

Universidad de las Ciencias Informáticas.

Facultad 4



Título:

Aplicación de administración y monitorización para  
la plataforma Génesis.

Autor:

Lissuan Fadruga Artiles

Tutor: Lic. Edisel Navas Conyedo.

Cotutor: Ing. Yunior Alberto Gonzales Nuzzo.

# Declaración de autoría

Declaramos que somos los únicos autores de este trabajo y autorizamos a la Facultad 4 de la Universidad de las Ciencias Informáticas a hacer uso del mismo en su beneficio.

Para que así conste firmamos la presente a los \_\_\_\_\_ días del mes de \_\_\_\_\_ del año \_\_\_\_\_.

Autor:

\_\_\_\_\_

Lissuan Fadruga Artilles  
Firma del Autor

Tutor:

\_\_\_\_\_

Lic. Edisel Navas Conyedo  
Firma del Tutor.

Agradecimientos:

Dedicatoria:

## Resumen:

Existe una tendencia al crecimiento del volumen de información administrado por los procesos de negocio de las empresas. Pequeñas, medianas y especialmente grandes instituciones invierten gran cantidad de recursos para el procesamiento de la información que generan sus actividades. En muchos casos la adquisición de superordenadores no es una alternativa viable, especialmente para países subdesarrollados como el nuestro. La computación paralela se presenta como una vía de solución menos costosa para el procesamiento de grandes cantidades de datos. Génesis, primer proyecto del Equipo de Investigación, Procesamiento y Análisis de Datos (EIPAD), es una plataforma para el cómputo paralelo de datos y cálculos que ofrece un conjunto de herramientas genéricas para desarrollar aplicaciones para procesamiento paralelo. Con la incorporación de un módulo de cálculo específico se genera un producto o un servicio que soluciona un problema puntual de cómputo paralelo. El presente trabajo está dirigido por la disyuntiva que tiene la plataforma Génesis sobre su interacción con los usuarios. Génesis no tiene definida una interfaz de administración y monitorización de los recursos que involucra, los usuarios que acceden a ella y los reportes que pueden generarse como resultado de los procesos de cálculo que realiza. Este trabajo tiene como objetivo fundamental realizar una investigación de las tecnologías y herramientas que hay en la actualidad para realizar un sistema informático que pueda llevar el control de los usuarios que acceden a la plataforma Génesis así como la administración y monitorización de los procesos que se llevan a cabo en la misma.

# Índice:

Introducción:.....	8
Capítulo 1 .....	11
Fundamentación teórica.....	11
1.1 Introducción .....	11
1.2 Presentación de los datos.....	11
1.3 Aplicaciones interfaces de usuarios para sistemas de cómputo paralelo actuales.....	12
1.3.1 Las Grid. Globus Toolkit .....	12
1.4 Tecnologías actuales a utilizar.....	14
1.4.7 Lenguaje Unificado de Modelado (UML) .....	26
1.4.8 SGBD.....	26
1.4.9 Herramienta CASE.....	28
1.5 Conclusiones .....	29
Capítulo 2 .....	30
Descripción del sistema.....	30
2.1 Introducción.....	30
2.2 Problema y Situación problemática.....	30
2.3 Objeto de automatización .....	30
2.3.1 Propuesta del sistema.....	31
2.3.2 Requerimientos del sistema.....	31
2.4 Modelo de sistema.....	34
2.4.1 Modelo de caso de uso del sistema.....	34
2.4.2 Descripción de los CU del sistema.....	34
Capítulo 3 .....	45
Análisis y Diseño del sistema.....	45
3.1 Introducción .....	45
3.2 Modelo de Análisis .....	45
3.3 Modelo de Diseño .....	46
3.4 Definiciones de diseño que se aplican .....	53
3.5 Tratamiento de errores.....	53
3.6 Seguridad .....	54
3.7 Interfaz.....	54
3.8 Conclusiones.....	54
Capítulo 4 .....	55
Implementación y prueba.....	55

4.1 Introducción .....	55
4.2 Implementación .....	55
3.3 Modelo de prueba .....	58
3.4 Conclusiones .....	60
Conclusiones.....	61
Recomendaciones .....	62
Anexos.....	66
Glosario de Términos.....	105

# Introducción:

En la actualidad, existe una tendencia al crecimiento de las prestaciones de las aplicaciones que necesitan procesar grandes volúmenes de datos en tiempos moderados, por lo que muchas instituciones han acelerado los procesos de investigación en el área del procesamiento paralelo de datos aprovechando su infraestructura tecnológica para mejorar el proceso de análisis de grandes volúmenes de datos, guiando su trabajo hacia una de las soluciones más brillantes a problemas de gran envergadura.

Por lo general la implementación de un algoritmo para cálculo paralelo necesita del conocimiento de los datos a procesar y las características del algoritmo que facilita la subdivisión del mismo, permitiendo que varios procesadores trabajando en conjunto para realizar una tarea en común pueda asumir cada uno una porción del problema, e intercambien datos a través de memorias o redes, para llegar a una solución en conjunto. El cómputo paralelo de datos ofrece grandes ventajas como la solución de problemas que parecían imposibles de resolver hasta la actualidad, la aceleración de la velocidad de ejecución de los cálculos y la disminución de los costos de prestaciones y servicios así como los recursos que se invierten. La computación paralela utilizada para el procesamiento de grandes volúmenes de datos tiene gran aplicación en áreas como la Biotecnología, los sistemas para Toma de Decisiones en Tiempo Real, Sistemas para predicciones económicas, la Física Computacional, la Minería de Datos, Astrofísica entre otras.

Génesis, primer proyecto del Equipo de Investigación, Procesamiento y Análisis de Datos, es una plataforma para el procesamiento paralelo de datos y el cálculo. Ofrece una arquitectura flexible que permite la creación de una línea de productos con estos requerimientos a partir de las especificaciones de un módulo de cómputo. Un módulo de cómputo es la documentación e implementación de un modelo matemático que plantea una solución paralelizable a un determinado problema y cuya implementación se basa en las especificaciones que ofrece Génesis.

Está diseñada en tres partes principales, un Motor de Procesamiento de Peticiones Paralelas, que constituye el núcleo de la plataforma para la planificación y control de los procesos de cómputo que se ejecutan; un Subsistema de Datos, cuyo objetivo es garantizar la disponibilidad, accesibilidad y persistencia de los datos ofreciendo soporte para el

procesamiento de grandes volúmenes de datos en la plataforma y una Aplicación de Administración y Monitorización para la plataforma.

En la actualidad Génesis no cuenta con la aplicación de Administración y Monitorización de los procesos de cómputo que se realizan en la plataforma, la colaboración de usuarios, un sistema de reportes de registros de eventos e historial de tareas y que permita facilidades de manipulación de los clientes que realizan las solicitudes de cómputo. A partir del análisis de la situación problemática presente se identifica el siguiente **problema**: ¿Cómo elaborar una interfaz de comunicación que permita facilidades de interacción de los usuarios con la plataforma Génesis? Este problema se enmarca en el **objeto de estudio**: Gestión del cómputo paralelo de datos y su **campo de acción** abarcado es la gestión del cómputo paralelo de datos de la plataforma Génesis.

**Objetivo general:**

Diseñar e implementar una aplicación de administración y monitorización para la plataforma Génesis.

**Objetivos específicos:**

1. Obtener la descripción del sistema a partir del análisis de los procedimientos establecidos para la utilización de la plataforma Génesis.
2. Diseñar e implementar una aplicación informática de administración y monitorización acoplada a la plataforma Génesis.
3. Desarrollar un proceso de pruebas de integración a los componentes al sistema para validar la solución propuesta.
4. Permitir al usuario la monitorización de las solicitudes de cómputo que realiza.

Para guiar la investigación se plantea la siguiente **hipótesis**: Si se desarrolla una aplicación informática de administración y monitorización para la plataforma Génesis se logrará una interfaz de comunicación con facilidades de interacción de los usuarios con la misma.

Las **tareas** que se llevaran a cabo para darle cumplimiento a los objetivos trazados son:

1. Fundamentación teórico-metodológica del problema:
  1. Analizar los procedimientos a seguir para la utilización de la plataforma Génesis.
  2. Realizar un estudio y análisis de las tecnologías y herramientas existentes y selección de las utilizadas para el desarrollo.

3. Escoger una metodología adecuada que rijan el proceso de desarrollo de software de la aplicación.
2. Elaboración de la propuesta.
  1. Identificar y documentar los requerimientos de la aplicación a desarrollar.
  2. Realizar el análisis y el diseño del sistema utilizando el proceso de software seleccionado a partir de los requerimientos identificados.
  3. Implementar la interfaz que facilite la comunicación de los usuarios con la plataforma Génesis.
3. Redacción del informe de investigación.

El presente trabajo consta de Introducción, 3 capítulos, 20 epígrafes, Conclusiones, Recomendaciones, Referencias Bibliográficas, Bibliografía y un Glosario de Términos.

En el **Capítulo 1 Fundamentación Teórica** se abordan los aspectos esenciales que intervienen en la creación de interfaces de usuarios y la necesidad de aplicarlos en el sistema. Se realiza un estudio de las aplicaciones para administración e interfaces de usuarios de las principales tecnologías para cómputo paralelo existentes y un análisis de las tendencias y tecnologías en la actualidad para el desarrollo del tipo de aplicaciones que se pretende elaborar justificando las seleccionadas.

En el **Capítulo 2 Características del sistema** se describe el objeto de estudio en el que se enmarca el problema, destacando la situación problemática, el objeto de automatización, la propuesta del sistema y la especificación de los requerimientos funcionales y no funcionales del software que sirven de base a las actividades posteriores al ciclo de desarrollo. Se describe el modelo de casos de usos del sistema que satisface los requerimientos identificados.

En el **Capítulo 3 Análisis y Diseño del Sistema** se presentan los artefactos resultantes de los flujos de trabajo Análisis y Diseño durante el ciclo de desarrollo, destacándose el modelo de análisis, el modelo de diseño y el diseño de la base de datos. Se hace referencia a los aspectos del diseño como las definiciones que se aplican, el tratamiento de errores, la seguridad e interfaz.

En el **Capítulo 4 Implementación y Prueba** se muestran los Diagramas de Componentes desarrollados a partir de los artefactos obtenidos en las actividades de análisis y diseño de la

aplicación. Se presenta el Diagrama de Despliegue propuesto y los Casos de Prueba diseñados a los que fue sometida la aplicación.

# Capítulo 1

## Fundamentación teórica

### *1.1 Introducción*

En el presente capítulo se abordan los aspectos esenciales que intervienen en la creación de interfaces de usuarios y la necesidad de aplicarlos en el sistema. Se realiza un estudio de las aplicaciones para administración e interfaces de usuarios de las principales tecnologías para cómputo paralelo existentes y un análisis de las tendencias y tecnologías en la actualidad para el desarrollo del tipo de aplicaciones que se pretende elaborar justificando las seleccionadas.

### *1.2 Presentación de los datos.*

La manipulación de grandes cantidades de datos involucra actividades como la validación de los mismos, el formato y observación de los resultados del procesamiento al cual fueron sometidos. Tener una forma de presentar los datos a los usuarios es una premisa en el desarrollo de las aplicaciones de software de hoy en día. La selección de las tecnologías y herramientas para implementar al máximo los requisitos de una buena interfaz es un reto para los desarrolladores de software en el mundo entero. Esta capa de presentación es de vital importancia porque representa la interacción del usuario con el resto de la plataforma Génesis. Para una buena presentación deben tenerse en cuenta algunas características como estas:

- **Validación de los datos de entrada.** A través de la validación de los datos se garantiza que la aplicación reciba los datos a procesar exactamente en el formato en el que los necesita, contribuyendo al tratamiento de errores en la aplicación y minimizando la carga de trabajo del resto de las capas.

- **Navegabilidad del sistema.** La navegabilidad debe estar perfectamente planeada, guiar a los usuarios hacia la información que requieren de forma rápida e intuitiva es el objetivo de toda aplicación desarrollada en tecnología web. [1].
- **Disponibilidad del servicio.** Lograr que el sistema este disponible cuando el usuario lo necesite es una de las premisas fundamentales para un buen servicio.
- **Presentación de los datos de salida.** Presentar los datos de salida se refiere a la variedad de formas en que se pueden mostrar como tablas, gráficas, reportes, donde el usuario le pueda dar una interpretación correcta y así darle una utilidad mayor a la aplicación.
- **Diseño consistente de las interfaces:** Las interfaces deben ser diseñadas con toda intención, de manera que sean fáciles de utilizar, permitan buena comunicación y tengan un diseño agradable a la vista del usuario. Una interfaz gráfica de usuario consistente reduce el tiempo de aprendizaje de las aplicaciones.

Una correcta elección de la arquitectura para la capa de presentación es fundamental para el éxito en general de la aplicación. La arquitectura seleccionada proporcionara el equilibrio entre la experiencia del usuario, la facilidad de desarrollo, y los requisitos operativos de la aplicación. Los usuarios demandan cada vez más que las aplicaciones construyan a partir de sus necesidades y requisitos.

### ***1.3 Aplicaciones interfaces de usuarios para sistemas de cómputo paralelo actuales.***

Existe una gran variedad de aplicaciones de software, técnicas y arquitecturas definidas para el cómputo paralelo de datos en la actualidad. La tendencia al desarrollo vertiginoso de este tipo de aplicaciones esta dada por la flexibilidad que permite la computación paralela y los beneficios asociados a esta, lo que posibilita a las instituciones científicas crear soluciones adaptadas a sus propios problemas . Sin embargo no todas las tecnologías traen asociadas interfaces cómodas para la interacción con los usuarios, algunas permiten la configuración a través de comandos ejecutados en una consola, otras proveen interfaces pero con muy pocos requerimientos cumplidos.

#### ***1.3.1 Las Grid. Globus Toolkit***

La computación Grid [2] es una tecnología que permite utilizar de forma coordinada todo tipo de recursos (entre ellos cómputo, almacenamiento y aplicaciones específicas) que no están

sujetos a un control centralizado. En este sentido es una nueva forma de computación distribuida, en la cual los recursos pueden ser heterogéneos (diferentes arquitecturas, supercomputadores, clúster...) y se encuentran conectados mediante redes de área extensa. La tecnología Grid actual ofrece la funcionalidad mínima necesaria para, de forma transparente y segura, compartir y explotar simultáneamente los recursos pertenecientes a diferentes organizaciones, siempre respetando sus propias políticas y procedimientos de seguridad y gestión de recursos. Sin embargo, y a pesar de la gran inversión científica, esta tecnología necesita de grandes mejoras para cumplir las altas expectativas que la comunidad científica, el entorno empresarial y la sociedad han puesto en ella. Como cada proyecto grid tiene un conjunto nuevo de necesidades que deben ser acercadas por una combinación única de componentes de software. Por ejemplo Globus Toolkit utiliza un conjunto de servicios y librerías para la monitorización de recursos, descubrimiento, administración, garantiza la seguridad y la colaboración entre proyectos distribuidos. [3] El desarrollo de estas aplicaciones para el desarrollo de grid es muy amplia, pero limitada a ese entorno de desarrollo, no son aplicaciones que se puedan adaptar a otros proyectos o campos que no sea el de las grid, no se puede implantar un middleware a un clúster que use PVM o otras aplicaciones.

### **1.3.2 PVM (Parallel Virtual Machine): XPVM**

PVM, acrónimo de Parallel Virtual Machine, crea una nueva abstracción, que es la máquina virtual paralela, consumiendo los recursos computacionales de los ordenadores libres de la red que se ponen a disposición de la librería PVM [4]. XPVM es el proveedor de una interfaz gráfica de la configuración local de la máquina virtual, así como una codificación visual de la actividad llevada a cabo en cada nodo de la máquina virtual, enviando órdenes e información conjuntamente con varias vistas animadas para controlar o depurar la ejecución de programas PVM [5]. Esta interfaz carece de una administración de los usuarios que actúan con la aplicación, está limitado sólo cuando se instala la aplicación PVM a partir de la versión 3.3 o posterior en el ordenador donde se tenga instalada la aplicación y no realiza una presentación de los datos para facilitarle el trabajo al usuario.

Estas tecnologías, tanto las herramientas de desarrollo de Grid Globus Toolkit como XPVM se utilizan en campos diferentes, pero limitadas al entorno de desarrollo que se enmarcan por ejemplo no se puede aplicar Globus Toolkit a aplicaciones que usen PVM ni viceversa, no son aplicaciones que se puedan adaptar a otros proyectos o campos fuera del cual están diseñadas. Marcadas son las diferencias que tienen estas herramientas con la presentación de una interfaz de usuario para la plataforma Génesis, que trata de administrar y proveer al

usuario de una herramienta con la que se pueda manejar el sistema, así como tener control de los usuarios que la utilizan, mostrar resultados al usuario así como de proveer información sobre el procesamiento o cálculo que se está realizando.

