



Proyecto de Grado

Sistema de Gestión del centro de cómputos de IBM

Daniela Moyano, Marcelo Sureda

Tutor: Ing. Gustavo Vázquez

Usuario Responsable: Ing. Diego López Roig

Junio 2005

Instituto de Computación
Facultad de Ingeniería
Universidad de la República Oriental del Uruguay

INGENIERIA





Resumen

En el presente trabajo se aborda la evaluación de las herramientas de administración utilizadas por las distintas áreas del centro de cómputos de IBM (administración de sistemas, bases de datos, redes y operaciones) y la construcción de una herramienta que permite mejorar el seguimiento de la gestión del mismo por medio de la evaluación de indicadores de productividad adecuadamente seleccionados.

El trabajo se dividió en tres etapas para el mejor desarrollo y seguimiento del proyecto. En la primer etapa se analizó el estado del centro de cómputos. Con este fin se realizó un relevamiento de las funcionalidades del sistema actual de gestión, los requerimientos de mejora planteados por el personal, la revisión de procesos y la modalidad de trabajo.

En la segunda etapa se realizó una investigación de indicadores de productividad utilizados por otros centros de cómputos en el mercado, evaluando y seleccionando los que mejor se adecuaran para el proyecto. Se presentó una propuesta con dichos indicadores y en conjunto con IBM se acordaron los indicadores a ser utilizados. Como resultado de la investigación y del análisis realizado quedó establecida la información que el nuevo sistema debe registrar para permitir la estimación de dichos indicadores.

En la tercer etapa se construyó un prototipo de la nueva herramienta de administración del centro de cómputos. El prototipo se ha desarrollado utilizando tecnología de última generación, lo que redundará en una actualización respecto a la herramienta empleada actualmente con este fin. Fue implementado íntegramente en la herramienta *open source* Eclipse utilizando lenguaje Java, servidor de aplicaciones Tomcat y base de datos DB2. Estas elecciones se basaron en un estudio realizado previamente sobre los diversos entornos, tecnologías y lenguajes más conocidos y utilizados en el mercado, los cuales por su parte deben cumplir con los estándares de IBM. En esta etapa también se definió el plan de releases para que en el futuro IBM pueda ir incorporando el resto de las funcionalidades al prototipo construido.

La herramienta que resulte de la implementación de los siguientes releases del prototipo será el pilar fundamental en lo que respecta al seguimiento de las tareas del centro de cómputos. Definirá los distintos flujos de trabajo y facilitará la ejecución de los diferentes procesos operativos además de generar y mantener un registro de todas las solicitudes que otras áreas de la compañía o clientes realizan al centro de cómputos.

Palabras Clave: procesos, ejecuciones, indicadores, métricas, productividad, bitácora, centro de cómputos, herramienta de administración, métricas de productividad, prototipo, operaciones.





Índice

1.	Introducción.....	7
1.1	Contexto del proyecto.....	7
1.2	Objetivos.....	7
1.3	Resultados esperados.....	7
1.4	Plan de trabajo.....	8
1.5	Metodología de trabajo.....	8
1.6	Organización del informe.....	9
2.	Interiorización en la gestión del SDC.....	10
2.1	Organización del SDC.....	10
2.2	Conceptos relevantes.....	11
2.3	Modalidad de trabajo y sistema utilizado.....	11
2.4	Problemas detectados.....	12
3.	Selección de métricas de productividad.....	13
4.	Solución al problema.....	15
4.1	Selección de lenguaje y entorno de desarrollo.....	15
4.2	Arquitectura.....	16
4.3	Diseño.....	17
4.4	Implementación.....	23
4.5	Alcance del prototipo.....	29
5.	Conclusiones y trabajo futuro.....	30
6.	Referencias.....	32
6.1	Referencias de Internet.....	32
6.2	Referencias bibliográficas.....	34
	Agradecimientos.....	35



Lista de anexos

Anexo A - Cronogramas

Anexo B - Modalidad de trabajo

Anexo C - Situación actual

Anexo D - Investigación de métricas

Anexo E - Investigación de entornos de desarrollo

Anexo F - Arquitectura del sistema

Anexo G - Diseño del sistema

Anexo H - Plan de releases

Anexo I - Manual de instalación

Anexo J - Manual de usuario

Anexo K - Glosario



1. Introducción.

1.1 Contexto del proyecto

El proyecto se realizó para la empresa IBM, específicamente para el área de “Delivery” de proyectos asociada a su centro de cómputos (SDC). Dicho sector está compuesto por un grupo de personas entrenadas en distintos perfiles de trabajo que en conjunto proveen servicio a múltiples empresas ya sean de pequeño, mediano o gran porte. Su principal objetivo es la administración y operación de la infraestructura informática necesaria para cumplir con los niveles de servicios acordados con los clientes.

Hoy en día IBM cuenta con una herramienta donde se registran las ejecuciones de los procesos operativos realizadas por los operadores. Esta herramienta permite obtener indicadores de cumplimiento de los niveles de servicio comprometidos. Desde hace algún tiempo IBM tiene la intención de aprovechar la información que se registra en dicha herramienta, para obtener indicadores que le permitan medir la productividad de su SDC y compararlas con los estándares del mercado. Dado que actualmente sólo se registran los procesos ejecutados por los operadores, para obtener indicadores de productividad de todo el SDC era necesario realizar modificaciones en la herramienta para permitir el registro de las tareas realizadas por los especialistas.

A partir de la necesidad planteada y la falta de recursos apropiados para llevar adelante este trabajo, surgió la idea de proponerlo como proyecto de grado a la Facultad de Ingeniería.

1.2 Objetivos

El proyecto presentado en este documento tiene por finalidad la elaboración de una herramienta de gestión que centralice la información de las distintas áreas permitiendo mejorar el seguimiento de las tareas desarrolladas en el sector.

Dicha herramienta deberá permitir un seguimiento de las tareas del SDC mediante la definición de distintos flujos de trabajo. A través de ella se busca mejorar la productividad del SDC, motivando la investigación y elaboración de indicadores de productividad adecuados que permitan realizar una estimación más exacta de la misma.

1.3 Resultados esperados

Los resultados esperados del proyecto son:

- Estudio de los procesos relacionados con la gestión del SDC
- Evaluación de las herramientas de administración utilizadas y a utilizarse
- Propuestas de forma de integración de todas las herramientas
- Definición de indicadores de productividad
- Diseño y desarrollo de la herramienta de gestión
- Adecuación de los procesos existentes



1.4 Plan de trabajo

Se definió un plan de trabajo a seguir con las etapas que se enumeran a continuación, cada una de las cuales son detalladas más adelante. En 'Anexo A – Cronogramas' se presenta el cronograma que se estableció inicialmente junto con su posterior redefinición.

- Interiorización en la gestión del SDC: se realizó un relevamiento de la situación actual, las herramientas utilizadas y a utilizar a corto plazo, además de un relevamiento de la modalidad de trabajo. Dicho relevamiento fue motivado en la adquisición de una visión global de la situación que permitiera evaluar las carencias del SDC en lo referente a su gestión. El resultado de esta etapa fue el análisis de requerimientos.
- Definición de métricas: se investigó acerca de las métricas utilizadas en otros centros de cómputos del mercado concernientes al área de Operaciones. En base a dicha investigación y al estudio realizado en la etapa anterior se realizó una propuesta de indicadores a utilizarse, orientados a mejorar el seguimiento de la gestión y la productividad del SDC. Esta propuesta fue presentada a IBM y junto a ellos se seleccionó el subconjunto de los indicadores más relevantes a los intereses de la empresa.
- Estudio de diferentes entornos de desarrollo: se relevaron los entornos y lenguajes más utilizados en el mercado, realizando una evaluación de los mismos para luego definir el entorno y lenguaje que mejor se adaptaran en todos los aspectos para el desarrollo del nuevo sistema.
- Desarrollo del prototipo: se llevó adelante en varias etapas, primero se definió el alcance del prototipo. Luego se formalizó el análisis en base a la información recavada en las etapas anteriores. Posteriormente se procedió a la definición del diseño del sistema, realizando la selección del entorno y lenguaje a utilizarse, definiendo la arquitectura e interface del prototipo. Por parte de IBM se realizó una validación de los requerimientos, diseño y arquitectura del nuevo sistema para luego comenzar con la fase de implementación seguida por la fase de testeo y elaboración de manual de usuario. La fase final de esta etapa estuvo comprendida por el testeo de aceptación por parte de IBM.
- Validación de las métricas utilizando el prototipo: se procedió al uso de la nueva herramienta por parte de IBM, de esta manera se cargaron datos reales por un período de dos semanas. Luego de dicho período y a partir de la información registrada fue posible validar las métricas por medio del prototipo, verificando que los resultados obtenidos a partir de los indicadores definidos en la herramienta se aproximaron a la percepción del sector durante este plazo.

1.5 Metodología de trabajo

Se siguió el plan de trabajo definido respetando cada una de sus etapas. Se fue avanzando de forma que el trabajo de cada etapa se basó en los resultados de la etapa anterior, documentando los mismos en entregables de utilidad como base para el informe final. También se escribieron como informes intermedios para la gente de IBM. Además de los documentos se tenían reuniones quincenales con el encargado del proyecto por parte de IBM donde se discutían los resultados obtenidos.

Durante la etapa de relevamiento de información concerniente a la modalidad de trabajo y herramientas utilizadas, se realizaron reuniones con integrantes del SDC cubriendo los distintos perfiles.

IBM siempre estuvo disponible para evacuar dudas, ya fueran de tipo funcionales o técnicas. También se nos brindó un curso básico de DB2 ya que la utilización de este producto fue un requisito por parte de IBM.

Por parte de Facultad desde el inicio del proyecto se mantuvieron reuniones quincenales con el tutor del proyecto. Dichas reuniones se motivaron en el seguimiento del proyecto y consulta de dudas.



1.6 Organización del informe

A continuación se presentará la organización de este documento de forma de facilitar la lectura del mismo.

En este capítulo se describió la motivación y contexto del proyecto, los objetivos y resultados esperados, así como el plan de trabajo y metodología empleados. Como complemento del plan de trabajo, en '*Anexo A – Cronogramas*' se presenta el cronograma que se estableció inicialmente junto con su posterior redefinición.

En el segundo capítulo se describe la organización del SDC y su modalidad de trabajo junto a los conceptos claves permitiendo una mejor comprensión. También se detallan las herramientas de administración utilizadas junto a las mejoras a incorporar en el nuevo sistema. Los documentos '*Anexo B - Modalidad de trabajo*' y '*Anexo C - Situación actual*' presentan en mayor detalle el relevamiento realizado de la modalidad de trabajo y las herramientas de administración utilizadas en el SDC respectivamente.

En el tercer capítulo se presenta una propuesta de indicadores fundamentada en un estudio realizado sobre las métricas utilizadas en el mercado. Estos indicadores se encuentran orientados a mejorar el seguimiento de la gestión y la productividad del SDC. En '*Anexo D - Investigación de métricas*' se expone el estado del arte de las métricas de productividad.

El cuarto capítulo expone una descripción de los distintos aspectos de la solución al problema. Se presentan las decisiones adoptadas respecto a la elección del lenguaje y entorno de desarrollo utilizado. Dichas decisiones se encuentran fundamentadas en una investigación sobre el tema que es presentada en '*Anexo E - Investigación de entornos de desarrollo*'. Este capítulo también presenta las decisiones más importantes concernientes a la arquitectura y diseño de la herramienta, fundamentadas en los análisis que se presentan en '*Anexos F - Arquitectura del sistema*' y '*Anexo G - Diseño del sistema*' respectivamente. Se presenta además el alcance del prototipo que se elaboró junto con los trabajos a realizar a futuro que continuarán el mismo, '*Anexo H - Plan de releases*' expone el plan de releases definido. Respecto a la herramienta, '*Anexo I - Manual de instalación*' presenta el correspondiente manual de usuario.

En el quinto capítulo se exponen las principales conclusiones obtenidas como resultado del proyecto así como trabajos futuros que se pueden desarrollar a partir del mismo.



