y evolución de la plataforma eclipse y los principales plugins existentes de interés para la tesina se detallarán a continuación.

### 3.2.1 Rich Client Platform (RCP)

Como se mencionó anteriormente Eclipse es un proyecto de desarrollo software de código abierto, cuyo propósito es proporcionar una plataforma de herramientas altamente integradas. El trabajo en Eclipse consiste en un proyecto central que incluye un framework genérico para la integración de herramientas. A esta parte central se la denomina comúnmente como Rich Client Platform (RCP) o "Aplicaciones de Cliente Enriquecido", opuesto a las aplicaciones "Cliente-liviano" basadas en navegadores y para diferenciarlas de los distintos ambientes de desarrollo (IDE) ofrecidos, que están contruidos sobre la plataforma eclipse.

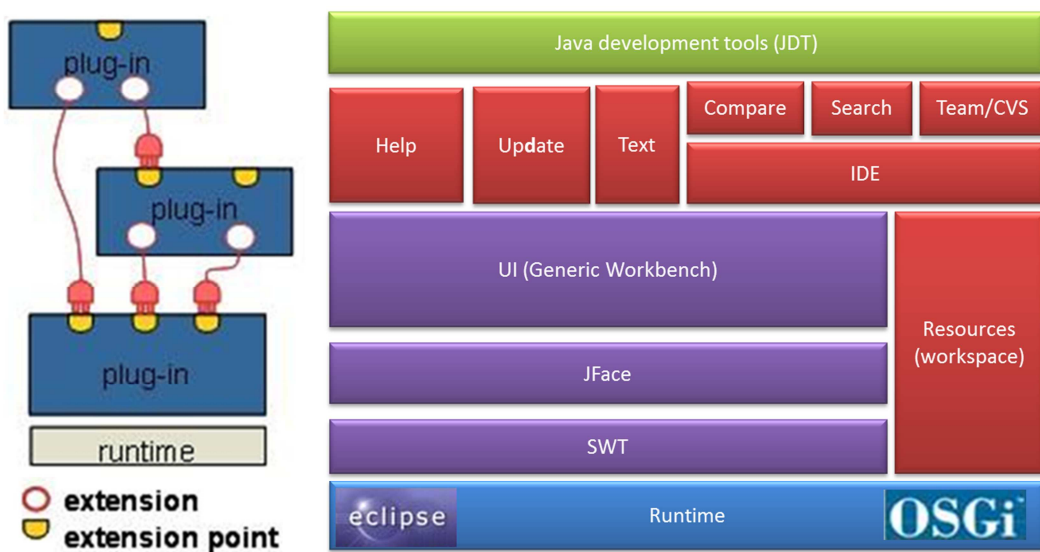


Figura 21 – La Plataforma Eclipse

La unidad básica de función o componente se denomina plug-in. La plataforma eclipse en sí misma y las herramientas que la extienden se componen de estas unidades de extensibilidad. Una sola herramienta puede consistir en un único plug-in, pero en general herramientas más complejas se dividen típicamente en varios. Desde una perspectiva de empaquetado, un plug-in incluye todo lo necesario para ejecutar un componente, como código Java, imágenes, texto traducido, etc. Una aplicación típica en eclipse se construye a partir del RCP extendiéndolo con estos componentes que aportan las nuevas funcionalidades. Las herramientas integradas en Eclipse suelen trabajar con carpetas y archivos ordinarios, pero utilizando una API de más alto nivel, basada en recursos y proyectos. Estos residen en un espacio de trabajo denominado *workspace* el cual puede verse como un pequeño repositorio de recursos local.

Ejemplos de aplicaciones construidas en eclipse RCP incluyen aquellas ofrecidas por IBM como WebSphere, Lotus (Notes, Sametime, Symphony) y Rational, u otras empresas importantes como SAS, NASA (Simulaciones de misiones), Oracle, Adobe, entre otras.

### **3.2.2 Eclipse Modeling Framework (EMF)**

La Ingeniería Dirigida por Modelos (IDM) es el último paso en la tendencia a elevar el nivel de abstracción en el que se diseña y construye el software. Así, en un proceso de desarrollo dirigido por modelos se parten de modelos con un alto nivel de abstracción que servirán para especificar el sistema obviando detalles tecnológicos. Dichos modelos serán refinados semi-automáticamente por medio de transformaciones de modelos, hasta que su nivel de abstracción permita utilizarlos para generar el código final.

El *Eclipse Modeling Framework* (EMF) es un framework de modelado para eclipse y generador de código que se ha transformado en un componente clave dentro de las aplicaciones generadas sobre la plataforma. Esto es debido principalmente a las capacidades que posee para generar rápidamente implementaciones a partir de especificaciones de metamodelos de alto nivel expresadas como diagramas UML. El código generado por el framework de EMF provee soporte para notificaciones, persistencia, comandos, reflexión y acceso a metadatos, enganche con componentes gráficos, entre otros. Además existen una serie de subproyectos para EMF que aportan extensiones como soporte para comparación entre modelos, transacciones, persistencia en bases de datos, soporte para objetos distribuidos, indizado y consultas, validación, etc. Otras tecnologías relacionadas basadas en EMF incluyen implementaciones de los estándares UML2 (Unified Modeling Lenguaje), OCL (Object Constraint Lenguaje), BPMN2 (Business Process Model and Notation), entre otras.

Ejemplos de aplicaciones que fueron generadas usando EMF incluyen componentes en las principales herramientas de IBM (Rational, Webshpere, Lotus, Jazz). El uso de EMF también resulta clave para otros subproyectos importantes como GMF que se detallara en la siguiente subsección.

### 3.2.3 Frameworks para Editores Gráficos (GEF, GMF, Graphiti)

Una característica importante de la plataforma eclipse fue el surgimiento de una serie de extensiones (plugins) que proveía un entorno que facilitaba la construcción de editores con capacidades gráficas. Este tipo de frameworks es una de las carencias grandes de hoy en día en las aplicaciones web que corren sobre navegadores.

Uno de los primeros frameworks en surgir fue el denominado Graphical Editor Framework (GEF) que es un plugin de eclipse para el desarrollo de editores gráficos que pueden ir desde procesadores de texto wysiwyg hasta editores de diagramas UML o interfaces gráficas para el usuario (GUI), etc. Permite hacer de manera relativamente sencilla diagramas de todo tipo: UML, circuitos lógicos, diagramas objeto-relacionales, diagramas de flujo, etc. Provee mecanismos para dibujar distintos tipos de figura, conexiones, drag-n-drop, redimensionado de figuras, eventos, cursores, etc. Utiliza el paradigma MVC (Modelo Vista Controlador) donde la Vista y el Controlador son proveídos por GEF mientras que el modelo lo debe implementar el programador. Generalmente para facilitar la construcción se utiliza EMF para la generación de los modelos mientras que manualmente se debe codificar los controladores y vistas gráficas. El desarrollo de editores gráficos utilizando GEF y EMF es muy común debido a las ventajas que proveen ambos frameworks y es utilizado por gran cantidad de empresas. Debido a la complejidad de la unión entre EMF y GEF surgió recientemente el denominado Graphical Modeling Framework (GMF).

GMF es un proyecto de eclipse que aporta principalmente dos cosas: por un lado enriquece GEF con una serie de capacidades como mejores visualizaciones, soporte de transacciones, etc. En segundo lugar provee un puente generativo entre GEF y EMF. Concretamente, GMF es un plug-in de eclipse que proporciona un componente generativo y una plataforma de ejecución para el desarrollo de la infraestructura de editores gráficos basados en EMF y GEF. Como vimos anteriormente EMF permite la generación de código eficiente partir de una representación de UML con soporte para notificaciones, persistencia, etc. GEF permite la creación de editores con funcionalidades gráficas. Lo que permite GMF es la generación de editores gráficos describiendo una serie de modelos de alto nivel, y generando el código necesario que conecta un editor GEF con el código generado por EMF a partir de la representación de sus modelos.



5. Definición del modelo que permite especificar los componentes de la paleta de herramientas del editor (contenida en un archivo con extensión `.gmftool`).

6. Generación del modelo de correspondencias (`.gmfmap`) entre los elementos gráficos, los controles de la paleta y los elementos del metamodelo de nuestro DSL. Para ello GMF proporciona un asistente que genera este modelo a partir de los anteriores.

7. Es habitual que las correspondencias recogidas en el modelo generado no sean correctas, con lo que es necesario refinar dicho modelo manualmente.

8. Generar el modelo generador de GMF (`.gmfgen`) a partir del modelo de correspondencias.

9. Generar el código Java que implementa el diagramador.

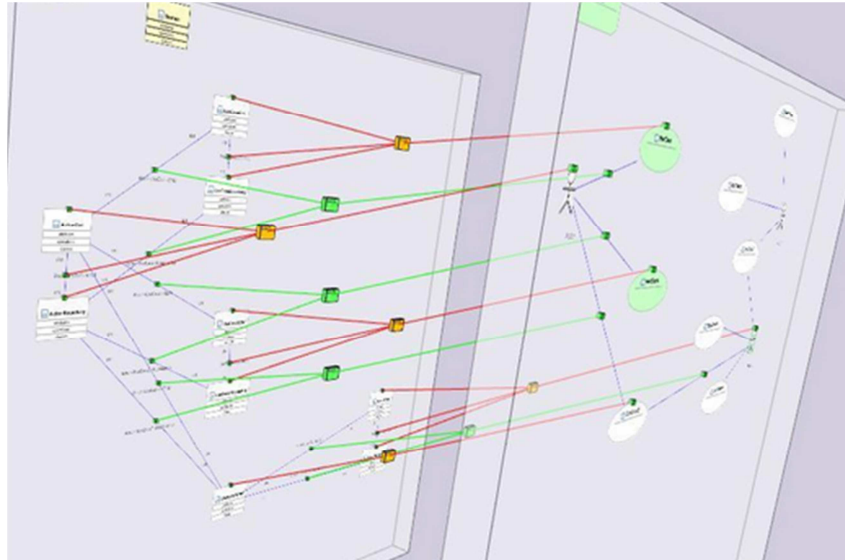
Al igual que GEF, GMF también puede aplicarse a variados dominios, por ejemplo, editores de UML, editores MINDMAP, etc.

En resumen, el proceso de generación de editores con GMF consiste en especificar un conjunto de modelos a partir de los cuales GMF generará automáticamente el código Java que implementa el diagramador y que podrá ser utilizado como un nuevo plugin de Eclipse

Por último, si bien GMF intenta resolver la problemática inherente a la generación de editores gráficos basados en GEF y EMF, el código resultante resulta aún muy complejo. Debido a ello es que están surgiendo otros frameworks como por ejemplo, el denominado Graphiti soportado principalmente por gente de SAP, que tienen como uno de sus objetivos ocultar los detalles complejos de las implementaciones ofreciendo una API mucho más simple. Además otro problema común de GMF es la desincronización más frecuente entre el código generado y las representaciones de nivel superior, por ello, si bien Graphiti acepta EMF como lenguaje de modelación, no sigue un enfoque generativo para el resto de la creación de los editores. [W11]

Un desarrollo relacionado que causo mucho interés en los tiempos en que se elaboró la propuesta de esta tesina fue el proyecto GEF 3D. Esta serie de plugins, era visto como el siguiente paso en la evolución de este tipo de las capacidades de este tipo de editores y permitía la representación de objetos en un espacio n-dimensional. Una de las posibles aplicaciones era la conexión de elementos de distintos diagramas como se muestra en la figura 23. [W12]

Lamentablemente hace un par de meses el proyecto GEF 3D fue puesto en suspenso debido a que solo soportaba un número limitado de controladores gráficos y además se encontraron problemas con las licencias de las librerías que permitirían utilizar implementaciones de controladores gráficos de código abierto.[W13]



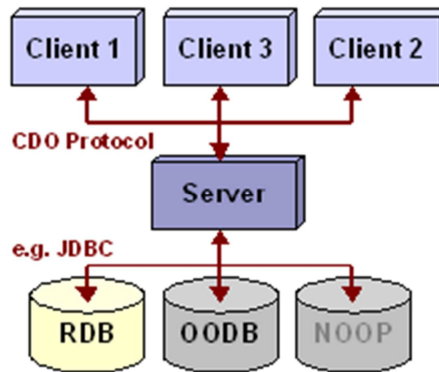
**Figura 23 – Ejemplo de aplicación de GEF 3D**

### **3.2.4 Soporte de modelos distribuidos (CDO y Dawn)**

EMF fue un proyecto exitoso de eclipse originalmente pensado para soluciones de escritorio con persistencia en archivos. Debido a sus ventajas y popularidad comenzó a ser usado en aplicaciones cada vez más complejas. En este contexto comenzaron encontrarse problemas que debían ser resueltos para que el uso de EMF continuase siendo viable, sobre todo en las implementaciones de modelos distribuidos. Las principales necesidades que surgieron fueron las siguientes:

1. Los modelos crecen al punto de que no entran en memoria.
2. Los recursos son demasiado grandes para residir en archivos.
3. La aplicación tiene que estar disponible para múltiples usuarios al mismo tiempo.
4. Es necesario poder hacer lock de objetos y ejecutar acciones como transacciones sobre un conjunto de objetos.
5. Son necesarios mecanismos que permitan que siempre se esté observando el estado corriente de los objetos.
6. Debería poder ser posible volver a cualquier momento de la historia de un modelo.

En este contexto se creó el sub-proyecto CDO (Connected Data Objects) que extiende de manera transparente a EMF permitiendo resolver las problemáticas descritas anteriormente. CDO tiene una arquitectura de 3 capas soportando aplicaciones cliente basadas en EMF, un server como repositorio central y un sistema intercambiable de persistencia en distintos tipos de base de datos (Figura 24).



**Figura 24 – Arquitectura de tres capas de CDO**

Por un lado CDO extiende los modelos EMF dando soporte para estados distribuidos. Cada objeto tiene un estado interno (clean, new, dirty, etc) sincronizado respecto del servidor y que es actualizado automáticamente mediante los mecanismos de notificaciones estándares de EMF basados en el patrón Observer. Por otro lado incorpora funcionalidades que permiten soportar conceptos típicos de aplicaciones cliente-servidor como son sesiones, vistas, transacciones, etc. También generaliza el mecanismo de notificaciones de EMF dando soporte para recibir información sobre eventos ocurridos remotamente y además incorpora conceptos como soporte para Branch (para trabajar con copias del modelo en el repositorio) o Auditoria (guardando versiones de los objetos de manera de poder recuperar la historia de los mismos). De este modo CDO se ha transformado en la evolución natural de EMF para aplicación en un ámbito de múltiples usuarios, y con una arquitectura escalable, extensible y flexible.

Por último, ha surgido recientemente el proyecto Dawn relacionado con la posibilidad de utilizar editores gráficos basados en GMF sobre modelos CDO. Así como GMF se construye sobre EMF, sobre CDO se construye el proyecto Dawn que generaliza GMF permitiendo la creación de editores gráficos con almacenamiento distribuido de los modelos, de modo que múltiples usuarios desde distintas máquinas pueden operar gráficamente y simultáneamente sobre el mismo recurso al mismo tiempo. La desventajas de CDO están relacionadas principalmente con que aún no



tiene soporte para login de usuarios ni presenta una arquitectura que permita acceder el repositorio desde múltiples fuentes (navegadores web o clientes de escritorio) utilizando por ejemplo servicios web.

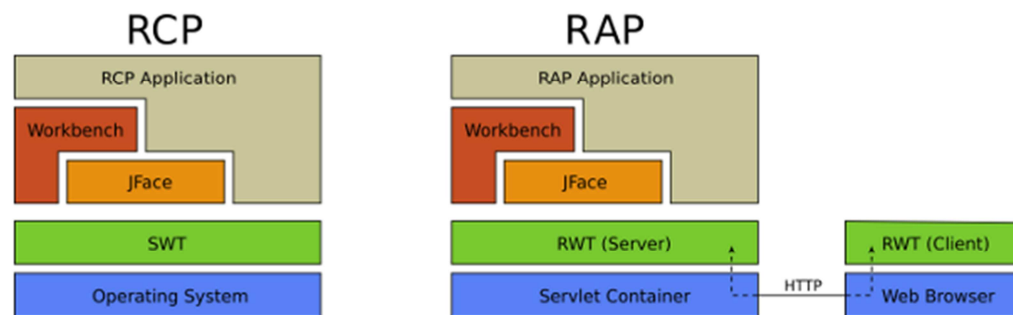
### 3.2.5 Eclipse 4

Hasta aquí se mencionaron las principales características de eclipse como plataforma para aplicaciones de escritorio. Además de los plugins aquí descritos, existen un gran número de extensiones de la plataforma en los más diversos ámbitos. Las últimas tendencias en cuanto a la evolución futura de la misma están dadas por la creciente necesidad de poder desarrollar aplicaciones en otros contextos, como navegadores web, teléfonos celulares o tablets.

Dentro de este contexto, existen proyectos en eclipse que dan soportes a distintas áreas como Eclipse SOA (para arquitecturas orientadas a servicios). Pulsar (para aplicaciones móviles) entre otras. Las tendencias futuras en la evolución de eclipse intentan reusar los conceptos existentes desarrollados en la plataforma, permitiendo la fácil y rápida migración para ejecución en otros contextos. En estos últimos meses han tomado forma 2 nuevas tecnologías que están siendo desarrolladas en eclipse y que atacan muchos de los problemas descubiertos durante el transcurso de esta tesina:

**Rich Ajax Platform (RAP)** – Es un sub-proyecto, que intenta permitir correr aplicaciones basadas en eclipse en el navegador, con los mínimos cambios posibles en el código fuente. La ventaja es que reusa todos los conceptos de extensibilidad de eclipse. La desventaja es que el modelo de eclipse no fue pensado para ejecutarse en un contexto web donde los requerimientos pueden ser distintos que los de las aplicaciones de escritorio, es decir, que no tiene una arquitectura que lo sustente, sino que más bien intenta reescribir algunos componentes de visualización de eclipse. Una característica interesante que introduce es el de Fuentes Únicos (Single Sourcing). Esto tiene dos características, por un lado para los programadores con experiencia en eclipse y RCP implica que pueden reusar sus habilidades y conocimiento sin mayores cambios. Por otro lado, el mismo código puede servir para crear una aplicación web o una de escritorio. Otro uso interesante del enfoque de Código Único podría ser por ejemplo el siguiente: hoy en día, debido a limitaciones tecnológicas, los editores gráficos basados en GMF no poseen una forma de ejecutarse en navegadores web, sin embargo, la funcionalidad no grafica si podría visualizarse y editarse desde un

navegador, por ello sería posible reusando el mismo código fuente para poder acceder a la parte común de la aplicación tanto desde una aplicación de escritorio ejecutándose en una laptop como desde un navegador web, mientras que los componentes que aportan los editores gráficos serían agregados solo a la versión de escritorio. RAP puede verse como una posible generalización de RCP para aplicaciones web reemplazando el componente de dibujo de interfaces gráficas SWT por uno denominado RWT que dibuja los componentes gráficos remotamente en el navegador web. [W13]



**Figura 25 - Arquitectura de RAP**

e4 – La generación actual de eclipse (las versiones 3.x) están fuertemente orientadas al desarrollo de aplicaciones de escritorio. La nueva generación (denominada e4) intenta incorporar una serie de conceptos que buscan llevar a eclipse al siguiente paso como framework para la construcción de aplicaciones e interfaces de usuario. La primera generación de eclipse (release 1.0 a 2.1) fue principalmente una plataforma de integración. Su principal fortaleza fue la de permitir poner juntos componentes (plugins) escritos por distintos autores, de modo de poder integrarlos en una aplicación común con una experiencia de usuario consistente. La segunda generación de eclipse (los reléase 3.X) estuvieron marcados por el runtime OSGi, el cual lo convirtió en un framework de aplicaciones basados en componentes de propósito general. Esta segunda generación tuvo buena escalabilidad para crear desde pequeñas aplicaciones embebidas, hasta grandes aplicaciones RCP y servidores web. Sin embargo, cada componente (plugin) típicamente es muy difícil de reusar fuera del ambiente específico para el cual fue diseñado y testado. Puede ser relativamente fácil agregar o remover componentes del sistema, pero es a menudo muy dificultoso tomar un componente diseñado para una aplicación y reusarlo en una ambiente o aplicación completamente diferente. Uno de los focos de la generación 4 de eclipse (e4) es hacer mucho más fácil la escritura de componentes que sean más

reusables, con menos dependencias y configurables para ser ejecutado en un gran rango de aplicaciones y ambientes de forma de permitir su uso en otros contextos y lenguajes. Entre los caminos de exploración actuales de e4 se pueden mencionar:

1. Un modelo de programación más orientado a servicios, basado en OSGi, que provea una mejor aislación para los plugins del ambiente que los rodea.
2. Una modelo de definición de GUI basado en XML y con soporte de estilos similar a las aplicaciones web (CSS)
3. Soportes para otros lenguajes y paradigmas como Javascript u OpenSocial.
4. Un modelo de administración de recursos más flexible.

### **3.2.6 Jazz**

Como vimos anteriormente, las tendencias respecto de eclipse como plataforma de desarrollo de aplicaciones están siendo influenciadas por grandes presiones para adaptarse a otros contextos en donde es necesario poder contar con arquitecturas que soporte el acceso a través de navegadores web. Vimos que RAP permite en parte soportar los mecanismos de extensibilidad de eclipse (plugins) pero accediéndolos dentro de un navegador web. Sin embargo RAP no resuelve los conceptos subyacentes inherentes a las arquitecturas de estos tipos de aplicaciones, con acceso de múltiples usuarios.

A partir de 2005, mientras IBM desarrollaba una nueva suite de herramientas de desarrollo de aplicaciones basadas en arquitecturas modernas, denominada RTC (Rational Team Concert), se vio el potencial de su visión como posible generalización de la plataforma eclipse para la aplicación en ambientes empresariales. De esta forma surgió *Jazz* como una plataforma para el desarrollo de aplicaciones para entornos colaborativos que podían ser accedidas tanto desde navegadores web como desde aplicaciones de escritorio. En esta sección se describirá brevemente las principales características más destacables de la visión original de Jazz, dejando para la sección 5 de implementación los detalles de todos los problemas encontrados en la implementación actual de Jazz.

Entre las principales características de la plataforma jazz se pueden mencionar:

1. Jazz tiene una arquitectura cliente-servidor orientada a servicios. En su visión define un conjunto de servicios básicos del lado del servidor denominados “*Jazz Foundation Services*” que abarcan manejo de usuarios (seguridad y acceso), soporte de áreas de proyectos y repositorio (persistencia), soporte de

roles, procesos o métodos y estructura de equipos, soporte de indexación, servicios de búsquedas y dialecto de creación de consultas, entre los más importantes.

2. Las aplicaciones extienden el servidor de Jazz definiendo nuevos *Componentes*. Un componente puede contribuir al servidor de Jazz opcionalmente con un modelo y con los nuevos servicios que expone.
3. Jazz permite que los clientes de escritorio sigan siendo construidos con la arquitectura de plugins anteriormente descritas, ofreciendo un conjunto de plugins que se instalan en los clientes y que permiten acceder a los servicios del servidor de manera transparente para el programador.
4. Jazz conserva la arquitectura de extensibilidad basada en plugins aun en el servidor. El servidor de Jazz se extiende con el mismo tipo de plugins que se venían utilizando en eclipse. Esto permite que ciertos componentes comunes puedan ser reusados tanto en los clientes como en el servidor. Por ejemplo, los modelos asociados con los distintos componentes, se generan usando una versión de EMF personalizada por Jazz. Los plugins que contienen la implementación de estos modelos, pueden ser reusados tanto del lado del cliente como del servidor permitiendo el tratamiento idéntico de los objetos en ambos lados.
5. El repositorio de jazz soporta tanto ítems simples como ítems auditables. Esto lo logra implementando un data warehouse para guardar la información histórica sobre los cambios en los ítems.
6. Un concepto interesante que también intenta de presentar Jazz aunque bastante limitado es el de obligar la definición de un método subyacente aplicado a cada proyecto. El método permite definir la inicialización del proyecto, roles de usuarios, y la asignación de permisos sobre distintos servicios en función de esos roles.
7. El servidor de jazz asume la existencia de un modelo de datos distribuido. Cada componente puede extender el repositorio de jazz agregando sus propios ítems y ofreciendo los mecanismos de acceso a los mismos. De esta forma el modelo de datos puede verse como la integración de una serie de modelos de datos y no como un modelo de datos monolítico predefinido.