## ***1.4 Tecnologías actuales a utilizar***

### ***1.4.1 Arquitectura Cliente / Servidor***

La arquitectura cliente/servidor es "la tecnología que proporciona al usuario final el acceso transparente a las aplicaciones, datos, servicios de cómputo o cualquier otro recurso del grupo de trabajo. El modelo soporta un medio ambiente distribuido en el cual los requerimientos de servicio hechos por estaciones de trabajo inteligentes o 'clientes', resultan en un trabajo realizado por otros computadores llamados servidores." "Es un modelo para construir sistemas de información, que se sustenta en la idea de repartir el tratamiento de la información y los datos por todo el sistema informático, permitiendo mejorar el rendimiento del sistema global de información". (GUTIÉRREZ 2005). [23]

Algunas ventajas de la arquitectura cliente/servidor:

- **Aumento de la productividad:** Los usuarios pueden utilizar herramientas que le son familiares, como hojas de cálculo y herramientas de acceso a bases de datos. Mediante la integración de las aplicaciones cliente/servidor con las aplicaciones personales de uso habitual, los usuarios pueden construir soluciones particularizadas que se ajusten a sus necesidades cambiantes.
- **Menores costes de operación:** Permiten un mejor aprovechamiento de los sistemas existentes, Protegiendo la inversión. Por ejemplo, el compartir los servidores (habitualmente caros) y dispositivos periféricos (como impresoras) entre máquinas clientes permite un mejor rendimiento del conjunto. Proporcionan un mejor acceso a los datos. La interfaz de usuario ofrece una forma homogénea de ver el sistema, independientemente de los cambios o actualizaciones que se produzcan en él. El movimiento de la ubicación de la información de funciones desde un ordenador central hacia servidores o clientes locales origina el desplazamiento de los costes de ese proceso hacia máquinas más pequeñas y por tanto, más baratas.
- **Mejora en el rendimiento de la red:** Las arquitecturas cliente/servidor eliminan la necesidad de mover grandes bloques de información por la red hacia los ordenadores

personales o estaciones de trabajo para su proceso. Los servidores controlan los datos, procesan peticiones y después transfieren sólo los datos requeridos a la máquina cliente. Entonces, la máquina cliente presenta los datos al usuario mediante interfaces amigables. Todo esto reduce el tráfico de la red, lo que facilita que pueda soportar un mayor número de usuarios. Tanto el cliente como el servidor pueden escalarse para ajustarse a las necesidades de las aplicaciones.

- **La existencia de varias unidades de procesamiento proporciona una red más fiable:** un fallo en uno de los equipos no significa necesariamente que el sistema deje de funcionar. En una arquitectura como ésta, los clientes y los servidores son independientes los unos de los otros con lo que pueden renovarse para aumentar sus funciones y capacidad de forma independiente, sin afectar al resto del sistema. La arquitectura modular de los sistemas cliente/servidor permite el uso de ordenadores especializados (servidores de base de datos, servidores de ficheros, etc.). Permite centralizar el control de sistemas que estaban descentralizados, como por ejemplo la gestión de los ordenadores personales que antes estuvieran aislados.

#### **1.4.2 Servidor web:**

- ❖ **Apache:** Es uno de los servidores web más utilizado en la actualidad. Su robustez y estabilidad hacen que cada vez millones de servidores reiteren su confianza en este software. En Abril del año 2009 se monitoreaban en la red alrededor de 231, 510,169 sitios web. De ellos Apache lideraba con 106 millones de sitios aproximadamente, seguido por Microsoft Internet Information Server con 67 millones. [19]

Presenta entre otras características: reporte de errores es altamente configurable, posee bases de datos de autenticación y negociado de contenido. Su flexible sistema modular, permite cargar y descargar módulos sin necesidad de tocar el núcleo de la aplicación. Dispone de una herramienta (APXS) que facilita la compilación e instalación de estos módulos, ya sean del mismo Apache o desde terceras partes. Los módulos se cargan en memoria cuando los necesita y se descargan automáticamente cuando dejan de utilizarse. El proceso de instalación de Apache varía dependiendo del tipo de sistema operativo que se emplea y de si se tienen o no privilegios de administrador. Apache permite personalizar la respuesta lo ante posible ante errores que se puedan dar en el servidor a través de la configuración de scripts para errores específicos.

Permite la creación de ficheros de registros de información para el control de los sucesos en el servidor. Soportado por varias plataformas como Unix (BSD, GNU/Linux, entre otros), Windows, Macintosh. Incluye módulos que se cargan de forma dinámica, soporta CGI, Perl, PHP, contiene soporte para Bases de datos. Es una tecnología gratuita de código abierto. [6]

### **1.4.3 Navegadores Web**

#### **Mozilla Firefox**

Mozilla Firefox es el navegador web de código abierto por excelencia. Sus principales potencialidades son:

- **Navegación con pestañas:** Presenta una forma rápida y eficiente de navegar por la web, que le permite abrir varias páginas en una misma ventana dentro de pestañas separadas. Abre enlaces en segundo plano mientras está leyendo otra página, continuando luego con los enlaces cuando haya terminado, y las páginas están disponibles cuando las necesite, produciendo una sensación de mayor velocidad incluso en conexiones lentas.[20]
- **Bloqueo de ventanas emergentes:** La publicidad mediante ventanas emergentes es una de los aspectos más molestos de la navegación en la web. Mozilla Firefox protege la publicidad de ventanas emergentes no solicitadas. También proporciona el control para permitir a ciertos sitios abrir ventanas emergentes, si es necesario para que funcionen.
- **Privacidad y seguridad:** Firefox mantiene el ordenador seguro del “spyware” malicioso al no cargar los controles ActiveX dañinos. Posee buenos índices de privacidad con un selecto conjunto de herramientas encargadas para mantener la actividad en línea segura.
- **Ajustado a las necesidades del usuario:** Simple e intuitivo, aunque con muchas características, Firefox tiene todas las funciones a las comunes que encuentran los usuarios en el resto de los navegadores como marcadores, historial, pantalla completa, aumento del tamaño del texto para hacer que las páginas con textos pequeños sean más fáciles de leer, entre otras.
- **Muy utilizado para el desarrollo:** Provee un conjunto de herramientas estándar para desarrolladores incluyendo una potente consola de errores/avisos de JavaScript y CSS, y un Inspector de documentos opcional que le proporciona una vista con información detallada sobre las páginas. (WILKINSONPC 1998-2007).[24]

Existen otras características que posee Mozilla Firefox por las que se ha incrementado considerablemente su número de usuarios, entre las que se destacan la barra de direcciones inteligente para el acceso rápido a sitios ya visitados, el rendimiento mejorado consumiendo menos memoria del ordenador, adición de marcadores y organización rápida de las páginas web favoritas, administrador de contraseñas. Evita los fraudes en línea, transacciones no seguras y falsificaciones mediante una identificación sencilla de sitios, mejoras en el motor gráfico, navegación en modo desconectado y posee asistencia en directo con un conjunto de expertos para el asesoramiento.

### **Internet Explorer.**

Internet Explorer es el navegador web más utilizado en la actualidad. Navega de forma sencilla entre las pestañas abiertas, aunque esta característica no estuvo soportada hasta su versión 7.0. Las pestañas se pueden agrupar y guardar en categorías lógicas, lo que posibilita abrir varias con un solo clic. Se puede establecer fácilmente un grupo de pestañas como grupo de la página principal, de modo que se abra todo el conjunto cada vez que se inicia Internet Explorer. Se ha vuelto a diseñar la interfaz y se ha mejorado para maximizar el área de la pantalla donde se muestra la página web, de modo que se ve mejor la parte necesaria y menos la que no lo es. Entre las opciones de impresión se encuentran los márgenes ajustables, los diseños de páginas personalizables, los encabezamientos y pies de página prescindibles y una opción para imprimir sólo el texto seleccionado. Permite realizar búsquedas en Internet a través del proveedor preferido desde el cuadro de búsqueda de la barra de herramientas, con lo que se evita la acumulación de barras. Es muy fácil elegir un proveedor: sólo hay que seleccionarlo en la lista desplegable, a la que además se pueden agregar otros proveedores. Permite acceso rápido y sencillo a los sitios favoritos, grupos de pestañas, historial de exploración y suscripciones a fuentes RSS. Permite la ampliación de una página web en concreto, incluidos el texto y los gráficos, para centrarse en un contenido específico o para facilitar la accesibilidad a usuarios con problemas de visión.

### ***1.4.4 Lenguajes de programación para la Web***

Existen dos grupos de lenguajes de programación para la Web clasificándose en dependencia de dónde se implementan siguiendo la arquitectura Cliente/Servidor. De esta forma un grupo es llamado lenguajes de lado del Cliente y el otro grupo lenguajes del lado del Servidor. En el

primer grupo entre los más usados se encuentran: **JavaScript, Script, XML, Java**. En el segundo grupo se destacan: **PHP, PERL, ASP, JSP**.

### ***Del lado del cliente:***

#### **JavaScript**

Es un lenguaje de programación del lado del cliente, porque es el navegador el que soporta la carga de procesamiento. Gracias a su compatibilidad con la mayoría de los navegadores modernos, es el lenguaje de programación del lado del cliente más utilizado. Con JavaScript se pueden crear efectos especiales en las páginas y definir interactividades con el usuario. El navegador del cliente es el encargado de interpretar las instrucciones JavaScript y ejecutarlas para realizar estos efectos e interactividades, de modo que el mayor recurso con que cuenta este lenguaje es el propio navegador. Es un lenguaje de programación bastante sencillo y pensado para hacer las cosas con rapidez, a veces con ligereza. Incluso las personas que no tengan una experiencia previa en la programación podrán aprender este lenguaje con facilidad y utilizarlo en toda su potencia con sólo un poco de práctica. Permite la programación de pequeños scripts, pero también de programas más grandes, orientados a objetos, con funciones y estructuras de datos complejas. Además, JavaScript pone a disposición del programador todos los elementos que forman la página Web, para que éste pueda acceder a ellos y modificarlos dinámicamente. (ALVAREZ 2003). [25]

#### **XML**

XML es un Lenguaje de Etiquetado Extensible muy simple, pero estricto que juega un papel fundamental en el intercambio de una gran variedad de datos. Es un lenguaje muy similar a HTML pero su función principal es describir datos y no mostrarlos como es el caso de HTML. XML es un formato que permite la lectura de datos a través de diferentes aplicaciones. Las tecnologías XML son un conjunto de módulos que ofrecen servicios útiles a las demandas más frecuentes por parte de los usuarios. XML sirve para estructurar, almacenar e intercambiar información.

Entre las tecnologías XML disponibles se destacan:

**XSL:** Lenguaje Extensible de Hojas de Estilo, cuyo objetivo principal es mostrar cómo debería estar estructurado el contenido, cómo debería ser diseñado el contenido de origen y cómo debería ser paginado en un medio de presentación como puede ser una ventana de un navegador Web o un dispositivo de mano, o un conjunto de páginas de un catálogo, informe o libro.

**XPath:** Lenguaje de Rutas XML, es un lenguaje para acceder a partes de un documento XML.

**XLink:** Lenguaje de Enlace XML, es un lenguaje que permite insertar elementos en documentos XML para crear enlaces entre recursos XML.

**XPointer:** Lenguaje de Direccionamiento XML, es un lenguaje que permite el acceso a la estructura interna de un documento XML, esto es, a sus elementos, atributos y contenido.

**XQL:** Lenguaje de Consulta XML, es un lenguaje que facilita la extracción de datos desde documentos XML. Ofrece la posibilidad de realizar consultas flexibles para extraer datos de documentos XML en la Web. ((W3C) 1994-2005). [26]

## **Ajax**

AJAX, acrónimo de **Asynchronous JavaScript And XML** (JavaScript y XML asíncronos), es una técnica de desarrollo web para crear aplicaciones interactivas. Éstas se ejecutan en el cliente, es decir, en el navegador del usuario, y mantiene comunicación asíncrona con el servidor en segundo plano. De esta forma, es posible realizar cambios sobre la misma página sin necesidad de recargarla. Esto significa aumentar la interactividad, velocidad y usabilidad en la misma. [7]

AJAX incorpora:

- Presentación basada en estándares usando XHTML y CSS
- Exhibición e interacción dinámicas usando el Document Object Model.
- Intercambio y manipulación de datos usando XML y JSON
- Recuperación de datos asíncrona usando XMLHttpRequest.
- JavaScript.

## **ExtJs.**

ExtJs. Es usado principalmente en aplicaciones web donde se requiere un alto nivel de interacción con el usuario. ExtJs posibilita la sencillez en el desarrollo de una aplicación web por las facilidades que brinda:

- Provee fácil navegación del usuario a través de la aplicación brindando elementos gráficos y de información como los grid, ventanas, combos y formularios.
- Interacción mejorada del usuario y el navegador a través del manejador de eventos respondiendo a tecleos del usuario, clic del ratón (mouse), y monitoreo de eventos en el navegador redimensionando ventanas y cambiando estilos.
- Combina la tecnología Ajax para acelerar la recuperación de la información de manera fácil y rápida, consumiendo pocos recursos.

- Brinda un conjunto de funciones predefinidas por la librería para comunicación con Ajax, usando JSON además de facilitar el trabajo con objetos, mejoramiento y rapidez en la construcción de interfaces más cómodas para los usuarios.
- Provee una documentación extensa con ejemplos sobre el trabajo con las funciones y propiedades de la librería.

ExtJs. Es compatible con casi todos de los navegadores más utilizados en la actualidad:

- Internet Explorer 6+
- Firefox 1.5+(PC, Mac)
- Safari 2+
- Opera 9 (PC, Mac). [8]

### ***Del lado del servidor:***

#### **PHP**

El PHP (acrónimo de PHP: Hypertext Preprocessor), es un lenguaje interpretado de alto nivel embebido en páginas HTML y ejecutado en el servidor. La meta del lenguaje es permitir rápidamente a los desarrolladores la generación dinámica de páginas. Con PHP se puede hacer la gran mayoría de las cosas que pueden realizarse con un script CGI, como el procesamiento de información en formularios, foros de discusión, manipulación de cookies y páginas dinámicas. Ofrece la integración con varias bibliotecas externas, que permiten que el desarrollador haga casi cualquier cosa, desde generar documentos en pdf, hasta analizar código XML.

PHP ofrece una solución simple y universal para las paginaciones dinámicas del Web de fácil programación. Su diseño elegante lo hace perceptiblemente más fácil de mantener y ponerse al día en comparación con el código de otros lenguajes. Debido a su amplia distribución PHP está perfectamente soportado por una gran comunidad de desarrolladores. Como producto de código abierto goza de la ayuda de un gran grupo de programadores, permitiendo que los fallos de funcionamiento se encuentren y se reparen rápidamente. Es actualizado continuamente con mejoras y extensiones de lenguaje para ampliar sus capacidades. (HENST 2001). [27]

PHP es uno de los lenguajes de lado servidor más extensibles en la web. Surgido en 1994, se trata de un lenguaje de creación relativamente creciente que ha tenido una gran aceptación en la comunidad debido sobre todo a la potencia y simplicidad que lo caracterizan. Permite insertar sus pequeños fragmentos de código dentro de la página HTML y realizar determinadas acciones de una forma fácil y eficaz sin tener que generar aplicaciones programadas

íntegramente en un lenguaje distinto al HTML. Por otra parte, ofrece un sinnúmero de funciones para la explotación de bases de datos de una manera llana, sin complicaciones.

Podríamos efectuar la quizás odiosa comparación de decir que PHP y ASP son lenguajes parecidos en cuanto a potencia y dificultad si bien su sintaxis puede diferir sensiblemente.

Algunas diferencias principales pueden, no obstante, mencionarse:

- PHP, aunque multiplataforma, ha sido concebido inicialmente para entornos UNIX y es en este sistema operativo donde se pueden aprovechar mejor sus prestaciones. ASP, siendo una tecnología Microsoft, está orientado hacia sistemas Windows, especialmente NT.
- Las tareas fundamentales que puede realizar directamente el lenguaje son definidas en PHP como funciones mientras que ASP invoca más frecuentemente los objetos. Por supuesto, esto no es más que una simple cuestión de forma ya que ambos lenguajes soportan igualmente ambos procedimientos. [9]

#### **Java:**

Java es uno de los pocos lenguajes que permiten desarrollar tanto aplicaciones de ejecución local (comúnmente llamadas 'aplicaciones de escritorio') como aplicaciones que se ejecutan en entornos de red, tanto a nivel de cliente como a nivel de servidor.

Con Java un programador puede realizar todo tipo de aplicaciones: aplicaciones de ejecución local (comúnmente denominadas aplicaciones de escritorio), pequeñas aplicaciones que son ejecutadas en una página web desde un navegador (Applets), aplicaciones cliente/servidor, aplicaciones Web, aplicaciones para dispositivos móviles (teléfonos, PDA's), etc. [10]

Los programas en Java pueden ejecutarse en cualquiera de las siguientes plataformas, sin necesidad de hacer cambios:

- Java siempre chequea los índices al acceder un arreglo.
- Java realiza chequeo de tipos durante la compilación (al igual que C). En una asignación entre punteros el compilador verifica que los tipos sean compatibles.

Java no es un lenguaje para hacer sistemas operativos o administradores de memoria, pero sí es un excelente lenguaje para programar aplicaciones.

Java combina flexibilidad, robustez y legibilidad gracias a una mezcla de chequeo de tipos durante la compilación y durante la ejecución. En Java se pueden tener punteros a objetos de un tipo específico y también se pueden tener punteros a objetos de cualquier tipo. Estos punteros se pueden convertir a punteros de un tipo específico aplicando un cast, en cuyo caso se chequea en tiempo de ejecución de que el objeto sea de un tipo compatible.

## **ASP**

Es la tecnología para la creación de páginas dinámicas del lado del servidor desarrollada por Microsoft. El tipo de servidores que emplean este lenguaje son aquellos que funcionan con sistema operativo de la familia de Windows NT. También pueden visualizarse páginas ASP sobre Windows 95/98.

Se basa fundamentalmente en Visual Basic Script, que deriva del conocido Visual Basic, aunque también se pueden escribir los scripts ASP en otro lenguaje: JScript, que deriva a su vez del conocido JavaScript [11].