2. Interiorización en la gestión del SDC

Este capítulo hace foco en el estado del SDC. Se detalla su organización, los conceptos básicos necesarios para la comprensión del tema de estudio, su modalidad de trabajo y las herramientas utilizadas relevantes a la gestión. El objetivo es tener una visión más amplia que permita identificar las debilidades a nivel operativo y carencias del sistema existente, de forma de considerar el fortalecimiento de estos puntos a la hora del desarrollo del nuevo sistema. Los documentos 'Anexo B - Modalidad de trabajo' y 'Anexo C - Situación actual' presentan con mayor detalle el relevamiento realizado del SDC.

2.1 Organización del SDC

Dentro del SDC el personal cumple distintas funciones cubriendo todas las áreas de especialización. A continuación se muestra el organigrama detallando sus correspondientes responsabilidades.

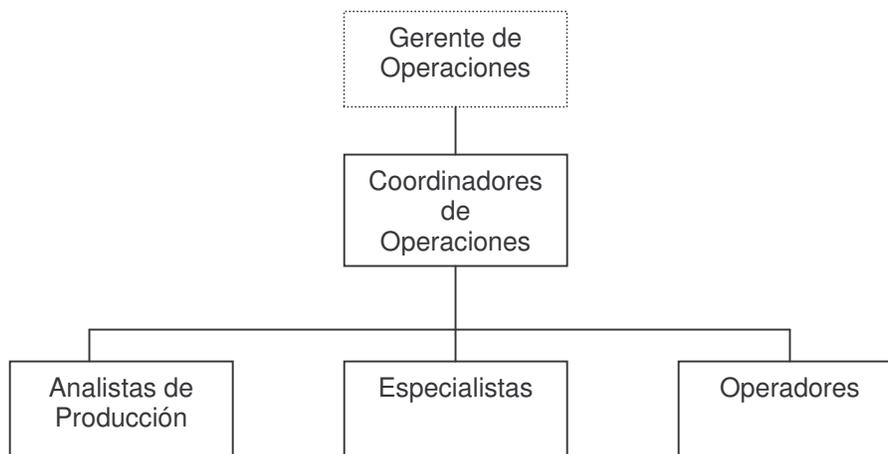


Figura 1 – Organigrama del SDC

El **Gerente de Delivery (de Operaciones)** es el responsable del sector de Operaciones en su totalidad. De él dependen todos los coordinadores.

El **Delivery Project Executive (Coordinador de Operaciones)** es el responsable de gerenciar el proyecto desde el punto de vista operativo, de coordinar el trabajo de los grupos de especialidad a su cargo. También es el responsable del proyecto frente al cliente.

El **Analista de Producción** es el responsable de la planificación de las tareas diarias de los proyectos que tiene asignado, de mantener un ambiente de producción controlado, y de hacer cumplir los estándares definidos para el ambiente de producción.

El **Especialista** posee los conocimientos y experiencia necesaria para satisfacer los requerimientos en un área de trabajo específica. Existen varios grupos de especialistas en función del área de trabajo, dentro de las tareas se destacan la administración y seguridad de las bases de datos, cambios de configuración de los servidores de aplicación, manejo de versiones de las aplicaciones web y administración de la red.

El **Operador** es responsable de ejecutar los procesos de acuerdo a la planificación realizada por el analista de producción.



2.2 Conceptos relevantes

Un proceso es un conjunto de fases sucesivas de acciones para cumplir un objetivo. Cada fase se denomina método y se deben ejecutar en determinado orden de acuerdo al manual de ejecución donde se describe el procedimiento que el operador debe seguir para la ejecución del mismo.

Los procesos se clasifican en cuatro tipos dependiendo de la frecuencia de ejecución y el cometido de los mismos:

Procesos regulares: se ejecutan con cierta frecuencia o a demanda (se sabe cuando ejecutarlo con cierta anticipación), por ejemplo el respaldo de la BD.

Procesos de ciclo: también se ejecutan con cierta frecuencia. Se diferencian de los anteriores por estar compuestos de varios procesos que tienen definido un orden de precedencia, tener un valor asociado (rama) que los categoriza en función del módulo sobre el cual ejecutan, y tener menor frecuencia de ejecución (la mayoría se ejecutan una o dos veces al mes).

Procesos no regulares: se ejecutan a demanda y generalmente se ejecutan una única vez o a lo sumo un par de veces, por ejemplo corregir datos en la base de datos que quedaron mal por algún proceso que no realizó lo que se esperaba.

Procesos de base de datos: se ejecutan directamente en la base de datos por parte de los administradores, por ejemplo cambios en procedimientos almacenados, en vistas o en estructuras de datos.

Procesos de puesta en producción: son las puestas en producción de cambios en las aplicaciones que residen en los servidores del centro de cómputos, por ejemplo cambios en las clases Java de una aplicación o cambios en los binarios de C/C++.

Los distintos métodos tienen un nombre asociado de acuerdo al cometido del mismo, y no todos son obligatorios. Por ejemplo el método "init" concierne a la configuración del ambiente en que se va a ejecutar el proceso, el método "exec" corresponde al que ejecuta el proceso en sí, el método "pos_cond" es el que verifica si la ejecución del proceso fue exitosa, entre otros.

2.3 Modalidad de trabajo y sistema utilizado

IBM para el manejo del SDC utiliza un sistema core en el cual lleva un registro de todos los procesos que se ejecutan en el ambiente de producción. Este sistema se llama Bitácora de Operación y está desarrollado en Forms de Oracle ejecutándose en modo carácter.

Se trata de un sistema de información relevante a la ejecución de los procesos, cuya información es utilizada posteriormente para generar informes internos y reportes a los clientes. La carga de los procesos que se deben ejecutar no la realiza el propio sistema, como complemento se debe realizar con un archivo de configuración que agenda las ejecuciones de los mismos para cada día de la semana.

A continuación se detallan las funcionalidades de la Bitácora:

Registro de ejecución de procesos: se detalla la ejecución de cada uno de los métodos que componen el proceso, registrando su fecha de inicio, fecha de fin, el detalle de lo ejecutado, como finalizó, y en caso de utilizar un medio magnético también se registra su etiqueta.

Consultas: el sistema permite listar las ejecuciones de procesos por varios filtros. Es de utilidad por ejemplo cuando un operador comienza su turno y desea revisar que procesos debe ejecutar y consultar información histórica de ejecuciones para observar si los tiempos y el comportamiento de una ejecución se encuentran dentro de los parámetros normales.

Mantenimiento: corresponde al mantenimiento de codigueras propias del sistema.



La funcionalidad de registrar información es utilizada en mayor parte por los operadores. La mayor parte de las tareas desempeñadas por los especialistas no son registradas en este sistema, solo son registradas vía mail al coordinador además de enviarle también reportes mensuales resumiendo la situación de los sistemas. Como consecuencia la productividad mensual obtenida a partir de la información registrada en el sistema no refleja la realidad.

Las consultas que brinda el sistema tienen problemas de performance. Las mismas se desarrollaron y ejecutan por fuera del sistema además de no utilizar índices apropiados. El caso más crítico es la consulta de los procesos pendientes de ejecución, que es una de las más utilizadas por los operadores. Este es uno de los motivos por el cual, en el cambio de turno de los operadores, se envían un mail con el detalle de la situación; como contingencia a problemas del correo también se deja un reporte por escrito. Una consecuencia directa es la existencia de redundancia de información.

Todos los cambios y mejoras a realizarse en el ambiente de producción son testeados previamente en los servidores del ambiente de testeo. En el caso de los cambios en las aplicaciones en el ambiente de producción, primero cumplen un ciclo en el sector Desarrollo pasando por distintas etapas: análisis, desarrollo, testeo interno, testeo y aprobación por parte del cliente del plan de ejecución. Luego se pasa al sector Operaciones y el analista de producción coordina la ejecución de la puesta en producción.

Los operadores también desempeñan labores de mesa de ayuda para las aplicaciones que están corriendo en el ambiente de producción y abren incidencias frente a reportes de problemas. Cuando la incidencia requiere la atención de un especialista se reporta por correo al especialista que corresponda. Algunos operadores además tienen asignadas otras responsabilidades como ser encargados de la seguridad e infraestructura, administración de la cintoteca y mantenimiento de la planilla correspondiente al inventario.

2.4 Problemas detectados

A continuación se detallan los principales problemas que se detectaron en el sistema existente y en la modalidad de trabajo del SDC:

- El sistema está desarrollado en una versión vieja de la herramienta utilizada y no tiene mantenimiento.
- Mala performance en las consultas.
- La agenda de las ejecuciones es realizada por medio de un proceso complementario al sistema.
- Hay definidos indicadores de calidad pero son elaborados en planillas de cálculo como complemento al sistema.
- El sistema no tiene administración de usuarios.
- Los especialistas no registran todas las tareas que realizan. Se desea implementar un ciclo interno para registrar todos los cambios y puestas en producción del sector Operaciones.
- Redundancia de información por parte del operador en el cambio de turno.
- No es posible realizar un seguimiento de la mesa de ayuda ya que el correo constituye la única herramienta donde son registradas las incidencias.
- El sistema no facilita el registro correspondiente a procesos de distintos proyectos ya que el sistema no permite trabajar con más de un proyecto a la vez.
- El sistema tampoco lleva registro del operador responsable de cada método, solo registra el operador que realizó la última modificación.
- Actualmente no existe una herramienta orientada a la administración de los medios magnéticos. A corto plazo está planificado incorporar el uso de una herramienta con este fin.



3. Selección de métricas de productividad

En este capítulo se presenta una propuesta de indicadores orientados a mejorar el seguimiento de la gestión y la productividad del SDC. Esta propuesta se encuentra fundamentada en un estudio en profundidad que se realizó sobre las métricas de productividad utilizadas en otros centros de cómputos del mercado relevantes al área de Operaciones y de los requerimientos que se desprenden del capítulo anterior. Dicho estudio se expone en 'Anexo D - Investigación de métricas'.

La única fuente de información que más se involucra en el tema de estudio es la norma COBIT ([1] y [2]), que es un modelo de gestión y control dirigido a las necesidades de la información y sus tecnologías relacionadas. Su objetivo es consensuar los riesgos del negocio, las necesidades de control y los aspectos tecnológicos, mediante la entrega de buenas prácticas aplicables a una estructura lógica de procesos y actividades. Dicha norma tiene una relación más estrecha con temas de seguridad. De todas maneras, se analizó detenidamente y en profundidad dada la carencia de información.

Luego de un análisis de la norma COBIT y del relevamiento realizado del SDC, se elaboró una propuesta de los posibles indicadores de productividad y se presentó ante IBM. Luego de un estudio por parte de la empresa se acordaron los indicadores a ser utilizados por el SDC.

Dichos indicadores permitirán proporcionar una visión global de todos los proyectos y a nivel de un proyecto en particular, permitiendo de esta forma realizar un seguimiento del SDC en función de las tareas. Sus valores deberán situarse entre una cota mínima y una máxima, ambas se deberán estimar a partir de los datos obtenidos durante el uso del sistema por un período de al menos tres meses.

A fin de poder presentar claramente los indicadores se realizó el siguiente glosario con las definiciones de las variables propias de los indicadores:

CPT = cantidad total de procesos

CPE = cantidad de procesos que finalizaron con error

CPO = cantidad de procesos que finalizaron exitosamente.

TEsp = suma de tiempo de espera para todos los procesos.

TM = suma de tiempo muerto para todos los procesos.

TEj = suma de tiempo de ejecución para todos los procesos.

