

ANEXO A. PROPUESTA DEL PROYECTO Plataforma de Inteligencia de Negocios

Planteamiento de la necesidad

Una gran cantidad de organizaciones, sobre todo las medianas y pequeñas, toman las decisiones de manera empírica, siendo una de las principales dificultades la ausencia de información a un nivel gerencial y estratégico. La baja calidad en las decisiones provoca pérdida de productividad y calidad en los procesos operativos y administrativos de la organización, haciéndolos más costosos e ineficientes.

Un ejemplo sencillo (y tomado de un caso real pero ligeramente modificado) es el de una empresa operadora de turismo que al hacer un análisis de sus servicios ofertados durante los cinco últimos años, descubre que en los meses m1 y m2 la demanda de adultos jóvenes que provienen de Europa por turismo de aventura es mayor que en el resto del año, y en cambio, los que provienen de USA lo hacen en los meses m6 y m7. La empresa, a partir del conocimiento obtenido, decide cambiar mejorar su estrategia de marketing para ofrecer los paquetes en los meses previos a m1 para Europa y en los meses previos a m6 para USA. Este cambio de estrategia le significó un ahorro significativo pues focalizó mejor las campañas para cada tipo de público.

Descripción General del Proyecto

La UNESCO declaró en el 2003 la necesidad de contribuir a la constitución de la Sociedad de la Información (UNESCO, 2003). En el Perú, se publicó en el 2005 el Plan de Desarrollo de la Sociedad de la Información, en donde se le da prioridad estratégica a este tema como pilar del desarrollo (Reyes, 2004). Para todos es aceptable que hoy se habla de la sociedad de la información y el conocimiento, y que la construcción de ese nuevo escenario viene soportado por el uso de diversas tecnologías de la información y la comunicación.

Las organizaciones en el Perú y el mundo, han venido usando diversos sistemas informáticos para registro de sus transacciones, que dan soporte a sus actividades diarias. Los datos recogidos por diversos procesos se consolidan en información que es útil para la toma de decisiones. El cuadro 1, que se muestra a continuación, es un extracto del análisis desarrollado por la empresa Apoyo Opinión y Mercado, para el año 2003, donde se muestra el conocimiento, uso e implementación de diversas tecnologías de la información en el Perú clasificadas por medianas y grandes empresas (Apoyo, 2003).

Se puede apreciar que las tecnologías de Bases de Datos y Herramientas de desarrollo son las más ampliamente usadas en ambos grupos, debido a que con ellas se suele construir las soluciones de sistemas de información necesarias para las empresas. Tecnologías como el ERP (*Enterprise Resource Planning*) y *Data Warehouse* son conocidas, pero tienen un menor nivel de aplicación en general; y en particular, a nivel de medianas organización, existe una demanda grande por atender. Finalmente, para el caso de tecnologías asociadas a soluciones

de CRM (*Customer Relationship Management*) o SCM (*Supply Chain Management*) son mucho menos usadas por su alta especialización y costo.

Software de	Conocimiento			Uso			Implementación		
	Total 2003	Entidad		Total 2003	Entidad		Total 2003	Entidad	
		Grande	Mediana		Grande	Mediana		Grande	Mediana
Base de Datos	94	100	92	90	96	88	37	27	40
Herramientas de Desarrollo	97	98	96	83	87	82	37	31	39
ERP		72	45	37	44	34	22	25	21
Datawarehouse	44	65	38	22	41	16	28	32	27
CRM	27	47	21	13	15	13	16	25	14
SCM	15	25	12	8	17	6	14	10	16
Antivirus	100	99	100	99	99	99	31	22	33

Cuadro 1. Conocimiento, Uso e Implementación de TI en el Perú (Apoyo, 2003).

La consultora internacional Gartner Dataquest realizó un pronóstico a cinco años, basado en una estimación preliminar del tamaño del mercado y una revisión de los inhibidores e impulsores, llegando a la conclusión de que el total del mercado de herramientas de BI proyecta un crecimiento de \$ 2.5 billones en 2004 a \$ 2.9 billones en 2009, con una tasa de crecimiento anual de 7.4% (Medina, 2004). Según un reporte electrónico se presume que el mercado de soluciones de *Business Intelligence* en la región Asia/Pacífico (sin contar al Japón) crecerá de US\$529.8 millones en el 2003 a US\$1.2 mil millones en el 2008 (BI-New, 2005). Durante el año 2003, en el Brasil, la inversión en servicios de consultoría de tecnologías de información relacionados a proyectos se incrementó; lo cual se muestra en el siguiente gráfico, donde se aprecia una mayor inversión en proyectos de *Business Intelligence* (IDC, 2004).



Gráfico 1: Servicios de consultoría de tecnologías de información

Considerando las tendencias antes presentadas a nivel nacional e internacional, se puede decir que hoy en día se hace muy necesario ir más allá de la recopilación de datos y se requiere hacer un tratamiento más elaborado de los mismos, complejas transformaciones de información y un análisis más profundo para obtener un mejor conocimiento de los hechos y a partir de ellos, tomar mejores decisiones.

Aplicación de Inteligencia de Negocios

Las empresas han sido las primeras en aplicar esta tecnología para lograr un mayor conocimiento de sus operaciones y sacar ventaja de ello. La información obtenida de manera inteligente es considerada hoy una ventaja competitiva y son cada vez más organizaciones las que buscan un mejor acercamiento a sus clientes individuales u organizaciones y han hecho de la Inteligencia de Negocios (BI de las siglas en inglés de *Business Intelligence*) una prioridad de inversión. Con esta tecnología se puede obtener análisis de mercados verticales *end-to-end*, soluciones del pronóstico y optimización del mercado; entre una gama muy amplia de posibilidades.

En el cuadro 2 se aprecia que ya existe en el mercado un uso de esta tecnología en empresas que son grandes, sin embargo para medianas y pequeñas organizaciones es restringido debido al alto costo.

Giro / Empresa	Análisis de Inteligencia de Negocios Aplicados
Comercializadora Ewong	Ventas específicas por Cliente Servicio al cliente (preferencias, alejamiento de clientes) Reclamos a proveedores
Entretenimiento Cine Planet	Utilización efectiva de su capacidad instalada. Programación de pedidos a proveedores. Fidelización de clientes (hábitos de clientes)
Seguros La Positiva	Características de siniestralidad (pólizas) Estrategias definidas por productos. Renovación de pólizas
Hoteles Casa Andina	Análisis de ventas (%ocupación, hotel-temporada) Análisis de reservas (razones de cancelación, porcentaje de cancelación por temporada). Características de clientes (nacionalidad, sexo, edad, etc).
Gobierno SUNAT	Cruce de información de contribuyentes.

Cuadro 2: Aplicaciones de Inteligencia de Negocios en el Perú

Definición

La inteligencia de negocios implica establecer una forma de hacer las cosas y de ser disciplinado para obtener el máximo provecho. Citando a CherryTree & Co se presenta una definición como sigue:

“Las aplicaciones de Business Intelligence (BI) son herramientas de soporte de decisiones que permiten en tiempo real, acceso interactivo, análisis y manipulación de información crítica para la empresa. Estas aplicaciones proporcionan a los usuarios un mayor entendimiento que les permite identificar las oportunidades y los problemas de los negocios. Los usuarios son

capaces de acceder y apalancar una vasta cantidad de información y analizar sus relaciones y entender las tendencias que últimamente están apoyando las decisiones de los negocios. Estas herramientas ayudan a que las organizaciones no pierdan información valiosa (conocimiento de la organización) que se va acumulando con el transcurrir de las actividades y que no resulta de fácil lectura” (Medina, 2005).

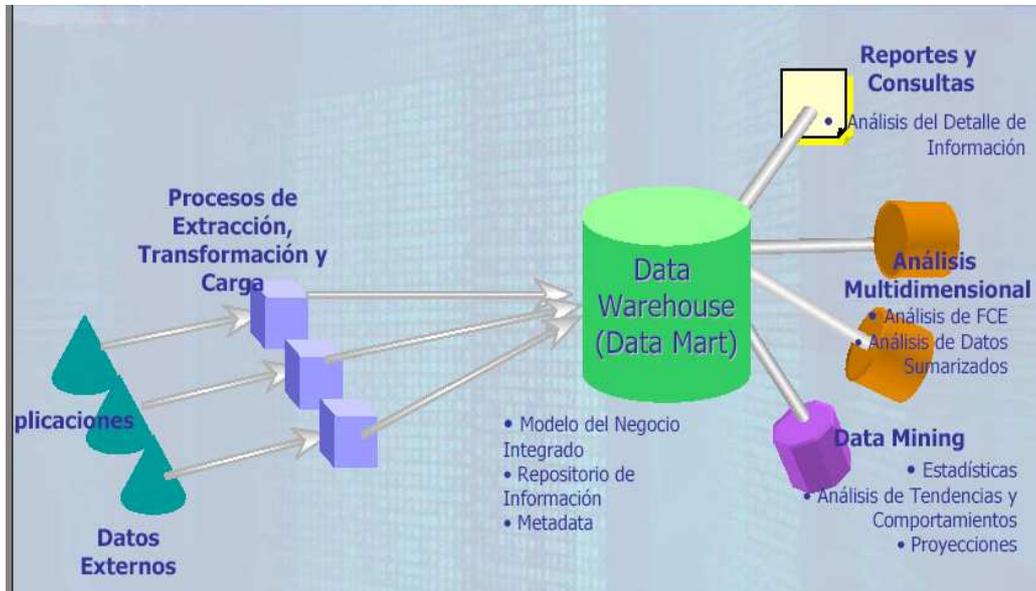


Gráfico 2: Modelo de aplicación de Inteligencia de Negocios (Medina, 2004).

Descripción Actual del Mercado

Para la implementación de un proyecto de Inteligencia de Negocios, se utilizan varios tipos de software que realizan una o varias etapas del proyecto, pero no integran todas ellas. Entre estos productos podemos destacar:

Producto	Descripción
Cognos:	Posee una suite de productos que permiten realizar la explotación de datos y la extracción
Business Object:	Posee una suite de productos que permite realizar la explotación y extracción de datos
Proclarity:	Producto que permite realizar la explotación de datos
Microsoft	Posee una suite de productos que permite extraer y explotar datos

En una implementación de Inteligencia de Negocios, la interacción con el usuario final es importante ya que es él quien, en base al modelo de negocios, va a analizar la información. Por lo que el analista, en la etapa de modelado, necesita contar con un prototipo que permita conocer el resultado final del modelo de negocios analizado. En el mercado sólo existen generadores de datos; pero no integrados a todas las etapas del proyecto.

Un proyecto de Data Warehouse necesita poder llegar a integrar todos los modelos de negocios de cada área de una empresa (Data Marts) para poder realizar la toma de decisiones en base a la información global del negocio.

Objetivo del Proyecto

1. Construir una plataforma informática para analistas de Inteligencia de Negocios, basados en un modelo de ventas de servicios de consultoría y procesamiento en Inteligencia de Negocios.
2. Implantar la plataforma en dos organizaciones de tamaño pequeño y mediano para evaluar el desempeño de la plataforma.

Componentes y Características del Producto: El producto final del proyecto es un software integrado que a nivel funcional cuenta con los siguientes Componentes:

1. **Modelador Dimensional**, que permite a un consultor de Data Warehouse poder modelar un tema de negocios e interactuar directamente con un usuario final mostrándole el resultado como prototipo.
2. **Extractor de Datos**, que permite al usuario del sistema, poder esquematizar y realizar el proceso de carga de datos desde la fuente misma hacia el repositorio final en base al resultados del modelador dimensional.
3. **Explotador de Datos**, permite al usuario final poder generar sus propios reportes en base a los datos obtenidos por el Extractor de Datos.

El software presentará a nivel técnico las siguientes características:

1. **Modular**, es decir, que cubre todas las etapas para el desarrollo de un proyecto de Data Warehouse.
2. **Escalable**, es decir, que permite modelar diversos temas de negocios (Data Marts) para así integrar todo el modelo de la organización (Data Warehouse). Práctica recomendada por Ralph Kimball.
3. **Multiplataformas**, es decir, un producto desarrollado con tecnologías que permita su uso sobre las plataformas de sistemas operativos más ampliamente usados en nuestro medio como son Windows y Unix.

Requisitos Funcionales del Producto: Los requisitos de alto nivel que cubrirá el software desarrollado en el proyecto son los siguientes:

Modelado Dimensional: es una actividad donde se realiza la representación de distintas entidades (de datos) que permiten un análisis de una determinada situación de la organización.

Generación del Modelo Dimensional

1. Administración de temas de análisis (Crear, modificar y eliminar)

2. Control el acceso de los usuarios por tema de análisis
3. Generación Modelo Dimensional (Estrella) de forma visual.
4. Administración de componentes del diseño dimensional (crear, modificar y eliminar): Dimensiones, Jerarquías y Atributos; Facts y Medidas.
5. Búsqueda de componentes del modelo (dimensiones, facts, atributos y medidas) para evitar generar entidades iguales
6. Reutilización de componentes en diferentes temas de análisis
7. Copiado de componentes en diferentes temas de análisis
8. Asistencia en la generación del modelo dimensional
9. Asignación el tipo de dato por cada atributo de una dimensión y las medidas de cada Fact
10. Visualización y generación de scripts de creación de tablas de diferentes manejadores de bases de datos
11. Bloque de un modelo dimensional en caso éste sea modificado por otro usuario
12. Visualización de mensajes de error en caso que el modelo dimensional no sea adecuado

Generación MetaData

1. Modificación y generación la metadata de los modelos en formato Excel, HTML y Texto
2. Visualización del documento preliminar de la metadata

Simulación de Datos

1. Configuración de reglas de generación de datos de un modelo dimensional indicado
2. Generación de datos a partir de las reglas de un modelo dimensional indicado
3. Almacenamiento de los datos generados
4. Visualización de los datos generados en el módulo de explotación de datos

Extracción

1. Administración de los proyectos y paquetes de extracción
2. Control el acceso de los usuarios por proyectos y paquetes
3. Creación de un paquete de extracción de forma visual, drag & drop de componentes desde la paleta de componentes
4. Control del tamaño de la vista del área de trabajo (zoom)
5. Administración de varias ventanas por área de trabajo
6. Presentación gráfica del flujo de datos y el progreso del proceso de extracción de un paquete
7. Almacenamiento de la historia del estado de los procesos de extracción ejecutados
8. Conexión de varias bases de datos como fuente de datos y repositorio de destino
9. Control de los accesos de la fuente de datos y el destino del repositorio de datos

10. Generación de tablas temporales antes de enviar los datos definitivos al repositorio destino
11. Limpieza y estandarización de de datos como paso previo para la carga de datos
12. Transformación mediante operaciones definidas por el sistema o por una operación personalizada con lenguaje SQL Standar
13. Visualización previa de los datos de origen de acuerdo a la extracción definida
14. Visualización previa de los datos a cargar de acuerdo a las reglas de transformación definida
15. Personalización de un log de errores por cada componente de extracción
16. Programación de las tareas de extracción en un calendario
17. Reubicación de las tareas en un calendario y cancelación de las ya programadas
18. Visualizar el estado de las tareas programadas
19. Presentación de reportes estadísticos sobre los tiempos de ejecución por tarea programada.

Explotación

1. Administración de los proyectos de explotación
2. Administración los Cubos Olap que forman parte de los proyectos
3. Administración los componentes de un cubo: Dimensiones, Jerarquías, Atributos, Medidas, Indicadores y Filtros.
4. Generación cubos a partir de la configuración de sus componentes
5. Creación reportes tabulares y/o gráficos a partir del cubo generado. Se puede utilizar Drg-Drop para la creación de los reportes
6. Ejecución de Drill-Down y Drill-Up a través de los informes generados
7. Control del acceso de los usuarios por proyectos, cubos e informes
8. Exportación de los datos a MS Excel, HTML y XML
9. Búsqueda de los componentes de los cubos configurados

Seguridad

1. Administración de Usuarios (Crear, modificar y eliminar)
2. Administración de Perfiles por opción del menú
3. Administración de contraseñas.

REFERENCIA BIBLIOGRAFICA.

- UNESCO, Annan Calls On Broadcasters to Help Create 'Open Inclusive' Information Society http://portal.unesco.org/ci/en/ev.php-URL_ID=13801&URL_DO=DO_TOPIC&URL_SECTION=201.html 2003.
- Apoyo Opinión y Mercado. Mercado Informático y Tecnología de Información, 2003.

- Dávila Abraham, Melgar Andrés, Modelado de la Metadata para el Desarrollo de Herramientas de Productividad sobre Múltiples Manejadores de Bases de Datos Relacionales. JIISIC04 (4ª Jornadas Iberoamericanas de Ingeniería del Software e Ing. del Conocimiento, España, 2004
- Reyes J., Plan de Desarrollo de la Sociedad de la Información en el Perú. LA agenda digital. Presidencia del Consejo de Ministros. CODESI: Comisión Multisectorial para el Desarrollo de la Sociedad de la Información, 2005.
- Medina Edison, Business Intelligence Un vistazo a la realidad peruana. Presentación COMMON DAY, 2004.
- Medina Jorge, Business Intelligence: Conceptos y Actualidad. 2005 <http://www.gestiopolis.com/recursos5/docs/ger/buconce.htm>.
- IDC, It Solution Market 2004, Agosto 2004.
- BI-New, BI booming in Asia / Pacific. 2005. <http://www.bi-news.com/>

Preparado por:

Abraham Dávila

Carla Basurto



ANEXO B. LISTA DE EXIGENCIAS

1. Lista de Exigencias de Proyecto AXEBit

N°	Características / Funcionalidades
No Funcional	
1.	Manejará los RBBMS: Sql Server, Oracle y MySQL.
2.	Tiempo de respuesta de búsquedas menor a 2 minutos.
3.	Repositorio de objetos será en XML.
4.	Herramienta WEB.
5.	Se desarrollará usando Eclipse.
6.	Se desarrollará usando Linux y soportar otros.
7.	Autosave por parámetro.
8.	Transacciones seguras.
9.	Confirmación de acciones en una barra de estado.
10.	Manual de usuario en web.
11.	Manual de instalación y configuración.
12.	Permite el manejo de drag&drop.
13.	Permite la configuración de apariencia gráfica de cada elemento o tipo de elemento (color de fondo, letra, tamaño, tipo de línea).
14.	Permite el zoom +/- y zap del modelo.
15.	Permite la configuración de reportes (fuentes, colores, gráficos, color de fondo, etc).
16.	Permite la generación de pistas (logs) de auditoria de los procesos.
17.	Permite la generación de pistas (logs) de errores de los procesos.
Análisis - Modelador	
1.	Entorno gráfico para el diseño del modelo estrella.
2.	Permite el manejo de elementos de manera gráfica.
3.	Permite la administración de elementos del modelo: temas de análisis, facts, dimensiones, atributos, jerarquías.
4.	Permite la administración de elementos del modelo: medidas, indicadores y relaciones.
5.	Permite el modelado físico y lógico
6.	Permite organizar el modelo según objetivos-temas-facts-dimensiones.
7.	Permite organizar el modelo según temas-facts-dimensiones.
8.	Permite organizar el modelo según facts-dimensiones.
9.	Permite organizar el modelo según temas-objetivos// facts-dimensiones.
10.	Permite organizar el modelo según temas-dimensiones.
11.	Permite definir una dimensión como histórica (subrogate-key).
12.	Permite el manejo de periodicidad de carga de una fact (política de actualización de fact).
13.	Entorno gráfico para el diseño del modelo snow-flake.
14.	Alarma para señalar uso de malas prácticas (por identificar).
	- mensaje por no tener una dimensión tiempo
	- mensaje por una dimensión que tenga más de una llave
15.	Soporte para la personalización de identificadores dentro del modelo: pre-fijo, postfijo para diversos elementos (reglas de identificadores)
16.	Permite identificar las medidas como aditivas, semi-aditivas y no-aditivas.
17.	Permite la importación de elementos entre modelos.
18.	Permite la copia de elementos (para modificación) de otros modelos (incluso diccionario).
19.	Permite la administración de formatos de datos para cada elemento.

20.	Permite la asignación de formatos de datos para cada elemento.
21.	Maneja todos los tipos de datos de acuerdo al RDBMS seleccionado.
22.	Permite registro de descripciones y notas sobre los elementos.
23.	Permite el bloqueo de un modelo dimensional en caso esté siendo modificado por otro usuario.
24.	Permite el manejo de plantillas de temas.
25.	Permite el manejo de plantillas de modelos.
26.	Generación de scripts para la creación de las tablas del modelo.
27.	Generación directa de las tablas del modelo en la base de datos.
28.	Visualización de scripts para la creación de las tablas del modelo.
29.	Almacenamiento de scripts para la creación de las tablas del modelo.
30.	Ingeniería reversa de la definición de las tablas.
31.	Exportar el modelo a un formato gráfico (jpeg, tiif, pdf).
32.	Visualización icónica del modelo.
33.	Identificación de usuario que tiene en uso un elemento.
Análisis - Metadata	
1.	Almacenamiento de la data (modelo) de los elementos (guardar)
2.	Permite que cada objeto tenga su diccionario de datos.
3.	Permite realizar búsquedas para todos los objetos (elementos): temas de análisis, facts, dimensiones, atributos, jerarquías.
4.	Permite realizar búsquedas para todos los objetos (elementos): medidas, indicadores y relaciones.
5.	Permite realizar búsquedas para los objetos: proyecto, reglas de extracción o cubos.
6.	Permite realizar búsqueda sobre objetos con patrones de cadenas en su significado o identificador
7.	Visualización gráfica de la dependencia de los objetos: - a partir de un objeto seleccionado. - a partir de un tipo de objeto.
8.	Permite registro de descripciones y notas sobre los elementos.
9.	Permite la generación de reportes a partir de la metadata (ver power designer).
10.	Permite la definición de TODOS los objetos del sistema (incluye la importación y definición de orígenes).
Análisis - Generador	
1.	Desarrollo basado en DatProducer (ya tiene desarrollado varias cosas, se debe pasar a web).
2.	Permite la configuración de reglas de formación de la data (GUI Web).
3.	Permite la generación de datos para RDBMS: MSSqlServer, Oracle, MySQL (en realidad falta MySQL).
4.	Permite el manejo de archivos de texto como base para generar datos.
5.	Permite el uso de datos generados previamente en bases de datos.
6.	Permite el manejo de reglas de integridad (ya tiene parcialmente).
7.	Permite la vista preliminar de la data parcial.
Análisis - Otros	
1.	Permite la estimación del tamaño del DataWarehouse (en base al modelo físico de acuerdo al RDBMS).
2.	Permite el manejo de privilegios por objetos.
3.	Permite el manejo de privilegios por empresas.
4.	Permite el manejo de privilegios por proyectos.
5.	Permite la administración de múltiples empresas.

6.	Permite la administración de múltiples proyectos por empresa.
7.	Permite la administración de usuarios por empresa (nota una persona puede estar en varias empresas y solo tener un id).
8.	Permite la administración de perfiles.
9.	Permite la asignación de usuarios a proyectos.
10.	Permite la administración de privilegios del usuario por objetos.
11.	Administración de pista de auditoria (log).
12.	Visualización de pista de auditoria.
13.	Administración de pista de errores (guardar, borrar).
14.	Visualización de pista de errores.
15.	Registro de esfuerzo por modelo.
16.	Permite definir la regla de limpieza de datos por cada objeto del modelo.
Extracción - Comunicación	
1.	Permite la comunicación con fuentes: Oracle, MSSql Server, Textos, XML, hojas de cálculo, MySQL, Archivo Hash.
2.	Permite la visualización de objetos desde las fuentes de datos: - tablas, campos, características de los campos (longitud, escala, tipo de datos, nulos, si es PK), atributos.
3.	Utiliza un parser para la comprobación de un script (de comandos).
4.	Ingeniería Reversa de la definición de tablas del modelo relacional.
5.	Agrupación de los programas de extracción por temas (jobs).
6.	Llamada de programas de extracción desde un programa de extracción.
Extracción - Extrae	
1.	Extracción de datos considerando las fuentes y los objetos para la consolidación.
2.	Permite el manejo de objetos de Extracción, Transformación y Carga.
3.	- Objetos pasivos (fuentes y destinos de información, stored procedure, ODBC, FTP, webservices, xml, etc.).
4.	- Objetos activos:
5.	- objeto que pueda extraer, transformar y cargar (genérico), incorpora el lookup.
6.	- objeto especializado Merge
7.	- objeto especializado Sum
8.	- objeto especializado Lookup
9.	- objeto especializado Transpuesta
10.	- objeto especializado Sort
11.	- objeto especializado Ejecutar comando (interno y del sistema operativo)
12.	- objeto de operación para mandar mensajes (alarmas) a usuarios específicos
13.	- objeto de operación para sincronizar procesos.
14.	El diseño de la extracción y la configuración de los objetos será en forma gráfica
15.	Ver modelador para la parte gráfica y de seguridad
16.	Permite el manejo de objetos de limpieza de datos durante la definición de los otros objetos activos
17.	Permite el manejo de reglas de limpieza como lista de comprobación.
18.	Permite la definición de pistas (log) de procesamiento y rechazos
19.	Permite generar log de eventos y errores de apoyo a la extracción y transformación.
20.	Permite generar log de eventos y errores de apoyo a la extracción y transformación.
Extracción - Ejecución	
1.	Integración: programación (schedule) y monitoreo.
2.	Permite programar la ejecución de los programas de carga de datos.
3.	Permite conocer los tiempos de ejecución promedio de cada programa.

4.	Permite visualizar el monitoreo de cada ejecución con sus tiempos de carga y número de filas por segundo por cada objeto activo.
5.	Permite visualizar el monitoreo de recursos de hardware y enviar alarmas.
Explotación - Cubos	
1.	Permite sugerir cubos a partir de las facts, agrupados por temas.
2.	Permite diseñar cubos en base a las dimensiones y jerarquías definidas en el análisis.
3.	Permite la creación de dimensiones, jerarquías, medidas calculadas definidas en el análisis.
Explotación - Reportes	
1.	Permite la generación de reportes gráficos y tabulares pre-definidos - pastel, líneas, barras, pareto
2.	Permite la generación de reportes gráficos y tabulares definido por el usuario
3.	Permite la administración de accesos para la visualización de reportes.
4.	Permite la definición de filtro.
5.	Permite la definición de ordenamiento, ranking y excepciones, los primeros, los últimos y entre dos o más posiciones.
6.	Permite el cálculo de y presentación de promedios, mínimos, máximos, curva de ajuste (mínimos cuadrados)
7.	Permite presentar gráficamente el cálculo de promedios, mínimos, máximos, curva de ajuste (mínimos cuadrados)
8.	Permite crear totales y subtotales de los reportes como: sumas, contadores, promedios, mínimos, máximos.
9.	Permitir exportar reportes bajo formato XML, HTML, pdf, hoja de cálculo electrónico, cvs, imágenes (giff, jpeg, bmp).
10.	Permite la visualización mdx del reporte.
Explotación - Integrado	
1.	Permite la generación de un cubo a partir de los datos de generación de data de análisis
2.	Permite presentar gráficamente el cálculo de promedios, mínimos, máximos, curva de ajuste (mínimos cuadrados)
3.	Permite programar la generación de cubos para el análisis.
Explotación - What-if	
1.	Permite la creación de reportes gráficos de proyección.
2.	Permite el análisis de sensibilidad a futuro, cambiando valores del presente.

ANEXO C - ESPECIFICACIÓN DE REQUISITOS DE SOFTWARE

1. Introducción

El presente documento, “Especificación de Requisitos de software”, contiene el total de requisitos encontrados para la elaboración de la herramienta AXEbit para el módulo de Extracción diagramado mediante Casos de uso y especificados para su mejor entendimiento. Este documento ha sido elaborado siguiendo las normas recomendadas en el documento “IEEE Recommended Practice for Software Requirements Specification IEEE Std 830-1998”.

1.1. Propósito

El objeto del presente documento es definir las funcionalidades y restricciones de la herramienta que se desea construir.

Se busca especificar de manera formal todas las funcionalidades y restricciones del sistema a construir, para evitar así dudas o confusiones que puedan existir por parte del cliente final o en el equipo desarrollador sobre alguna funcionalidad de la aplicación.

1.2. Ámbito del Sistema

El futuro sistema recibirá el nombre de AXEbit.

Este proyecto ha sido propuesto con la finalidad de implementar una solución que permita diseñar flujos de transformación de datos para el manejo y desarrollo de un data warehouse.

La motivación que impulsa el desarrollo del sistema es el de implementar una herramienta de Inteligencia de Negocios que facilite y mejore el acceso a la información, brindando este servicio desde una central en la cual se ejecutarán los procesos de transformación de datos, y que permitirá la conexión remota de clientes los cuales podrán realizar las configuraciones necesarias utilizando su navegador web.

La herramienta en general permitirá manejar de manera flexible todas las operaciones relacionadas a la visualización y acceso a la información.

1.3. Definiciones, Acrónimos y Abreviaturas

1.3.1. Definiciones

Dimensión	Es la descripción de una dimensión de la base de datos. Contiene la información de sus campos (metadata)
Miembro	Suele representar cualquier campo de una dimensión. Dentro de los miembros podemos encontrar : ID: se refiere al campo que será utilizado para hacer las consultas dentro del sistema. Descripción: es el campo que servirá para ser mostrado en los informes o gráficos. Alias: Es el nombre que el miembro tiene dentro de un proyecto.
Jerarquía	Agrupar a miembros que pertenecen a una o más dimensiones, y permite la navegabilidad en el informe.
Hechos	Son los campos de la fact.
Indicador	Elemento de medida definido por los hechos. Se expresa mediante fórmulas matemáticas que tienen como operandos a los campos de la fact.
Filtro	Se distinguen los tipos de filtro: Por rangos: Los que se encuentran en el rango por valor numérico o cadena. Los que no se encuentran en el rango por valor numérico o cadena. Posición : Los n primeros Los n últimos Los que no son n primeros Los que no son n últimos Se encuentra en Lista : Solo los que pertenecen a la lista No se encuentra en la Lista Compuesto: es la agrupación de uno o más filtros simples o compuestos.
Informe	En la agrupación de los siguientes elementos: Miembros Indicadores Filtros Jerarquías Muestra la información que el usuario solicita.
Proyecto	Es la agrupación de uno o más temas de análisis.
Rol de Sistema	Es un conjunto de privilegios que competen al sistema. Los roles establecen diferentes niveles de acceso a la aplicación.
Rol de Proyecto	Es un conjunto de privilegios que competen al proyecto.
Perfil	Es el conjunto de uno o más roles del sistema y algún rol de proyecto asociado a un proyecto.
Usuario	Es la persona que usa la aplicación. Un usuario puede tener uno o más perfiles.
Gráfico	Contiene los mismos elementos que un Informe, sin embargo se presenta visualmente distinto.
Objetos del Sistema	Son todos los Miembros, Filtros, Indicadores, Hechos, Dimensiones, Roles, Perfil, Jerarquías, Informes y Gráficos, definidos en este documento.
Objetos del Informe	Son todos los Miembros, Filtros, Indicadores y Jerarquías definidos en este documento.

1.3.2. Acrónimos

ERS	Especificación de Requisitos de Software
-----	--

1.4. Referencias

- IEEE Recommended Practice for Software Requirements Specification IEEE Std 830-1998.

1.5. Visión General de la ERS

Este documento consta de 3 partes. Esta primera parte introductoria da una visión general de lo que se quiere lograr con el uso de la ERS.

La segunda parte hace una descripción general de las funciones que realizará la herramienta sin entrar en mucho detalle, pues sólo incluye las principales tareas y se muestran las relaciones y dependencias que puedan existir entre los módulos más importantes. Finalmente, en la tercera parte se detallará paso a paso como debe funcionar la herramienta, desarrollando todos los casos de uso con el fin de cubrir con todos los requerimientos.

2. Descripción General

Esta herramienta deberá permitir que los sistemas de almacenamiento de datos (Data Warehouses) se puedan diseñar, crear, poblar y probar de un modo sencillo para el usuario.

En esta sección se presenta una descripción a alto nivel del sistema para la herramienta AXEbit. Se presentará el modelo de casos de uso (modelo que muestra la funcionalidad de la herramienta), las características de los usuarios, las suposiciones y dependencias del sistema.

Para poder identificar los tipos de usuarios a la herramienta, estos serán divididos de acuerdo a las funcionalidades que pueden utilizar en la misma. Adicionalmente se tendrá un módulo de seguridad que permitirá personalizar, en caso lo requieran los clientes, los permisos por actor.

2.1. Modelo de Casos de Uso

En esta sección se presentan los diagramas de casos de uso de la herramienta obtenidos durante el proceso de especificación de requisitos, los cuales permiten mostrar a alto nivel las funcionalidades que la herramienta realizará.

Inicialmente, se indica el catálogo de actores que interactúan con el sistema y posteriormente la descripción de cada uno de los paquetes con sus respectivos diagramas de casos de uso.

2.1.1. Catálogo de Actores

Administrador Sistema

El administrador del sistema puede registrar nuevos usuarios en el sistema, configurar los perfiles de usuario y asignarles perfiles a los usuarios dentro del ámbito del sistema. Además puede configurar conexiones a Base de Datos.

Analista

Persona encargada de crear, modificar y ejecutar paquetes de transformación de datos usando el software.

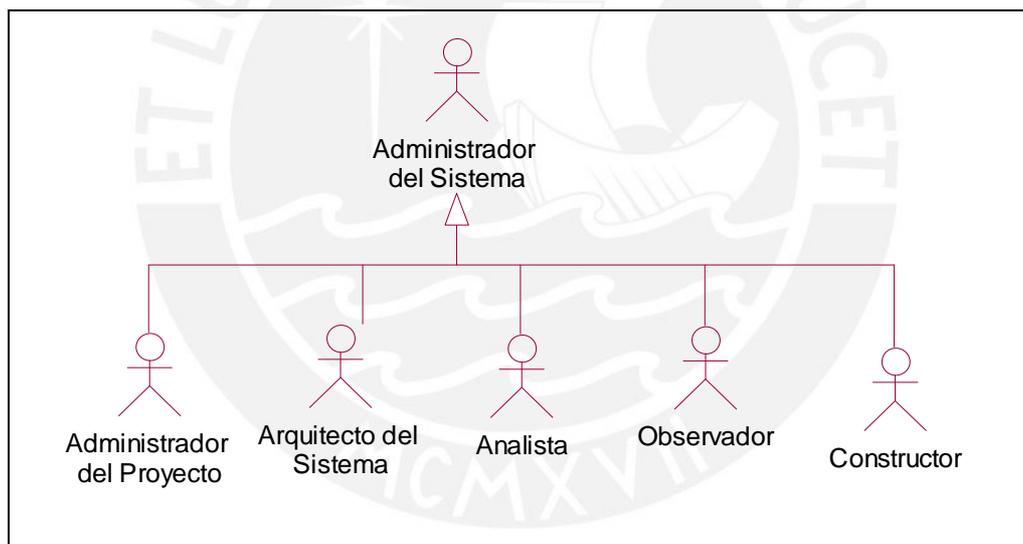


Ilustración 1: Catálogo de actores

2.1.2. Casos de Uso por Módulo

Módulo Seguridad

Paquete Seguridad

El paquete de seguridad es único para el módulo de seguridad, el cual interactuará con los tres módulos restantes (Análisis, Extracción y Explotación) y sus funcionalidades respectivas con los actores del sistema.

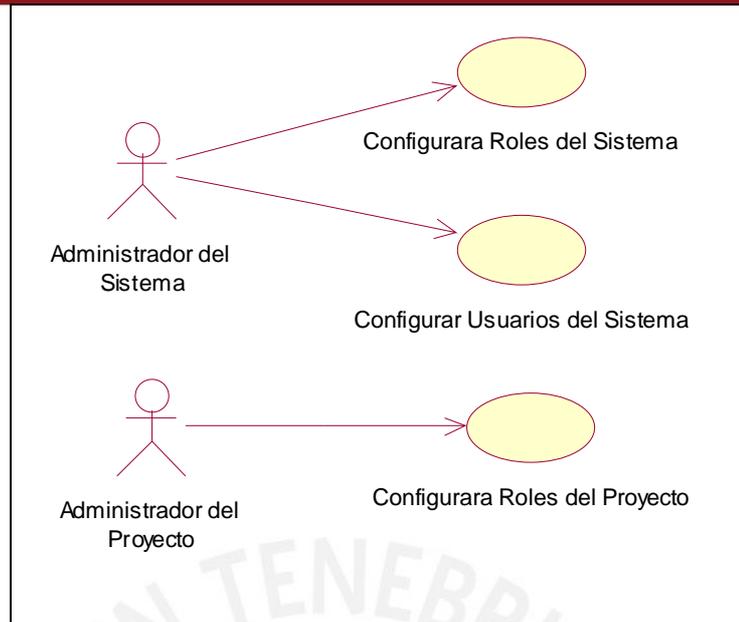


Ilustración 2: Casos de uso del paquete seguridad

Módulo Extracción

Paquete Comunicación

Contiene la las operaciones relacionadas con la comunicación entre la herramienta AXEbit, módulo de extracción y las fuentes de datos. Se inicia con la definición de parámetros de fuente de datos, tales como el manejador de base de datos, servidor, usuario, contraseña y otros. Después de tener estos datos, la comunicación podrá darse sin problemas y los objetos de las fuentes de datos podrán ser utilizados para la construcción del flujo de transformación.

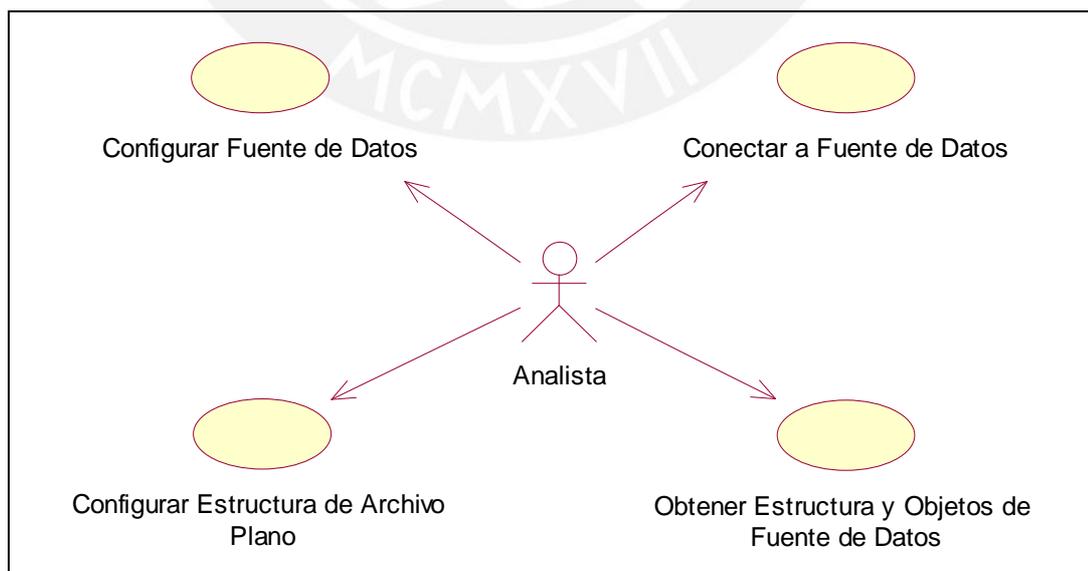


Ilustración 3: Casos de uso del paquete comunicación

Paquete Extracción

El paquete de extracción contiene los casos de uso más importantes del módulo, ya que son los que permiten definir los flujos de transformación, filtros de limpieza y personalizaciones de las operaciones de extracción y transformación de datos.

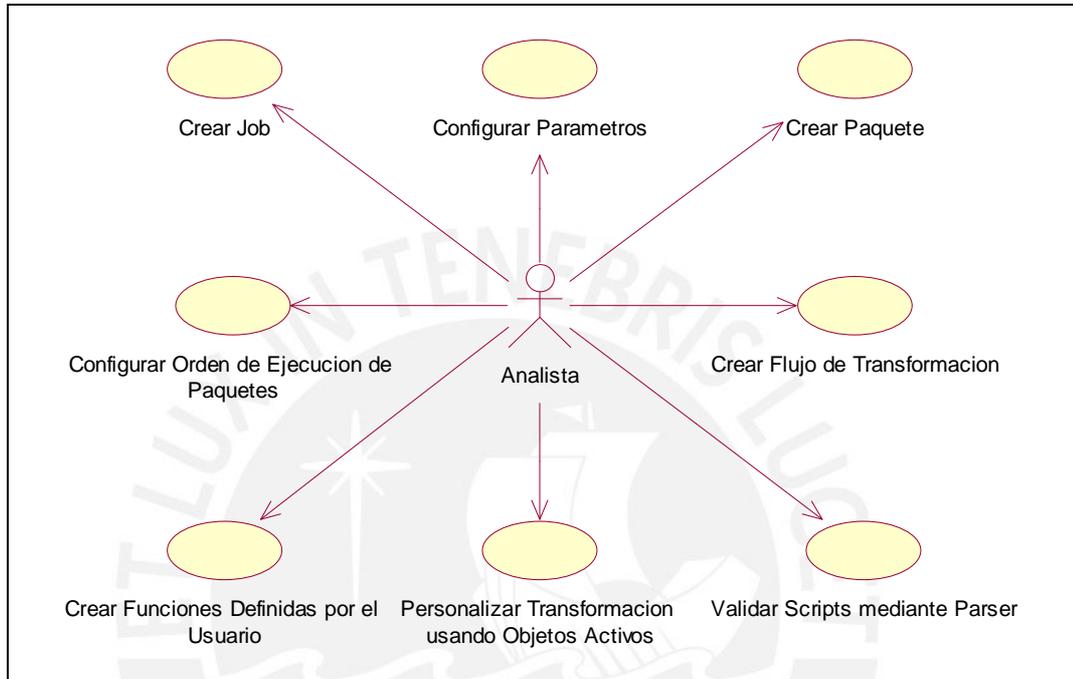


Ilustración 4: Casos de uso del paquete extracción

Paquete Ejecución

Paquete que contiene los casos de uso de control de ejecución de la extracción de datos.