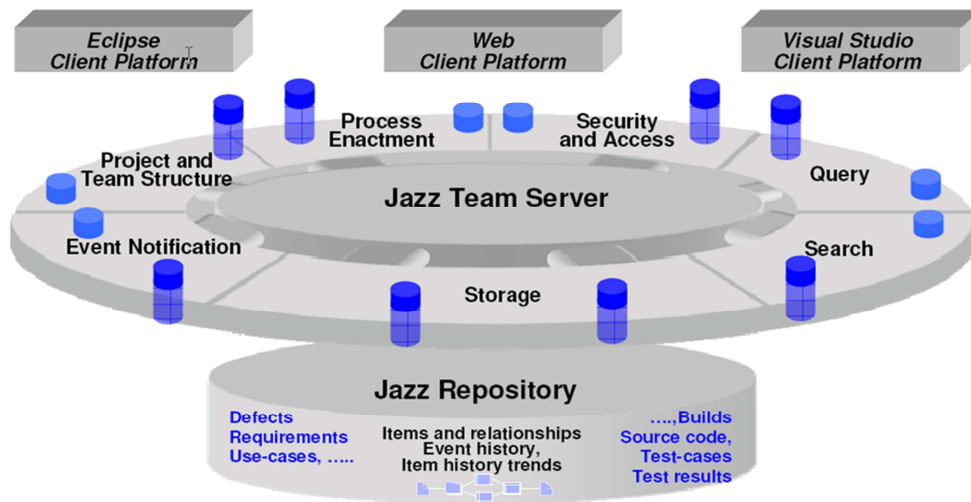


Figura 26 – Arquitectura de Integración de Jazz (JIA)

Si bien estas visiones originales en la definición de la plataforma jazz son muy potentes, resulta una realidad distinta la encontrada en la implementación ya que presenta grandes dificultades debido a las tecnologías subyacentes utilizadas. Por ejemplo, uno de los problemas es la dificultad para combinar soluciones. Las soluciones actuales generan un servidor de Jazz por cada herramienta lo que provoca duplicación de funcionalidad y dificultades en la integración y coordinación de los múltiples servidores. Estos problemas serán tratados con más detalles en las secciones restantes.

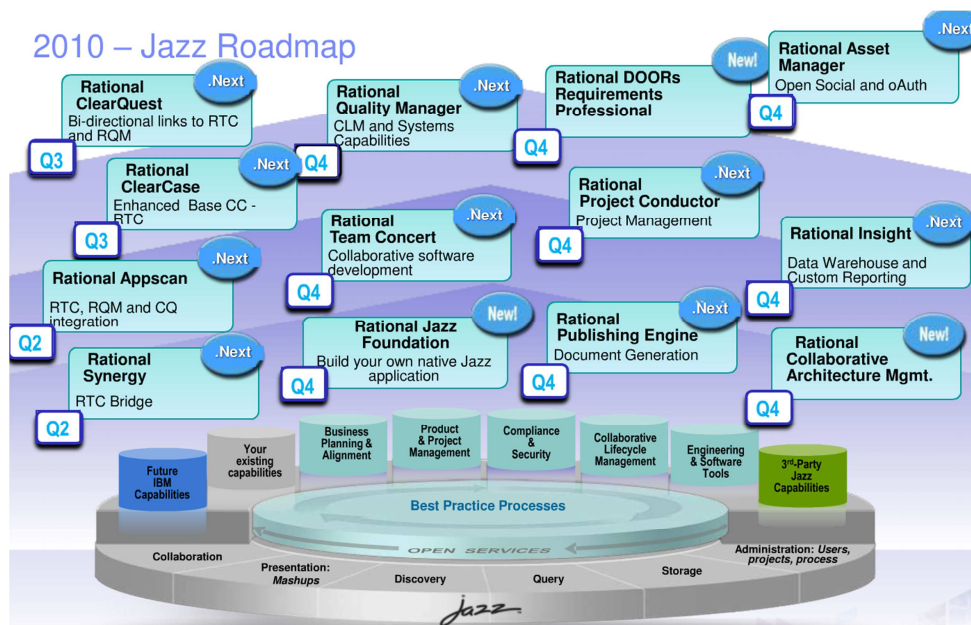


Figura 27 -- Ejemplo de Suite de Rational basada en Jazz

El ejemplo más claro de utilización de Jazz, es como plataforma de integración de toda la Suite de Rational para control del ciclo de desarrollo de aplicaciones. En la figura 27 se muestran en detalle las distintas aplicaciones actualmente disponibles y aquellas planeadas para los próximos meses. [w15]

### **3.3 Conclusiones del capítulo**

En este capítulo se presentó un estudio tanto de las herramientas existentes para soporte del área de CPM y como así también del estado del arte de las tecnologías de software basadas en eclipse para el desarrollo de aplicaciones.

En cuanto a las herramientas comerciales, si bien brindan muy poca información sobre su funcionamiento o arquitectura subyacente, en general se observan dos tendencias. O bien se ofrecen como complemento de complejas soluciones de BI y sin poder usarse de manera separada, o bien como herramientas muy simplificadas y con poca conectividad o sustento de arquitectura de software.

Las herramientas de las empresas más importantes (IBM, SAP, Oracle, etc) son en general complejas, difíciles de instalar y caras, sobre todo si se piensa su utilización en Pymes. Estas herramientas si bien permiten ser extendidas, poseen en general lenguajes propietarios o restricciones que dificultan la personalización o integración con otras tecnologías. Herramientas como IBM Cognos BI trabajan con niveles bajos de abstracción (tablas de Bases de Datos) que dificultan el tiempo y recursos necesarios para crear extensiones o integrarlas con sistemas basados en métodos ventajosos como los de desarrollo dirigido por modelos (MDD). Por otro lado se describieron ejemplos de herramientas más simples de instalar pero que carecen de una arquitectura (al menos no la describen) que provea principios básicos de extensibilidad, personalización, etc. Esto genera además problemas de sincronización al tener que conectar la información con otras fuentes de datos externas o al tener que realizar adaptaciones a las técnicas.

En base al análisis de la información encontrada, se pudo clasificar a las herramientas en tres grandes tipos lo que ayuda a clarificar sus fortalezas y debilidades. Esta clasificación además permitirá descubrir un GAP importante y posibilitara posicionar la arquitectura de herramientas ATID propuesta en el siguiente capítulo.

En la segunda parte de este capítulo se presentó la evolución de eclipse como plataforma para el desarrollo de aplicaciones, exponiendo el fundamento de su éxito

desde sus primeras versiones hasta los más recientes desarrollos. Eclipse surgió como una plataforma que permitía a partir de un núcleo central (RCP) crear aplicaciones de escritorio con una arquitectura modular y extensible. Este esquema fue altamente escalable y permitió la creación de complejas suite de herramientas integradas. Frameworks de modelación como EMF o frameworks de editores gráficos como GMF aprovecharon al máximo las capacidades de extensibilidad y permitieron la rápida generación de soluciones de calidad. Sin embargo en el último año la plataforma ha sido empujada por la necesidad de contar con una mayor flexibilidad para adaptarse a las aplicaciones que no corren necesariamente localmente en una maquina específica. Esta necesidad fue una de las motivaciones para el desarrollo de esta tesina y la mayoría de los aporte desarrollados en los capítulos siguientes han sido respaldados en los últimos meses por el surgimiento de una nueva serie de tecnologías que intentan extender eclipse en otras áreas como CDO, Dawn, o e4, también mencionados anteriormente en este capítulo.

Por último se presentó una breve introducción a la plataforma Jazz. Se describieron aquí solamente los principios más valiosos descriptos en la documentación de la visión de Jazz como plataforma de colaboración. Estos principios servirán como fundamento para el capítulo 4. Se dejan para el capítulo 5 de esta tesina los detalles de problemas encontrados en la implementación de Jazz. Ambas áreas, la de la definición y la de la implementación de Jazz, son dos mundos completamente distintos. Desde el punto de vista de los principios fundamentales de su arquitectura Jazz provee un enfoque muy potente. Sin embargo, lo que en ningún lado está documentado y que pocos llegan a enfrentarse es la realidad de su implementación actual. Estos problemas y posibles soluciones serán planteados en los capítulos siguientes.

Es de aclarar que los capítulos 2 y 3 fueron terminados de redactar en estos últimos meses y después de las experiencias de implementación o sea que muchas de las conclusiones que se mencionan hasta aquí es gracias a los desarrollos que se verán en los últimos capítulos. También es de destacar el cambiante mundo tecnológico en que nos encontramos hoy en día. Tendencias que surgieron durante el transcurso de la tesina como Google Wave, GEF3D y desarrollos en el Cloud, si bien fueron analizadas y testeadas, no fueron incluidas debido a que o bien fueron discontinuados o bien aún no está completamente clara su reales utilidades o necesidades.

## 4. Solución propuesta

En este capítulo se describe en detalle la solución de software planteada para el área previamente estudiada. Se ha visto en el capítulo 2 el incipiente interés por herramientas para una nueva área denominada CPM. Se han comentado en el capítulo 3 los tipos de herramientas existentes para el área y las ventajas de las tecnologías existentes para el desarrollo de aplicaciones basadas en la plataforma eclipse. En las primeras dos partes de este capítulo se comenzaran enumerando los requerimientos de un caso de uso típico para una metodología basada en CPM. Luego se describirán en detalle las ideas que sustentan la arquitectura. Por último se dará un ejemplo de utilización de la arquitectura a manera de validación de la misma.

### 4.1 Consideraciones Iniciales

Como se mencionó anteriormente una característica importante del área de CPM es la aplicación de la técnica en un marco conceptual bien definido. Ese framework define las actividades más importantes que deben ser llevadas a cabo en la aplicación de CPM. Ya vimos la descripción del proceso de 6 pasos denominado “*Execution Premium*” en la sección 2.3.4. Existen gran cantidad de métodos y variantes. Mencionaremos aquí brevemente, como ejemplo adicional, el caso del framework denominado “*Nine steps to success*” del Balanced Scorecard Institute (BSI) [W16]. Esto servirá para ilustrar un poco mejor la dinámica del área lo que servirá como base para la definición de los requerimientos. Es importante tener en cuenta que la arquitectura que se presentara en este capítulo no debe depender de un método específico, sino que intenta capturar los requerimientos genéricos comunes más importantes.

El framework presentado por BSI se ilustra en la figura 28 y se caracteriza por un ciclo continuo que se describe a continuación:

**1 Evaluación.** Uno de los primeros pasos es hacer una evaluación de la *Misión* y *Visión* de la organización, desafíos, facilitadores y valores. Herramientas que ayudan y soportan a estas actividades son los análisis de la competencia, análisis económicos, análisis del entorno general, análisis FODA, etc.

**2 Estrategia.** en este paso se define los elementos de la estrategia de la organización, incluyendo resultados estratégicos y perspectivas, temas estratégicos, poniendo atención en las necesidades de los clientes y las proposiciones de valor de



la organización. Los temas estratégicos permiten enfocar y agrupar un conjunto de objetivos. Generalmente son un número reducido de frases que tienen una vida más corta que la misión y la visión, por ejemplo la FIFA, recientemente anuncio su dirección estratégica con 3 temas: “Desarrollar el juego, Llegar al mundo, Construir un futuro mejor” [W17]

**3 Objetivos..** En este paso se crean los objetivos estratégicos, alineados con la estrategia definida en los pasos anteriores. Los objetivos se clasifican por tema y perspectiva, y se asocian mediante relaciones causa-efecto.



**Figura 28 – Ejemplo de Aplicación de CPM**

**4 Mapa estratégico.** En los pasos anteriores suele generarse una mapa estratégico por cada tema estratégico. Los objetivos y relaciones causa-efecto más generales que afectan a toda la empresa se consolidan en un mapa estratégico de alcance empresarial. Los mapas de los distintos temas se fusionan en un mapa empresarial resumido que muestra como la organización crea valor para los clientes e inversores.

**5 Medidas de rendimiento.** En el paso 5 se determinan las medidas de rendimiento (métricas) que se usara para evaluar si las iniciativas están contribuyendo o no a alcanzar un objetivo particular. Se deben determinar las metas y rangos. Para la elección se debe tener en cuenta varios aspectos como que la

métrica nos permita ver si el objetivo es alcanzado o no, que permita a los empleados saber que es lo más importante en lo que se tienen que enfocar, debe reflejar no solo el trabajo hecho sino los logros alcanzados, deben estar definidos explícitamente en términos de responsable, unidad de medición, frecuencia de actualización, etc, deben ser verificables para poder comprobar la precisión de los datos recopilados.

**6 Iniciativas.** En este paso, se desarrollan las iniciativas estratégicas que serán desarrolladas para soportar los objetivos. Para poder hacer un seguimiento de las mismas, estas se asignan al personal adecuado. Además deben servir como guías para el desarrollo de los distintos planes de acción y la asignación de recursos.

**7 Automatización.** En este paso se debe garantizar que se dispone de la infraestructura de software adecuada para que la información real de los distintos sistemas pueda conectarse con las métricas definidas en los tableros de control y para que pueda realizarse la comunicación y seguimiento efectivo de los objetivos, metas e iniciativas en toda la empresa.

**8 Cascada.** Este paso se refiere a la propagación de la información generada en los niveles más altos de gerencia hacia abajo en las distintas unidades de negocio de la empresa. De este modo la estrategia de los niveles superiores se refina en los niveles subsiguientes. Esto permite la generación de objetivos y métricas de bajo nivel, es decir, más cercanos a los detalles operativos de la empresa. Esta capacidad es clave para obtener un alineamiento de toda la empresa en el logro de los mismos objetivos que los planteados en la estrategia. Los mapas de los equipos e individuos son los que relacionan el trabajo diario con los objetivos del departamento. Métricas de rendimiento son generadas para todos los objetivos de todos los niveles. A medida que los mapas se propagan en cascada hacia abajo través de la organización, los objetivos así como las mediciones de rendimiento utilizadas se vuelven más operacionales y más tácticos. En todos los niveles los objetivos y mediciones tienen uno o más responsables asignados. El análisis de toda esta información permite encontrar problemas de falta de alineamiento. Además también sirve para controlar el comportamiento esperado de los empleados, pudiendo por ejemplo generar mecanismos de reconocimiento y recompensa para los que más contribuyen a los lineamientos.

Cada miembro de la empresa debe entender claramente cuál es la misión de la empresa y hacia donde esta quiere ir. Por lo tanto, emerge la necesidad de sistemas de software capaces de recuperar, resumir, e interpretar los datos para los usuarios

finales y relacionarlos con los objetivos y visiones de la empresa de manera de lograr un mejor seguimiento de los mismos. Todo esto contribuye a que además de los logros financieros, los empleados piensen de una manera diferente, se facilite la transición cultural de la empresa y se logre que la mayoría de los recursos humanos avancen en la misma dirección. Nos permite establecer las vías para que la estrategia y la planificación definidas por la dirección de la empresa se trasladen directamente a cada empleado, de manera que estos tengan visibilidad sobre sus objetivos personales específicos y sobre los objetivos globales de la empresa y de esta forma se sientan involucrados y partícipes de la estrategia. Sientan que entienden lo que la empresa está intentando conseguir y lo que espera de ellos, y tengan la certeza de si su trabajo avanza o no en la dirección adecuada. [W18]

**9 Validación.** En este paso se debe analizar los resultados e indicadores para responder la pregunta de si la estrategia está funcionando adecuadamente. Se debe validar que se estén midiendo las cosas correctas, que las medidas realmente reflejen el camino hacia los objetivos, etc. La organización se debe comprometer a monitorear los resultados del desempeño una vez que la estrategia ha sido desarrollada, planeada e implementada, permitiendo validar si esta ha sido correctamente ejecutada. Se debe:

*Monitorear y aprender.* Se requiere monitoreo y aprendizaje sobre los problemas, obstáculos, y retos a que se enfrenta la organización y que se ve reflejado en las desviaciones detectadas en los indicadores. Este proceso integra información acerca de las operaciones y la estrategia en una estructura cuidadosamente diseñada para las reuniones de revisión gerencial. En esta fase también es de ayuda la utilización de sistemas informáticos, los cuales deben tener una clara orientación hacia el monitoreo y seguimiento. La retroalimentación de la información de la BS hacia el nivel ejecutivo es muy importante ya que la información de monitoreo recibido en base a las mediciones permite efectuar los cambios fundamentales necesarios a la estrategia original. Los indicadores de los distintos objetivos permiten hacer los análisis de correlación que podrán llevarnos a entender los orígenes de los problemas a través de las relaciones causa-efecto. Por ejemplo, reducir dramáticamente el tiempo de procesamiento de las órdenes, reducir el tiempo de desarrollo de los productos y mejorar las capacidades de los empleados, son objetivos que se relacionan con los mismos a través de una serie de relaciones causa-efecto. Problemas en unos objetivos pueden tener sus causas en otros objetivos.

*Validar y Adaptar.* La organización debe también validar las hipótesis y los supuestos fundamentales para determinar si tiene la estrategia correcta. Esto involucra validar y adaptar la estrategia, utilizando datos operacionales internos y datos de los cambios del entorno y la competencia. Estas reuniones no son disparadas por los desfases en los indicadores sino que se refieren a un análisis más profundo sobre factores externos cambiantes. Los resultados de esta etapa pueden variar desde el ajuste de metas, la re-priorización de iniciativas, o la comunicación de nuevas expectativas a los distintos departamentos, o en casos de cambios externos, regulaciones gubernamentales, o cambios tecnológicos podría implicar el desarrollo un ciclo para una nueva estrategia.

Hasta aquí se mencionó la dinámica del área de CPM representada en una metodología específica. Es de destacarse, que en general las iniciativas relacionadas con el área de CPM están actualmente en su infancia y cambiando rápidamente. Por ejemplo, un seguimiento fino del método de 9 etapas planteado anteriormente podría encontrar ciertas superposiciones entre las etapas y también es común que en variaciones de estas metodologías se utilicen términos distintos aunque con una semántica parecida, definiciones poco claras o no consistentes entre los métodos, o uso de términos que suenan de manera similar pero cuyos significados son diferentes, lo que genera gran confusión [9]. Estas indefiniciones en el área también son un aspecto muy importante a tener en cuenta en la definición de la arquitectura ATID como se verá más adelante.

## **4.2 Requerimientos**

En esta sección se describen los principales requerimientos que se espera que puedan surgir y que deberían ser tenidos en cuenta a la hora de plantear una arquitectura de software para el área. Para estructurar esta sección se siguió un orden similar al desarrollado en la sección anterior, es decir, expresando los requerimientos en función de los nueve pasos del método presentado. Al final de la sección se resumen y clasifican los requerimientos más importantes en una tabla. Cabe aclarar nuevamente, que la arquitectura que se describirá más adelante no intenta ser una solución específica para la metodología presentada en la sección anterior, sino que se generan teniendo en cuenta también otras metodologías como las enunciadas en el capítulo 2.

En primer lugar en la sección anterior se vio que en general hay un método o proceso continuo que rige las actividades que se realizan en el área de CPM. Por ejemplo, el método de 9 pasos descrito anteriormente. Este proceso debería estar formalizado, documentado y accesible por todos los miembros de la organización.

Durante la etapa de *evaluación* se crea la visión y misión. Estos valores no pueden ser editados por cualquier usuario. En principio se requiere que exista el concepto de rol de usuario y permisos exclusivos para los distintos tipos de operaciones. Es decir, que parte integral de la plataforma debe ser la posibilidad de definir para cada par de acción y rol, los permisos de escritura, lectura, o ejecución correspondientes. Además surge la necesidad lograr armar distintos paquetes de funcionalidades en función del usuario, es decir, que de esta forma ciertos editores y aplicaciones solo estarán disponibles para los usuarios que tengan los roles adecuados. Es deseable que esto se logre teniendo que instalar solamente los editores con la funcionalidad usada.

Otro requerimiento que surge durante la etapa de evaluación, es el de proporcionar asistencia especializada para actividades suplementarias como acceso a editores de análisis de mercado, editores especializados de FODA, etc. En la mayoría de los casos también se destaca la posibilidad de obtener sugerencias o permitir hacer búsquedas en repositorios especializados, como por ejemplo, sería deseable durante la redacción de la visión o la visión, poder visualizar comparaciones respecto de las que han hecho otras compañías de la misma industria, o recibir ejemplos de la información que se ha considerado más exitosa, de manera de poder acelerar el desarrollo de las mismas o descubrir aspectos no considerados. Este acceso a información valiosa puede ser tanto a través de un repositorio especializado local como a través de una conexión remota con un repositorio más general.

En la etapa de *estrategia*, se definen los temas estratégicos más importantes para la empresa y las perspectivas relevantes. Nuevamente es un diferenciador clave contar con funcionalidad de asistencia, permitiendo sugerir casos típicos agrupados por industria, área, etc. También en esta etapa comienza a entrar en juego la necesidad de un editor gráfico para los mapas estratégicos o para expresar las perspectivas con otras notaciones como la mostrada en la figura 4.

En el paso de la definición de *objetivos* aparece tanto la necesidad de edición gráfica para dibujar y conectar objetivos, como mecanismos que faciliten la búsqueda reuso y publicación de objetivos. Además los controles de edición, locking y

colaboración también resultan de gran valor en esta etapa. Los objetivos se separan por temas, pudiéndose crear distintos mapas por cada uno de ellos. Estas tareas pueden ser asignadas a distintos usuarios y podrían ser útil, permitir la edición offline de los mapas permitiendo la publicación y sincronización diferida. También seguramente sería un beneficio adicional contar con templates para la creación de los mapas agrupados por ejemplo por industria de modo de acelerar el desarrollo de los mismos.

En la etapa anterior pueden actuar varias personas desarrollando cada una un tema estratégico particular con algún mapa estratégico asociado. En la siguiente etapa se deben consolidar los distintos mapas estratégicos en un mapa estratégico empresarial. Aquí es donde la utilización de un sistema con una arquitectura adecuada puede marcar cambios, ya que al soportar capacidades de colaboración y edición simultánea, los usuarios encargados de la construcción del mapa empresarial pueden tener conocimiento de los desarrollos del resto de los usuarios, por lo que la consolidación en un mapa estratégico empresarial resulta casi inmediata. Además el soporte para la creación de múltiples vistas y editores a partir de un mismo modelo también es un requerimiento clave manteniendo siempre la consistencia subyacente.

En la etapa 5 se deben determinar las *métricas* que serán los indicadores de rendimiento clave para cada objetivo. En este caso es indispensable contar con soporte que facilite encontrar, para una determinada industria y objetivo, las medidas de rendimiento claves típicas para esa industria. Las métricas pueden estar relacionadas entre sí mediante funciones matemáticas, el editor gráfico de métricas, debería permitir el soporte para este tipo de relaciones. De esta forma las hojas del árbol de métricas serán las que reciban los datos, que pueden ser tomados desde distintas fuentes, ya sea por ingreso manual, como por el sincronismo automático con valores específicos obtenidos de otros sistemas. Además es muy importante contar con indicadores gráficos adecuados para cada métrica pudiendo elegir entre varias forma de visualizar los indicadores. Esto permite armar tableros de control especializados para cada área. Las métricas pueden tener uno o más responsables. Se debe poder crear rangos y metas asociadas a las mismas y asignar notificaciones y alertas ante desfasajes.

Similarmente en la etapa de definición de *iniciativas*, se requiere de editores especializados en el tratamiento y análisis de las mismas. Estos editores deben poder acceder a información compartida como por ejemplo objetivos, facilitando la

búsqueda de duplicaciones o posibles conflictos o inconsistencias entre ellas y se debe garantizar poder asignar cada una de ellas a los usuarios responsables de llevarlas a cabo y hacer la comunicación y seguimiento..

En el siguiente paso de *automatización*, aparecen requerimientos relacionados tanto con la obtención y actualización de la información de las métricas, como las notificaciones de alertas o desfasajes. Mucho del éxito del alineamiento estratégico depende de contar con una arquitectura que permita poder hacer el siguiente paso de cascada adecuadamente. Surge la necesidad de representar información de departamentos, roles y unidades.

En la etapa de *cascada* se intenta alinear a toda la organización detrás de los mismos objetivos. Esto permite evitar que los departamentos soporten sus propios silos en lugar de encausarse en la misión definida para la organización. Se debe poder dividir un mapa estratégico permitiendo la asignación a las distintas subunidades de negocio. La necesidad de hacer la reconciliación y análisis de los distintos mapas generados por las distintas subunidades es otra de las tareas importantes. Durante este paso se generaran nuevos objetivos, nuevas iniciativas, métricas y metas y se debe garantizar que todas estén correctamente definidas y soportando los mismos objetivos. Puede ser que cada departamento requiera de sus propias herramientas extras de análisis, permitiendo la personalización de las herramientas, o requiera tener acceso a información de otros departamentos para un mejor alineamiento. En la consolidación de los elementos se deben tener en cuenta la detección de posibles contradicciones entre distintos departamentos.

En la etapa de *validación*, se requieren mecanismos que permitan monitorear y reportar desfasajes, metas no alcanzadas, iniciativas no completadas en tiempo, etc. Además se debe poder generar análisis para poder entender porque no se están alcanzando las metas planeadas, como por ejemplo, estudios de dependencias entre objetivos, o dependencias entre métricas. Reportes más generales también son útiles. En el caso de las revisiones de validación, puede ser necesario el uso de otras herramientas de análisis especializadas, como mapas de capacidades, que permitan establecer correlaciones entre los datos tanto externos como internos. Estas herramientas deben poder integrarse y estar disponibles para los usuarios adecuados. Además pueden ser muy útiles para estas etapas las capacidades de realizar simulaciones de cambios y estudios de impacto.