THO = total de horas de operadores = cantidad de operadores x cantidad de horas mensuales (160 horas)

THE = total de horas de especialistas = cantidad de especialistas x cantidad de horas mensuales

TTot = tiempo total de procesamiento = TEsp + TM + TEj

A continuación se detallan los indicadores acordados con IBM a ser utilizados por el SDC:

- Porcentaje de tiempo de espera respecto al tiempo total: es la porción del tiempo total de procesamiento que se debe esperar por una respuesta de algo que está estipulado. Este indicador refleja el tiempo ocioso no debido a problemas en la ejecución de los respectivos procesos.

$$\text{Se calcula como: } 100 \times \frac{T_{esp}}{T_{Tot}}$$

- Porcentaje de tiempo muerto respecto al tiempo total: es la porción del tiempo total de procesamiento que se debe esperar si sucede algún imprevisto. Este indicador refleja el tiempo ocioso debido a problemas en la ejecución de los respectivos procesos.

$$\text{Se calcula como: } 100 \times \frac{T_M}{T_{Tot}}$$



- Porcentaje de tiempo de ejecución respecto al tiempo total: refleja el tiempo real de ejecución con respecto al tiempo total de procesamiento, con tiempo real se refiere al tiempo que la ejecución está consumiendo recursos del sistema.

$$\text{Se calcula como: } 100 \times \frac{TEj}{TTot}$$

- Porcentaje de tiempo de ejecución por hora de operador: refleja la carga de trabajo de los operadores. El valor objetivo debe ser mayor que 100, si es menor indica que hay tiempo ocioso, si es muy grande indica que existe sobrecarga en las tareas de los operadores.

$$\text{Se calcula como: } 100 \times \frac{TEj}{THO}$$

- Porcentaje de tiempo muerto por hora de especialista: refleja el tiempo ocioso de las ejecuciones de procesos debido a problemas en las mismas, también refleja en cierta medida parte de la eficiencia de los especialistas.

$$\text{Se calcula como: } 100 \times \frac{TM}{THE}$$

- Porcentaje de procesos que finalizaron con error: corresponde a la porción de ejecuciones que finalizaron con error sin importar la causa.

$$\text{Se calcula como: } 100 \times \frac{CPE}{CPT}$$

- Promedio de ejecuciones de los procesos antes de obtener resultado satisfactorio: refleja en promedio la cantidad de veces que se ejecutó el mismo proceso antes de obtener resultado satisfactorio.

$$\text{Se calcula como: } \frac{\sum (\text{ejecuciones para los procesos que finalizaron con error})}{CPE}$$



4. Solución al problema

En esta sección se presenta una descripción de los distintos aspectos de la solución construida, como los lenguajes y entornos de desarrollo utilizados en la implementación del prototipo, arquitectura, diseño y decisiones de implementación del sistema. A la vez se plantea el alcance del prototipo desarrollado y los trabajos a realizar a futuro.

4.1 Selección de lenguaje y entorno de desarrollo

La rápida evolución de la informática ha propiciado la aparición constante de nuevos lenguajes de programación, así como de paquetes de software que proporcionan un entorno completo de desarrollo, con todas las utilidades necesarias para el programador.

Los entornos de desarrollo son una útil herramienta de trabajo cuando tenemos que realizar una aplicación compleja que requiera de una interfaz gráfica. Esta interfaz gráfica viene dada por el propio entorno, quien ahorra el trabajo de desarrollo, permitiendo centrarse en la aplicación en sí.

En esta sección se plantean el lenguaje y entorno de desarrollo que se decidió utilizar en base a las necesidades del SDC y un estudio de los más utilizados en el mercado. Dicho estudio se presenta en '*Anexo E - Investigación de entornos de desarrollo*'. De la gran cantidad de lenguajes de programación existentes se seleccionaron los siguientes: Perl ([4]), Lotus Domino ([3]), Java basándose en los estándares de J2EE ([5], [6], [7], [8], [9], [10]) y Visual Studio .Net ([12], [13]).

IBM se encuentra desarrollando el módulo correspondiente al manejo de incidencias de la mesa de ayuda del SDC, el cual está siendo implementado en el lenguaje Perl. Es importante destacar que el nuevo sistema deberá interactuar con el mencionado módulo. Cabe destacar que la base de datos que se utilizó en el nuevo sistema es DB2. Esta decisión fue tomada por parte de IBM.

El nuevo sistema debe estar disponible para las plataformas Windows y Linux. Debe brindar interface web, en este sentido una buena solución la brinda Java con sus nuevas tecnologías como Servlet y JSP ([14], [15], [16]) que son componentes de fácil uso y amplia portabilidad, dominantes de la especificación de J2EE, orientados a implementar la presentación web permitiendo separar el diseño de la página de su lógica.

También debe brindar interface por línea de comando. Esta opción está orientada al registro de información. En este sentido Perl es el lenguaje que mejor satisface este requerimiento debido al poderoso manejo de texto y expresiones regulares que brinda.

Otro punto importante a considerar en la elección del lenguaje es que el mantenimiento del nuevo sistema queda a cargo del SDC. Desde este punto de vista también es importante considerar que el proyecto se limita a la entrega de un prototipo con sus funcionalidades de mayor relevancia, quedando a cargo del SDC la extensión de las funcionalidades del mismo. Cabe destacar que su personal no posee fuertes conocimientos en J2EE, teniendo sí conocimientos y experiencia en programación con Java y Perl.

Considerando los puntos expuestos, las características relevadas en las etapas anteriores, que la complejidad extra que agrega el uso de EJB es mayor que el beneficio que brinda a nuestro producto en particular, el corto plazo existente para la implementación del prototipo, y la adquisición extra de conocimientos técnicos a utilizar, se decidió implementar el nuevo sistema utilizando el lenguaje Java basándose en los estándares de J2EE sin la utilización de EJB. En consecuencia se decidió utilizar Eclipse ([11]) como entorno de desarrollo para la aplicación, motivado fundamentalmente por ser una de las herramientas *open source* más utilizadas en el mercado, de fácil instalación y uso, y estar orientada a Java. Se decidió hacer una excepción para la interface por línea de comando por lo expuesto anteriormente, seleccionando el lenguaje Perl para su implementación.



4.2 Arquitectura

El sistema fue desarrollado en una arquitectura de tres capas independizando la presentación, de la lógica de negocio, y de los datos. Este desacoplamiento permite que cada capa se especialice absolutamente en la funcionalidad que debe brindar sin importar cual es el origen de los datos procesados, bajo costo de mantenimiento y reusabilidad de los componentes. El documento 'Anexo F - Arquitectura del sistema' presenta en mayor detalle la arquitectura.

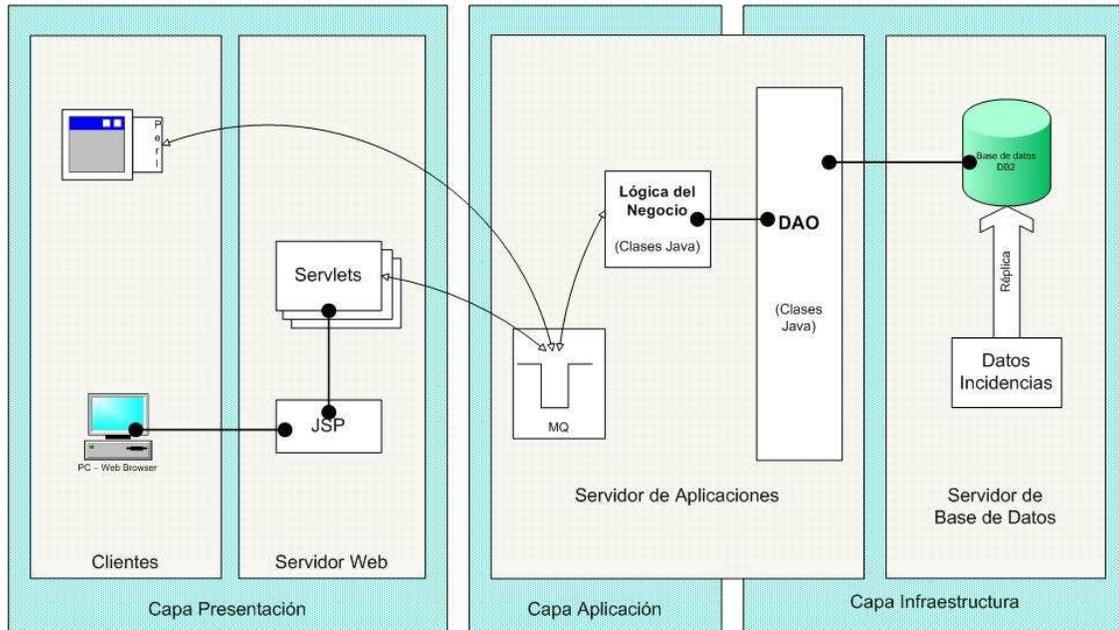


Figura 2 – Diagrama de la Arquitectura de Componentes

Capa de Presentación

Esta capa contiene al servidor web y al cliente, incluyendo todos los componentes gráficos que sirven al usuario para lograr la interacción con el sistema, quien tendrá una interface web y una interface por línea de comando. En la interface gráfica el cliente utilizará un browser para interactuar con las JSP's en el servidor web. Las JSP's se utilizarán para desplegar la información que reciba de los Servlets quienes se encargarán de despachar y controlar los requerimientos del cliente. Para la interface por línea de comando el cliente utilizará una herramienta desarrollada con Perl.

Capa de Aplicación

Esta capa implementa la lógica de negocio, es el nexo entre la capa de presentación y la capa de infraestructura. Se encarga de realizar todas las validaciones necesarias a cada acción solicitada por los clientes a través de la capa de presentación, de proveer los datos a la misma, además de controlar el flujo del programa y asegurar tanto la seguridad como la integridad de los datos. Se desarrolló utilizando clases Java cuyos métodos implementan las validaciones a realizar.

Como mecanismo de comunicación con la capa de presentación, ya sea a través de los Servlets o de la aplicación correspondiente a la interface por línea de comando, se decidió utilizar una cola de mensajes MQ ([19], [20], [21]). Dicha decisión brinda una gran interoperabilidad y cumple con el estándar JMS ([5]) de J2EE, reduciendo de esta forma el impacto si en el futuro se desea implementar la lógica de negocio con EJB por estar totalmente desacoplada de la capa de presentación.



Capa de Infraestructura

Esta capa consta de dos componentes: el acceso a datos y los datos. El acceso a datos fue implementado por clases Java vía JDBC, ya sea a nivel de lectura como de escritura, garantizando la integridad de la información. Los datos se encuentran almacenados en una base DB2, a la cual se replicará la información correspondiente a registro de incidencias, medios magnéticos e inventarios. Esta replicación se realizará una vez al mes ya que esta información es relevante solo para la estimación de los indicadores que forman parte en la generación de los reportes mensuales.

4.3 Diseño

En esta sección se presenta una breve descripción del negocio detallando el modelo del dominio del sistema y el funcionamiento del mismo. También se presenta una breve descripción de los casos de uso formalizando de esta manera las funcionalidades que brinda el sistema, y por último se detalla su diseño modular. En 'Anexo G - Diseño del sistema' se puede apreciar con mayor profundidad este tema.

4.3.1 Descripción del negocio

Los procesos de negocio relativos al sistema son los siguientes:

- Administración de Usuarios (P1): engloba un conjunto de procesos simples encargados de la administración de los usuarios del sistema. Permite la incorporación de nuevos usuarios, eliminación de los mismos, asignación y modificación de permisos.
- Ejecución de Procesos (P2): involucra un conjunto de procesos relacionados a la gestión de solicitudes de ejecuciones, registro de ejecución y consultas de las mismas.
- Control de Operaciones (P3): engloba a los procesos encargados de calcular los indicadores definidos.
- Administración de Codigueras (P4): concierne a la gestión de las codigueras propias del sistema.

Los procesos P2 y P3 son los más importantes del sistema y son los que tienen mayor impacto en su diseño.

Proceso P2

A continuación se ilustra el modelo de dominio para familiarizarse con el vocabulario del dominio y facilitar posteriormente el entendimiento de su diseño. Los términos específicos se describen en el glosario.