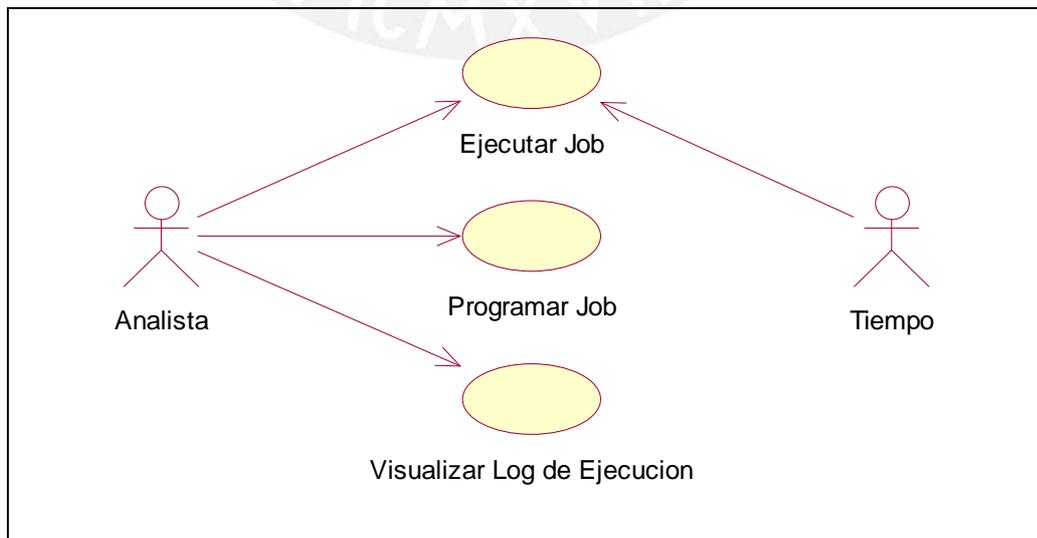


Ilustración 5: Casos de uso del paquete ejecución

2.2. Características de los Usuarios

Administrador del Sistema

El administrador del sistema será un usuario con conocimientos técnicos, por lo tanto deberá tener avanzados conocimientos sobre computación. Éste será el encargado de administrar a los demás usuarios, así como configurar sus accesos al sistema. Además podrá configurar las conexiones al repositorio de datos.

Analista

En este grupo estarán los encargados de crear, modificar y ejecutar paquetes de transformación de datos usando el software. Debe ser una persona que tenga el conocimiento técnico adecuado pero que también conozca de la lógica del negocio para poder obtener los resultados esperados.

2.3. Suposiciones y Dependencias

2.3.1. Suposiciones y Dependencias

- Se asume que se cuenta con las licencias necesarias para la respectiva instalación de la herramienta.
- Se asumirá que el servidor permanecerá en funcionamiento durante el uso de la herramienta.
- Cualquier petición de cambios en la especificación debe ser aprobada por todas las partes.
- Se asume que los requisitos descritos en este documento son estables una vez que sea aprobado.
- La herramienta AXEbit depende del motor OLAP y del servidor de aplicaciones Tomcat.
- La herramienta se desarrollará utilizando una arquitectura web, accediendo a nuestro servidor mediante la interfaz JOLAP

3. Requisitos Específicos

Esta sección contiene a nivel detallado la funcionalidad y restricciones del sistema. El detalle de la funcionalidad se determina con la especificación de casos de uso y las restricciones a la funcionalidad se especifican con los requisitos de interfaces externos, los requisitos de rendimiento, los requisitos de desarrollo y los miembros de la herramienta.

3.1. Especificación de Casos de Uso – Extracción

3.1.1. Paquete Comunicación

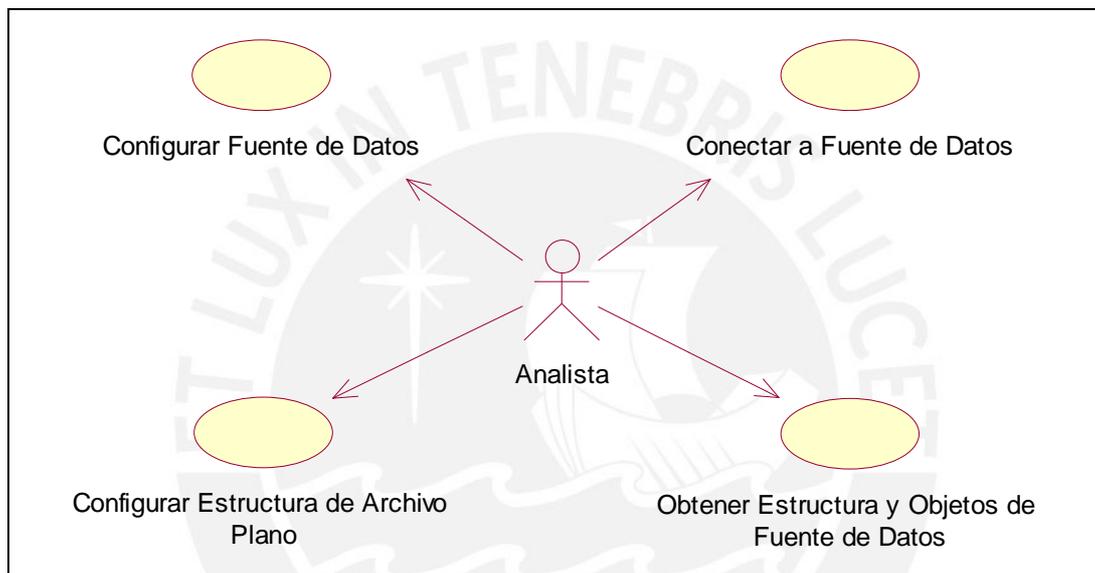


Ilustración 6: Casos de uso del paquete comunicación

3.1.1.1. Caso de Uso Configurar Fuente de Datos.

El propósito de este caso de uso es el de permitir al usuario configurar una fuente de datos.

1. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

1.1. Flujo Básico

El caso de uso se inicia con el usuario.

1. El caso de uso se inicia cuando el usuario selecciona "Fuente de datos".
2. El usuario elige "Nueva".
3. El sistema muestra los tipos de fuentes de datos a las cuales se puede conectar.

4. El usuario selecciona “Conexión a Base de datos”
5. El sistema muestra las bases de datos a las que se puede conectar: Oracle, MS Sql Server, MySQL
6. El usuario selecciona el tipo de base de datos.
7. El sistema muestra una pantalla para configurar los datos necesarios para la conexión a la BD: Nombre del servidor, nombre de la BD, usuario, contraseña.
8. El usuario ingresa los datos solicitados.
9. El sistema guarda los datos de la configuración.
10. Los pasos 3 a 9 se repiten para todas las fuentes de datos que el usuario desee tener.
11. El usuario selecciona “Salir” y el caso de uso finaliza.

1.2. Flujos Alternativos

i. Conexión a archivos locales

1. Este flujo alternativo se realiza en el paso 4 del flujo básico, cuando el usuario selecciona “Conexión a archivos locales”.
2. El sistema muestra los tipos de archivos a los que se puede conectar: archivos planos, archivos “.xls”, “.txt”
3. El usuario selecciona el tipo de archivo.
4. El sistema muestra una pantalla para configurar la ruta de los archivos, pidiendo ingresar los siguiente campos: Ruta, nombre Fuente de datos, descripción.
5. El usuario ingresa los datos solicitados.
6. El sistema guarda los datos de la configuración.
7. Los pasos 1 a 6 se repiten para todas las fuentes de datos que el usuario desee tener.
8. El caso de uso finaliza cuando el usuario selecciona “Salir” en la ventana.

ii. Modificar Configuración de Fuente de Datos

1. Este flujo alternativo se realiza en el paso 3 del flujo básico, cuando el usuario selecciona “Modificar”.
2. El sistema muestra las Fuentes de Datos ya existentes.
3. El usuario selecciona una Fuente de Datos.
4. El sistema muestra los datos de la configuración para que el usuario pueda modificarlos.
5. El usuario ingresa los datos a modificar.

6. El sistema guarda los datos de la configuración.
7. Los pasos 1 a 6 se repiten para todas las fuentes de datos que el usuario desee modificar.
8. El caso de uso finaliza cuando el usuario selecciona "Salir" en la ventana.

2. Requerimientos Especiales

No existen requerimientos especiales para este caso de uso.

3. Precondiciones

Validación de Usuario: El usuario debe haber ingresado al sistema.

4. Post condiciones

Se ha configurado la fuente de datos con éxito.

5. Puntos de extensión

No hay puntos de extensión para este caso de uso.

3.1.1.2. Caso de Uso Conectar a fuente de datos

El propósito de este caso de uso es el de permitir al usuario conectarse a una fuente de datos configurada en el sistema.

1. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

1.1. Flujo Básico

1. El caso de uso se inicia cuando se desea acceder a una fuente de datos y no existen conexiones abiertas.
2. El sistema crea una cadena de conexión dependiendo de la configuración realizada por el usuario para la fuente de datos.
3. El sistema abre una conexión con la fuente de datos y el caso de uso finaliza.

1.2. Flujos Alternativos

i. *Buscar conexión activa*

1. Este flujo alternativo se realiza en el paso 1 del flujo básico, cuando se inicia la búsqueda de una conexión activa para la fuente de datos.
2. El sistema realiza la búsqueda en el pool de conexiones activas.
3. En caso encuentre una conexión para la fuente, devuelve esta conexión y el caso de uso finaliza. Caso contrario, se retoma el paso 2 del flujo básico.

2. Requerimientos especiales

No existen requerimientos especiales en este caso de uso.

3. Precondiciones

Validación de usuario: El usuario debe haber ingresado al sistema.

Configuración previa de Fuente de Datos: La fuente de datos a conectar debe haber sido configurada correctamente.

4. Postcondiciones

Se realizó la conexión a la fuente de datos exitosamente.

5. Puntos de Extensión

No existen puntos de extensión para este caso de uso.

3.1.1.3. Caso de Uso Obtener estructuras y objetos de fuente de datos

El propósito de este caso de uso es el de permitir al usuario obtener las estructuras y objetos de las fuentes de datos para poder utilizarlos en la creación del flujo de transformación.

1. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

1.1. Flujo Básico

1. El usuario selecciona la fuente de datos de la cual desea obtener la estructura.
2. Si se trata de una base de datos y existe conexión abierta con la fuente de datos se continúa en el paso 4.
3. Incluir Caso de uso Conectar a Fuente de Datos.
4. El sistema muestra en una ventana las tablas que contiene la fuente de datos y sus atributos con sus respectivos tipos de datos.
5. El usuario selecciona "Salir" y el caso de uso finaliza.

1.2. Flujos Alternativos

No hay flujos alternativos para este caso de uso.

2. Requerimientos Especiales

No existen requerimientos especiales para este caso de uso.

3. Precondiciones

Validación de Usuario : El usuario debe haber ingresado al sistema.

Fuente de Datos Configurada : Las fuentes de datos deben haber sido configuradas correctamente.

4. Postcondiciones

Se obtuvo la estructura y objetos de la fuente de datos exitosamente.

5. Puntos de Extensión

No existen puntos de extensión para este caso de uso.

3.1.1.4. Caso de Uso Configurar estructura de archivo plano

Permite al usuario definir la estructura de campos que conforman un archivo plano y convertirlo a objetos que puedan ser utilizados en la construcción del flujo de transformación.

1. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

1.1. Flujo Básico

1. El usuario selecciona el archivo que desea configurar.
2. El sistema muestra una pantalla para configurar la estructura del archivo, pidiendo ingresar los siguientes campos: Primera Línea Nombre de campos, Carácter de fin de registro, Carácter de separación, Ancho Fijo.
3. El usuario ingresa los datos solicitados y selecciona “Cargar Estructura”.
4. El sistema muestra un listado de los campos encontrados en el archivo, y solicita por cada uno de ellos los siguientes datos: Nombre, tipo, tamaño, decimales, nulidad, Primary Key.
5. El usuario ingresa los datos solicitados y selecciona “Analizar Archivo”.
6. El sistema valida la estructura del archivo y guarda la configuración del mismo.

1.2. Flujos Alternativos

No hay flujos alternativos para este caso de uso.

2. Requerimientos Especiales

No existen requerimientos especiales para este caso de uso.

3. Precondiciones

Validación de Usuario: El usuario debe haber ingresado al sistema.

Fuente de Datos Configurada: Las fuentes de datos deben haber sido configuradas correctamente.

4. Postcondiciones

Se obtuvo la estructura y objetos de la fuente de datos exitosamente.

5. Puntos de Extensión

No existen puntos de extensión para este caso de uso.

3.1.2. Paquete Extracción

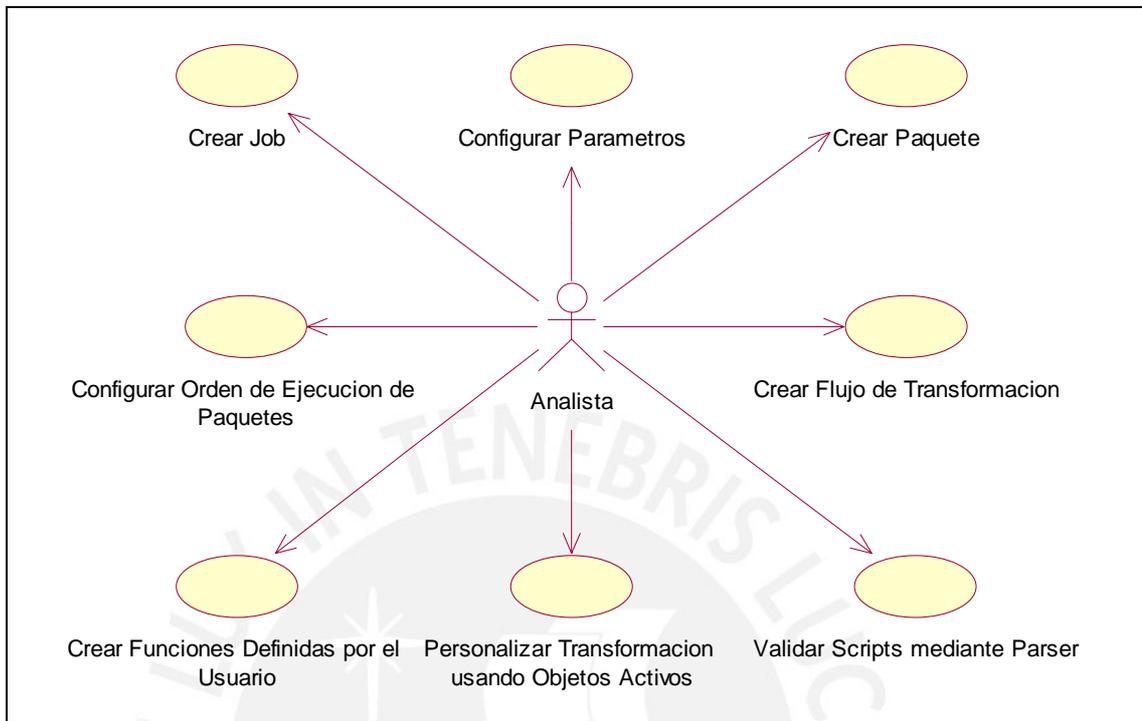


Ilustración 7: Casos de uso del paquete extracción

3.1.2.1. Caso de Uso Crear Paquete

El propósito de este caso de uso es el de crear un paquete de extracción dentro de un job previamente configurado.

1. Flujo de Trabajo

1.1. Flujo Básico

- 1 El usuario selecciona la opción 'Agregar nuevo paquete'.
- 2 El sistema muestra un formulario con los datos a llenar para el nuevo paquete: nombre, descripción y el tema de Análisis al que pertenece.
- 3 El usuario ingresa los datos solicitados.
- 4 El sistema guarda los datos del nuevo Paquete y muestra el área de dibujo para la creación del flujo de transformación.
- 5 El caso de uso finaliza.

1.2. Flujos Alternativos

- i. *Cancelar la creación del nuevo paquete*

1. El usuario selecciona la opción “Cancelar” en cualquier paso del flujo básico.
2. El sistema cierra el formulario, regresa al área de trabajo inicial y el caso de uso finaliza.

2. Requerimientos especiales

No existen requerimientos especiales en este caso de uso.

3. Precondiciones

Validación de Usuario: El usuario ha ingresado al sistema.

Job abierto: El usuario ha abierto un job.

4. Postcondiciones

Se ha creado exitosamente el nuevo paquete.

5. Puntos de extensión

No existen puntos de extensión para este caso de uso.

3.1.2.2. Caso de Uso Crear Job

El propósito de este caso de uso es el de crear un job que permite la administración de paquetes dentro de un proyecto.

1. Flujo de Trabajo

1.1. Flujo Básico

1. El usuario selecciona la opción “Administrar Jobs”.
2. El sistema muestra un formulario de Administración de Jobs con un listado de los jobs existentes para el proyecto, y dos opciones: Buscar y Crear.
3. El usuario selecciona la opción “Crear”.
4. El sistema muestra un formulario con los datos a llenar para el nuevo job: nombre, descripción y la relación de parámetros.
5. El usuario ingresa el nombre, la descripción. Incluir caso de uso Configurar Parámetros.
6. El usuario selecciona la opción “Grabar” y el sistema guarda los datos del nuevo Job.
7. El caso de uso finaliza.

1.2. Flujos Alternativos

i. *Buscar un job*

1. El usuario ingresa un nombre y/o una descripción en el formulario de Administración de Jobs.

2. El usuario selecciona la opción “Buscar”.
3. El sistema muestra el listado de los jobs que contengan el nombre y descripción ingresado por el usuario.
4. El caso de uso finaliza.

ii. Modificar un job

1. El usuario selecciona un job del listado en el formulario de Administración de Jobs.
2. El sistema muestra un formulario con los datos del job.
3. El usuario modifica cualquier campo de información del job.
4. El usuario selecciona la opción “Grabar” y el sistema guarda las modificaciones hechas para el job.
5. El caso de uso finaliza.

iii. Eliminar un job

1. El usuario selecciona la casilla de verificación de los jobs que desea eliminar en el listado que aparece en el formulario de Administración de Jobs.
2. El usuario selecciona la opción “Eliminar”.
3. El sistema elimina los jobs seleccionados
4. El caso de uso finaliza.

iv. Cancelar la creación del nuevo job.

1. El usuario selecciona la opción “Cancelar” en cualquier paso del flujo básico.
2. El sistema cierra el formulario, regresa al área de trabajo inicial y el caso de uso finaliza.

2. Requerimientos especiales

No existen requerimientos especiales en este caso de uso.

3. Precondiciones

Validación de Usuario: El usuario ha ingresado al sistema.

Proyecto abierto: El usuario ha abierto un proyecto.

4. Postcondiciones

Se ha creado/modificado/eliminado exitosamente un job.

5. Puntos de extensión

No existen puntos de extensión para este caso de uso.

3.1.2.3. Caso de Uso Configurar Parámetros

El propósito de este caso de uso es el de crear los parámetros que serán usados en la ejecución de los paquetes.

1. Flujo de Trabajo

1.1. Flujo Básico

1. El usuario selecciona la opción “Agregar Parámetro”.
2. El sistema muestra un formulario con los datos a llenar para el nuevo parámetro: nombre, etiqueta y valor por defecto.
3. El usuario ingresa los datos solicitados.
4. El usuario selecciona la opción “Agregar”.
5. El sistema agrega el parámetro al listado de parámetros del job.
6. El usuario repite los pasos 1-4 por cada parámetro que desee agregar.
7. El caso de uso finaliza.

2. Requerimientos especiales

No existen requerimientos especiales en este caso de uso.

3. Precondiciones

Validación de Usuario: El usuario ha ingresado al sistema.

Job en creación: El usuario se encuentra en proceso de creación de un job.

4. Postcondiciones

Se han agregado exitosamente los nuevos parámetros al job.

5. Puntos de extensión

No existen puntos de extensión para este caso de uso.

3.1.2.4. Caso de Uso Crear flujo de transformación

El propósito de este caso de uso es la definición del flujo de transformación de un paquete.

2. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

2.1. Flujo Básico

1. El caso de uso se inicia cuando el usuario abre un paquete.
2. El sistema muestra el área de dibujo para el flujo de transformación.
3. El usuario agrega objetos pasivos (orígenes y destinos de datos) y activos (procesos de transformación) al área de dibujo seleccionándolos del navegador de fuentes de datos y la paleta de componentes, y los conecta con flujos de acuerdo al orden de ejecución de los componentes que se necesite.

4. El sistema guarda la información del flujo formado y calcula las precedencias de los componentes activos automáticamente de acuerdo a como se han conectado.
5. El sistema ha creado una relación entre los orígenes y destinos, ahora es posible configurar los subcomponentes que permiten la limpieza y la transformación de datos. Finaliza el caso de uso.

2.2. Flujos Alternativos

i. Eliminar componente del flujo de transformación

1. En cualquier momento el usuario puede seleccionar un componente (objeto activo o pasivo) del flujo de transformación y eliminarlo del flujo presionando el botón correspondiente.
2. El sistema elimina las relaciones con otros componentes y se regresa al flujo de trabajo básico.

3. Requerimientos especiales

No existen requerimientos especiales en este caso de uso.

4. Precondiciones

Validación de usuario: El usuario ha ingresado al sistema.

Paquete abierto: El usuario ha abierto un paquete.

Fuente de datos configurada: La conexión a las fuentes de datos debe haberse configurado correctamente para poder colocar objetos pasivos en el flujo de transformación.

5. Postcondiciones

La relación entre los objetos pasivos y activos queda definida para formar el flujo de transformación del paquete.

6. Puntos de Extensión

No existen puntos de extensión para este caso de uso.

3.1.2.5. Caso de Uso Personalizar transformación usando objetos activos

El propósito de este caso de uso es el de permitir configurar la forma como se realizará la transformación de datos desde los orígenes a los destinos del flujo de transformación.

1. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

1.1. Flujo Básico

1. El caso de uso se inicia cuando el usuario selecciona el objeto activo a personalizar dentro del paquete abierto.

2. El usuario selecciona la opción “Configurar Transformación”.
3. El sistema muestra la lista de los campos de las tablas de origen y destino involucrados en el flujo de transformación.
4. El usuario selecciona los campos que desea operar e ingresa las sentencias y funciones que desea aplicar a cada uno.
5. Si el analista lo desea puede abrir una ventana de ayuda con todas las funciones posibles, se muestran las funciones predefinidas y las definidas por el usuario.
6. En el caso de que sea necesario, el analista puede indicar la sentencia WHERE cuando se trate de realizar una transformación con join de tablas.
7. En el caso de que sea necesario, el analista puede indicar los campos GROUP BY destino cuando se utilice funciones acumulativas tales como “sum()” y “prom()”.
8. El usuario elige grabar los cambios.
9. El sistema guarda los cambios aplicados o muestra mensaje de error en el caso se esté tratando de aplicar cambios no permitidos.
10. Se vuelve a la pantalla de creación del flujo y el caso de uso finaliza.

1.2. Flujos Alternativos

i. Cancelar la definición de una transformación

1. En cualquier paso a partir del paso 3 el analista elige la opción “Cancelar” y la definición se interrumpe.
2. El sistema regresa al área de trabajo inicial y el caso de uso finaliza.

ii. Configurar filtrado de datos

1. Este flujo alternativo se inicia en el paso 2 del flujo básico si el usuario selecciona la opción “Configurar Filtro”.
2. El sistema muestra los campos de las tablas origen conectadas al objeto activo seleccionado.
3. El usuario ingresa los criterios de filtrado para los campos seleccionados seleccionando filtros predefinidos o personalizados, o ingresa la sentencia directamente.
4. El analista elige grabar los cambios.
5. El caso de uso continúa en el paso 9 del flujo básico.

iii. Configurar estandarización de datos

1. El flujo alternativo se inicia en el paso 2 del flujo básico si el usuario selecciona la opción “Configurar Estandarización”.
2. El sistema muestra los campos de la tabla destino conectada al objeto activo.

3. El usuario ingresa las funciones de estandarización a aplicar para cada campo.
 4. Si el analista lo desea puede abrir una ventana de ayuda con todas las funciones posibles, se muestran las funciones predefinidas y las definidas por el usuario.
 5. El caso de uso continúa en el paso 9 del flujo básico.
- iv. *Configurar lookup*
1. El flujo alternativo de inicia en el paso 2 del flujo básico si el usuario selecciona la opción “Configurar Lookup”.
 2. El sistema muestra los campos de la tabla destino conectada al objeto activo.
 3. El usuario selecciona para cada campo destino los siguientes datos: Campo origen con valor a buscar, tabla relacionada al campo, campo de la tabla relacionada para comparar y campo de la tabla relacionada de valor de retorno.
 4. El caso de uso continúa en el paso 9 del flujo básico.
- v. *Configurar mediante script*
1. El flujo alternativo se inicia en el paso 1 del flujo básico si el objeto activo a configurar es un Script.
 2. El sistema muestra un formulario para seleccionar un script de las fuentes de datos y la opción para ingresar el script manualmente.
 3. El usuario ingresa el script ya sea manualmente o seleccionándolo de una fuente de datos.
 4. Incluir caso de uso Validar Scripts mediante Parser.
 5. En caso de que la validación haya sido exitosa, el sistema graba los datos y se vuelve a la pantalla de definición del flujo.
 6. El caso de uso finaliza.
- 2. Requerimientos especiales**
- No existen requerimientos especiales en este caso de uso.
- 3. Precondiciones**
- Validación de usuario: El usuario debe haber ingresado al sistema.
- Paquete abierto : Debe haber un paquete abierto.
- Flujo de transformación existente: Debe haberse creado un flujo de transformación con por lo menos un origen de datos y un destino conectados al objeto activo.
- 4. Postcondiciones**
- La configuración del objeto activo se ha completado exitosamente.
- 5. Puntos de Extensión**
- No existen puntos de extensión para este caso de uso.

3.1.2.6. Caso de Uso Validar Scripts mediante Parser

El propósito de este caso de uso es el de verificar que no haya errores de sintaxis en los scripts ingresados por el usuario o incluidos en el flujo de transformación.

1. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

1.1. Flujo Básico

1. El caso de uso es iniciado cuando el usuario selecciona la opción “Validar Script”
2. El usuario ingresa el script a validar por el sistema, así como la fuente de datos relacionada al script.
3. El sistema revisa el script ingresado de acuerdo con la fuente de datos seleccionada.
4. Si el resultado de validación del script es positiva, el sistema muestra un mensaje diciendo que la validación ha sido exitosa. Caso contrario, se muestra un mensaje diciendo que el script no es correcto.

1.2. Flujos Alternativos

No hay flujos alternativos para este caso de uso.

2. Requerimientos especiales

No existen requerimientos especiales en este caso de uso.

3. Precondiciones

Validación de usuario: El usuario debe haber ingresado al sistema.

Script ingresado: Debe haberse ingresado o seleccionado un script antes de llamar al caso de uso.

4. Postcondiciones

El script ha sido validado y se mostró el resultado al usuario.

5. Puntos de Extensión

No existen puntos de extensión para este caso de uso.

3.1.2.7. Caso de Uso Configurar Orden de Ejecución de Paquetes

El propósito de este caso de uso es el de permitir al usuario definir el orden en que se ejecutarán los paquetes que pertenecen a un job.

1. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

1.1. Flujo Básico

1. El caso de uso se inicia cuando el usuario selecciona la opción “Configurar orden de paquetes” para el job abierto.

2. El sistema muestra la lista de paquetes que pertenecen al job abierto y el orden de prioridad asignado a cada uno.
3. El usuario ingresa el orden en que desea que se ejecuten los paquetes, siendo el de menor número el que se ejecutará primero.
4. El sistema guarda el orden de ejecución de los paquetes para el job.
5. El sistema regresa al área de trabajo inicial y el caso de uso finaliza.

1.2. Flujos Alternativos

i. Cancelar la configuración de orden de paquetes

6. En cualquier paso a partir del paso 2 el analista elige la opción “Cancelar” y la configuración se interrumpe.
7. El orden de prioridad que tenían los paquetes antes de hacer cualquier cambio se mantiene.
8. El sistema regresa al área de trabajo inicial y el caso de uso finaliza.

2. Requerimientos especiales

No existen requerimientos especiales en este caso de uso.

3. Precondiciones

Validación de usuario: El usuario debe haber ingresado al sistema.

Job abierto: El usuario ha abierto un job.

Job tiene paquetes: El job debe contener paquetes o no se mostrará nada en la lista para configurar el orden de ejecución.

4. Postcondiciones

El orden de ejecución de los paquetes ha sido configurado exitosamente.

5. Puntos de Extensión

No existen puntos de extensión para este caso de uso.

3.1.2.8. Caso de Uso Crear funciones definidas por el usuario

El propósito de este caso de uso es el de permitir al usuario definir sus propias funciones para la transformación y estandarización de datos en la configuración de los objetos activos del flujo de transformación en un paquete. Hay dos tipos principales de funciones que se pueden definir: por sentencia y por reglas.

1. Flujo de Trabajo

1.1. Flujo Básico

1. El usuario selecciona la opción “Funciones definidas por el usuario”.

2. El sistema muestra un formulario de Administración de Funciones con un listado de las funciones existentes para el paquete, y dos opciones: Buscar y Crear.
3. El usuario selecciona la opción “Crear”.
4. El sistema muestra un formulario con los datos a ingresar para la nueva función: nombre, descripción y tipo de función.
5. El usuario ingresa el nombre, la descripción, y selecciona función de tipo “Sentencia”.
6. El sistema muestra los datos a ingresar para ese tipo de función: número de parámetros, tipo de salida, los tipos de los parámetros de entrada, y un área de texto para ingresar la sentencia.
7. El usuario ingresa los datos solicitados.
8. El usuario selecciona la opción “Grabar” y el sistema guarda los datos de la nueva Función.
9. El caso de uso finaliza.

1.2. Flujos Alternativos

i. Crear una función de tipo Reglas

1. El usuario selecciona la opción “Regla” en el paso 5 del flujo básico.
2. El sistema muestra un listado de reglas, y el usuario selecciona la opción “Agregar Regla”.
3. El sistema muestra un formulario para ingresar los datos de la nueva regla: valor de entrada y valor de salida.
4. El usuario ingresa los datos de la nueva regla y selecciona la opción “Agregar”.
5. El sistema muestra el formulario de la función con el listado actualizado con la nueva regla.
6. El usuario repite los pasos 2-4 por cada regla que desee agregar a la función.
7. El usuario selecciona “Grabar” y el sistema guarda los datos de la nueva función.
8. El caso de uso finaliza.

ii. Buscar una función

1. El usuario ingresa un nombre y/o una descripción en el formulario de Administración de Funciones.
2. El usuario selecciona la opción “Buscar”.
3. El sistema muestra el listado de las funciones que contengan el nombre y descripción ingresado por el usuario.

4. El caso de uso finaliza.

iii. Modificar una Función

1. El usuario selecciona una función del listado en el formulario de Administración de Funciones.
2. El sistema muestra un formulario con los datos de la función.
3. El usuario modifica cualquier campo de información de la función.
4. El usuario selecciona la opción “Grabar” y el sistema guarda las modificaciones hechas para la función.
5. El caso de uso finaliza.

iv. Eliminar una función

1. El usuario selecciona la casilla de verificación de las funciones que desea eliminar en el listado que aparece en el formulario de Administración de Funciones.
2. El usuario selecciona la opción “Eliminar”.
3. El sistema elimina las funciones seleccionadas
4. El caso de uso finaliza.

v. Cancelar la creación de la nueva función.

1. El usuario selecciona la opción “Cancelar” en cualquier paso del flujo básico.
2. El sistema cierra el formulario, regresa al área de trabajo inicial y el caso de uso finaliza.

2. Requerimientos especiales

No existen requerimientos especiales en este caso de uso.

3. Precondiciones

Validación de Usuario: El usuario ha ingresado al sistema.

Paquete abierto: El usuario ha abierto un paquete.

4. Postcondiciones

Se ha creado/modificado/eliminado exitosamente una función.

5. Puntos de extensión

No existen puntos de extensión para este caso de uso.

3.1.3. Paquete Ejecución

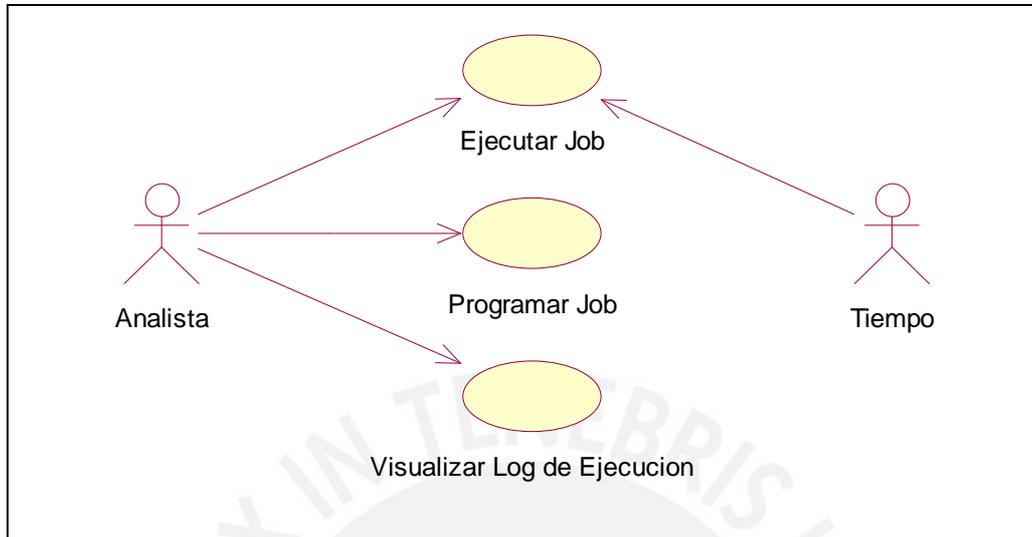


Ilustración 8: Casos de uso del paquete ejecución

3.1.3.1. Caso de Uso Programar Job

El propósito de este caso de uso es el de permitir al usuario programar un job para su ejecución en el servidor en una fecha y hora definidas, con una periodicidad específica.

1. Flujo de Trabajo

1.1. Flujo Básico

1. El usuario selecciona la opción "Programar jobs".
2. El sistema muestra un formulario de Administración de Programaciones con un listado de las programaciones existentes para el proyecto, y dos opciones: Buscar y Crear.
3. El usuario selecciona la opción "Crear".
4. El sistema muestra un formulario con los datos a ingresar para la nueva programación: nombre, descripción, job, activo o no, la hora de ejecución y la periodicidad.
5. El usuario ingresa los datos solicitados para la nueva programación, y selecciona una periodicidad en particular.
6. El sistema muestra los datos a ingresar para ese tipo de periodicidad, si es diaria, se muestra un listado de los días, si es semanal se muestra el día de la semana y la separación entre semanas, si es mensual se muestra el día del

mes, y la separación entre meses, y si es anual se muestra el día del año y la separación entre años.

7. El usuario ingresa los datos para la periodicidad.
8. Según el job seleccionado, el sistema muestra la relación de parámetros que contiene ese job y un área de texto para ingresar valor a cada parámetro.
9. El usuario ingresa los valores de los parámetros que desee, a los que no ingrese valor el sistema toma el valor por defecto para dichos parámetros.
10. El usuario selecciona la opción “Grabar” y el sistema guarda los datos de la nueva Programación.
11. El caso de uso finaliza.

1.2. Flujos Alternativos

i. Buscar una Programación

1. El usuario ingresa un nombre y/o una descripción en el formulario de Administración de Programaciones.
2. El usuario selecciona la opción “Buscar”.
3. El sistema muestra el listado de las programaciones que contengan el nombre y descripción ingresado por el usuario.
4. El caso de uso finaliza.

ii. Modificar una Programación

1. El usuario selecciona una programación del listado en el formulario de Administración de Programaciones.
2. El sistema muestra un formulario con los datos de la programación.
3. El usuario modifica cualquier campo de información de la programación.
4. El usuario selecciona “Grabar” y el sistema guarda las modificaciones hechas para la programación.
5. El caso de uso finaliza.

iii. Eliminar una Programación

1. El usuario selecciona la casilla de verificación de las programaciones que desea eliminar en el listado que aparece en el formulario de Administración de Programaciones.
2. El usuario selecciona la opción “Eliminar”.
3. El sistema elimina las programaciones seleccionadas
4. El caso de uso finaliza.

iv. *Cancelar la creación de la nueva programación.*

1. El usuario selecciona la opción “Cancelar” en cualquier paso del flujo básico.
2. El sistema cierra el formulario, regresa al área de trabajo inicial y el caso de uso finaliza.

2. Requerimientos especiales

No existen requerimientos especiales en este caso de uso.

3. Precondiciones

Validación de Usuario: El usuario ha ingresado al sistema.

Proyecto abierto: El usuario ha abierto un proyecto.

4. Postcondiciones

Se ha creado/modificado/eliminado exitosamente una programación.

5. Puntos de extensión

No existen puntos de extensión para este caso de uso.

3.1.3.2. Caso de Uso Ejecutar Job

El propósito de este caso de uso es el de ejecutar un job programado en el servidor y generar el log de ejecución.

1. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

1.1. Flujo Básico

1. El usuario selecciona la opción “Ejecutar job”.
2. Si el job tiene parámetros configurados, se muestra una pantalla para ingresar los valores de los parámetros.
3. El usuario ingresa los valores de los parámetros y selecciona la opción “Ejecutar”.
4. El sistema inicia la ejecución del job seleccionado utilizando los valores de los parámetros proporcionados (si tenía parámetros configurados).
5. Se incluye el caso de uso Visualizar Log de Ejecución, flujo alternativo 1.
6. El caso de uso finaliza.

1.2. Flujos Alternativos

i. Ejecución automática del job

1. Este flujo alternativo se inicia cuando se llega a la hora programada de ejecución de algún job.
2. El sistema ejecuta el job en el servidor, con los valores de los parámetros

configurados en la programación, y siguiendo el orden de ejecución de los paquetes indicado, y guarda un registro de la ejecución paso por paso.

3. Al finalizar la ejecución se guarda el log de ejecución en el servidor y el caso de uso finaliza

2. Requerimientos especiales

No hay requerimientos especiales..

3. Precondiciones

Validación de usuario: El usuario debe haber ingresado al sistema.

Proyecto abierto: El usuario ha abierto un proyecto

Job abierto: El usuario ha abierto un job.

En el caso de la ejecución automática, solo se requiere que haya un job programado en el servidor.

4. Postcondiciones

La ejecución del job se completó y se grabó el registro de ejecución.

5. Puntos de Extensión

No existen puntos de extensión para este caso de uso.

3.1.3.3. Caso de Uso Visualizar Log de Ejecución

El propósito de este caso de uso es el de permitir al usuario visualizar los pasos que se han ejecutado en el servidor durante la ejecución de un job.

1. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

1.1. Flujo Básico

1. El caso de uso se inicia cuando el analista selecciona la opción "Ver logs".
2. El sistema muestra un listado de los jobs que pertenecen al proyecto abierto.
3. El analista selecciona el job del cual desea visualizar su log de ejecución.
4. El sistema muestra la lista de logs de ejecución del job seleccionado, ordenados por fecha de ejecución.
5. El analista selecciona el log que desea visualizar.
6. El sistema muestra los detalles del log ejecutado, paso a paso, indicando la hora y una descripción de la acción realizada. Si ocurrió algún error en la ejecución, también se muestra con la palabra 'ERROR' al inicio para identificarlo.
7. El caso de uso termina cuando el usuario cierra la ventana de visualización de log de

ejecución.

1.2. Flujos Alternativos

i. Visualizar el log de ejecución durante la ejecución

1. El usuario ejecuta manualmente un job. (Caso de uso Ejecutar Job)
2. El sistema muestra la pantalla de visualización de log de acciones y la actualiza periódicamente para mostrar los pasos de la ejecución en curso hasta que finaliza la ejecución.
3. El caso de uso termina cuando el usuario cierra la ventana de visualización de log de ejecución.

2. Requerimientos especiales

No existen requerimientos especiales.

3. Precondiciones

Validación del usuario: El usuario ha ingresado al sistema.

Proyecto abierto: El usuario ha abierto un proyecto.

4. Postcondiciones

El log de ejecución se visualizó exitosamente.

5. Puntos de Extensión

No existen puntos de extensión para este caso de uso.

3.2. Requisitos de Interfaces Externos

3.2.1. Interfaces de Usuario

La interfaz de usuario será gráfica, a través del explorador web, y el manejo del programa se realizará a través de teclado y ratón.

3.2.2. Interfaces Hardware

Servidor de aplicaciones que proveerá el acceso a la aplicación web.

3.2.3. Interfaces Software

Interfaz JOLAP para el acceso al servidor de aplicaciones.

Interfaz de comunicación con la herramienta DatProducer.

3.2.4. Interfaces de Comunicación

Se requiere una conexión a internet desde el servidor de aplicaciones para poder brindar el servicio a los clientes.

De igual manera se debe contar con acceso desde internet a las fuentes de datos que se configuren en la herramienta para poder realizar la ejecución.

3.2.5. Seguridad

Cuando un usuario intente conectarse al sistema deberá introducir su usuario (login) y clave de acceso (contraseña), y el sistema deberá comprobar que se trata de un usuario autorizado. Si el identificador introducido no corresponde a un usuario autorizado o la clave no coincide con la almacenada, se dará una indicación de error.

La herramienta tendrá distintos tipos de usuarios y a cada uno de ellos se le permitirá únicamente el acceso a aquellas funciones que le correspondan. Esta información se encuentra contenida en los casos de uso del negocio y en las especificaciones de casos de uso del negocio.

El Caso de Uso asociado a este miembro es Validar Usuario.

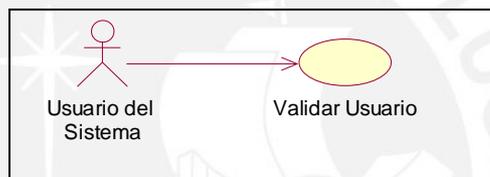


Ilustración 9: Caso de uso Validar usuario

3.2.5.1. Caso de Uso Validar usuario

El propósito de este caso de uso es permitir a un usuario que ingrese al sistema, evitando que cualquier persona ajena al proyecto pueda acceder a la información de dicho proyecto.

1. Flujo de Trabajo

A continuación se detalla el flujo básico de este caso de uso.

1.1. Flujo Básico

1. El usuario ingresa su identificación de usuario (login) y su contraseña.
2. El sistema verifica que los datos ingresados coincidan con los almacenados en el sistema.
3. Si la validación es exitosa concede los privilegios de acuerdo al perfil de usuario y muestra la pantalla principal de la aplicación. Caso contrario vuelve a la pantalla inicial y muestra un mensaje de error.
4. El caso de uso finaliza.

2. Requerimientos especiales

No existen Requerimientos especiales para este caso de uso.

3. Precondiciones

No existen precondiciones para este caso de uso.

4. Postcondiciones

Si la información ingresada es correcta el usuario podrá acceder a utilizar la herramienta y usar las funcionalidades para las cuales tiene privilegios habilitados.

5. Puntos de Extensión

No existen puntos de extensión para este caso de uso.



ANEXO D. DOCUMENTO DE ANALISIS

1. Introducción

El presente es el documento de análisis elaborado por el grupo BI-PUCP para la elaboración de la herramienta BI-PUCP.