ASP.NET es un ambiente de programación construido sobre el entorno NGWS (New Generation Windows Services, o sea, "Servicios de la Nueva Generación de Windows"), que permite crear poderosas aplicaciones de Internet. ASP.NET ofrece varias ventajas importantes sobre los modelos previos de desarrollo para Internet:

### **Mejor eficiencia:**

ASP.NET ejecuta código compilado sobre el entorno NGWS en el servidor. Distinto a sus predecesores interpretados, ASP.NET usa amarres tempranos ("early bindings"), así como compilación justo a tiempo ("just-in-time compilation"), optimización nativa, y servicios de caché, sin configuración adicional. Para los desarrolladores, esto significa eficiencia dramáticamente superior antes de escribir la primera línea de código.

ASP.NET hace fácil el ejecutar tareas comunes, desde el simple envío de un formulario o la autenticación de un cliente, hasta el despliegue y la configuración de una Web. Por ejemplo, el entorno de paginado de ASP.NET le permite construir interfaces de usuario que separan limpiamente la lógica de su aplicación del código de su presentación, y maneja eventos con un modelo sencillo de procesamiento de formularios al estilo de Visual Basic. Adicionalmente, el CLR simplifica el desarrollo con servicios de código gerenciado, como el conteo automático de referencias y la limpieza automática de la memoria utilizada por su aplicación.

### **Gerenciabilidad:**

ASP.NET usa un sistema jerárquico de configuración, basado en archivos de texto, que simplifica la aplicación de parámetros de configuración al servidor y sus aplicaciones. Porque la información de configuración es almacenada como texto, nuevos parámetros pueden ser configurados sin recurrir a herramientas de administración locales. Esta filosofía de "cero administración local" también se extiende al despliegue de aplicaciones de ASP.NET. Una aplicación de ASP.NET se despliega a un servidor simplemente copiando los archivos

necesarios al servidor. No hay que reiniciar el servidor, ni siquiera para reemplazar código compilado que ya está en servicio.

### **Escalabilidad y Disponibilidad**

ASP.NET ha sido diseñado para la escalabilidad con características específicamente dirigidas a mejorar el funcionamiento de servidores en un clúster y de servidores con procesadores múltiples. Los procesos del servidor son vigilados y gerenciados por el entorno del ambiente de ejecución de ASP.NET, así que si algún proceso se entorpece o se detiene, un nuevo proceso puede ser creado para reemplazarlo, lo cual ayuda a mantener la disponibilidad de la aplicación para manejar solicitudes de servicio. [12]

### **1.4.5 Herramientas para el desarrollo:**

#### **Eclipse:**

Eclipse es un IDE que no está orientado específicamente hacia ningún lenguaje de programación en concreto. El uso de un determinado lenguaje, estará supeditado a la existencia de un plugging que le de soporte. Como una plataforma de código abierto que se puede utilizar para diseñar programas de primer nivel, componentes, sitios web y muchos otros elementos.

Una de las características más curiosas del IDE Eclipse es el modo en que se compilan los proyectos. No existe en Eclipse ningún botón que permita compilar individualmente un fichero concreto. La compilación es una tarea que se lanza automáticamente al guardar los cambios realizados en el código. Por esta razón es prácticamente innecesario controlar manualmente la compilación de los proyectos.

Eclipse facilita la tarea de crear y ejecutar pruebas unitarias utilizando la plataforma JUnit Eclipse y mantiene, de forma local, un registro de los cambios realizados en cualquier archivo del proyecto. Es posible comprobar los cambios que se han ido realizando y, en caso de error, volver a versiones anteriores.

IDE EasyEclipse `easyeclipse-php-1.2.2.2` es un plugging de Eclipse para programar en entornos PHP. EasyEclipse es:

- Libre y open-source.
- Multiplataforma.
- Fácil de descargar e instalar.
- Integrado con un plugging de SubVersion (SVN) para el control de versiones.
- Simple de mantener, sin problemas de versiones ni dependencias.

**Zend:**

Zend es un producto de software impulsor de la tecnología PHP, optimizado para el desarrollo de aplicaciones web. El programa, además de servir de editor de texto para páginas PHP, proporciona una serie de ayudas que cubren desde la creación y gestión de proyectos hasta la depuración de código.

Zend está desarrollado con Java, lo que a veces supone que no funcione tan rápido como otras aplicaciones de uso diario. Sin embargo, esto ha permitido a Zend lanzar con relativa facilidad y rapidez versiones del producto para Windows, Linux y MacOS, aunque el desarrollo de las versiones de este último sistema se retrase un poco más.

Zend Studio consta de dos partes en las que se dividen las funcionalidades de parte del cliente y las del servidor. Las dos partes se instalan por separado, la del cliente contiene el interfaz de edición y la ayuda. Permite además hacer depuraciones simples de scripts, aunque para disfrutar de toda la potencia de la herramienta de depuración habrá que disponer de la parte del servidor, que instala Apache y el módulo PHP o, en caso de que estén instalados, los configura para trabajar juntos en depuración.[13]

El editor que permite escribir los scripts es bastante útil para la programación en PHP. La interfaz está compuesta por varias partes, en las que encontramos un explorador de archivos, una ventana de depuración, los menús y otra para mostrar el código de las páginas. Zend Studio es un editor de texto para páginas PHP que proporciona un buen número de ayudas desde la creación y gestión de proyectos hasta la depuración del código. [13]

**Notepad++:**

Las ventajas de este programa son innumerables. Identifica muchísimos tipos de archivo de código fuente (C, C++, Java, Python, XML, JavaScript, PHP...) y resalta la sintaxis correspondiente a cada uno de ellos. Dispone de navegación por pestañas por lo que es posible tener abiertos varios archivos en una sola instancia del programa. Contador de líneas para cada uno de los archivos, posibilidad de selección de líneas y marcadores para estas. Cuando modificamos un archivo ya abierto por Notepad++ por detrás con algún otro programa, Notepad++ lo reconoce y ofrece la opción de recargar el archivo desde el disco.

**PgAdminIII:**

PgAdminIII es una completa herramienta multiplataforma de diseño y administración de Bases de Datos. Se distribuye libremente bajo los términos de la licencia Artistic License [14]. PgAdminIII es la aplicación mas popular, disponible, al igual que PostgreSQL para la mayoría

de Sistemas Operativos. Este está diseñado para responder a todas las necesidades de los usuarios, desde la realización de consultas SQL hasta la elaboración de complejas sentencias. Este hace fácil la administración, además de incluir un editor de SQL, entre otras características.

### **Control de versiones (Subversion (SVN) v1.5.1)**

SVN es un sistema de control de versiones, que mantiene los registros de todos los cambios que se han realizado a los archivos de un software, lo que permite el trabajo de distintos desarrolladores en un mismo proyecto, esta herramienta es muy usada por los programadores de software libre.

### **1.4.6 Proceso Unificado de Desarrollo (RUP)**

El Proceso Unificado de Desarrollo (RUP) es una propuesta de proceso para el desarrollo de software orientado a objetos que utiliza UML para describir un sistema. Es un marco de trabajo genérico que puede especializarse para una gran variedad de sistemas software, para diferentes áreas de aplicación, diferentes tipos de organizaciones, diferentes niveles de aptitud y diferentes tamaños de proyecto. Este proceso tiene como características fundamentales que está dirigido por casos de uso, es iterativo e incremental y centrado en la arquitectura. Está constituido por cinco flujos de trabajo fundamentales: requisitos, análisis, diseño, implementación y prueba, los que tienen lugar sobre 4 fases: inicio, elaboración, construcción y transición.

#### **Características de RUP**

- **Dirigido por Casos de Uso:** El proceso de desarrollo lleva un hilo, avanza a través de una serie de flujos de trabajo que parten de los casos de uso. Los casos de uso se especifican, se diseñan y los casos de uso finales son la fuente a partir de la cual los ingenieros de prueba construyen sus casos de prueba. Los casos de usos guían la arquitectura del sistema.
- **Centrado en la arquitectura:** La arquitectura surge de las necesidades de la empresa, como las perciben los usuarios y los inversores, y se refleja en los casos de uso. Incluye los aspectos más significativos del sistema y se ve influida por muchos otros factores como la plataforma en la que tiene que funcionar el software (arquitectura hardware,

sistema operativo, sistema de gestión de base de datos), las consideraciones de implantación, sistemas heredados y requisitos no funcionales.

- **Iterativo e incremental:** Es práctico dividir el trabajo en partes más pequeñas o miniproyectos. Cada miniproyecto es una iteración que resulta en un incremento. Las iteraciones hacen referencia a pasos en el flujo de trabajo, y los incrementos, al crecimiento del producto. En cada iteración, los desarrolladores identifican y especifican los casos de uso relevantes, crean un diseño utilizando la arquitectura seleccionada como guía, implementan el diseño mediante componentes, y verifican que los componentes satisfacen los casos de uso. Si una iteración cumple con sus objetivos el desarrollo continúa con la siguiente iteración. Cuando una iteración no cumple sus objetivos, los desarrolladores deben revisar sus decisiones previas y probar con un nuevo enfoque. (JACOBSON 2000).[28]

#### ***1.4.7 Lenguaje Unificado de Modelado (UML)***

Es el lenguaje gráfico para modelado de sistemas con tecnología orientada a objeto que permite especificar, visualizar, construir y documentar. Soporta todo el ciclo de vida de desarrollo de software: especificaciones de analistas, arquitectura, diseño, implementación e implantación. Soporta además distintas áreas de aplicación: sistemas distribuidos, tiempo real, aplicaciones monoproceso, sistemas de información corporativos, Banca/Finanzas, Telecomunicaciones, Defensa/Espacio, Transporte, Distribución, Electromedicina, Ciencia, entre otros. Es un lenguaje de modelado visual fácil de aprender pero rico en significado, es estándar, estable y configurable. (MESTRAS 2004).

#### ***1.4.8 SGBD***

Un Sistema de Gestión de Bases de Datos es aquel que se encarga de controlar el acceso concurrente, evitar redundancia, que se cumplan las restricciones y reglas de integridad, usar elementos que aceleren el acceso físico a los datos (índices, agrupamientos, funciones de dispersión), distribuir los bloques del disco del modo más adecuado para el crecimiento y uso de los datos, controlar el acceso y los privilegios de los usuarios y recuperar ante fallos. (PESCADOR 2007). [29]

**PostgreSQL:**

PostgreSQL es un Sistema de Gestión de Bases de Datos Objeto-Relacionales (ORDBMS) que ha sido desarrollado desde 1977. Está considerado como la base de datos de código abierto más avanzada del mundo. PostgreSQL proporciona un gran número de características que normalmente sólo se encontraban en las bases de datos comerciales tales como DB2 u Oracle. Soporta transacciones y desde la versión 7.0, claves ajenas (con comprobaciones de integridad referencial). Tiene mejor soporte para triggers y procedimientos en el servidor. Además, tiene ciertas características orientadas a objetos. (SUAU 2006) [30]. En la actualidad continua un proceso de desarrollo activo a nivel mundial por un equipo de desarrolladores y contribuidores de código abierto. Es multiplataforma. Utiliza una estrategia de almacenamiento de filas para conseguir mejores respuestas en ambientes de grandes volúmenes. Presenta interfaces de comunicación con diferentes lenguajes, incluyendo C++ y PHP. Tiene altos niveles de escalabilidad, más rápido que otros gestores en la ejecución de transacciones. Tiene certificada la implementación ACID de sus transacciones. Presenta buenos niveles de administración de concurrencia de usuarios incluso en transacciones y consultas muy complejas. Se han desarrollado múltiples herramientas para ambientes de clusterización con PostgreSQL. Presenta mejores niveles de rendimiento en entornos Linux.

#### **Oracle:**

Es uno de los gestores de base de datos mas confiables, seguros y difundido en el mundo entero, protege del fracaso del servidor, fracaso del sitio, el error humano, y reduce el tiempo fuera de servicio planeado, afianza los datos y habilita la complacencia con la única seguridad fila-nivelada, interviniendo de grano fino, encriptación de los datos transparente y llamada del total de datos. El almacenaje de los datos alto rendimiento, proceso analítico en línea, y minería de datos [15]. La principal dificultad que lo aparta de la presente investigación es que Oracle no es una herramienta libre.

#### **MySQL**

MySQL es un sistema de Gestión de Bases de Datos Relacional de código abierto.

Las principales ventajas que resaltan de MySQL se refieren a [21]:

- **Costo:** MySQL es gratuito para la mayor parte de los usos y su servicio de asistencia resulta económico.
- **Asistencia:** MySQLAB ofrece contratos de asistencia a precios razonables y existe una nutrida y activa comunidad MySQL.

- **Funcionalidad:** MySQL dispone de muchas de las funciones que exigen los desarrolladores profesionales, como compatibilidad completa con ACID, compatibilidad para la mayor parte de SQL ANSI [19], duplicación, funciones SSL e integración con la mayor parte de los entornos de programación.
- **Portabilidad:** MySQL se ejecuta en la inmensa mayoría de sistemas operativos y, la mayor parte de los casos, los datos se pueden transferir de un sistema a otro sin dificultad.
- **Facilidad de uso:** Resulta fácil de utilizar y de administrar. Las herramientas de MySQL son potentes y flexibles, sin sacrificar su capacidad de uso.

Si se desea sencillez y velocidad, nada es mejor que MySQL. Ampliamente documentada en la red, destaca sobre todo en su utilización sobre plataformas Web donde el número de conexiones es muy elevado, y su tarea principal son consultas sencillas. Si se desea una base de datos que soporte ACID no es una buena elección de momento (tal vez en futuras versiones). [22]

#### **1.4.9 Herramienta CASE**

La utilización de herramientas CASE es muy importante para la creación de aplicaciones de software que requieren un diseño previo.

##### **Algunas ventajas de su utilización:**

- Aumento de la productividad y de la calidad de las aplicaciones desarrolladas. Las herramientas CASE ayudan a los desarrolladores a expresar gráficamente los procesos de negocio y sus correspondientes modelos de datos, validando la integridad y consistencia de los mismos y generando en muchos casos el código necesario para convertir dichos modelos en nuevas aplicaciones.
- Pueden producir rápidamente prototipos de las aplicaciones a desarrollar, los cuales se pueden probar, corregir e incorporar a la aplicación final.
- La tecnología CASE también permite partir los proyectos en múltiples piezas que se pueden trabajar individualmente y que luego pueden ser reensambladas en una única aplicación que funciona perfectamente.
- Suponen una gran ayuda a la hora de gestionar la documentación sistemas de información, ya que proporcionan un entorno donde además de generar buena parte de la misma. (LÓPEZ 2000).[31]

## **Visual Paradigm-UML**

Visual Paradigm para UML es una de las herramientas UML CASE del mercado, considerada como muy completa y fácil de usar, es multiplataforma y proporciona excelentes facilidades de interoperabilidad con otras aplicaciones. Fue creada para el ciclo vital completo del desarrollo del software que lo automatiza y acelera, permitiendo la captura de requisitos, análisis, diseño e implementación. Visual Paradigm-UML también proporciona características tales como generación del código, ingeniería reversa y generación de informes. Tiene la capacidad de crear el esquema de clases a partir de una base de datos y crear la definición de base de datos a partir del esquema de clases. Permite invertir código fuente de programas, archivos ejecutables y binarios en modelos UML al instante, creando de manera simple toda la documentación. Apoya los estándares más recientes de las notaciones de Java y de UML. Incorpora el soporte para trabajo en equipo, que permite que varios desarrolladores trabajen a la vez en el mismo diagrama y vean en tiempo real los cambios hechos por sus compañeros.

## ***1.5 Conclusiones***

En la primera parte de este capítulo se hizo referencia a conceptos relacionados con el objeto de estudio definido en la introducción del documento. En la segunda parte, se desarrolla el estudio de algunas de las tecnologías actuales que pueden ser empleadas durante el proceso de desarrollo del software, mencionando ciertas características y ventajas que las mismas poseen, para la elección de las utilizadas fue tomada siguiendo fundamentalmente la política de uso de herramientas con soporte multiplataforma y licencias de utilización libre.

Las herramientas a utilizar son las siguientes:

- Gestor de Base Datos: PostgreSQL 8.3
- Navegador: Mozilla Firefox 3.0
- Lenguaje de programación del lado del servidor: PHP v5.2.5
- Lenguaje de programación del lado del cliente: JavaScript , Ajax y librería ExtJs-2.2
- Herramienta de desarrollo del lado del servidor: EasyEclipse.
- Servidor Web: Apach2
- Metodología de desarrollo: Proceso Unificado del Software (Rational Unified Process) con notación UML.
- Herramienta CASE de Modelado UML: VisualParadimg para UML 6.1
- Para apoyar el proceso de implementación del software se utilizarán herramientas de desarrollo como Notepad++ y ZendStudio.

# Capítulo 2

## Descripción del sistema.

### ***2.1 Introducción.***

Para comprender lo que es el sistema se realiza una descripción de los detalles que tiene, de las características particulares y generales que lo diferencian de otros. En este capítulo se abordan aspectos relacionados con el objeto de automatización y los documentos específicos que se procesan. Se realiza una descripción general de la propuesta de sistema y cómo debe funcionar. Se especifican los requisitos funcionales y no funcionales así como la definición de los casos de uso. Además de los modelos de análisis y diseño presentando los artefactos obtenidos en las actividades comprendidas en estas etapas del desarrollo

### ***2.2 Problema y Situación problemática.***

EIPAD (Equipo de Investigación Procesamiento y Análisis de Datos) tiene como primer proyecto a Génesis, una plataforma para el procesamiento paralelo de datos y el cálculo. Génesis es una capa superior de abstracción; un conjunto de librerías, aplicaciones servidoras y clientes, bases de datos y componentes distribuidos que constituyen trabajo adelantado a la hora de enfrentar un proyecto de cómputo paralelo. Al acoplarle módulos de cálculo se crea un producto o servicio que responde a una solución paralelizable a un determinado problema. Génesis no cuenta con una interfaz para la validación y transformación de los datos, la presentación de resultados, administración de usuarios, configuración de servicios, reportes de registros del sistema, lo que hace engorrosa la interacción de los usuarios con la plataforma para la ejecución de las solicitudes de cómputo y la administración del sistema.