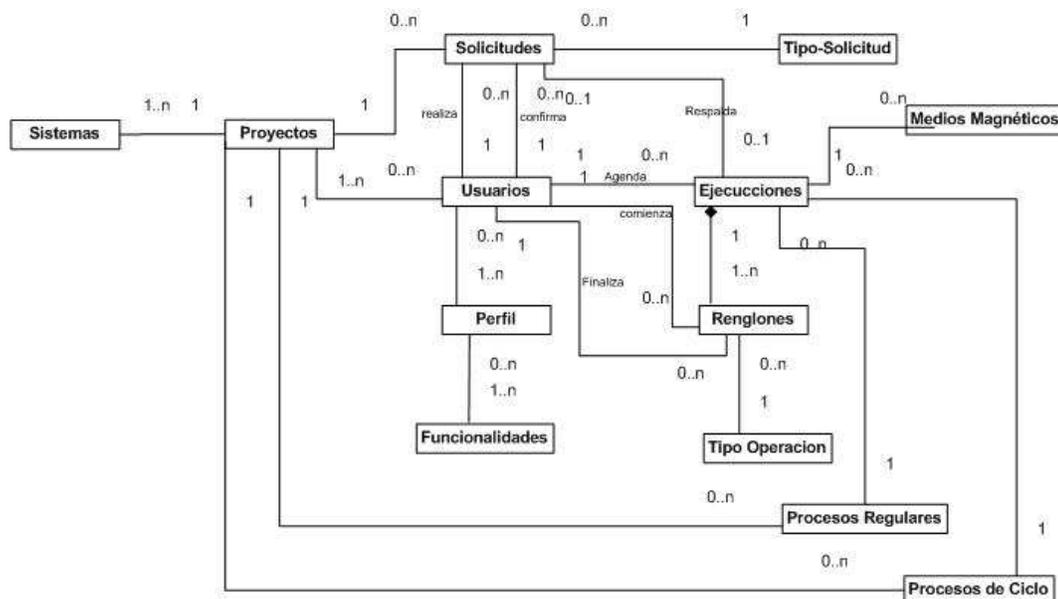


Figura 3 – Modelo de Dominio



Como se mencionó en el capítulo 2, los procesos se categorizan según su frecuencia de ejecución y su cometido. Las peticiones de sus ejecuciones son originadas de distintas formas dependiendo del tipo de proceso en cuestión.

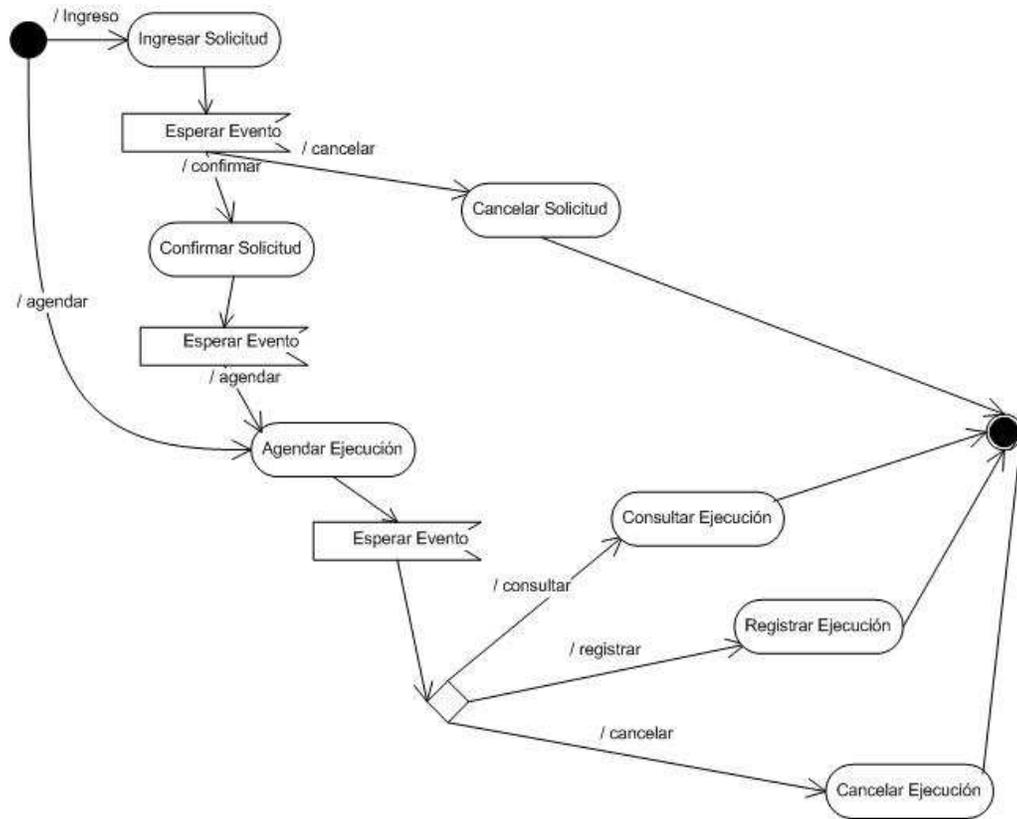


Figura 4 – Proceso P2

En el caso de los procesos no regulares, procesos de puesta en producción, y procesos de cambio, la intención de ejecutar tal proceso es manifestada por medio del ingreso de una solicitud de ejecución. Dicha solicitud debe ser aprobada o descartada por el responsable del área donde se generó la misma, es decir se confirma o cancela respectivamente. En caso de confirmarse dicha solicitud, llegará al sector Operaciones donde el analista de producción decide y agenda la ejecución para el momento que crea conveniente en función de las tareas del SDC.

En el caso de los procesos regulares se les asoció un cronograma indicando los días de la semana y hora que debe ejecutarse cada uno. En base a dicho cronograma, el analista de producción puede consultar para un período determinado todos los procesos que corresponde ejecutar, debiendo autorizar solamente las deseadas, incluso modificando el momento de su ejecución si cree necesario. También es posible solicitar una ejecución de un proceso regular fuera del cronograma establecido; dicha petición se genera por medio del ingreso de una solicitud de ejecución.

En el caso de los procesos de ciclo, sus ejecuciones no son solicitadas por los usuarios ni se les asoció un cronograma como en el caso de los procesos regulares. Por este motivo el analista de producción es el responsable de tener conocimiento y realizar la agenda de ejecución para un proceso de esta naturaleza cuando corresponde. Este tipo de proceso se caracteriza por estar compuesto de varios subprocesos con un orden de precedencia establecido. En base a esto, el analista de producción al realizar la agenda de una ejecución de un proceso de ciclo está realizando la agenda de la/s ejecución/es de los subprocesos iniciales, es decir los que no tienen ningún subproceso previo. A medida que las ejecuciones se concretan y el orden



de precedencia establecido lo permite, las agendas de los restantes subprocesos se realizan en forma automática. De esta forma se continúa con la ejecución del ciclo de una forma fluida sin las demoras que existirían en caso de requerirse la intervención del analista de producción para que realice dichas agendas.

Luego que se agenda una ejecución se puede tomar uno de dos caminos distintos, se cancela la misma ó se concreta dicha ejecución, en cuyo caso se registran todos los datos relevantes a la misma (ver "Registrar ejecuciones" en 'Anexo I - Manual de instalación'; por más detalles).

Tanto las solicitudes como las ejecuciones en sí, tienen asociados distintos estados dependiendo de la etapa en la que se encuentren. A continuación se detallan sus diagramas de estado.

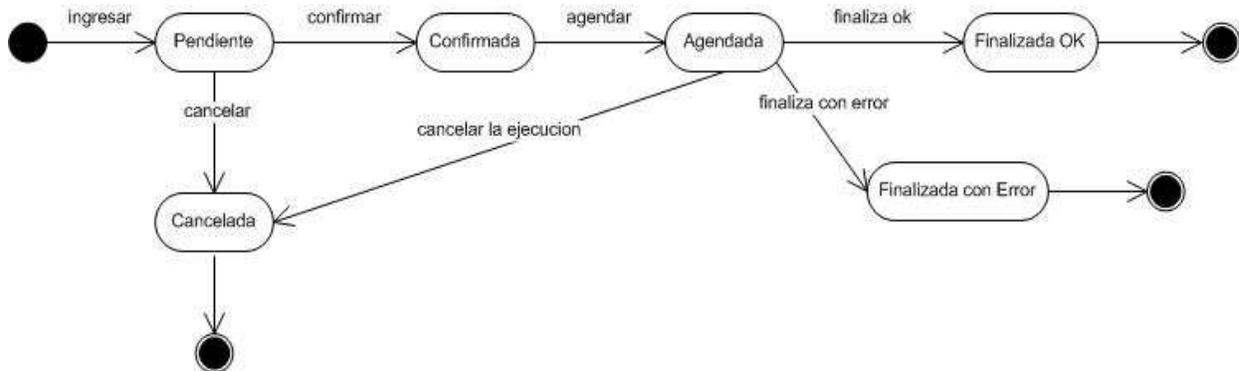


Figura 5 – Diagrama de estado de las solicitudes

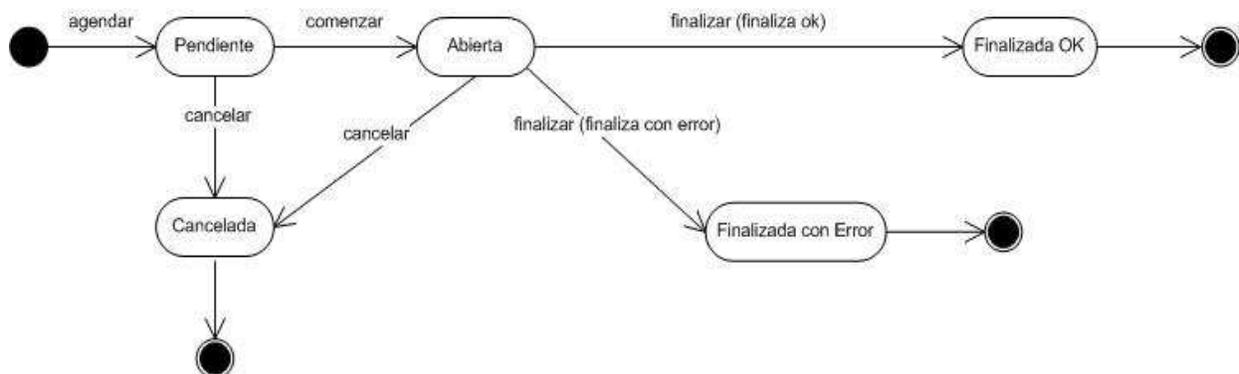


Figura 6 – Diagrama de estado de las ejecuciones

Proceso P3

Este proceso concierne al cálculo de los indicadores en base a la información registrada por el proceso P2 principalmente, también hay información que es replicada de otros sistemas. IBM aprobó la decisión de utilizar archivos XML ([14], [17], [18]) como especificación de cada indicador, dicha decisión contempló la ventaja y la desventaja que esto implica. La ventaja obtenida es la flexibilidad en el sentido que se pueden incorporar indicadores de forma sencilla y con un bajísimo costo. La desventaja es que cada indicador se debe resolver únicamente por medio de consultas sobre la base de datos sin la utilización de código extra.