Otro factor a tener en cuenta, es el relacionado con la inmadurez, la gran cantidad de variaciones de terminologías y las diferencias semánticas que pueden surgir en el área de CPM según la técnica particular empleada. En una arquitectura pensada para soportar múltiples metodologías este aspecto es importante y debe considerarse de antemano. En general, la mejor solución para evitar estos conflictos es la definición de estándares. Por ejemplo, la OMG (consorcio creador de estándares como UML o BPMN) define el llamado Business Motivation Model (BMM) que intenta definir un lenguaje común para desarrollar y comunicar planes de negocios de una manera organizada. En el caso de CPM se nota la falta de marcos conceptuales más formales o estándares. Si bien la formalización de CPM sería un requerimiento importante, se deja como trabajo futuro debido a su complejidad inherente. Otros tipos de estándares son los que permiten la interoperabilidad entre las distintas herramientas. Por ejemplo, uno de los intentos por lograr esta interoperabilidad, fue la definición del estándar XMI, para intercambio de información. Veremos que para arquitecturas más complejas es importante la definición de otros tipos de estándares más relacionados con las tecnologías involucradas. Se comentara en la siguiente sección la propuesta del estándar OSLC como forma de lograr interoperabilidad en entre las distintas herramientas. De esta forma será posible tener en cuenta el requerimiento que permite la interacción con otras aplicaciones más complejas como Enterprise Resource Planning (ERP) o Customer Relationship Management (CRM)

Comúnmente en la construcción de este tipo de herramientas se utiliza un esquema de meta-modelo único, es decir, un esquema de datos monolítico sobre el cual se desarrolla toda la arquitectura. Esta característica termina generalmente siendo un problema para la extensibilidad, dado que no es posible el fácil agregado de nuevos conceptos sin tener que modificar los esquemas. Un requerimiento que pocas herramientas tiene en cuenta es la posibilidad de soportar el agregado dinámico de múltiples lenguajes específicos de dominio (DSL), es decir, que permita la definición de un DSL pero sin acotar la extensión, ampliación o interacción con otros DSL, cosa muy común en el área. Además de estas facilidades de extensibilidad, la arquitectura debería soportar la fácil personalización del comportamiento, por ejemplo, para poder reemplazar funcionalidades generales por implementaciones específicas y propietarias de las empresas.

Otro requerimiento importante es el de la capacidad de realizar validaciones y chequeo de ciertas restricciones. Por ejemplo, según muestreos entre empresas que



tuvieron éxito en la implementación del CMI se señala que los indicadores de gestión deberían tener idealmente esta distribución:

1. Financieros 20%
2. Clientes 24%
3. Procesos internos 38%
4. Aprendizaje e Innovación 18%

Como se ve, para que los indicadores sean balanceados se deberían manejar indicadores que en un 80% no deben ser financieros, a diferencia de empresas que dicen usar CMI pero que tienen un 75% de indicadores financieros. La validación del cumplimiento de reglas de negocio tan simples como estas o mucho más complejas debería ser soportada por la herramienta con extensiones a la arquitectura que no comprometan ni limiten las posibilidades de personalización de las reglas aplicadas

Se debe tener en cuenta que la herramienta final podría tener distintos esquemas de aplicación. Por ejemplo, una forma de instanciación de la arquitectura sería como herramienta utilizada por consultores a los que les es requerida hacer una auditoría sobre una empresa. Otro esquema implicaría la implementación de la arquitectura en toda una empresa o podría solo requerirse la instalación solo en ciertos niveles o unidades de negocio. Si bien lo ideal es desarrollo completo de la arquitectura de manera de lograr un alineamiento estratégico de toda la empresa, también deberían tenerse en cuenta estos requerimientos alternativos.

Una última tendencia que cabe la pena destacar es el creciente interés de poder acceder a los sistemas informáticos desde otros dispositivos como celulares y tablets,

En base a los requerimientos enumerados anteriormente, se detallara en la siguiente sección una arquitectura dinámica denominada ATID que permite contemplar estos y otras necesidades del área. Se intentara dar el mayor grado de detalle sobre las tecnologías empleadas en cada parte de la arquitectura. En el capítulo 5, se describirán las experiencias encontradas al implementar los componentes más importantes de la ATID. Dada la complejidad que alcanzo la arquitectura, no se describirán todos los componentes. Aquellos que queden fuera del alcance de la tesina por razones de tiempo, son enumerados en la sección de trabajos futuros.

#### Requerimientos Centrales

- Soporte para extensibilidad mediante nuevos editores dependiendo de la metodología empleada.
- Plataforma integrada basada en estándares
- Armado de distintos clientes de software según el usuario.
- Editores con capacidades gráficas de edición (mapas estratégicos, árbol de métricas)
- Soporte para múltiples usuarios con distintos permisos de accesos y distintas funcionalidades disponibles para cada usuario.
- Creación de múltiples editores y vistas sobre el mismo modelo por múltiples usuarios al mismo tiempo.
- Soporte de múltiples metamodelos.
- Conexión con repositorios para hacer búsquedas, comparaciones y recibir sugerencias.

#### Requerimientos de Valor Agregado

- Repositorios de búsqueda externo (visión, misión, temas, perspectivas, objetivos, iniciativas)
- Búsquedas de duplicaciones o conflictos entre iniciativas (similar objetivos métricas etc.)
- El método debería estar formalizado, documentado y accesible por todos los miembros de la organización
- Personalización de las funcionalidades para adaptarlas a las técnicas propias de cada empresa
- Soporte de roles y modelación de la estructura de la empresa
- Locking de objetos en los editores
- Colaboración para que múltiples usuarios pueda editar el mismo diagrama
- Mecanismos de mezclado de modelos.
- Asistencia en la Comunicación (asignación de responsabilidades)
- Facilitar la propagación y la mezcla del efecto cascada
- Notificaciones y alertas

#### Requerimientos de Análisis y Reportes

- Simulaciones y análisis de Impacto de cambios
- Soporte de validaciones (reglas de negocio)
- Soporte para armado de tableros de control
- Generación de informes y reportes

#### Interoperabilidad y otros

- Interoperabilidad con otras herramientas y estándares
- Conectividad con sistemas externos como BI, ERP para obtener valores de métricas
- Acceso offline para edición de información con mecanismos de sincronización entre la información local y la compartida
- Arquitectura empresarial (escalabilidad, robustez)
- Soporte de múltiples configuraciones
- Soporte de tablets, celulares

Figura 29 – Categorización de los requerimientos más importantes

### 4.3 Arquitectura de Integración Dinámica (ATID)

Como se describió en el primer capítulo y en el comienzo de esta sección, existe una incipiente área denominada CPM con características y requerimientos específicos del ámbito y para la cual resultan ventajosos contar con herramientas que asistan a los usuarios en las distintas etapas de aplicación de la metodología. En el capítulo 3 se presentaron las principales herramientas existentes que se clasificaron en tres grandes tipos de acuerdo a características en común. Por un lado, en las de tipo I se clasificaban a las soluciones por lo general complejas y pesadas de las grandes empresas que estaban fuertemente atadas a las soluciones de BI. Por otro lado en las de tipo II se clasificaron a las soluciones más pequeñas que solo ofrecían soporte para BSC pero sin una arquitectura subyacente de integración claramente definida. Las de tipo III eran extremadamente simples.

Una de las aseveraciones de esta tesina es que ninguno de los tipos de herramientas presentados resulta adecuado para la gran mayoría de las empresas, en especial para las pequeñas y medianas empresas (pymes). Por un lado, es importante reconocer que el concepto de Balanced Scorecard no es tan complejo (típicamente se obtienen no más de 20 indicadores por mapa) por lo cual muchas organizaciones continúan usando simplemente Planillas de Calculo y Procesadores de Texto o papel. Sin embargo el proceso de distribución y recopilación de feedback si puede requerir mas esfuerzo, sobre todo a medida que aumenta el tamaño de las empresas. Por lo tanto si bien es muy ventajoso contar con una solución de software para el área, no resulta indispensable tener que instalar una compleja solución de BI para lograr aplicar los beneficios del área de CPM. Por otro lado es importante contar con una arquitectura flexible que permita, de ser necesario, poder hacerla crecer fácilmente según las necesidades y recursos de la empresa, alcanzando en el caso limite una integración completa como las de las grandes soluciones de BI.

En esta sección se presenta la *Arquitectura Tecnológica Integrada Dinámica* (ATID) que se propone en esta tesina como base para la construcción exitosa de cualquier herramienta para el área anteriormente descrita. La misma hace hincapié en la aplicación de la tecnología, con énfasis en la integración pero con el dinamismo necesario para poder armar distintas configuraciones según las necesidades como se mostrara más adelante. Es decir, que la ATID puede también verse también como la instanciación de una meta-arquitectura que permite armar múltiples variaciones de acuerdo a las necesidades de los usuarios.

La arquitectura así presentada abarca todo el espectro de tipos descrito en la figura 20, es decir, ofreciendo una solución desacoplada de BI, simple en su instalación y uso, pero con una arquitectura robusta y extensible que permita dinámicamente agregar los componentes requeridos por la empresa, de modo de poder en el caso extremo, armar una solución tan compleja como las de tipo I.

Se comenzara describiendo la arquitectura aquí propuesta con un diagrama de alto nivel (Figura 30). En este diagrama se representan los componentes mínimos esenciales para el funcionamiento de la arquitectura y ejemplos de los componentes más importantes que extienden la misma. En las siguientes subsecciones se dará una descripción de cada uno de los componentes, mencionando las tecnologías más adecuadas para la implementación de los mismos. En el capítulo 5 se darán los detalles técnicos de las experiencias de implementación para cada componente.

Como se muestra en la figura 30 la primer gran division es la de la arquitectura cliente-servidor en donde se muestra la posibilidad de que los usuarios interactuen con un servidor centralizado a traves de distintas tecnologias como telefonos moviles, tablets, laptops, o navegadores web embebidos en distintos dispositivos.

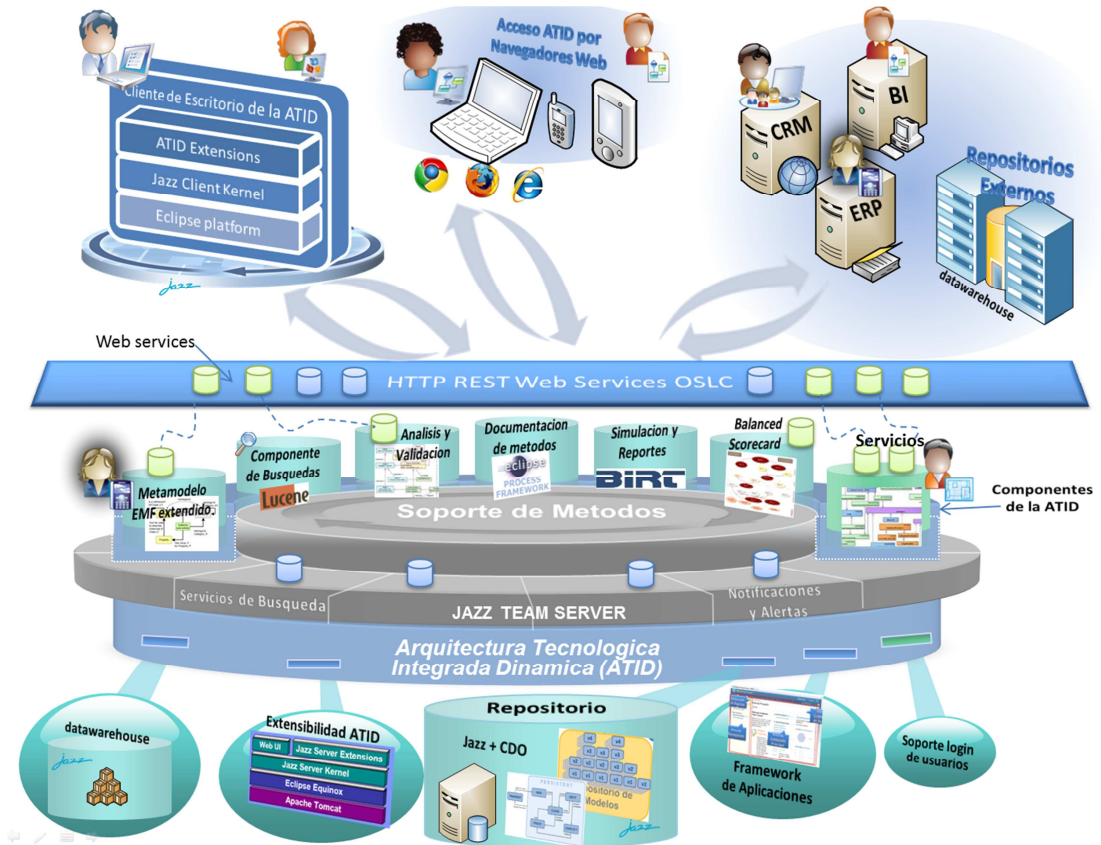


Figura 30 – Diagrama de alto nivel de la arquitectura ATID

A continuación se describirán los componentes principales de la arquitectura.

#### 4.3.1 Componentes Centrales

Se comenzara describiendo aquí los conceptos centrales que si o si son requeridos pues ellos forman la base para poder armar dinámicamente configuraciones más complejas. Más adelante, en la sección 4.3.5 se describirán las distintas variaciones en la instanciación de la arquitectura, es decir, que no todo lo mostrado en la figura 30 es requerido para el funcionamiento mínimo del sistema.

##### Mecanismo de extensibilidad (OSGi).

Vimos en el capítulo 3 que eclipse posee una arquitectura robusta para el desarrollo de aplicaciones de escritorio. Esto se dio principalmente debido a su mecanismo de extensibilidad basado en plugins, el cual se hizo compatible con el estándar OSGi.

Los primeros intentos por construir una plataforma extensible usando J2EE no fueron exitosos, el aislamiento de los cargadores de clases (classloaders) para cada uno de los archivos “.war” iba en contra de la extensibilidad deseada. Una solución era empaquetar todos los componentes en un gran archivo war. La plataforma Jazz definió un esquema basado en el proyecto Equinox de eclipse denominado OSGi Bridge Servlet [W19]. Esto permitió mantener el exitoso esquema de extensibilidad usado en eclipse también del lado del servidor, con lo cual además es posible reutilizar los plugins desarrollados para las aplicaciones de escritorio en el servidor. Los mismos mecanismos de extensibilidad y puntos de extensión aplican para la plataforma Jazz. En la arquitectura ATID se mantiene esta ventaja arquitectónica tanto del lado del cliente como del lado del servidor. Esto permite desarrollar componentes (plugins) que pueden ser reutilizados en distintos contextos, permitiendo aplicaciones la fácil creación de aplicaciones que corran en distintos dispositivos como laptops o navegadores web. Se verán más detalles de este mecanismo en la sección de implementación.

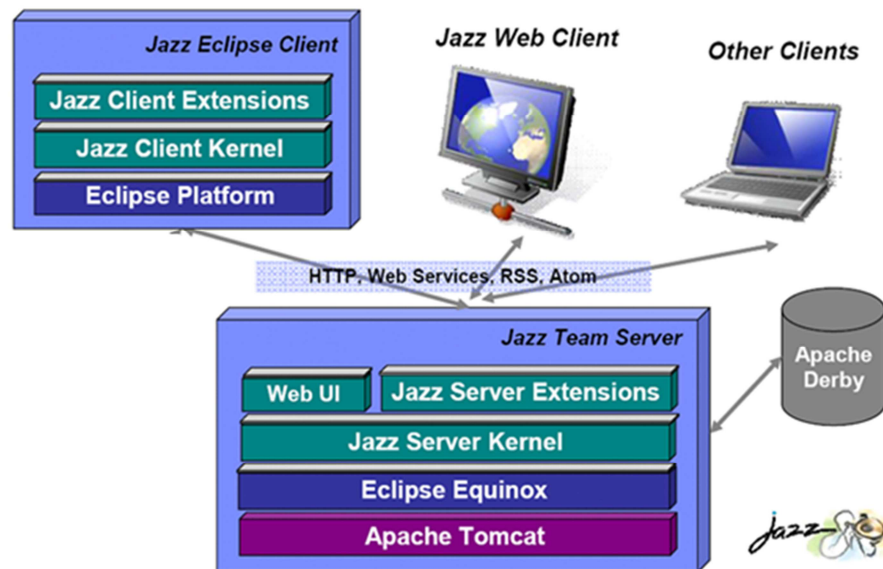


Figura 31 – Mecanismos de extensibilidad de la ATID basado en Jazz

### Comunicación cliente–servidor.

En este caso también se toma como referencia la implementación desarrollada para la plataforma Jazz debido a sus ventajas de abstracción de los protocolos web, y la utilización de web services para el transporte de recursos e información. Esto es, existen componentes tanto del lado del cliente como del lado de los servidores que

facilitan la intercomunicación. Permiten la abstracción de los protocolos y capas intermedias [W20].

Estos mecanismos de comunicación son mantenidos en la ATID como protocolos de comunicación entre los distintos componentes de manera de lograr una arquitectura alineada con las tendencias de implementaciones orientadas a servicios (SOA). Así como en las aplicaciones RCP la comunicación era a través de puntos de extensión, en el servidor los componentes del lado del cliente se comunican con los componentes en el servidor exponiendo servicios. Con cada componente además es posible registrar un modelo EMF, de modo que los servicios web automáticamente transforman los objetos EMF en representaciones transportables mediante los protocolos web. De esta forma, el programador se abstrae de los detalles de transporte recibiendo objetos EMF de alto nivel.

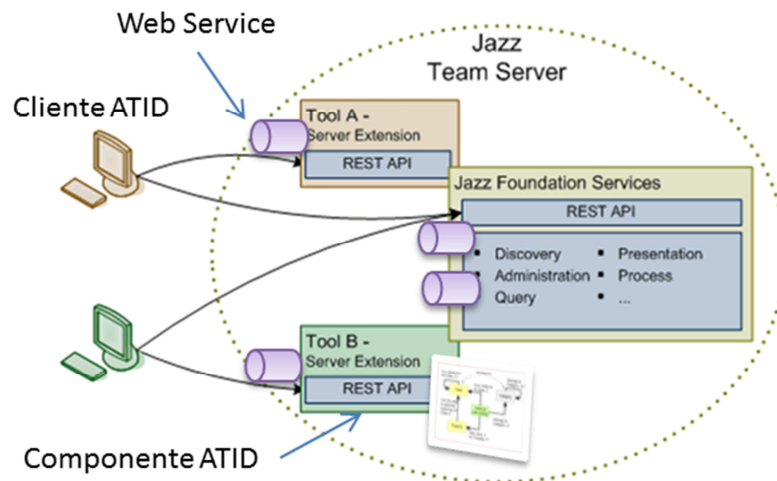


Figura 32 – Comunicación entre los clientes y los servidores

### Soporte para usuarios

Otra componente típica de todo sistema es el soporte para validación de usuarios. Esta componente es muy importante pues para la mayoría de las acciones realizadas en el sistema debería quedar registrado el usuario responsable de las mismas. En este sentido, las ventajas de la plataforma Jazz es que provee servicios de manipulación de usuarios bastante genéricos, que puede ser utilizado con la instanciación de la plataforma.

La arquitectura ATID mantiene estos mecanismos de login de usuarios tomando como referencia los provistos por jazz. De esta forma se proveen los mecanismos de manejo de los datos usuarios, autenticación de los mismos y permisos. Se permite

tanto la creación de administradores de servidores, como usuarios con acceso a determinados proyectos. Este módulo de manejo de usuarios se combinara luego con el de módulo de procesos y roles para permitir armar tablas de control de acceso al sistema más complejas, que determinara que operaciones puede realizar cada usuario según el contexto en que se encuentre y el rol asignado.

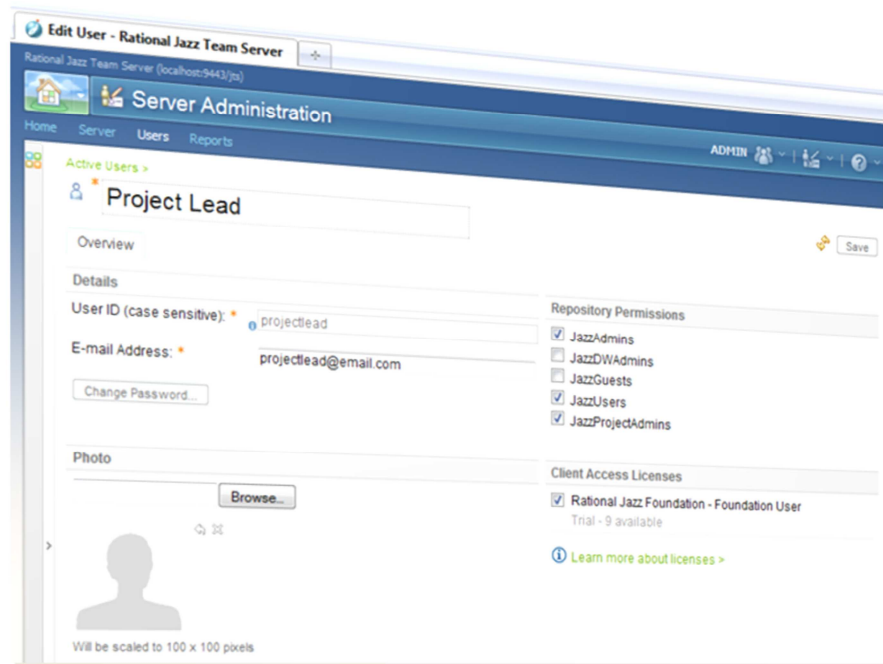


Figura 33 – Interface de administracion de usuarios de Jazz reusada por la ATID

### Framework de Aplicaciones

Una característica importante de la arquitectura es la posibilidad de contar con frameworks para rápido desarrollo de las aplicaciones. En el caso de los clientes de eclipse basados en RCP, se disponen de componentes como swt, jface, y eclipse.ui que facilitan el desarrollo y extensibilidad del paradigma de vistas y múltiples editores de interfaz. De manera similar es importante contar con un paradigma extendido para la arquitectura ATID. Jazz provee un framework para las aplicaciones web denominado JazzUI. Este provee puntos de extensión que permiten extender la interfaz del navegador web con nuevas páginas de información. Esto facilita acortar los tiempos de desarrollo ya que no debemos preocuparnos por tareas rutinarias como la codificación de los mecanismos de login, creación de proyectos, etc, pudiendo reusar los mismos y enfocándonos en el desarrollo de extensiones.

La ATID reusa tanto el framework RCP para las aplicaciones de escritorio como el framework de JazzUI para las aplicaciones con acceso desde el navegador web.

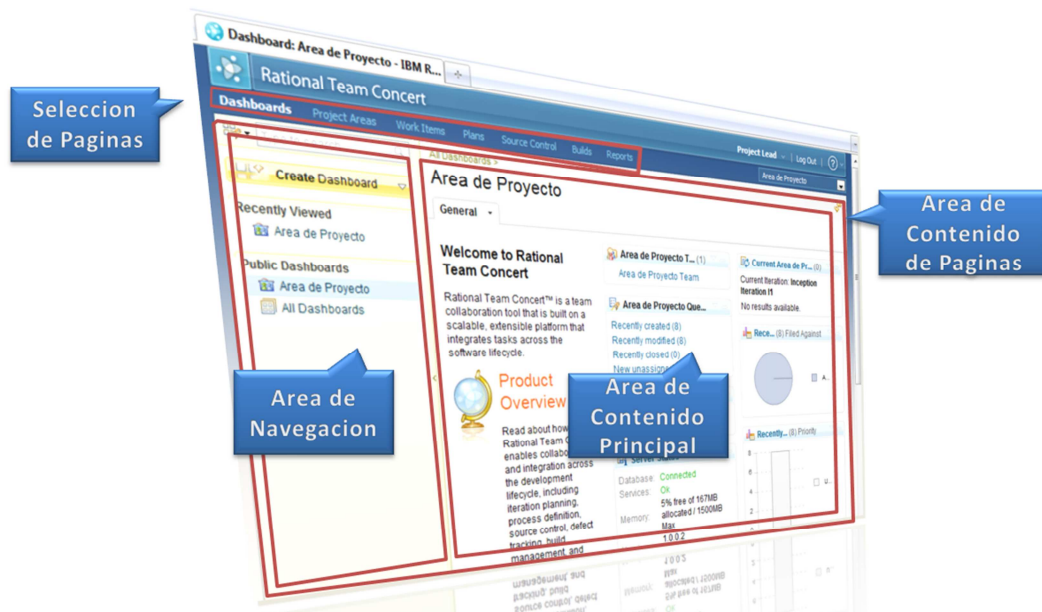


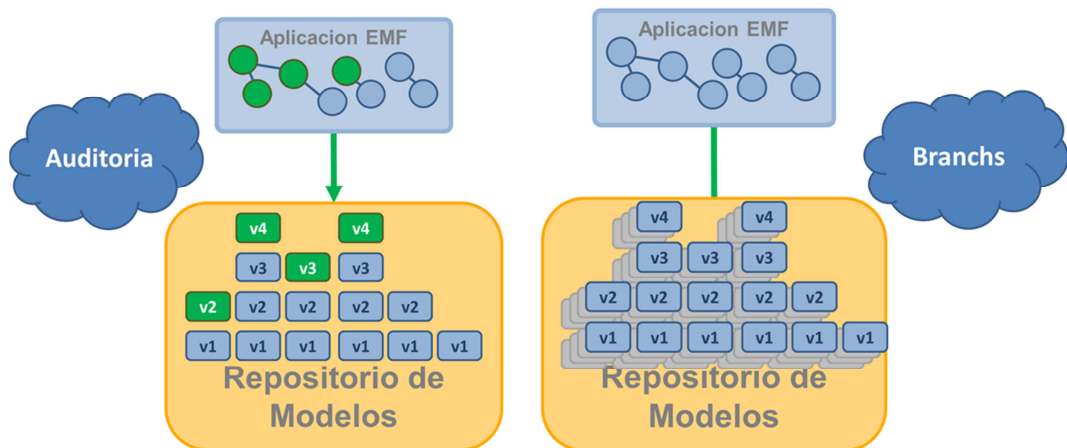
Figura 34 – Framework de aplicaciones web de Jazz

### Componente de Repositorio.

Esta es una de las componentes centrales de la arquitectura. Una de las ventajas de la plataforma eclipse para las aplicaciones RCP de escritorio, era la posibilidad de reutilizar un conjunto de plugins que ofrecían la posibilidad de manipular en alto nivel un área de trabajo (workspace) donde era posible organizar la información en múltiples recursos, pudiendo extender la plataforma con editores especializados para operar sobre cada recurso. Sobre este principio de recursos, fue construido EMF, el cual permitía trabajar con objetos de alto nivel abstrayendo la manera en que los recursos eran serializados para su persistencia, de manera de poder optar entre almacenarlos en el repositorio de archivos de eclipse o en una base de datos. Este esquema de archivos y editores fue exitoso para aplicaciones de escritorio y sistemas monousuarios. Con la necesidad de soportar múltiples usuarios y almacenamiento remoto, aparece la necesidad de generar repositorios de información más complejos. Este es un punto crítico de la arquitectura y para el cual se concluyó en esta tesina que no existe una implementación adecuada que pueda ser reusada. En principio se estudió la utilización del repositorio de Jazz. El mismo ofrecía la ventaja fundamental de la posibilidad que ofrecía para la separación de modelos y capacidades de indexación y auditoría entre otras. Esto es, cada componente podía contribuir al repositorio definiendo su propio meta-modelo. De esta forma, se lograría que cada extensión de la herramienta sea fácil de acoplar y desacoplar de la arquitectura, contribuyendo a la necesidad de dinamismo de la ATID. Toda la implementación de



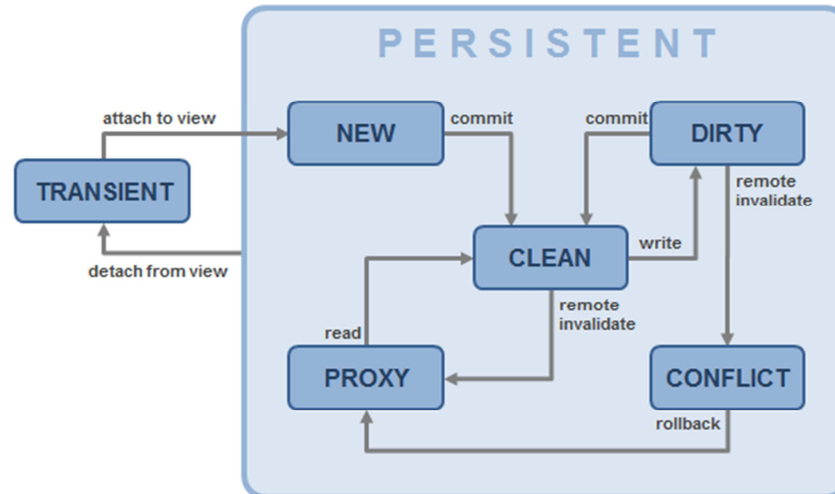
los componentes de jazz, como por ejemplo los de gestión de usuario, están implementados usando los conceptos del repositorio de jazz. La ventaja de esto es que permite a los nuevos modelos hacer referencia a cualquier concepto de la plataforma de manera de poder definir entidades nuevas y relacionarlos con otros como los de área de proyecto o usuario. Sin embargo, como se detallará en la sección de implementación, las restricciones no documentadas de la implementación de jazz, como la imposibilidad de definir jerarquías de herencia complejas o relaciones entre modelos externos y las falta de soporte para notificaciones compatible con EMF, además de la necesidad de regeneración de la base de datos con cada cambio con la consecuente pérdida de información, hace muy dificultosa cualquier implementación y más compleja aun la integración con otras tecnologías como GMF. Esta fue la restricción más importante encontrada para la implementación de un prototipo de la arquitectura basado en la plataforma Jazz. En los últimos meses, surgió una mejor opción para la implementación dada por la tecnología CDO.