### ***2.3 Objeto de automatización***

Los procesos que serán objeto de automatización son la administración de usuarios para el uso de la plataforma por usuarios debidamente autorizados, control de las tareas que el usuario posee y puede ejecutar, establecer un entorno colaborativo donde los usuarios puedan

compartir información resultante del proceso de cómputo que desarrollan, establecimiento de permisos y creación de perfiles a usuarios.

### **2.3.1 Propuesta del sistema.**

Después de haber hecho un análisis del problema y la situación problemática y debido a la falta de un sistema que pueda solucionar el mismo se propone desarrollar un sistema que aporte soluciones reales, concretas, eficientes a las situaciones planteadas, el mismo brindará acceso a usuarios que tengan los permisos requeridos para su utilización. También debe brindar facilidades para que el usuario efectúe sus tareas de cómputo y ofrecer información del estado de procesamiento en que se encuentran las mismas. El sistema debe permitir que el usuario pueda compartir información con otros usuarios del sistema. Debe ser fácil y sencillo de utilizar.

### **2.3.2 Requerimientos del sistema.**

A continuación se muestran los requerimientos funcionales y no funcionales que tiene el sistema.

#### **Requisitos funcionales:**

- RF1. Añadir nuevos Usuarios al sistema.
- RF2. Asignar permisos de Usuarios.
- RF3. Denegar permisos de Usuarios.
- RF4. Modificar Datos de Usuarios.
- RF5. Consultar Datos de Usuario.
- RF6. Obtener reporte de registros de información (logs) por nodo.
- RF7. Obtener reporte de registros de información (logs) del sistema.
- RF8. Imprimir reporte de registros de información.
- RF9. Insertar nuevas solicitudes de cálculo.
- RF10. Obtener reportes de las solicitudes de cálculo.
- RF11. Listar mensajes recibidos.

RF12. Enviar mensajes.

RF13. Eliminar mensajes.

**Requisitos no funcionales:**

**Apariencia o interfaz externa:**

- El sistema debe tener una interfaz fácil de usar y amigable para que pueda ser utilizada por usuarios que no necesiten de entrenamiento prolongado.
- Estará diseñado para la resolución deseada por el usuario.
- Empleo de imágenes y colores identificados con el negocio donde se implantará el sistema.

**Usabilidad.**

- El sistema estará disponible las 24 h del día.
- En los controles tipo lista que posean más de 20 posibles selecciones se debe emplear algún mecanismo que permita su rápida localización (búsqueda por teclado).
- Empleo de barras de progreso u otros para indicar el estado del proceso en cuestión.
- Barra de herramientas de acceso directo con las opciones más usadas.
- Aunque la plataforma sugiere el uso para usuarios con conocimientos informáticos, el sistema estará diseñado para usuarios que no necesiten de muchos conocimientos para su interacción.

**Rendimiento.**

- Los tiempos de respuesta y velocidad de procesamiento de la información serán rápidos, no mayores de 20 segundos para las recuperaciones.
- Se deben emplear mecanismos que resuelvan el problema de la carga de datos para ejecución de las solicitudes de cómputo.

**Soporte.**

- Se necesita un servidor de bases de datos que soporte grandes volúmenes de datos (PostgresSql) y ejecución de múltiples transacciones en períodos de tiempo cortos.
- EL sistema va incorporado como un paquete de instalación de la plataforma.

**Portabilidad.**

- El sistema será multiplataforma, podrá ser ejecutado por clientes (Linux o Windows).

**Seguridad.**

- El usuario debe autenticarse para entrar al sistema y realizar cualquier operación.
- Garantizar el acceso controlado a la información. Este debe influir sobre cómo se presentan las interfaces para cada usuario dependiendo del nivel de acceso a la información.

**Software.**

- Se utilizará para el cliente Sistema operativo Windows 98 o superior, o Linux y navegador Mozilla Firefox.
- Se utilizará para el servidor Sistema Operativo Windows Advancer Server (2000 o superior) o Linux en cualquiera de sus distribuciones. Un servidor Apache/2.2.6 (Win32) con módulo PHP 5.2.5 disponible, este debe estar configurado con las extensiones "pgsql", "pdo" y "pdo\_pgsql". Un servidor de bases de datos PostgresSql v8.3.

**Hardware:*****Para el servidor:***

- Requerimientos mínimos: Procesador Pentium IV a 2.0GHz de velocidad de procesamiento y 1Gb de memoria RAM.
- Al menos 40Gb de espacio libre en disco duro.
- Tarjeta de red.

***Para el cliente:***

- Requerimientos mínimos: Procesador Pentium IV a 133Mhz con 128 Mb de memoria RAM.
- Tarjeta de red.

**Restricciones para el diseño e implementación:**

- Utilizar los estándares establecidos de codificación, diseño, entre otros.
- Emplear como servidores web y de bases de datos Apache y PostgresSql.
- Utilizar como lenguaje del lado del servidor al PHP v5.2.5 y del lado del cliente el JavaScript.

## 2.4 Modelo de sistema.

### 2.4.1 Modelo de caso de uso del sistema.

Se presenta a continuación el modelo de caso de uso del sistema obtenido a partir de los requerimientos de la aplicación.

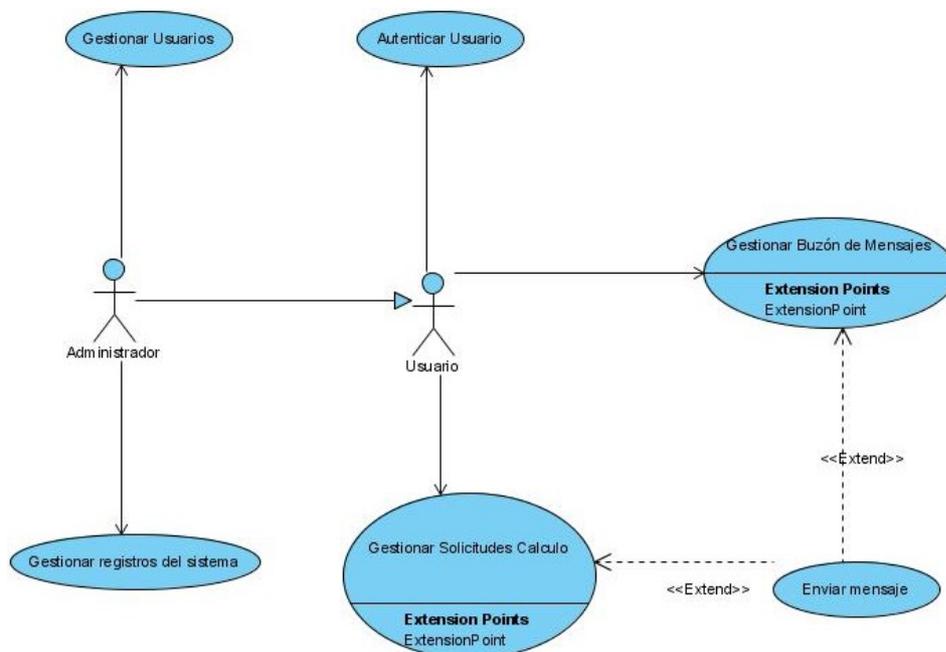


Figura 2.1 Diagrama de casos de uso del sistema.

### 2.4.2 Descripción de los CU del sistema

<b>Caso de Uso:</b>	Gestionar Usuarios
<b>Actores:</b>	Administrador
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador desea introducir un nuevo usuario al sistema o actualizar los datos de uno existente.
<b>Precondiciones:</b>	El administrador debe haberse autenticado en el sistema correctamente.

### Flujo Normal de Eventos

Flujo Normal de Eventos	
	<p>1. El sistema muestra un listado con todos los usuarios, y las opciones de Insertar un nuevo Usuario, Modificar los datos de un usuario existente, y Asignar o Denegar permisos a los usuarios.</p>
<p>2. El administrador selecciona la opción a ejecutar. En caso de:</p> <p>a. Insertar datos de usuario. Sección Insertar datos de Usuario.</p> <p>b. Modificar datos de usuario. Sección Modificar datos de Usuario.</p> <p>c. Asignar permisos al usuario. Sección Asignar permisos a Usuario.</p> <p>d. Modificar permiso de usuario. Sección Modificar permisos de Usuario.</p>	
Sección Insertar nuevo usuario.	
Acción del Actor	Respuesta del Sistema
<p>1. El administrador solicita insertar un nuevo usuario.</p>	<p>2. El sistema muestra un formulario con los campos a completar del usuario.</p>
<p>3. El administrador introduce los datos del usuario y selecciona la opción enviar.</p>	<p>4. El sistema valida los datos de entrada</p> <p>5. El sistema verifica que el usuario no este registrado.</p> <p>6. El sistema guarda en la base de datos los datos del usuario.</p> <p>7. El sistema muestra la opción de actualizar el grid de usuarios.</p>

8. El administrador presiona el botón de actualizar el grid.	9. El sistema muestra el grid actualizado con el usuario nuevo al final del grid.
<b>Flujos Alternos</b>	
	4.1 El sistema muestra el mensaje “Datos de entrada no válidos” y pasa a la acción 2 del flujo normal de los eventos.
	5.1 El sistema muestra el mensaje “El usuario ya existe en el sistema” y pasa a la acción 2 del flujo normal de los eventos.
<b>Sección Modificar Datos del usuario.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema busca los datos del usuario seleccionado y los muestra en un formulario
2. El administrador modifica los datos deseados y selecciona la opción enviar.	3. El sistema valida los datos de entrada y actualiza los datos del usuario.
<b>Poscondiciones</b>	Quedan registrados los datos del usuario.
<b>Sección Asignar permiso al usuario.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema busca los datos del usuario seleccionado y muestra los permisos en un formulario
2. El administrador modifica los datos deseados y selecciona la opción enviar.	3. El sistema valida los datos de entrada y actualiza los datos del usuario.
<b>Poscondiciones</b>	Quedan registrados los datos del usuario.
<b>Sección Modificar permiso al usuario.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema muestra los permisos del usuario seleccionado y los muestra en un formulario

2. El administrador modifica los datos deseados y selecciona la opción enviar.

3. El sistema valida los datos de entrada y actualiza los datos del usuario.

### Interfaz

**Usuarios Registrados**

**Registro de usuarios**

nombre	apellidos	contrasena	usuario
Yadira	Lizama	YadiraLizama	yadir
Hector	Perez	Hector	hperez
Lissuan	Fadraga	Lissuan	lfadr
Luisa	Mue	Luisa	lmue
Ernesto	Figueredo	erensto	emfig
Katrina	Lopez	Kati	katrin

**Permisos**

Nombre permiso	Descripcion
<input type="checkbox"/> permiso2	Acceso al Módulo de tamaño de particu:
<input type="checkbox"/> permiso1	Acceso al Módulo de administracion
<input type="checkbox"/> permiso0	Acceso al Módulo Megamatrices
<input type="checkbox"/> perimiso4	Acceso Básico de Usuarios

**Detalles del usuario**

Fecha: 05/01/2011 Usuario: hperez Foto :  
 Nombre: Hector Entidad: UCI  
 Apellidos: Perez Correo: hperez@sd.cd  
 Contraseña: Hector

Insertar usuario Actualizar Usuarios Borrar formulario Asignar Permisos Suspender permiso Modificar Usuarios

### Poscondiciones

Quedan registrados los datos del usuario.

Cuadro 2.1 Caso de uso Gestionar Usuarios

<b>Caso de Uso:</b>	Autenticar Usuarios
<b>Actores:</b>	Usuarios
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea entrar al sistema.
<b>Precondiciones:</b>	El usuario debe tener comunicación con el sistema.
<b>Flujo Normal de Eventos</b>	
	1. El sistema muestra un formulario para que el usuario se autentifique.

2. El usuario entra los datos de la autenticación y presiona aceptar.

3. El sistema valida los datos de entrada  
4. El sistema verifica que el usuario no este registrado.

5. El sistema le muestra una interfaz al usuario dependientemente a los permisos que tenga con las funcionalidades que puede ejecutar.

### Interfaz

Cuadro 2.2 Caso de Uso Autenticar Usuarios

<b>Caso de Uso:</b>	Generar Registro del Sistema
<b>Actores:</b>	Administrador
<b>Resumen:</b>	El caso de uso se inicia cuando el administrador desea hacer un reporte de los registros del sistema existentes.
<b>Precondiciones:</b>	El administrador deber estar autenticado correctamente.
<b>Flujo Normal de Eventos</b>	
	1. El sistema muestra un grid con el listado de los registros lanzados por el sistema ordenados por fecha.
2. El administrador selecciona la opción a ejecutar. En caso de:	
a. Generar reporte de registros. Sección Generar reporte	

Registros  
 b. Eliminar secuencia de registros. Sección Eliminar Registros.

**Sección Generar Reporte de Registros**

Acción del Actor	Respuesta del Sistema
	1. El sistema muestra los registros en un formulario.
2. El administrador selecciona reporte de registros y presiona aceptar.	3. El sistema crea un pdf con el reporte de los registros.

**Poscondiciones** | Quedan reportados los registros del sistema

**Sección Eliminar Registros.**

Acción del Actor	Respuesta del Sistema
	1. El sistema muestra los registros en un formulario.
2. El administrador selecciona reporte de registros y presiona aceptar.	3. El sistema crea un pdf con el reporte de los registros.

**Interfaz**

**Registros del Sistema**

**Registros almacenados**

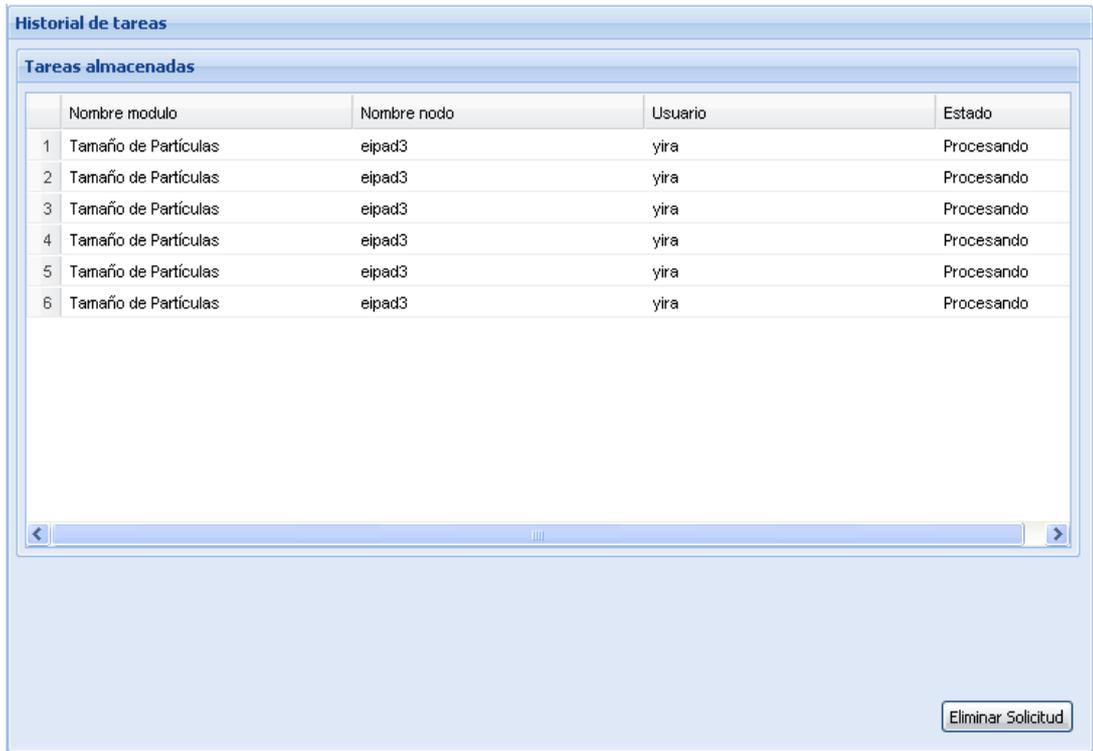
#	Fecha	Mensaje	Descripcion	Nodo
1	Fri Feb 05 1909 00:00:00 GMT-0500	Error en el nodo servidor	Este es un error ,debe reportarlo	1
2	Sat Feb 06 1909 00:00:00 GMT-0500	Se insertaron las solicitudes correctame	Se lanzo para indicar que fueron inserta	1
3	Fri Mar 05 1909 00:00:00 GMT-0500	Conexion correcta	Se creo correctamente la conexion a la l	1

<b>Poscondiciones</b>	Quedan eliminados los registros del sistema que el usuario desee.
-----------------------	---