4.3.2 Casos de Uso

A continuación se describen los casos de uso del sistema:

- Ingreso al sistema: involucra al usuario que desea ingresar al sistema, el acceso al sistema se realiza por autenticación del usuario. Cada usuario tendrá al menos un perfil asociado, donde cada perfil implica un conjunto de funcionalidades habilitando a dichos usuarios hacer uso de las mismas. Este desacoplamiento entre los usuarios y las funcionalidades facilita la administración de usuarios y sus permisos.
- Ingresar solicitud: involucra al usuario que desea solicitar una ejecución, cuyo proceso deberá ser de uno de los siguientes tipos: no regular, puesta en producción, de cambio, regular (a demanda). Este caso de uso genera una solicitud en estado “pendiente”, y permite asignarle la prioridad de su ejecución.
- Confirmar solicitud: involucra al usuario que desea confirmar o cancelar una o varias solicitudes de ejecución en estado pendiente, cambiando a estado confirmada o cancelada respectivamente.
- Agendar ejecución: se entiende por agendar como la orden de ejecutar un proceso en el momento que se indica. Este caso de uso involucra al usuario que desea agendar una o varias solicitudes para su ejecución cambiando su estado de confirmada a agendada. También habilita la agenda de ejecuciones de los procesos regulares según el cronograma establecido así como para los procesos de ciclo cuando corresponde. Como resultado se genera una ejecución en estado pendiente y con el momento que debe concretarse.
- Cancelar ejecución: corresponde a la cancelación de la ejecución de un proceso sin importar su tipo. La precondition es que la ejecución debe estar en estado pendiente o abierta. Como consecuencia cambia el estado de la ejecución a cancelada y en caso de existir una solicitud que ampare dicha ejecución también cambia el estado de la misma a cancelada.
- Registrar ejecución: involucra al usuario que desea registrar información relevante a la ejecución de un proceso. Este caso de uso permite llevar registro, paso a paso, en la ejecución de cada uno de los métodos según lo indicado en el respectivo manual de ejecución del proceso. El registro de información debe ser en tiempo real. Cuando la ejecución del proceso finaliza, en caso de tratarse de la finalización de un subproceso de un proceso de ciclo, automáticamente realiza la agenda de ejecución de los subprocesos correspondientes de acuerdo a las precedencias establecida entre los mismos.
- Consultar ejecución: involucra al usuario que desea consultar información relevante a una o varias ejecuciones. Permite consultar por distintos filtros, por ejemplo por estado de la ejecución, tipo de proceso, proyecto, parte del nombre (tanto prefijo como sufijo), etc.
- Mantenimiento de codigueras: corresponde al mantenimiento de las codigueras propias del sistema.
- Consultar codigueras: corresponde a la consulta de las codigueras propias del sistema.
- Administración de usuarios: corresponde a la administración de los permisos asignados a los usuarios, permitiendo dar el alta a nuevos usuarios para hacer uso del sistema, dar de baja usuarios, administrar los perfiles asociados a cada usuario y administrar las funcionalidades asociadas a cada perfil.
- Control de Operaciones: corresponde a la estimación de los indicadores de productividad en la operativa de trabajo. Dichos indicadores se estiman en base a la información registrada correspondiente a la ejecución de los procesos y de las incidencias registradas referentes al sector Operaciones. De aquí la importancia que el registro de información se realice en tiempo real, lo que redundará en minimizar errores en la estimación de los indicadores.



4.3.3 Diseño modular

El enfoque utilizado es un diseño modular y orientado a objetos. A continuación se detallan los módulos presentes en cada capa de la arquitectura del sistema y como estos se relacionan.

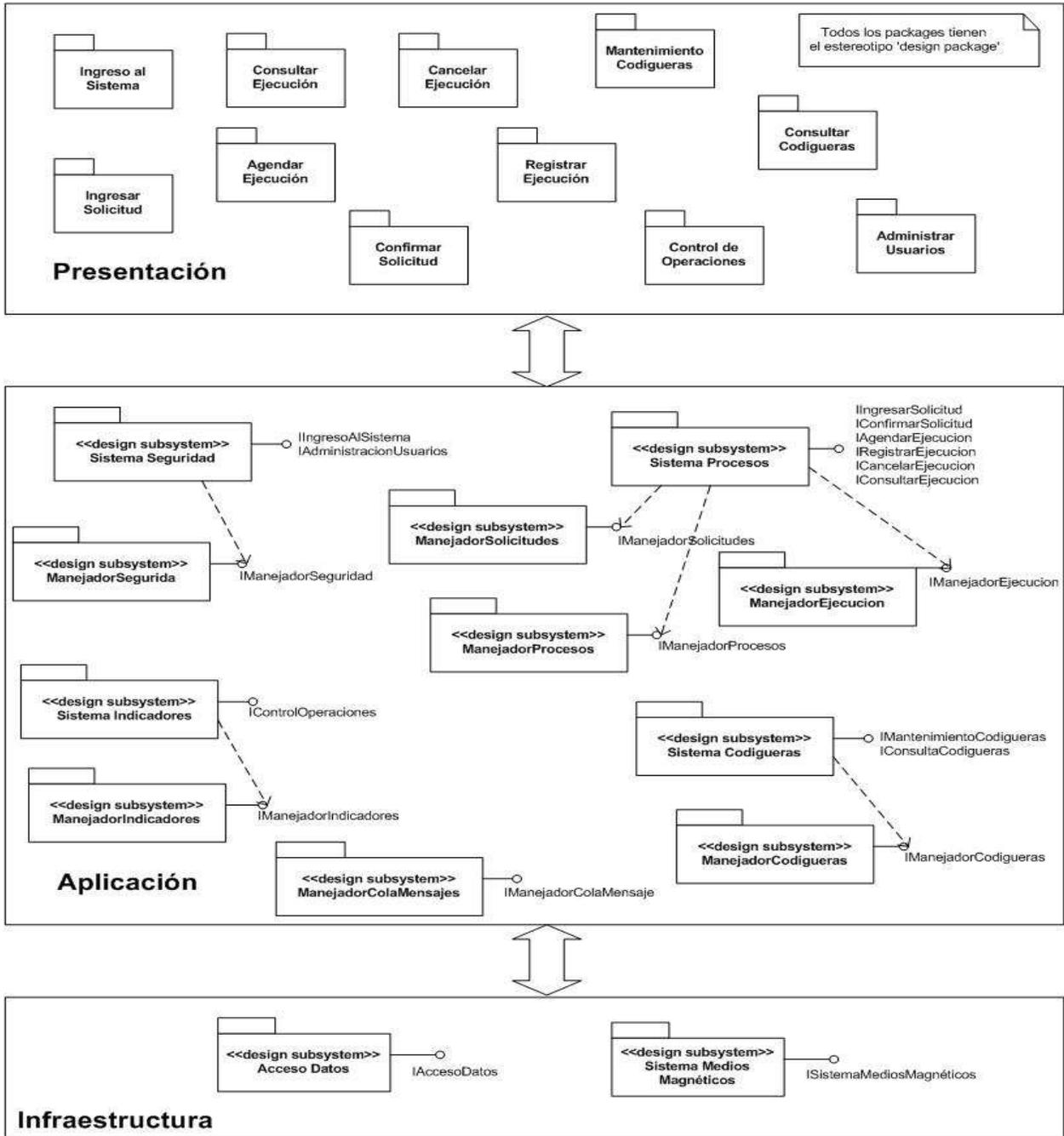


Figura 7 – Módulos del sistema

Presentación

Este sistema es una aplicación basada en el web, por esta razón cuenta con un grado de distribución a nivel de interfaz de usuario. A nivel de usuario final, corre en su estación de trabajo una aplicación llamada



browser. Esta aplicación es la encargada de presentar al usuario la interfaz de la aplicación y de enviar al servidor las acciones que el usuario realiza.

Por el lado del servidor se encuentra otra aplicación general, llamada servidor web, capaz de generar páginas web en forma dinámica. El servidor web recibe las solicitudes del usuario, utilizando el protocolo HTTP, y redirige dichas solicitudes a las páginas dinámicas ubicadas en los módulos de la capa de presentación. Existe un módulo para cada caso de uso identificado, que contiene la lógica que lleva adelante el caso de uso y un conjunto de JSP's utilizadas por dicha lógica. De esta forma la vista de casos de uso muestra el front-end del sistema, el cual es generado dinámicamente utilizando tecnología JSP. Desde el punto de vista del back-end se tiene un conjunto de Servlets generados a partir de los procesos llevados a cabo por el sistema. El servidor web además asocia a cada usuario información de sesión y es él quien se encarga de gestionar la atención a múltiples clientes en forma simultánea.

Aplicación

Cada módulo de la capa superior (capa de presentación) utiliza servicios de esta capa. Dichos servicios son ofrecidos por medio de interfaces, una por cada caso de uso. Por GRASP facade controller ([21], [22], [23]) restringido a las operaciones de un subsistema, se creó un módulo por subsistema que realiza todas las interfaces requeridas por los casos de uso asociados al mismo. También se creó un módulo que realiza las interfaces que administran cada tipo de negocio y los tipos que los categorizan, encapsulando la forma en que los datos están almacenados, la distribución de estos datos y el acceso a ellos. Se cuenta además con el módulo de mensajería que encapsula todo lo relativo con la comunicación entre las capas de aplicación y la de presentación por medio de la cola de mensajes MQ.

A continuación se presenta una breve descripción para cada uno de los subsistemas existentes:

- Subsistema de seguridad: el ingreso al sistema se realiza por medio de usuario y password, y la seguridad se basa en seguridad por funcionalidades.
- Subsistema de procesos: brinda todos los servicios que conciernen a la ejecución de procesos. Esto lo realiza por medio de la utilización de los servicios brindados por el subsistema del dominio. La realización de las interfaces provistas por este módulo, en una clase llamada *Controller*, se logra mediante solicitud de los servicios respectivos a los módulos que administran cada tipo de negocio y los tipos que lo categorizan. La mayoría de los métodos simplemente delegan la solicitud del servicio respectivo a los módulos del tipo de negocio correspondiente.
- Subsistema del dominio: brinda la definición de los objetos del dominio que mapean la realidad y sus interdependencias. Además brinda los servicios que permiten la manipulación de los objetos logrando de esta forma incrementar el desacoplamiento entre los subsistemas y un comportamiento controlado de los mismos.
- Subsistema de codigueras: brinda los servicios relevantes al mantenimiento de las codigueras del sistema.
- Subsistema control de operaciones: brinda los servicios concernientes al cálculo de los indicadores por medio de la interface que este módulo realiza. Sus métodos delegan la solicitud del servicio al módulo que encapsula el tipo de negocio correspondiente a los indicadores. Dicho módulo encapsula el manejo de los archivos XML y el cálculo de los respectivos indicadores, en una clase llamada *Controller* que utiliza los servicios respectivos a los módulos que administran cada tipo de negocio y los tipos que lo categorizan.

Infraestructura

Los módulos localizados en esta capa son de dos tipos, adaptadores a sistemas existentes ó servicios generales útiles para cualquier aplicación. Estos están disponibles para todos los módulos de la capa de aplicación. Dentro de la primer categoría se encuentra el sistema de medios magnéticos, este módulo encapsula todo lo relativo con la comunicación con dicho sistema. Los componentes que realizan las interfaces que soportan dichos módulos cumplen la función de adaptador (*Adapter Design Pattern*), adaptando los parámetros y tipos de retorno y el mecanismo de invocación al sistema real o módulo



correspondiente. Dentro de la segunda categoría se encuentra el módulo de acceso a datos, el cual encapsula la conexión al motor relacional.

4.4 Implementación

Para la implementación se utilizó una metodología de desarrollo incremental. Se realizó mediante aproximaciones sucesivas respetando el orden de los casos de uso en función de importancia respecto al negocio y a la vez que permitiera a IBM observar el avance del proyecto dado el atraso del mismo y el escaso tiempo para esta etapa. Esto permitió realizar los ajustes necesarios en función de las necesidades y grado de satisfacción de IBM a lo largo de su desarrollo, minimizando el trabajo que implicó las mejoras que IBM fue planteando a medida que se entregaban las nuevas versiones.

El sistema se desarrolló en Java y se utilizó:

- Java Development Kit versión 1.4.07 de Sun
- Servidor web Apache Tomcat versión 5.0 ([15], [16])
- Especificaciones de JSP 2.0 y Servlet 2.4
- Entorno de desarrollo Eclipse versión 3.0
- Base de datos DB2 versión 8.0

A continuación se describen cuales fueron las decisiones de implementación que permitieron reflejar las clases que se realizaron en la etapa de diseño al programa final.

Presentación

Cada caso de uso puede requerir más de una interacción del usuario con el sistema. En esta capa la implementación de cada uno de dichos casos de uso se encuentra encapsulada en un módulo respetando el diseño. Se utilizó el patrón Model-View-Controller (MVC) ([21], [22], [23]), cada módulo contiene una o varias JSP dependiendo de las interacciones que demanda el caso de uso. Dichos módulos también contienen al menos un Servlet que se encarga de la lógica de presentación del caso de uso en cuestión, es decir tiene el control sobre las JSP correspondientes, y utiliza los servicios brindados por el respectivo módulo en la capa de aplicación.

Cuando un usuario ingresa al sistema almacena en su información de sesión las funcionalidades y proyectos que tiene asignados. De esta forma se evita invocar los métodos correspondientes cada vez que el usuario desea realizar una acción, ya que esta información prácticamente no sufre actualizaciones, redundando en menor tiempo de ejecución y menor carga al servidor de aplicación y base de datos. En la sesión de usuario también se almacenan los parámetros que deben ser recordados a lo largo de la ejecución del caso de uso, eliminando dicha información cuando el caso de uso finaliza.