1.1 Propósito

El propósito principal de este documento es el de proporcionar al equipo de desarrollo de la herramienta BI-PUCP y a los que lean este documento un mayor entendimiento sobre la distribución de elementos de programación que se utilizarán mediante diagramas de clases, usando técnicas de análisis orientado a objetos, de esta manera se podrá presentar las entidades que formarán parte del producto elaborado y las relaciones entre éstas, esto representado mediante diagramas de clases.

1.2 Alcance

Este documento abarcará principalmente los diagramas de clases para los módulos de Análisis, Extracción y Explotación, incluyendo además un diagrama global que muestre las principales relaciones entre los tres módulos. Además versiones posteriores se mostrará información y diagramas de análisis complementarios, tales como diagramas de estados de las principales clases.

1.3 Referencias

Las referencias aplicables son:

1. Catalogo de requisitos BI-PUCP, 2005
2. Especificación de requerimientos de software, BI-PUCP, 2006

1.4 Visión general del Documento de análisis

Este documento consta de dos secciones. Esta sección es la introducción y proporciona una visión general del Documento de análisis. En la sección dos se muestra los diagramas de clases de análisis de la Herramienta BI-PUCP y un diccionario de datos por cada tabla.

2. Diagramas de clases de análisis

En esta sección se presenta los diagramas de clases de análisis, además se presenta una breve descripción de las clases y sus atributos.

2.1. Diagrama de clases de análisis del módulo de extracción

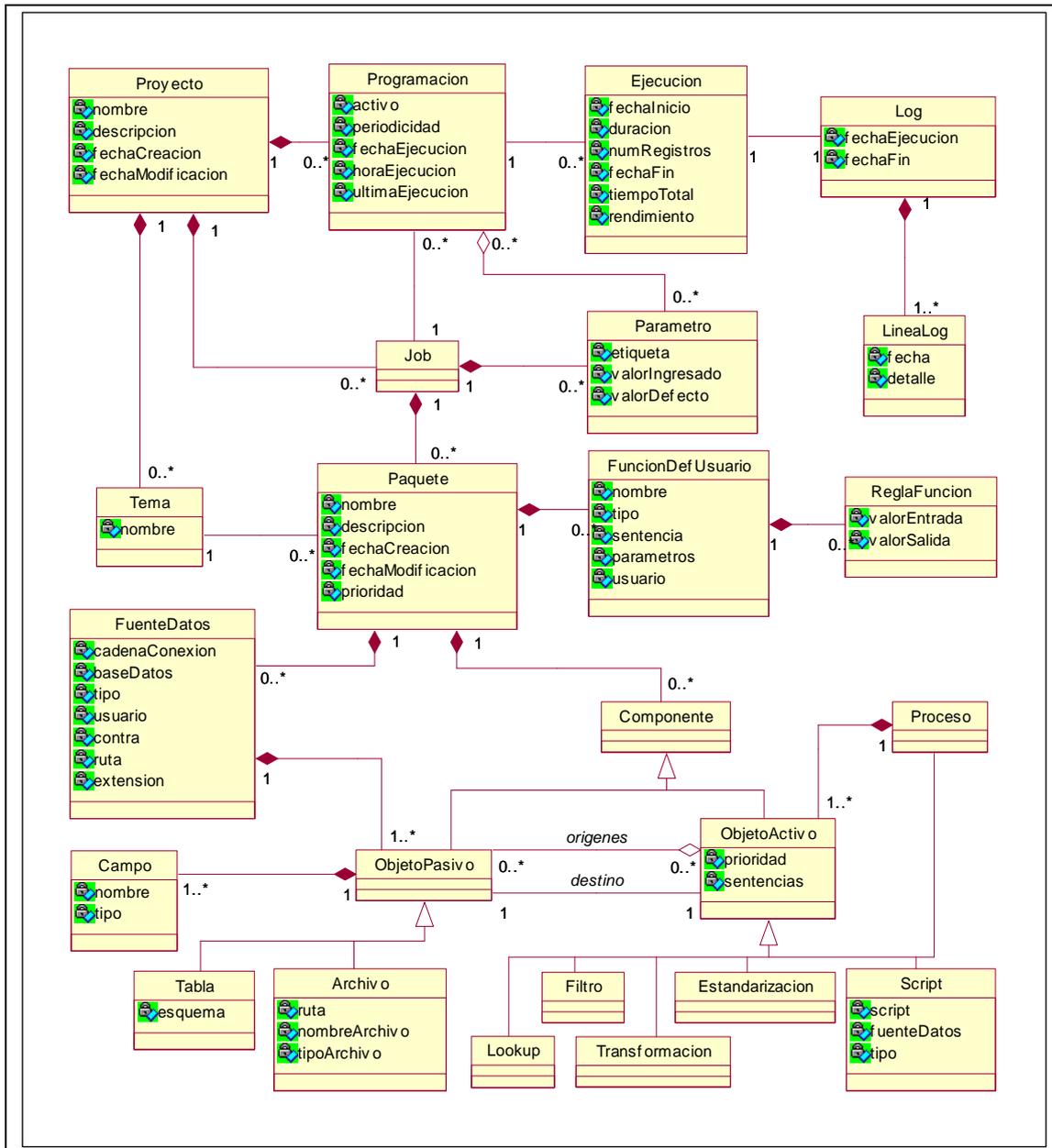


Ilustración 1: Diagrama de Clases de análisis del módulo extracción

2.2. Diccionario de datos de las clases de análisis

Nombre de la clase	Proyecto
Descripción	Clase que contiene la información del proyecto de inteligencia de negocios, común para todos los módulos.
Principales atributos	
Atributo	Descripción
nombre	Nombre del proyecto
descripcion	Breve descripción del proyecto
fechaCreacion	Fecha de creación del proyecto.
fechaModificacion	Fecha de última modificación del proyecto.
Relaciones	
Relación	Descripción
Tema	Un proyecto está compuesto por muchos temas.
Programación	Un proyecto contiene muchas programaciones para ejecución de jobs.
Job	Un proyecto contiene muchos jobs de extracción.

Nombre de la clase	Tema
Descripción	Clase que contiene la información de un tema de análisis para el proyecto de inteligencia de negocios.
Principales atributos	
Atributo	Descripción
nombre	Nombre del tema
Relaciones	
Relación	Descripción
Proyecto	Un tema pertenece a un proyecto de inteligencia de negocios.
Paquete	Un tema puede estar asociado a muchos paquetes de extracción.

Nombre de la clase	Programación
Descripción	Clase que contiene la información de una programación de ejecución para un job. Define la periodicidad de ejecución del job: diaria, semanal, mensual, anual; y la separación entre ejecuciones en días, semanas, meses y años.
Principales atributos	
Atributo	Descripción
activo	Indica si la programación está activa. Si es así, cuando llegue la fecha programada se realizará la ejecución automática.
periodicidad	La periodicidad de ejecución de la programación, así como la separación entre ejecuciones.
fechaEjecucion	Fecha calculada para la próxima ejecución.
horaEjecucion	Hora en la cual se desea que se ejecute el job programado.
ultimaEjecucion	Fecha de última ejecución de la programación.
Relaciones	
Relación	Descripción
Proyecto	Una programación pertenece a un proyecto.
Job	Una programación permite indicar cuando se ejecutará el job asociado.
Parámetro	Una programación establece los valores de ejecución para los parámetros del job asociado.
Ejecución	Una programación puede ejecutarse varias veces.

Nombre de la clase	Ejecucion
Descripción	Clase que contiene la información de una ejecución de una programación y permite calcular el rendimiento de la misma.
Principales atributos	
Atributo	Descripción
fechaInicio	La fecha de inicio de la ejecución.
duracion	La duración de la ejecución.
numRegistros	El número de registros procesados en la ejecución.
fechaFin	La fecha de fin de la ejecución.
tiempoTotal	Tiempo total de la ejecución en segundos.
rendimiento	Rendimiento de la ejecución en registros por segundo.
Relaciones	
Relación	Descripción
Programación	Los datos de rendimiento de la ejecución se generan de la ejecución de una programación.
Log	Los datos obtenidos de la ejecución son almacenados en un log.

Nombre de la clase	Parámetro
Descripción	Clase que representa un parámetro configurable para un job.
Principales atributos	
Atributo	Descripción
etiqueta	Etiqueta del parámetro a ser mostrada en las pantallas de la aplicación al ser configurado por el usuario.
valorDefecto	Valor por defecto del parámetro.
valorIngresado	Valor ingresado por el usuario para el parámetro.
Relaciones	
Relación	Descripción
Programación	Un parámetro contiene el valor configurado para una programación de un job.
Job	Un parámetro pertenece a un job y permite configurar un valor variable.

Nombre de la clase	Log
Descripción	Clase que contiene la información de salida en la ejecución de una programación.
Principales atributos	
Atributo	Descripción
fechaEjecucion	La fecha de inicio de los datos del log.
fechaFin	La fecha de fin de registro de datos en el log.
Relaciones	
Relación	Descripción
Ejecución	Un log se genera de la ejecución de una programación.
LineaLog	Un log está compuesto por muchas líneas de log.

Nombre de la clase	LineaLog
Descripción	Clase que contiene la información de una línea del log.
Principales atributos	
Atributo	Descripción
fecha	La fecha en que se registró la línea del log.
detalle	El detalle con la información de la línea del log.
Relaciones	
Relación	Descripción
Log	Una línea log pertenece a un log de ejecución.

Nombre de la clase	Job
Descripción	Clase que agrupa varios paquetes relacionados.
Principales atributos	
Atributo	Descripción
-	-
Relaciones	
Relación	Descripción
Proyecto	Un job pertenece a un proyecto.
Paquete	Un job contiene muchos paquetes.
Programación	Un job puede estar asociado a varias programaciones.
Parámetro	Un job puede contener varios parámetros configurables.

Nombre de la clase	Paquete
Descripción	Esta clase contiene la información de la relación entre las fuentes de datos de origen y destino, define el flujo de transformación.
Principales atributos	
Atributo	Descripción
nombre	Nombre del paquete.
descripcion	Breve descripción del paquete.
fechaCreacion	La fecha de creación del paquete.
fechaModificacion	La fecha de última modificación del paquete o de sus componentes.
prioridad	Indica la prioridad de ejecución del paquete dentro del job al que pertenece.
Relaciones	
Relación	Descripción
Tema	Un paquete trabaja con tablas para un tema específico.
FuenteDatos	Un paquete tiene muchas fuentes de datos.
Componente	Un paquete está compuesto por muchos componentes, los cuales forman el flujo de transformación.
Job	Un paquete pertenece a un job.
FuncionDefUsuario	Un paquete puede tener muchas funciones definidas por el usuario.

Nombre de la clase	FuncionDefUsuario
Descripción	Clase que contiene la información de una función definida por el usuario. Puede ser del tipo sentencia o tipo reglas.
Principales atributos	
Atributo	Descripción
nombre	Nombre de la función definida por usuario.
tipo	Tipo de la función: sentencia o reglas.
sentencia	La sentencia de transformación o de operaciones definidas por el usuario.
parametros	Lista de parámetros que acepta la función.
usuario	Usuario que creó la función.
Relaciones	
Relación	Descripción
Paquete	Una función definida por el usuario pertenece a un paquete.
ReglaFuncion	Una función definida por el usuario de tipo reglas tiene una o más reglas.

Nombre de la clase	ReglaFuncion
Descripción	Clase que representa una regla de función definida por el usuario.
Principales atributos	
Atributo	Descripción
valorEntrada	El valor de entrada de la regla.
valorSalida	El valor que se devuelve al recibir el valor de entrada.
Relaciones	
Relación	Descripción
FuncionDefUsuario	Una regla de función pertenece a una función definida por el usuario.

Nombre de la clase	FuenteDatos
Descripción	Clase que contiene la información para conectarse a las fuentes de datos que utiliza la herramienta para realizar la transformación. La fuente de datos puede ser una base de datos o un archivo.
Principales atributos	
Atributo	Descripción
cadenaConexion	La cadena la cual se utiliza para conectarse a la base de datos.
baseDatos	El nombre de la base de datos.
tipo	Indica el tipo de fuente de datos: SQL, Oracle, MySQL.
usuario	El usuario con el cual se conecta a la fuente de datos.
contra	La contraseña que el usuario ingresa para conectarse a la base de datos.
ruta	La ruta del archivo que es la fuente de datos.
extension	La extension del archivo que se usa como fuente de datos.
Relaciones	
Relación	Descripción
Paquete	Una fuente de datos pertenece a un paquete.
ObjetoPasivo	Una fuente de datos contiene muchos objetos pasivos.

Nombre de la clase	Componente
Descripción	Clase que contiene la información de un componente utilizado para realizar la transformación.
Principales atributos	
Atributo	Descripción
-	-
Relaciones	
Relación	Descripción
Paquete	Un componente pertenece a un solo paquete, formando el flujo de transformación.
ObjetoActivo	Un componente puede ser de tipo objeto activo.
ObjetoPasivo	Un componente puede ser de tipo objeto pasivo.

Nombre de la clase	ObjetoActivo
Descripción	Clase que contiene la información de un objeto activo para la transformación. Un objeto activo es aquel que se encarga de procesar los datos de las fuentes.
Principales atributos	
Atributo	Descripción
prioridad	Prioridad de ejecución del objeto activo dentro del paquete.
sentencias	Sentencias que configuran la ejecución del objeto.
Relaciones	
Relación	Descripción
Componente	Un objeto activo es un tipo de componente.
ObjetoPasivo	Un objeto activo se relaciona con muchos objetos pasivos, pues con ellos intercambia los datos a procesar y luego de ser procesados. Puede tener varios orígenes de datos, pero solo un destino.
Proceso	Un objeto activo puede ser de tipo proceso.
Script	Un objeto activo puede ser de tipo script.
Estandarizacion	Un objeto activo puede ser de tipo estandarización.
Filtro	Un objeto activo puede ser de tipo filtro.
Transformacion	Un objeto activo puede ser de tipo transformación.

Nombre de la clase	Proceso
Descripción	Clase que contiene el proceso de transformación entre dos orígenes de datos. Contiene a los subcomponentes de transformación: filtro, transformación y estandarización.
Principales atributos	
Atributo	Descripción
-	-
Relaciones	
Relación	Descripción
ObjetoActivo	Un proceso es un tipo de objeto activo. Además contiene otros objetos activos como subcomponentes.

Nombre de la clase	Filtro
Descripción	Clase que contiene la información del subcomponente de filtro de un proceso.
Principales atributos	
Atributo	Descripción
-	-
Relaciones	
Relación	Descripción
ObjetoActivo	Una operación de Filtro es un tipo de objeto activo.

Nombre de la clase	Transformacion
Descripción	Clase que contiene la información del subcomponente de transformación de un proceso.
Principales atributos	
Atributo	Descripción
-	-
Relaciones	
Relación	Descripción
ObjetoActivo	Una operación de Transformación es un tipo de objeto activo.

Nombre de la clase	Estandarizacion
Descripción	Clase que contiene la información del subcomponente de estandarización de un proceso.
Principales atributos	
Atributo	Descripción
-	-
Relaciones	
Relación	Descripción
ObjetoActivo	Una operación de Estandarización es un tipo de objeto activo.

Nombre de la clase	Script
Descripción	Clase que contiene la información de un script que forma parte de un paquete.
Principales atributos	
Atributo	Descripción
script	El script SQL a ser ejecutado.
fuentesDatos	La fuente de datos en la cual se ejecutará el script.
tipo	El tipo de script a ejecutarse. Puede ser tipo programa o tipo procedure.
Relaciones	
Relación	Descripción
ObjetoActivo	Un script SQL es un tipo de objeto activo.

Nombre de la clase	Lookup
Descripción	Clase que contiene la información del subcomponente de lookup de un proceso.
Principales atributos	
Atributo	Descripción
-	-
Relaciones	
Relación	Descripción
ObjetoActivo	Una operación de Lookup es un tipo de objeto activo.

Nombre de la clase	ObjetoPasivo
Descripción	Clase que contiene la información de un objeto pasivo para la transformación. Un objeto pasivo es aquél que representa un origen o destino de datos para la transformación.
Principales atributos	
Atributo	Descripción
-	-
Relaciones	
Relación	Descripción
Componente	Un objeto pasivo es un tipo de componente.
ObjetoActivo	Un objeto pasivo se relaciona con muchos objetos activos de acuerdo a como se defina el flujo de transformación. Puede ser el origen o destino de un objeto activo.
FuenteDatos	Un objeto pasivo pertenece a una fuente de datos.
Tabla	Un objeto pasivo puede ser de tipo tabla.
Archivo	Un objeto pasivo puede ser de tipo archivo.
Campo	Un objeto pasivo esta compuesto por campos, que indican la estructura que tiene.

Nombre de la clase	Tabla
Descripción	Clase que contiene la información de una tabla.
Principales atributos	
Atributo	Descripción
esquema	Contiene el esquema al que pertenece la tabla.
Relaciones	
Relación	Descripción
ObjetoPasivo	Una tabla es un tipo de objeto pasivo.

Nombre de la clase	Archivo
Descripción	Clase que contiene la información de un archivo.
Principales atributos	
Atributo	Descripción
ruta	Ruta de acceso al archivo.
nombreArchivo	Nombre del archivo.
tipo	Indica el tipo de archivo que es: archivo plano, archivo excel.
Relaciones	
Relación	Descripción
ObjetoPasivo	Un archivo es un tipo de objeto pasivo.



ANEXO E. DOCUMENTO DE DISEÑO

1. Introducción

El presente “Documento de Diseño” contiene los diagramas de clases de diseño para las funcionalidades que serán implementadas en la construcción de la herramienta AXEbit correspondientes al módulo de extracción. Los diagramas se han elaborado utilizando el lenguaje de modelamiento UML.

1.1. Propósito

El objeto del presente documento es definir la estructura de los datos que manejará la herramienta, así como la relación entre éstos.

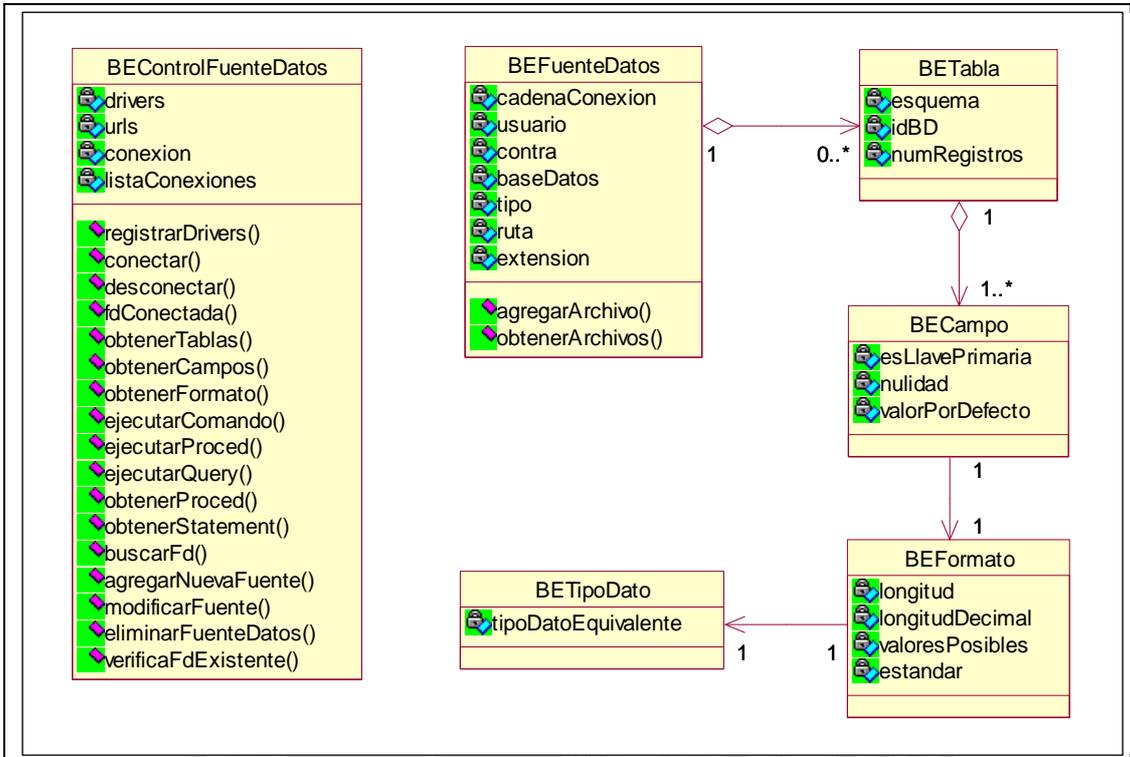
Se busca obtener una definición clara de los controles y gestores que se encargarán de realizar las funcionalidades haciendo un manejo ordenado de las clases que interactúan.

1.2. Visión General del Documento

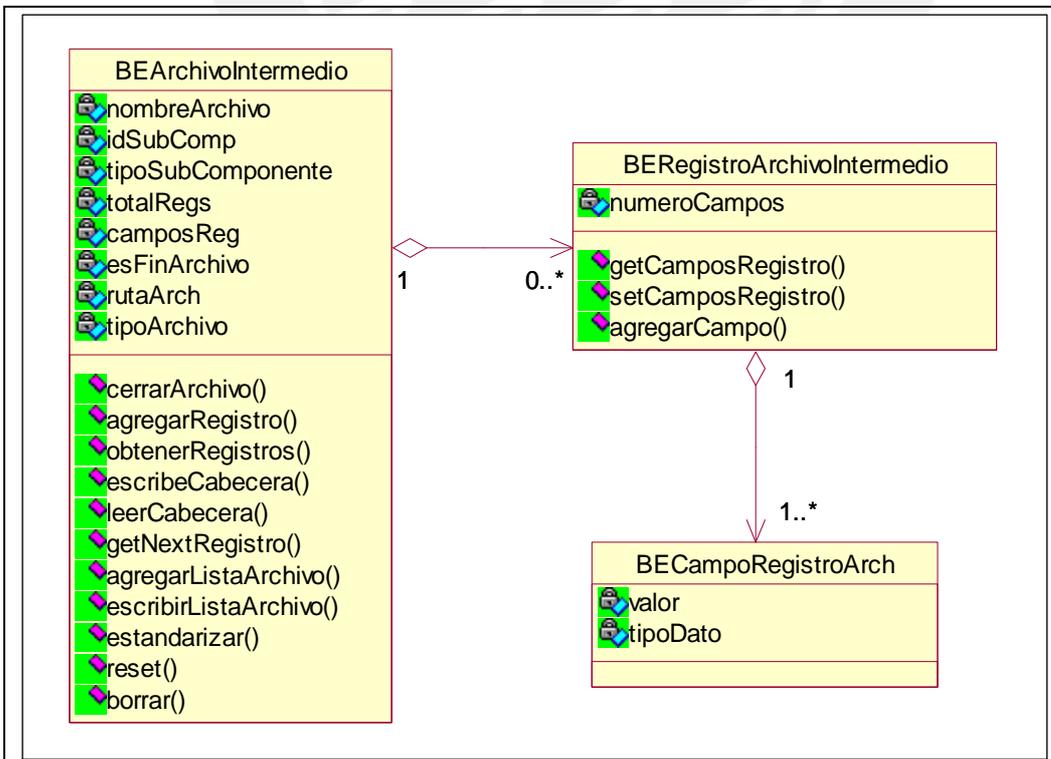
Este documento consta de 2 partes. Esta primera parte introductoria da una idea general de lo que se busca lograr con el uso de este documento.

La segunda parte muestra en detalle los diagramas de clases de diseño para las funcionalidades a ser implementadas.

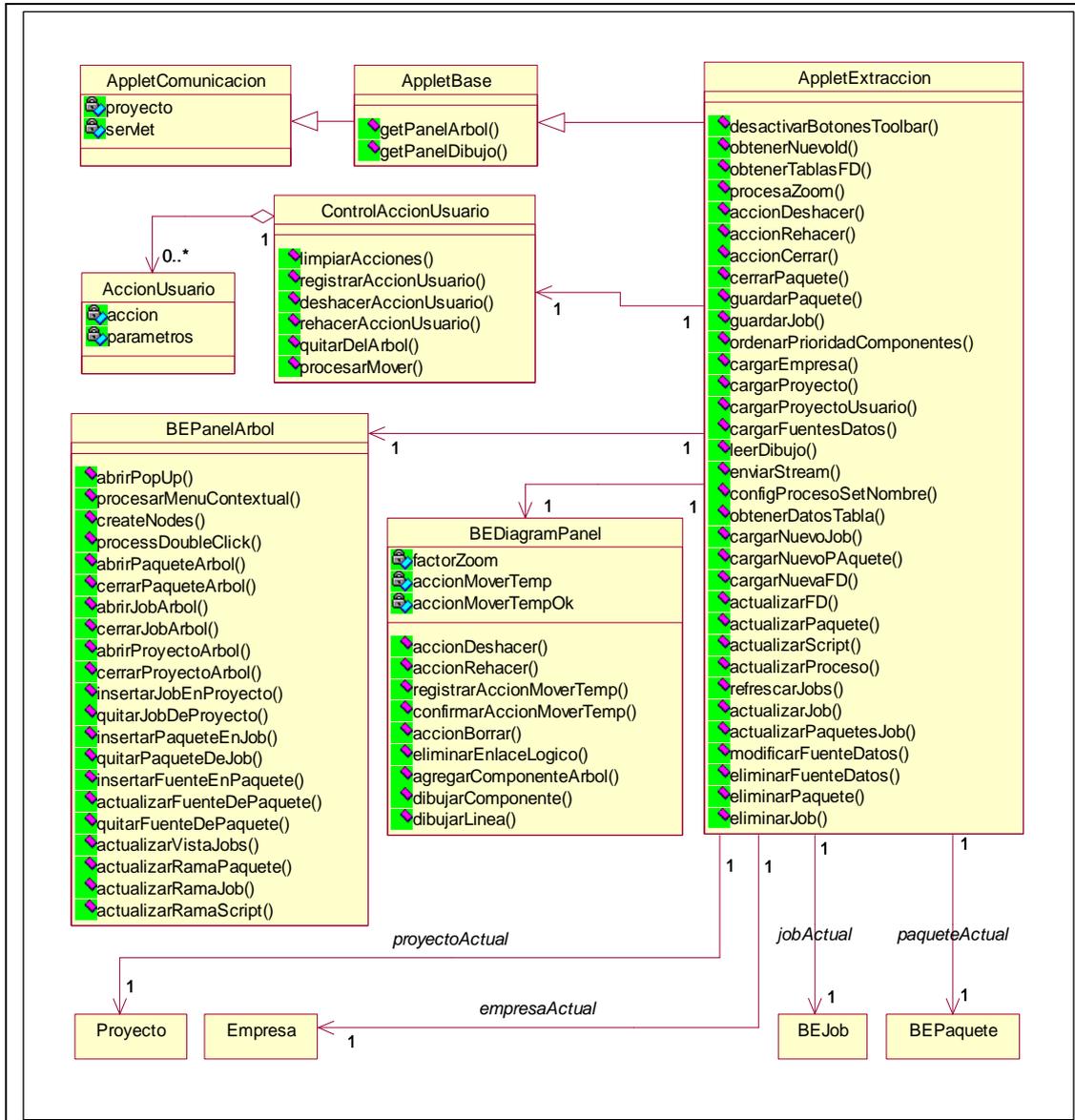
2.2. Clases de conexión a base de datos



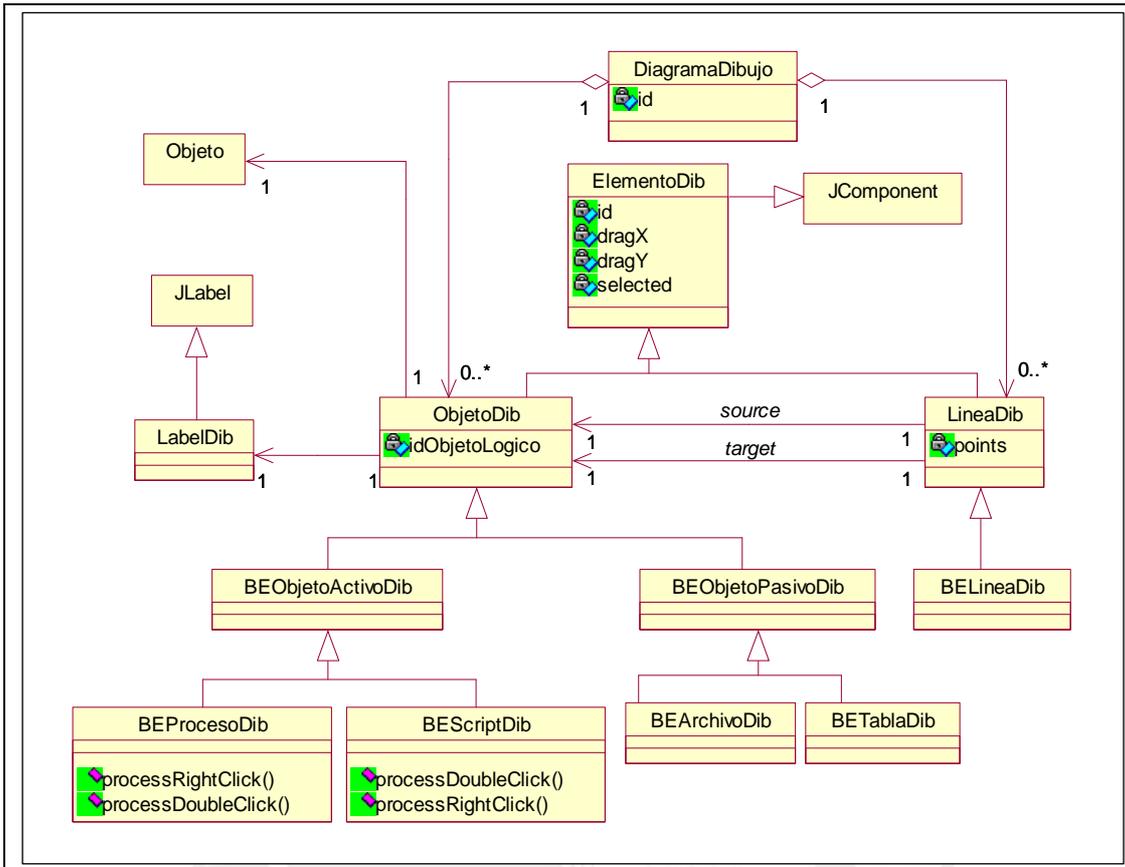
2.3. Clases de manejo de archivos intermedios



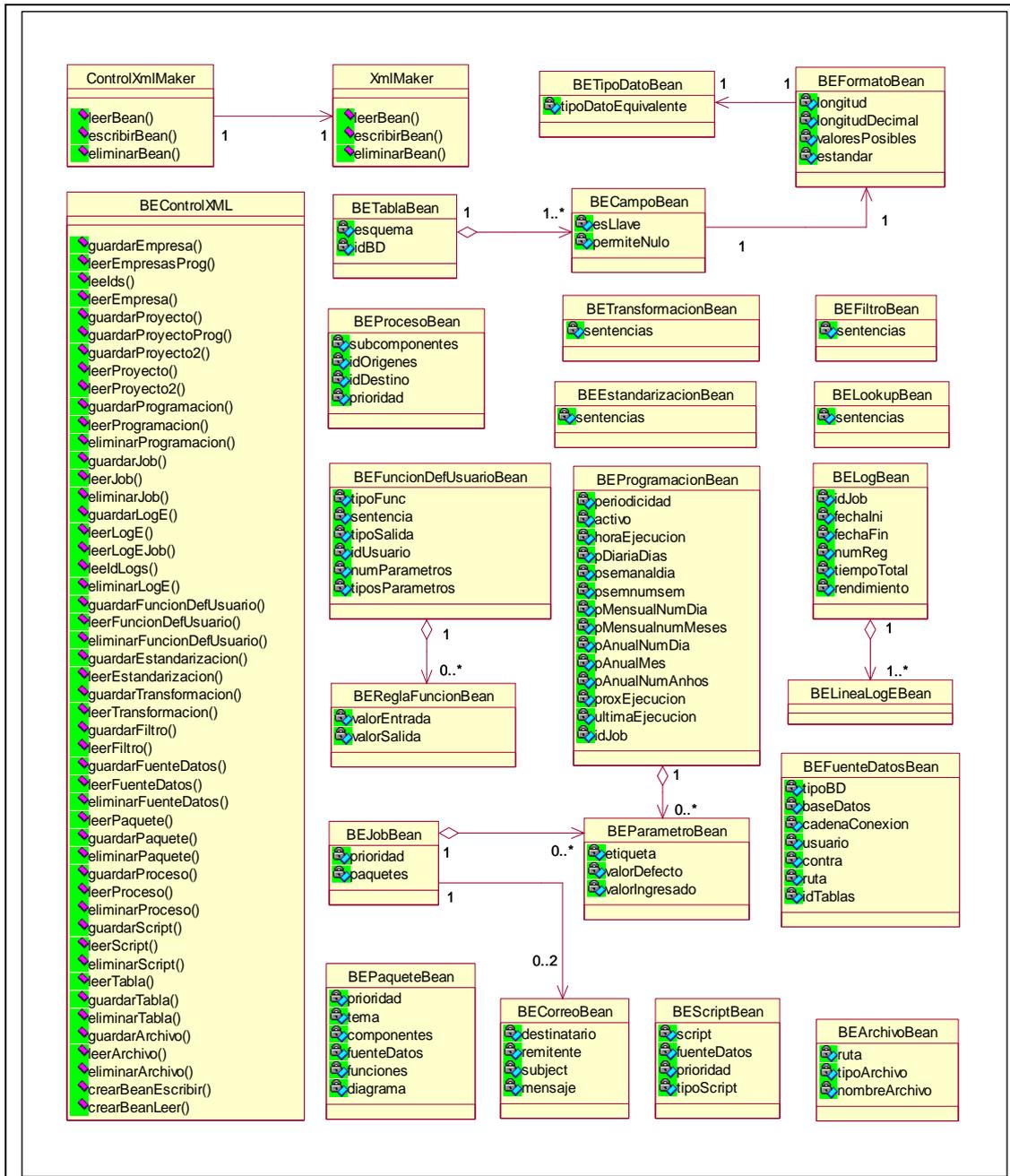
2.4. Clases del applet



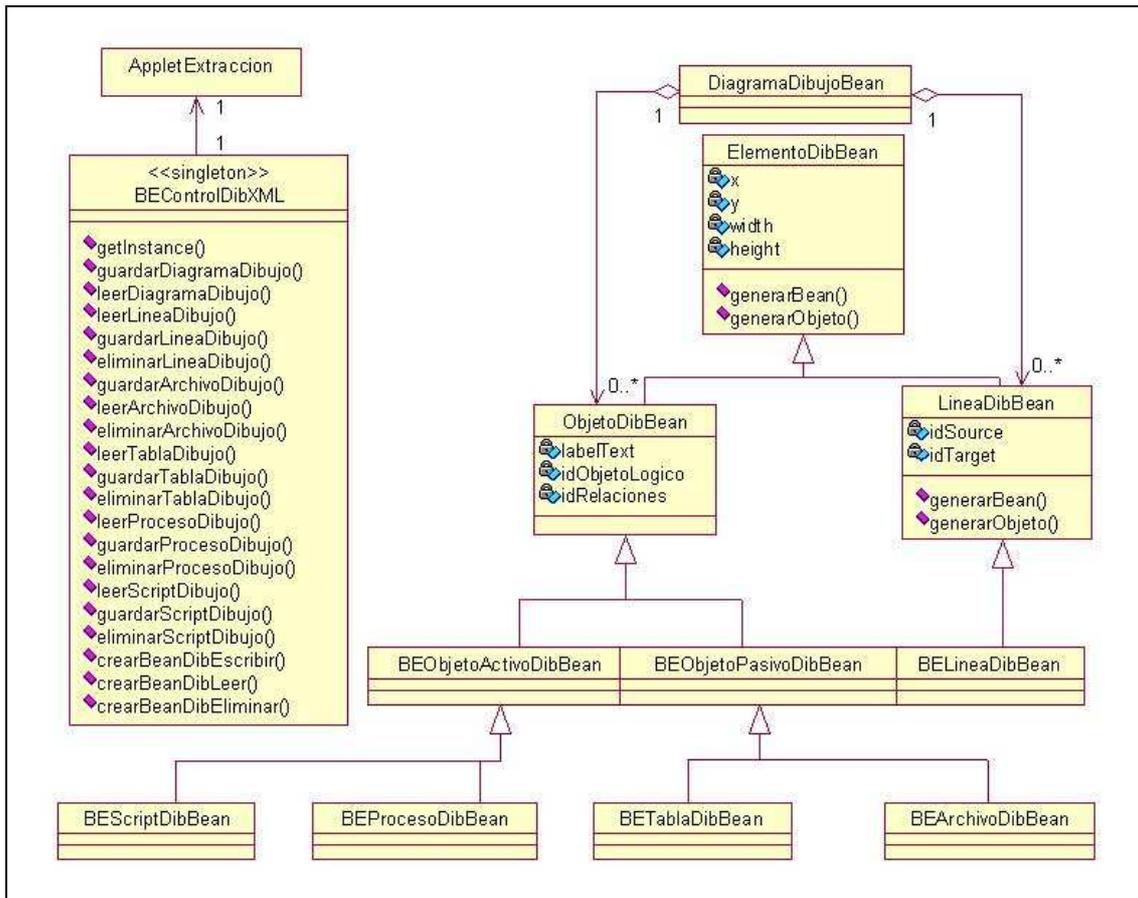
2.5. Clases de dibujo



2.6. Clases de persistencia



2.7. Clases de persistencia de dibujo



ANEXO F. DOCUMENTO DE ARQUITECTURA

1. Introducción

1.1. Objetivo

Este documento nos proporcionará una descripción de la arquitectura de la aplicación desarrollada por el proyecto BI-PUCP a través de una serie de diferentes vistas arquitectónicas del mismo para representar diferentes aspectos del sistema.

Con la arquitectura se podrá también presentar las relaciones existentes entre los 3 módulos de Análisis, Extracción y Explotación y la manera de verificar dichas relaciones, lo que ayudará al equipo de desarrollo a validar la consistencia de las mismas.

1.2. Alcances

Este documento describirá las vistas de Caso de Uso, Lógica, de Despliegue y de Implementación de la representación arquitectónica del software que se desarrollará. Este documento se aplica al desarrollo del prototipo de arquitectura y sienta las bases para el diseño e implementación de las iteraciones de la fase de construcción de la herramienta AXEbit.

1.3. Referencias

Para la elaboración del presente documento, se tomaron como referencias a los siguientes documentos:

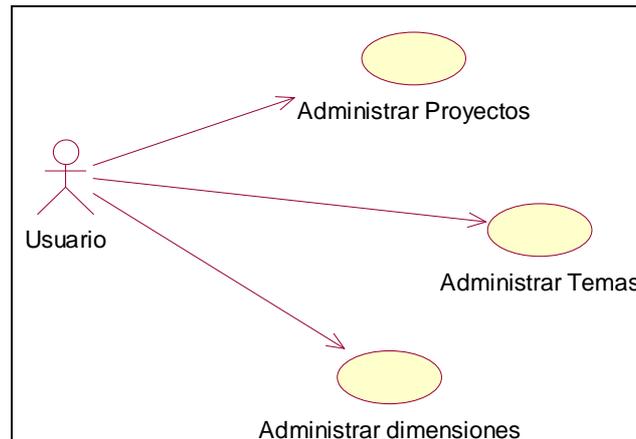
1. Especificación de requerimientos de software, v1, BI-PUCP, 2006.
2. Plan de proyecto, v1, BI-PUCP, 2006.
3. Documento de análisis, v1, BI-PUCP, 2006.

2. Vista de casos de uso

Estos son los casos de uso que se implementaron en el prototipo de arquitectura de manera que se pudiera comprobar su viabilidad.

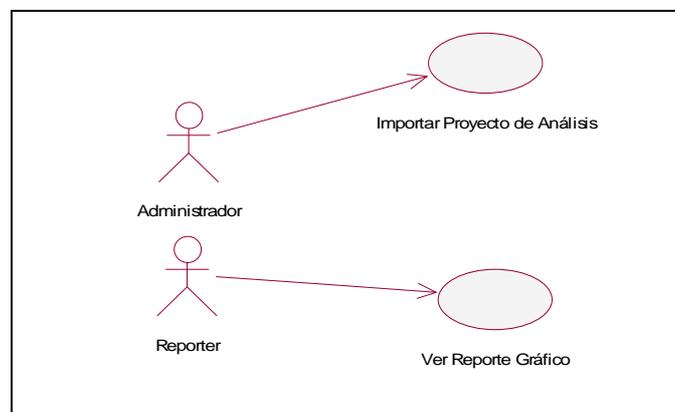
Se utilizaron los componentes mencionados en la Vista de Despliegue del presente documento en los tres módulos del prototipo y se comprobó el efectivo funcionamiento del mismo.

2.1. Módulo de Análisis



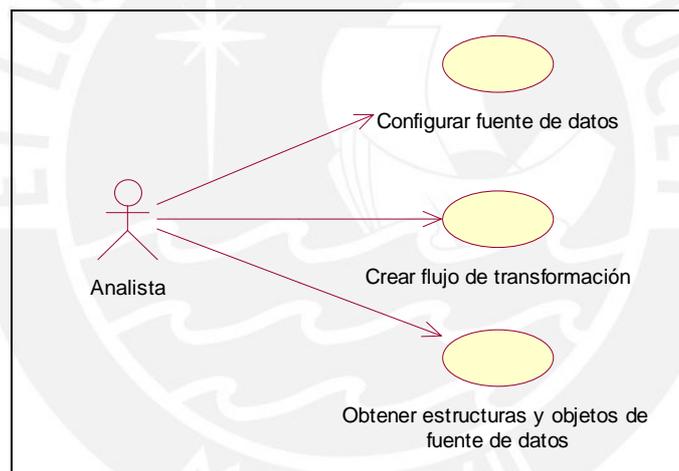
- **Administrar proyectos:** Se implementó la parte de creación de un proyecto para el sistema, el cual administrará los temas de análisis y sus componentes.
- **Administrar temas de análisis:** Consiste en la creación de un tema de análisis dentro de un proyecto, para contener tablas relacionadas entre sí, las cuales tendrán la información del tema, en las dimensiones y facts.
- **Administrar dimensiones:** En esta etapa consiste en crear tablas que contendrán la información que generará la metadata. Estas tablas tendrán aquellos valores que serán objeto de análisis.
- **Persistencia en xml:** Consiste en un modulo especializado para el manejo de la información de los proyectos que se generen al momento de utilizar la herramienta. Básicamente el Servlet llamaría a este modulo utilizando un controlador para que este guarde y recupere la información de los proyectos. Este no es propiamente un caso de uso pero es un tema importante dentro de nuestra implementación.