Cuadro 2.3 Caso de uso Generar Registro Sistema

<b>Caso de Uso:</b>	Gestionar Solicitudes de Calculo
<b>Actores:</b>	Usuarios
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea gestionar las solicitudes de cálculo que tiene realizándose.
<b>Precondiciones:</b>	El usuario debe estar autenticado correctamente y tener los permisos necesarios para ejecutar la acción.
<b>Flujo Normal de Eventos</b>	
	1. El sistema muestra un grid con todas las solicitudes a las cuales el usuario pueda acceder.
2. El usuario selecciona la opción a	

ejecutar. En caso de: a. Eliminar solicitud. Sección Eliminar Solicitud. b. Ver resultado .Sección Ver Resultados	
<b>Sección Eliminar solicitud.</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema muestra las tareas en un grid.
2. El usuario selecciona la tarea y presiona la eliminar.	3. El sistema elimina la tarea de la base de datos y actualiza el grid con las tareas restantes.
<b>Poscondiciones</b>	Quedan eliminadas las tareas que el usuario necesite
<b>Sección Ver Resultados</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema muestra las tareas en un grid.
2. El usuario selecciona la tarea y presiona ver resultados.	3. El sistema muestra los resultados de la solicitud de calculo en otra interfaz o pantalla
<b>Interfaz</b>	



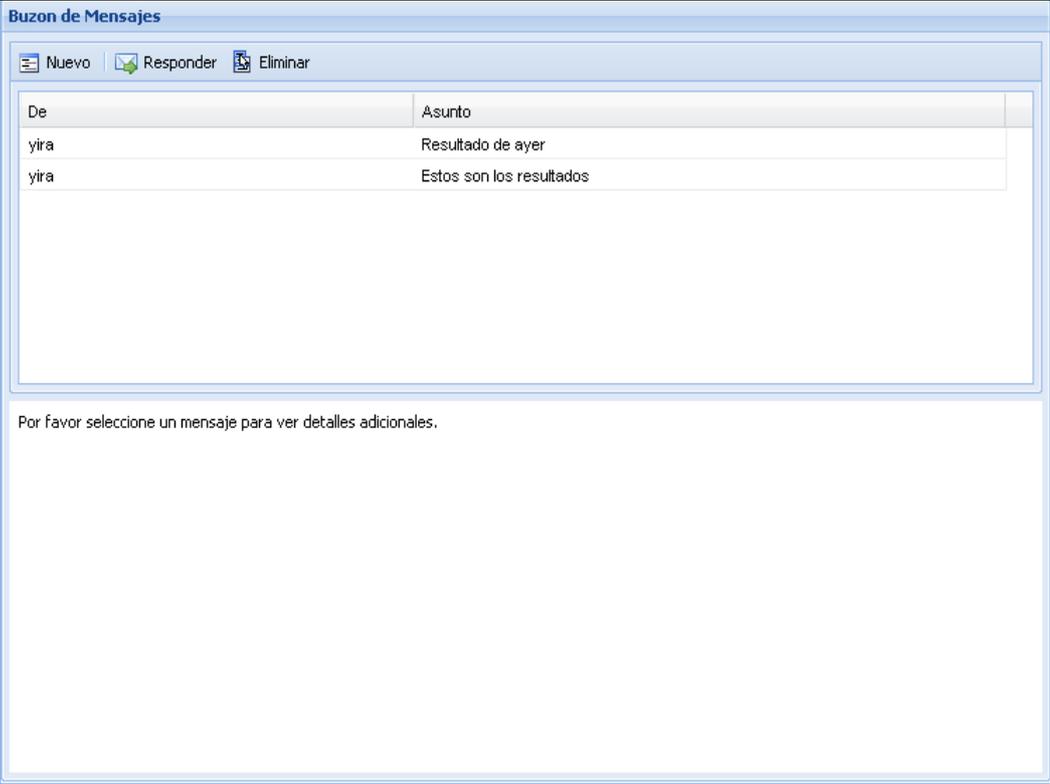
**Poscondiciones**

Quedan mostradas las tareas que el usuario desee ver.

Cuadro 2.4 Caso de uso Gestionar Solicitud Calculo

<b>Caso de Uso:</b>	Gestionar Buzón de Mensajes
<b>Actores:</b>	Usuarios
<b>Resumen:</b>	El caso de uso se inicia cuando el usuario desea gestionar los mensajes que se le han enviado.
<b>Precondiciones:</b>	El usuario debe estar autenticado correctamente y tener los permisos necesarios para ejecutar la acción.
<b>Flujo Normal de Eventos</b>	
	1. El sistema muestra el buzón de mensajes que le han llegado al usuario.
2. El usuario selecciona la opción a ejecutar. En caso de:	
a. Ver mensajes. Sección Ver Mensajes	

b. Responder Mensajes. Sección Responder Mensajes.	
c. Enviar Nuevo Mensaje. Sección Enviar Mensajes.	
d. Eliminar Mensaje. Sección Eliminar Mensaje.	
<b>Sección Mostrar Mensaje</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema muestra los mensajes que tiene el usuario en el buzón de entrada.
2. El usuario selecciona el mensaje.	3. El sistema muestra el cuerpo del mensaje
<b>Poscondiciones</b>	Quedan visualizados los mensajes que el usuario necesite.
<b>Sección Responder Mensaje</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema muestra los mensajes en el buzón de entrada.
2. El usuario selecciona el mensaje.	3. El sistema muestra el mensaje y la opción de responder.
4. El usuario presiona la tecla responder mensaje.	5. El sistema envía el mensaje al usuario deseado.
<b>Poscondiciones</b>	Quedan respondidos los mensajes deseados por el usuario.
<b>Sección Enviar Nuevo Mensaje</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema muestra los mensajes que tiene el usuario en el buzón de entrada.
2. El usuario selecciona el mensaje.	3. El sistema muestra el cuerpo del mensaje, con la opción de enviar nuevo mensaje. 4. El sistema muestra la ventana de

	nuevo mensaje con los datos a completar.
5. El usuario completa las especificaciones de la ventana y presiona la opción de enviar.	
<b>Poscondiciones</b>	Quedan enviados todos mensajes que el usuario necesite.
<b>Sección Eliminar Mensaje</b>	
<b>Acción del Actor</b>	<b>Respuesta del Sistema</b>
	1. El sistema muestra los mensajes que tiene el usuario en el buzón de entrada.
2. El usuario selecciona el mensaje.	3. El sistema muestra el cuerpo del mensaje, con la opción de eliminar el mensaje.
<b>Interfaz</b>	
	
<b>Poscondiciones</b>	Quedan eliminados todos mensajes que el usuario necesite.

Cuadro 2.5 Caso de uso Gestionar Buzón Mensajes

# Capítulo 3

## Análisis y Diseño del sistema.

### ***3.1 Introducción***

En el flujo de trabajo que corresponde al análisis y diseño del sistema se profundiza en los casos de usos detallándolos de manera que permitan reflejar una vista interna del sistema descrita con el lenguaje de los desarrolladores. En esta vista interna se especifican mejor los casos de uso y se determinan las clases necesarias para llevar a cabo las funcionalidades en ellos contenidas. En el presente capítulo se abordarán aspectos relacionados con este flujo de trabajo y se presentarán los artefactos vinculados al mismo.

### ***3.2 Modelo de Análisis***

#### ***3.2.1 Definición del modelo de análisis.***

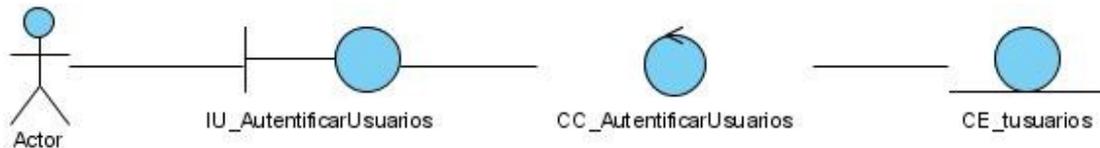
El Modelo de análisis contiene clases de análisis y sus objetos organizados en paquetes que colaboran entre sí.

Las clases de análisis se centran en los requisitos funcionales y son evidentes en el dominio del problema porque representan conceptos y relaciones del dominio. Tienen atributos y entre ellas se establecen relaciones de asociación, agregación/ composición, generalización especialización y tipos asociativos. RUP propone clasificar las clases en:

- Clases de Interfaz: Modelan la interacción entre el sistema y sus actores.
- Clases entidad: Modelan información que posee larga vida y que es a menudo persistente.
- Clases de control: Coordinan la realización de uno o unos pocos casos de uso coordinando las actividades de los objetos que implementan la funcionalidad del caso de uso.

#### ***3.2.2 Diagrama de clases del análisis.***

El objetivo principal del flujo de análisis son los diagramas de clases de análisis, los cuales muestran que clases participan en las realizaciones de los distintos casos de usos. A continuación se muestra el diagrama de clases del análisis para el Caso de Uso Autenticar Usuario. El resto de los diagramas se muestran en el Anexo I.



**Figura 2 Diagrama de clases del análisis caso de uso Autenticar Usuarios.**

Uno de los artefactos que genera el flujo de análisis son los diagramas de interacción del análisis. En el Anexo II se muestran estos diagramas.

### ***3.3 Modelo de Diseño***

#### ***3.3.1 Definición del modelo de diseño.***

Durante el flujo de trabajo de diseño, se modela el sistema de manera que soporte todos los requerimientos, incluyendo a diferencia del análisis, a los requerimientos no funcionales. Este modelo se puede utilizar para visualizar la implementación y para soportar las técnicas de programación gráfica de la aplicación. El modelo de diseño es un modelo de objetos que describe la realización física de los casos de uso centrándose en cómo los requisitos funcionales y no funcionales, junto con otras restricciones relacionadas con el entorno de implementación, tienen impacto en el sistema a considerar. Además, el modelo de diseño sirve de abstracción de la implementación del sistema y es, de ese modo, utilizada como una entrada fundamental de las actividades de implementación. [16]

#### ***3.3.2 Diagramas de clases de diseño.***

A través de este flujo uno de los artefactos más importantes a obtener son los diagramas de clases de diseño, donde se exponen las clases que intervienen en las realizaciones de los casos de uso del sistema.

Para acceder a estos diagramas vea el Anexo III. Para la confección de los diagramas de clases se respetaron los patrones de diseño, se pueden identificar en los diagramas patrones de diseño como el Fachada (Facade) que consiste en proveer una interfaz única a un conjunto de interfaces en un subsistema, o sea define una interface a alto nivel que hace más fácil el uso del subsistema [17]. Además se usa el patrón de Asignación de Responsabilidades Bajo Acoplamiento el cual se encarga de asignar las responsabilidades de modo que se mantenga un bajo acoplamiento, o sea el modo de dar soporte a poca dependencia y a una mayor reutilización. [18]

En los diagramas de clases del diseño elaborados se utilizan los estereotipos para la web que ofrece la herramienta Visual Paradigm con el objetivo de esclarecer el papel que juega cada clase en el diseño.

A continuación se presenta un diagrama de clases del diseño del caso de uso Autenticar Usuarios.

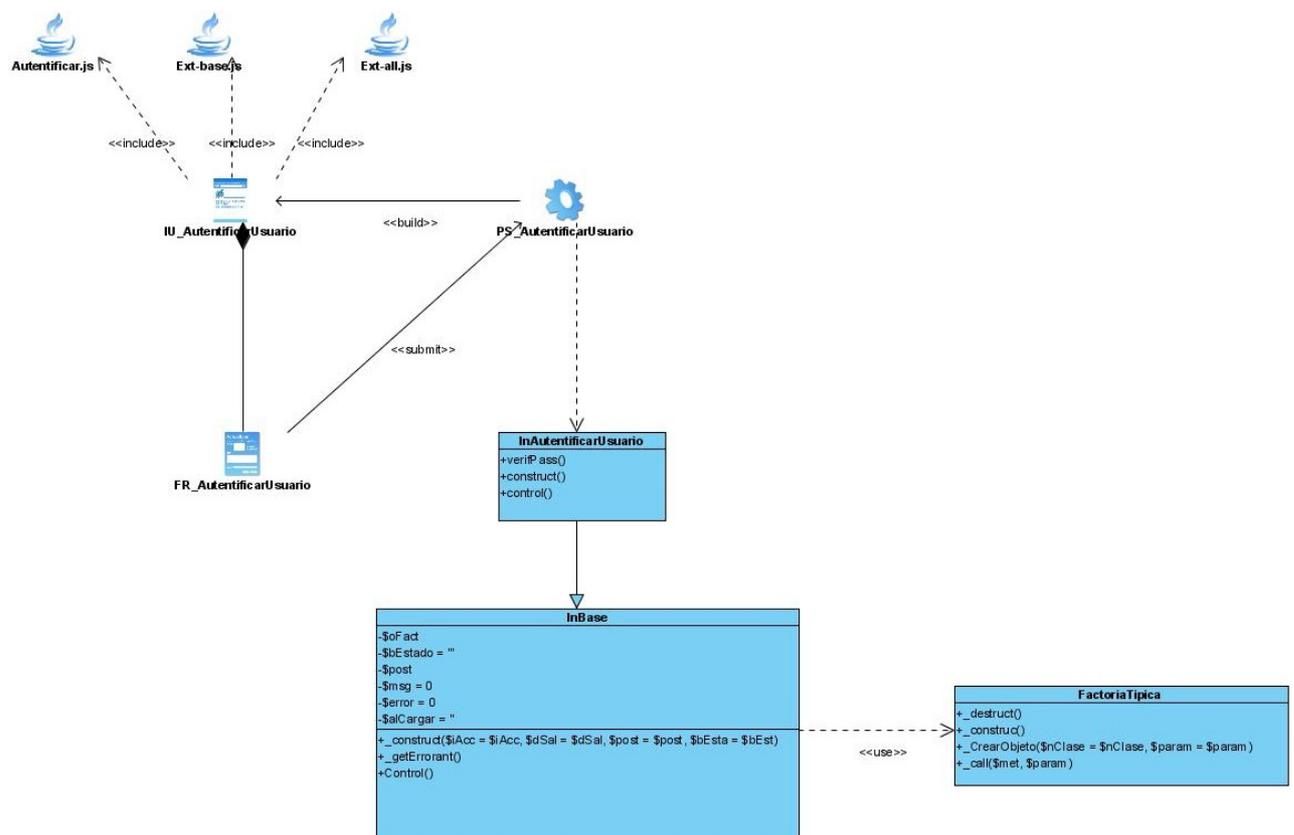


Figura 3 Diagrama de clases del diseño referente al caso de uso Autenticar Usuarios.

### 3.3.3 Diagramas de interacción de clase del diseño:

Los diagramas de interacción de clases del diseño se encuentran disponibles en el Anexo IV.

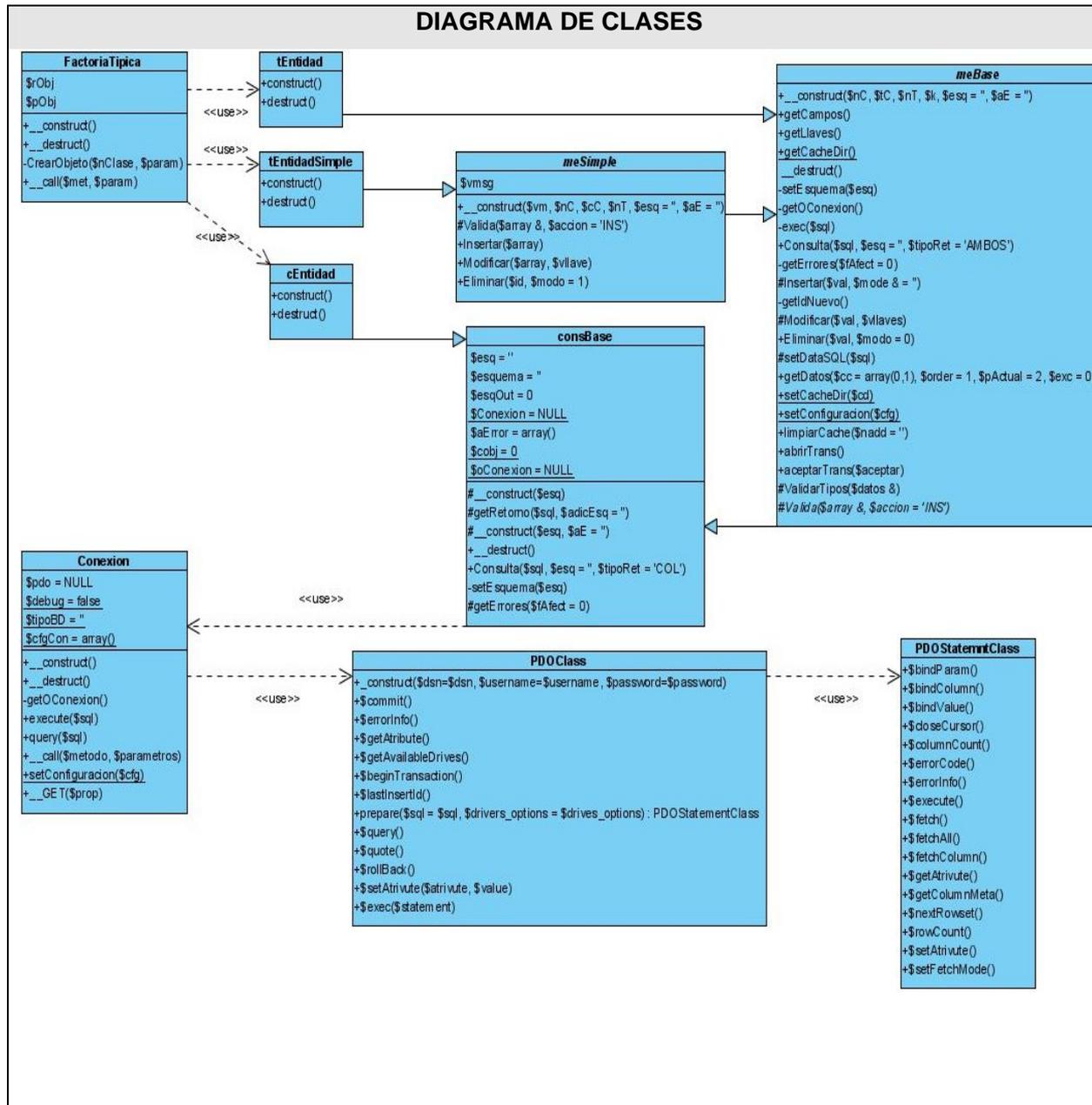


Figura 3 Diagrama de clases del diseño-Mecanismo de Diseño sobre Acceso a Datos

### 3.3.4 Descripción de las clases del diseño.

A continuación se aborda una descripción de las principales clases que deben estar implementadas en el sistema para el correcto funcionamiento de la lógica de negocio y el acceso de datos de la aplicación.