A continuación se detallan las decisiones adoptadas en la generación de las JSP:

- Para nombrar los parámetros y principales funcionalidades de la aplicación, se utilizaron los términos que emplea el usuario.
- En los casos en los que se requiere ingreso de información de los usuarios, se utilizaron campos de texto, los cuales están habilitados ó no en función de los permisos asociados al mismo.
- En los casos que se requiere registro de información cuyos valores posibles corresponden a una lista predeterminada se utilizaron combos. Esto redundo en fácil ingreso de información eliminando los errores de tipeo si se utilizara un campo de texto para su registro, ó en minimización de costo asociado al diseño si por ejemplo se utilizaran radio buttons y hubiera que agregar o quitar opciones.
- Se emplearon “radio buttons” cuando el usuario debe elegir una opción entre varias y “check boxes” cuando es posible seleccionar más de una opción.
- Los valores por defecto fueron definidos por el usuario.
- Los mensajes presentados al usuario sólo desaparecen cuando este lo decide.



- En el caso de uso correspondiente a control de operaciones, los valores por defecto de los parámetros de entrada y los campos correspondientes a la información a desplegar son configurados en los archivos XML respectivos.

Aplicación

Respetando las decisiones de diseño, por cada caso de uso se creó una interface que brinda las operaciones necesarias a la capa de presentación para que sea capaz de llevar adelante la ejecución de los mismos. Dichas interfaces se encuentran agrupadas en función de su dominio, y cada una fue implementada por una única clase cuyos métodos simplemente delegan la solicitud del servicio a los módulos del tipo de negocio correspondiente. Esto se debe a la baja complejidad de los casos de uso, ya que en su mayoría corresponden solo a consultas ó simplemente a registro de información.

Cada uno de los módulos que encapsula un tipo de negocio provee una sola interface, cuya realización se encuentra en un controlador façade ([21], [22], [23]) llamado *Manejador* que coordina el tipo de negocio respectivamente. Estos módulos además contienen las clases que modelan los componentes del tipo de negocio respectivo y encapsulan su comportamiento, solo los métodos de la clase *Manejador* acceden a dichos objetos. Con esto se logra un bajo acoplamiento y las cantidades de controladores y de operaciones del sistema por controlador se mantienen manejables. El *Manejador* también incluye el manejo de excepciones para permitir la detección de errores y recuperación ante fallas.

El uso de interfaces permite la abstracción de la implementación en una especificación. Con esto se evita la dependencia de una implementación particular, permitiendo su reemplazo por otra implementación que realice las mismas interfaces que la anterior con un mínimo esfuerzo.

A continuación se presentan los modelos de información utilizados. La figura 8 presenta los componentes que administra la clase *ManejadorSeguridad*, la figura 9 presenta los componentes que administra la clase *ManejadorProcesos*, la figura 10 presenta los componentes que administra la clase *ManejadorSolicitudes*, la figura 11 presenta los componentes que administra la clase *ManejadorEjecuciones*, y la figura 12 presenta los componentes que administra la clase *ManejadorIndicadores*.

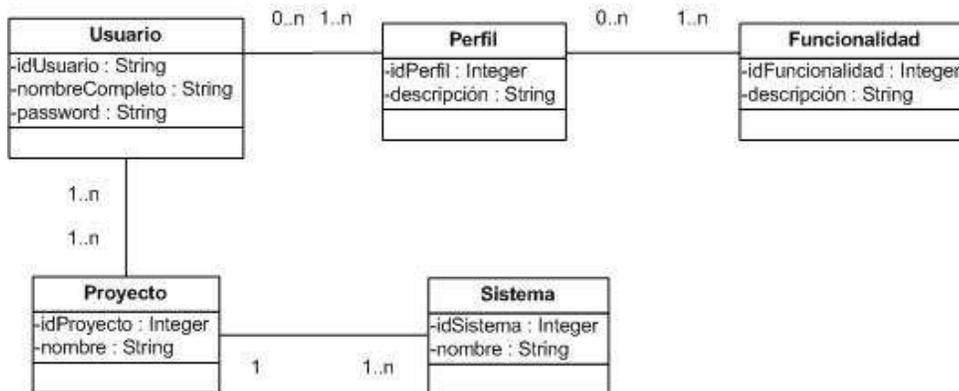


Figura 8 – Modelo de información correspondiente a seguridad

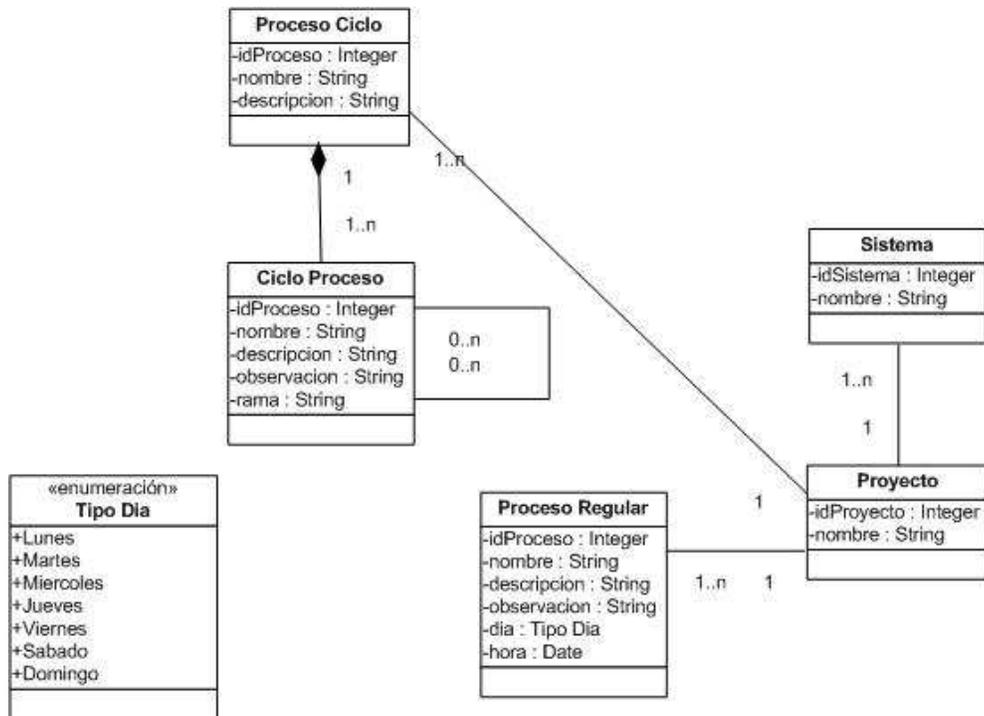


Figura 9 – Modelo de información correspondiente a los procesos de ciclo y regulares