**Figura 35 – Capacidades de Auditoría y creación de Branchs en CDO**

Como se describió en la sección 3.2.4 CDO es una generalización de los conceptos de EMF para soportar el almacenamiento distribuido de información. La gran ventaja es que la implementación es transparente para los desarrolladores y compatible con todas las características de EMF. El servidor de CDO soporta el almacenamiento de múltiples versiones de los mismos objetos lo que es útil para hacer auditorías y además permite crear branches para que los usuarios puedan operar con sus propias copias de la información durante el tiempo que consideren necesario, conservando la *traceability* con las entidades al instante de originar el branch de modo de poder hacer una mezcla futura de la información modificada (Figura 35).

Otro concepto importante que aporta CDO es el mecanismo de estados y notificación automáticos. Por ejemplo, en la figura se muestra el diagrama de estados asociado a cada objeto EMF. Los cambios de estado son notificados automáticamente por el servidor de CDO al resto de las conexiones.



**Figura 36 – Diagrama de estados de un objeto CDO**

Existen muchos otros conceptos importantes en CDO como soporte para transacciones, consultas, colaboración. La única desventaja de CDO es la falta del mecanismo de extensibilidad de lado del servidor y la falta de mecanismos de comunicación orientados a servicios como los mencionados anteriormente en esta subsección. Por ello, para la ATID se propone la necesidad de un repositorio con características comunes a ambas implementaciones, es decir, las ventajas mencionadas del repositorio de Jazz pero con una implementación con las facilidades de CDO. Uno de los principales resultados esperados de la implementación del repositorio de la ATID sería la posibilidad de la rápida prototipación de editores gráficos basados en GMF sobre modelos distribuidos con soporte para múltiples usuarios, lo cual es un requerimiento muy importante para el tipo de aplicaciones empresariales para los que está pensada la arquitectura.

#### **4.3.2 Componentes de Valor Agregado**

Los componentes descritos en la sección anterior son la base para lograr las implementaciones de la ATID. En esta sección se describen componentes que si bien no son esenciales, son muy importantes para agregar funcionalidades de valor agregado y cuya inclusión o no dependen de los recursos disponibles o necesidades puntuales de cada organización.

### Componente de Búsquedas.

Este es otro de los componentes que podría ser trabajo de una futura tesina ya que requiere una sub-arquitectura por sí mismo. Se vio durante los requerimientos que durante las distintas etapas de CPM sería muy útil contar con mecanismos de búsquedas y sugerencia de objetivos, métricas, relaciones típicas entre las entidades, y otro tipo de información. Este componente intenta mostrar la necesidad por la capacidad diferenciadora de contar con mecanismos de asistencia para búsquedas y sugerencia. Por ejemplo, contar con el tipo de sugerencias que ciertos buscadores realizan a medida que escribimos una consulta, o la capacidad de corregir errores de tipeado (Figura 37), hasta complejos mecanismos de indexación de cualquier dato generado por las aplicaciones.



Figura 37 – Ejemplo de funcionalidades de asistencia en búsquedas

Lo que sugiere este componente, es que exista un mecanismo integrado en la arquitectura que determine cuál es la mejor forma de indexar la información para facilitar no solo las búsquedas, sino también las sugerencias. Lo idea es que estos mecanismos sean lo más transparentes para los desarrolladores. Por ejemplo, se investigó y testeó para esta tesina los mecanismos ofrecidos por Jazz, el cual provee servicios de indexación y búsquedas. Sin embargo las grandes desventajas de esa implementación son dos:

1. La indexación está disponible solo para los modelos EMF guardados en el repositorio de Jazz, para lo cual se requiere anotar los modelos con metadatos que proveen información de indexación de bajo nivel.
2. El lenguaje de expresión de consultas es de API propietaria y sin una sintaxis y semántica clara.

La arquitectura de la ATID destaca la incorporación de un componente que permita realizar desde complejas consultas hasta simple búsquedas. Se busca que el programador se abstraiga de los detalles de estos componentes salvo por

optimizaciones de rendimiento muy particulares. En estas áreas la ATID aconseja investigar el uso de librerías como las del proyecto Lucene o EMF Index.

Es importante hacer notar que existen varios tipos de información que debería estar accesible para búsquedas. Por un lado la información propia de la herramienta en cuestión, por ejemplo, objetivos o métricas actualmente almacenadas en el sistema. Por otro lado debería poder conectarse y buscar información en repositorios externos de información preparados para almacenar y consolidar la información de múltiples proyectos de manera de estar disponibles para los usuarios. Por ejemplo, dado un objetivo, podría querer buscar en un repositorio externo cuales son los objetivos que frecuentemente se relacionan con él, e importar esa información. También debería poder buscarse información en el datawarehouse histórico, que se detalla más adelante, si es que se utiliza ese componente. Siempre es importante conservar la referencia al origen o fuente desde donde se obtuvo la información de manera de poder mantener sincronización de ser necesario.

#### **Documentación de Metodologías y Procesos.**

Vimos anteriormente que debe existir una forma de poder acceder y referenciar la formalización del método de CPM utilizado, de manera que esté disponible para cualquier usuario dentro de la organización. Esta componente puede implementarse con tecnologías con *eclipse Process Framework* (EPF) o con *Rational Method Composer* (RMC). Estas herramientas generan sitios web navegables que pueden ser instalados en el servidor. La desventaja es que no son fácilmente editables o personalizables, aunque no es común que la metodología cambie muy frecuentemente. Las actividades asociadas con un método particular pueden ser almacenadas en el sistema para su seguimiento y su control. RMC provee la capacidad de exportar las actividades del método como templates para Rational Team Concert (RTC) de manera de poder utilizar personalizaciones de los workitems de RTC para hacer el seguimiento de la metodología.

#### **Componente de Gestión de Metodologías.**

Otra parte importante de la arquitectura es el soporte para la metodología subyacente al proyecto utilizado. En el caso de la plataforma Jazz, se exige que cada área de proyecto, se asocie con un método en particular. En jazz, el método controla varios aspectos de un proyecto. Por ejemplo, determina los roles de los usuarios que participan en ese tipo de proyecto. Además las acciones que se realizan cuando se inicializa el proyecto. Controla los permisos para todas las operaciones sobre el

proyecto así como las precondiciones y pos-condiciones que se ejecutan luego de las operaciones.

En la ATID se tiene en cuenta los beneficios de poder contar con una formalización de la metodología subyacente utilizada por la herramienta. Sin embargo se considera que la implementación de Jazz está demasiado orientada a las metodologías de desarrollo de software y no es completamente reutilizable en el contexto como CPM. Esta componente debería lograr una mejor separación de los conceptos. Por ejemplo, la definición de los roles en el caso de CPM podría hacerse de acuerdo a la estructura organizativa de la empresa. Configuraciones propias de la arquitectura, como las aplicaciones que un rol en particular puede ejecutar también deberían ser soportadas por la metodología, como no es el caso en Jazz.

Otra característica deseable sería la recolección de información de otros componentes, como los que definen el metamodelo, los comandos, los editores gráficos de ui, etc, para permitir establecer claramente los permisos de cada una de las contribuciones y su relación con los roles. En la sección de implementación se mostrara cómo es posible utilizar los mecanismos de jazz para desarrollar el método exponiendo estos detalles anteriormente descriptos.

### **Componentes para Colaboración.**

Una tendencia que es muy común encontrar en los sistemas modernos es la posibilidad de que distintos usuarios editen el mismo modelo al mismo tiempo. Para ellos es importante contar con mecanismos que permitan controlar el acceso compartido de recursos. Durante el desarrollo de la tesina se estudiaron los mecanismos de colaboración provistos por Jazz los cuales solo permiten enviar emails ante ciertos eventos o colaborar a través de un chat embebido, pero no dispone de mecanismos de notificación en tiempo real. Ya vimos que CDO provee mecanismos para controlar el estado de los objetos compartidos y provee además mecanismos de notificación automáticos para mantener el sincronismo global. Uno de los requerimientos encontrados tiene que ver con la necesidad de desarrollar la estrategia en *cascada* en toda la organización con el mayor nivel de comunicación posible. Para lograr esto se intentó crear en la arquitectura mecanismos de colaboración basados en Jazz. Las experiencias de implementación se detallan más adelante pero se observó la complejidad y dificultad para implementar dichos mecanismos. Con el surgimiento hace un par de meses de CDO en su versión 3 se abre la posibilidad de lograr una rápida y fácil implementación de los mecanismos de

colaboración, notificación y sincronización antes descriptos, aunque se requiere bastante trabajo para integrar los conceptos de CDO con la arquitectura basada en Jazz como se describió en la sección del repositorio. La ATID propone los componentes de CDO3 para soporte de colaboración. Esto permitiría que múltiples usuarios trabajen sobre un modelo compartido al mismo tiempo, reflejando los cambios al modelo lo antes posible para evitar inconsistencias o conflictos. Se deja como tema de futuro estudio el de los problemas asociados con esta dinámica de la colaboración aunque se deduce que contando con este tipo de componentes se podría lograr un alineamiento empresarial con una rapidez sin precedentes. La ATID reconoce la importancia de que los mecanismos de comunicación, colaboración y sincronización deben estar disponibles y pensados desde un principio. Sin una arquitectura subyacente preparada es muy difícil agregar esta capacidad más adelante.

### Componente de alertas y notificaciones



Figura 38 – Tablero de control que permite el seguimiento de las métricas.

En la sección 2 y en la sección 4.1 vimos que una de los objetivos de la metodología CPM es el de descubrir los indicadores claves de rendimiento (KPI) que permitan evaluar el alcance o no de los objetivos empresariales. Para estas métricas se fijan metas y rangos. Típicamente, con esta información se arman los tableros de control como los de la figura 38 que permiten monitorear si el valor de los indicadores está dentro de los rangos esperados.

Dado la gran cantidad de tableros y para evitar las posibilidades de errores de interpretación humanos, es conveniente contar con mecanismos que permitan enviar

notificaciones a las personas adecuadas ante la ocurrencia de ciertos eventos, como el desfasaje de los indicadores. La ATID propone la incorporación de un componente especial para soportar estos casos. Este módulo debe soportar la definición de diversos tipos de notificaciones, como ser el envío de emails, alertas en tiempo real u otro tipo de mecanismos. Los tipos de eventos pueden ser tan simples como caer por debajo de un valor base, hasta eventos más complejos como que para cierto rango de fechas el valor deba estar comprendido entre dos valores específicos. El tipo de alerta también debería poder configurarse en función del nivel de gravedad. También este módulo puede relacionarse con el módulo de workitems de RTC, de manera de definir acciones de seguimiento de las alertas.

### **4.3.3 Componentes de Análisis y Reportes**

Una vez ingresada la información, surgirán una gran cantidad de análisis útiles y reportes que se podría realizar sobre la información almacenada en el sistema. Es importante que la arquitectura esté preparada para lidiar con estos requerimientos. Se mencionan aquí algunos componentes importantes relacionados con estas capacidades.

#### **Componente de Datawarehouse**

Como vimos en la sección 2 este componente es muy común en las arquitecturas de BI. Jazz también incorpora un almacén de datos de solo lectura cuyo principal objetivo es servir de fuente de datos eficiente para la generación de reportes. Cada cierta cantidad de tiempo, la información del sistema se consolida en el datawarehouse. A diferencia de Jazz, en la ATID esta componente no es requerida pudiendo optarse por las implementaciones más adecuadas. Dependiendo de la cantidad de usuarios, la cantidad de información histórica contemplada, el tipo de reportes, etc deberá decidirse si incluir o no esta componente en la arquitectura.

Otro interesante uso de esta componente es en la definición de repositorios de información externos. Vimos que el reuso del conocimiento es una parte importante para acelerar el desarrollo de la metodología de CPM. Poder hacer búsquedas de información típicas para una industria particular, o consultar una base de datos con las métricas típicas de rendimiento para un objetivo dado, son funcionalidades diferenciadoras para el área de CPM. Un forma de lograr esto es compartiendo

información no confidencial con repositorios externos de información implementados como datawarehouse de propósitos generales.

### Componentes de Análisis y Validación

Otra funcionalidad muy importante que generalmente no se tiene en cuenta pero que es aplicable a esta y otras áreas es la posibilidad de realizar análisis con la información que se vean reflejados en los distintos editores con capacidades de coloración o “heatmapping”. En esta técnica el componente provee capacidades para anotar con distintas representaciones graficas o simplemente usando una escala de colores el comportamiento de algún indicador representativo. Por ejemplo en la figura 39, se muestra un heatmap simple.

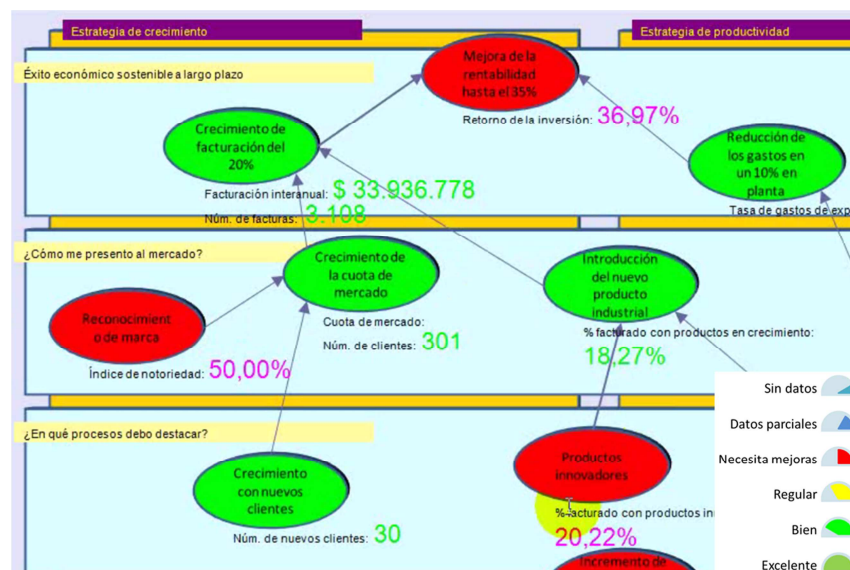


Figura 39 – Ejemplo de utilización de la componente de análisis

También puede ser importante poder definir reglas de validación de cierta información, ya sea por la expresión de reglas de negocio (Business rules), precondiciones, pos-condiciones, etc. Herramientas como ILOG proveen librerías basadas en eclipse para tal fin.

### Componente de Simulación:

Una vez que se cuenta con este tipo de software es muy útil poder hacer simulaciones de las variables de entrada para estudiar tanto posibilidades de cambios, como análisis de respuesta a distintos eventos. El poder responder a preguntas del tipo “que pasaría si” puede ser muy importante para validar distintos escenarios relacionados con la estrategia. Por ello también debería ser parte integral de la arquitectura la capacidad de tomar la información existente y usarla para simulaciones sin afectar los datos reales de la empresa. Capacidades como el branch



de CDO explicado anteriormente pueden ayudar a la implementación de este componente.

#### **Componente de reportes.**

Por último es importante destacar que la ATID nos ofrece un marco integrado que abarca el ciclo completo de gestión del rendimiento de negocios de principio a fin, desde la definición del modelo de negocio hasta la revisión de las previsiones, pasando por la planificación, los presupuestos, la definición de métricas e indicadores y por supuesto, el análisis de los datos y la elaboración de informes.

La generación de reporte tiene varios niveles y una subarquitectura propia. Por un lado existen reportes a manera de tableros de control, para reflejar los indicadores determinados durante la aplicación de la técnica de BSC. Por otro lado existe una necesidad de las organizaciones de poder contar con sistemas de soporte para la generación de documentación. Esta componente de la ATID debe contemplar todos estos casos. Una posible implementación de esta componente con tecnologías basadas en eclipse es a través del proyecto BIRT (Business Intelligence and Reporting Tools). La exitosa inclusión de BIRT en gran cantidad de herramientas en el mercado actual demuestra su valor para la implementación de la componente. Inclusive Jazz mismo utiliza BIRT para la generación de reportes. Debido a la complejidad de esta componente no se implementó y se dejó como trabajo futuro.

#### **4.3.4 Componentes de Interoperabilidad y Otros Conceptos**

Por último se mencionan en esta sección otros aspectos importantes relativos a la arquitectura ATID como aspectos de interoperabilidad e integración, entre otros.

##### **Interoperabilidad.**

Este es uno de los puntos más críticos de la arquitectura y que la mayoría de los vendedores actuales de soluciones usan en su beneficio. En general, las grandes soluciones para el área existentes presentadas en el capítulo 3, utilizan mecanismos o lenguajes propietarios que solo se integran bien con las soluciones de la misma marca. Por ejemplo, si quisiese usar el sistema de BSC de SAP pero usando BI de Oracle sería muy difícil de lograr. La ATID ve esto como un problema para las organizaciones y plantea un esquema flexible, abierto y estándar para lograr la interoperabilidad. En este contexto, al utilizar una arquitectura basada en SOA, la ATID permite que cualquier solución pueda acceder a sus servicios. De la misma

forma, si otros sistemas como ERP (Enterprise Resource Planning) o BI exponen su funcionalidad a través de servicios, será posible que la arquitectura ATID interactúe con ellos fácilmente. De esta manera se evita la proliferación de silos de datos dispersos.

Para simplificar aún más la conectividad entre las distintas componentes, sería deseable la estandarización de los servicios y recursos expuestos, de manera que puedan ser fácilmente intercambiados entre las distintas herramientas. Un paso en esta dirección lo da OSLC (Open Services) que es una comunidad abierta de individuos y compañías en la que se elaboran y publican estos estándares.

#### **Otros Conceptos.**

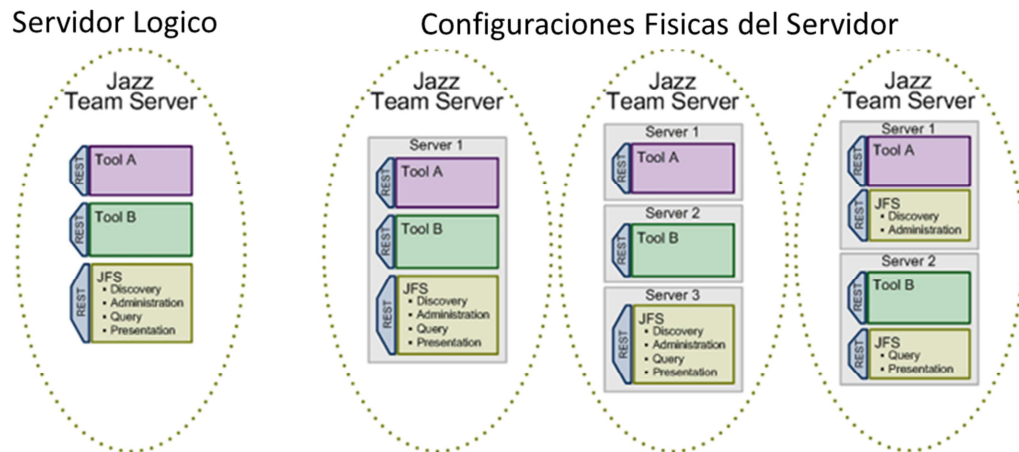
Además de todos estos componentes descritos, existen otros conceptos que deberían tenerse presente a la hora de instanciar la arquitectura ATID. Por ejemplo, muchas implementaciones fracasan porque el tiempo necesario para cargar la información mínima para poder usar las funcionalidades es tan grande que los usuarios se sienten frustrados. Un diferenciador importante es contar con conectividad con algún repositorio de templates predefinidos que permitan estructurar e inicializar el espacio de trabajo de acuerdo a una metodología específica o un caso de industria específico. La posibilidad de ofrecer soluciones pre-empaquetadas o asistentes de software que permitan rápidamente guiar al usuario en el uso de las componentes de la arquitectura puede ser muy importante.

Otros conceptos que deberían tenerse en cuenta son los mecanismos de seguridad, el soporte para la internalización, componentes de organigramas o estructura de la empresa, componentes específicos de propagación en cascada, etc. Gracias al concepto central de dinamismo de la arquitectura muchos de estos mecanismos pueden ser agregados a la plataforma en trabajos futuros.

#### **4.3.5 Configuraciones de la arquitectura**

La arquitectura ATID puede verse como una arquitectura de arquitecturas, dado que su estructura no intenta ser rígida, sino que con el agregado de distintos componentes se van generando distintas arquitecturas intermedias, o instanciaciones de la misma. Para lograr escalabilidad, la ATID sugiere una arquitectura de n-capas, multi-server y multi-thread para poder ser escalada de modo de soportar a miles de usuarios si fuese necesario. También la ATID debe tener en cuenta la disponibilidad, para lo cual se soporta el balance de carga dinámico y se debe tener en cuenta la rápida recuperación

ante fallas de hardware u otras anomalías. Es decir, que si bien desde el punto de vista externo la arquitectura puede verse como compuesta de un único servidor lógico, físicamente puede estar implementado con múltiples componentes del servidor distribuidos de acuerdo a funcionalidades y necesidades específicas (Figura 40). Esto también permite controlar la capacidad, es decir, que tanto hardware se requiere en función de la sobrecarga de procesamiento esperada.



**Figura 40 – Escalabilidad y configuraciones del servidor basadas en Jazz**

### **Soporte para trabajar desconectado (offline)**

Estas facilidades para armar distintas configuraciones, también permitirían resolver un problema común de estas arquitecturas que es lo que sucede cuando no tenemos posibilidad de conectarnos al servidor remoto. Por ejemplo, Jazz exige conectividad para poder operar con el área de trabajo remota. Si se pierde la conectividad se pierde la posibilidad de trabajar aun en los clientes de escritorio. Una ventaja de la ATID proviene de la utilización de mecanismos basados en CDO 3, ya que el mismo provee la capacidad para clonar el repositorio y sincronizar los clones de manera transparente al programador. Esto es, de ser necesario un usuario puede crear un replica local del repositorio o parte de él, de modo que esta imagen se mantenga sincronizada automáticamente con el repositorio maestro. Como se muestra en la figura 41, si se pierde la conectividad, es posible seguir trabajando y guardando la información en el repositorio local. Al recuperar la conectividad esa nueva información es enviada al repositorio maestro.