2.2. Módulo de Explotación



- **Importar proyecto de análisis:** Engloba las siguientes funcionalidades:
- **Importar Tema de Análisis:** Consiste en formar un cubo de explotación en base a un tema creado por el módulo de análisis. El cubo sería creado en base a la metadata del tema de análisis.
- **Crear Esquema de explotación:** Consiste en la creación de un esquema XML basado en la información obtenida del cubo creado en base al esquema de análisis. Dicho esquema sigue el formato específico de comunicación con el motor OLAP.
- **Administrar Reporte (flujo de creación) y Ver Reporte (gráfico):** Consiste en mostrar el resultado gráfico de un reporte. la implementación de la comunicación con el motor OLAP por medio de sentencias MDX.

2.3. Módulo de Extracción



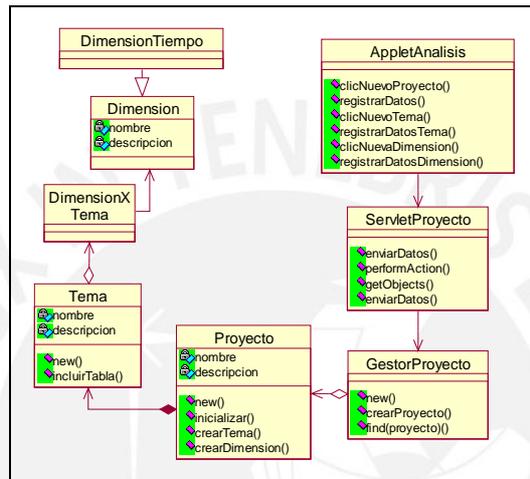
- **Configurar fuente de datos:** Consiste en ingresar los datos necesarios para realizar una conexión a la base de datos que se está configurando para su uso en la herramienta.
- **Crear flujo de transformación:** Consiste en formar un flujo de transformación en el área de dibujo de la herramienta de manera gráfica.
- **Obtener estructuras y objetos de fuente de datos:** Consiste en visualizar la estructura de tablas y campos de las fuentes de datos configuradas en la herramienta.

3. Vista Lógica

Esta sección es una descripción de la vista lógica de la arquitectura. Describe las clases más importantes y la interacción entre las mismas.

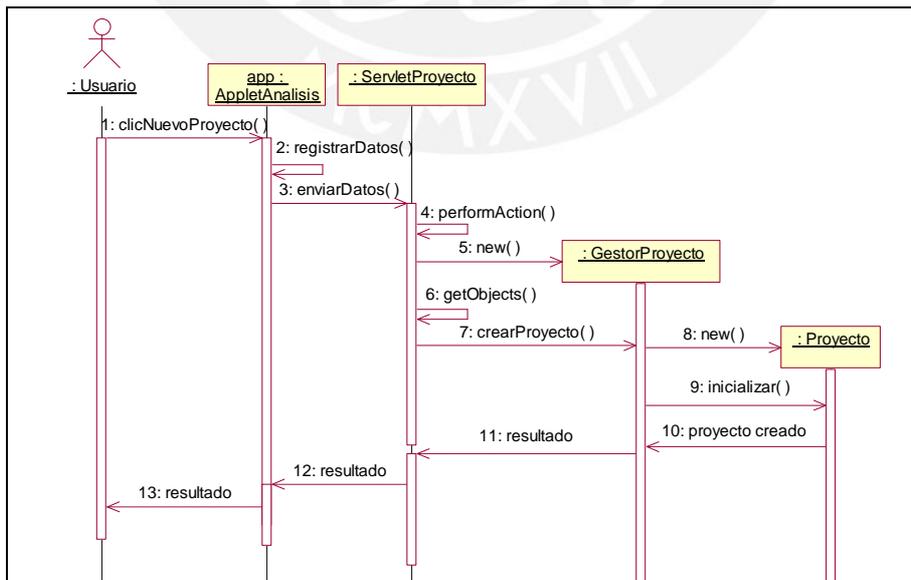
3.1. Módulo de Análisis

3.1.1. Diagrama de Clases

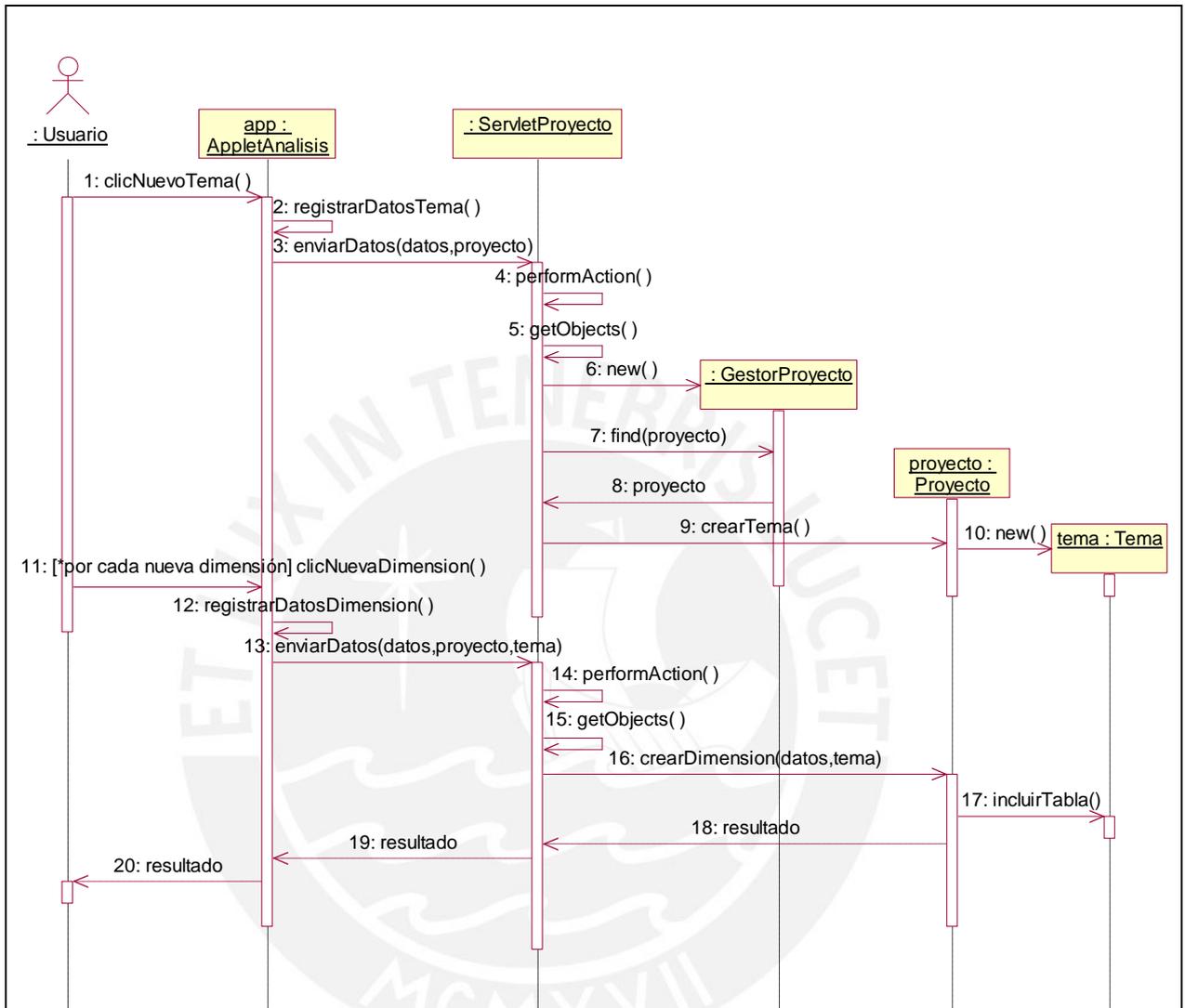


3.1.2. Diagramas de secuencias

Caso de uso “Administrar proyecto” (flujo creación)

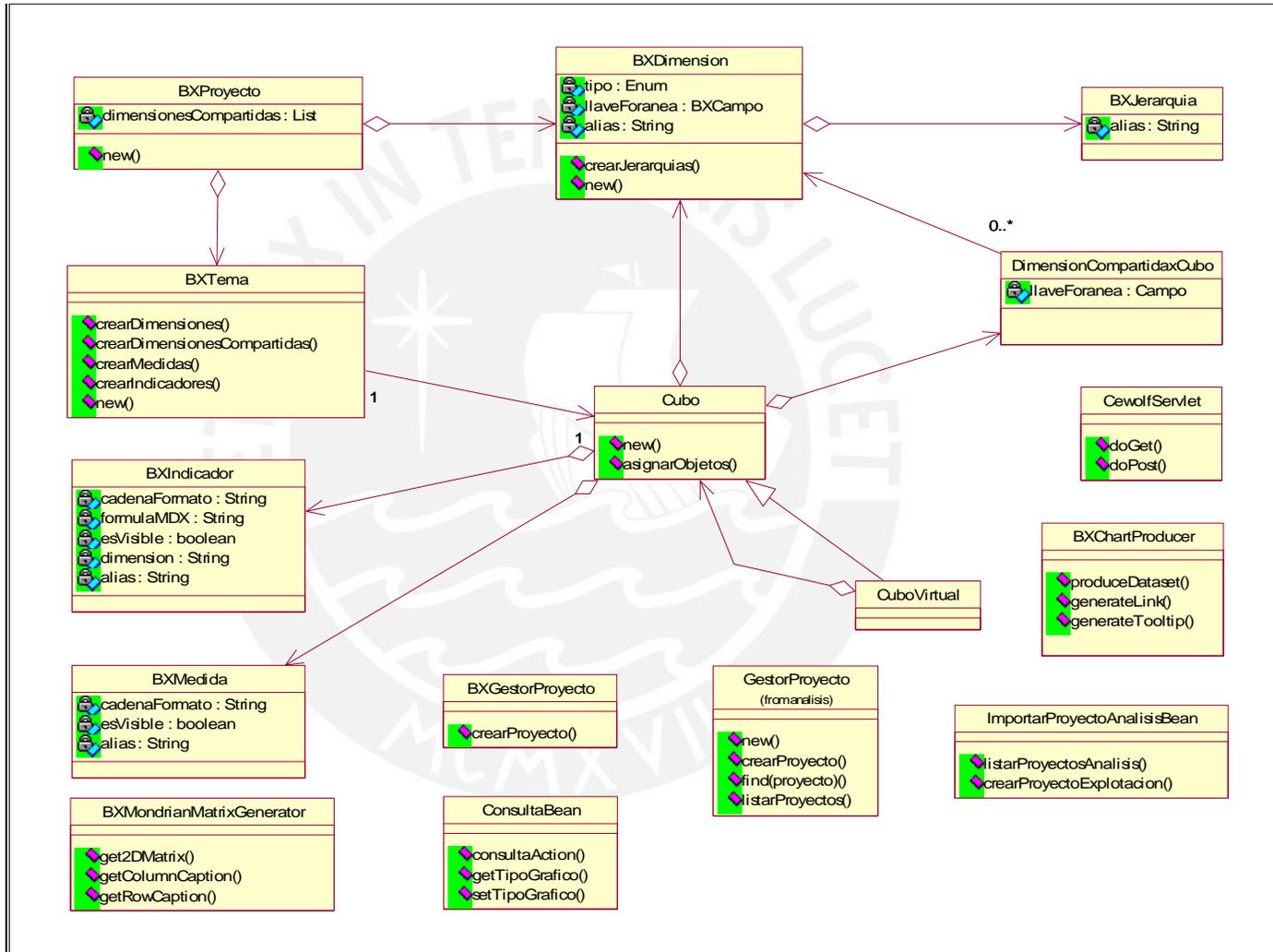


Casos de uso “Administrar dimensiones” y “Administrar temas de análisis” (flujos creación).



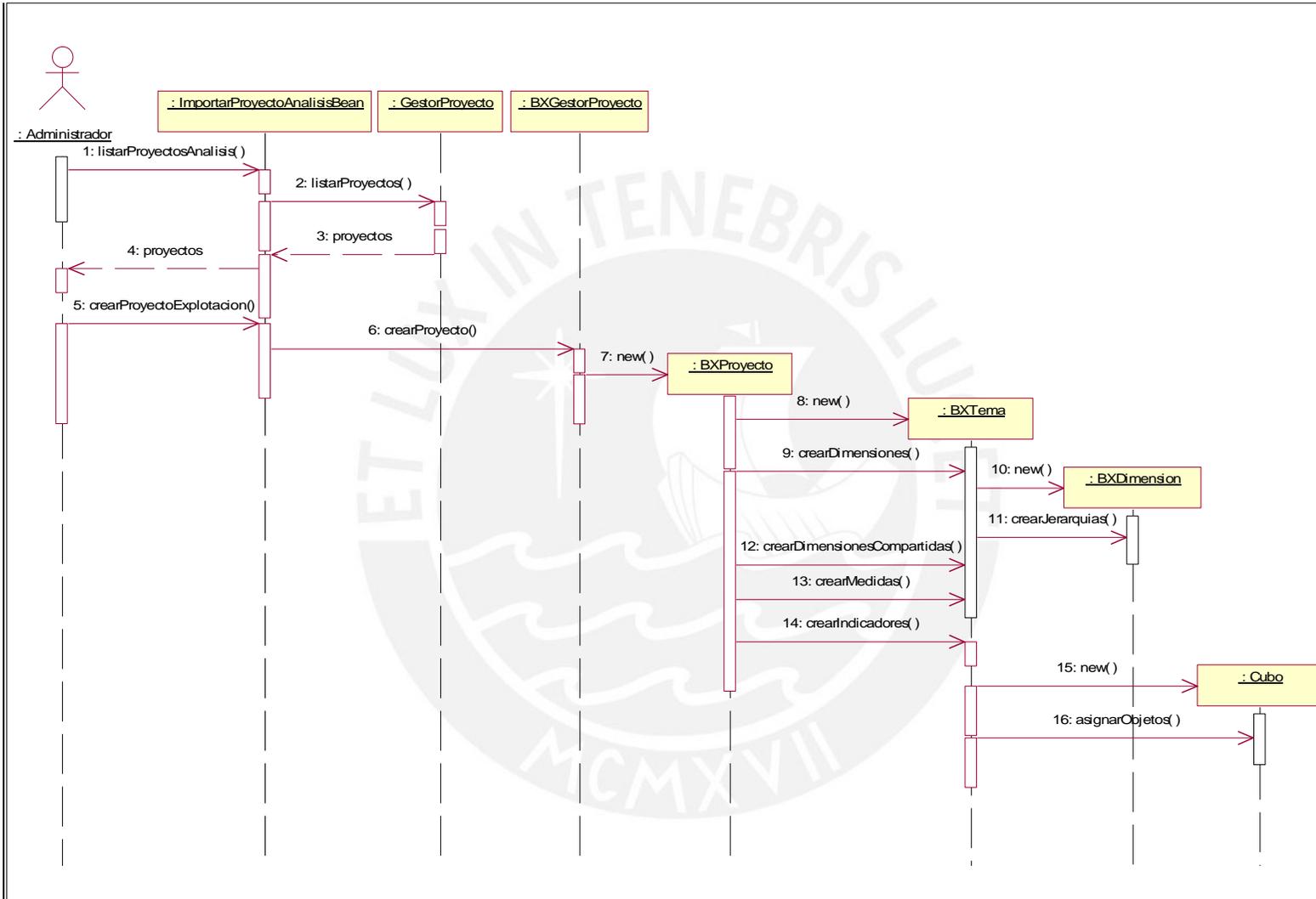
3.2. Módulo de Explotación

3.2.1. Diagrama de Clases

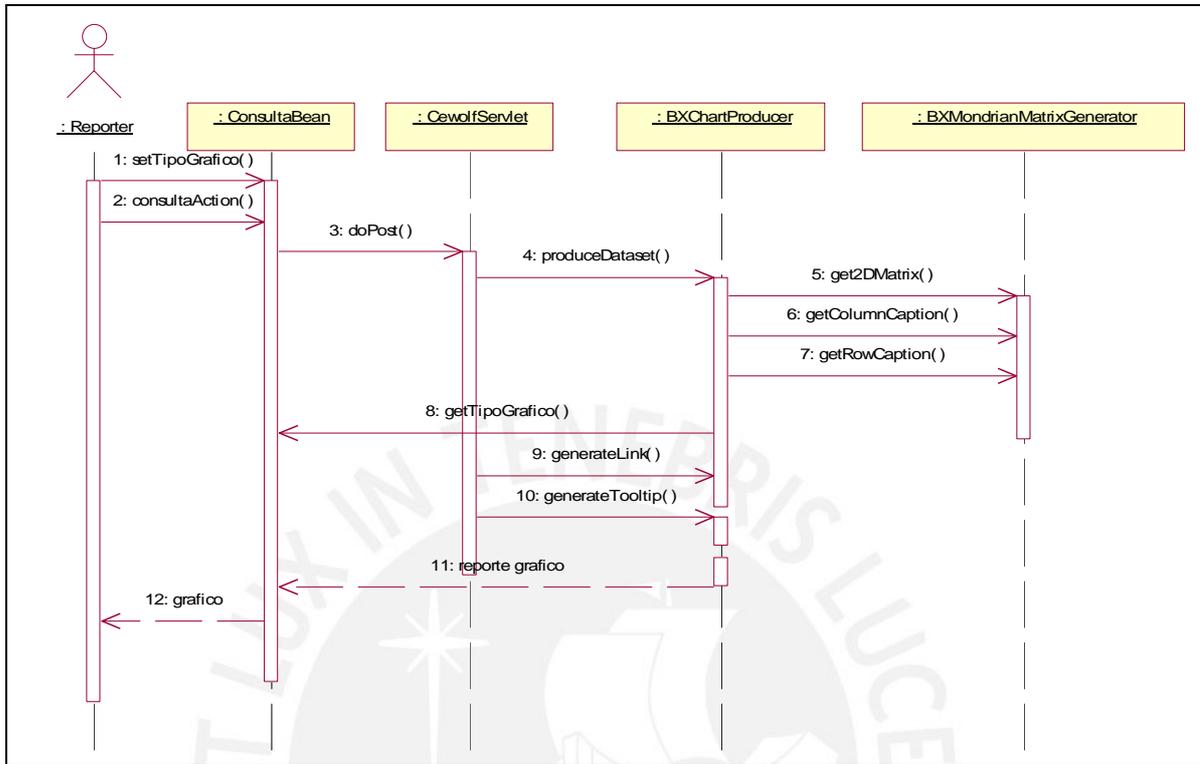


3.2.2. Diagrama de Secuencias

Caso de Uso Importar Proyecto de Análisis

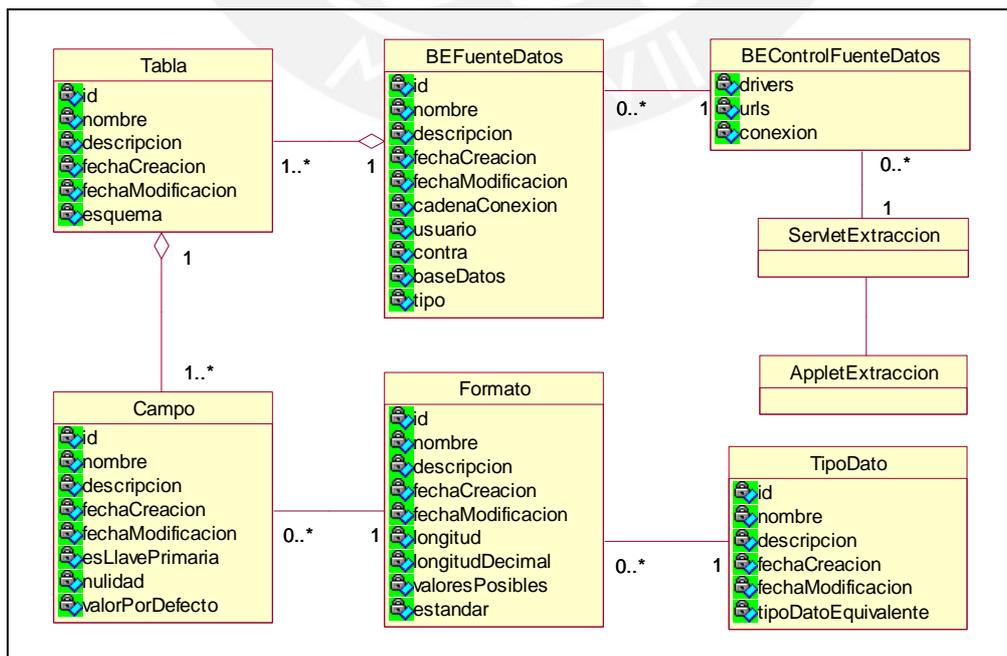


Caso de Uso Ver Reporte Gráfico



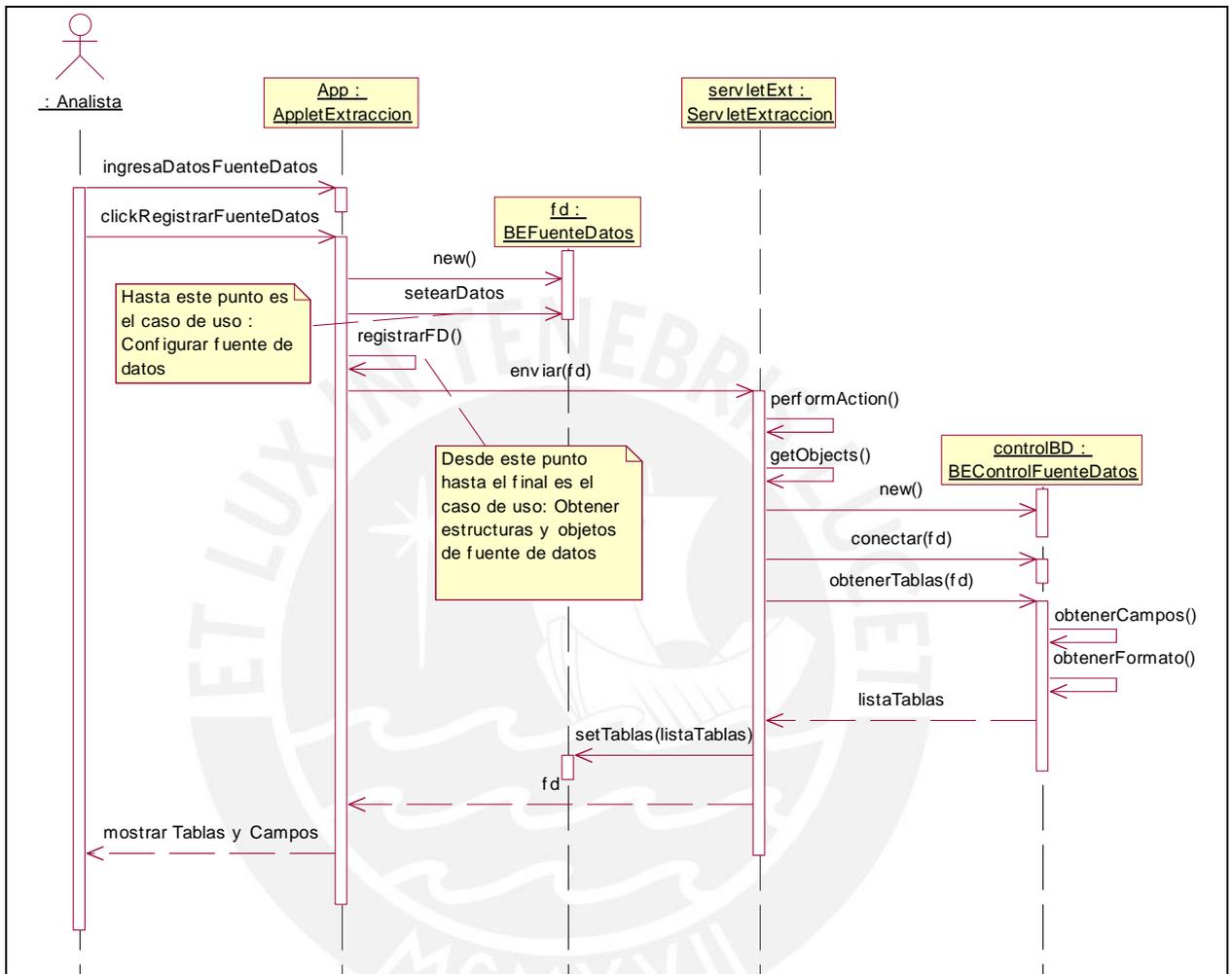
3.3. Módulo de Extracción

3.3.1. Diagrama de Clases

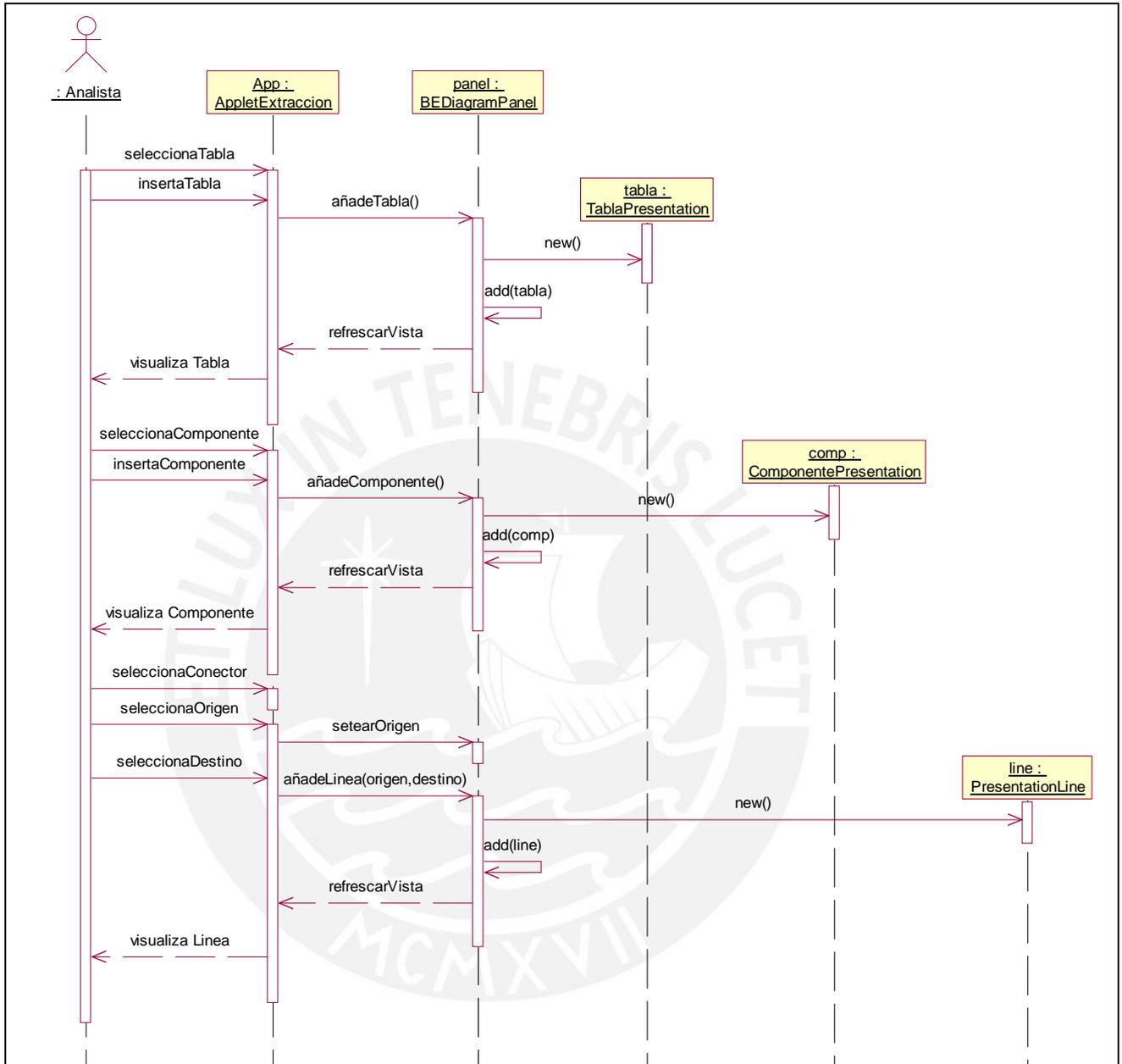


3.3.2. Diagramas de secuencias

Casos de uso “Configurar fuente de datos” y “Obtener estructuras y objetos de fuente de datos”.

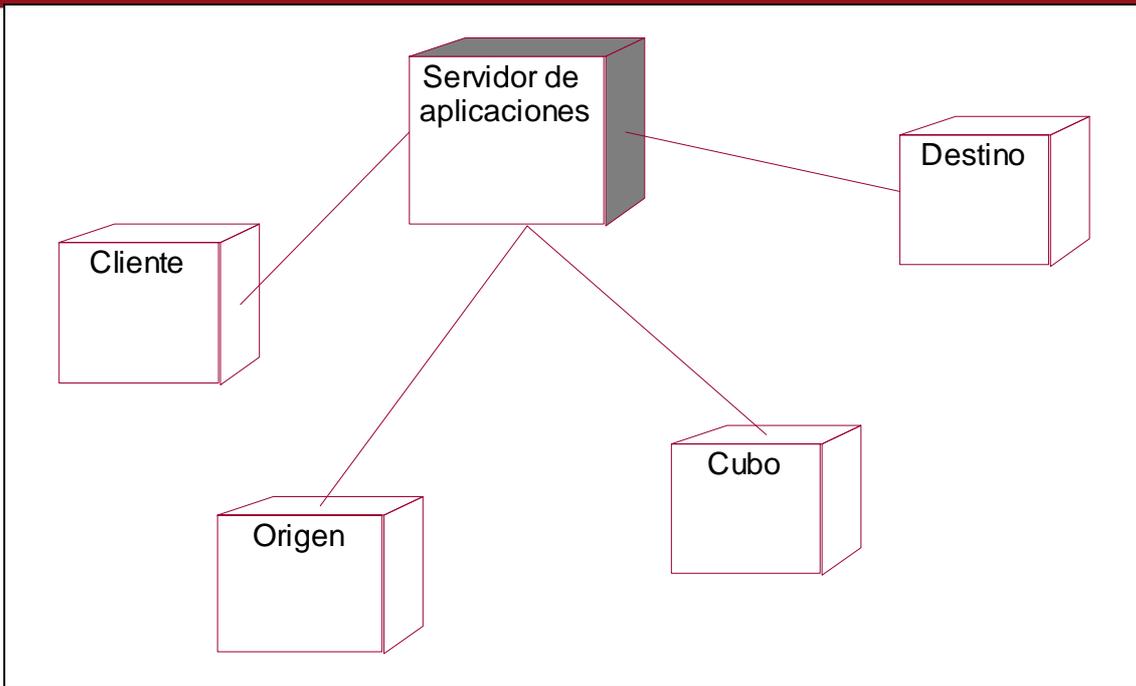


Caso de uso "Crear flujo de transformación".



4. Vista de despliegue

En esta vista se describe la configuración de red física sobre la cual se ejecuta la herramienta AXEbit. Indica los diferentes nodos físicos en los cuales se ejecuta el software y sus interconexiones.



Paquetes	
Nombre	Descripción
Servidor de aplicaciones	Es el servidor que procesa la información que envía el cliente y devuelve los resultados después de las operaciones que se realizan. Es el nexo entre los demás nodos y administra el flujo de trabajo entre ellos.
Cliente	Los usuarios de la herramienta harán uso de ésta a través de Internet por medio de un navegador que es donde se descargarán los applets para trabajar con la herramienta, además se visualizarán las páginas jsp y jsf.
Origen	Es la fuente de datos de donde se obtiene la data a ser procesada. Viene a ser un servidor de base de datos con bases de datos con la data origen.
Cubo	Guardará información de la estructura de los cubos generados para la explotación, serán utilizados por el motor OLAP para obtener la data desde las fuentes definidas para dicho cubo.
Destino	Es la fuente de datos a donde se deriva la data luego de ser procesada. Viene a ser un servidor de base de datos con bases de datos que contendrán la data transformada y la que surge de la explotación de datos.

5. Vista de implementación

Esta sección muestra el conjunto de paquetes reutilizables de cada módulo de la herramienta AXEbit, así como las respectivas interfaces que deben exponer para que se interrelacionen de una manera clara.

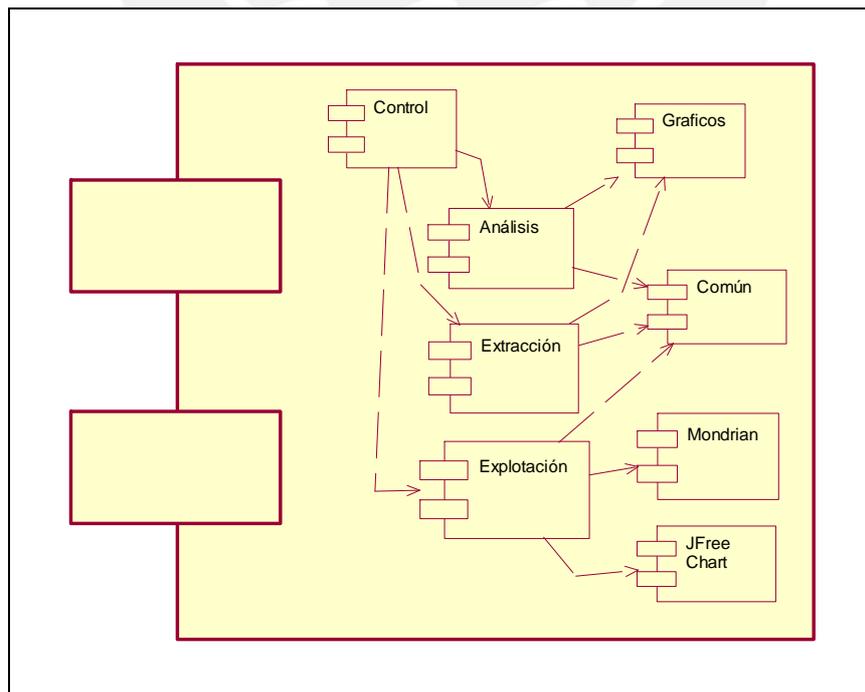
Entonces, la arquitectura del producto software esta conformado por los siguientes componentes:

- Aplicación
- Control
- Común
- Gráficos
- Análisis
- Extracción
- Explotación

Estos componentes se describen a continuación:

5.1. Componente: Aplicación

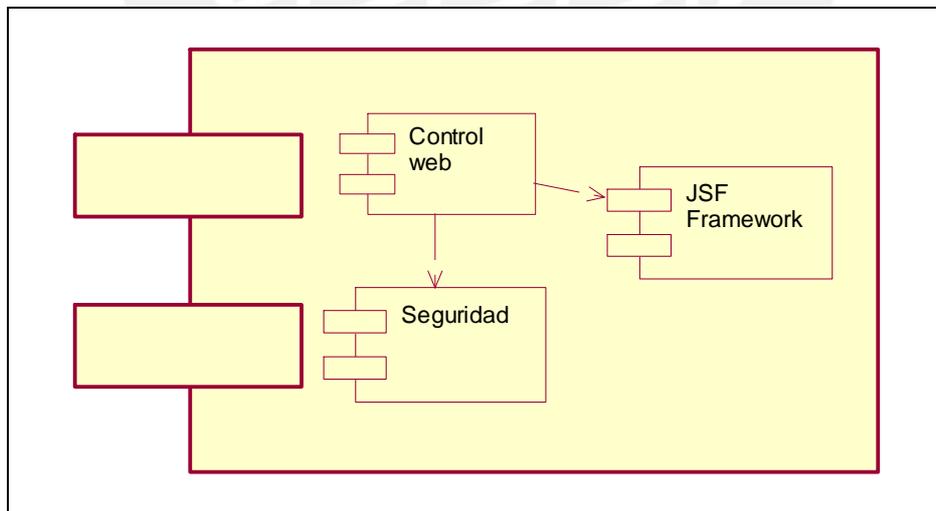
Este constituye el componente principal de la herramienta ya que contiene a todos los demás y en su interior se define el flujo de trabajo de los demás componentes. En caso de que la herramienta se instale para un nueva empresa o grupo de usuarios se deberá usar este componente para acceder a la funcionalidad de la herramienta en su conjunto.



Aplicación	
Componente	Descripción
Análisis	Este componente contiene todos los subcomponentes que se utilizan en el módulo de análisis de la herramienta.
Común	Este componente contiene todos los subcomponentes comunes que son usados en todos los módulos de la herramienta.
Control	Este componente contiene todos los subcomponentes encargados del flujo de trabajo entre los tres módulos principales de la herramienta, además permite el acceso a estos componentes.
Explotación	Este componente contiene todos los subcomponentes que se utilizan en el módulo de explotación de la herramienta.
Extracción	Este componente contiene todos los subcomponentes que se utilizan en el módulo de extracción de la herramienta.
Gráficos	Este componente contiene todos los subcomponentes necesarios para realizar los gráficos en los applets.
CeWolf / JFreeChart	Componente utilizado para realizar los gráficos de explotación (charts). JFreeChart realiza los gráficos basados en datasets, y CeWolf permite mostrarlos vía web.
Mondrian	Motor OLAP utilizado por explotación, permite la conexión al DataMart y obtener los resultados de la explotación.

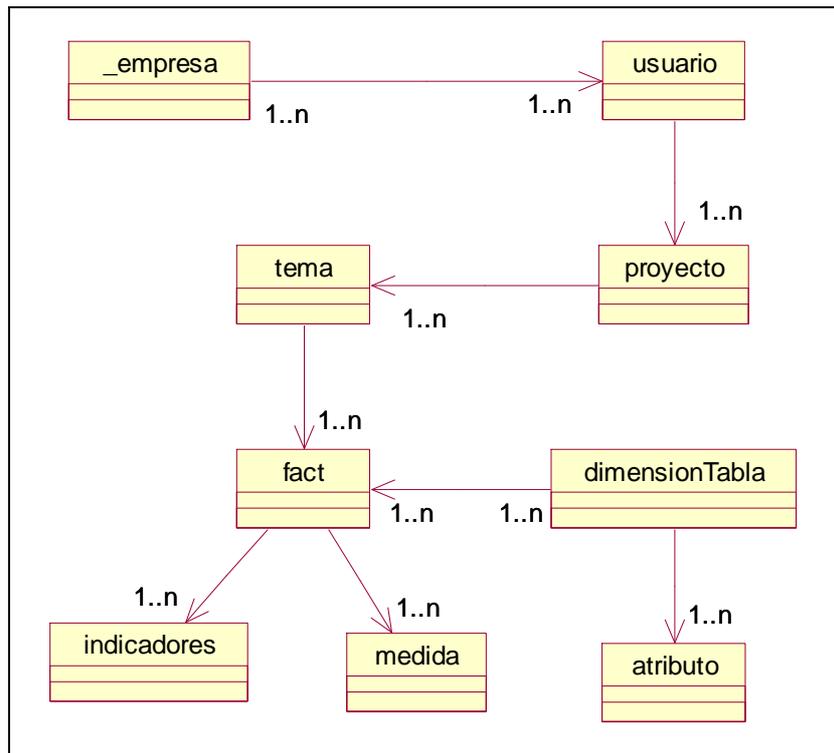
5.2. Componente: Control

Este componente contiene las clases necesarias para garantizar un acceso seguro a la herramienta, así como el framework usado en ésta, por último contiene el control web que es usado para poder interactuar con la herramienta.



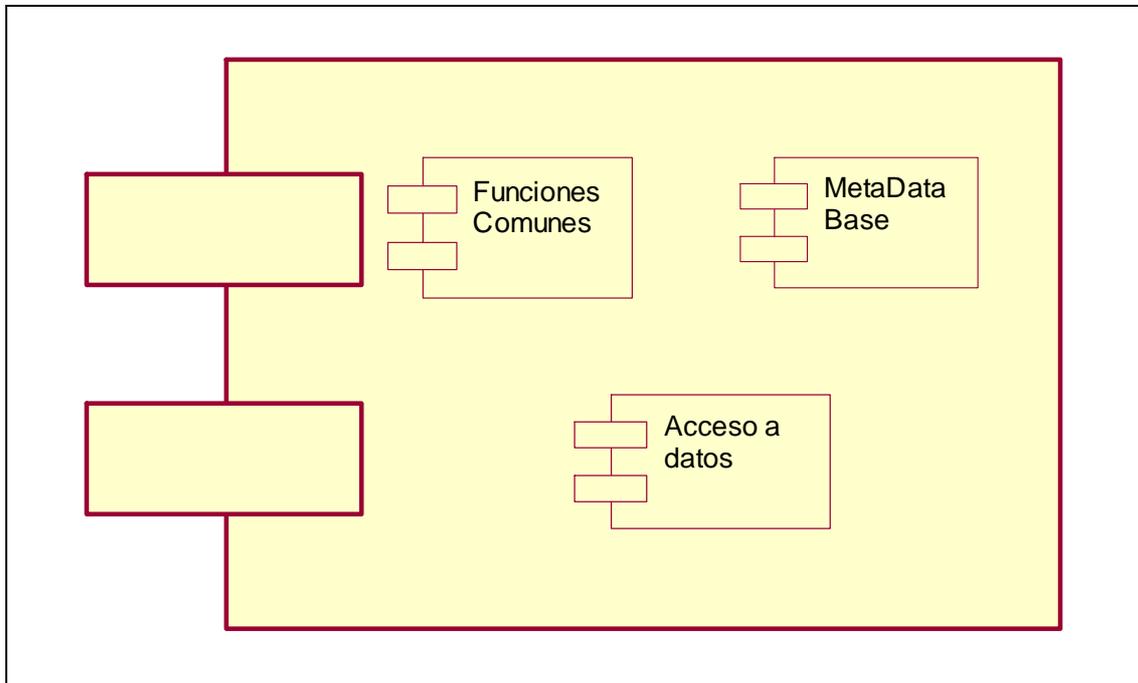
Aplicación	
Componente	Descripción
Control web	Componente utilizado para interactuar con la aplicación, implementa la interfaz web que permitirá el acceso a la lógica del negocio.
JSF Framework	Componente que maneja el flujo de navegación y comunicación con la aplicación. Implementa el patrón MVC.
Seguridad	Componente de seguridad y control de accesos. Implementará e manejo de LDAP.