<b>Nombre:</b> Global	
<b>Descripción:</b> Automatiza la carga de las clases. Configura el modelo de acceso a datos y verifica los permisos para hacer uso de un fichero determinado.	
<b>Tipo de clase:</b> public	
<b>Atributo</b>	<b>Tipo</b>
<b>Para cada responsabilidad:</b>	
Nombre:	script runtime ( \$round )
Descripción:	Cálculo del tiempo de ejecución de un script
Nombre	__autoload(\$nomClase)
Descripción:	Función mágica autoload, garantiza la carga automática de las clases al instanciar

Cuadro 3.1 Clase Global

<b>Nombre:</b> Factoría Típica	
<b>Descripción:</b> Clase que implementa la interfaz del modelo de persistencia con el resto de los subsistemas. A través de esta clase se crean y se manipulan los objetos de las típicas simples, los nomencladores y las demás típicas. Es una puerta entre la capa de Acceso a Datos y la capa de Lógica de Negocio.	
<b>Tipo de clase:</b> public	
<b>Atributo</b>	<b>Tipo</b>
\$rObj (Registro de objetos. Arreglo que contiene a los objetos que se hayan creado)	objeto
\$pObj (Chequeo de objetos persistentes. Arreglo que contiene a los objetos persistentes que se hayan creado)	objeto
<b>Para cada responsabilidad:</b>	
Nombre:	__construct()
Descripción:	Constructor de la clase

<b>Nombre</b>	__destruct()
<b>Descripción:</b>	Destructor de la clase
<b>Nombre</b>	CrearObjeto(\$nClase , \$param)
<b>Descripción:</b>	Se devuelve un objeto ya creado, en caso contrario se crea, se guarda en el registro de objetos y se devuelve. Es llamado únicamente desde el método __call.
<b>Nombre</b>	__call(\$met, \$param)
<b>Descripción:</b>	Función mágica que se ejecuta cuando se hace referencia a un método que no existe en la clase. Permite de forma transparente la interacción con el resto de las clases del modelo de persistencias. Implementa de forma explícita tres métodos base: Crear, pCrear, Destruir.

**Cuadro 3.2 Clase Factoría Típica**

<b>Nombre:</b> InBase	
<b>Descripción:</b> Se encarga de darle salida a los mensajes de alerta o error registrados.	
<b>Tipo de clase:</b> abstract	
<b>Atributo</b>	<b>Tipo</b>
\$dSalidaCU	string
\$oFact	objeto
\$bEstado	string
\$post	array
\$msg	array
\$error	array
\$alCargar	string
\$iAccion	string
<b>Para cada responsabilidad:</b>	
<b>Nombre:</b>	__construct(\$iAcc, \$dSal, \$post , \$bEst )
<b>Descripción:</b>	Constructor de la clase.
<b>Nombre</b>	getErrores()
<b>Descripción:</b>	Permite dar salida a los mensajes de alerta o error registrados anteriormente en la clase.
<b>Nombre</b>	getDatosRet(& \$datos, \$idp)
<b>Descripción:</b>	Incorpora a \$datos chequeo errores

<b>Nombre</b>	vRetorno(& \$sess)
<b>Descripción:</b>	Verifica si se desea salir del CU
<b>Nombre</b>	getXML(& \$datos)
<b>Descripción:</b>	devuelve los datos en un XML
<b>Nombre</b>	movActualizacion(& \$sess, \$Elim)
<b>Descripción:</b>	Resuelve el tema del movimiento por un navegador.

Cuadro 3.3 Clase InBase

### 3.3.5 Diseño de la Base de Datos

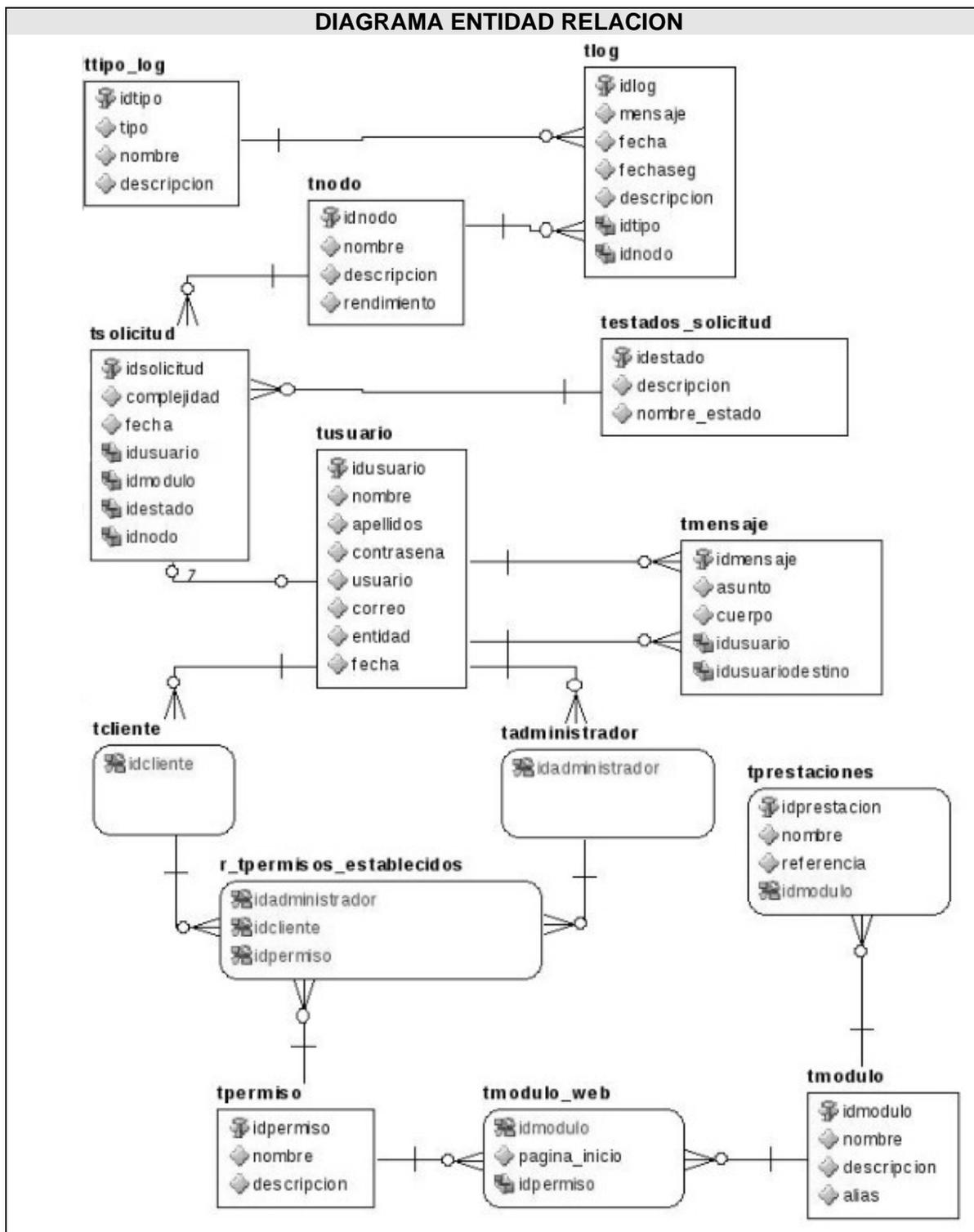


Figura 3.1 Diagrama Entidad-Relación del sistema.

### **3.3.6 Descripción de las tablas.**

La descripción de las tablas que intervienen en el Modelo de Datos del Sistema se describe en el Anexo V.

## **3.4 Definiciones de diseño que se aplican**

### **Estándares en la interfaz de la aplicación**

La apariencia del sistema está definida por el diseño visual que se realiza, sumándole originalidad al software, y es de gran importancia para lograr que el usuario se sienta satisfecho con la información que obtiene y con la forma en que lo hace. La interfaz de la aplicación debe ser fácil de usar para quienes trabajan con ella de manera que el usuario pueda sentirse cómodo al intercambiar con el sistema que es un producto para él.

- **Menú**

El menú principal de la aplicación será vertical, localizado a la izquierda de la región de la página. La denominación de las opciones principales o visibles debe referirse o encapsular a un grupo de acciones afines y no a una acción en particular. Cada módulo de cómputo es un nodo del árbol del que se despliegan sus prestaciones.

- **Diseño de Página**

Las páginas interfaces del sistema debe llevar una cabecera que lo identificará, imagen no muy grande representativa del sitio, barra de menú con sus opciones, área de trabajo, área de pie o barra de estado, además se debe dar información sobre el usuario que está trabajando en la aplicación como nombre de usuario y tipo de acceso.

- **Hojas de Estilo**

Se utilizarán colores basados sobre tonalidades claras fundamentalmente en azul combinado con el blanco. Las letras serán en color negro, y las señalizaciones de error en color rojo.

## **3.5 Tratamiento de errores**

Para el tratamiento de errores se utilizó la técnica de validación de los datos introducidos por los usuarios en el sistema según la arquitectura del mismo. Tendremos por tanto dos tipos de validaciones para los tipos de errores existentes:

1. Validación en el cliente

En este caso las validaciones son a través de código Java Script, con el objetivo de evitar realizar viajes innecesarios al servidor. Identificar los errores directamente en el cliente permite advertir de la existencia de estos sin la necesidad de esperar a que el servidor realice dichas validaciones.

2. Validación en el servidor

En el servidor las validaciones tienen una gran importancia para los casos en que el cliente, por ejemplo, tenga deshabilitada la ejecución de script del lado del cliente y con esto la validación del lado del cliente no sería ejecutada. En este caso las funciones de validación en el servidor se encargan de impedir que se procese información no correcta si omitimos este paso.

De manera general en ambos casos la validación incluye errores en los tipos de datos esperados así como la omisión de algunos de los mismos. En cualquiera de los casos, se informa a los usuarios por medio de mensajes lo más claros posibles para que los mismos conozcan y solucionen el problema.

### ***3.6 Seguridad***

Para garantizar el acceso de los usuarios autorizados al sistema, se implementa un subsistema de seguridad, acoplado al sistema. Este subsistema de seguridad se encuentra en una fase inicial de desarrollo, cuenta con una clase Perfil.php que dada un usuario y su categoría construye un archivo de extensión **.json**, utilizado por la clase Portada.js para mostrar las opciones a las que tiene acceso el nivel del usuario autenticado. Cuando el usuario registra sus credenciales en la página principal, son enviados a la clase InAutenticar.php que se encarga de consultar la base de datos para comprobar la veracidad de los datos y en caso que el registro sea correcto se carga el perfil del usuario.

### ***3.7 Interfaz***

Mediante el uso del lenguaje de programación del lado del servidor PHP y la tecnología AJAX se logró desarrollar una interfaz cómoda para el usuario creando una experiencia interactiva que no requiere recargar la página cada vez que el mismo realiza una petición al servidor, la misma es además, sencilla, con tonalidades de colores claros predominando el azul claro y fácil de usar. Los prototipos de interfaz podrán verse en las descripciones del Anexo I.

### ***3.8 Conclusiones.***

En este capítulo se trataron aspectos relacionados con el flujo de trabajo análisis y diseño obteniendo los artefactos correspondientes al mismo, además de puntos importantes que tienen que ver con la seguridad, , interfaz, estándares en la interfaz de la aplicación y tratamiento de errores que nos ayudan a definir cómo quedará el sistema propuesto.

# Capítulo 4

## Implementación y prueba.

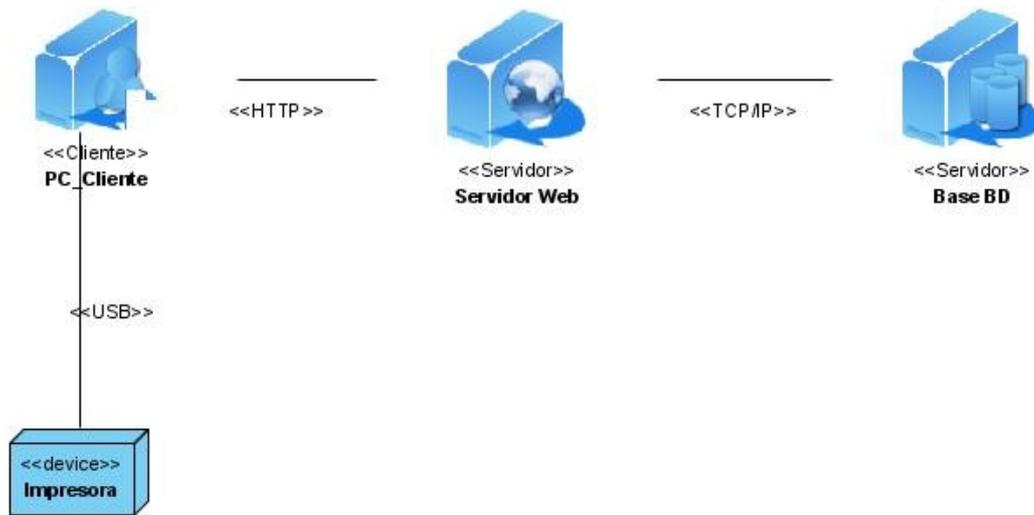
### ***4.1 Introducción***

El resultado principal de la implementación es el Modelo de Implementación. En este flujo de trabajo se empieza con el resultado del diseño y se implementa el sistema en términos de componentes, es decir, ficheros de código fuente, de código binario, ejecutables, scripts, y similares y cómo estos se organizan de acuerdo a los nodos específicos en el modelo de despliegue. El flujo de implementación está fuertemente determinado por el lenguaje de programación. En este capítulo se muestran los diagramas componentes relacionados con la aplicación, los casos de prueba a los que fue sometida la aplicación, y el diagrama de despliegue.

### ***4.2 Implementación***

El modelo de implementación incluye el diagrama de despliegue que define la configuración de los nodos de procesamiento y las correspondencias entre ellos en los que funcionará la aplicación, véase figura 4.1, los casos de prueba para validar los casos de uso y el diagrama de componentes a construir y las relaciones entre los mismos el cual se explicará a continuación. Para realizar este diagrama primeramente se hizo uno general que mostrara la distribución lógica de los componentes por capas representando las tres capas de la arquitectura definida y la fuente de datos donde están almacenados los mismos. Vea figura 4.2.

Los restantes diagramas de componentes se pueden acceder a ellos en el anexo V.



**Figura 4.1 Diagrama de despliegue Aplicación Interfaz de Génesis**

El usuario accede a la aplicación ubicada en el servidor web desde la computadora cliente a través del protocolo <<HTTP>> .La base de datos se modela en un único servidor que representa la dirección IP a la que se conecta el servidor web para realizar las transacciones. En la mayoría de los casos la base de datos estará clusterizada y la cantidad de nodos varía en dependencia de la carga de procesamiento actual del sistema, por lo que no se modela la cantidad de nodos del clúster de base de datos. La impresora se recomienda para imprimir los reportes del sistema.