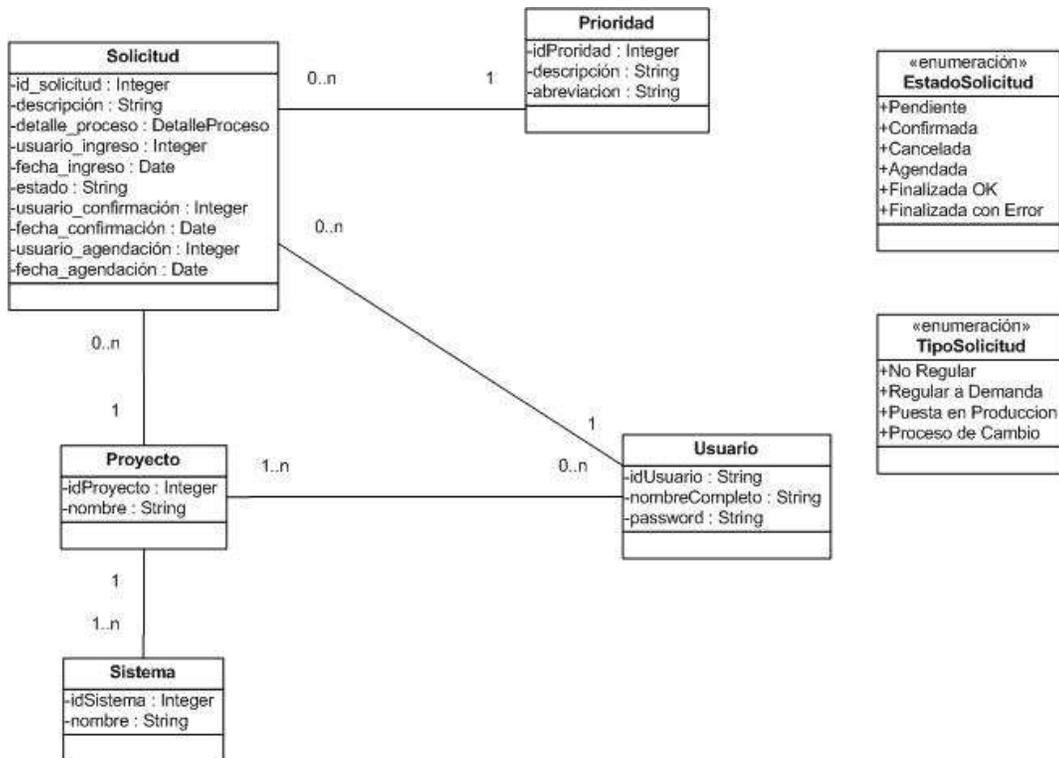


Figura 10 – Modelo de información correspondiente a las solicitudes

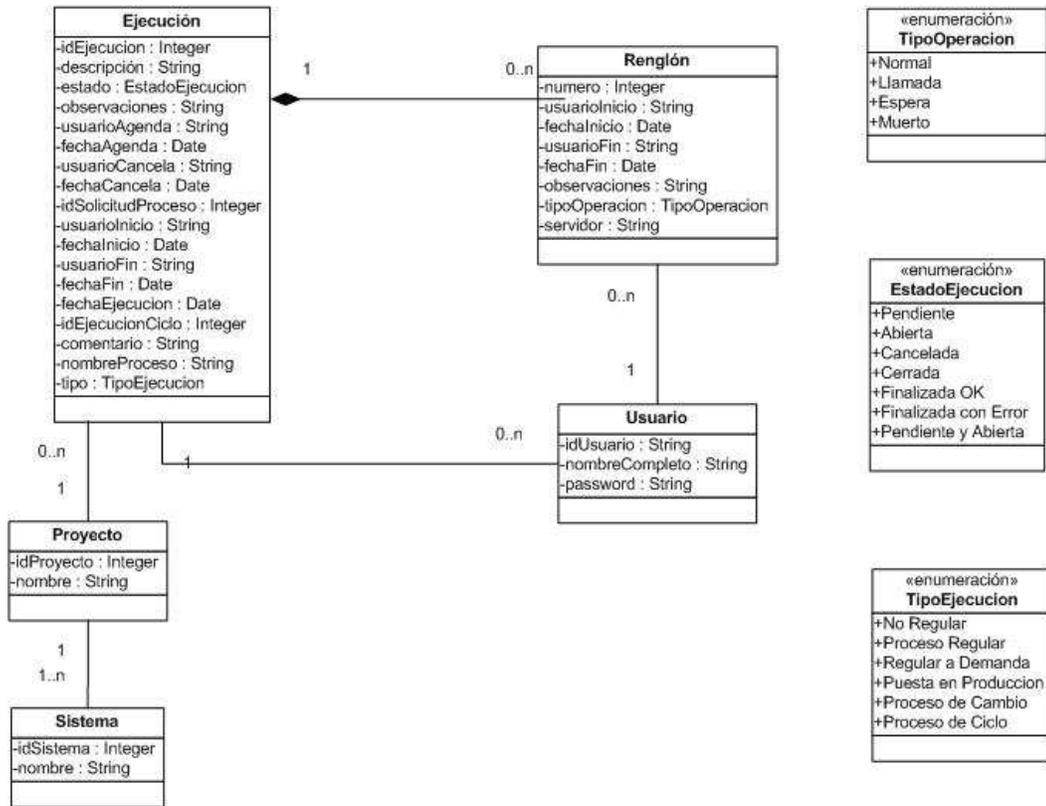


Figura 11 – Modelo de información correspondiente a las ejecuciones

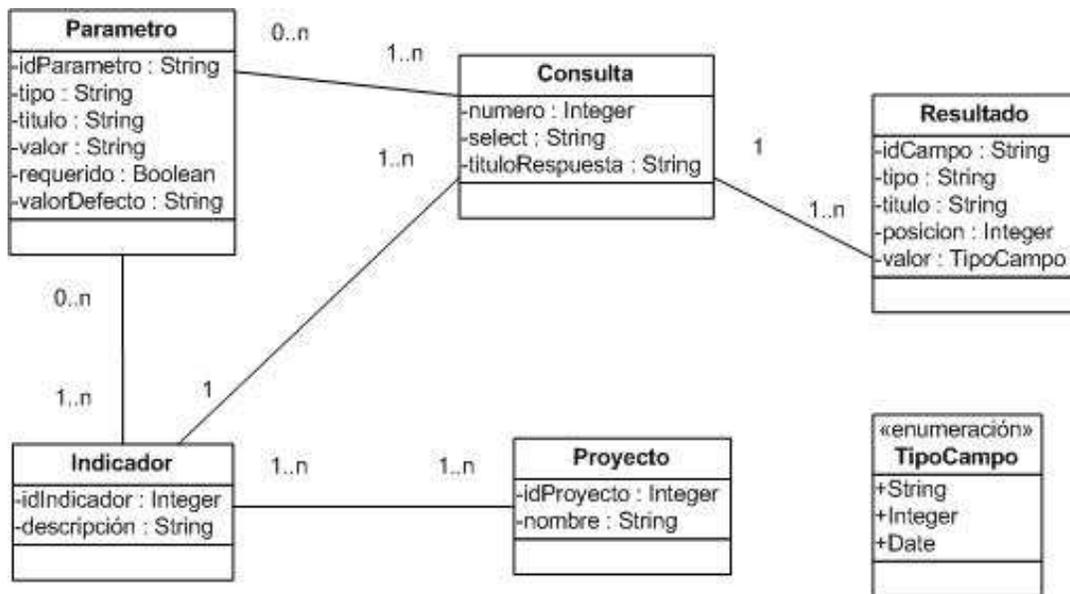


Figura 12 – Modelo de información correspondiente a indicadores



En el caso del módulo correspondiente al tipo de negocio de indicadores, se implementó una clase para procesar el respectivo archivo XML. Dicho archivo contiene: la secuencia de consultas que corresponde ejecutar definidas para el cálculo del indicador, la información necesaria sobre los parámetros para la ejecución de cada una de las consultas y para devolver el resultado correspondiente. Ya que no existe un estándar al parsear un documento XML, se implementó utilizando JAXP, que es un conjunto de interfaces abstractas que uniformizan el manejo de parsers tanto SAX como DOM haciéndolo independiente del fabricante del parser. Es decir que se puede cambiar el parser en el futuro sin repercusiones en este módulo. Además JAXP es la API estándar para procesar documentos XML que provee Java siendo compatible a la especificación J2EE. La implementación se centró en la filosofía de los parsers DOM ya que representan al documento por un modelo de objetos, facilitando su manejo y no viendo comprometida la performance en este caso por tratarse de estructuras simples y pequeñas.

Se implementaron tres archivos XML con el objetivo de calcular los indicadores seleccionados y presentados en el capítulo 3. Dichos archivos no calculan directamente estos indicadores, sino que proveen de la información necesaria para su cálculo, permitiendo a la vez obtener una visión más amplia en el sentido que es posible observar con un mayor grado de refinamiento cada uno de los indicadores. A continuación se presentan los tres archivos que se implementaron; por mayor detalle relativo a los parámetros y formato de salida de la información ver '*Anexo J - Manual de usuario*'.

- Total tiempo por tipo operacion.xml: calcula la cantidad de tiempo insumido por las ejecuciones discriminado por tipo de operación. Esta información permite calcular los porcentajes de tiempo muerto/espera/ejecución respecto al tiempo total, es decir los tres primeros indicadores presentados en el capítulo 3.
- Total tiempo por perfil.xml: calcula la cantidad de tiempo y cantidad de usuarios involucrados en las ejecuciones, discriminado por tipo de operación y perfil. Permite calcular el cuarto y quinto indicador presentado en el capítulo 3.
- Total ejecuciones por estado.xml: calcula la cantidad de ejecuciones discriminada por el estado de las mismas y también la cantidad de reejecuciones. Esta información permite calcular el sexto y séptimo indicador presentado en el capítulo 3.

Infraestructura

Se utilizó una base de datos relacional que corre sobre el motor de base de datos DB2, el modelo de datos se ilustra en la figura 13.

El componente correspondiente al acceso a datos brinda los servicios de persistencia de datos, dichos servicios se agruparon en un módulo por cada tipo de negocio logrando un alto grado de desacoplamiento entre los mismos. Cada uno de estos módulos ofrece los servicios por medio de una única interface, evitando también la dependencia de una implementación particular, permitiendo por ejemplo cambiar toda la implementación de esta capa sin comprometer la capa de aplicación.

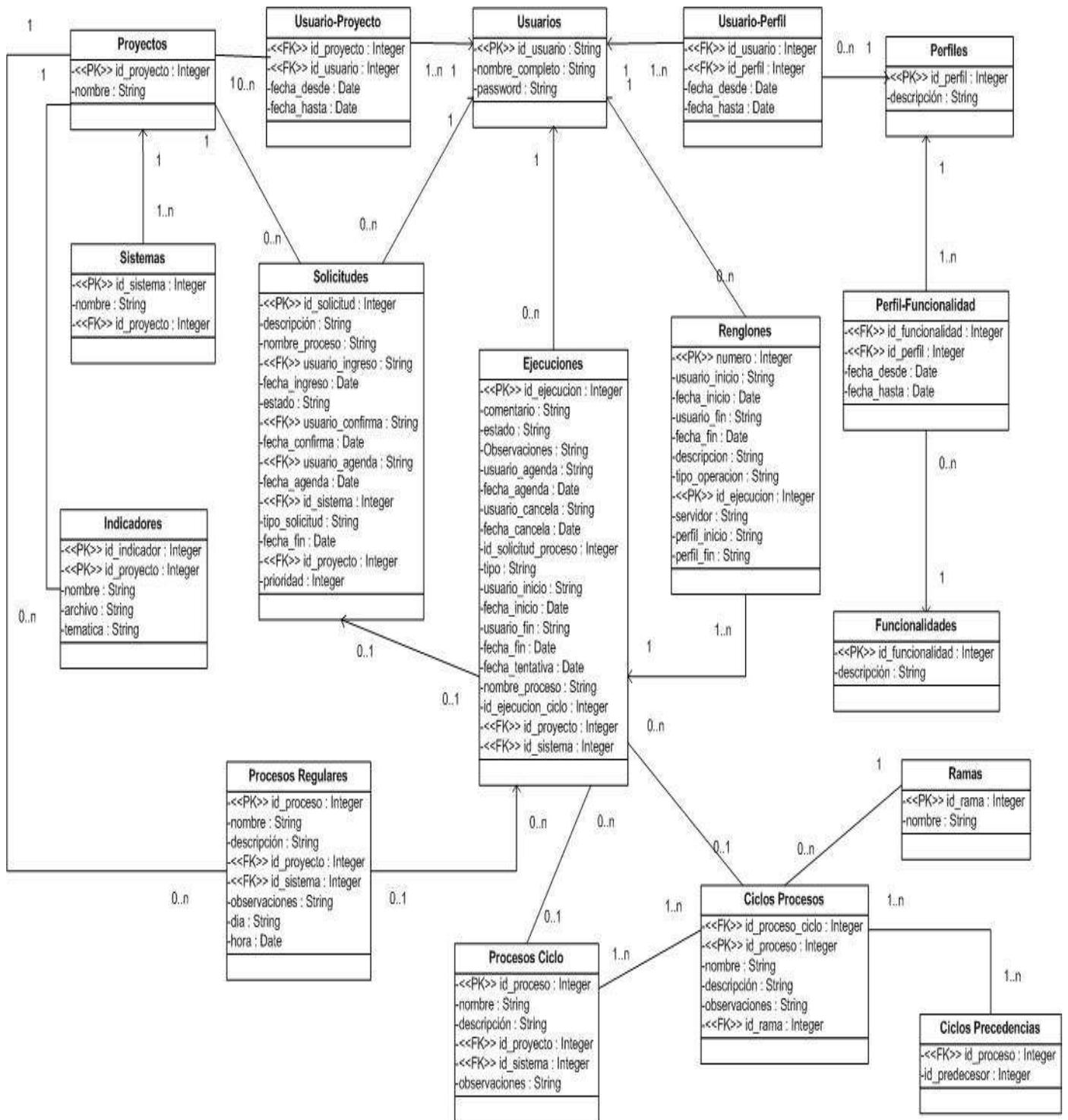


Figura 13 – Modelo de Datos



4.5 Alcance del prototipo

En la etapa de implementación del sistema se definió un plan de releases en conjunto con IBM por razones de tiempo. El alcance del prototipo desarrollado se limitó al primer release que se detalla a continuación.

Release 1

Este release excluyó los casos de uso menos importantes: mantenimiento de codigueras, consulta de codigueras y administración de usuarios, ya que los mismos no cubren requerimientos importantes del sistema y están desacoplados del resto de los casos de uso.

Si bien el caso de uso correspondiente a los indicadores constituye uno de los requerimientos de mayor importancia del sistema, este primer release solo cubrió el subconjunto de indicadores más relevantes que IBM desea estimar a corto plazo, relegando la implementación de los demás indicadores en uno de los próximos releases.

La comunicación entre las capas de presentación y de aplicación no se implementó por medio de colas de mensajes MQ como se definió en la arquitectura. Esto redujo la interoperabilidad del sistema acotando el acceso a la aplicación con acceso web. La decisión de utilizar MQ se basó en el requerimiento de permitir el acceso al sistema por medio de línea de comando. La implementación de la herramienta orientada a brindar la interface por línea de comando está a cargo de IBM y aún no tiene una fecha de finalización a corto plazo debido a manejo de las prioridades dentro de sus planificaciones. En consecuencia el impacto de esta decisión es mínimo a corto plazo.

El prototipo que se desarrolló cumplió con los alcances definidos de acuerdo a este release. Se relegó para el futuro la implementación de los restantes releases, en el mismo orden que se enumeran.

Release 2

Este release introduce el uso de colas de mensajes MQ para la comunicación entre la capa de presentación y la capa de aplicación. Con esto se cubre el requerimiento correspondiente a la interoperabilidad. El diseño contempló esto, respecto a la capa de aplicación impacta en los Servlets, las invocaciones a las interfaces de la capa de aplicación deberán realizarse por medio de MQ. Respecto a la capa de aplicación, deberá implementarse el módulo correspondiente que se encargará de despachar los mensajes de la cola MQ.

Release 3

El alcance de este release es incorporar los indicadores no considerados en el primero. El impacto de este release está limitado al módulo correspondiente a indicadores y es mínimo ya que cada indicador es implementado por un archivo XML, garantizando de esta forma su total desacoplamiento de los demás indicadores. Es decir, solo deberá generarse los archivos correspondientes a cada uno de los indicadores a incorporar; en *'Anexo I – Manual de instalación'* se detallan los pasos a seguir para esto. También deberá planificarse y comenzar con la replicación de la información necesaria para la estimación de los indicadores seleccionados concernientes a las incidencias.

Release 4

Este release incorpora los siguientes casos de uso: mantenimiento de codigueras, consulta de codigueras, y administración de usuarios, cubriendo de esta forma la totalidad de los requerimientos del sistema. El impacto de dicho release es menor ya que cada caso de uso es desarrollado en módulos independientes, limitando también el testeado a los módulos correspondientes. Esto se contempló en la definición del diseño. Se especificó la interacción del usuario con el sistema en cada caso de uso en lo que respecta a capa de presentación. A nivel de presentación, se deberá generar el conjunto de JSP de acuerdo al diseño y respetando las decisiones de diseño gráfico que se utilizó en las demás ventanas. También se deberá generar los Servlets que controlaran dichas JSPs. Respecto a la capa de aplicación, se deberá implementar las clases que implementarán las interfaces de sus módulos, las mismas fueron especificadas en el diseño.