Para entender mejor el diagrama de componentes anterior se tiene el siguiente diagrama de clases:



5.3. Componente: Común

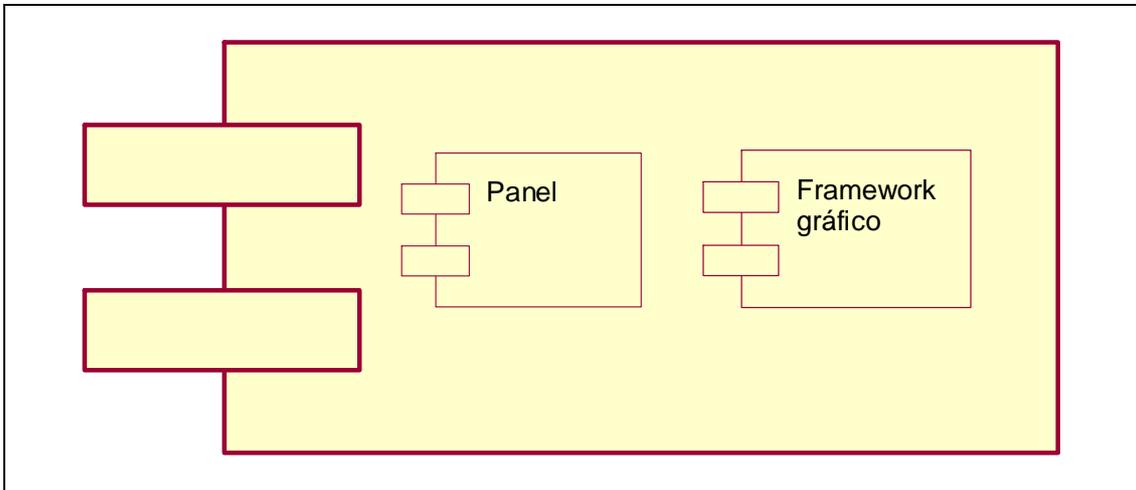
Este componente contiene todos los subcomponentes que se utilizan en todos los módulos de la herramienta.



Aplicación	
Componente	Descripción
Funciones Comunes	Este componente agrupa las clases que contienen funciones que son accedidas por todos los módulos. Ejemplo: El servlet de persistencia que se encarga de la grabación de datos en archivos XML utilizando el Castor.
MetaData Base	Este componente contiene los objetos de datos comunes a todos los módulos. Estas son las clases comunes como Tabla, Campo, Dimensión, etc.
Acceso a datos	Este componente permite a los módulos el acceso a bases de datos. Contiene los JARs con los drivers que permiten interactuar con manejadores de bases de datos y el control de conexiones a bases de datos.

5.4. Componente: Gráficos

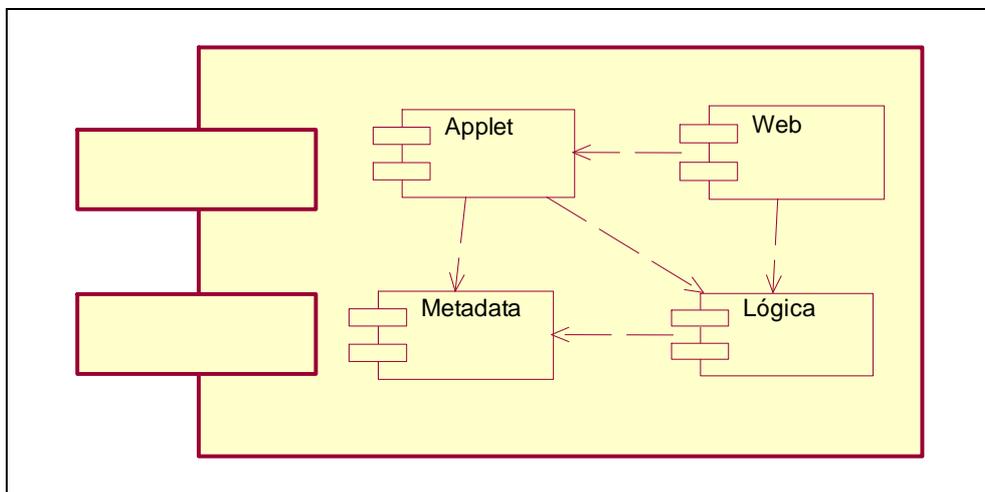
Este componente contiene todos los subcomponentes necesarios para realizar los gráficos en los applets.



Aplicación	
Componente	Descripción
Framework gráfico	Este componente contiene los elementos que representan los gráficos en el área de dibujo y tiene la implementación de acciones básicas sobre el área de dibujo como el drag and drop.
Panel	Este componente contiene el área donde se realizara el gráfico y la paleta de herramientas que permite seleccionar los elementos a graficar.

5.5. Componente: Análisis

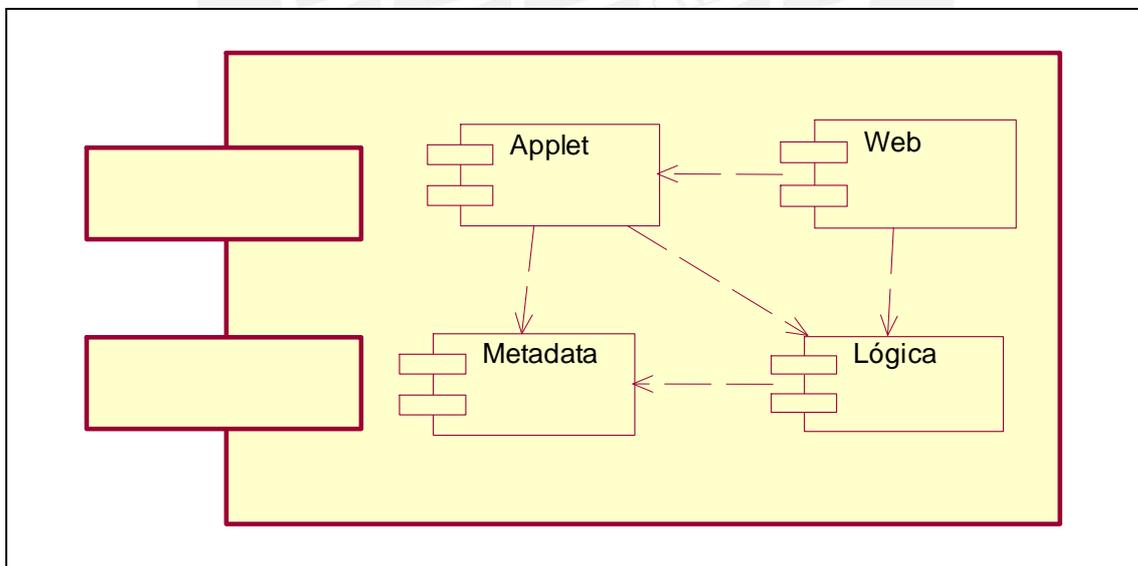
Este componente contiene todos los subcomponentes que se utilizan exclusivamente en el módulo de Análisis de la herramienta.



Aplicación	
Componente	Descripción
Applet	Este componente se descargará al equipo del cliente y permitirá realizar operaciones que gráficamente son más convenientes de realizar que empleando páginas JSF o web. El applet actuará con el sistema mediante servlets. Sus partes principales son el árbol de navegación de objetos y el área de dibujo. El modo de trabajo es muy similar al del módulo de Extracción.
Lógica	Este componente contiene las clases que tienen definida la lógica del negocio y que trabajan con la metadata. Ejemplo: El control de análisis.
Metadata	Este componente contiene los objetos que guardan la información de los elementos de análisis que hayan sido configurados por el usuario. Ejemplo: Reglas de Generación de Datos, Preferencias de Usuario, Definición de Tipos de Datos, etc. También contiene los beans de persistencia utilizados para enviar los datos de los objetos al servlet de persistencia el cual guarda estos datos en archivos XML.
Web	Este componente está conformado por los servlets y páginas JSF alojadas en el servidor.

5.6. Componente: Extracción

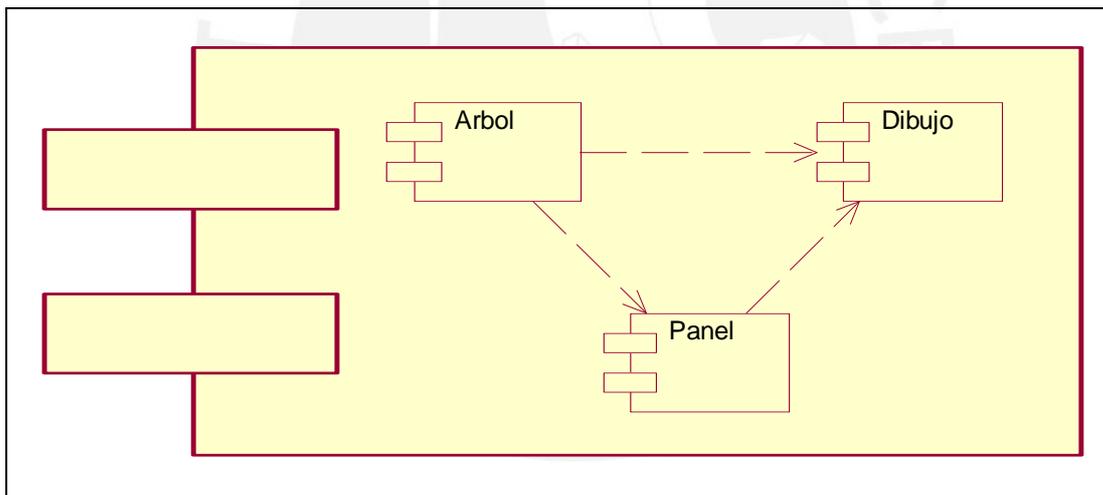
Este componente contiene todos los subcomponentes que se utilizan en el módulo de extracción de la herramienta.



Aplicación	
Componente	Descripción
Applet	Este componente se descargará al equipo del cliente y permitirá realizar operaciones que requieren uso de gráficos que no puedan ser realizadas por medio de formularios web o paginas JSF. Interactuará con un servlet ubicado en el lado del servidor, el cual recibirá los datos de las operaciones realizadas en el applet. Sus partes principales son el árbol de navegación de objetos y el área de dibujo.
Lógica	Este componente contiene las clases que tienen definida la lógica del negocio y que trabajan con la metadata. Ejemplo: El control de extracción y el ejecutor de jobs programados.
Metadata	Este componente contiene las clases de dominio, que se usan para la lógica del negocio, y las clases que representan el mapeo correspondiente para su persistencia y recuperación en cualquier momento.
Web	Este componente esta conformado por los servlets, las páginas JSF alojadas en el servidor, los beans que representan las páginas y los archivos de recursos.

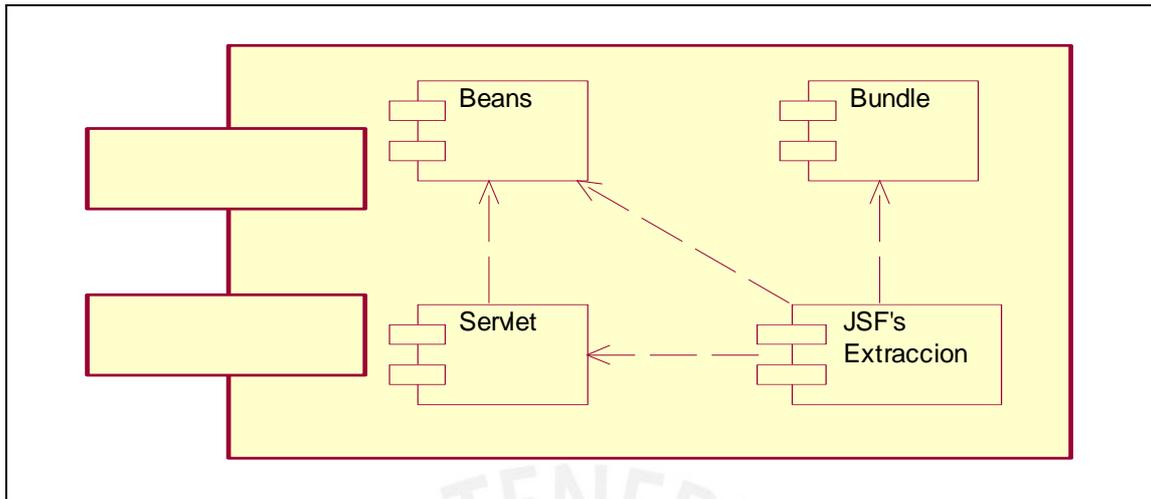
Estos subcomponentes se subdividen según lo siguiente:

5.6.1. Componente: Applet



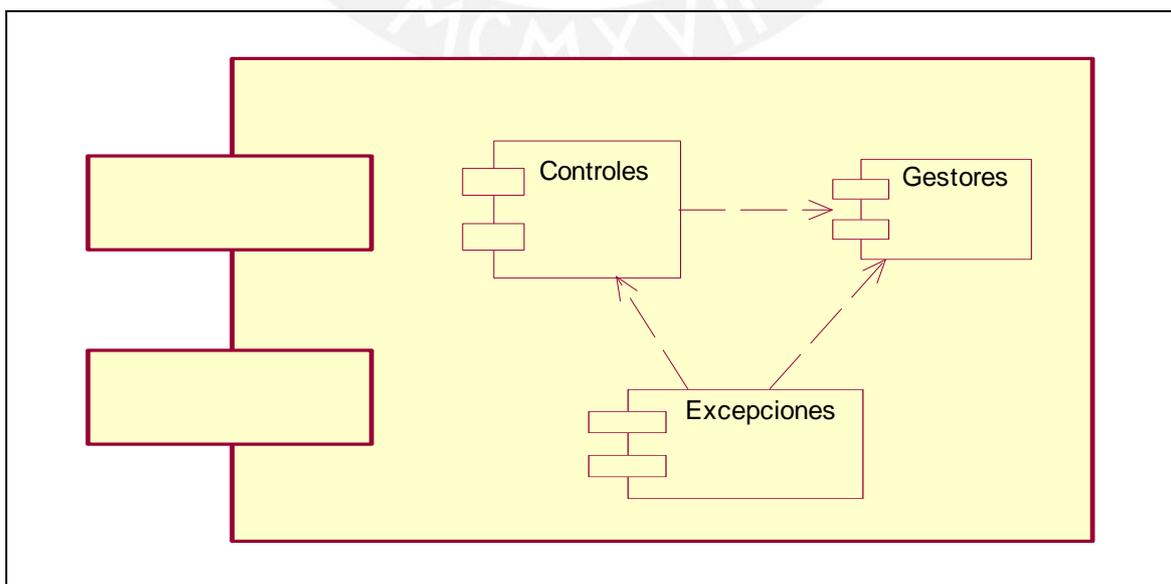
Aplicación	
Componente	Descripción
Arbol	Este componente contiene las clases necesarias para manejar el árbol de navegación de objetos que contendrá la estructura de los proyectos que se muestra en el applet. Se encarga de definir los menús y las operaciones que realiza el usuario en dicha área.
Dibujo	Este componente contiene las clases que definen como se dibujarán los objetos sobre el área de dibujo y las operaciones a realizar con ellos. Ejemplo: La clase del objeto activo Script : BEScriptDib
Panel	Este componente contiene los objetos que controlan la forma como se muestra el área de dibujo, por ejemplo se encarga de la funcionalidad del deshacer y rehacer con las acciones que realiza el usuario sobre el área de dibujo en el applet.

5.6.2. Componente: Web



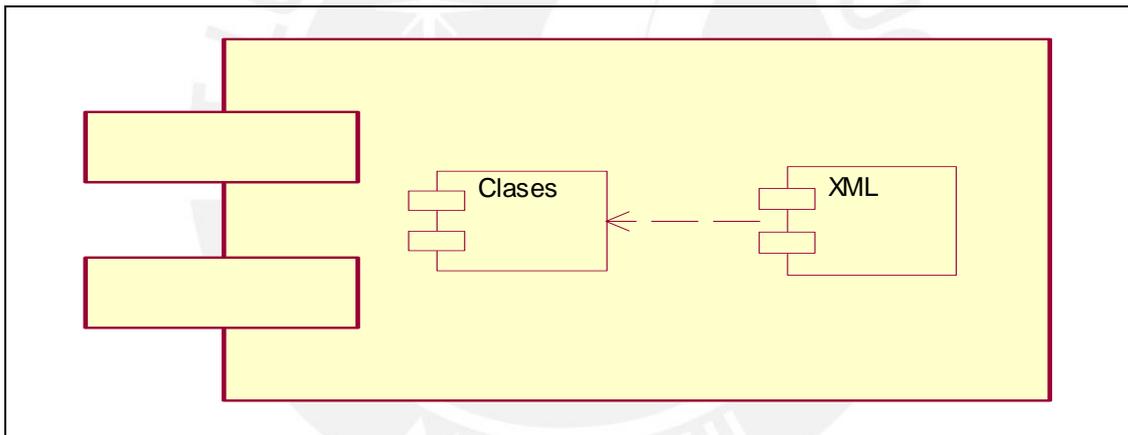
Aplicación	
Componente	Descripción
Beans	Este componente contiene las clases de los beans de respaldo de las páginas JSF. Estos beans definen las propiedades y las lógicas de manejo asociadas con los componentes de interfaz gráfica utilizados en la página.
Servlet	Este componente contiene las clases que definen las operaciones que debe realizar el servidor en la interacción con las páginas JSF. Ejemplo: El servlet de extracción aparte de ejecutar las programaciones, recibe solicitudes de las páginas JSF para saber el estado de ejecución de los jobs.
Bundle	Este componente contiene los archivos de recursos de las páginas JSF, por ejemplo para la internacionalización de los mensajes.
JSF's Extraccion	Este componente contiene las páginas JSF, que se mostrarán del lado del cliente en el navegador Web.

5.6.3. Componente: Lógica



Aplicación	
Componente	Descripción
Controles	Este componente contiene las clases que se encargarán de controlar las operaciones y la comunicación entre las distintas clases, que se realizan en la capa de la lógica del negocio en el sistema. Ejemplo: BEControlEjecución, clase que maneja la ejecución de una transformación.
Gestores	Este componente contiene las clases que manejan la lógica de los objetos del dominio. Ejemplo: BEGestorJob, clase que maneja las operaciones a realizar sobre los jobs que pertenecen a un proyecto.
Excepciones	Este componente contiene las clases de las excepciones personalizadas que se manejan en el sistema para controlar los distintos errores que podría presentar la aplicación.

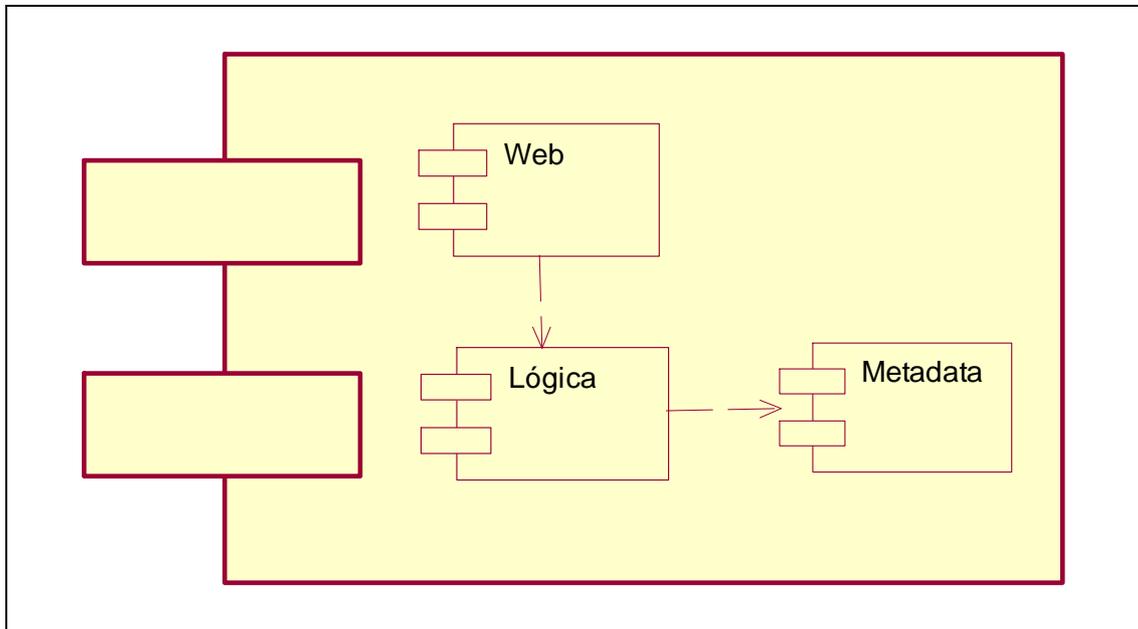
5.6.4. Componente: Metadata



Aplicación	
Componente	Descripción
Clases	Este componente contiene los objetos que guardan la información de los elementos de extracción que hayan sido configurados por el usuario. Ejemplo: clases de datos de extracción tales como Transformación, Fuente de Datos, etc.
XML	Este componente contiene los beans de persistencia utilizados para enviar los datos de los objetos al servlet de persistencia el cual guarda estos datos en archivos XML.

5.7. Componente: Explotación

Este componente contiene todos los subcomponentes que se utilizan en el módulo de explotación de la herramienta.



Aplicación	
Componente	Descripción
Lógica	Este componente contiene las clases que tienen definida la lógica de la explotación y que trabajan con la metadata. Esto incluye el manejo de reportes, gráficos (charts) y las clases encargadas de comunicarse con el motor OLAP. También contiene los beans de persistencia utilizados para enviar los datos de los objetos al servlet de persistencia el cual guarda estos datos en formato XML.
Metadata	Este componente contiene los objetos que guardan la información de los elementos de explotación configurados por el usuario. Incluye información de cubos y objetos de explotación (esquemas, reportes, filtros, etc).
Web	Este componente esta conformado por los servlets y paginas JSF alojadas en el servidor.

6. Requerimientos de desempeño

La arquitectura de software del sistema satisface los requerimientos clave de tamaño y desempeño:

- El sistema permitirá que las transacciones que se realicen sean seguras, de modo que ningún ente externo al sistema pueda acceder a la información que éstas manejen. Para tal fin se definirán varios niveles de acceso.
- El sistema permitirá el acceso concurrente de varios usuarios, de modo que puedan realizar transacciones simultáneas.
- Las computadoras personales del lado del cliente deberán contar con un procesador Pentium III, 128 MB de memoria RAM, acceso a Internet y un navegador Web.
- El módulo de extracción requerirá de un mínimo de 1Gigabyte de RAM en el servidor principal. Los requerimientos del servidor se incrementarán de acuerdo con la capacidad que se quiera dar al mismo, es decir, cuantos flujos simultáneos se necesita que pueda ejecutar.

ANEXO G. DOCUMENTO DE XML

1. Introducción

1.1 Objetivos

El presente documento forma parte del diseño final de la herramienta AXEbit, en éste se muestra el esquema XML usado para almacenar la información persistente requerida dentro de la herramienta. Se provee la descripción de la estructura de los archivos XML usados en la herramienta AXEbit, de esta forma se logrará transmitir una visión más clara sobre la forma en que se trabajo la construcción de la herramienta.

1.2 Alcance

El documento describirá el esquema XML usado dentro de la herramienta AXEbit para almacenar la información, indicando las carpetas en las cuales se distribuirá los archivos y la estructura de un archivo de ejemplo.

1.3 Visión general del documento

Este documento consta de dos secciones. Esta sección es la introducción y proporciona una visión general del Documento de XML. En la siguiente sección se detalla el esquema usado en la herramienta para el almacenamiento de información.

2. Esquema XML

En esta sección se muestra el detalle de la estructura usada para almacenar la información en archivos XML, comenzando por la estructura de carpetas y mostrando además el contenido de un archivo XML de ejemplo.

2.1 Estructura de carpetas XML

Las carpetas usadas para almacenar los archivos XML son las siguientes:

jobs
fuentedatos
paquetes
procesos
scripts
archivos
tablas
proyectos
empresas
logs
fusuario

programación

- **Jobs:** Contiene los archivos XML de los jobs configurados en la herramienta
- **FuenteDatos:** Contiene los archivos XML que describen las fuentes de datos configuradas en la herramienta.
- **Paquetes:** Contiene los archivos XML que describen los paquetes configurados en la herramienta, incluyendo referencias a los componentes internos del mismo (objetos activos y objetos pasivos)
- **Procesos:** Contiene los archivos XML que describen los componentes activos tipo proceso que pertenecen a los paquetes de la herramienta. Cada XML contiene además la información de los subcomponentes configurados para cada proceso. (filtro, transformación, estandarización y lookup)
- **Scripts:** Contiene los archivos XML que describen los scripts configurados en la herramienta.
- **Archivos:** Contiene los archivos XML que describen los archivos planos que sirven como origen y destino de datos para los paquetes configurados en la herramienta.
- **Tablas:** Contiene los archivos XML que describen las tablas de base de datos que sirven como origen y destino de datos para los paquetes configurados en la herramienta.
- **Proyectos:** Contiene los archivos XML que describen los proyectos de inteligencia de negocios configurados en la empresa.
- **Empresas:** Contiene los archivos XML que describen a las empresas con las cuales se trabaja en la herramienta.
- **Logs:** Contiene los archivos XML que contienen la información de ejecución de los jobs de extracción.
- **Fusuario:** Contiene los archivos XML que describen las funciones definidas por el usuario en la herramienta.
- **Programacion:** Contiene los archivos XML que contienen la información de programación para la ejecución de los jobs de la herramienta.

2.2 Ejemplo archivo XML

```

<?xml version="1.0" encoding="UTF-8" ?>
- <BEFuenteDatosBean tipo-bD="0" id="1154479273390113">
  <base-datos>axe</base-datos>
  <contra>biex</contra>
  <usuario>biex</usuario>
  <id-tablas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="java:java.lang.String">1154479284078649</id-tablas>
  <id-tablas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="java:java.lang.String">1154479284562499</id-tablas>
  <id-tablas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="java:java.lang.String">1154479284593474</id-tablas>
  <id-tablas xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:type="java:java.lang.String">1154479284609509</id-tablas>
  <cadena-conexion>localhost</cadena-conexion>
  <descripcion>maparam</descripcion>
  <fecha-modificacion>2006-08-01T19:41:13.390-05:00</fecha-modificacion>
  <fecha-creacion>2006-08-01T19:41:13.390-05:00</fecha-creacion>
  <nombre>fdParam</nombre>
</BEFuenteDatosBean>

```

Todos los archivos XML tendrán como mínimo los siguientes atributos:

- id
- nombre
- descripcion
- fechaCreacion
- fechaModificacion

ANEXO H. DIAGRAMAS DE SECUENCIA

1. Introducción.

El presente documento “Diagramas de Secuencias” contiene los diagramas de secuencias para las funcionalidades que serán implementadas en la construcción de la herramienta AXEbit correspondientes al módulo de extracción. Los diagramas se han elaborado utilizando el lenguaje de modelamiento UML.

1.1. Propósito

El objeto del presente documento es definir la interacción entre las clases de diseño para la implementación de las funcionalidades que cubrirán los requisitos establecidos para la herramienta que se desea construir.

Se busca mostrar claramente las dependencias entre clases y las llamadas que se hacen, así como el flujo global de datos entre las mismas, de manera que el equipo desarrollador pueda realizar la etapa de construcción de forma ordenada. Además, mediante el uso de estos diagramas se facilita la tarea de ubicar los posibles puntos en que se produce un error en el software sin necesidad de estar revisando línea por línea el código fuente.

1.2. Visión General del Documento

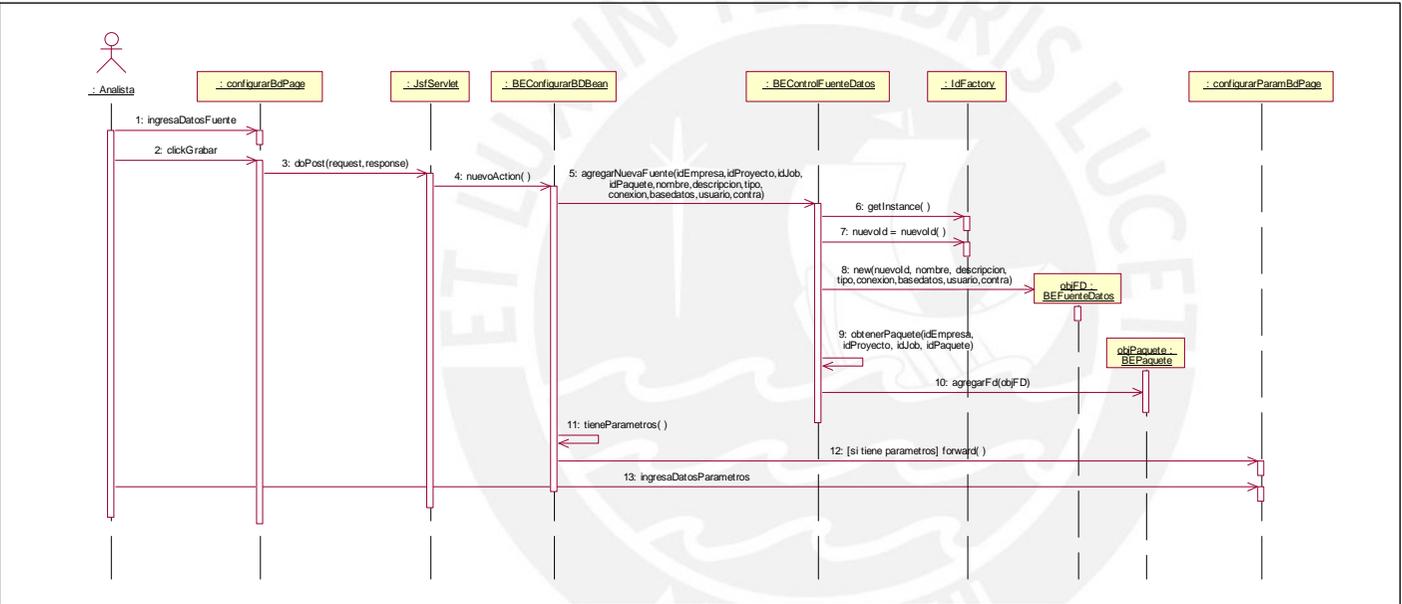
Este documento consta de 2 partes. Esta primera parte introductoria da una idea general de lo que se busca lograr con el uso de este documento.

La segunda parte muestra en detalle los diagramas de secuencias para las funcionalidades a ser implementadas.

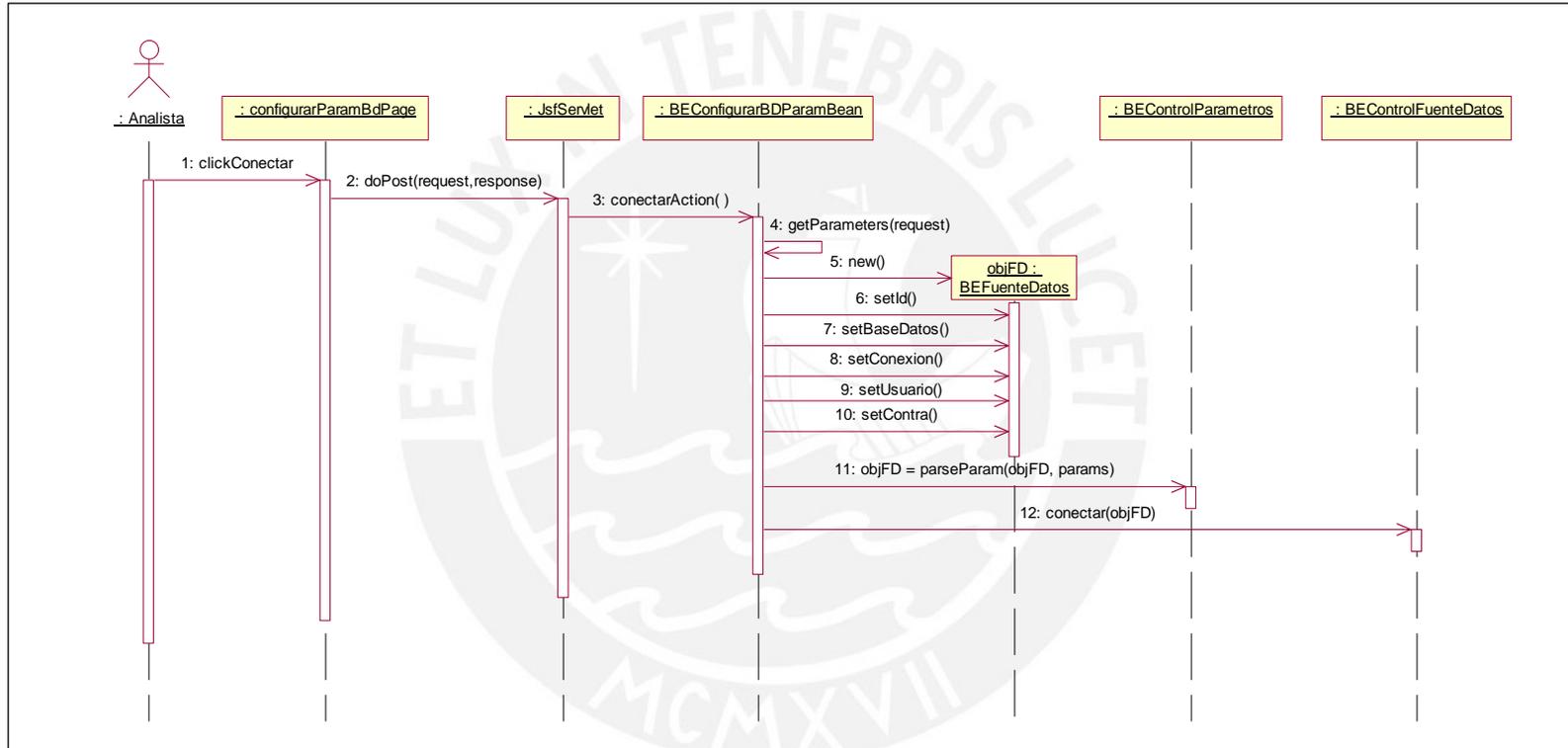
2. Diagramas de Secuencias del módulo de extracción

2.1. Paquete Comunicación

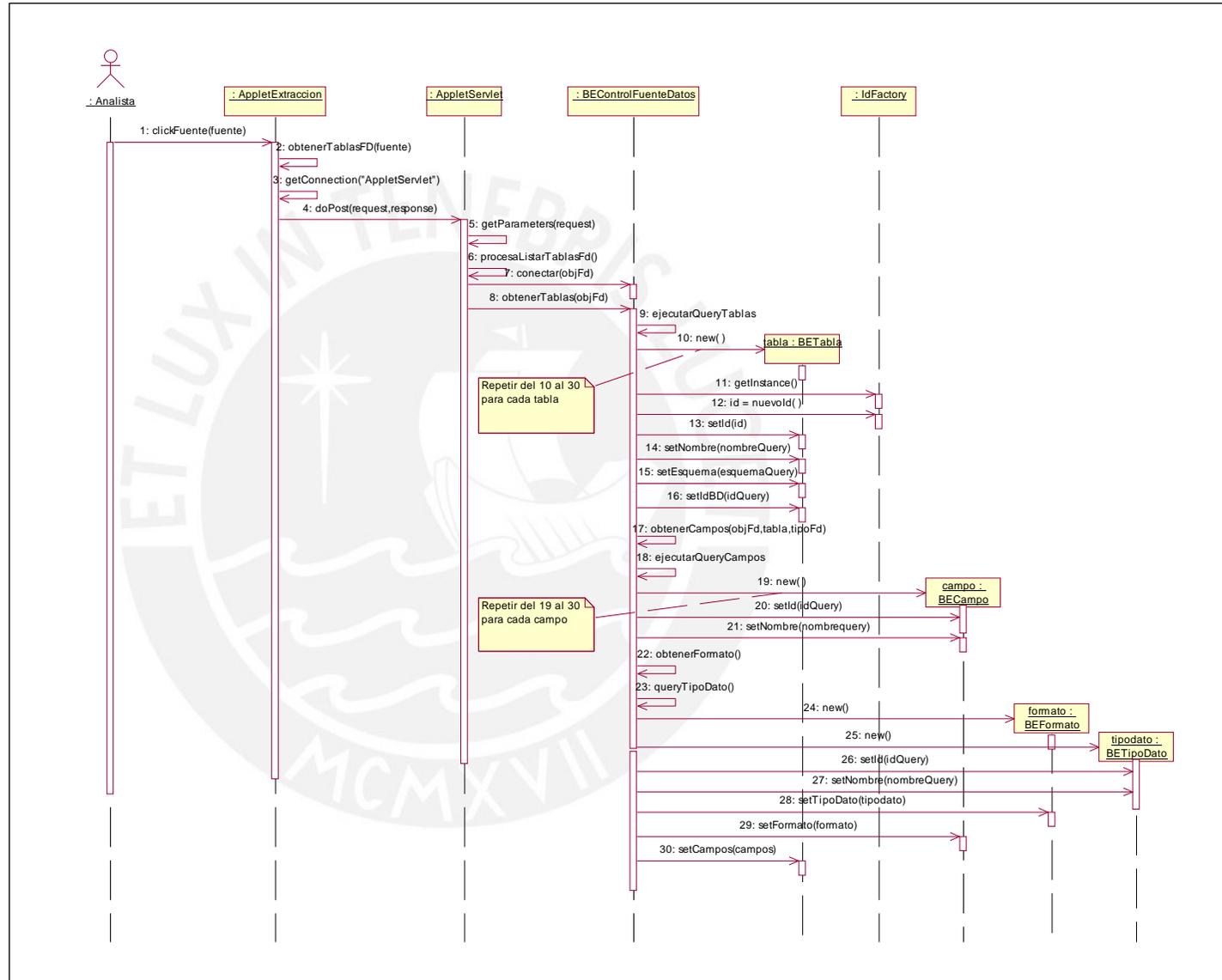
2.1.1. Configurar fuente de datos – Nueva Fuente de datos