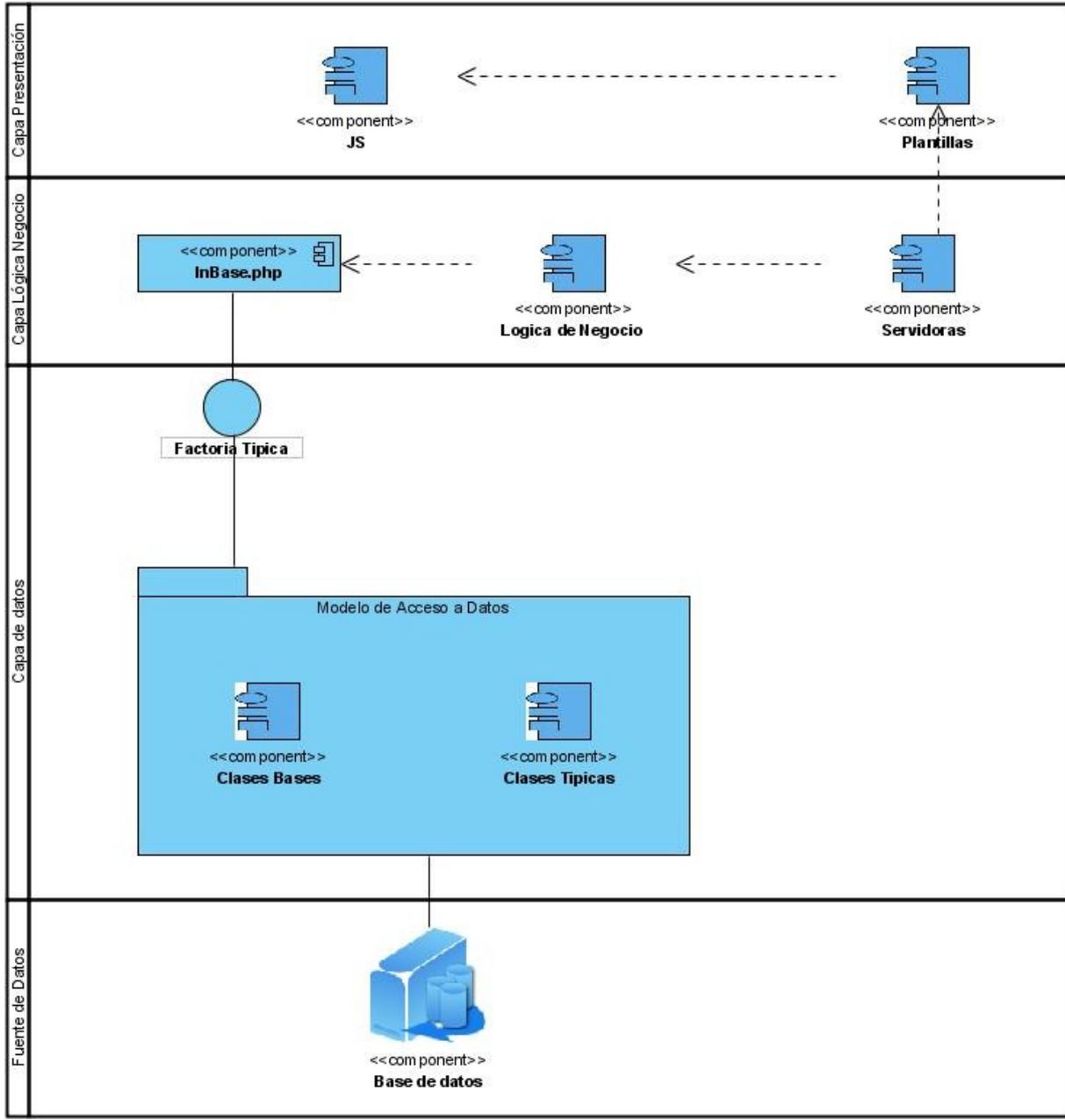


Figura 4.2 Diagrama de componentes general por capas.

### 3.3 Modelo de prueba

Una de las últimas fases del ciclo de vida antes de entregar un programa para su explotación, es el flujo de trabajo de pruebas. Esta es una actividad en la cuál un sistema o componente es ejecutado bajo unas condiciones o requerimientos especificados, los resultados son observados y registrados, haciendo finalmente una evaluación de algún aspecto del sistema o componente.

El método de prueba aplicado fue el de caja negra el cual se refiere a las pruebas que se llevan a cabo sobre la interfaz del software. Los casos de prueba pretenden demostrar que las funciones del software son operativas, que la entrada se acepta de forma adecuada y que se produce un resultado correcto, así como que la integridad de la información externa se mantiene. Este tipo de prueba permite obtener un conjunto de condiciones de entrada que ejerciten de forma completa todos los requisitos funcionales de un programa, es un enfoque complementario que intenta descubrir diferentes tipos de errores como por ejemplo errores de interfaz, errores en estructuras de datos o acceso a las bases de datos entre otros.

**CP1: Nombre del CU** AutenticarUsuarios.

**CP1.1:** Autenticar Usuario.

**Descripción general:** El usuario queda autenticado.

Entrada	Resultados	Condiciones
El usuario entra el usuario y la contraseña y presiona aceptar.	El usuario queda autenticado y el sistema le muestra su perfil dependiendo de los permisos que tiene.	El usuario y contraseña que entra son correctos.
El usuario entra el usuario y la contraseña y presiona aceptar.	El sistema muestra un mensaje en la interfaz indicándole que el usuario o la contraseña no corresponden con el correcto.	El usuario o la contraseña no coinciden con los correctos.
El usuario no completa los campos correspondientes y presiona aceptar.	El sistema muestra un mensaje de error solicitando el usuario o la contraseña para entrar al sistema.	El usuario no debe haber introducido el usuario o la contraseña dejando en blanco algún espacio o ambos del

		formulario.
--	--	-------------

**Tabla 1 Caso de Prueba para Autenticar Usuarios**

**CP2: Nombre del CU GestionarUsuarios.**

**CP2.1: Insertar Usuario.**

**Descripción general:** Gestión de los usuarios y los permisos del sistema.

<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El administrador entra todos los datos del formulario y presiona enviar.	El usuario nuevo queda añadido y sus datos guardados en la base de datos.	Los datos de entrada del formulario son correctos.
El usuario entra todos los datos del formulario y presiona aceptar	El sistema muestra un mensaje en la interfaz indicándole que la entrada en algún campo esta incorrecta.	Los datos tienen errores de formato o de validación.
El usuario no completa los campos correspondientes y presiona aceptar.	El sistema muestra un mensaje de error solicitando al usuario que llene los campos del formulario.	El usuario no debe haber introducido algún campo dejando en blanco algún espacio del formulario.

**Tabla 2 Caso de Prueba para Insertar Usuarios**

**CP2: Nombre del CU GestionarUsuarios.**

**CP2.2: Modificar Usuario.**

**Descripción general:** Gestión de los usuarios y los permisos del sistema.

<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El administrador entra todos los datos del formulario y presiona enviar.	El sistema muestra un mensaje en la interfaz indicándole que no ha introducido ningún cambio o modificación en el usuario forzándolo a hacerla o a dejar la	Los datos de entrada del formulario son los mismos del que antes existía y no existe modificación alguna.

	interfaz de modificación.	
El usuario entra todos los datos del formulario y presiona aceptar	El sistema muestra un mensaje en la interfaz indicándole que la entrada en algún campo esta incorrecta.	Los datos tienen errores de formato o de validación.
El usuario no completa los campos correspondientes y presiona aceptar.	El sistema muestra un mensaje de error solicitando al usuario que llene los campos del formulario.	El usuario no debe haber introducido algún campo dejando en blanco algún espacio del formulario.

Tabla 2 Caso de Prueba para Modificar Usuario

**CP2: Nombre del CU GestionarUsuarios.**

**CP2.2: Modificar Permisos.**

**Descripción general:** Gestión de los usuarios y los permisos del sistema.

Entrada	Resultados	Condiciones
El administrador selecciona el usuario y los permisos a modificar y presiona aceptar.	El sistema muestra confirmación de la asignación de los permisos	Los permisos que selecciona el administrador están correctos.
El administrador selecciona el usuario y presiona aceptar.	El sistema muestra un mensaje en la interfaz indicándole que los permisos seleccionados ya el usuario los tiene asignados.	Los permisos son los mismos que tiene el usuario ya asignado.

Tabla 2 Caso de Prueba para Modificar Usuario

**3.4 Conclusiones.**

En este capítulo se presentaron artefactos como el diagrama de despliegue, el diagrama de componentes y el diseño de los casos de prueba correspondientes al flujo de implementación obteniendo como resultado el modelo del mismo teniendo en cuenta el lenguaje de programación utilizado.

# Conclusiones.

Al concluir el presente trabajo podemos afirmar que se hizo un análisis de las principales tecnologías para diseñar e implementar un sistema informático capaz de administrar y hacer más fácil la interacción de los usuarios con la plataforma Génesis. El diseño y la implementación de este sistema se logró sobre la base de la elección de las herramientas elegidas por la investigación realizada en capítulos anteriores. Los objetivos propuestos se cumplieron a partir de los indicadores:

1. Se identificaron los requerimientos funcionales y no funcionales en los procedimientos establecidos para la utilización de la plataforma Génesis para la modelación del sistema.
2. Se realizó el análisis y diseño de la aplicación informática de administración y monitorización acoplada a la plataforma Génesis.
3. Se logró la implementación del diseño elaborado alcanzando un sistema para administración de usuarios, que favorece la colaboración entre los clientes a partir del intercambio de resultados, con reportes de registros de información del sistema para las actividades de monitorización.
4. Se realizó el proceso de pruebas de integración a los componentes del sistema para la validación del sistema.

# Recomendaciones.

A partir de las investigaciones realizadas se recomienda seguir desarrollando la aplicación para que cubra los requerimientos posteriores de la plataforma Génesis como son la configuración de los nodos del clúster, configuración de servicios de comunicación entre nodos. Continuar el proceso de mejora del entorno colaborativo para obtener una mayor aceptación por parte de los usuarios y en el módulo de seguridad del sistema. Añadir herramientas para la monitorización del rendimiento de los nodos y proponer una solución para configuración de servicios genéricos.

# Referencias Bibliográficas:

[1] Redacción Alvanova.com .La navegabilidad de la web .2007 .Disponible en:  
[http://www.creaciondempresas.com/serv\\_gratuitos/albanova/disenio/art1.asp](http://www.creaciondempresas.com/serv_gratuitos/albanova/disenio/art1.asp). Consultado en [febrero 2009].

[2] Computación Grid. 2007 .Disponible en:  
<http://www.textoscientificos.com/redes/computacion-grid>. Consultado en [febrero 2009]

[3]The Globus Toolkit.2007. Disponible en URL: [http://www.globus.org/grid\\_software/](http://www.globus.org/grid_software/). Consultado en [febrero 2009].

[4]I. Olea. Introducción a PVM. 2004 .Disponible en:  
<http://www.ibiblio.org/pub/Linux/docs/LuCaS/Manuales-LuCAS/doc-cluster-computadoras/doc-cluster-computadoras-html/node43.html>. Consultado en [febrero 2009].

[5]I. Olea. Introducción a XPVM. 2004. Disponible en:  
<http://www.ibiblio.org/pub/Linux/docs/LuCaS/Manuales-LuCAS/doc-cluster-computadoras/doc-cluster-computadoras-html/node48.html>. Consultado en [febrero 2009].

[6] Una Introducción a Apache. 2006. Disponible en:  
[http://linux.ciberaula.com/articulo/linux\\_apache\\_intro/](http://linux.ciberaula.com/articulo/linux_apache_intro/). Consultado en [febrero 2009].

[7]Qué es Ajax. 2006. Disponible en:  
<http://www.webexperto.com/articulos/articulo.php?cod=223>. Consultado en [febrero 2009].

[8]S. Frederick. Learning ExtJs: Disponible en: <http://extjs.com/learn/Tutorials>. Consultado en [febrero 2009].

[9]M. A. Alvarez. Qué es PHP. 2001. Disponible en  
<http://www.desarrolloweb.com/articulos/392.php>. Consultado en [febrero 2009].

[10] M. Enjolras. Beneficio del uso de Java en las aplicaciones modernas de bibliotecas. 1998. Disponible en: [http://fesabid98.florida-uni.es/Comunicaciones/m\\_enjolras.htm](http://fesabid98.florida-uni.es/Comunicaciones/m_enjolras.htm). Consultado en [febrero 2009].

[11] El lenguaje Java. 2007. Disponible en <http://www.dcc.uchile.cl/~lmateu/Java/Apuntes/java.htm>. Consultado en [febrero 2009]

[12] M. A. Alvarez. Que es ASP. 2001. Disponible en: <http://www.desarrolloweb.com/articulos/393.php>. Consultado en [febrero 2009]

[13] M. A. Alvarez. Editor web orientado a la programación de páginas PHP. 2003. Disponible en: <http://www.desarrolloweb.com/articulos/1178.php>. Consultado en: [febrero 2009].

[14] Artistic Licenses. 2008. Disponible en: <http://www.opensource.org/licenses/artistic-license.php>. Consultado en [febrero 2009].

[15] Oracle Database. Disponible en: [http://www.oracle.com/database/standard\\_edition.html](http://www.oracle.com/database/standard_edition.html). Consultado en [febrero 2009].

[16] Conferencia 6 Flujo de Analisis y Diseño. *teleformacion.cu*. [En línea] 2007.] [http://teleformacion.uci.cu/file.php/42/Clases\\_Curso\\_2007-2008/conferencias/Conferencia\\_6/Conferencia\\_6.pdf](http://teleformacion.uci.cu/file.php/42/Clases_Curso_2007-2008/conferencias/Conferencia_6/Conferencia_6.pdf). Consultado en [febrero 2009].

[17] **Gamma, Erich, y otros.** *Design Patterns Elements of Reuseable Object-Oriented Software*. s.l.: Addison Wesley, 1994. Consultado en [febrero 2009].

[18] **Larman, Craig.** *UML y Patrones. Introducción al análisis y diseño orientado a objetos*. México: Prentice Hall, 1999. 970-17-0261-1. Consultado en [febrero 2009].

[19] **Netcraft.** "April 2009 Web Server Survey". 2009 Disponible en: [http://news.netcraft.com/archives/2009/04/06/april\\_2009\\_web\\_server\\_survey.html](http://news.netcraft.com/archives/2009/04/06/april_2009_web_server_survey.html). [Consultado en: Mayo 2009]

[20] Mozilla Foundation. Características de Firefox. 2009. Disponible en <http://www.mozilla-europe.org/es/firefox/features/#tabs> [Consultado en Mayo 2009]

[21] D. J. Burbano. Análisis comparativo de bases de datos de código abierto Vs código cerrado (Determinación de índices de comparación). Consultado en [febrero 2009].

[22] Grupo Linux. Comparativa de distintos SGBQ bajo Linux. 2003. Disponible en: [www.gvpontis.gva.es/fileadmin/conselleria/images/Documentacion/migracionSwAbierto/Anexos/Anexo\\_C17.pdf](http://www.gvpontis.gva.es/fileadmin/conselleria/images/Documentacion/migracionSwAbierto/Anexos/Anexo_C17.pdf) [Consultado en Marzo 2009]

[23] J. G. Gutiérrez. Definición arquitectura cliente servidor, 2005. Disponible en: <http://www.monografias.com/trabajos24/arquitectura-cliente-servidor/arquitectura-clienteservidor.Shtml>. Consultado en [febrero 2009].

[24] (WILKINSONPC 1998-2007). *Navegador Firefox: Mucho mejor que IE*, 1998-2007. [Disponible en: <http://www.wilkinsonpc.com.co/free/firefox.html>]. Consultado en [febrero 2009].

[25] (ALVAREZ 2003). ALVAREZ, M. A. *Qué es Java Script?*, 2003. [Disponible en: <http://www.desarrolloweb.com/articulos/25.php>]. Consultado en [febrero 2009].

[26] ((W3C) 1994-2005). L. O. D. W. W. W. C. *Guía Breve de Tecnologías XML*, 1994-2005. [Disponible en: <http://www.w3c.es/divulgacion/guiasbreves/TecnologiasXML>]. Consultado en [febrero 2009].

[27] (HENST 2001). C. V. D. *¿Qué es el PHP?*, 2001. [Disponible en: <http://www.maestrosdelweb.com/editorial/phpintro/>]. Consultado en [febrero 2009].

[28] (JACOBSON 2000). *El Proceso Unificado de Desarrollo de Software*, 2000. [Disponible en: LÓPEZ, D. H. Y. F. R. *Ventajas de las Herramientas CASE*, 2000. [Disponible en: <http://www.csi.map.es/csi/silice/Sgcase5.html>]]. Consultado en [febrero 2009].

[29] (PESCADOR 2007). *Manual: ¿Qué es una Base de Datos?*, 2007. [Disponible en: [http://www.mihosting.net/faqmysql/que\\_es\\_base\\_de\\_datos.shtm](http://www.mihosting.net/faqmysql/que_es_base_de_datos.shtm)]. Consultado en [febrero 2009].

[30] (SUAU 2006). *MySQL vs. PostgreSQL*, 2006. [Disponible en: <http://www.fedora-es.com/node/189>]. Consultado en [febrero 2009].

[31] (LÓPEZ 2000). *Ventajas de las Herramientas CASE*, 2000. [Disponible en: <http://www.csi.map.es/csi/silice/Sgcase5.html>]. Consultado en [febrero 2009].