El servidor local es el que guarda temporalmente los cambios y se comunica con el servidor remoto así como también escucha por los pedidos de notificación de

cambios del servidor remoto. De esta forma la sincronización se lograría mediante la comunicación transparente entre los servidores.

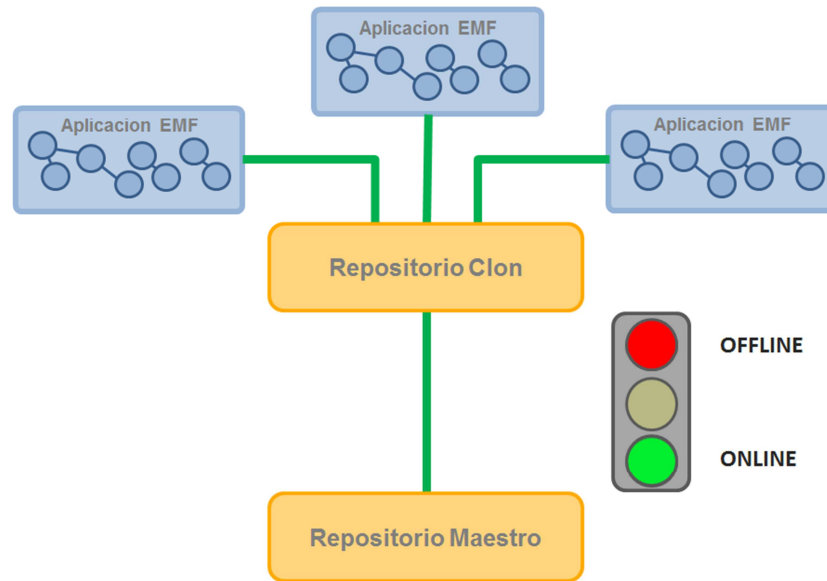


Figura 41 – Soporte para trabajar offline con CDO

#### 4.4 Validación de la Arquitectura

Se describió en las secciones anteriores un ejemplo de la dinámica de la metodología CPM, se describieron los requerimientos típicos de esa dinámica y se propuso una arquitectura de software denominada ATID para el área. En esta sección se explicara a manera de validación, como la ATID funciona para resolver los problemas claves típicos del área. Para ello, se utilizara como contexto el de una organización mediana con un cierto número de departamentos o subunidades de negocio.

Una de las primeras etapas es la de definición de la misión, visión y otros análisis como el FODA. Esto es llevado a cabo por el CEO y el equipo de nivel de gerencia ejecutivo de la organización. En este caso se proveen de módulos que proporcionan editores gráficos que permiten crear y almacenar esta información. Utilizando el módulo de usuarios y metodología es posible configurar los permisos para este tipo de tareas en esos roles. Esto combinado con la extensibilidad de la arquitectura permite que la funcionalidad de edición este solo visible para los usuarios adecuados. Una vez que esta información se encuentra lista para compartir con el resto de la organización se propaga en cascada a las distintas unidades de negocio utilizando los módulos de propagación y configuración adecuada del repositorio. De esta forma es posible recibir retroalimentación o cualquier comentario acerca de la información enviada.

Otro de los pasos iniciales es la definición del mapa estratégico, para eso vimos que en la etapa de estrategia se requiere un mecanismo de búsquedas y sugerencias para las perspectivas, esto es aportado por el componente de búsquedas y repositorio externo. También es necesario poder abrir editores tanto individuales para las perspectivas como un editor de mapas estratégicos, todos integrados. En este caso un desarrollador debería contribuir al módulo de metamodelo (EMF) y al módulo de editores (GMF) para generar la funcionalidad adecuada. Es decir que se generaran plugins que enriquecen el repositorio con las entidades de metamodelado necesarias para soportar los conceptos como perspectivas, objetivos, métricas y relaciones. Por otro lado se generaran los editores gráficos necesarios para que los usuarios puedan editar la información. En general cada etapa o funcionalidad requiere el almacenamiento de información. Por lo tanto la extensibilidad del repositorio de la ATID es clave y debe permitir el agregado de ítems y sus relaciones con objetos existentes.

Otro componente importante que entra en juego tanto durante la creación del mapa estratégico como en el resto de los editores es el módulo de colaboración, además de los ya mencionados como búsquedas y repositorio, permitiendo que el grupo de usuarios adecuados sea consciente de la presencia o tareas realizadas por otros integrantes del mismo grupo. Por ejemplo, varios usuarios al mismo tiempo podrían estar agregando objetivos al mismo mapa estratégico.

Una vez definidos los objetivos estratégicos, uno de los pasos siguientes, es la creación de métricas de rendimiento clave. En este caso los módulos de búsqueda y sugerencia son nuevamente utilizados. También aquí será posible generar editores gráficos específicos que permitan por ejemplo relacionar las métricas. Hoy en día, los editores gráficos se generarían mediante el acceso de clientes de escritorio, pero gracias al reuso de plugins de la ATID, sería posible también generar visualizaciones de esta información en navegadores web. También es importante, que en etapas posteriores cada usuario o unidad de negocio pueda crear sus propios tableros de control utilizando componentes gráficos como los presentados en la sección 4.3.3. Es aquí donde el componente de reportes entra en juego.

En la etapa de definición de iniciativas, el módulo de búsquedas o sugerencias podría llegar a ser de utilidad, aunque en este caso es más importante que las iniciativas surjan desde las distintas unidades, subiendo y bajando en la cadena de responsabilidades de manera dinámica, asignando además el control de las mismas a

los distintos usuarios. En este caso también podrían ser importantes los módulos de simulaciones, ya que sería importante poder analizar si los resultados esperados de las iniciativas realmente contribuyen o no a alcanzar la metas.

En el paso de automatización entran en juego varios subsistemas. Por un lado los mecanismos de propagación de la información a las distintas unidades, la posibilidad de definir notificaciones y alertas ante eventos como desfasajes de valores esperados para métricas, etc. Por otro lado entra en juego la conectividad con información de otros subsistemas dado que mucho de los valores reales de las métricas provienen de otros subsistemas de la organización.

En la etapa de cascada es en donde se prueba realmente las capacidades de la arquitectura. Aquí es donde los distintos subsistemas deben funcionar coordinadamente como una orquesta. Se requiere ligar la estrategia de la compañía con las estrategias de las distintas unidades de negocio, a la vez que se intenta alinear y motivar al personal para optimizar la estrategia de ejecución. En cada unidad de negocio se podrá tomar un mapa estratégico del nivel superior y refinar los objetivos creando nuevos mapas que luego serán compartidos con el nivel superior. Nuevamente aquí intervienen los módulos de propagación de la información, los de notificación a las distintas subunidades, los de colaboración para garantizar la consistencia y permitir la supervisión, entre otros. De esta forma se logra resultado esperado de alinear a todos los empleados por medio de procesos de comunicación formales y ligando los objetivos personales e incentivos con los objetivos estratégicos. Esto permite evitar las barreras personales (metas personales, incentivos no ligados a la estrategia), las barreras de visión (estrategia no entendida por la implementación) y las propias barrera de la administración (control operativo sin estrategia).

En la etapa de evaluación, es donde intervienen los componentes de alertas automáticas. Las metas no alcanzadas o los indicadores fuera de rango son síntomas de problemas, los cuales son importantes de corregir a la brevedad. Aquí también intervienen los componentes de análisis, notificaciones, alertas, y los de navegabilidad.

Durante todas estas etapas también es importante mencionar, que la disponibilidad del módulo de validación y reglas de negocio, permitiría ajustar el comportamiento a las necesidades particulares de la empresa, con reglas específicas o

personalizaciones del método. Las reglas pueden ser tan simples como no permitir más de 10 objetivos por mapa, hasta reglas muchos más complejas.

En este caso se evaluó la instanciación completa de la arquitectura, es posible otros escenarios como por ejemplo, el de las empresas de consultoría que solo requieren desarrollar la estrategia y entregarla a manera de reportes y paquetes que luego puedan ser importados en los sistemas propietarios de la empresa. Para ese caso sería posible crear configuraciones de la arquitectura más específicas y con menos componentes para los consultores. Esta es la gran ventaja del dinamismo de la ATID.

Por último, se plantearan aquí las consultas enumeradas inicialmente en la propuesta de la tesina y se mostrara como la ATID, responde a todas ellas.

**Consultas de Alineamientos de Objetivos:** Uno de los primeros pasos en el planeamiento estratégico es determinar cuáles serán los propósitos del programa tanto a corto como mediano plazo. Que objetivos estratégicos de la organización serán estudiados. Con que misión o visión organizacional se relacionan. Se debe producir una hipótesis que detalle las iniciativas que eventualmente mejorara el rendimiento y los resultados (Es decir, desarrollar un mapa estratégico). *Precisamente toda la descripción inicial de esta sección permite la validación clara de este punto.*

**Consultas de línea base:** Definir cómo lograr monitorear o recolectar información desde las más importantes fuentes, que datos necesitamos recolectar, cuales almacenar y cuales descartar. *En este caso la ATID presentada permite claramente determinar las métricas de rendimiento clave y provee mecanismos basados en estándares como SOA para conectarlas con distintas fuentes de información.*

**Consultas de costo y riesgo:** Las consecuencias financieras de una nueva iniciativa debido a BI deberían ser estimadas. Es necesario evaluar el costo actual de las operaciones y el incremento en costos asociados con las iniciadas por BPM. Se debe poder determinar cuál es el riesgo que una iniciativa falle. La evaluación de riesgos debería ser convertida en una métrica financiera e incluirla en el planeamiento. *Para este caso la ATID cuenta con el módulo de simulación, y una arquitectura de repositorio basada en CDO 3 que lo soporta como se detalló en la sección 4.3.*

**Consultas sobre los clientes e inversores.** Determinar quién se beneficiara con las iniciativas y quién pagará. Quién se beneficiara indirectamente. Cuáles son los beneficios cuantitativos y cualitativos. Es la iniciativa especificada la mejor para incrementar la satisfacción para todos los tipos de clientes o existe una mejor forma? Como se monitorea si los clientes están siendo beneficiados? *La ATID ataca esta problemática de dos formas, por un lado, el editor de iniciativas debe contar con un metamodelo y funcionalidades que contemplen la búsqueda, el descubrimiento y el almacenamiento de esta información, por otro lado, la arquitectura permite la conexión con fuentes externas a la organización que podrían aportar por ejemplo resultados de encuestas a los usuarios, entre otras mediciones.*

**Consultas relacionadas con métricas:** Los requerimientos de información deben ser transformados operacionalmente en métricas definidas claramente. Se debe decidir que métricas usar para cada pieza de información que está siendo recolectada. Son estas las mejores métricas? Como sabemos eso? Cuantas métricas necesitan ser controladas? Si es un numero grande de métricas (como usualmente lo es), que tipo de sistema se utilizara para controlarlas? Se utilizan métricas estandarizadas, de modo de poder compararlas contra aquellas de otras organizaciones? Cuáles son las métricas industriales disponibles? *Para facilitar encontrar las métricas adecuadas, la ATID provee un módulo de búsqueda y sugerencias basadas en repositorios externos que contienen precisamente información histórica de las que por experiencias previas han resultado las mejores métricas para una industria o segmento en particular. En ciertas situaciones, el módulo de proceso o metodología, disparara durante la etapa de evaluación, los análisis necesarios para saber si las métricas elegidas realmente contribuyen a los objetivos planteados. Además, el módulo de monitoreo, enviara alertas y notificaciones ante el descubrimiento de cualquier desfasaje.*

**Consultas orientadas a los resultados:** El programa CPM debería ser monitoreado para asegurar que los objetivos están siendo alcanzados. Es común que sean necesarias correcciones. Se debe intentar demostrar que las iniciativas, son las que efectivamente están contribuyendo a los cambios en los resultados y que estos no se producen aleatoriamente. *Como se mencionó en el punto anterior, la ATID contempla y permite resolver estos problemas.*



## 4.5 Conclusiones del capítulo

Se presentó en este capítulo, la ATID, una arquitectura de arquitecturas que propone resolver ampliamente las necesidades de los usuarios del área de CPM.

En la primer parte de este capítulo se ejemplificó las principales actividades realizadas durante la aplicación de la metodología CPM. Para ello se tomó como ejemplo, la metodología de nueve pasos “The Nine Steps to Success”. La misma tiene una estructura de ciclo continuo que abarcar toda las etapas de desarrollo de CPM desde los primeros análisis de evaluación, las actividades de definición de la estrategia desarrollada por temas específicos, el desarrollo de objetivos, métricas metas, la generación y consolidación de mapas estratégicos, la elaboración de iniciativas, la definición de los sistemas de automatización requeridos para la ejecución, y las etapas de monitoreo y validación general de la estrategia. Con esta descripción del método más los conceptos detallados en el capítulo 2, e intentando capturar las necesidades fundamentales de cualquier variación en las metodologías, se enumeraron en la sección 4.2 los principales requerimientos que debería satisfacer una arquitectura de software para el área.

A continuación se planteó que uno de los resultados de las experiencias de implementación desarrolladas durante esta tesina fue el descubrimiento de los problemas que impiden implementar herramientas que respondan a todos los requerimientos planteados. Se encontró que las tecnologías actuales no son suficientes y por ello se presentó la arquitectura ATID como solución integrada dinámica basada en las tecnologías de eclipse y que permitiría ofrecer tanto a las pequeñas y medianas como a las grandes empresas, de una arquitectura que pueda adaptarse y crecer al ritmo de las necesidades de la empresa. Se detallaron los componentes de la arquitectura mencionando las tecnologías testeadas para cada uno de ellos expresando además las ventajas, desventajas y posibles implementaciones alternativas.

Se mencionó además el carácter de meta arquitectura de la ATID por el hecho de permitir el instanciado de múltiples configuraciones dado por su dinamismo, permitiendo esto a su vez resolver problemáticas comunes como es la posibilidad de trabajar transparentemente de si existe o no conectividad con la web.

Por último se mostró a manera de validación de la arquitectura, como las distintas componentes se utilizan en las distintas situaciones. También fue posible

responder a todas las preguntas planteadas inicialmente durante la propuesta de la tesina, relacionadas con las capacidades esperadas de la herramienta de software

Como resultado se logró presentar una arquitectura que permite un tratamiento integrado de la gestión del rendimiento, aportando a las organizaciones un valor fundamental. En el siguiente capítulo se detallarán los desarrollos y las experiencias de implementación.

## **5. Resultados de Implementación**

En esta sección se detallan las experiencias de implementación de la arquitectura ATID presentada en el capítulo anterior. Se intentó mantener una estructura similar a la del capítulo 4, es decir, que para cada componente se explicaran las tecnologías testeadas, las extensiones hechas y los resultados o propuestas.

Para probar los conceptos fundamentales de la arquitectura se pensó en implementar los casos de uso más desafiantes. Uno de ellos, y que impactaba en los componentes principales, eran el de utilizar los mecanismos de extensibilidad de Jazz para poder conectar múltiples editores GMF con el repositorio de Jazz. Esto requería la creación de editores GMF, instanciación de meta-modelos, instanciación de servicios de jazz y extensión del repositorio de jazz, es decir, que involucraba a los componentes centrales. Otros de los casos de uso mencionados se relacionan con la modelación de las metodologías de CPM, estudio de los mecanismos de búsquedas, extensibilidad web, edición múltiple en editores, que se detallan a continuación.

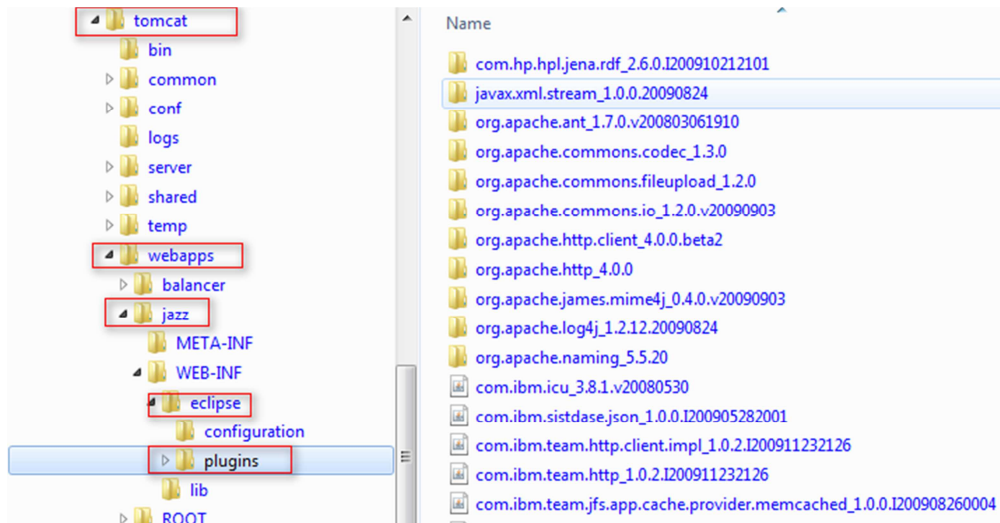
### **5.1 Componentes centrales**

#### **5.1.1 Mecanismo de extensibilidad de Jazz**

Como se explicó en la sección 4.1, el mecanismo central de extensibilidad que hizo tan popular a eclipse, es decir, su mecanismo de plug-ins, se conserva en la implementación de jazz. De esta forma, Jazz permite definir plug-ins que se instalan tanto en las aplicaciones de escritorio como en el servidor. Por ejemplo, el servidor de jazz puede correrse con Tomcat. En este caso, en la estructura de directorios de Tomcat, en la carpeta webapps veremos que la aplicación jazz, contiene en su carpeta web-inf un directorio llamado eclipse que contiene el mismo tipo de plugins genéricos de eclipse. (Figura 42). El proceso por el cual se transforman los plugins desarrollados para el servidor, en la estructura requerida para correr en tomcat se denomina “aprovisionamiento” y es aún un procedimiento relativamente complejo ya que requiere la creación de una “features” (grupos de plugins), un update-site, se debe detener el servidor de tomcat, crear una serie de folders y archivos de configuración .ini y rearrancar el servidor.[W23]

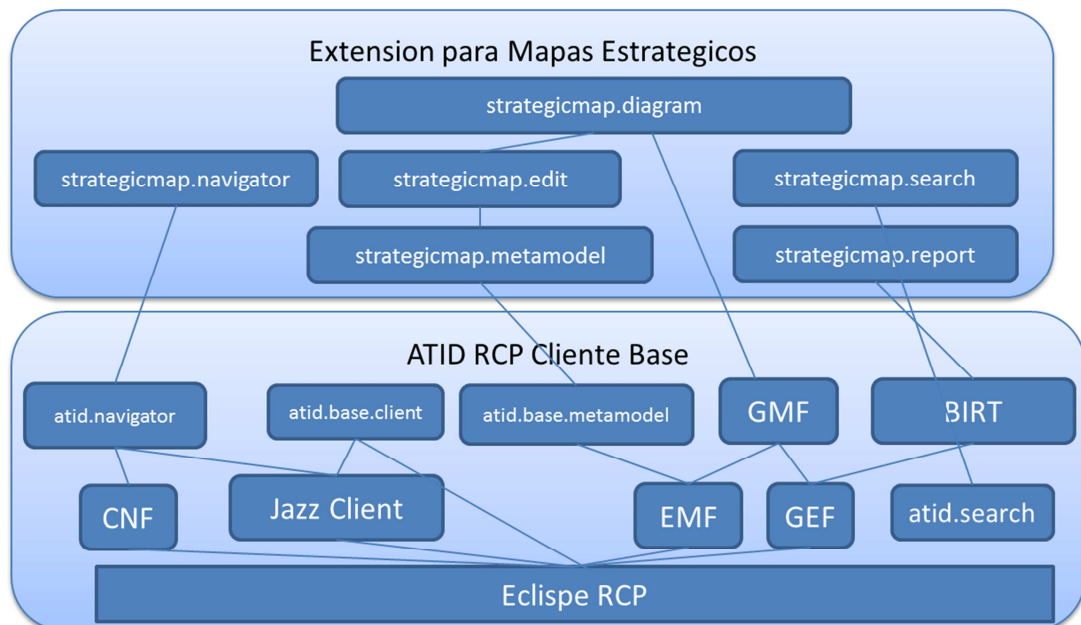
Los plugins del lado del cliente se comunican entre sí mediante el mecanismo clásico de eclipse de dependencias y definición de puntos de extensión. Del mismo modo los plugins en el lado del servidor se interrelacionan con dependencias a través de puntos

de extensión. En la siguiente sección se verá la estructura basada en servicios que permite relacionar los plugins del cliente con los del servidor.



**Figura 42 - Mecanismo de extensibilidad de jazz**

Por ejemplo, en la figura 43 se muestra un esquema extremadamente simplificado de la arquitectura para armar una aplicación de escritorio que utilice la ATID para proveer funcionalidad que permita crear y editar los mapas estratégicos. Un esquema similar de extensibilidad se lograría del lado del servidor. Como se verá más adelante para la tesina se desarrollaron varios plugins que corren tanto del lado del cliente como del lado del servidor.



**Figura 43 – Esquema simplificado de una extensión del cliente.**

Si bien esta fue la idea original, se encontraron dos grandes problemas a la hora de implementar estos conceptos. Primero, jazz está construido sobre el eclipse IDE y no el eclipse RCP, con lo cual habría que determinar los requerimientos mínimos de Jazz. Por otro lado Jazz provee un navegador denominado “Tema Artifacts” no compatible con frameworks de eclipse para la navegación como el CNF mucho más difícil de extender. Por último, como se detallara en las secciones siguientes, GMF es un framework orientado a recursos EMF mientras que en Jazz se usa EMF como fachada (pattern facade) de la base de datos, por lo que se hace muy difícil las integraciones. Se demostrara en el prototipo que se llegó a extender Jazz y conectar GMF con el repositorio, pero no se recomiendan las implementaciones de Jazz. Solo se rescata para la ATID el mecanismo de extensibilidad presentado en esta sección.

### **5.1.2 Comunicación cliente-servidor**

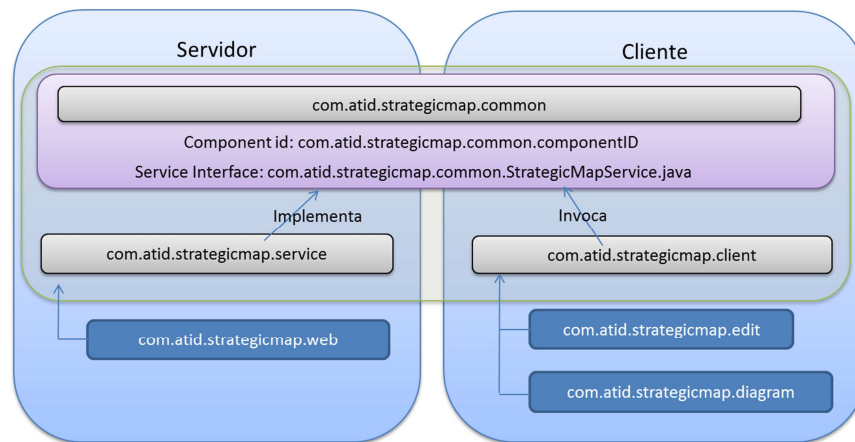
En esta sección se describirá el mecanismo y las decisiones tomadas en el desarrollo de la comunicación cliente-servidor. Vimos en la sección anterior que la extensibilidad del lado del cliente y del servidor se logra mediante plugins. Tanto dentro del cliente como internamente en el servidor la comunicación entre los plugins se logra estableciendo dependencias y utilizando los puntos de extensión. Ahora bien, para lograr la comunicación entre el cliente y el servidor, se estudió el mecanismo provisto por Jazz basado en servicios que se describe a continuación.

Si bien el plugin es la unidad mínima de extensibilidad, Jazz define el concepto de “*Component*” para referenciar un conjunto de plugins que permiten extender la plataforma jazz en un dominio en particular. Los componentes exponen su funcionalidad a otros componentes a través de servicios. Un componente puede tener uno o más servicios y típicamente está compuesto por cinco plugins. Existe un plugin cuyo nombre termina generalmente en “.common” que define al componente utilizando un punto de extensión proveido por jazz. Este punto de extensión además permite registrar los servicios que presta el componente. El plugin “.common” contiene además la declaración de interface del servicio. Este plugin es común a ambas partes (cliente y servidor). Del lado del servidor se implementa un plugin cuyo nombre termina generalmente en “.service”. Este plugin implementa la interfaz de los servicios definidos en el plugin “.common” y define las extensiones a la funcionalidad específica del lado del servidor teniendo acceso a los demás implementaciones de los componentes del servidor. También es común que del lado

del servidor se implemente un plugin denominado “.web” que implementa la funcionalidad que se accede desde el navegador web.

Del lado del cliente se implementa el plugin “.client”. Este extiende jazz proveyendo una interfaz de servicios que expone operaciones disponibles desde el punto de vista del cliente. Este plugin permite recuperar un proxy de la implementación del servicio en el servidor e invocar operaciones sobre él. [W22]

En la figura 44 se muestra un ejemplo simple del uso de los plugins para implementar un servicio de comunicación para el componente que implementa el Mapa Estratégico.