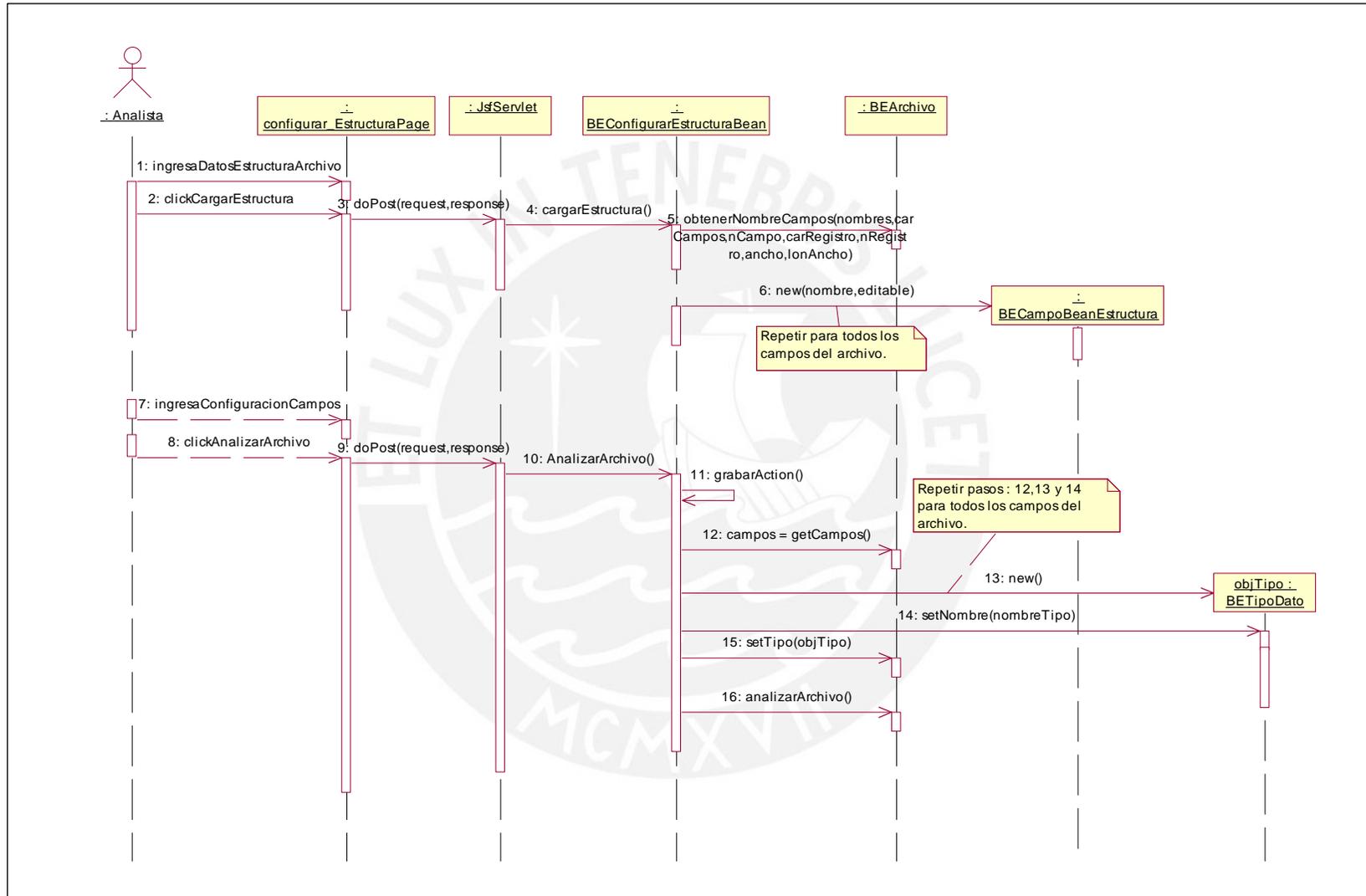
2.1.2. Conectar a fuente de datos



2.1.3. Obtener estructuras y objetos de fuente de datos

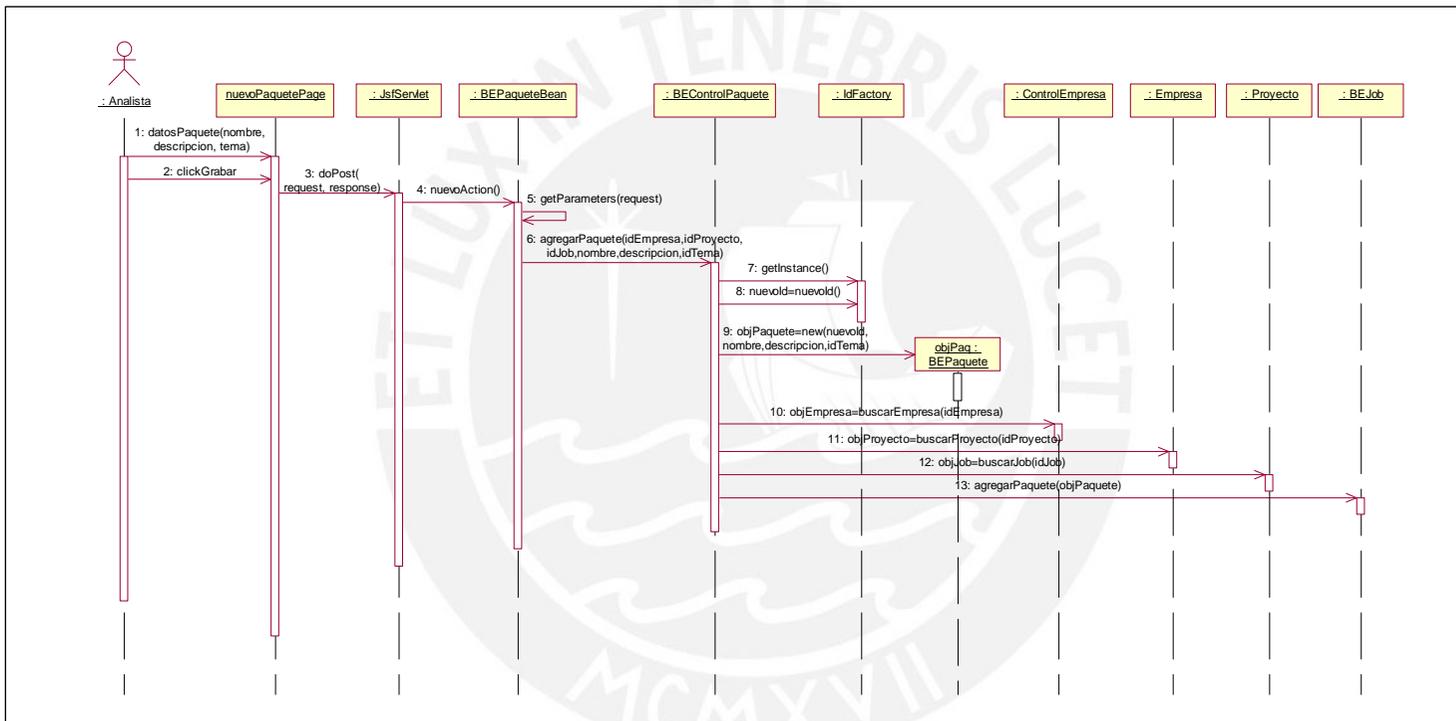


2.1.4. Configurar estructura de archivo plano

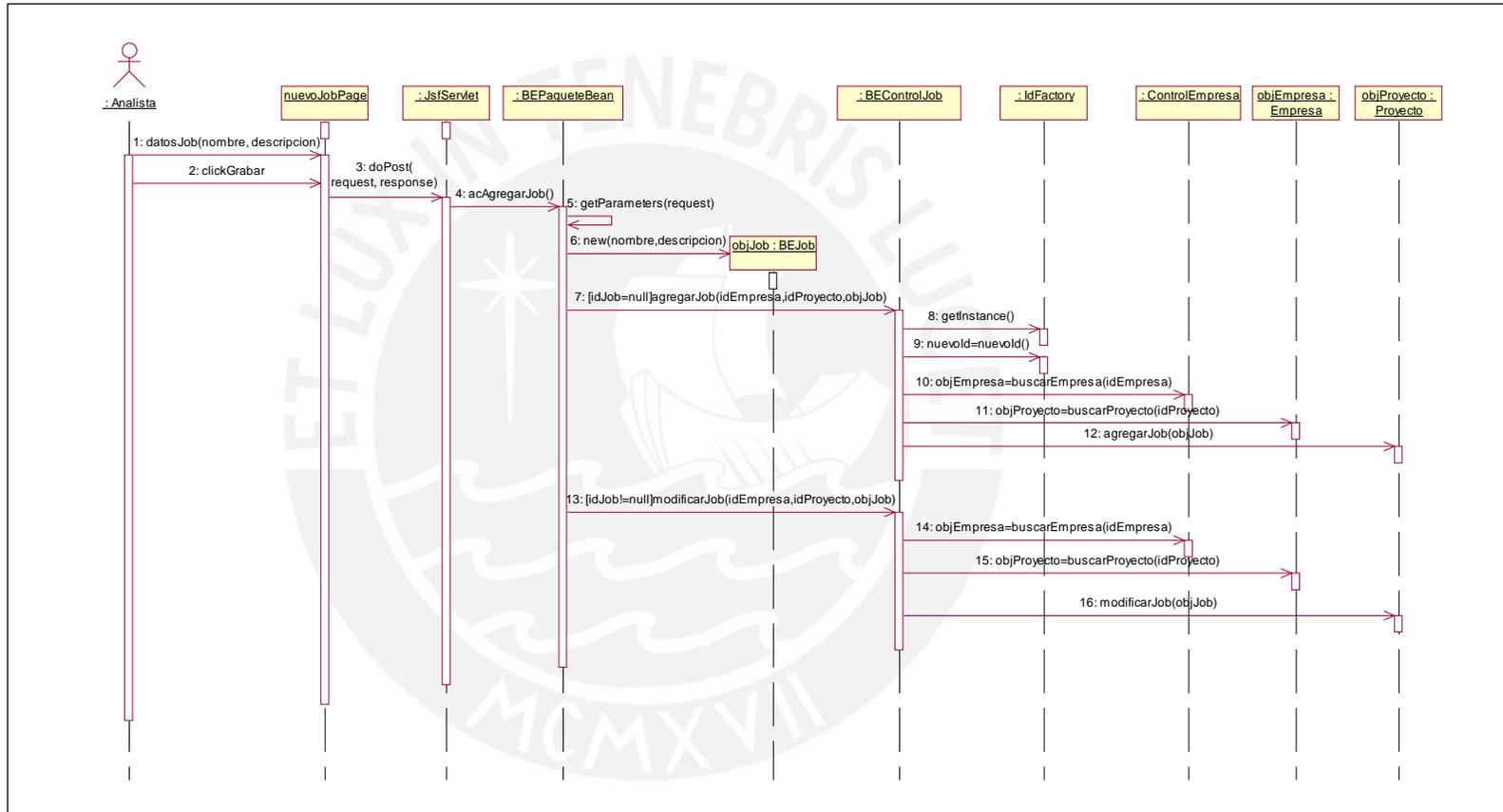


2.2. Paquete Extracción

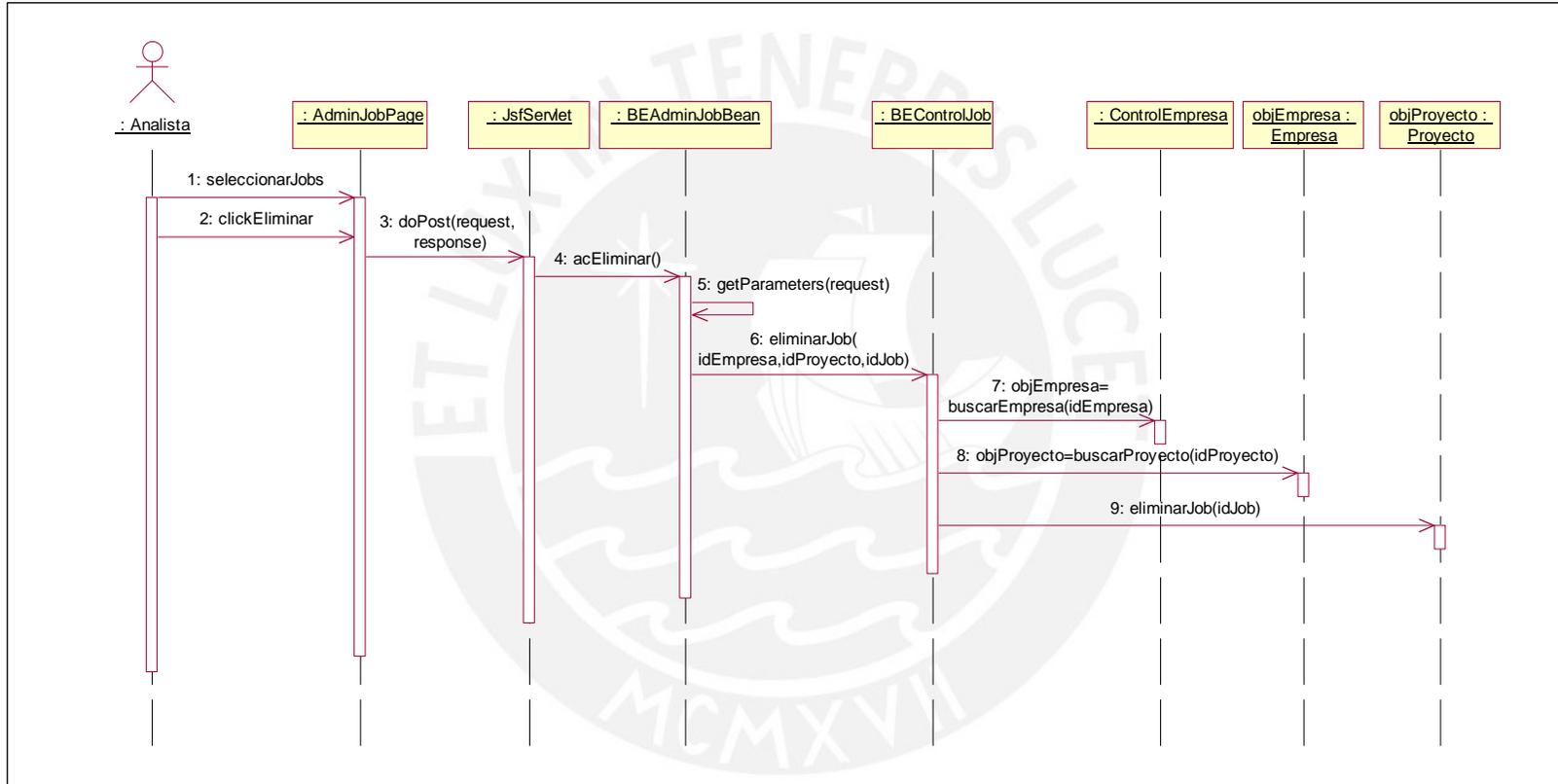
2.2.1. Crear Paquete – Nuevo Paquete



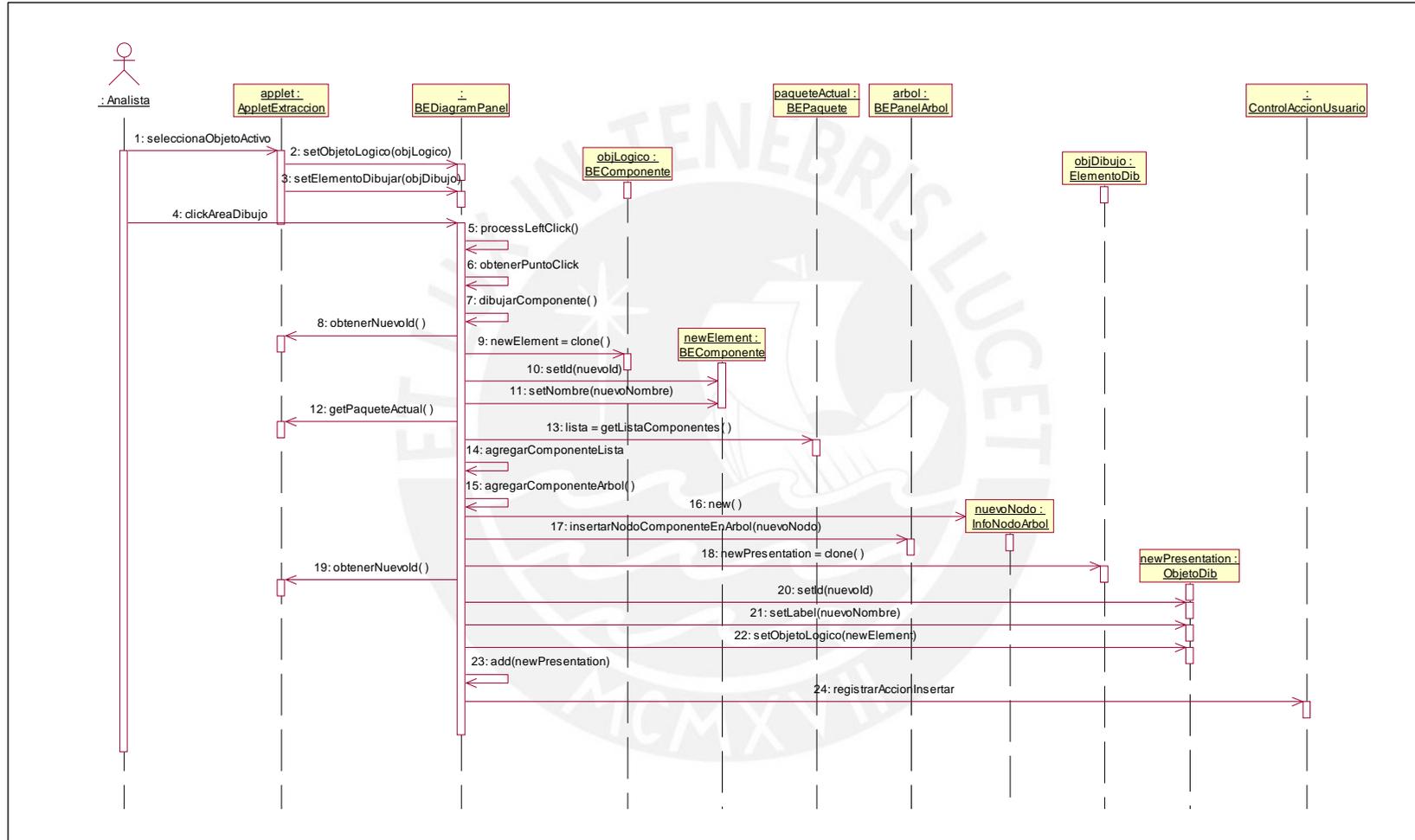
2.2.2. Administrar Job – Nuevo/Modificar Job



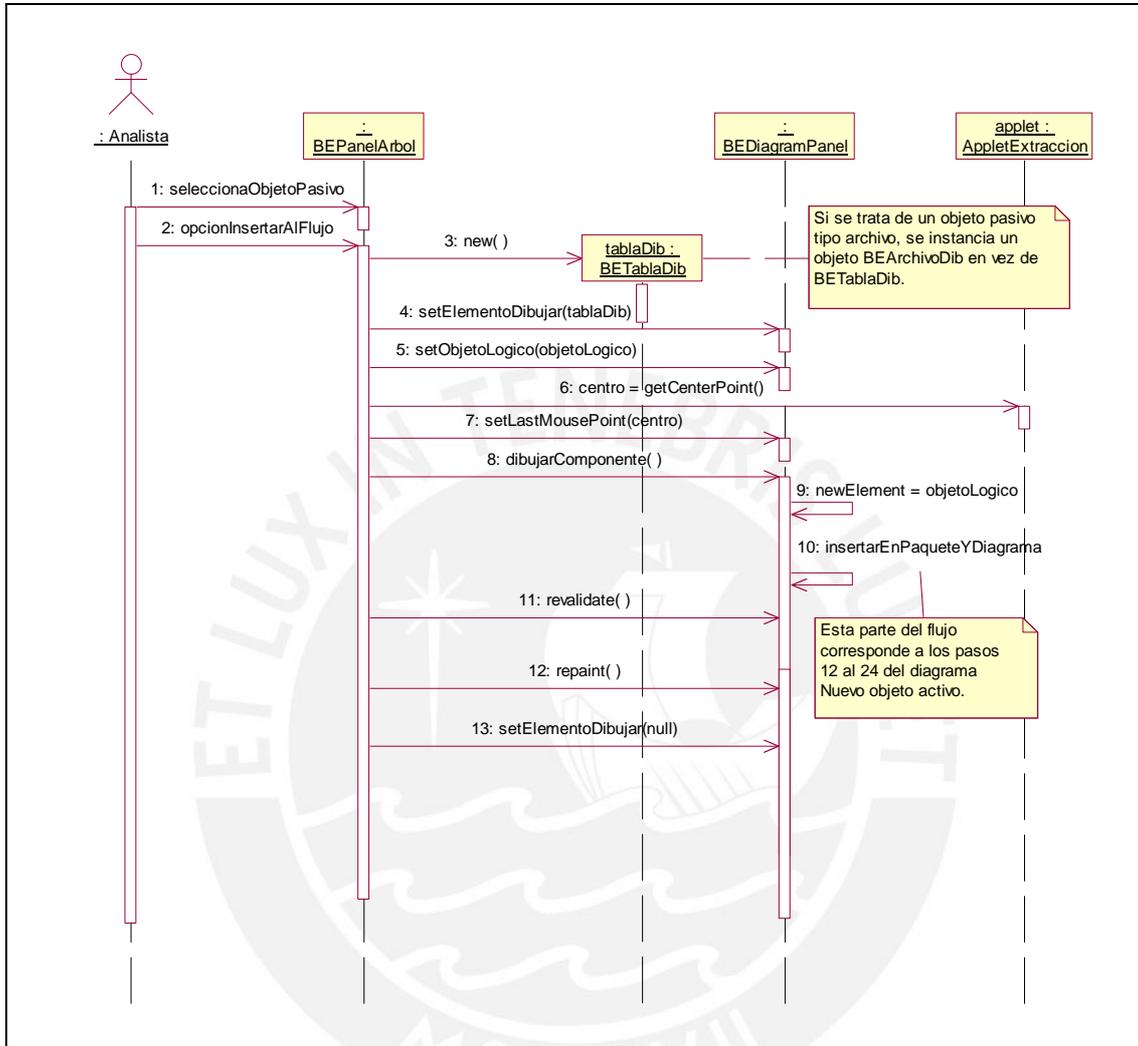
2.2.3. Administrar Job – Eliminar Job



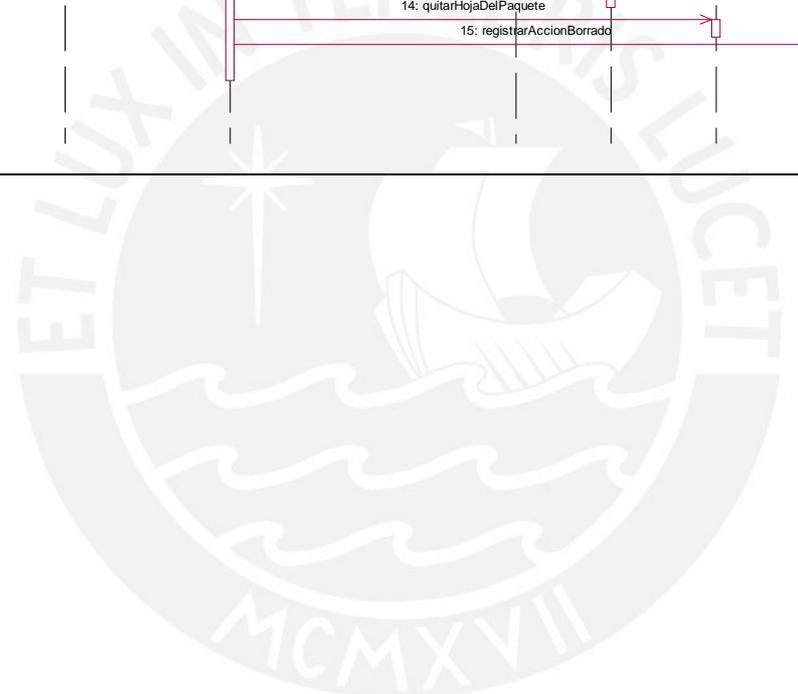
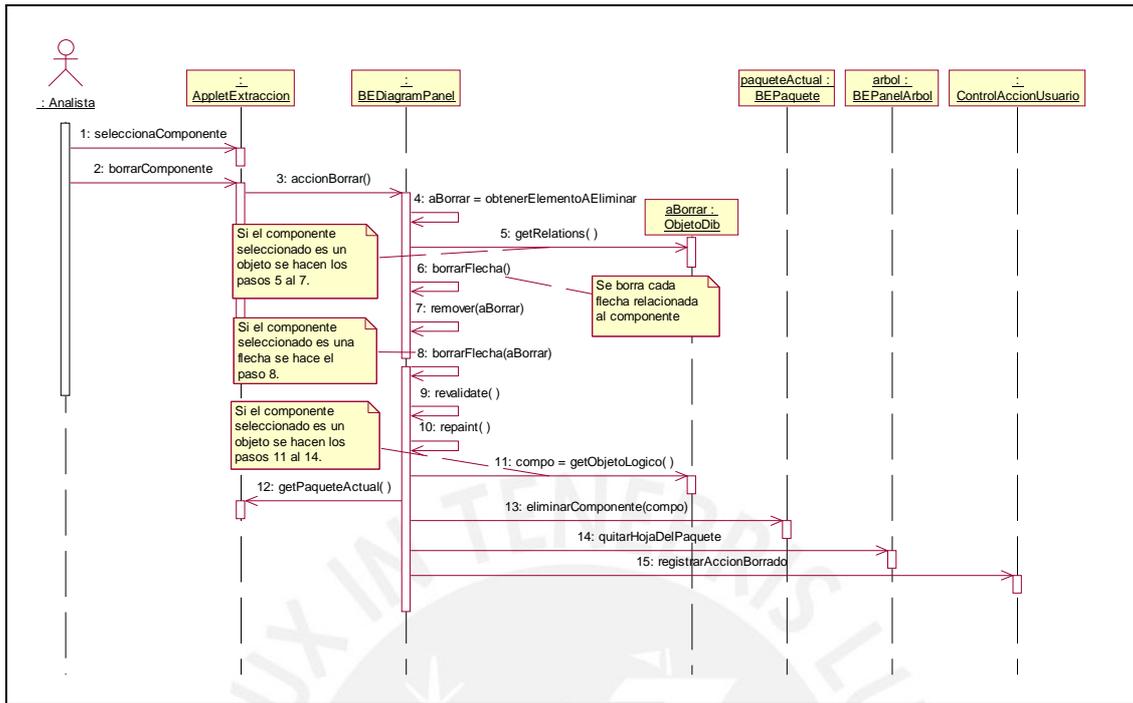
2.2.4. Crear Flujo de Transformación – Nuevo objeto activo



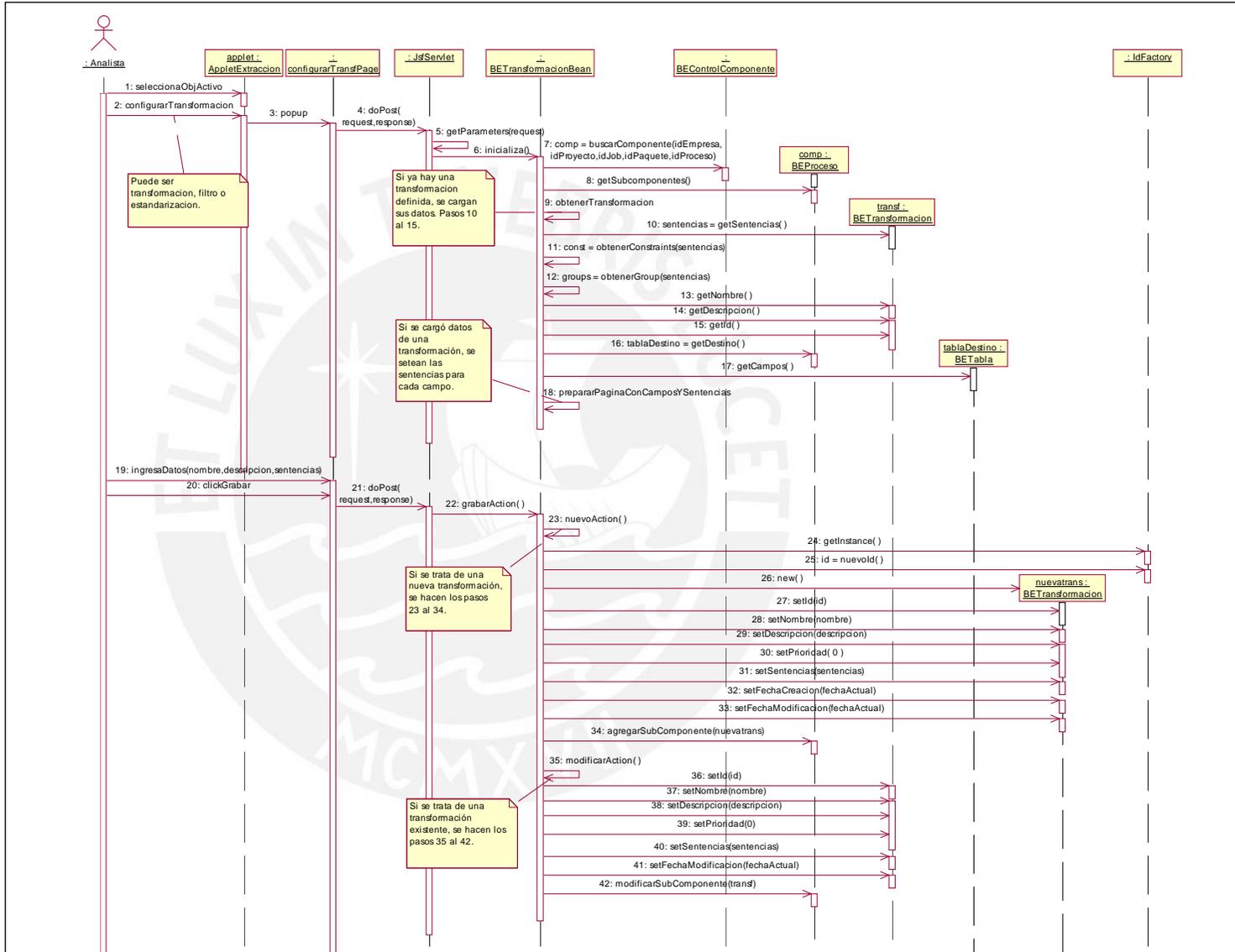
2.2.5. Crear Flujo de Transformación – Nuevo objeto pasivo



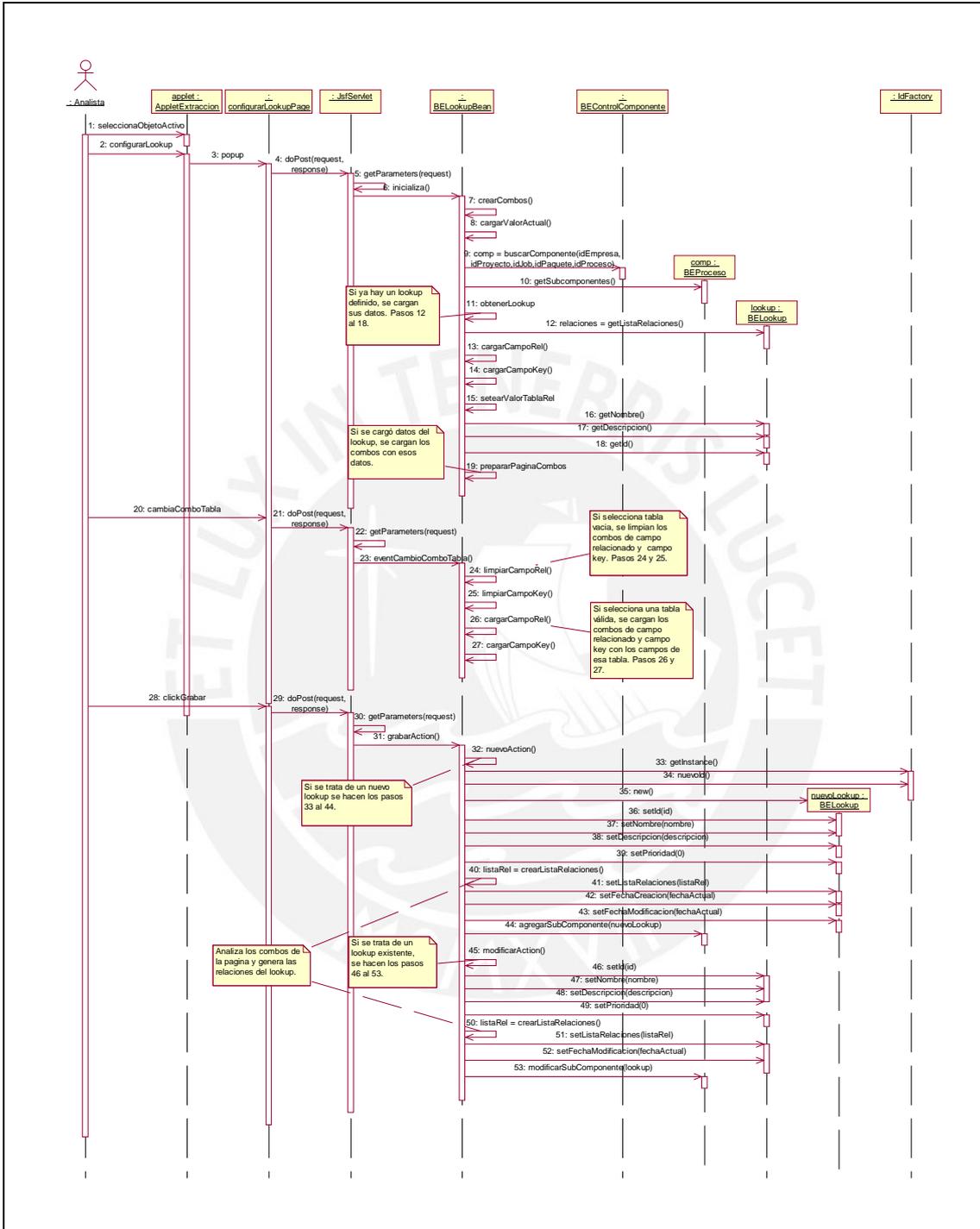
2.2.6. Crear Flujo de Transformación – Eliminar componente



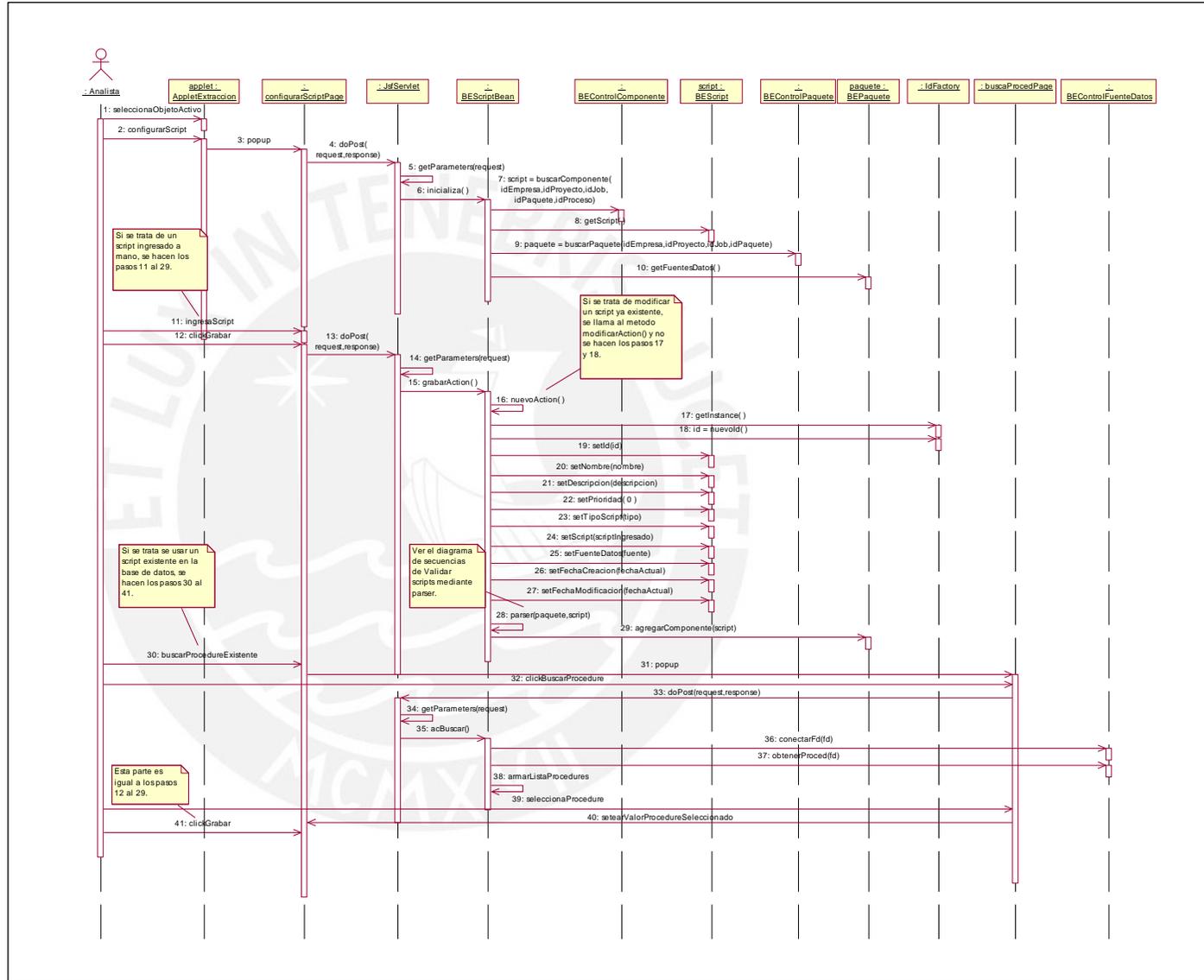
2.2.7. Personalizar Objetos Activos – Filtro, Transformación y Estandarización.



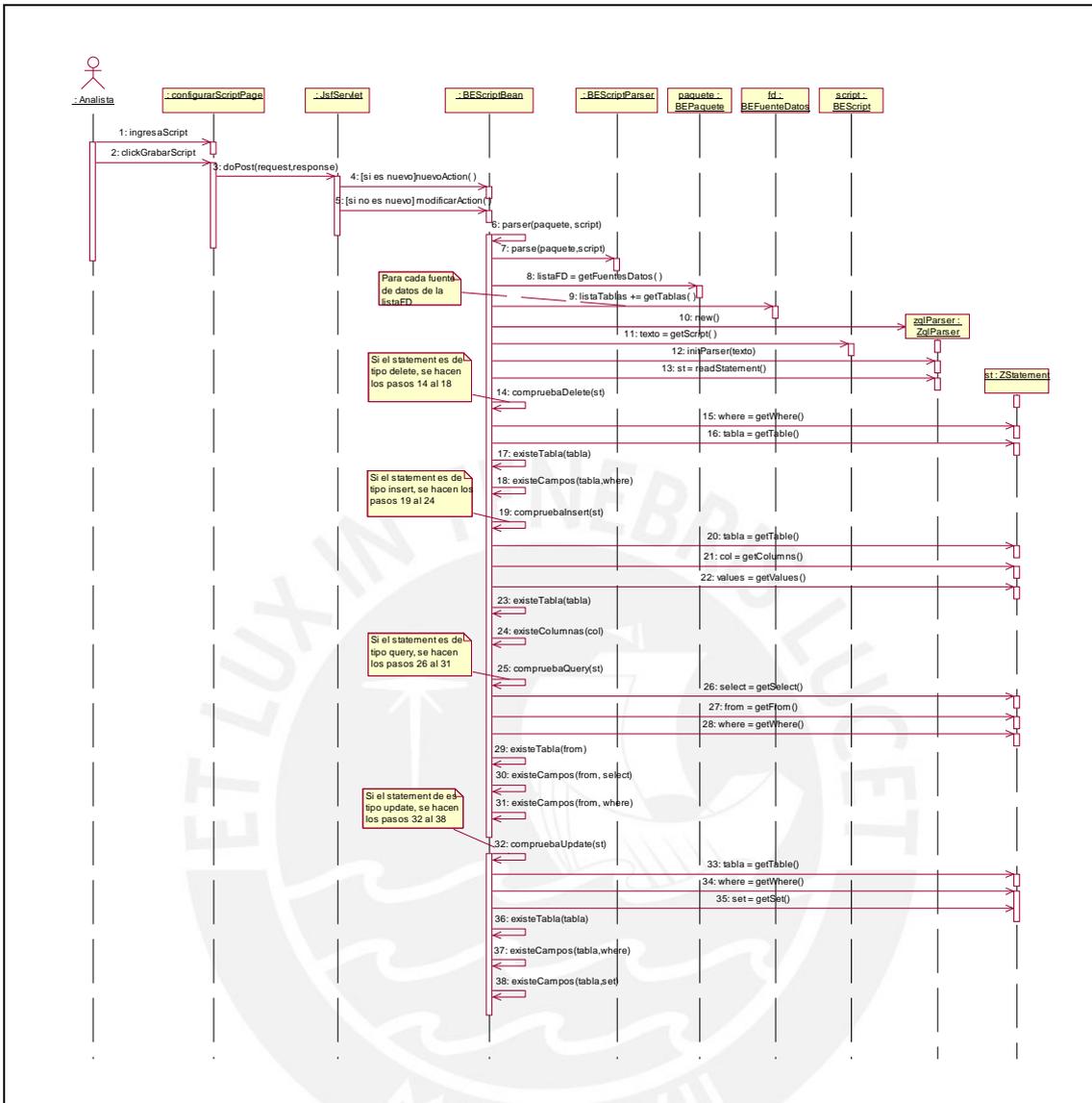
2.2.8. Personalizar Objetos Activos – Lookup