5. Conclusiones y trabajo futuro

En la presente sección se exponen los resultados obtenidos y los aportes brindados por la tarea realizada en cada una de sus etapas, tanto para IBM como a título personal. También se narran las principales dificultades que se enfrentaron durante la realización del proyecto.

Al inicio del proyecto se elaboró un cronograma a seguir estableciendo además de las etapas los entregables correspondientes a cada una. En la etapa inicial se realizó un relevamiento del estado del SDC con el objetivo de fortalecer las debilidades a nivel operativo y tener presente las carencias del sistema anterior a la hora de desarrollar el nuevo sistema. Esta etapa fue fundamental para el conocimiento del negocio constituyendo una de las bases en la toma de decisiones en las etapas posteriores. La falta de documentación de la herramienta anterior se compensó con la constante disposición y firmes conocimientos del personal sobre la misma, toda la información relevada se obtuvo de las reuniones concretadas con los distintos perfiles de usuario existentes. En consecuencia, además de adquirir el conocimiento del negocio y entender la forma de trabajo, contribuyó en encontrar una solución adecuada a todos los problemas planteados ya que como usuarios del sistema transmitieron parte de los problemas existentes. Durante esta etapa se generó un pequeño atraso de unos pocos días respecto al cronograma establecido.

En la siguiente etapa se obtuvo una selección de indicadores luego de un exhaustivo estudio del estado del arte de métricas de productividad relevantes al área de Operaciones en otros centros de cómputos. Dicha selección se realizó junto a IBM respetando las necesidades e intereses de la empresa. Durante la etapa de investigación, se enfrentó un factor que no se había considerado, la falta de información en esta rama de la informática. Esto demostró que el área de Operaciones aún no ha llegado a considerarse un factor relevante en la productividad de una empresa, por lo que IBM ha considerado explotar la información de dicha área en beneficio del cliente y propio, permitiendo un seguimiento más aproximado de los niveles de servicio ofrecidos así como la optimización de sus recursos. El tiempo insumido en la búsqueda de información fue mayor del esperado, siendo muy grande el volumen de información referente a métricas de productividad pero ninguna relevante al área deseada. Cabe aclarar que luego de un mes de intensa investigación aún no se había logrado obtener un mínimo de información concerniente al tema de estudio del proyecto. El tiempo total insumido por esta etapa fue de tres meses aproximadamente, mientras que el tiempo estimado en el cronograma fue de un mes.

La última etapa consistió en la construcción de un prototipo con el objetivo de centralizar la información y utilizarla en la estimación de los indicadores seleccionados en la etapa anterior. Para la elaboración del prototipo se incursionó en las tecnologías de desarrollo de última generación, analizando y seleccionando las que mejor se adecuaban a los requerimientos desprendidos de las etapas anteriores. En consecuencia el prototipo se desarrolló utilizando herramientas *open source* que garantizan tener disponibles en Internet una gran cantidad de clases y de documentación de las cuales extraer ideas y conocimientos. Se desarrolló una aplicación web implementada en Java respetando la especificación J2EE, lo que garantiza su gran portabilidad, escalabilidad y bajo costo de mantenimiento.

En cuanto al diseño se entendió que era importante realizarlo de una manera modular para facilitar el mantenimiento y adición de funcionalidades al sistema en el futuro. De esta forma se logró obtener un diseño que separara con claridad los diferentes bloques del sistema, permitiendo realizar la implementación de cada módulo en forma independiente de los demás.

Debido al atraso del proyecto se definió un plan de releases, desglosando la implementación del sistema en cuatro etapas. IBM estuvo de acuerdo que el prototipo a entregar fuera el resultado de la primera de ellas, brindando las funcionalidades más relevantes. La implementación se realizó en función de su importancia de forma de entregar cada caso de uso a medida que se desarrollaban. Esto permitió realizar los ajustes necesarios en función de las necesidades y grado de satisfacción de IBM, minimizando el trabajo que implicaron las mejoras planteadas a medida que se entregaban las nuevas versiones.

El tiempo disponible para la implementación del prototipo fue escaso, por este motivo se realizó un testeo mínimo relegando esta tarea a IBM. En este sentido IBM comenzó a utilizar la herramienta con dos objetivos,



testear con información real y a la vez aprovechar dicha información para la validación de las métricas seleccionadas. Los resultados obtenidos a partir de los indicadores definidos en el prototipo fueron de satisfacción para IBM ya que se aproximaron a la percepción subjetiva del sector durante dicho período. Cabe destacar que el volumen de información registrada por la herramienta no fue el más adecuado debido al corto período de tiempo disponible para esta etapa. Si bien los resultados obtenidos fueron satisfactorios es recomendable realizar una nueva evaluación de resultados en un período de tiempo más amplio obteniendo un volumen de información bastante más generoso.

Es importante resaltar la generalidad de la herramienta respecto a los indicadores, obtenida como resultado de la utilización de XML para su implementación. Los indicadores se pueden incorporar mediante su propia definición en archivos XML, incrementando considerablemente la flexibilidad de la herramienta en lo relativo a esta funcionalidad. Además permite crear indicadores específicos para cada proyecto, lo cual es importante para una organización como IBM, que brinda servicios a muchos clientes.

Con respecto a las interfaces de usuario que se realizaron, se consiguió un alto grado de amigabilidad. Esto se debe a que las mismas se basaron en la herramienta anterior y términos conocidos por el usuario, facilitando el ingreso de información ya que cuando es posible se genera automáticamente. La experiencia mostró que los usuarios de la herramienta anterior prácticamente no tuvieron la necesidad de leer un manual de usuario para poder utilizar el prototipo, el manejo de la misma les resultó intuitivo, sencillo y rápido por lo mencionado anteriormente. En este sentido se obtuvo un alto grado de satisfacción por parte del cliente.

Resumiendo, se obtuvo una herramienta que permite estimar de una forma más precisa la productividad del sector, siendo posible realizar un seguimiento más detallado de las actividades allí desarrolladas. En cuanto a la extensibilidad de la herramienta, no se limita a la incorporación de las funcionalidades contempladas en los releases definidos, sino que también abarca muchas otras funcionalidades, es decir, la herramienta se adapta perfectamente a los nuevos requerimientos que puedan surgir sin mayor esfuerzo. Esto fue logrado tanto por el diseño creado como por el lenguaje de programación utilizado.

Trabajos futuros

De los resultados obtenidos se abre un conjunto amplio de posibles trabajos a realizar. Hubiera sido deseable la implementación de todos los releases presentados en el punto 4.5. Cabe destacar que se realizó el diseño de la herramienta incluyendo el alcance de los mismos. Permitiendo así, la implementación e incorporación de los objetivos que plantean dichos releases, de una forma controlada y segura desde el punto de vista de su impacto en el sistema. El resultado que se obtenga de la extensión de la herramienta, convertirá a la misma en uno de los pilares fundamentales en lo referente a la gestión del centro de cómputos de IBM.

Hubiera sido deseable la implementación y/o diseño de algunos otros puntos que permitirán incrementar la eficiencia del sector y que no fueron posible realizar. En primer lugar, que la ejecución de los procesos se realice utilizando un scheduler estableciendo las acciones a realizar. Es decir, en función del manual de ejecución correspondiente a dicho proceso y la salida generada por la ejecución de uno de sus métodos, la herramienta establezca la directiva a seguir. De esta forma se facilitará la tarea de los operadores y se minimizará tiempo y errores en la operativa.

Otro punto que se puede atacar en un futuro es la interacción con el sistema de correo electrónico y/o la implementación de un módulo de notificaciones, para mantener informado a los usuarios sobre el estado de sus solicitudes y/o avisar de tareas pendientes. De esta forma, se notificaría al usuario correspondiente de dicha información; el módulo de noticias desplegaría la información pertinente en alguna sección de la herramienta con este fin. Por ejemplo, un usuario que ingresó una solicitud sería informado de sus cambios de estado, es decir cuando se confirme, agende y ejecute, o incluso si se cancelara dicha ejecución; el coordinador de desarrollo sería informado cuando un usuario ingresó una solicitud y la misma debe ser confirmada por él.



6. Referencias

6.1 Referencias de Internet

- [1] – Datasec Cobit (última visita, 30/08/2004)
<http://www.datasec.com.uy/index.php?option=content&task=view&id=21&Itemid=73>
- [2] – Isaca Stgio de Chile, Cobit 3rd Edition (última visita, 30/08/2004)
<http://www.isaca.cl/cobit.html>
- [3] – IBM - Internacional Business Machines Corporation (última visita, 10/10/2004)
<http://www-306.ibm.com/software/lotus>
- [4] – Geneura, Perl (última visita, 10/10/2004)
<http://www.geneura.ugr.es/>
- [5] – The J2EE (última visita, 30/10/2004)
<http://java.sun.com/j2ee>, <http://www.fing.edu.uy/inco/proyectos/linkall/seminarioj2ee.html>
- [6] – JSP, Servlets, J2EE (última visita, 10/04/2005)
<http://www.java.sun.com/j2ee/index.jsp>
- [7] – Javahispano Tutoriales (última visita, 15/04/2005)
<http://www.javahispano.org/>
- [8] – Java Community Process, Community Development of Java technology Specifications (última visita, 15/04/2005)
<http://jcp.org/en/introduction/overview>
- [9] – Java Boutique, Tutoriales (última visita, 15/04/2005)
<http://javaboutique.internet.com>



- [10] – Tecnologías de Internet y Tutoriales de Java, Javascript (última visita, 15/04/2005)
<http://www.abcdatos.com/>, <http://es.wikipedia.org/>, <http://www.aulambra.com/>, <http://www.htmlweb.net/>,
<http://www.javaworld.com/>
- [11] – Eclipse (última visita, 30/03/2005)
<http://www.eclipse.org>
- [12] – The Microsoft .NET (última visita, 10/10/2004)
<http://www.microsoft.com/net/>
- [13] – Microsoft .NET vs. J2EE: How Do They Stack Up? by Jim Farley 08/01/2000
(última visita, 30/10/2004)
http://java.oreilly.com/news/farley_0800.html
- [14] – JSP, Servlets, XML (última visita, 20/04/2005)
<http://www.servlets.com/>
- [15] – Tomcat (última visita, 30/03/2005)
<http://jakarta.apache.org>
- [16] – Apache (última visita, 30/03/2005)
<http://www.apache.org>
- [17] – Applying XML and Web Services Standards in Industry (última visita, 20/04/2005)
<http://www.xml.org/>
- [18] – XML (última visita, 20/04/2005)
<http://www.jcp.org/aboutJava/communityprocess/review/jsr063/jaxp-pd2.pdf>
- [19] – MQ Web site (última visita, 16/12/2004)
<http://www-4.ibm.com/software/ts/mqseries/messaging>
- [20] – MQ White papers & Specification Sheets (última visita, 16/12/2004)
<http://www-306.ibm.com/software/integration/wmq/library/index.html>



6.2 Referencias bibliográficas

[21] – James W. Cooper – Java Design Patterns – Primera Edición Enero 2000
Editorial Addison-Wesley

[22] – Erich Gamma, Richard Helm, Ralph Johnson, John Vlissides –Design Patterns – Edición 1995
Editorial Addison-Wesley

[23] – Erich Craig Larman – Applying UML and patterns – Edición 1997
Prentice Hall



Agradecimientos

Deseamos agradecer a las personas que nos ayudaron a realizar este proyecto:

- Edgardo Maissonave, de IBM, por su soporte en los temas de programación con las nuevas tecnologías y que lo llevaron a quedarse varias veces fuera de su horario de trabajo.
- Diego López, usuario responsable por parte de IBM, quien tuvo la idea original del proyecto, por su apoyo y amplia disponibilidad.
- Gustavo Vázquez, tutor del proyecto, por su aporte y amplia disponibilidad.
- Erica Rudolph, ex-integrante del grupo, por su dedicación y aporte en gran parte del proyecto.
- Antranik Hagopian, de IBM, por su soporte en los temas de configuración y administración de la base de datos DB2.
- y a todos los que de una manera u otra también colaboraron con la realización de este proyecto.