**Figura 44 – Ejemplo de implementación de una comunicación cliente-servidor**

Jazz exige para los métodos de la clase que implementa el servicio de lado del cliente (subclases de `ITeamModelledRestService`) comiencen con alguno de los siguientes prefijos: `POST`, `GET` o `PUT`. Esto guarda relación con el tipo de servicio WEB que se invoca, dado que se hace una conversión (Marshaling) entre el método y la operación HTTP que representa. Del mismo modo, existen clases implementan la interfaz `IParameterWrapper` que representan los parámetros enviados en el pedido HTTP. En la figura 45 se muestra de manera general, las interfaces anteriormente descritas. En el caso particular de la tesina, este mecanismo se utilizó en dos componentes de la ATID. Por un lado, para implementar un mecanismo de colaboración en tiempo real, enviando información de acciones realizadas al servidor. En la otra componente en que se utilizó este mecanismo es en la comunicación del editor de GMF con el repositorio de Jazz como se detallara más adelante. Respecto de su uso se puede mencionar la ventaja que significa poder abstraerse de los protocolos de comunicación, pensándola en términos más abstractos

de invocaciones a servicios. Uno de las posibles desventajas sería la falta de algún mecanismo que permita seleccionar el proveedor de servicios dinámicamente.

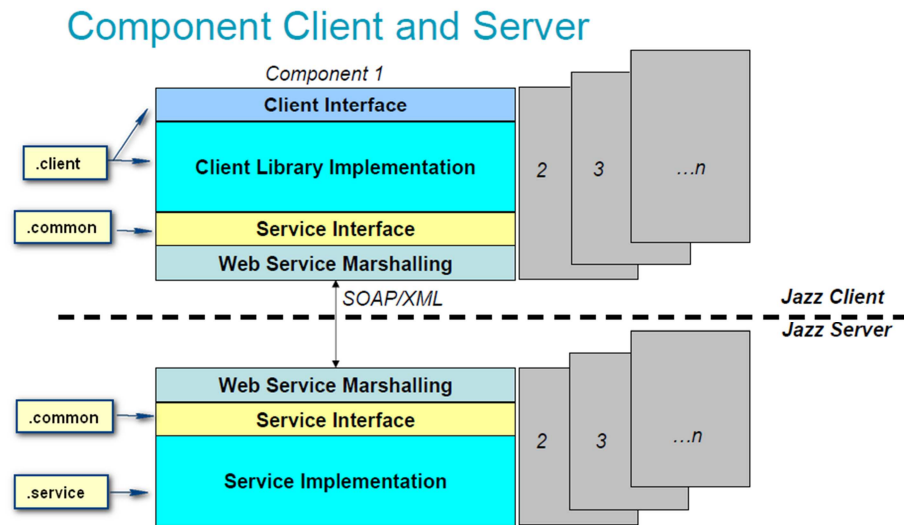


Figura 45 – Esquema de comunicación cliente-servidor de un componente.

### 5.1.3 Gestión de Usuarios

En este caso se experimentó con los mecanismos de manejo de usuarios de Jazz. La gran ventaja es que toda la funcionalidad para crear usuarios, asignar contraseñas, y los mecanismos de chequeo de identidad son provistos por la plataforma jazz. Los mecanismos se basan en el estándar LDAP para aplicaciones de servidor.

En el prototipo se experimentaron con varios aspectos de los usuarios. Por ejemplo, al estudiar mecanismos de colaboración, se vio la necesidad de asignar distintos colores a cada usuario. En este caso, fue muy útil la vista denominada "Team Organization" que permite visualizar a los todos los usuarios de un proyecto. Extendiendo esta vista se logro el efecto deseado (Figura 46)

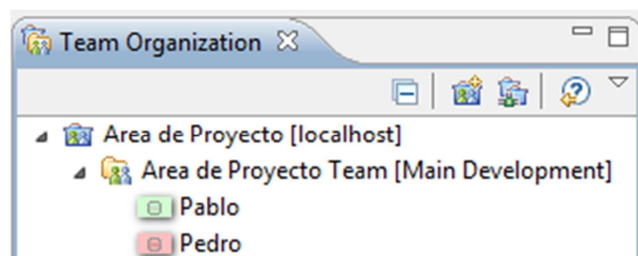


Figura 46 - Soporte para coloración de usuarios

Otra característica de Jazz, es que permite tener múltiples conexiones a distintos repositorios con distintos usuarios. Esta API si bien es muy flexible, dificulta la programación de los clientes pues para cada acción hay que llevar el rastro de quien

es el usuario que la originó y jazz no da soporte para ello. En general se concluye que el mecanismo de gestión de usuarios de Jazz, con mejoras como mejores soportes para colaboración, o un control de usuario más simple, sería adecuado para la ATID.

#### **5.1.4 Framework de aplicaciones**

En esta sección se detallan las facilidades que se pueden encontrar para la construcción de aplicaciones basadas en eclipse. Se estudiaron principalmente los mecanismos provistos por jazz para extensión de las interfaces y la construcción de editores GMF pero conectados con un repositorio remoto.

##### **5.1.4.1 Dominios**

Jazz provee una vista que permite visualizar los artefactos en el repositorio denominada "Team Artifacts". En este navegador se establece la conexión con el repositorio y se visualizan los proyectos asociados con el usuario actual. Uno de los mecanismos propuestos por jazz para la extensión de los clientes, es a través de un punto de extensión, que permite agregar un "Dominio". Los dominios en general se visualizan a nivel de los proyecto y mediante "Content Providers" que proporcionan el contenido del dominio a manera de árbol. [W24]

En el caso particular de la ATID se prototipo un dominio denominado Business Domain. Al crear un dominio, se debe proveer la clase que lo implementa. Esta clase debe ser subclase de Domain, la cual obliga a proveer un Content Provider y Label Provider, que son las clases que determinaran que información se muestra en la vista Team Artifact. En particular se buscó de implementar la posibilidad de crear un "Mapa Estratégico" agregando un menú contextual al dominio. La particularidad de esta información es que debe ser recuperada desde el servidor. El resultado de la implementación de la acción fue exitoso y permitió descubrir algunos problemas con la implementación de Jazz. Por un lado, el mecanismo del "Team Artifacts" no es tan flexible ni potente como los frameworks especializados para navegadores que posee eclipse (por ejemplo, el CNF) . por otro lado se mezcla información como las conexiones al repositorio con los elementos de modelado. Como conclusión se dedujo que sería más simple una implementación basada en CNF, lo cual se recomienda para la ATID.



#### 5.1.4.2 Diagrama de Mapas Estratégicos

El objetivo de esta sección no es el desarrollo de un complejo editor para mapas estratégicos, lo cual se sabe que sería posible dados los amplios ejemplos de complejos editores basados en GMF, sino que se enfocó el estudio y prototipación de los mecanismos para poder conectar GMF con el repositorio de Jazz.

El proceso de generación de editores basados en GMF se detalló en la sección 3.2.3. Siguiendo esos pasos se generó un editor simplificado de GMF, el mismo se muestra en la figura 47. Durante el desarrollo se encontraron una serie de problemas, que se resumen a continuación. En las siguientes secciones se explicaran otros detalles relacionados con el metamodelo y la conectividad con el repositorio.

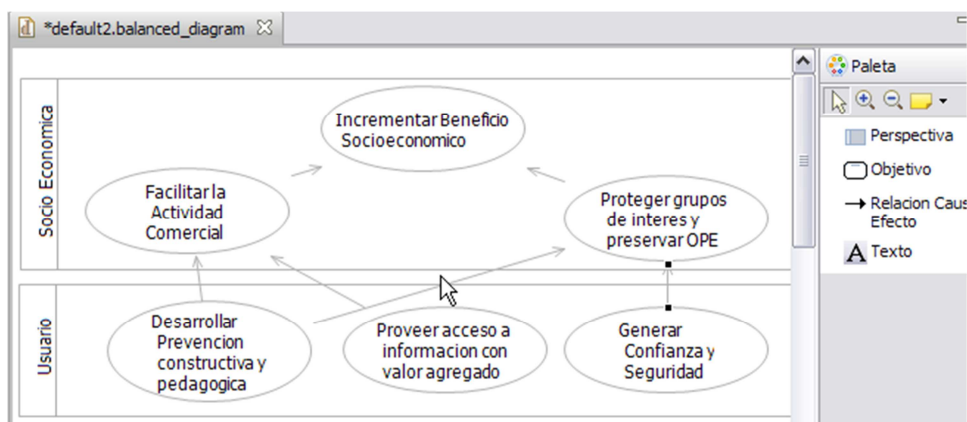


Figura 47 - Editor de Mapas Estratégico prototipado

- 1- El proceso de definición de GMF, posee poco soporte de herramientas lo que genera dificultades para detectar las causas de los errores que se generan. Las interfaces de edición son poco intuitivas y con escasa ayuda.
- 2- El código generado es bastante complejo y se hace difícil hacer personalizaciones del código generado sin arriesgarse a perder la posibilidad de sincronismo entre las especificaciones de alto nivel y el código.
- 3- Las complejas relaciones del metamodelo tuvieron que ser relajadas para poder reusar el mismo modelo EMF utilizado para el intercambio de información con el servidor, como se verá en la siguiente sección. Esto limitó muchísimo la implementación.
- 4- La implementación de GMF está fuertemente orientada a ver recursos como archivos lo cual dificultó la prototipación de la conectividad con el servidor. Por ejemplo existe una clase llamada WorkspaceSynchronizer que es invocada frecuentemente para recuperar el archivo asociado con el recurso

La solución implementada aprovecha las ventajas de la independencia de los mecanismos de serialización de EMF para inyectar la información obtenida desde el servidor de Jazz. Con esto se logró implementar el save y load remoto en el servidor. La clave fue la redefinición de dos objetos importantes de GMF: una es la clase EditorInput que es capaz de proveer el objeto a editar, la otra es el DocumentProvider, responsable por guardar o recuperar el objeto.

- 5- Una de las primeras diferencias que uno descubre que son necesarias respecto de la implementación tradicional de GMF es que el mecanismo que permite determinar si un editor tiene o no modificaciones necesita ser adaptado. Actualmente GMF utiliza listeners a nivel de resource para determinar si hubo algún cambio o no. Si hubo alguna modificación en el archivo entonces el estado de un editor puede ser marcado como Dirty. Sin embargo al trabajar en un modo remoto, el concepto de Dirty debe ser mejor definido. Por un lado, ya no existe el concepto de recurso como archivo, y la información que se recupera desde el servidor es solo información parcial del modelo. El estado Dirty solo debería verse modificado por el hecho de que el usuario realizó alguna acción desde los editores. Sino, cambios hechos desde otros editores podrían modificar el estado de los otros editores.
- 6- Un problema que se encontró aquí está relacionado con la forma en que se agregan objetos al repositorio ya que Jazz exige que las operaciones de agregación se hagan en una única transacción en el servidor. Esto obliga a definir un método genérico de agregación del lado del servidor el cual es invocado cada vez que se agregan objetos en el diagrama. De esta forma esos cambios también pueden ser inmediatamente notificados al resto de los clientes

La ventaja del enfoque utilizado es que logra aportar una capa de abstracción debajo de GMF que permitiría la conectividad con el repositorio de jazz. La gran desventaja como se detallara más adelante son debidas a las carencias de jazz para soportar metamodelos complejos, sumándose otros factores como el hecho de no poseer jazz soporte para notificaciones de eventos lo que hace que la cantidad de esfuerzo requerido para lograr una implementación sea muy grande. Estas desventajas se hicieron más notorias con la aparición de CDO en los últimos meses en su versión 3

junto con el proyecto Dawn, el cual permitiría obtener los resultados esperados en tiempos mucho más cortos.

### **5.1.4.3 Metamodelos**

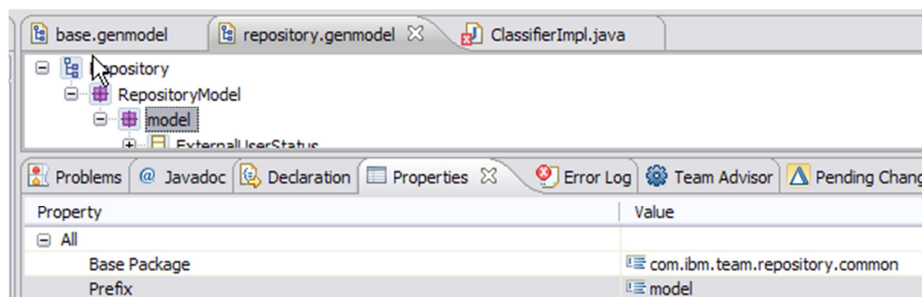
Esta es también una de las áreas que es diferenciadora en el planteo de la ATID respecto de las arquitecturas tradicionales. El objetivo buscado por la ATID es la posibilidad de que las distintas extensiones enriquezcan el metamodelo dinámicamente con sus propias entidades. Se reconoce la necesidad de una fácil extensibilidad y la posibilidad de mecanismos y API específicos para el agregado e integración de nuevas técnicas. Para ello se eligió la tecnología EMF para el desarrollo de los mismos dado las ventajas explicadas en el capítulo 2. Un detalle que hay que tener en cuenta es que existen al menos dos niveles de modelación, el de las entidades de negocio propiamente dichas, y los metamodelos de soporte para los editores gráficos.

En este contexto, uno de los primeros desarrollos de implementación fue el meta-modelo para los editores de GMF usando EMF. Aquí se pensó originalmente el desarrollo de una arquitectura de plugins que permitiese el reuso de entidades. De esta forma se crearon 2 editores de GMF cada uno con 3 plugins (metamodelo, .edit y .diagram), a su vez se creó un metamodelo general denominado base en donde se pensó poner las entidades comunes compartidas que los otros editores podían reusar. Lo mismo sucedía para los metamodelos de los editores gráficos, creando 3 plugins adicionales más (con extensión .viewmodel) en donde se representaban las abstracciones requeridas por los editores gráficos. Se llegó a la conclusión que la mayoría de los diagramas que se representaban (sino todos) pueden ser pensados abstractamente como grafos. Por ello se generó un plugin reusable general para la representación genérica de grafos (con extensión .graph), permitiendo de esta forma la construcción rápida de cualquier tipo de diagramas. Por ejemplo, en este esquema, las perspectivas empresariales son también nodos de grafo y las restricciones visuales entre los objetivos y las perspectivas se imponen con algoritmos de layout. Este esquema parecía funcionar bien dado las características estudiadas de GMF y se llegó a completar el proceso de generación de los dos editores.

El problema surgió al estudiar el mecanismo de soporte de metamodelos de jazz. No era de esperarse que jazz pusiese restricciones en la manera que se generan los modelos EMF, menos aún descubrir que jazz re implementaba la forma en que se

generaban los modelos EMF haciéndolos incompatibles con los necesarios por GMF. Se darán más detalles en la sección 5.1.5 pero se comenta aquí una de las formas en que se logró poder generar los modelos GMF usando el EMF modificado por Jazz. Tanto EMF como Jazz comienzan con la declaración del metamodelo en un archivo con extensión `.ecore`, la diferencia es que Jazz requiere una estructura interna particular con anotaciones especiales para cada entidad relacionadas con parámetros de generación de una base de datos, como opciones de indexación, representación en BD, etc. El proceso de generación del código a partir del metamodelo, que en EMF se logra creando un nuevo archivo denominado `.genmodel`, en Jazz se realiza directamente con una opción de menú contextual sobre el archivo `.ecore` denominada “Generate Jazz Component”. Para poder usar el mismo `.ecore` en el proceso de generación de GMF, se requiere una cuidadosa implementación y seteo de las propiedades denominadas `BasePackage` y `Prefix` (Figura 48)

Hasta aquí se había logrado conectar los modelos de EMF de jazz con los de GMF evitando usar complejas transformaciones. Logrado este objetivo se descubrieron nuevas restricciones respecto de las representaciones EMF de jazz que se detallan a continuación. A pesar de la importancia de estas restricciones, las mismas se encontraron documentadas solo al final de un tutorial de ejemplo [W25].



**Figura 48 – Propiedades importantes para la generación de GMF a partir de Jazz.**

Una de las restricciones que impone Jazz para la generación de metamodelos es respecto de los supertipos de las entidades y está relacionado con el primitivo esquema de generación de base de datos que utiliza. Jazz exige que cualquier entidad del repositorio sea subclase directa de las clases `Auditable`, `SimpleItem`, `Helper` o alguna clase abstracta que herede de ellas. Los ítems auditables son los que conservan una historia de los estados previos. Los ítems simples solo llevan un id que indica si fueron modificados o no. La otra restricción importante esta relacionada con las referencias a otros Items. Jazz solo permite referenciar a entidades del

componente de repositorio de jazz (SimpleItem, Auditable, UUID, etc) o a entidades definidas en el propio componente. Es decir que no se pueden hacer referencias directas a entidades en otros componentes.

Estas limitaciones atacan los conceptos de libre extensibilidad de metamodelos planteados inicialmente en esta sección y lamentablemente son insalvables. Por ello una de las decisiones para intentar seguir adelante con el prototipo, fue la de replicar los metamodelos en cada componente. Finalmente se concluyó que todas estas restricciones de Jazz no tienen fundamento e implementaciones como CDO o Hibernate demuestran que estas limitaciones son innecesarias, por ello es que como conclusión se propone para la ATID mecanismos de repositorios basados en CDO descartando los de Jazz.

#### 5.1.4.4 Creación de una extensión web usando Jazz

Por ultimo se comenta brevemente la capacidad testeada de Jazz para generar aplicaciones web. El mismo se describió brevemente en la sección 4.3.1. El paso principal es la definición de un plugin del lado del servidor con extensión (.web). Es necesario agregar las correctas dependencias a plugins de la plataforma jazz. La implementación está fuertemente basada en Ajax y Dojo. La forma en que jazz permite extender las interfaces web es a través de la definición de aplicaciones. Cada aplicación aporta un conjunto de páginas, Los pasos necesarios para extender la ui son los siguientes:

1- se debe especificar que el plugin desarrollado extenderá la arquitectura web de jazz

```
<!-- Jazz Ajax Framework web bundle marker -->
<extension point="net.jazz.ajax.webBundles"></extension>
```

2- Se debe especificar la aplicación y el archivo javascript que la implementa

```
<!-- ATID Web App -->
<extension id="atidApplicationID" point="net.jazz.ajax.applications">
  <application alias="/web"
    jsclass="atid.base.web.ui.internal.ATIDApplication"
    icon="/ui/graphics/JazzFoundation/JazzIcon.ico"
    statusHandler="atid.base.web.internal.JFSStatusHandler"/>
</extension>
```

De esta forma al seleccionar un proyecto desde la página web se visualizará nuestra aplicación, pudiendo así luego agregar páginas. El principal problema que presenta

este mecanismo es la falta de editores especializados para código JavaScript y notación Dojo usada por jazz.

### **5.1.5 Repositorio**

En esta sección se documentan los detalles de la implementación del repositorio de la ATID para lo cual se intentó utilizar las principales características del repositorio de Jazz. Parte de los aspectos de la generación de los metamodelos y archivos EMF requeridos se detallaron en las secciones anteriores. El foco de esta sección se pone en los detalles de la generación de las componentes de jazz que extienden la base de datos del repositorio y las características de las operaciones permitidas en la implementaciones del lado del servidor.

#### **Extensión del Repositorio de Jazz**

En Jazz cada componente que se desea agregar al repositorio del servidor puede contribuir con un metamodelo. Para ello se deben realizar las acciones que se describen en esta sección [W26].

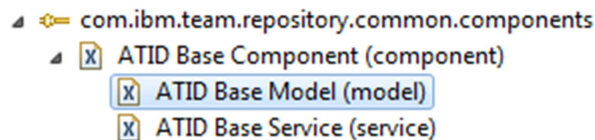
Como se describió anteriormente, para poder almacenar objetos en el repositorio de jazz aprovechando las capacidades automáticas de indexación, historial, búsquedas y servicios asociados, jazz provee un mecanismo propio para la generación de código. El mecanismo es muy similar al empleado por EMF enfocado en MDD. Jazz incluye un repositorio extensible que provee una ubicación centralizada para la información específica de cada herramienta. La información es almacenada en el repositorio en términos de objetos de alto nivel denominados ítems. El repositorio incluye dos tipos de ítems, los auditables, que mantienen la historia de la creación y subsecuente modificación de los mismos para propósitos de auditoría. Esto incluye un registro de los estados anteriores, el usuario que lo guardo, y el tiempo en el que se lo modificó. Para los ítems que no requieren historia, el repositorio almacena solamente el último estado.

Cada ítem del repositorio tiene un identificador único, que puede ser usado como clave para recuperarlo. Estos se denominan identificadores únicos universales (UUIDs) El repositorio de jazz provee una API (IRepositoryItemService) del lado del servidor que permite crear, recuperar, actualizar y borrar ítems del repositorio. El repositorio de jazz se implementa mediante un mapping sobre una base de datos relacional. (Derby, IBM DB2, Oracle, SQL Server). Se utiliza una combinación de

modelos ecore de EMF mas anotaciones para hacer un mapping relacional y almacenar la información en la base de datos del servidor [w27].

La creación, actualización y borrado de items en el repositorio solo puede ser hecho usando los mecanismos del lado del servidor.

Para extender la BD se necesita declarar el modelo usando el punto de extensión que declara componentes. Automáticamente jazz buscará el modelo ecore y lo registrará en la plataforma.



Con esta declaración del modelo que apunta al archivo .ecore que contiene la definición del metamodelo, es suficiente para utilizar alguno de los métodos proporcionados por Jazz para crear la base de datos [W28]. Todos estos métodos son bastante primitivos, complejos y con gran probabilidad de fallar si no se tiene mucho cuidado. Uno de ellos utiliza configuraciones basadas en el framework de testing jUnits Otros método consiste en una herramienta de línea de comandos denominada repotools provista por Jazz para tal fin. El proceso requiere detener el servidor y la base de datos y se regeneran por completo todas las tablas de todos los componentes con lo que toda la información almacenada se pierde debiendo ser migrada con posterioridad.

Debido a la arquitectura de separación de los modelos ecore que se eligió para el prototipo se debió crear un componente por cada plugin definido. Además jazz permite exponer el modelo solo a través de un servicio. Es por eso que debemos definir un servicio de acceso para cada componente.

```
<extension
  point="com.ibm.team.repository.service.serviceProvider">
  <serviceProvider
    componentId="atid.strategicmap.viewmodel.common.component"
    implementationClass="atid.strategicmap.viewmodel.service.ABAStrategicmapeditorService">
    <provides>
      <providedService
        interface="atid.strategicmap.viewmodel.common.IABAStrategicmapeditorService">
      </providedService>
    </provides>
    <prerequisites>
      <requiredService
        interface="atid.team.repository.common.service.IQueryService">
      </requiredService>
      <requiredService
        interface="atid.team.repository.service.IRepositoryItemService">
      </requiredService>
    </prerequisites>
  </serviceProvider>
</extension>
```

La implementación de los servicios se realiza del lado del servidor. Para ello se crea por cada plugin “.common” un plugin “.service” que provee una implementación para los servicios. Estos plugins solo serán instalados en el servidor. Se debe crear la clase que implementa el servicio y contribuir al punto de extensión que declara el servicio provisto como se detalló en el segmento de declaración XML anterior.

Uno de los servicios que se implemento era el que permitía guardar el modelo asociado con el mapa estratégico en el servidor. El problema que se encontró aquí es una restricción de Jazz para que el estado de un ítem sea modificado dentro de una transacción y la falta de soporte para notificaciones. Cuando se recupera un ítem del repositorio de jazz se recupera un handler al mismo. Del lado del cliente se crea el concepto de workingCopy, que es una estrategia que está ampliamente sobreusada en toda la implementación de Jazz. El uso de la misma intenta evitar la posibilidad de instanciar más de una vez el mismo objeto en el cliente. Esto sería mejor implementado con una componente genérica del framework. Sin embargo no resuelve una parte importante del manejo compartido como la edición simultanea de cada objeto por parte de los editores. Esto hace que por ejemplo, si quisiésemos hacer el save del editor no tenemos garantía que el estado que queremos guardar no haya sido modificado previamente por otros usuarios, por lo que habría que implementar algún algoritmo para detectar posibles conflictos y resolverlos. La solución que se encontró para este caso era implementar una marca de estado para cada Item de modo de poder saber para un ítem dado si había sido modificado o no. Esto conjuntamente con un mecanismo de notificaciones a los clientes, podía mantener bajo control los posibles conflictos. Esta implementación propuesta es muy similar a la lograda por CDO, por ello, se concluyó que convenía más explorar el uso de CDO como repositorio antes que realizar toda la compleja implementación de una funcionalidad similar en el servicio de Jazz.

## **5.2 Componentes de Valor Agregado**

Para los componentes de valor agregado se exploraron los soportes de Jazz y otras herramientas complementarias como el RMC o EPF. como se detallarán en la siguientes secciones.



### 5.2.1 Búsquedas

Respecto de las búsquedas se exploraron los mecanismos de queries proveídos por jazz. La API que permite realizar consultas sobre los ítems en el repositorio se denomina Query Repository API. Por ejemplo el siguiente es un fragmento de código que permite recuperar el valor de un atributo.

```
IItemQuery query= IItemQuery.FACTORY.newInstance(WorkItemQueryModel.ROOT);
IPredicate keyPredicate=
WorkItemQueryModel.ROOT.stringExtensions().key()._eq(query.newStringArg());
IPredicate valuePredicate=
WorkItemQueryModel.ROOT.stringExtensions().value()._like(query.newStringArg(),
ESCAPE_CHARACTER);
query.filter(keyPredicate._and(valuePredicate));
IItemQueryPage page= queryService.queryItems(query, parameters, pageSize);
```

Lo que se ve en este ejemplo, es que jazz define un dialecto propio para este tipo de operaciones no fácil de entender, y además la API no es estable lo que puede tener variaciones con las distintas versiones. Por ello se recomienda para estos componentes explorar la posibilidad de incorporar plugins en eclipse basados en otras tecnologías como Lucene.