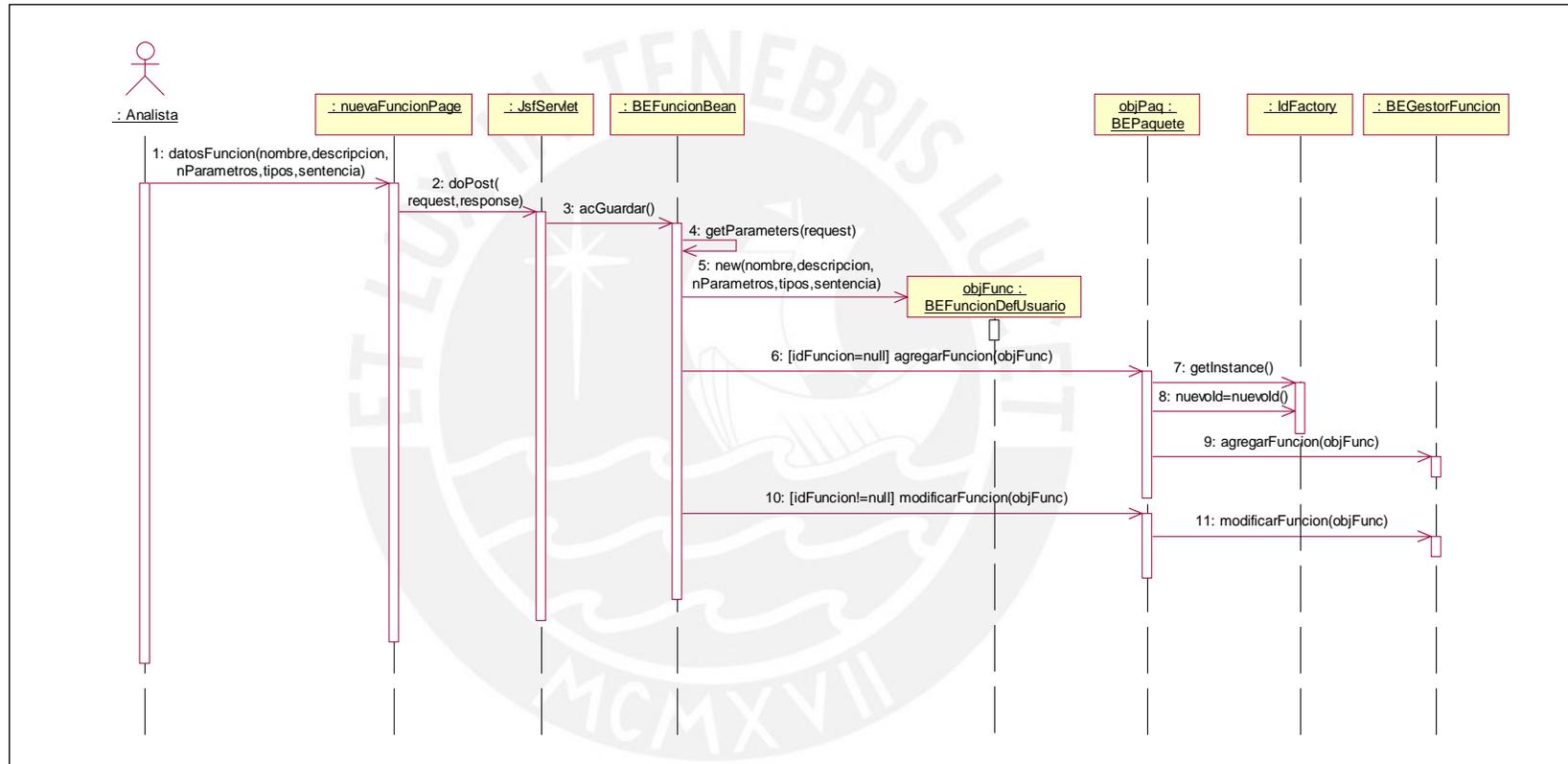


2.2.9. Personalizar Transformacion usando Objetos Activos – Script

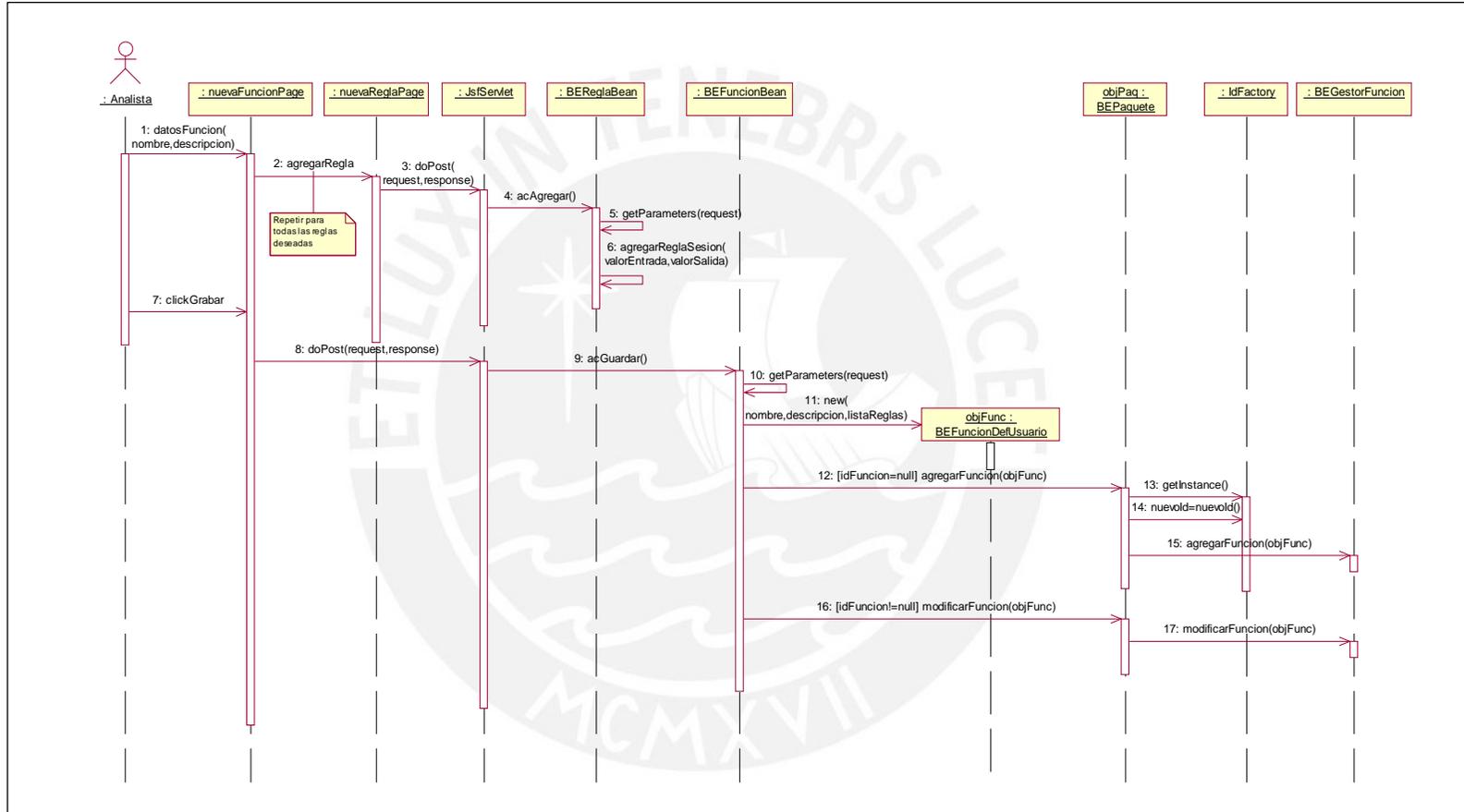


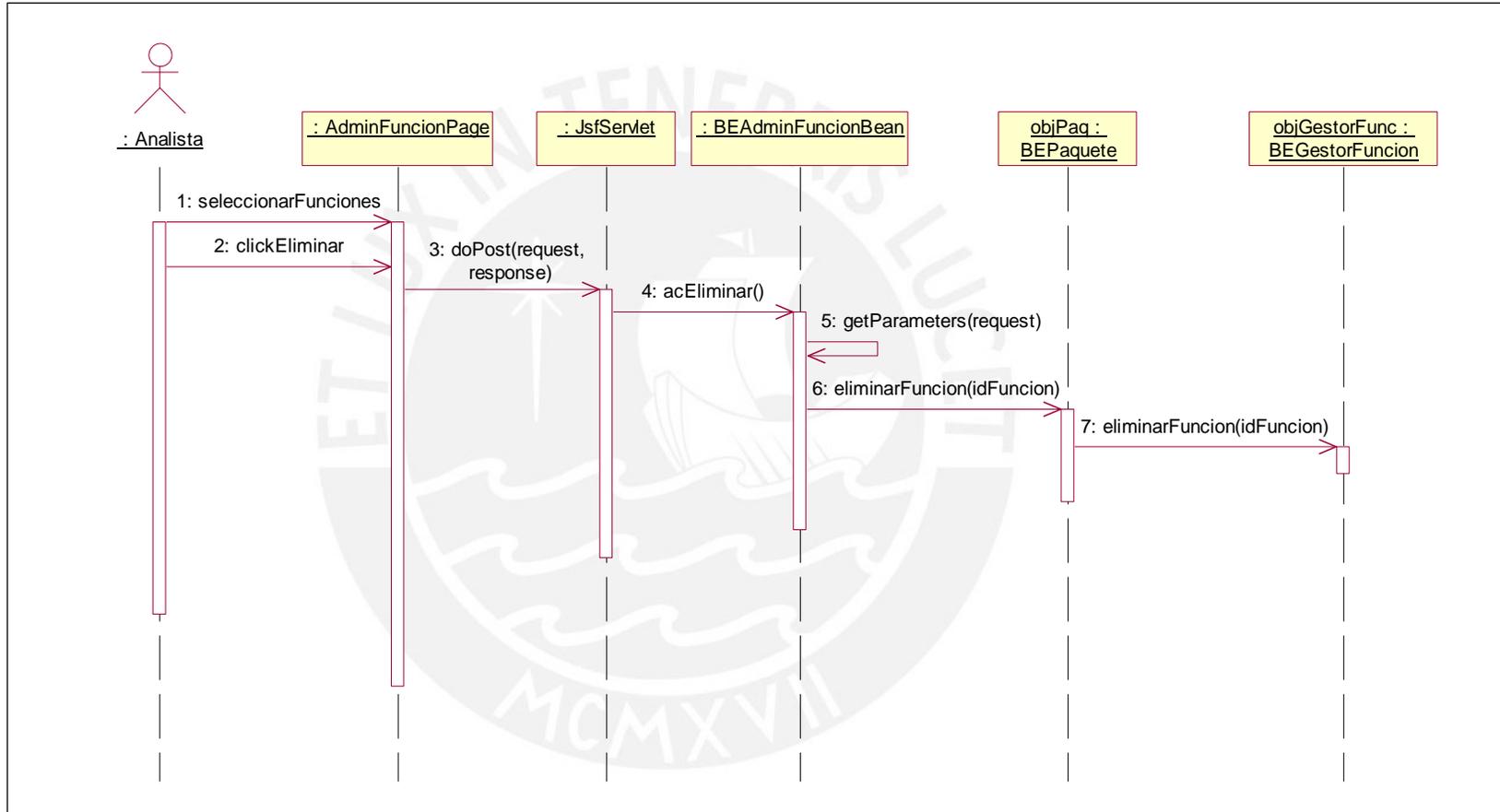
2.2.10. Validar Scripts mediante Parser



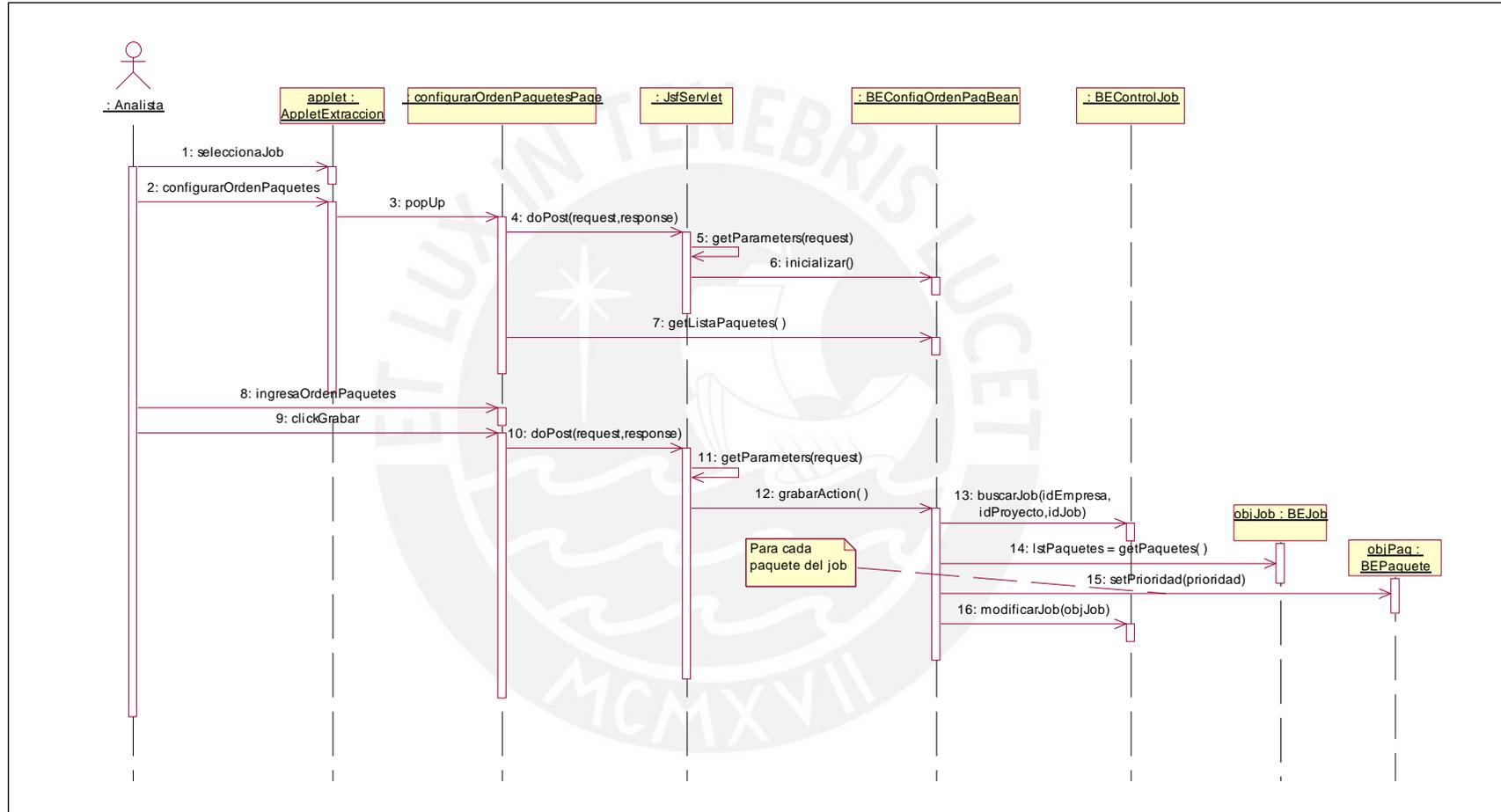


2.2.12. Administrar Funciones – Crear/Modificar Función Tipo Regla

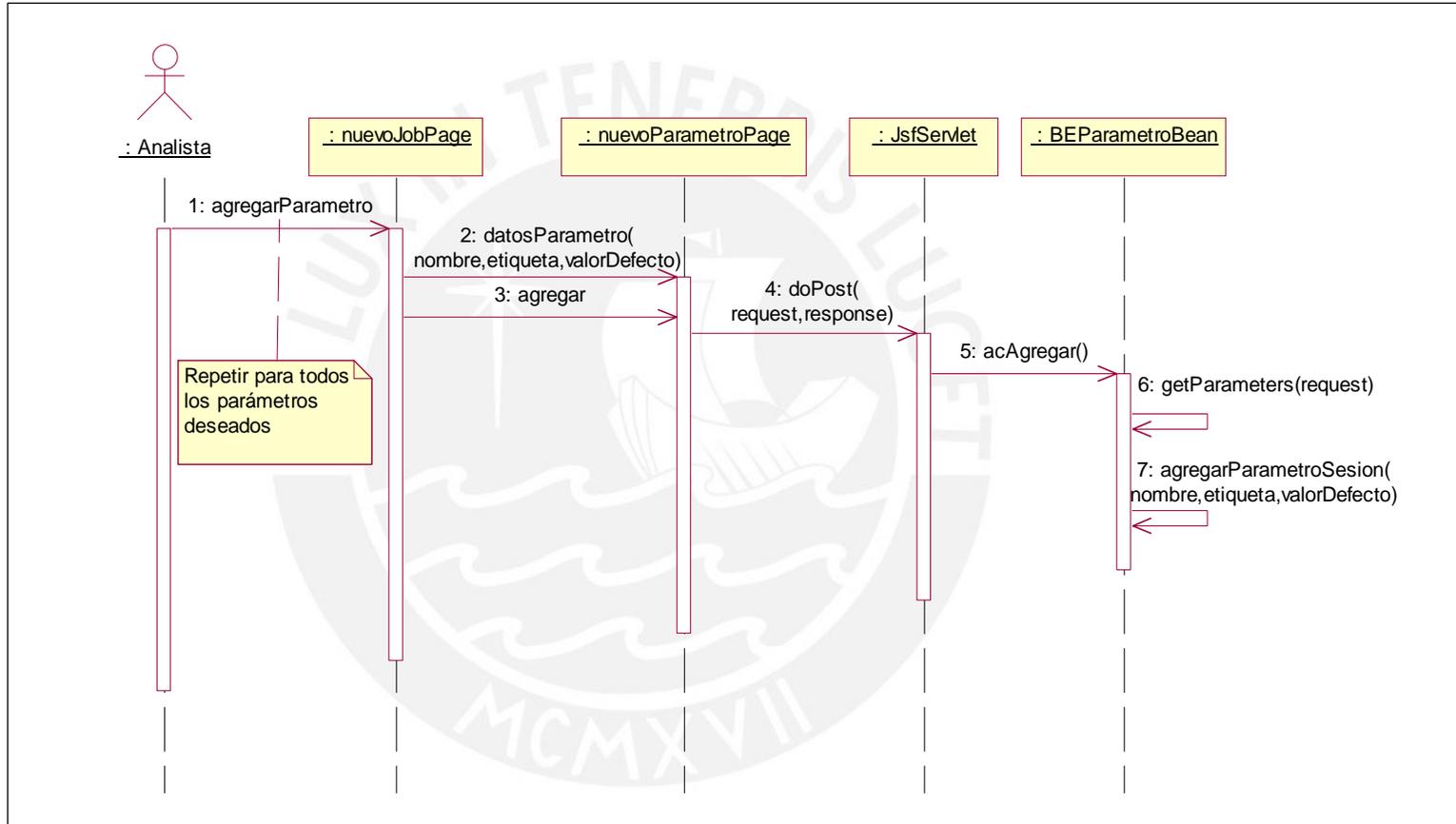




2.2.14. Configurar orden de ejecución de paquetes

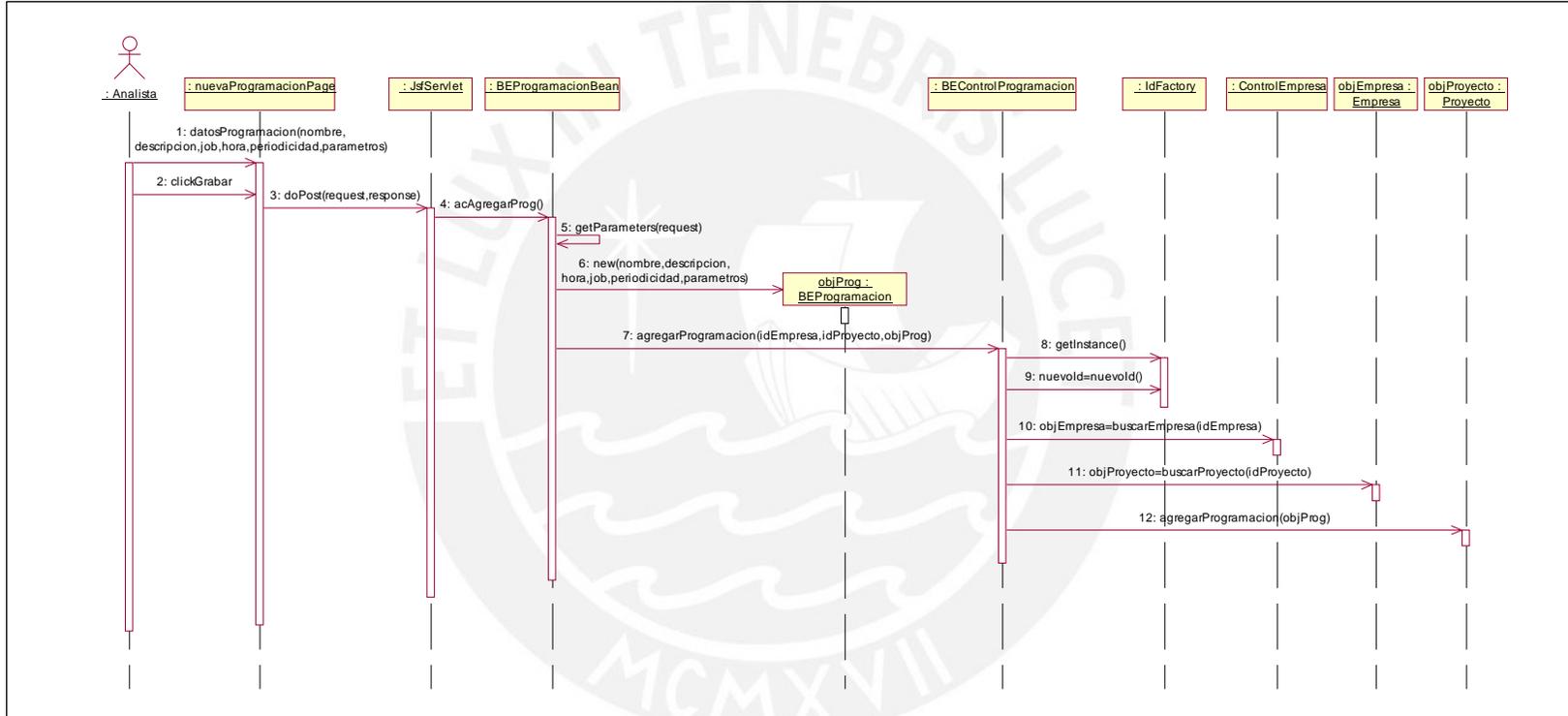


2.2.15. Configurar parámetros

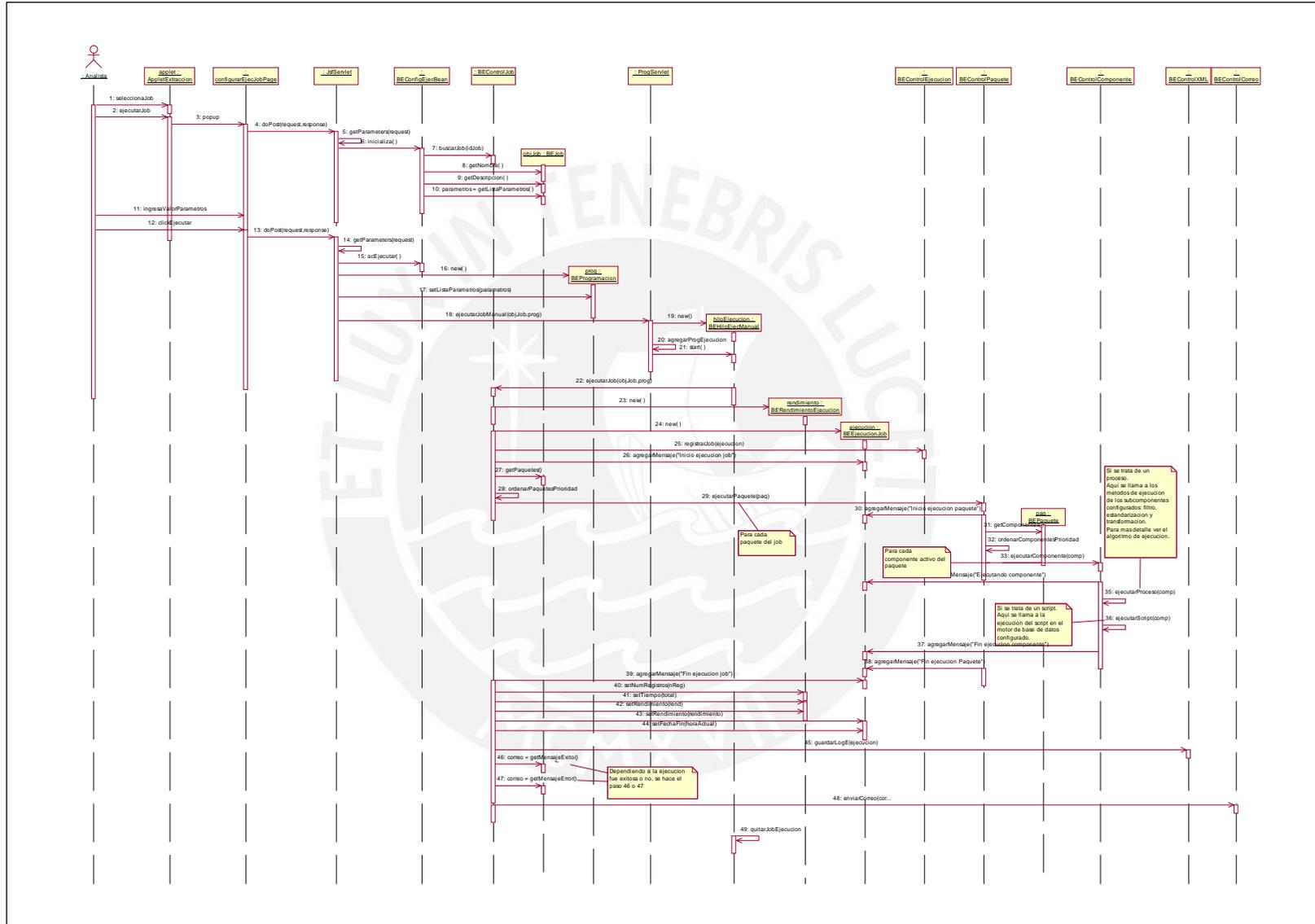


2.3. Paquete Ejecución

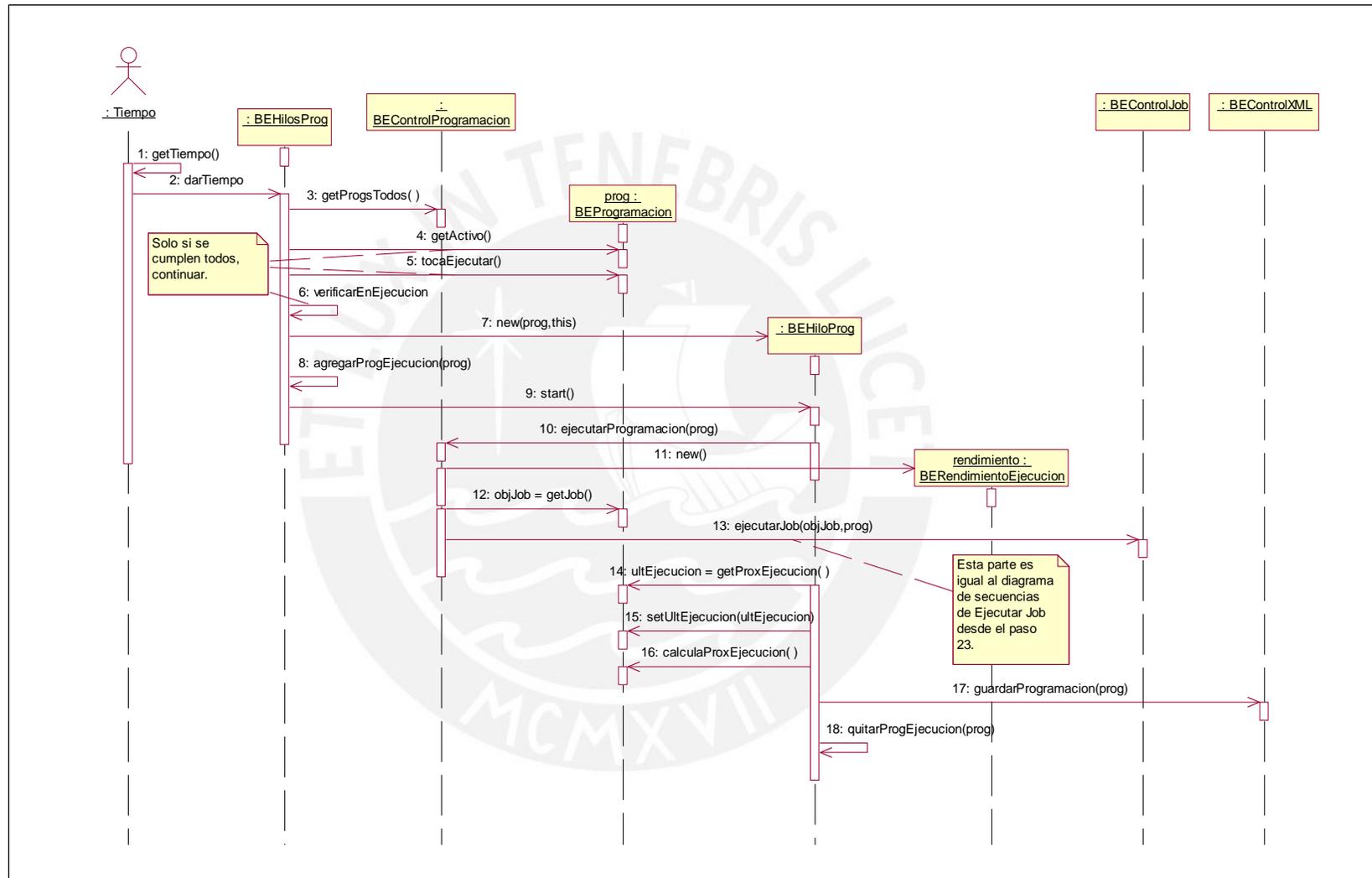
2.3.1. Programar job



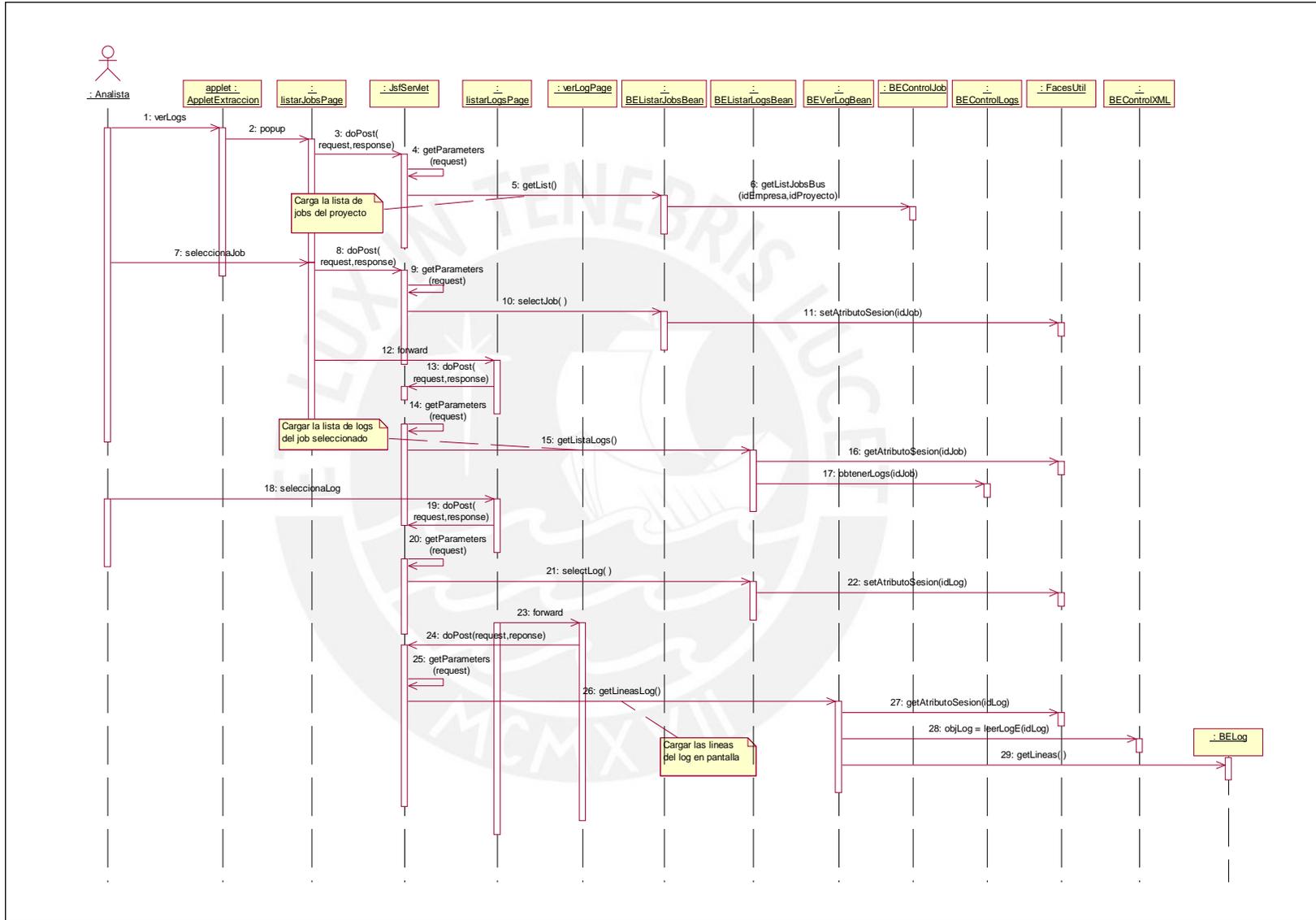
2.3.2. Ejecutar job



2.3.3. Ejecutar Job automático



2.3.4. Visualizar log de ejecución



ANEXO I. DISEÑO DE PANTALLAS

1. Introducción

1.1 Objetivos

El presente documento ha sido elaborado con la finalidad de presentar los diseños de pantalla para la herramienta AXEbit, módulo de extracción.

1.2 Alcance

El presente documento de diseño de pantallas es aplicable al módulo de extracción de la herramienta AXEbit y comprende tanto las pantallas del *applet* como las páginas JSF para la configuración de los diversos componentes de la herramienta.

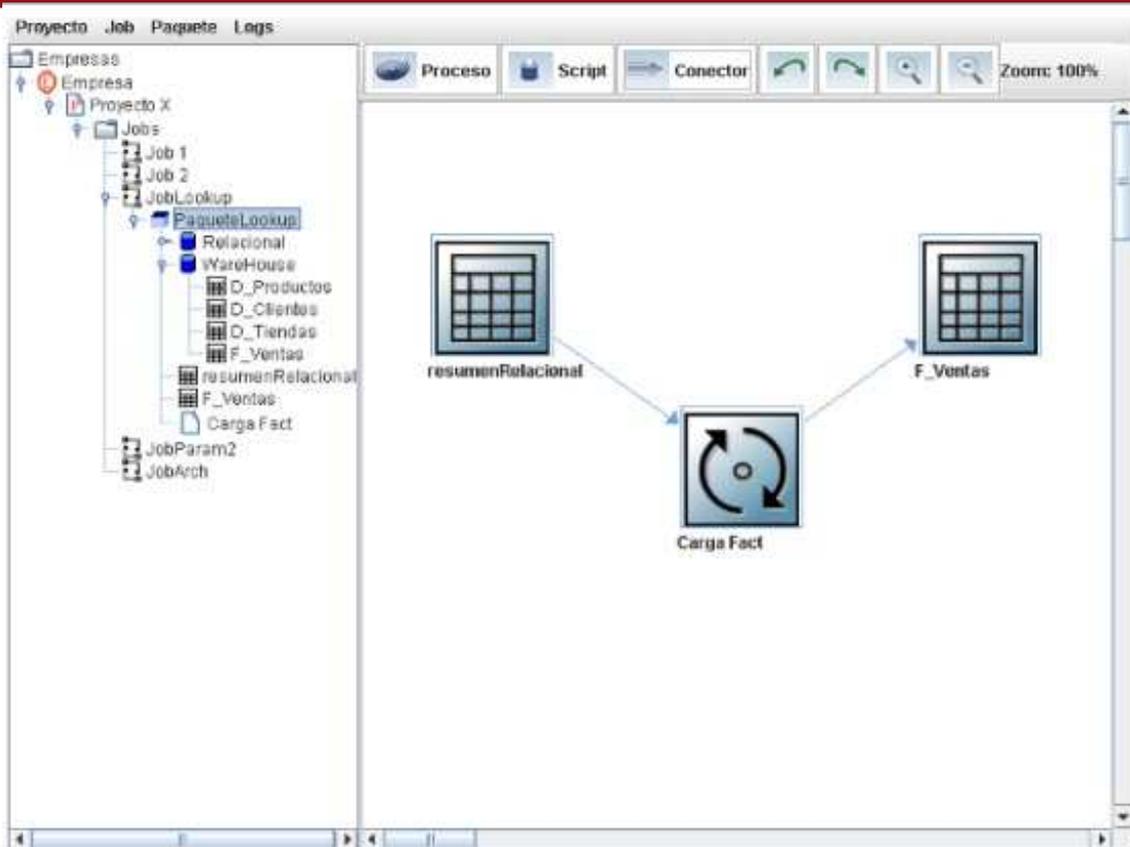
2. Estándares de programación

A continuación se muestran las pantallas de la herramienta:

2.1 Applet

El *applet* de la herramienta contiene la interfaz principal con la que interactúa el usuario para el módulo de extracción de la herramienta. Las partes que componen el *applet* son:

- a) **Árbol de proyecto:** Ubicado al lado izquierdo del *applet*, permite navegar entre los objetos del proyecto de forma jerárquica, primero la raíz donde se ubican las empresas con las que trabaja el usuario, luego los proyectos que pertenecen a la empresa, en el siguiente nivel los *jobs* y luego los paquetes. Cada elemento tiene un menú contextual que permite configurarlo.
- b) **Área de dibujo:** Corresponde al área más grande del *applet* y es donde el usuario puede formar el flujo de transformación para un paquete. Cuenta con una barra de herramientas en la cual se pueden seleccionar los componentes a agregar al flujo.
- c) **Menú principal:** Ubicado en la parte superior del *applet* permite acceder rápidamente a todas las funcionalidades de la herramienta que estén activas para el elemento actual.



Applet del módulo de extracción

2.2 Páginas JSF

Para configurar un componente del flujo de transformación se llama a una ventana popup con una página *JSF* apropiada. Del mismo modo se hace para cada funcionalidad del menú, administración de *jobs*, programaciones, muestra de *logs* de ejecución, etc.

Las características principales de las páginas *JSF* que se han implementado para la herramienta son las siguientes:

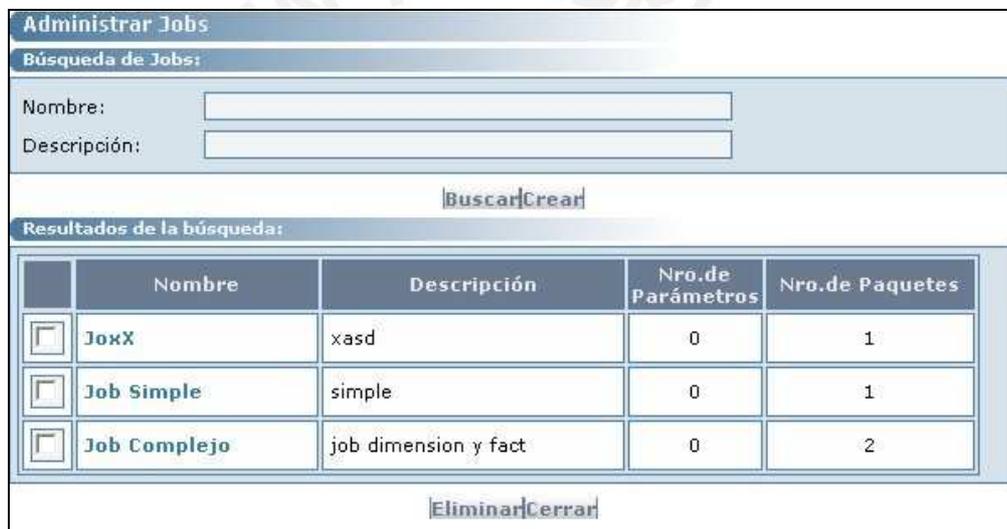
- a) Se utiliza las hojas de estilo o *cascading style sheets* (CSS) para unificar la definición de estilos de todas las páginas y facilitar de esta forma el mantenimiento de las páginas. Un cambio en la hoja de estilos se verá reflejado en todas las páginas inmediatamente.
- b) Se ha seleccionado un color tenue para evitar el agotamiento visual luego de usar la herramienta por tiempo prolongado.

Se hace uso de archivos properties o bundle para obtener el texto que se muestra en las páginas, de esta forma se puede establecer un archivo properties por idioma y permitir que se seleccione dinámicamente el archivo de idioma apropiado según la configuración del explorador web que accede a la herramienta. Esto permite que la herramienta soporte múltiples idiomas con un mínimo esfuerzo, pues no hay que modificar las páginas JSF, sino solamente definir nuevos archivos properties para los idiomas adicionales.

A continuación se muestra la lista de páginas que forman la aplicación organizadas por funcionalidad y acompañadas de una breve descripción.

2.2.1 Administrar jobs

Esta pantalla permite administrar los jobs del proyecto actual. Se pueden crear, modificar o eliminar los jobs.



Administrar Jobs				
Búsqueda de Jobs:				
Nombre:	<input type="text"/>			
Descripción:	<input type="text"/>			
Buscar Crear				
Resultados de la búsqueda:				
	Nombre	Descripción	Nro.de Parámetros	Nro.de Paquetes
<input type="checkbox"/>	JobX	xasd	0	1
<input type="checkbox"/>	Job Simple	simple	0	1
<input type="checkbox"/>	Job Complejo	job dimension y fact	0	2
Eliminar Cerrar				

Página de administración de jobs

2.2.2 Nuevo job

Esta pantalla permite crear y modificar los datos de un job.

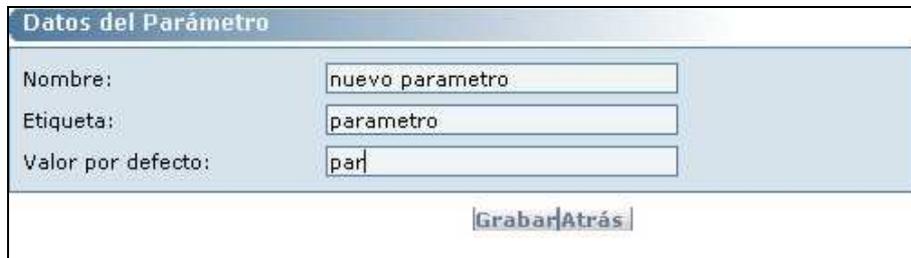


Nuevo Job			
Nombre:	<input type="text" value="JobNuevo"/>		
Descripción:	<input type="text" value="nuevo job"/>		
Relación de parámetros			
	Nombre	Etiqueta	Valor por Defecto
<input type="checkbox"/>	nuevo parametro	parametro	par
Agregar			
Eliminar			
Grabar Atrás			

Página de creación y modificación de jobs

2.2.3 Nuevo parámetro

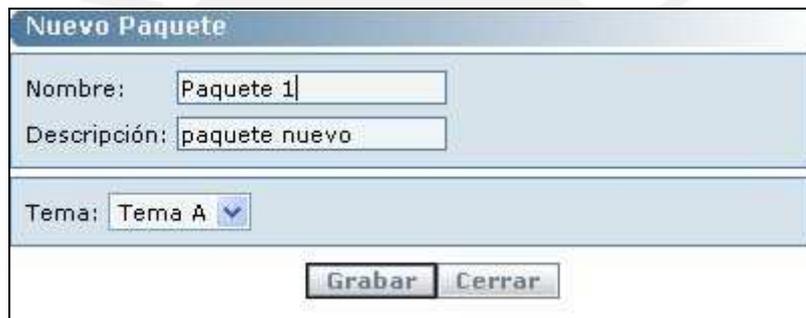
Esta pantalla permite crear o modificar los datos de un parámetro para un job.



Página de creación y modificación de parámetros

2.2.4 Nuevo paquete

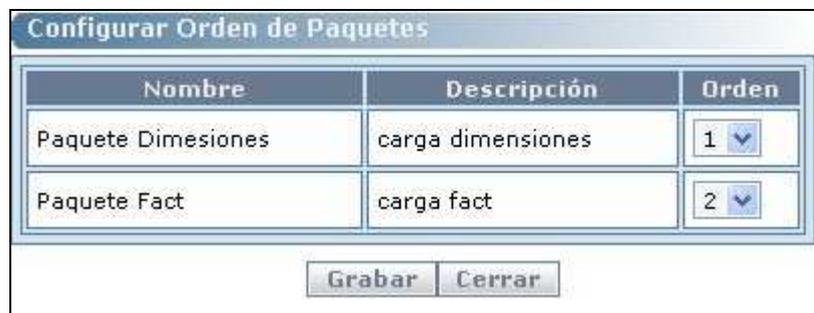
Esta pantalla permite crear un nuevo paquete para un job.



Página de creación de paquete

2.2.5 Configurar orden de paquetes

Esta pantalla permite configurar el orden en que se ejecutarán los paquetes que pertenecen a un job.



Nombre	Descripción	Orden
Paquete Dimensiones	carga dimensiones	1
Paquete Fact	carga fact	2

Página de configuración del orden de paquetes

2.2.6 Configurar fuente de datos base de datos

Esta pantalla permite configurar los datos de conexión a una base de datos para ser utilizada como fuente de datos del paquete.



Configurar Base de Datos

Nombre:

Descripción:

Tipo:

Conexión:

Base Datos:

Usuario:

Contraseña:

Página de configuración de base de datos

2.2.7 Seleccionar tablas de base de datos

Esta pantalla permite seleccionar las tablas de la base de datos que serán agregadas al paquete para poder formar parte del flujo de transformación.



Seleccionar Tablas

Tablas de la Base de Datos

	Nombre
<input type="checkbox"/>	Ventas
<input type="checkbox"/>	Clientes
<input type="checkbox"/>	ResumenVentas
<input type="checkbox"/>	resumenRelacional
<input type="checkbox"/>	TextoDestino
<input type="checkbox"/>	PersonasA
<input type="checkbox"/>	pruebaMalo
<input type="checkbox"/>	pruebaMaloDestino
<input type="checkbox"/>	Datos1
<input type="checkbox"/>	Datos2
<input type="checkbox"/>	DatosComb
<input type="checkbox"/>	persGen
<input type="checkbox"/>	persGenESTD

Página de selección de tablas de la base de datos

2.2.8 Configurar fuente de datos archivo

Esta pantalla permite seleccionar la carpeta desde la cual se leerán los archivos planos para ser agregados al paquete.