### 5.2.2 Gestión de metodologías

Jazz permite organizar el espacio del servidor organizándolo en proyectos. La particularidad es que para cada proyecto Jazz exige que se asocie un proceso o método. De esta forma es posible controlar ciertos aspectos como la definición de roles, permisos para los usuarios y soporte para la definición de iteraciones de procesos basados en ciclos continuos. También permite definir pre y pos condiciones para acciones registradas en el servidor. Otra característica importante es que controla la inicialización del espacio de trabajo, de manera de poder crear templates de la estructura del proyecto [W29].

Todo esto es aún más valioso si se posee la documentación de la metodología, disponible en un sitio que sea adecuado para búsquedas y navegación. Para el prototipo se analizó la herramienta Rational Method Composer (RMC), que permite definir una metodología y genera un sitio web navegable [W32]. Si bien su utilización no es simple ni intuitiva, sería adecuado para intentar formalizar las prácticas de CMI, como por ejemplo, el proceso de ciclo continuo de 6 etapas visto en el capítulo 2, o el proceso iterativo de 9 etapas descritas en el capítulo 4.

Además RMC permite exportar las actividades típicas del proceso como workitem templates. Los workitem templates son moldes para actividades de Rational Team Concert (RTC) una herramienta especializada en el seguimiento de ciclos de desarrollo de sistemas de software. RTC permite la personalización de los workitems de manera de poder adecuarlos a otras prácticas como CMI [W31]. Los workitem templates se asociarían el proceso definido en Jazz, de manera de poder generar las actividades típicas para una iteración en particular y realizar así el seguimiento. Todas estas opciones fueron testeadas encontrando como problema principal, la dificultad de las herramientas y mecanismos de extensión, dado la poca documentación o dificultad de uso y la complejidad de las interfaces que se obtienen a pesar de las simplificaciones que permite realizar RTC sobre los editores de workitems.

### **5.2.3 Extensiones para Colaboración**

La plataforma jazz, solo provee mecanismos de integración con sistemas de chat y servidores de email. No provee mecanismos de colaboración en tiempo real. Para explorar la posibilidad de extender las capacidades de colaboración de la plataforma jazz, se prototipó un framework que interactuaba a nivel de la pila de comandos de GMF creando servicios y entidades que representaban a los mismos, enviando las representaciones de los comandos efectuados sobre los archivos al servidor. En el servidor se apilaban los comandos y todos los editores consultaban esta pila mediante un servicio especial, cada cierto intervalo de tiempo para saber si en otros editores se habían producido cambios. Con este mecanismo se intentó crear una funcionalidad similar a la que proveía Google wave, de colaboración en tiempo real, sin embargo, se presentan dos graves problemas, uno es el constante pedido de servicios por parte de los editores para preguntar si ha habido algún cambio. El otro problema es que no resuelve situaciones anómalas en donde varios usuarios realizan acciones conflictivas sobre los mismos elementos del Mapa Estratégico. La implementación más avanzada de estos mecanismos son las presentadas por CDO, que posee una arquitectura robusta para soportar transacciones, notificaciones en tiempo real y lock de objetivos.

Una conclusión interesante de esta experiencia de implementación es la posibilidad que se planteaba en el prototipo desarrollado de asignar distintos colores a los usuarios de manera de poder diferenciar su selección en pantalla. Esto serviría

por ejemplo, si dos usuarios están operando en distintas computadoras sobre el mismo Mapa Estratégico, de manera de que cada uno sepa sobre qué objetivo o perspectiva está operando de manera de minimizar los conflictos. Esto no es proveído actualmente ni por jazz ni por CDO y se deja como futuro trabajo.

#### 5.2.4 Alertas y notificaciones

En este contexto solo se encontraron en Jazz componentes que permiten la notificación por email, utilizado generalmente para alertas de poca gravedad. En caso de extrema gravedad de algún indicador sería deseable poder recibir alertas en tiempo real que desvíen la atención para resolver algún tema urgente. Se encontró y testeó un punto de extensión de Jazz que permite abrir pequeñas ventanas de notificación en la pantalla cerca del borde inferior derecho [W33]

```
<extension
  point="com.ibm.team.foundation.rcp.core.notification">
  <eventcategory
    id="atid.rtc.launcher.eventcategory"
    name="Categoria">
    <eventtype
      id="atid.rtc.launcher.eventtype"
      name="Ha ocurrido un evento">
    </eventtype>
  </eventcategory>
  <trigger enabled="true" eventCategoryId="atid.rtc.launcher.eventcategory"
    eventType="atid.rtc.launcher.eventtype" id="com.ibm.aba.rtc.launcher.trigger"
    minimumPriority="Normal"
    name="Descripcion de evento" notifierId="com.ibm.team.jface.AlertNotifier"
    showInfos="true" />
</extension>
```

Como conclusión la ATID sugiere poseer componentes dedicados para soportar la posibilidad de enviar y recibir alertas y notificaciones. No se han encontrado componentes equivalentes o que puedan usarse como referencia para la implementación. Debido a limitaciones de tiempo, el desarrollo de esta area se deja como posible extensión.

### 5.3 Componentes de Análisis y Reportes

Los componentes de análisis de la ATID son los que facilitan armar visualizaciones que simplifiquen el análisis sobre los distintos editores. En este contexto se destaca la capacidad de crear heatmaps o mapas de colores para representar el estado de distintas variables gráficamente. En este caso la idea era crear una serie de plugins que provean de diálogos y vistas reusables que permitan la fácil creación y armado de este tipo de visualizaciones. De este modo, la persona que esté realizando una

extensión como por ejemplo, desarrollando un análisis sobre un mapa estratégico, podría crear un plugin de análisis que aprovechara esta funcionalidad genérica personalizándola para el análisis particular. Nuevamente debido a restricciones de tiempo y a la falta de otras componentes que deben ser desarrolladas previamente se dejan estas componentes como extensiones futuras.

Respecto de los componentes de reportes, se comprobó que Jazz se beneficia de los desarrollos de otro proyecto denominado Business Intelligence and Reporting Tools (BIRT) para permitir la creación de Dashboards en sus interfaces web. Estas funcionalidades de BIRT tienen años de desarrollos y podrían ser aprovechadas por la ATID. También podría ser beneficioso el análisis de los mecanismos provistos por RTC para la generación de reportes basados en BIRT [W34].

Por último, respecto de los componentes de Datawarehouse (DW) se comprobó que jazz define un DW extensible [W35], con lo cual es posible definir ciertos intervalos de tiempo en los cuales la información presente en el repositorio se consolidara en el DW. Esta información puede estar luego accesible para los reportes y análisis. Se deja como trabajo futuro el análisis de estos mecanismos de extensibilidad y las conclusiones referentes a las ventajas o desventajas de la estrategia utilizada por Jazz para resolver este problema.

## **5.4 Componentes de Interoperabilidad y otros conceptos**

En esta sección se hace referencia principalmente a tres conceptos. Uno de ellos, el de la interoperabilidad utilizando tecnologías como OSLC ya fue detallado en la sección 4.3.4. Al momento de desarrollar la tesina estas tecnologías estaban en su infancia con solo 3 herramientas basadas en el estándar. Actualmente se está incrementando rápidamente la aparición de nuevos productos y sus necesidades de integración por lo que los mecanismos de OSLC se refinan y mejoran constantemente. El estudio de los estándares de OSCL para interoperabilidad de deja como trabajo futuro dado su complejidad inherente.

En segundo lugar se mencionan las pruebas exitosas para permitir la internacionalización de las extensiones. Por cada plugin, se genera un plugin adicional, por ejemplo, con extensión “nl\_es” que es el que contiene las traducciones específicas de lenguaje. Luego en la configuración de arranque de la herramienta se pasa como parámetro el lenguaje con el que se desea arrancar la herramienta (por ejemplo -nl es para español). Las traducciones de los plugins de la plataforma

eclipse fueron donadas para la versión 3.2. Las aplicaciones RCP deberían implementar sus propias traducciones.

El tercer concepto para destacar es referente a la escalabilidad y robustez de la plataforma jazz, hoy en día herramientas como RTC basadas en jazz son usadas internamente por miles de empleados de IBM, y las mejoras al respecto se publican con cada nueva versión de la plataforma.

## **5.5 Configuraciones de la Arquitectura**

Esta sección se estudió principalmente dos conceptos. Uno referido a las posibilidades que brinda la plataforma para armar distintas configuraciones como se explicó en la sección 4.3.5 En este contexto se considera que la arquitectura provista por Jazz es lo suficientemente flexible como para soportar todos esos casos, por lo que se podría reusarse para la ATID.

Respecto del soporte para trabajar desconectado, se comprobó que la plataforma Jazz no provee en general mecanismos que permita trabajar sin conexión al servidor. Esto se ve reflejado en RTC, donde por ejemplo, si no se dispone de conectividad no se puede ver la estructura del proyecto. Peor aún, un problema de usabilidad muy grave de Jazz en general es esperar hasta último momento para hacer las verificaciones de permisos o problemas de conexión. Podría pasara que luego de haber editado un workitem, la plataforma informe al usuario que no tiene permiso para guardarlo. De la misma forma si se pierde la conectividad con el servidor y se han realizado cambios en un editor, estos no podrán ser guardados hasta que se restablezca la conexión. La arquitectura ATID intenta preservar la posibilidad de guardar localmente el modelo. La propuesta más prometedora para soportar esta funcionalidad la provee nuevamente el proyecto CDO. Como ya vimos, CDO contempla la posibilidad de crear clones de los repositorios, en donde los clones se mantienen en sincronismo con el repositorio maestro. De esta forma si la conexión con el repositorio maestro se cae, aún existe la posibilidad de trabajar temporalmente con el repositorio clone. Sin embargo surgen varias preguntas relacionadas con la forma en que se resuelven situaciones no deseadas como la edición múltiple, la duplicación de información, etc. Todas estas situaciones representan un interesante caso de estudio para futuros desarrollos.

## 5.6 Conclusiones del Capitulo

En esta sección se detallaron los principales desarrollos de implementación que se efectuaron para intentar implementar los conceptos de la ATID.

Respecto de los mecanismos de extensibilidad, se vio el potencial de conservar el exitoso paradigma de extensibilidad de las aplicaciones eclipse de escritorio, utilizando el mismo concepto de plugins, dependencias y puntos de extensión para extender el servidor. Sin embargo se corroboró al intentar prototipar una extensión para un editor de mapas estratégicos, que los plugins de Jazz construidos con estos principios no tienen en cuenta a las extensiones más populares de eclipse como el CNF o GMF, generando complicaciones a la hora de poder armar las arquitecturas flexibles que la ATID propone. Por ello se concluyó mantener en la ATID el principio de extensibilidad pero no todos los plugins de Jazz construidos sobre estos mecanismos. La comunicación entre el cliente y servidor se logra mediante la implementación de componentes de Jazz que son conjuntos de plugins que brindan servicios relacionados con un dominio en particular. En este caso, los tests de prototipación de los mecanismos de Jazz, resultaron ventajosos para poder establecer una comunicación cliente servidor, abstrayendo los protocolos y conversiones intermedias. En cuanto al manejo de usuarios, se probaron los mecanismos provistos por Jazz observando las ventajas del reuso de esta implementación con los mecanismos de autenticación y seguridad avanzados. Se propuso como posible extensión mecanismos de coloración que fueron prototipados exitosamente. Se detectó la excesiva complejidad de los mecanismos de conexión con servidores remotos del lado de los clientes.

El framework de aplicaciones de la ATID es uno de los componentes claves en el éxito de la arquitectura ya que determina la posibilidad de rápida prototipación. Sin embargo en los resultados de implementación, si bien se puede crear un editor de mapas estratégicos en un tiempo corto, se encontraron obstáculos no previstos para conectarlo con Jazz, debido a carencias tanto del lado de GMF como del repositorio de Jazz, que mostraron como estas tecnologías aun no están listas para lograr la rápida prototipación de editores, La aparición de mejores tecnologías como CDO y Dawn hacen necesario la investigación de la conectividad remota usando otras tecnologías de implantación. Esto de deja como un posible trabajo futuro.

Respecto de la gestión de metamodelos los resultados de implementación ayudaron a clarificar las verdaderas restricciones de la plataforma jazz para los tipos de metamodelos soportados. Aparte de las restricciones relacionadas con la estructura interna de los archivos .ecore de EMF y las confusas meta anotaciones exigidas por jazz relacionadas con conceptos de base de datos de bajo nivel, se agregan las limitaciones de herencia simple y no poder referencias a elementos en otros componentes, limitaciones que atentan contra los principios fundamentales de extensibilidad y dinamismo de la ATID. Nuevamente se reitera que el surgimiento de tecnologías como CDO abren las puertas a explorar nuevas forma de implementar los componentes principales de la ATID.

El repositorio es otro de los puntos débiles encontrados en la implementación de Jazz. Tanto el proceso de generación de las tablas de la base de datos a partir del modelo .ecore como los mecanismos que obliga a implementar jazz para poder operar con los ítem del repositorio representan graves dificultades para la implementación de las capacidades expresadas en la visión de la ATID.

Si consideramos los componentes de valor agregado, se determinó que el soporte para búsquedas provisto por jazz lo realiza con una API propietaria, difícil de comprender y con un estado beta, por ello, se sugiere para la ATID la exploración de otras tecnologías como EMF Index, o Lucene. Los componentes para soporte de metodologías son muy útiles como guías y control del seguimiento adecuado de las metodologías. Mecanismos más simples que los estudiados para implementar los métodos y procesos podrían ser convenientes para la ATID. A los mecanismos de colaboración que provee jazz como soporte para chat y envió de emails, se sugiere agregar mecanismos de colaboración en tiempo real como los provistos por CDO más la posibilidad de agregar awareness, cosa que no es brindada actualmente por ninguna de las tecnologías estudiadas.

Por último se comentaron desarrollos relacionados con los componentes de Análisis, Reportes, interoperabilidad y otros. Como conclusión de todos los intentos de desarrollo se comentan las dos caras percibidas para la plataforma Jazz. Por un lado las ideas subyacentes, y por otro lado la implementación. Pueden verse como dos mundos completamente desconexos. Desde el punto de vista de los principios fundamentales de su arquitectura Jazz provee un enfoque muy potente. Sin embargo, lo que en ningún lado está documentado y que pocos llegan a enfrentarse es la realidad de su implementación. Por ejemplo, el solo hecho de instalar el ambiente

SDK de jazz constituye un filtro para el 99% de los programadores sin experiencia en el desarrollo en eclipse, y los pocos que tienen éxito se encuentran con gran cantidad de restricciones escasamente documentadas y una muy mala calidad de implementación de código fuente que hacen muy difícil cualquier extensión o desarrollo o mejora. Principios básicos de programación aprendidos durante el transcurso de la carrera son reiteradamente violados en el código fuente de Jazz. Las API no siempre tienen un fundamento coherente, es común encontrarse con métodos que requieren gran cantidad de parámetros no relacionados con la acción que se intenta ejecutar, son algunos de los casos enfrentados durante los desarrollos.

Sin embargo algunos de los principios de jazz son muy poderosos y se rescataron durante las descripciones de cada una de las secciones anteriores.



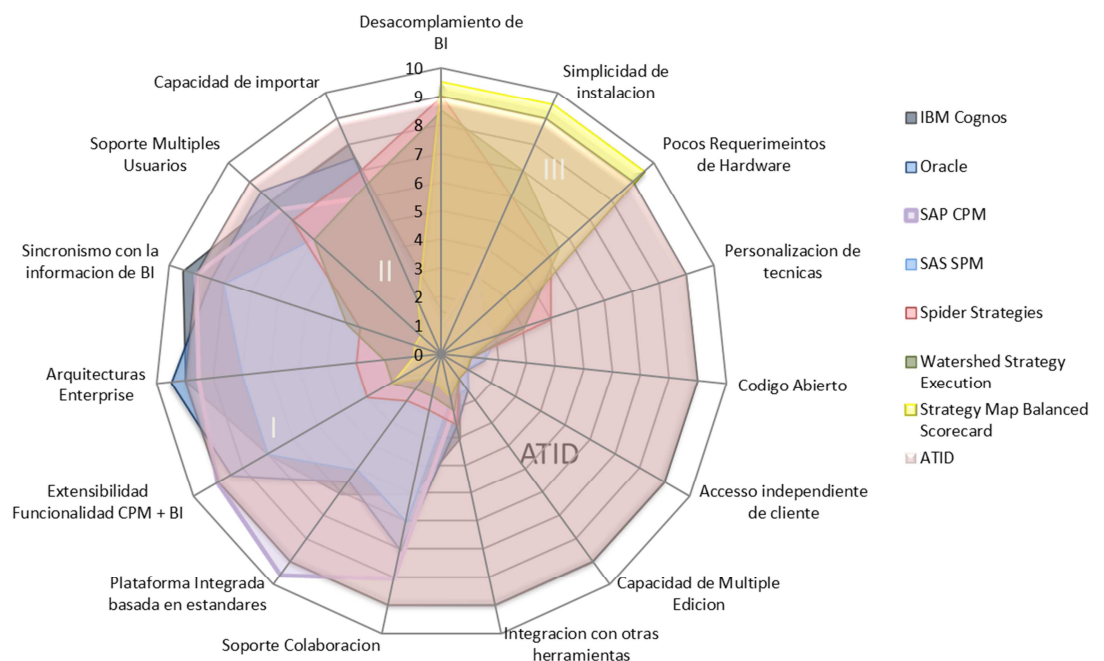
## **6. Conclusiones y Trabajo Futuro**

En esta sección se resumen las principales conclusiones, experiencias y aportes generados durante el desarrollo de la tesina. Anteriormente se finalizó cada capítulo con un resumen en el que se destacan las ideas principales capturadas durante el desarrollo de cada capítulo. En esta sección se enumeran las conclusiones más importantes, los principales aportes tanto para el campo de las ciencias informáticas como el del área estudiada y los posibles trabajos y extensiones futuros.

### **6.1 Conclusiones**

- 1- El área de gestión del rendimiento de negocios (CPM) permite modelar estrategias, traducirlas en planes y monitorear su ejecución con el objetivo de mejorar el rendimiento financiero y operacional de las empresas. Las técnicas de esta área, representadas por la de Balanced Scorecard han pasado de ser simples mecanismos de creación de tableros de control, hasta convertirse hoy en día en complejas metodologías que permiten desarrollar, comunicar y ejecutar la estrategia de manera de lograr el alineamiento de toda la empresa. Sin embargo la aplicación de CPM en todo su alcance se ve reducida por la falta de software adecuado para el tratamiento de la misma y que esté al alcance de todas las empresas. En esta tesina se creó la arquitectura dinámica empresarial ATID basada en los mecanismos de extensibilidad de eclipse, jazz y otros desarrollos como respuesta a las necesidades del área.
- 2- En muchos casos se confunde CPM asignándola como parte de otras áreas como por ejemplo, la inteligencia de negocios (BI) En base al estudio en esta tesina se concluye que CPM debería desarrollarse como un área independiente. Un de las diferencias clave es que BI intenta dar soporte para la toma de decisiones enfocándose en extraer valor de la información histórica o diaria mientras que CPM va un paso más allá al ofrecer mecanismos de modelación de estrategia que puede tener en cuenta tendencias y predicciones que no siempre dependen de los datos históricos de las empresas sino de eventos externos.
- 3- Otra importante conclusión está relacionada con el análisis de las herramientas actualmente existentes y que intentan brindar funcionalidades para cubrir el área de CPM. La información analizada se usó para crear un

gráfico en el que se muestran las características necesarias y deseables de las herramientas. Se manifiesta la carencia en el área de las arquitecturas de este tipo de implementaciones que sean más flexibles y que provea todas las componentes necesarias para contemplar los casos de uso relevantes. Esta falta de claridad en el área de herramientas existentes, es uno de los problemas que dio origen a la propuesta de esta tesina, es decir, explicar una arquitectura de software que sustente las herramientas necesarias para el area, teniendo en cuenta todos los detalles importantes como la extensibilidad, personalizaciones y soporte de casos de uso típicos como repositorios de información, integración o extensiones de otras técnicas, etc. Un riesgo cuando se comenzó la tesina era la posibilidad de encontrar que existía alguna herramienta con una arquitectura sólida que responda a las inquietudes planteadas. Luego de más de un año de investigación, la respuesta sigue siendo la presunción original. No existe una arquitectura sólida que permita llenar ese gap. Por ello se definió la ATID la cual logra completar las faltantes encontradas para las necesidades de los usuarios y esa es una de las contribuciones más importantes de esta tesina. (Figura 49)



**Figura 49 - Posicionamiento de la ATID relativo al diagrama de la Fig.20**

- 4- Respecto de la tecnología GMF usada para el desarrollo de los editores gráficos como el editor de mapas estratégicos se concluye que si bien en el mejor de los casos permite acelerar los tiempos de desarrollo, su mecanismo

de generación de código presenta graves falencias. Por un lado, el código generado se mezcla con el personalizado sin poseer mecanismos de trazabilidad avanzados que permitan mantener el sincronismo entre los modelos transformados, corriéndose el riesgo de perder cambios con cada generación de código. Por otro, lado si bien la generación de editores del tipo grafos conectados por líneas resulta bastante simple, el problema surge al requerir personalizaciones más avanzadas en los layouts, como es el caso del editor de mapas estratégicos creado. Para estos casos, se requiere la edición manual del código resultado lo que termina siendo complejo. Estas dificultades encontradas son las que están haciendo aparecer nuevos proyectos como Graphiti, que intentan simplificar estas complejidades.

- 5- En base a las últimas tendencias en el área de las arquitecturas de software se concluye que eclipse necesita evolucionar para mantenerse como una plataforma atractiva para el desarrollo de aplicaciones. Esto hace que construir una aplicación basada en eclipse hoy en día deba hacerse con mucho cuidado y con una arquitectura bien fundada debido al momento de transición que está experimentando la plataforma. Esta es otra de las guías que intenta aportar esta tesina
- 6- En relación a la plataforma Jazz, se encontraron aspectos desalentadores no esperados en la implementación, los cuales se dejaron documentados pues representan problemas serios de usabilidad y calidad de software. Por ejemplo, llama la atención encontrarse en situaciones donde luego de haber hecho una gran cantidad de cambios en un editor se recibe un mensaje de que no es posible guardarlos porque no se tienen los permisos o el rol requerido. También el manejo del repositorio es complejo y lleno de restricciones. Los metamodelos con los que permite trabajar son muy simples, forzando las superclases permitidas para los objetos, suprimiendo la posibilidad de utilización de jerarquías de herencia, o no dejando referenciar objetos en otros componentes, por lo que se hace muy difícil lograr la extensibilidad requerida y el trabajo con modelos complejos como los utilizados para los editores de GMF. Además, el agregado de nuevos modelos al repositorio requieren un proceso de actualización de la BD bastante largo, tedioso y con muy poco soporte. A estas y otras conclusiones se llegaron gracias a los esfuerzos de implementación dado que estas restricciones están escasamente

documentadas. En la arquitectura ATID se incorporan solo los aspectos ventajosos de la plataforma Jazz, sugiriendo el uso de otras tecnologías como por ejemplo la de reciente aparición denominada CDO 3 para el caso específico del repositorio.

- 7- Por último se concluye que la ATID puede verse por un lado como una solución integrada dinámica basada en las tecnologías de eclipse y que permitiría ofrecer tanto a las pequeñas y medianas como a las grandes empresas, de una arquitectura que pueda adaptarse y crecer al ritmo de las necesidades de la organización, pero por otro lado puede verse como una arquitectura de arquitecturas o meta-arquitectura permitiendo armar distintas configuraciones e instancias según lo requerido de modo de resolver no solo las necesidades de los usuarios de esta área, sino de otras ramas con dinamismos similares.

## **6.2 Aportes**

Los objetivos alcanzados que se presentan en esta sección se relacionan con los desafíos y problemas que se presentaron en el área de CPM descripta. Se buscó con esta tesina plantear una solución desde el punto de vista informático mostrando cómo es posible utilizar las tecnologías más recientes basadas en la plataforma eclipse para desarrollar una arquitectura que pueda ser implementada en distintos contextos de organizaciones o empresas y que ayude a los usuarios en la aplicación de las técnicas del área. De esta forma se intenta dar un posible respuesta a las preguntas de cómo llenar el GAP de herramientas especializadas que existe hoy en día en esta área, demostrando cómo es posible soportar el planeamiento estratégico con una arquitectura que satisface todas las necesidades y tendencias del área. Se enumeraran a continuación los aportes principales del trabajo.

1. Se clarifico el área de dominio denominada Gestión de Rendimiento de Negocios (CPM) exponiendo su dinámica y estableciendo un marco formal para la misma, detallando además su relación y diferencia con otras áreas como la Inteligencia de negocios. Además se dejó documentado el estado del arte de los últimos desarrollos relacionados con la metodología de Balanced Scorecard exponiendo algunas inconsistencias entre las variantes de aplicación.

2. Otro aporte está relacionado con el análisis de las principales herramientas que ofrecen soluciones para el área, pudiendo encontrar una clasificación en base a las propiedades comunes y estableciendo claramente en una figura las faltantes respecto de las necesidades para el área. Esto permitió el planteo de la ATID como solución para cubrir estos GAPS.
3. Se realizó una detallada explicación de la evolución de la plataforma eclipse desde sus orígenes y fundamentos hasta los más recientes desafíos y tendencias en la evolución de la plataforma. En el mismo contexto se incluyó a la plataforma Jazz y se la relaciono con el futuro de las arquitecturas modernas basadas en servicios. Por otro lado se descubrió que el concepto tradicional de suite de herramientas basadas en eclipse no son adecuadas para la dinámica del área, presentando una arquitectura más robusta que tolera otros tipos de paradigmas como acceso por navegadores web, etc.
4. Se detalló una arquitectura denominada ATID que no solo permite la implementación del área de CPM sino que además contempla el dinamismo y la versatilidad para trabajar con variaciones o nuevas ideas sin requerir la reconstrucción total de la herramienta. De esta forma se planteó un esquema general que permitiría operar dinámicamente con distintos lenguajes de dominio (DSL). Para el desarrollo de la ATID se estudiaron las arquitecturas más prometedoras basadas en eclipse y sus gaps, aportando una serie de conceptos relacionados con dinamismo y extensibilidad que hace más valiosa a la ATID para la implementación de soluciones en áreas tan incipientes como CPM. Se describieron las principales componentes genéricas necesarias para el rápido desarrollo de prototipos. Por ejemplo, reusando el framweork de aplicaciones y los componentes de navegación, repositorio, búsquedas, y editor gráficos se podría construir rápidamente un editor especializado en el almacenamiento y visualización de métricas claves de rendimiento (KPI) incluyendo las posibilidades de visualización y edición grafica de las posibles operaciones matemáticas entre ellas. Se implementaron los componentes principales presentando las carencias actuales y soluciones posibles. Además, se validó el uso de la arquitectura ejemplificando su utilización para el método “Nine steps to success” del Balanced Scorecard Institute.

5. Se encontró y menciono brevemente cierta relación entre las metodologías usadas en áreas de calidad de software como CMMI y algunos intentos por establecer niveles de madurez respecto de CPM (SMMM)

### **6.3 Trabajos Futuros**

Se describen en esta sección posibles trabajos y líneas de investigación futura relacionados con el área de estudio, con las carencias encontradas en las tecnologías analizadas, y con las posibles extensiones a la arquitectura presentada.

1. Respecto de las metodologías para el área de CPM se podría generar un documento que sirva como base para estandarizar los conceptos del área. Esto implicaría desarrollar el marco conceptual definiendo su área de aplicación, alcance, etc. Por ejemplo, para realizar ello, habría que analizar si el enfoque de formalización sería del estilo de los documentos de la OMG (UML, BPMN) es decir definiendo la sintaxis y semántica del área, o si en lugar sería más beneficioso realizarlo como algo del estilo de CMMI, definiendo niveles de madurez y áreas de proceso. Relacionado con esto también se podrían formalizar los métodos con herramientas como Rational Method Composer (RMC) o el Eclipse Process Framework (EPF) analizando ventajas y desventajas para la integración con la ATID. Además, dentro del marco teórico se podría también investigar la relación entre CPM y otras metodologías y modelos (como por ejemplo SOMA).
2. Se podría analizar la semántica de las relaciones causa-efecto entre objetivos estratégicos de manera de determinar si existe la posibilidad de plantear un formalismo (como por ejemplo su relación con la lógica de predicados) de modo de poder realizar análisis, validaciones o teoremas basados en esta formalización, que resuelvan problemas específicos del área.
3. Crear un framework genérico (componente de la ATID o servicio) que permita realizar operaciones de registro de cambios. de modo de poder implementar un playback de los cambios hechos, similarmente a lo que planteo Google con su propuesta Wave.
4. Los frameworks de editores gráficos como GMF o GEF, son muy necesarios en otros ámbitos como por ejemplo para el desarrollo de editores que se ejecuten dentro de los navegadores web. Sería importante estudiar mecanismos quizás basados en MDD para la generación automática de

código (por ejemplo basado en DOJO, de manera de poder generar los editores gráficos de modo que sean accesibles desde el navegador web)

5. Así como GMF cuenta con 3 modelos principales, sería posible generar nuevos lenguajes tanto para la especificación de layouts como para las políticas de edición, de manera de poder generar editores más específicos mediante la generación de código. Las restricciones OCL también podrían ingresarse durante la etapa de generación.
6. Definida en este documento la arquitectura ATID y sus componentes principales se plantea la posibilidad como trabajo futuro de trabajar en los generadores de código y transformaciones necesarias para derivar instancias particulares de la misma rápidamente para cualquier caso. De esta forma se podría además formalizar la metodología o proceso de desarrollo necesario para construir una extensión. En este proceso es donde se deberían tener en cuenta aspectos de calidad del software mostrando como esos conceptos influyen o no en la metodología, la arquitectura y la herramienta propuesta.
6. Si bien la real utilidad de estos modelos como el Balanced Scorecard es muy difícil de demostrar, dado la complejidad del mundo de negocios en el que es aplicado, se podría explorar la posibilidad de testear el método aplicando técnicas científicas, como por ejemplo la generación de micromundos, de manera de obtener datos de aplicación de la herramienta sobre ejemplos concretos aportando datos científicos válidos. También se encontró la existencia de estándares que podrían guardar relación con el área de CPM como el Business Motivation Model (BMM) o el de reglas de negocios (SVBR) Se deja como trabajo futuro el análisis de estos estándares y documentar como podrían influir sus conceptos en la ATID.
7. En base al estudio teórico se concluyó que una de las necesidades fundamentales de los consultores pasan por proveerles capacidades graficas de desarrollo de modelos (Por ejemplo, editores de mapas estratégicos). Por ello se describió una arquitectura que permite de una manera extensible agregar nuevos editores basados en GMF a la plataforma. Respecto de la solución encontrada, existen algunas alternativas como posibles campos de estudio futuro. Por ejemplo, en la configuración actual de la arquitectura se utilizó un solo servidor. La desventaja es que el agregado de nuevas entidades implica cambiar el esquema de la base de datos, tarea que quizás requiera

detener la BD actual, modificarla y volver a arrancar el servidor. Es posible extender la arquitectura para contemplar información distribuida en múltiples servidores. CALM es la tecnología de jazz que permite realizar estas actividades y cuya relación con la ATID podría ser investigada. Además, las posibilidades de interoperabilidad entre herramientas abren la posibilidad de proponer nuevos tipos de análisis en base a las ventajas de integración con otros sistemas y metodologías. Futuras líneas de investigación podrían contemplar estas posibles extensiones.



## Referencias

- [1] Jürgen Strohhecker, Does a Balanced Scorecard Management Cockpit Increase Strategy Implementation Performance - Findings of a series of experiments using a system dynamics based micro-world - Frankfurt School of Finance and Management - July 2007
- [2] Zbigniew Michalewicz et al, “Adaptive Business Intelligence” - Springer ISBN-13-978-3-540-32928-2 – 2007
- [3] Joerg Reinschmidt, Allison Francoise - Business Intelligence Certification Guide - International Technical Support Organization - January 2000
- [4] Robert B. Walford - Business Process Implementation for IT Professionals - Artech House © 1999
- [5] Praveen Gupta -Six Sigma Business Scorecard - Ensuring Performance for Profit - McGraw-Hill – 2004
- [6] Mark C. Tibergien and Rebecca Pomeroy - Practice Made Perfect - The Discipline of Business Management for Financial Advisers — Bloomberg Press - Princeton – 2005
- [7] Michael S. Zambruski - A Standard for Enterprise Project Management - ESI international project management series – 2009
- [8] H. P. Luhn - A Business Intelligence System — IBM Journal 1958
- [9] Chuck Ballard et al, “Business Performance Management. Meets Business Intelligence” - IBM Redbook - July 2005
- [10] Chuck Ballard et al - Dimensional Modeling: In a Business Intelligence Environment - IBM Redbook - March 2006
- [11] Jürgen Strohhecker - Does a Balanced Scorecard Management Cockpit Increase Strategy Implementation Performance - Findings of a series of experiments using a system dynamics based micro-world Frankfurt School of Finance and Management - July 2007
- [12] Larissa T. Moss, Shaku Atre - Business Intelligence Roadmap: The Complete Project Lifecycle for Decision-Support - Addison Wesley - February 2003
- [13] Bruce Walters - IT Enabled Strategic Management: Increasing Returns for the Organization - Louisiana Tech University – 2006
- [14] Philip Sadler - Strategic Management - Cogan Page – 2003

- [15] Paul R. Niven - Balanced Scorecard Step-by-Step - Maximizing Performance and Maintaining Results - John Wiley & Sons, Inc. – 2002
- [16] Robert Kaplan - David Norton – “The Balanced Scorecard - Translating Strategy into Action” - Harvard Business School – 1996
- [17] Kevin Laframboise - Business Performance and Enterprise Resource Planning - Concordia University, Montreal, Canada ECIS 2002
- [18] Dan Volitich - IBM Cognos 8 Business Intelligence: The Official Guide - McGraw-Hill – May 2008
- [19] Chuck Ballard et al - Improving Business Performance Insight with Business Intelligence and Business Process Management - IBM Redbook – August 2006
- [20] Craig Utley - Business Intelligence with Microsoft® Office PerformancePoint Server 2007 - McGraw-Hill Osborne - January 2008
- [21] Doug Harts - Microsoft ® Office 2007 Business Intelligence - McGraw-Hill Osborne - November 2007
- [22] Mohan Nair - Essentials of Balanced Scorecard - John Wiley & Sons – 2004
- [23] Nils Rasmussen - Financial Business Intelligence - Trends, Technology, Software Selection, and Implementation - John Wiley & Sons – 2002
- [24] Gary Chin - Agile Project Management: How to Succeed in the Face of Changing Project Requirements - Amacom © 2004
- [25] Mark Marone and Seleste Lunsford - Strategies That Win Sales: Best Practices of the World’s Leading Organizations - Dearborn Financial © 2005
- [26] Robert Kaplan and David Norton - The Balanced Scorecard Measures that Drive Performance – Harvard Business Review – February 1992
- [27] Harold Kerzner - Strategic Planning For Project Management Using a Project Management Maturity Model - John Wiley & Sons Inc – 2001
- [28] Dick Smith and Jerry Blakeslee - Strategic Six Sigma Best Practices From Executive Suite - John Wiley & Sons, INC – 2002
- [29] Michel Crouhy, Robert Mark - The Essentials of Risk Management - McGraw-Hill – 2006
- [30] Dick Grote - The Performance Appraisal Question and Answer Book - Amacom – 2002
- [31] R.A.Hill - Strategic Financial Management – Ventus Publishing - 2008
- [32] Carlo Verzellis, “Business Intelligence: Data Mining and Optimization for Decision Making”- John Wiley & Sons, Ltd.—2009, ISBN: 978-0-470-51138-1

- [33] Mamdouh Refaat ,“Data preparation for data minig using SAS” -- Morgan Kaufmann Publishers – 2007, ISBN 13: 978-0-12-373577-5
- [34] Mark N. Frolick and Thilini R. Ariyachandra, “Business Performance Management: One Truth”, Information Systems Management, 2006
- [35] ¿Que es, en definitiva, el balanced scorecard? Symnetics Business Transformation – PDF -- [www.symnetics.com.ar](http://www.symnetics.com.ar)
- [36] Robert Kaplan - David Norton – “The Strategy-Focused Organization: How Balanced Scorecard Companies Thrive in the New Business Environment” Harvard Business School – 2001
- [37] Robert Kaplan - David Norton – “Strategy Maps: Converting Intangible Assets into Tangible Outcomes” - Harvard Business School – 2004
- [38] Robert Kaplan - David Norton – “Alignment: Using the Balanced Scorecard to Create Corporate Synergies” - Harvard Business School – 2006
- [39] Robert Kaplan - David Norton – “The Execution Premium: Linking Strategy to Operations for Competitive Advantage” - Harvard Business School – 2008
- [43] Paul R. Niven “Balanced Scorecard Step by Step – Maximizing Performance and Maintaining Results”, John Wiley & Sons, Inc. ISBN 0-471-07872-7., 2002
- [44] IBM Cognos 8 Business Intelligence-Version 8.4.1 - Architecture and Deployment Guide - 2009
- [45] Marco Sisfontes-Monge - CPM and Balanced Scorecard with SAP - Galileo Press – 2006

## Referencias web

[W1] Inteligencia de Negocios (BI):

<http://www.monografias.com/trabajos14/bi/bi.shtml>

[W2] Teoría de Data Mining

<http://churriwifi.wordpress.com/2010/02/26/teoria-de-data-mining>

[W3] El tablero de comando (Balanced Scorecard)

<http://www.3w3search.com/Edu/Merc/Es/GMerc094.htm>

[W4] Oracle Strategy Management Solutions

<http://www.oracle.com/us/solutions/ent-performance-bi/051192.html>

[W5] The Challenge of Strategic Alignment

[http://www.fsn.co.uk/channel\\_kpi\\_environment/the\\_challenge\\_of\\_strategic\\_alignment](http://www.fsn.co.uk/channel_kpi_environment/the_challenge_of_strategic_alignment)

[W6] Product Review SAS SPM Strategic Performance Management

[http://www.fsn.co.uk/channel\\_bi\\_bpm\\_cpm/pr\\_sas\\_spm\\_strategic\\_performance\\_management](http://www.fsn.co.uk/channel_bi_bpm_cpm/pr_sas_spm_strategic_performance_management)

[W7] Documentation for Strategic Performance Management

<http://support.sas.com/documentation/onlinedoc/spm/>

[W8] SAS Strategy Management

<http://www.sas.com/solutions/spm/>

[W9] Spider Strategies

<http://www.spiderstrategies.com/>

[W10] Watershed Strategy Execution Software

<http://handscribe.com/watershed>

[W11] Overview of Graphiti

<http://www.eclipse.org/graphiti/documentation/overview.php>

[W12] GEF goes 3D

<http://jaxenter.com/gef-goes-3d-10076.html>

[W13] GEF3D development blog

<http://dev.eclipse.org/blogs/gef3d/>

[W14] RAP Rich Ajax Platform

<http://www.eclipse.org/rap/>

[W15] Innovate2010 jazz keynote

<http://www.slideshare.net/oslc/innovate2010-jazz-keynote-4528964>

- [W16] Balanced Scorecard Institute  
<http://www.balancedscorecard.org/>
- [W17] Incorporating Strategic Themes into Your Next Strategic Plan  
<http://www.fastmeetings.com.au/articles/strategic-themes.htm>
- [W18] CPM - Corporate Performance Management  
[http://www.ibermatica.com/ibermatica/cpm\\_cognos](http://www.ibermatica.com/ibermatica/cpm_cognos)
- [W19] A brief history of the Jazz Team Server interface: Our journey from a J2EE server towards a RESTful server  
<http://jazz.net/blog/index.php/2008/02/15/a-brief-history-of-the-jazz-server-interface-our-journey-from-a-j2ee-server-towards-a-restful-server/>
- [W20] Jazz Integration Architecture Overview  
<http://jazz.net/projects/DevelopmentItem.jsp?href=content/project/plans/jia-overview/index.html>
- [W21] Deploying HelloJfs into Jazz Foundation Server using Jazz Provisioning  
<https://jazz.net/wiki/bin/view/Main/ProvisionAppInJFS>
- [W22] Hello Jazz -- How to write a simple Jazz component  
<http://jazz.net/library/article/118>
- [W23] Jazz Platform Technical Overview  
<http://jazz.net/library/LearnItem.jsp?href=content/docs/platform-overview/index.html>
- [W24] Contributing to the Team Artifacts view in Rational Team Concert 2.0]  
<http://jazz.net/library/article/190/>
- [W25] The JazzBot component development walkthrough  
<https://jazz.net/wiki/bin/view/Main/JazzBotWalkthrough>
- [W26] Component Development Home  
<https://jazz.net/wiki/bin/view/Main/ComponentDevelopment>
- [W27] JazzTalk component development walkthrough  
<https://jazz.net/wiki/bin/view/Main/JazzTalkWalkthrough>
- [W28] Building a Jazz Repository database  
<https://jazz.net/wiki/bin/view/Main/HowToBuildRepositoryDatabase>
- [W29] Process awareness and customizability  
<https://jazz.net/projects/rational-team-concert/features/process-aware>
- [W30] Getting Started with Rational Team Concert: A Deployment Guide  
<http://jazz.net/library/article/398>
- [W31] Work Item Templates

<http://jazz.net/wiki/bin/view/Main/WorkItemTemplates>

[W32] Agile practices enactment with Rational Team Concert and Rational Method Composer

<http://jazz.net/library/article/410>

[W33] Using & Extending the Notifier for Events

<https://jazz.net/wiki/bin/view/Main/FoundationNotifierTutorial>

[W34] Writing your own reports in Rational Team Concert 2.0

<http://jazz.net/blog/index.php/2009/07/24/writing-your-own-reports-in-rational-team-concert-20/>

[W35] Extending the Data Warehouse

<https://jazz.net/wiki/bin/view/Main/DataWarehouseExtending>

[W36] Teach Your Eclipse to Speak the Local Lingo

<http://www.eclipse.org/articles/Article-Speak-The-Local-Language/article.html>

[W37] How to Internationalize your Eclipse Plug-In

<http://www.eclipse.org/articles/Article-Internationalization/how2I18n.html>

[W38] JDojo

<https://jazz.net/wiki/bin/view/Main/JDojo>

[W39] Database vendor is not supported by the server license

<https://jazz.net/wiki/bin/view/Main/DbVendorIsNotSupported>

[W40] Open Social Design

<https://jazz.net/wiki/bin/view/Main/OpenSocialDesign>

[W41] Balanced Scorecard

[http://estudiantesadministracionuss.blogspot.com/2009\\_10\\_01\\_archive.html](http://estudiantesadministracionuss.blogspot.com/2009_10_01_archive.html)

[W42] Jazz Foundation Application SDK

<https://jazz.net/wiki/bin/view/Main/ServerSDKMain>

[W43] Metodología BSC

<http://www.monografias.com/trabajos17/metodologia-bsc/metodologia-bsc.shtml>

[W44] Executive Strategy Manager

<http://www.executivestrategymanager.com/>

[W45] Running and Debugging a Jazz server for component development

<https://jazz.net/wiki/bin/view/Main/JazzServerRunAndDebug>

[W46] Balanced Scorecard

<http://www.5campus.com/LECCION/bsc/INICIO.HTML>

[W47] GMF Tutorial

[http://wiki.eclipse.org/index.php/GMF\\_Tutorial](http://wiki.eclipse.org/index.php/GMF_Tutorial)

[W48] Epsilon

<http://www.eclipse.org/gmt/epsilon/cinema/#Eugenia>

[W49] GMF Tutorial and Guide

<http://www.docstoc.com/docs/2165225/GMF-Tutorial-and-Guide>

[W50] Implementación de un Balanced Scorecard en corporación editorial

<http://www.scribd.com/doc/3016165/Implementacion-de-un-Balanced-Scorecard>

[W51] GMF Model Modifications

[http://support.phitechnology.it/wiki/index.php/GMF\\_Model\\_Modifications](http://support.phitechnology.it/wiki/index.php/GMF_Model_Modifications)

[W52] El Balanced Scorecard y la Gerencia Estrategica

<http://www.slideshare.net/alfonsogu/cmi-2642431>

[W53] Balanced Scorecard

<http://www.slideshare.net/JAARIASV1/balanced-scorecard-4069908>

[W54] Dev Setup

<https://jazz.net/wiki/bin/view/Main/DevSetupMain>

[W55] RTCSDK20\_DevelopmentEnvironmentSetup

[https://jazz.net/wiki/bin/view/Main/RTCSDK20\\_DevelopmentEnvironmentSetup](https://jazz.net/wiki/bin/view/Main/RTCSDK20_DevelopmentEnvironmentSetup)

[W56] Hello Jazz - How to write a simple Jazz component

<http://jazz.net/library/article/118>

[W57] CDO Hibernate Store

[http://wiki.eclipse.org/CDO\\_Hibernate\\_Store](http://wiki.eclipse.org/CDO_Hibernate_Store)

## Glosario

En esta sección se describen brevemente el significado de las abreviaciones utilizadas. Las explicaciones detalladas se encuentran en el desarrollo de la tesina.

**ATID – Arquitectura Tecnológica Integrada Dinámica.** Arquitectura presentada en la sección 4 de esta tesina como solución a las necesidades del área.

**BSC – Indicadores Balanceados de Desempeño – (Balanced Scorecard).** Técnica para la gestión del rendimiento empresarial presentada en la sección 2.3

**BI – Inteligencia de Negocios (Business Intelligence).** Conjunto de técnicas de soporte para la toma de decisiones detallada en la sección 2.1

**BIRT (Business Intelligence and Reporting Tools).** Extensiones de la plataforma eclipse para la generación de reportes.

**BMM – Business Motivation Model.** Estándar definido por la OMG para la generación de planes de negocios.

**BPM – Gestión de Rendimiento de Negocios (Business Performance Management).** Area de gestión descrita en la sección 1.1.

**BPMN -- Business Process Model and Notation.** Estándar para la definición de procesos de negocios.

**CDO -- Connected Data Objects.** Extensión de eclipse para soporte de objetos distribuidos detallada en la sección 3.2.4

**CMI – Cuadro de Mando Integral.** Traducción utilizada comúnmente para referenciar al área de Balanced Scorecard.

**CNF – Common Navigator Framework.** Extensión de eclipse para la rápida generación de navegadores con estructura de árbol.

**CPM – Gestión del Rendimiento Corporativo (Corporate Performance Management).** Area de gestión descrita en la sección 2.2

**CRISP -- CRoss-Industry Standard Process for Data Mining.** Técnica utilizada en la Inteligencia de Negocios para la minería de datos.

**CRM -- Customer Relationship Management.** Sistemas de gestión de las relaciones con los clientes.

**Dashboard – Cuadro de Mando - Tablero de Control.** Conjunto gráficos que expresan los valores de los indicadores de rendimientos.

**Data Mining – Minería de datos. .** Técnica utilizada en la Inteligencia de Negocios para extraer información útil de un conjunto grande de datos.



**Data Warehousing – Almacén de datos.** Bases de datos especializadas en el almacenamiento de información históricas optimizadas para consultas..

**DSL – Lenguaje específico de Dominio - Domain Specific Language .** Expresa la sintaxis y semántica de las entidades de una área de dominio en particular

**Eclipse** – Plataforma de integración de software [www.eclipse.org](http://www.eclipse.org)

**EMF – Eclipse Modeling Framework.** Extensión de eclipse para la expresión de modelos en alto nivel (metamodelos) generando código java a partir del mismo.

**EPM – Gestión el Rendimiento Empresarial (Enterprise Performance Management).** Area de gestión descrita en la sección 1.1

**ERP -- Enterprise Resource Planning.** Sistemas de gestión para la planificación de recursos.

**EPF – Eclipse Process Framework. .** Extensiones de eclipse para la expresión de metodologías con capacidades de generación de documentación navegable (sitios web).

**ETL - Extracción, transformación y carga. (Extract, transform, load).** Etapa utilizada en la Inteligencia de Negocios. (Sección 2.2)

**Gartner S.A.-** Empresa especializada en consultoría de Negocios.

**IT – Tecnología de Información.**

**Jazz** – Plataforma de integración colaborativa -- [www.jazz.net](http://www.jazz.net).

**KPI – Indicadores claves de rendimiento (Key Performance Indicators).** Métricas que tienen un valor significativo para la indicación de alguna aspecto relacionado con el rendimiento de un concepto específico.

**MDD – Desarrollo Dirigido por Modelos -- Model Driven Development.** Arrea en la que se formalizan los conceptos que permiten derivar modelos a partir de transformaciones de los mismos.

**OCL (Object Constraint Lenguaje).** Lenguaje formal para la expresión de restricciones lógicas sobre los modelos UML

**OLAP -- Procesamiento analítico en línea (Online analytical processing).** Técnicas utilizadas en la Inteligencia de Negocios.

**OMG – Object Management Group.** Organización que define y regula los estándares como UML, BPMN, BMM, etc

**OSGi -- Open Services Gateway Initiative .** Organización que define un estándar de interoperabilidad entre componentes.

**OSLC -Open Services for Lifecycle Collaboration** . Iniciativa que intenta ofrecer especificaciones estándares de expresión de recursos a través de servicios web.

**RCP – Rich Client Platform.** Tecnologías de eclipse relacionadas con la utilización de la plataforma para la generación de aplicaciones genéricas de escritorio.

**RMC – Rational Method Composer.** Herramienta que permite la expresión de metodologías y su posterior publicación como sitio web navegable.

**RTC – Rational Team Concert** . Herramienta construida sobre la plataforma jazz para coordinación de las actividades de desarrollo de software.

**SEMMA -- Sample, Explore, Modify, Model, Asses.** (<http://www.sas.com/>). Técnica utilizada en la inteligencia de negocios.

**SMMM - The Strategic Management Maturity Model.** Intento de definición de un modelo de madurez para el área de Balanced Scorecard.

**SOA -- Service Oriented Architecture.** Arquitecturas orientadas a servicios.

**Strategic Map – Mapa Estratégico.** Mapa construido durante la aplicación de la técnica de Balanced Scorecard detallado en la sección 2.3.3

**UML (Unified Modeling Lenguaje)** . Lenguaje de expresión de modelos unificado. Estándar definido por la OMG para la expresión de modelos de manera abstracta.

**Value Driver** -- Elementos que favorecen la creación de valor.

**XMI - XML Metadata Interchange.** Define un formato XML para el intercambio de datos y metadatos entre distintas herramientas.