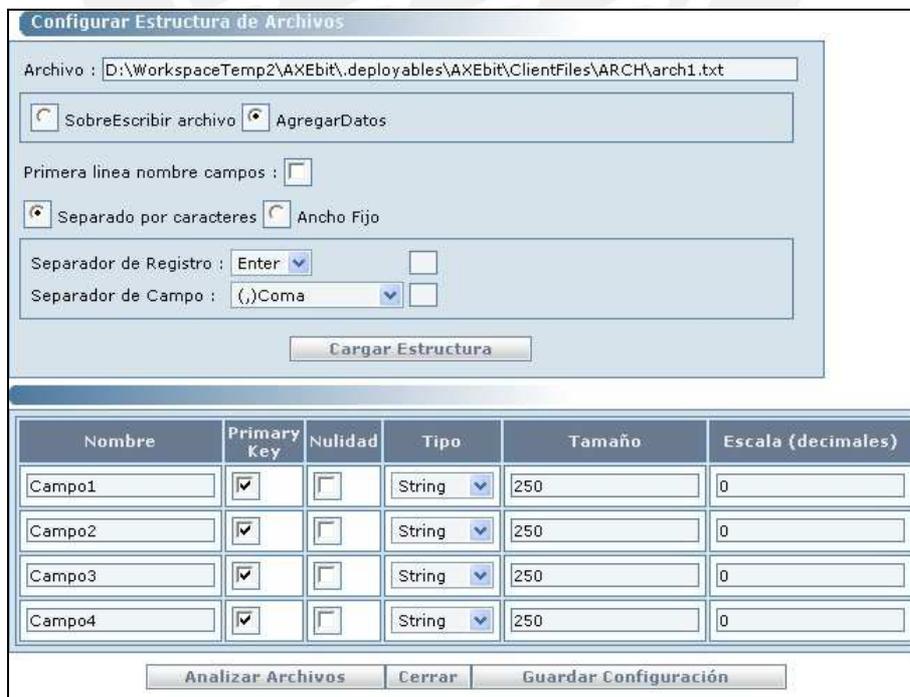


Página de configuración de ruta de archivos planos

2.2.9 Configurar estructura de archivo plano

Esta pantalla permite indicar la estructura de un archivo plano agregado al paquete. Se debe especificar la separación entre campos y registros del archivo, o si es que tiene ancho fijo para sus campos. También se debe indicar si al momento de escribir en el archivo se reemplazarán todos los datos o se agregará al final.

Finalmente, cuando se tienen los campos, se debe indicar el nombre y tipo de dato de cada uno.

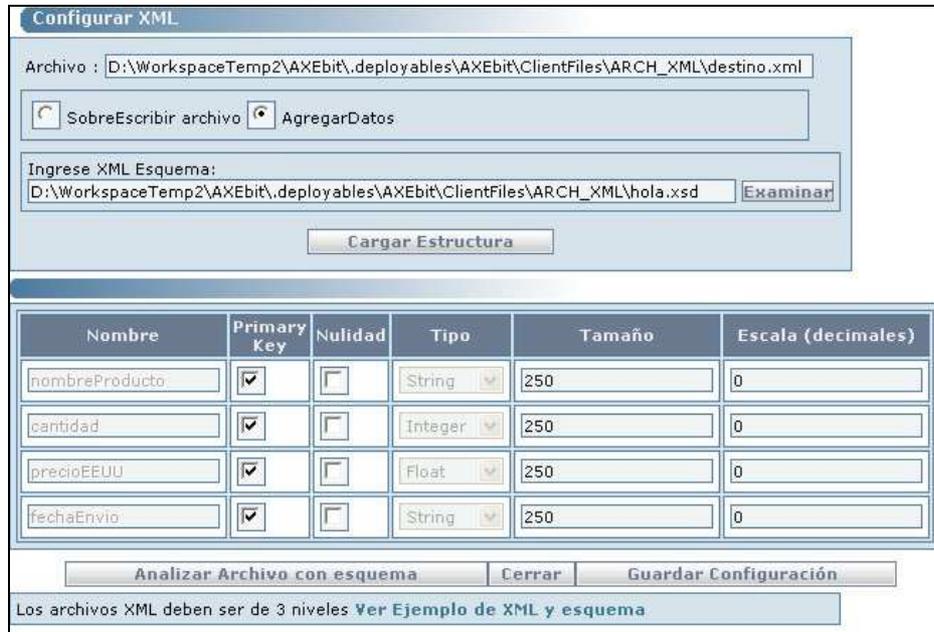


Nombre	Primary Key	Nulidad	Tipo	Tamaño	Escala (decimales)
Campo1	<input checked="" type="checkbox"/>	<input type="checkbox"/>	String	250	0
Campo2	<input checked="" type="checkbox"/>	<input type="checkbox"/>	String	250	0
Campo3	<input checked="" type="checkbox"/>	<input type="checkbox"/>	String	250	0
Campo4	<input checked="" type="checkbox"/>	<input type="checkbox"/>	String	250	0

Página de configuración de estructura de archivo plano

2.2.10 Configurar estructura de archivo xml

Esta pantalla permite especificar la ruta desde la cual se cargará el archivo XSD (esquema XML) para obtener la información de los campos que conforman un archivo XML agregado al paquete.



Configurar XML

Archivo : D:\WorkspaceTemp2\AXEbit\deployables\AXEbit\ClientFiles\ARCH_XML\destino.xml

SobreEscribir archivo AgregarDatos

Ingrese XML Esquema:
D:\WorkspaceTemp2\AXEbit\deployables\AXEbit\ClientFiles\ARCH_XML\hola.xsd

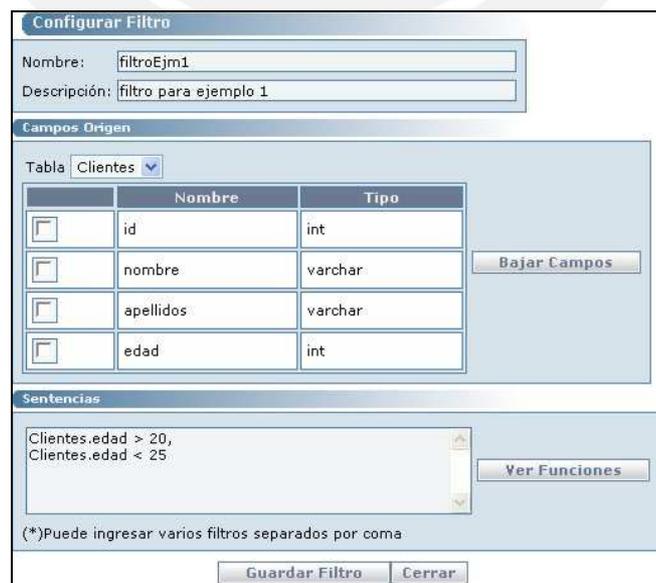
Nombre	Primary Key	Nulidad	Tipo	Tamaño	Escala (decimales)
nombreProducto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	String	250	0
cantidad	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Integer	250	0
precioEEUU	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Float	250	0
fechaEnvio	<input checked="" type="checkbox"/>	<input type="checkbox"/>	String	250	0

Los archivos XML deben ser de 3 niveles [Ver Ejemplo de XML y esquema](#)

Página de configuración de estructura de archivo XML

2.2.11 Configurar filtro

Esta pantalla permite configurar el subcomponente 'Filtro' del flujo de transformación.



Configurar Filtro

Nombre: filtroEjm1

Descripción: filtro para ejemplo 1

Campos Origen

Tabla: Clientes

	Nombre	Tipo
<input type="checkbox"/>	id	int
<input type="checkbox"/>	nombre	varchar
<input type="checkbox"/>	apellidos	varchar
<input type="checkbox"/>	edad	int

Sentencias

Clientes.edad > 20,
Clientes.edad < 25

(*)Puede ingresar varios filtros separados por coma

Página de configuración de filtro

2.2.12 Configurar transformación

Esta pantalla permite configurar el subcomponente 'Transformación' del flujo de transformación.



Nombre	Tipo	Campo Origen	Transformación
idCliente	int	---	ClienteMod.idCliente =Clientes.id
nombreCompleto	varchar	---	ClienteMod.nombreCompleto=Clientes.nombreCompleto

Página de configuración de transformación

2.2.13 Configurar estandarización

Esta pantalla permite configurar el subcomponente 'Estandarización' del flujo de transformación.



Nombre	Tipo	Estandarización
idCliente	int	ClienteMod.idCliente =nothing
nombreCompleto	varchar	ClienteMod.nombreCompleto=mayuscula

Página de creación y modificación de parámetros

2.2.14 Configurar lookup

Esta pantalla permite configurar el subcomponente 'Lookup' del flujo de transformación.

Configurar Lookup

Nombre:

Descripción:

Relaciones Lookup

Nombre	Campo Origen	Tabla Relacionada	Campo Relacionado	Campo Llave
idCliente	---	---		
idProducto	---	---		
idTienda	---	---		
valorVenta	---	---		

Página de configuración de lookup

2.2.15 Ver funciones

Esta pantalla permite visualizar las funciones que aplican al subcomponente que se está configurando.

Posibles Funciones a aplicar:		
Funciones predefinidas:		
Nombre	Descripción	Aplicable a
+	Operador de Concatenación de Cadenas	Tipos Cadena
+	Operador de suma de valores	Tipos Numéricos
-	Operador de resta de valores	Tipos Numéricos
*	Operador de multiplicación de valores	Tipos Numéricos
/	Operador división de valores	Tipos Numéricos
^	Operador de potencia	Tipos Numéricos
sum(x)	Función de suma acumulativa de un campo	Tipos Numéricos
prom(x)	Función de promedio de valores de un campo	Tipos Numéricos
directo(x)	Función de paso directo de datos. Bloquea la estandarización.	Cualquier tipo
substring(x,y,z)	Función de recorte de cadenas	Tipos Cadena

Página de listado de funciones

2.2.16 Administrar funciones definidas por el usuario

Esta pantalla permite administrar las funciones definidas por el usuario. Permite crear, modificar y eliminar funciones.



Funciones definidas por el Usuario

Búsqueda de funciones:

Nombre:

Descripción:

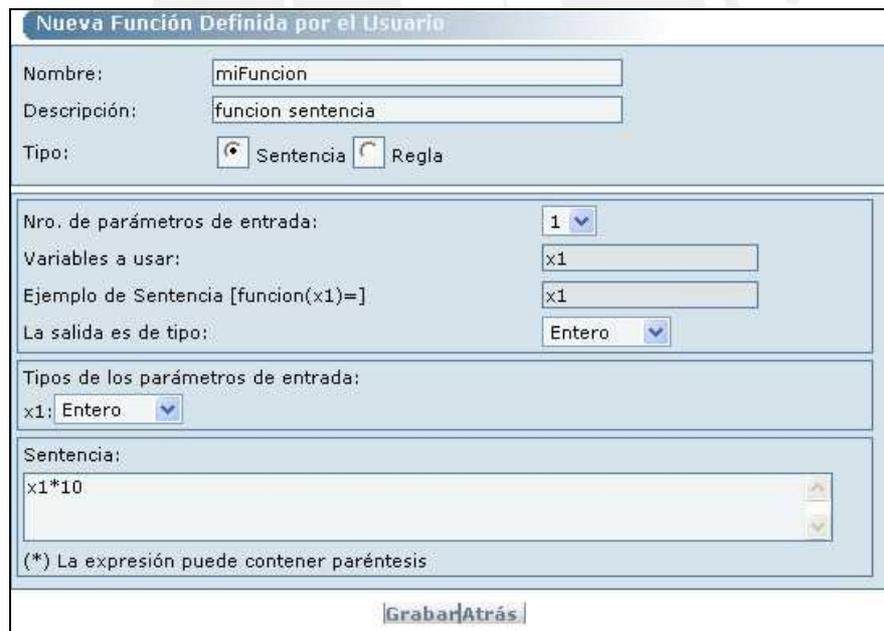
Resultados de la búsqueda:

	Nombre	Descripción	Usuario
<input type="checkbox"/>	funcion	nueva	10

Página de administración funciones definidas por el usuario

2.2.17 Nueva función definida por el usuario tipo sentencia

Esta pantalla permite crear nuevas funciones definidas por el usuario de tipo sentencia.



Nueva Función Definida por el Usuario

Nombre:

Descripción:

Tipo: Sentencia Regla

Nro. de parámetros de entrada:

Variables a usar:

Ejemplo de Sentencia [funcion(x1)=]:

La salida es de tipo:

Tipos de los parámetros de entrada:

x1:

Sentencia:

(*) La expresión puede contener paréntesis

Página de creación de función definida por el usuario tipo sentencia

2.2.18 Nueva función definida por el usuario tipo reglas

Esta pantalla permite crear nuevas funciones definidas por el usuario de tipo reglas.



Nueva Función Definida por el Usuario

Nombre:

Descripción:

Tipo: Sentencia Regla

Reglas de la Función:

	Valor de Entrada	Valor de Salida
<input type="checkbox"/>	x	1

Página de creación de función definida por el usuario tipo reglas

2.2.19 Formar sentencia where

Esta pantalla permite formar fácilmente la sentencia where para una transformación que involucre más de una tabla origen.



Configurar Constraints

Constraints

Nombre		Campo Relacionado
LineasFactura.id	=	---
LineasFactura.idFactura	=	Facturas.id Facturas.idCliente Facturas.idTienda Facturas.fecha Facturas.total
LineasFactura.idProducto	=	---
LineasFactura.cantidad	=	---
LineasFactura.monto	=	---

Página de formación de sentencia where

2.2.20 Administrar programaciones

Esta pantalla permite administrar las programaciones del proyecto actual. Permite agregar, modificar y eliminar programaciones.



Programaciones					
Búsqueda de programaciones:					
Nombre:		<input type="text"/>			
Descripción:		<input type="text"/>			
<input type="button" value="Buscar"/> <input type="button" value="Crear"/>					
Resultados de la búsqueda:					
	Nombre	Descripción	Activo	Ultima ejecución	Periodicidad
<input type="checkbox"/>	miProg	nueva prog	Si	No ejecutado	Semanal
<input type="button" value="Eliminar"/> <input type="button" value="Cerrar"/>					

Página de administración de programaciones

2.2.21 Programar job

Esta pantalla permite configurar una nueva programación para ejecutar automáticamente un job.



Nuevo Job			
Nombre:		<input type="text" value="JobNuevo"/>	
Descripción:		<input type="text" value="nuevo job"/>	
Relación de parámetros			
	Nombre	Etiqueta	Valor por Defecto
<input type="checkbox"/>	nuevo parametro	parametro	par
<input type="button" value="Agregar"/> <input type="button" value="Eliminar"/>			
<input type="button" value="Grabar"/> <input type="button" value="Atrás"/>			

Página de programación de job

2.2.22 Ejecutar job

Esta pantalla permite ejecutar manualmente un job.



Ejecución Manual del Job		
Nombre:		<input type="text" value="Job Simple"/>
Descripción:		<input type="text" value="simple"/>
Ingreso de valores para los parámetros del job		
Etiqueta	Valor por defecto	Valor para la ejecución
<input type="button" value="Ejecutar"/> <input type="button" value="Cerrar"/>		

Página de ejecución manual de job

2.2.23 Ver ejecución de job

Esta pantalla permite visualizar la ejecución del job en tiempo real.

Ejecución del Job

Log de ejecución:

```
Sat Jan 13 22:13:58 COT 2007 Inicio de ejecución del
job: Job Simple
Sat Jan 13 22:13:58 COT 2007 Inicio de ejecución del
paquete: Ejemplo
Sat Jan 13 22:13:58 COT 2007 Inicio de ejecución del
componente: nuevo proceso 0
Sat Jan 13 22:13:58 COT 2007 Ejecutando filtro
filtroEjm1
Sat Jan 13 22:13:59 COT 2007 Ejecutando
transformacion transformacionEjm
```

Fecha y hora de inicio: Sat Jan 13 22:13:58 COT 2007
 Fecha y hora de fin: Sat Jan 13 22:13:59 COT 2007
 Tiempo en ejecución (segundos): 0,52

Registros procesados: 4
 Tasa de procesamiento (reg/seg): 7,75

[Cerrar]

Página de vista de ejecución de job

2.2.24 Listar jobs

Esta pantalla muestra la lista de jobs del proyecto, desde aquí se selecciona el job para el cual se desea ver los logs de ejecución.

Jobs del Proyecto

Nombre	Descripción
JobX	xasd
Job Simple	simple
Job Complejo	job dimension y fact

[Cerrar]

Página de listado de jobs para un proyecto

2.2.25 Listar logs por job

Esta pantalla muestra la lista de logs de ejecución para un job específico.

Logs de Ejecución

Fecha Ejecución
31/12/06 18:10:32
13/01/07 22:13:58

[Cerrar]

Página de listado de logs para un job

2.2.26 Ver log

Esta pantalla muestra el contenido de un log de ejecución.

Ver Log de Ejecución	
Hora	Mensaje
13/01/07 22:13:58	Inicio de ejecución del job: Job Simple
13/01/07 22:13:58	Inicio de ejecución del paquete: Ejemplo
13/01/07 22:13:58	Inicio de ejecución del componente: nuevo proceso 0
13/01/07 22:13:58	Ejecutando filtro filtroEjm1
13/01/07 22:13:59	Ejecutando transformacion transformacionEjm
13/01/07 22:13:59	Ejecutando estandarizacion estandarizacionEjm
13/01/07 22:13:59	Ejecutando carga de datos.
13/01/07 22:13:59	Fin de ejecución del componente: nuevo proceso 0
13/01/07 22:13:59	Fin de ejecución del paquete: Ejemplo
13/01/07 22:13:59	Fin de ejecución del job: Job Simple

Tiempo en ejecución (segundos): 0,52
Registros procesados: 4

Página de vista de log de ejecución



ANEXO J. DOCUMENTO DE ESTIMACIÓN

1. Introducción

1.1. Objetivos

El presente documento tiene como objetivo mostrar el esfuerzo previsto para la realización del proyecto en su segunda etapa. Teniendo como experiencia previa la primera etapa en la cual la herramienta tenía una arquitectura cliente-servidor.

1.2. Alcance

El presente documento de estimación se aplicará al módulo de Extracción de AxeBit tomando en consideración la segunda etapa de la realización de la herramienta, la misma que es descrita a lo largo de todo el documento de tesis.

Las técnicas empleadas para la estimación son las siguientes: Puntos de función y Cocomo II.

1.3. Definiciones y acrónimos

Acrónimo	Significado
Puntos de función	
EI	Entrada externa, datos o información de control que procede de fuera de los límites de la aplicación.
EO	Salida externa, datos o información de control que sale fuera de los límites de la aplicación mediante un proceso lógico.
EQ	Consulta externa, procesos elementales que obtienen una combinación de entrada/salida como resultado de una recuperación de datos, pero no mantienen ningún ILF.
ILF	Fichero lógico interno, son grupos de datos lógicamente relacionados identificables por el usuario, o información de control contenida dentro de los límites de la aplicación.
EIF	Fichero de interfaz externo, información de control utilizada en la aplicación pero mantenida por medio de otra.
DET	Elemento de tipo dato, campo no recursivo y reconocible por el usuario.
RET	Elemento de tipo registro, subgrupo de DETs reconocibles por el usuario.
FTR	Tipo de fichero referenciado, fichero al que una transacción hace referencia.
PFSA	Puntos de función sin ajustar.
Cocomo II	
Factor Producto	

RELY	Fiabilidad del producto requerida.
DATA	Tamaño de la base de datos.
CLPX	Complejidad del producto.
RUSE	Reutilización requerida.
DOCU	Adecuación de la documentación a las necesidades del ciclo de vida.
Factor Plataforma	
TIME	Limitaciones en el tiempo de ejecución.
STOR	Limitaciones en el almacenamiento principal.
PVOL	Volatilidad de la plataforma.
Factor Personal	
ACAP	Capacidad de los analistas.
AEXP	Capacidad del programador.
PCAP	Continuidad del personal.
PEXP	Experiencia en aplicaciones.
LTEX	Experiencia en la plataforma
PCON	Experiencia con el lenguaje y las herramientas.
Factor Proyecto	
TOOL	Uso de herramientas software.
SITE	Desarrollo en varios sitios.
SCED	Calendario de desarrollo requerido.

2. Determinación de puntos de función

Con el fin de obtener el cálculo de puntos de función se determinarán los siguientes puntos: archivos internos lógicos, entradas externas, consultas externas y salidas externas.

2.1. Archivos Internos Lógicos

Se determinan los DETs y RETs de cada uno de los IFL. El resultado de este cálculo se muestra a continuación:

Archivos Internos Lógicos	
ILF	DET
Archivo	ruta extension tipo nombres ancho lonAncho caracterCampo caracterRegistro append valido nCampo nRegistro nombreArchivo
Filtro	SentenciasFiltro tipo baseDatos cadenaConexion usuario contra ruta idTablas esquema idBD listaCampos
FuenteDatos	tipo baseDatos cadenaConexion usuario contra ruta idTablas
Job	prioridad List paquetes listaParametros correoOK correoERR
Log	idJob fechaIni fechaFin lineas numReg tiempoTotal rendimiento
Paquete	prioridad tema componentes fuenteDatos funciones diagrama guardaLog
Script	script fuenteDatos prioridad tipoScript
Tabla	esquema idBD listaCampos

Transformación	SentenciasTransformacion tipo baseDatos cadenaConexion usuario contra ruta idTablas esquema idBD listaCampos
Parámetro	nombre etiqueta valor

En el sistema planteado no encontramos la presencia de EIF, ya que no se contará con aplicaciones externas que interactúan en la aplicación.

Usando el resultado de la tabla anterior y la tabla de cálculo de complejidad para ILF (ver anexo 4.1) se obtendrá a complejidad de cada uno de ellos.

Calculo de Complejidad			
ILF	#DET	#RET	Complejidad
Archivos	13	1	BAJA
Filtros	11	3	BAJA
Fuentes de datos	5	2	BAJA
Jobs	7	3	BAJA
Logs	7	3	BAJA
Paquetes	4	3	BAJA
Scripts	3	2	BAJA
Tablas	1	2	BAJA
Transformaciones	11	3	BAJA
Parámetro	3	1	BAJA

A continuación se realizará el cálculo de los puntos de función para los ILF (ver anexo 4.5).

Puntos de Función			
Tipo de Función	Complejidad Funcional	Totales de Complejidad	Totales de Tipo de Función
ILF	10 BAJA x 7	70	70
	0 MEDIA x 10	0	
	0 ALTA x 15	0	

2.2. Entradas Externas

Se determinarán las entradas externas para los casos de uso del sistema.

Entradas Externas	
Casos de Uso	Entrada Externa
Configurar Fuente de Datos	Crear Fuente de Datos Modificar Fuente de Datos Eliminar Fuente de Datos
Conectar a fuente de datos	Conectar Fuente de Datos
Obtener estructuras y objetos de fuente de datos	Obtener estructuras Obtener objetos de fuente de datos
Configurar estructura de archivo plano	Configurar archivo Configurar tipo de datos de registros
Crear Paquete	Crear Paquete Configurar Flujo Eliminar Paquete
Crear Job	Crear Job Modificar Job Eliminar Job
Configurar Parámetros	Configurar Parámetros
Crear flujo de transformación	Crear flujo de transformación
Personalizar transformación usando objetos activos	Configurar flujo de transformación con objetos activos
Validar Scripts mediante Parser	Validar Scripts mediante Parser
Configurar Orden de Ejecución de Paquetes	Configurar Orden de Ejecución de Paquetes
Crear funciones definidas por el usuario	Crear funciones definidas por el usuario
Programar Job	Programar Job

Usando la tabla anterior y la tabla de cálculo de complejidad para EI (ver anexo 4.2) se obtendrá las DETs y FTRs como la complejidad de cada uno de ellos.

Calculo de Complejidad					
Entrada Externa	DET	FTR	#DET	#FTR	Comp.
Crear Fuente de Datos	tipo baseDatos cadenaConexion usuario contra ruta idTablas	FuenteDatos	7	1	BAJA
Modificar Fuente de Datos	tipo baseDatos cadenaConexion usuario contra ruta idTablas	FuenteDatos	7	1	BAJA
Eliminar Fuente de Datos	tipo baseDatos cadenaConexion usuario contra ruta idTablas	FuenteDatos	7	1	BAJA

Conectar Fuente de Datos	tipo baseDatos cadenaConexion usuario contra ruta idTablas	FuenteDatos	7	1	BAJA
Obtener estructuras	tipo baseDatos cadenaConexion usuario contra ruta idTablas esquema idBD listaCampos	Tabla FuenteDatos	10	2	MEDIA
Obtener objetos de fuente de datos	tipo baseDatos cadenaConexion usuario contra ruta idTablas esquema idBD listaCampos	Tabla FuenteDatos	10	2	MEDIA
Configurar archivo	ruta extension tipo nombres ancho lonAncho caracterCampo caracterRegistro append valido nCampo nRegistro nombreArchivo	Archivo	13	1	BAJA
Configurar tipo de datos de registros	ruta extension tipo nombres ancho lonAncho caracterCampo caracterRegistro append valido nCampo nRegistro nombreArchivo	Archivo	13	1	BAJA
Crear Paquete	prioridad tema componentes fuenteDatos funciones diagrama guardaLog	Paquete	7	1	BAJA

Configurar Flujo	prioridad tema componentes fuenteDatos funciones diagrama guardaLog SentenciasFiltro SentenciasTransf ormación prioridad List paquetes listaParametros correoOK correoERR	Paquete Filtro Transformación Job	14	4	ALTA
Eliminar Paquete	prioridad tema componentes fuenteDatos funciones diagrama guardaLog	Paquete	7	1	BAJA
Crear Job	prioridad List paquetes listaParametros correoOK correoERR	Job	5	1	BAJA
Modificar Job	prioridad List paquetes listaParametros correoOK correoERR	Job	5	1	BAJA
Eliminar Job	prioridad List paquetes listaParametros correoOK correoERR	Job	5	1	BAJA
Configurar Parámetros	nombre etiqueta valor	Parámetro	3	1	BAJA
Crear flujo de transformación	Sentencias Transformación tipo baseDatos cadenaConexion usuario contra ruta idTablas esquema idBD listaCampos	Transformación FuenteDatos Tabla	11	3	ALTA

Configurar flujo de transformación con objetos activos	Sentencias Transformación tipo baseDatos cadenaConexion usuario contra ruta idTablas esquema idBD listaCampos	Transformación FuenteDatos Tabla	11	3	ALTA
Validar Scripts mediante Parser	script fuenteDatos prioridad tipoScript	Script FuenteDatos	4	2	BAJA
Configurar Orden de Ejecución de Paquetes	prioridad tema componentes fuenteDatos funciones diagrama guardaLog	Paquete	7	1	BAJA
Crear funciones definidas por el usuario	nombre descripcion aplicableA	FuncionPred	3	1	BAJA
Programar Job	prioridad List paquetes listaParametros correoOK correoERR	Job	5	1	BAJA

A continuación se realizará el cálculo de los puntos de función para las EI (ver anexo 4.5).

Puntos de Función			
Tipo de Función	Complejidad Funcional	Totales de Complejidad	Totales de Tipo de Función
EI	16 BAJA x 3	45	74
	2 MEDIA x 4	8	
	3 ALTA x 6	18	

2.3. Consultas Externas

Se determinarán las consultas externas para cada uno de los casos de uso del sistema.

Salidas Externas	
Casos de Uso	Salida Externa
Visualizar Log de Ejecución	Log de Ejecución

Usando la tabla anterior y la tabla de cálculo de complejidad para EQ (ver anexo 4.4) se obtendrá las DETs y FTRs como la complejidad de cada uno de ellos.

Calculo de Complejidad					
Consulta Externa	DET	FTR	#DET	#FTR	Comp.
Visualizar Log de Ejecución	idJob fechaIni fechaFin lineas numReg tiempoTotal rendimiento	Log	7	1	BAJA

A continuación se realizará el cálculo de los puntos de función para las EQ (ver anexo 4.5).

Puntos de Función			
Tipo de Función	Complejidad Funcional	Totales de Complejidad	Totales de Tipo de Función
EQ	1 BAJA x 4	4	4
	0 MEDIA x 5	0	
	0 ALTA x 7	0	

2.1. Salidas Externas

Se determinarán las salidas externas para cada uno de los casos de uso del sistema.

Salidas Externas	
Casos de Uso	Salida Externa
Ejecutar Job	Ejecutar Job

Usando la tabla anterior y la tabla de cálculo de complejidad para EO (ver anexo 4.4) se obtendrá las DETs y FTRs como la complejidad de cada uno de ellos.

Calculo de Complejidad					
Salida Externa	DET	FTR	#DET	#FTR	Comp.
Ejecutar Job	prioridad correoOK correoERR prioridad tema componentes fuenteDatos funciones diagrama guardaLog tipo baseDatos cadenaConexion usuario contra ruta idTablas nombre etiqueta valor	Job Paquete Parámetro FuenteDatos	20	4	ALTA

A continuación se realizará el cálculo de los puntos de función para las EO (ver *anexo 4.5*).

Puntos de Función			
Tipo de Función	Complejidad Funcional	Totales de Complejidad	Totales de Tipo de Función
EO	0 BAJA x 4	4	7
	0 MEDIA x 5	0	
	1 ALTA x 7	0	

Realizando la sumatoria de todos los tipos de puntos de función se tiene:

Tipo Punto de Función	Cantidad PF
ILF	74
EI	68
EQ	4
EO	7
Total (sumatoria)	153

Según la tabla de transformación (*ver anexo 4.6*) para el lenguaje Java que será utilizado en la realización de este proyecto, se tienen 53 líneas de código por punto de función. Haciendo los cálculos respectivos tenemos:

$$53 \text{ LCF/PF} * 153 \text{ PF} = 8109 \text{ LCF} = 8.109 \text{ KLCF}$$

3. Modelo de estimación Cocomo II

3.1. Factores del proyecto

Criterios de aplicación	
Factor escala	Valor
Precedencias	2.48
Flexibilidad de desarrollo	4.05
Arquitectura / Solución de riesgo	2.83
Cohesión del equipo / Interacción	3.29
Madurez del proceso	4.68
Total (sumatoria)	17.33

3.2. Multiplicadores de esfuerzo

Multiplicadores de esfuerzo	
Factor criterio de costo PRODUCTO	Valor
RELY	0.82
DATA	1.00
CLPX	1.17
RUSE	1.00
DOCU	1.00
Total (productoria)	0.9594

Multiplicadores de esfuerzo	
Factor criterio de costo PERSONAL	Valor
ACAP	0.71
AEXP	1.00
PCAP	0.88
PEXP	1.09
LTEX	1.09
PCON	0.81
Total (productoria)	0.6013

Multiplicadores de esfuerzo	
Factor criterio de costo PLATAFORMA	Valor
TIME	1.00
STOR	1.00
PVOL	1.00
Total (productoria)	1.00

Multiplicadores de esfuerzo	
Factor criterio de costo PROYECTO	Valor
TOOL	0.78
SITE	1.00
SCED	1.00
Total (productoria)	0.78

3.3. Determinación del esfuerzo requerido

Con los datos obtenidos se puede finalmente encontrar el Esfuerzo, a continuación se muestran los cálculos finales:

$$\text{Exponente} = 0.91 + 0.01 * (\sum \text{Factores}) = 0.91 + 0.01 * 17.33 = 1.0833$$

$$\begin{aligned} \text{Esfuerzo} &= 2.94 * (\text{KLOC})^{\text{EXPONENTE}} * (\prod \text{Multiplicadores}) \\ &= 2.94 * (8.109)^{1.0833} * (0.9594 * 0.6013 * 1.00 * 0.78) \\ &= 2.94 * 9.6535 * 0.44997203 \\ &= 2.94 * (8.109)^{1.0833} * 0.44997203 \\ &= 2.94 * (8.109)^{1.0833} * 0.44997203 \\ &= 12.77 \text{ meses hombre} \end{aligned}$$

Debido a que el equipo de trabajo esta constituido por 5 personas, el esfuerzo por persona sería el siguiente:

ESFUERZO = 2.554 meses por integrante

Lo cual constituye tres meses de desarrollo por integrante.

4. Tablas referenciales

4.1. Complejidad de un ILF o EIF

Complejidad de un IFL o EIF			
	1 a 19 DET	20 a 50 DET	51 a más DET
1 RET	BAJA	BAJA	MEDIA
2 a 5 RET	BAJA	MEDIA	ALTA
6 o más RET	MEDIA	ALTA	ALTA

4.2. Complejidad de un EI

Complejidad de un EI			
	1 a 4 DET	5 a 15 DET	16 o más DET
0 a 1 FTR	BAJA	BAJA	MEDIA
2 FTR	BAJA	MEDIA	ALTA
3 o más FTR	MEDIA	ALTA	ALTA

4.3. Complejidad de un EQ

Complejidad de un EQ			
	0 a 5 DET	6 a 19 DET	20 o más DET
0 a 1 FTR	BAJA	BAJA	MEDIA
2 a 3 FTR	BAJA	MEDIA	ALTA
4 o más FTR	MEDIA	ALTA	ALTA

4.4. Complejidad de un EO

Complejidad de un EO			
	1 a 5 DET	6 a 19 DET	20 o más DET
0 a 1 FTR	BAJA	BAJA	MEDIA
2 a 3 FTR	BAJA	MEDIA	ALTA
4 o más FTR	MEDIA	ALTA	ALTA

4.5. Tabla de Transformación

Tabla de Transformaciones					
	IFL	EIF	EI	EQ	EO
BAJA	7	5	3	3	4
MEDIA	10	7	4	4	5
ALTA	15	10	6	6	7

4.6. Tabla de Transformación LCF/PF

Tabla de Transformaciones	
Lenguaje	LCF/PF
Ensamblador	320
C	150
Cobol	106
Pascal	91
Basic	64
TCL	64
Java	53
C++	29

ANEXO K. ESTRUCTURA DE PAQUETES

1. Introducción

El presente es el documento que muestra en detalle la estructura de paquetes que se manejó para el uso organizado de las clases. Se agrupó de acuerdo a la funcionalidad que representan para el Sistema.

1.1 Propósito

El propósito principal de este documento es el de proporcionar un mayor entendimiento sobre la estructura que se utilizó para organizar las clases en paquetes.

1.2 Alcance

Este documento consta de un listado detallado de todos los paquetes que se utilizaron además de una pequeña descripción del contenido de cada paquete.

1.3 Visión general del Documento

Este documento consta de dos secciones. Esta sección es la introducción y proporciona una visión general del contenido del documento. En la sección dos se muestra el listado detallado de los paquetes que se usaron.

2. Descripción de Paquetes

Nombre de Paquete	Descripción
comun	Contiene las clases en común para los 3 módulos del proyecto: Análisis, Extracción y Explotación. La estructura de este paquete es la misma usada para cada módulo, la cual se detalla luego.
extraccion	Es el paquete global del módulo de Extracción. En el siguiente nivel de este directorio se encuentra los paquetes principales del módulo.
applet	Contiene las clases que administran el funcionamiento del applet. Se encuentra subdividido en 3 paquetes: arbol, dib, panel.
arbol	Contiene las clases que administran el árbol de navegación de objetos, además de las clases que permiten el drag & drop del árbol al área de trabajo.
dib	Contiene las clases que administran la lógica de los componentes que se dibujan en el área de trabajo.
panel	Contiene las clases que administran el panel principal del applet, controlan diversas funciones como el zoom en el área de trabajo por ejemplo.
beans	Contiene las clases que representan los beans de respaldo de las páginas JSF que se muestran al usuario.
bundle	Contiene los archivos de recursos que administran los mensajes que se muestran al usuario en las páginas JSF.
clases	Contiene las clases que representan el dominio del Sistema, guarda la información de los elementos de extracción configurados por el usuario.
controles	Contiene las clases que administran las diversas funciones que se realizan con las clases de Dominio y la comunicación entre éstas.
excepciones	Contiene las clases que administran las excepciones que se presentan en la ejecución del sistema.
gestores	Contiene las clases que manejan la lógica de los objetos del dominio.
servlet	Contiene el programa que se ejecuta en el servidor para administrar las programaciones que se han hecho en el Sistema.
xml	Agrupar las clases que controlan la persistencia de la metadata. Contiene sólo un paquete: beans.
beans	Contiene las clases que se leen y escriben a disco, para una posterior recuperación.
seguridad	Contiene las clases que administran la seguridad de la herramienta.



ANEXO L. ESTRUCTURA DE DESAGREGACIÓN DEL TRABAJO

1. Propósito

1.1. Objetivos

El presente documento tiene como objetivo brindar un panorama general de todo el proyecto, mostrando la manera como se han descompuesto y distribuido las tareas necesarias para completarlo.

Mediante el uso de este documento se espera tener mayor facilidad para el planeamiento, organización y control del proyecto pues permite visualizar directamente que tareas deben realizarse para poder completar cada parte del desarrollo.

1.2. Alcance

El presente documento de estructura de desagregación del trabajo se aplica al módulo de Extracción de AxeBit tomando en consideración todas las etapas de la realización de la herramienta, la misma que es descrita a lo largo de todo el documento de tesis.

2. Estructura de Desagregación del Trabajo

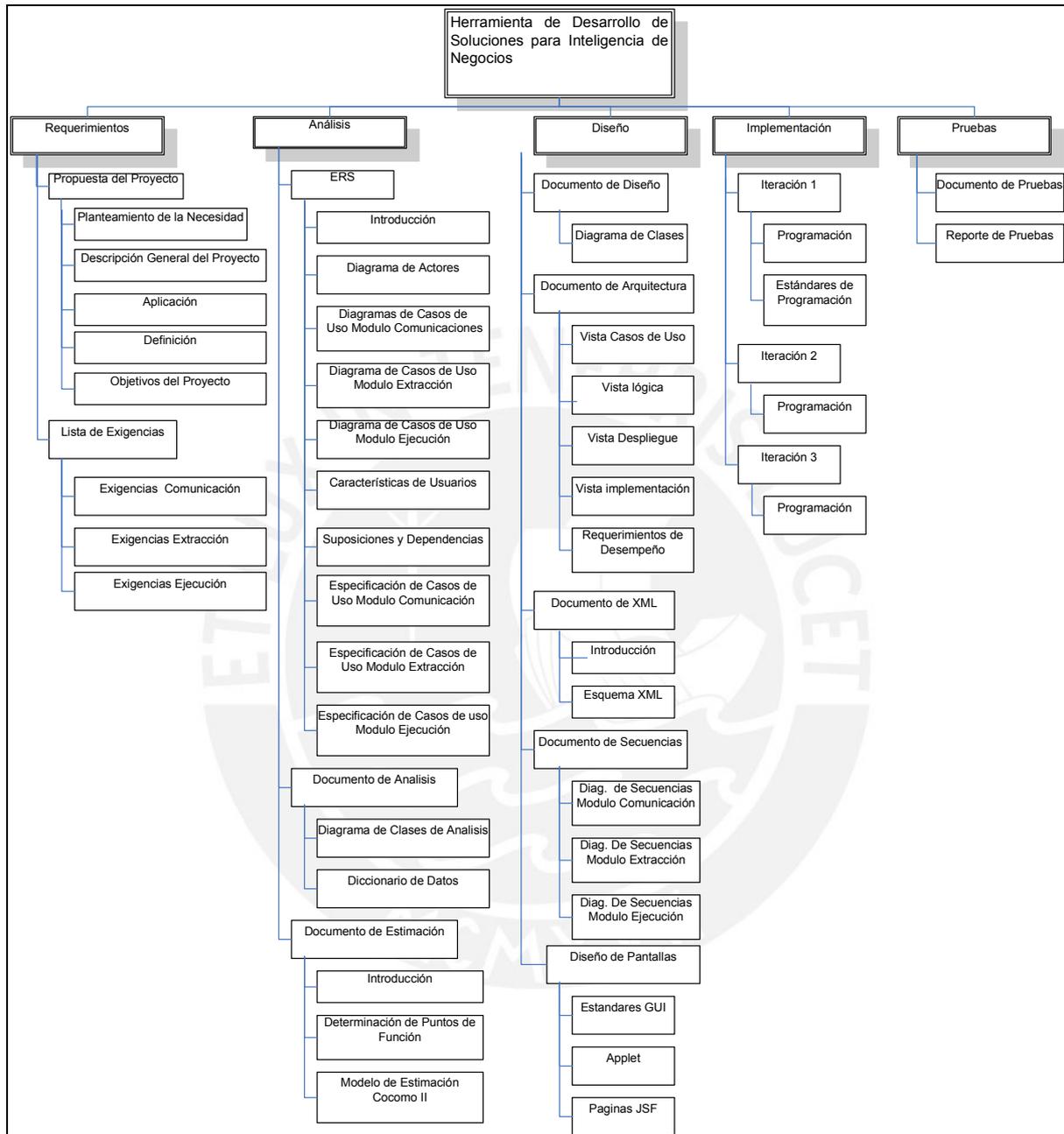


Ilustración 2: Diagrama de Estructura de Desagregación del Trabajo

2.1 Lista de Tareas

1. Requisitos
 - 1.1. Propuesta del Proyecto
 - 1.1.1 Planteamiento de la Necesidad
 - 1.1.2 Descripción General del Proyecto
 - 1.1.3 Aplicación
 - 1.1.4 Definición

- 1.1.5. Objetivos del Proyecto
- 1.2. Lista de Exigencias
 - 1.2.1. Exigencias de Comunicación
 - 1.2.2. Exigencias de Extracción
 - 1.2.3. Exigencias de Ejecución
- 2. Análisis
 - 2.1. ERS
 - 2.1.1. Introducción
 - 2.1.2. Diagrama de Actores
 - 2.1.3. Diagrama de Casos de Uso Modulo Comunicación
 - 2.1.4. Diagrama de Casos de Uso Modulo Extracción
 - 2.1.5. Diagrama de Casos de Uso Modulo Ejecución
 - 2.1.6. Características de Usuarios
 - 2.1.7. Suposiciones y Dependencias
 - 2.1.8. Especificación de Casos de Uso Módulo Comunicación
 - 2.1.9. Especificación de Casos de Uso Módulo Extracción
 - 2.1.10. Especificación de Casos de Uso Módulo Ejecución
 - 2.2. Documento de Análisis
 - 2.2.1. Diagramas de Clases de Análisis
 - 2.2.2. Diccionario de Datos
 - 2.3. Documento de Estimación
 - 2.3.1. Introducción
 - 2.3.2. Determinación de Puntos de Función
 - 2.3.3. Modelo de Estimación Cocomo II
- 3. Diseño
 - 3.1. Documento de Diseño
 - 3.1.1. Diagrama de Clases
 - 3.2. Documento de Arquitectura
 - 3.2.1. Vista de Casos de Uso
 - 3.2.2. Vista Lógica
 - 3.2.3. Vista de Despliegue
 - 3.2.4. Vista de Implementación
 - 3.2.5. Requerimientos y Desempeño
 - 3.3. Documento de XML
 - 3.3.1. Introducción
 - 3.3.2. Esquema XML
 - 3.4. Documento de Secuencias
 - 3.4.1. Diagrama de Secuencias Módulo Comunicación
 - 3.4.2. Diagrama de Secuencias Módulo Extracción
 - 3.4.3. Diagrama de Secuencias Módulo Ejecución
 - 3.5. Diseño de Pantallas
 - 3.5.1. Estándares GUI
 - 3.5.2. Applet
 - 3.5.3. Páginas JSF
- 4. Implementación
 - 4.1. Iteración 1
 - 4.1.1. Programación
 - 4.1.2. Estándares de Programación
 - 4.2. Iteración 2
 - 4.2.1. Programación
 - 4.3. Iteración 3
 - 4.3.1. Programación
- 5. Pruebas
 - 5.1. Documento de Pruebas
 - 5.2. Reporte de Pruebas