# Anexos

## Anexo I: Diagramas de clases del análisis

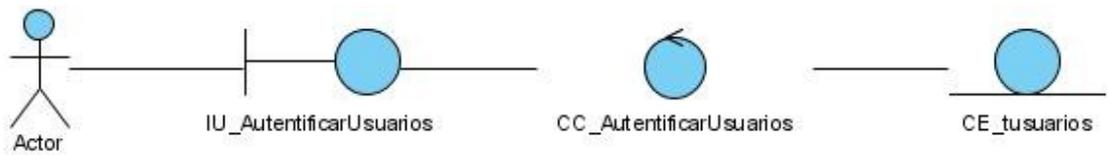


Figura 1 Diagrama de clases del análisis del caso de uso AutenticarUsuarios

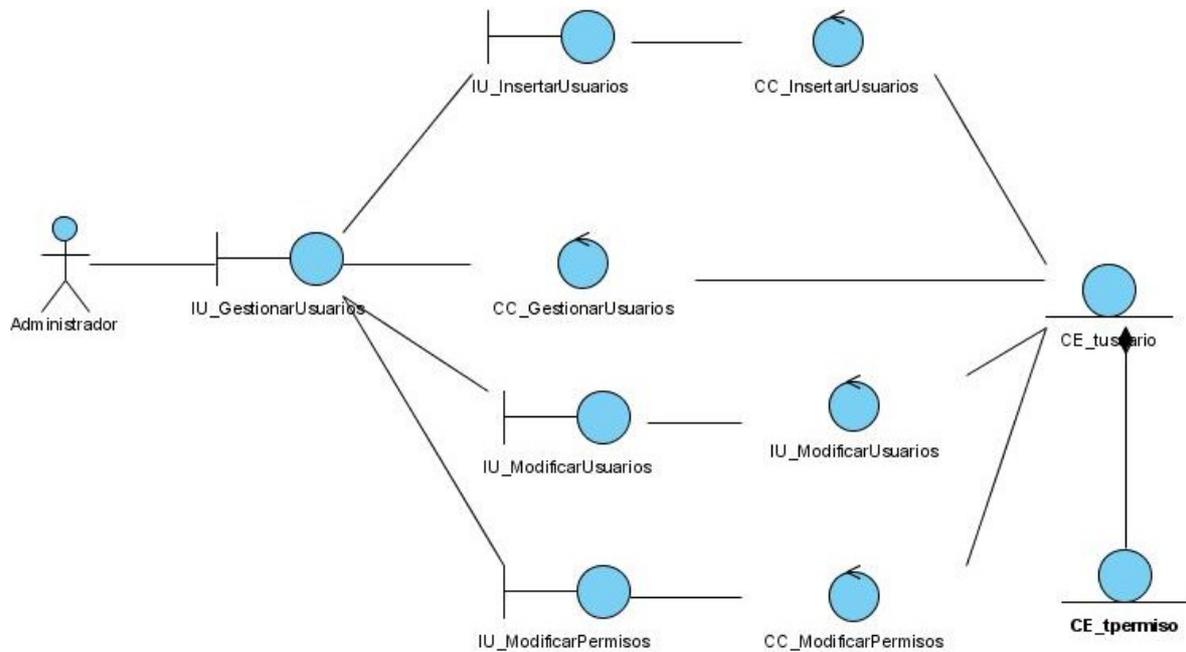
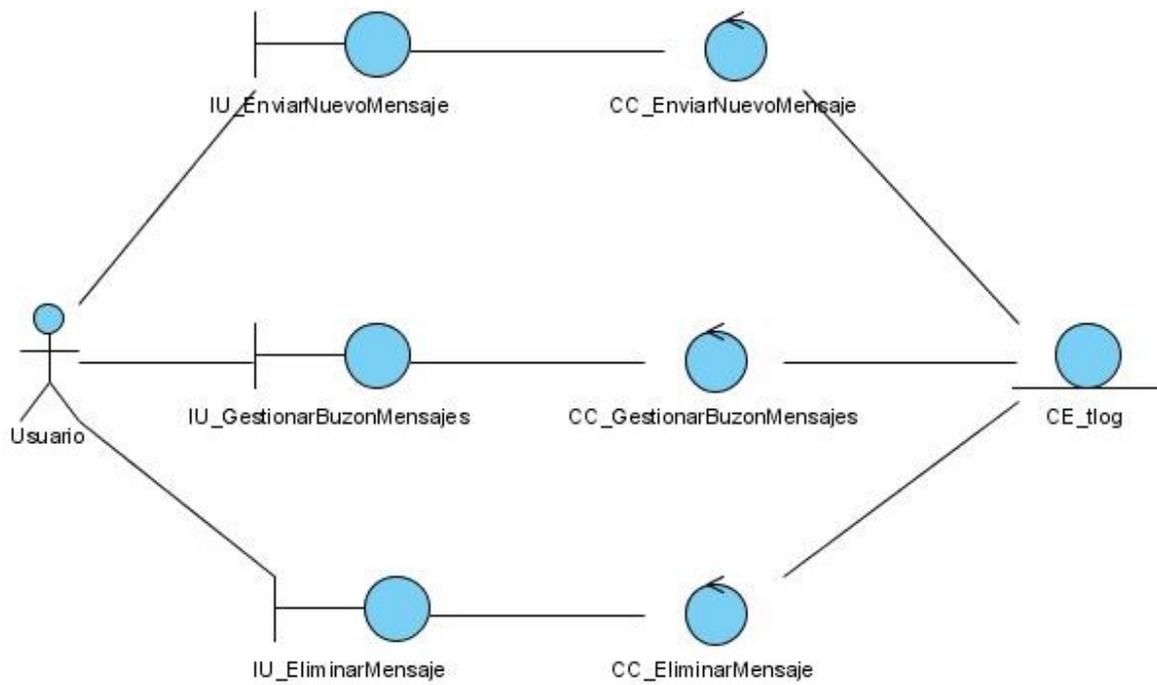
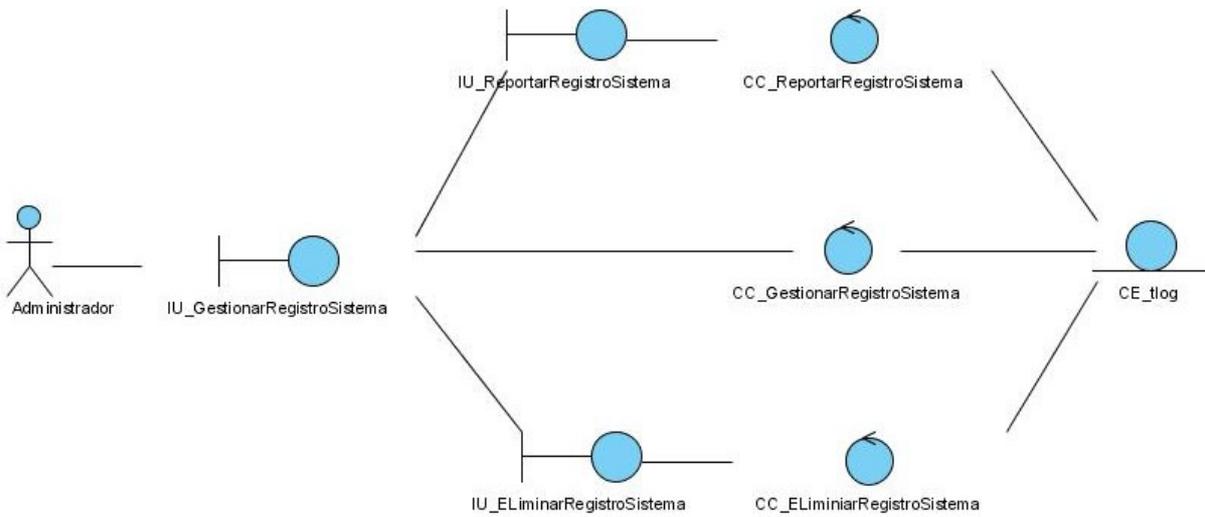


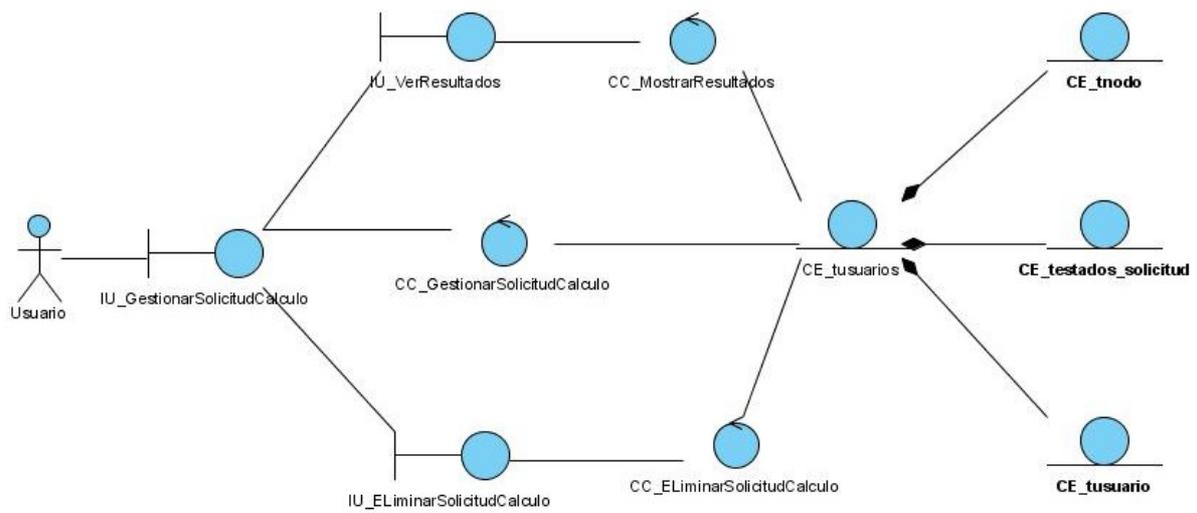
Figura 2 Diagrama de clases del análisis del caso de uso GestionarUsuarios



**Figura 3. 0.1 Diagrama de clases de análisis del caso de uso GestionarBuzonMensajes**



**Figura 4 Diagrama de clases del análisis del caso de uso GestionarRegistroSistema**



**Figura 5 Diagrama de clases del análisis del caso de uso GestionarSolicitudCalculo**

## Anexo II: Diagramas de interacción del análisis.

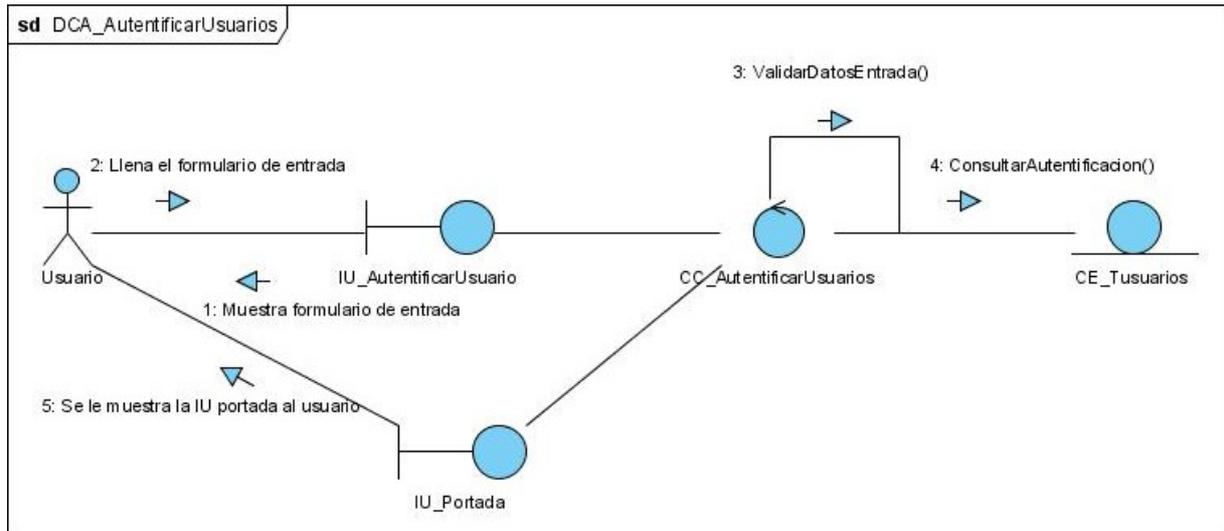
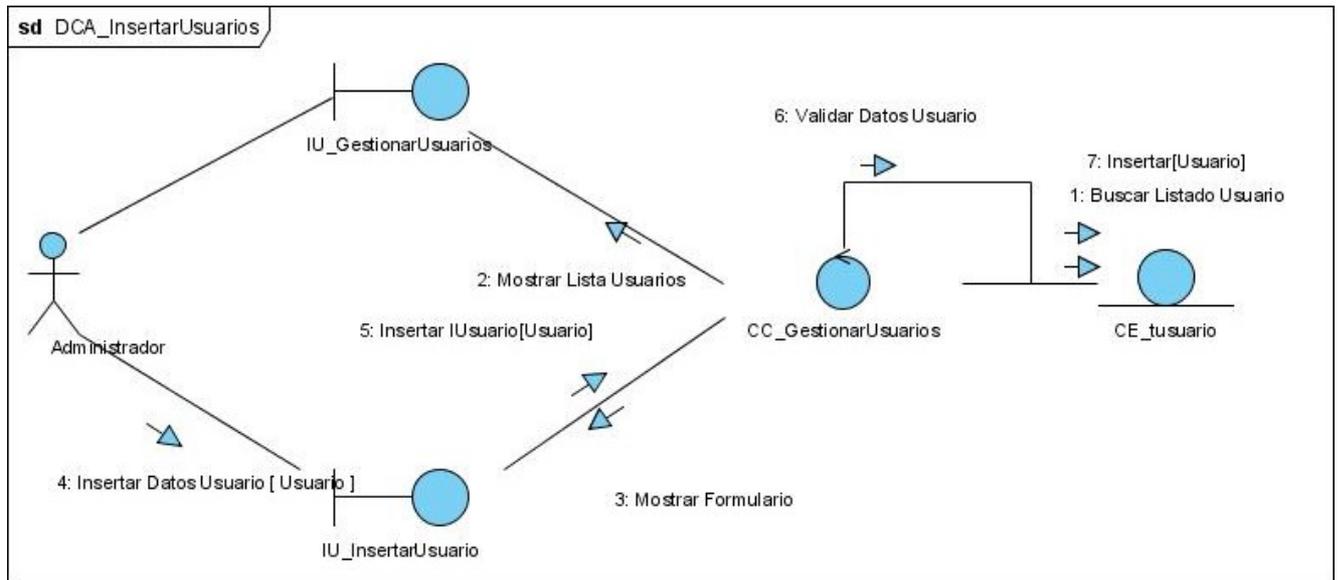
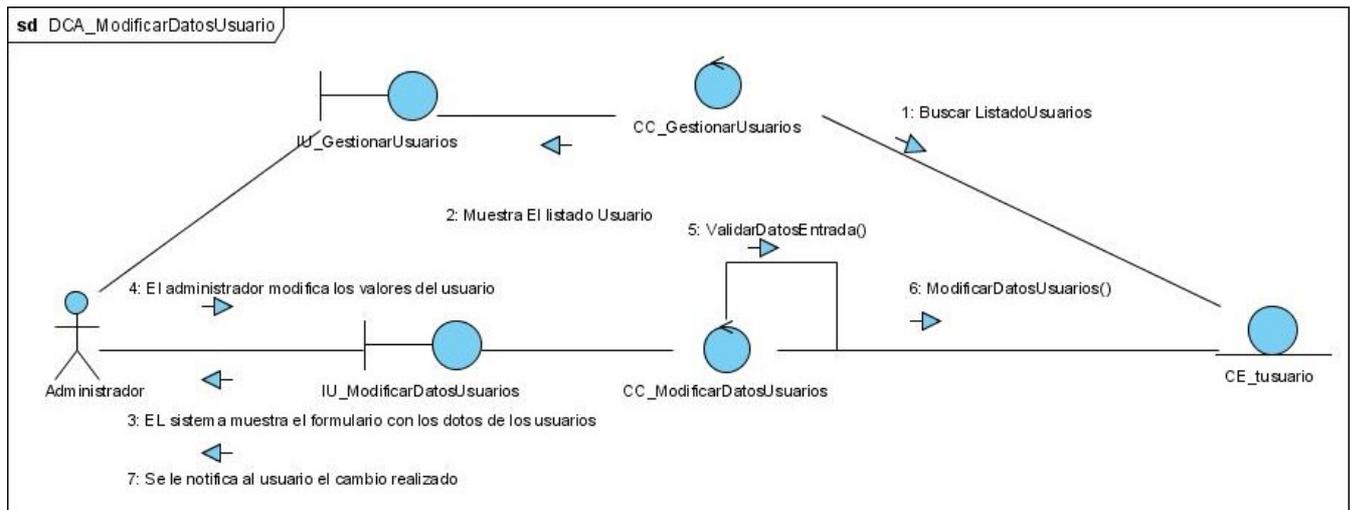


Figura 1 Diagrama de colaboración del caso de uso Autenticar Usuarios

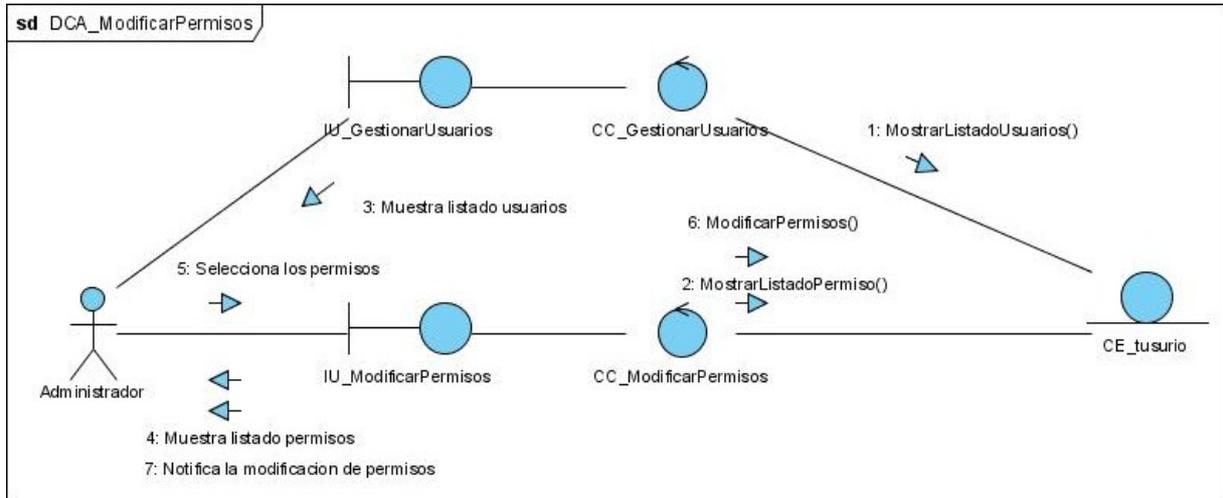
Diagramas de colaboración por escenarios del caso de uso GestionarUsuarios:



**Figura 2 Diagrama de colaboración del escenario Insertar Usuarios del caso de uso Gestionar Usuarios**

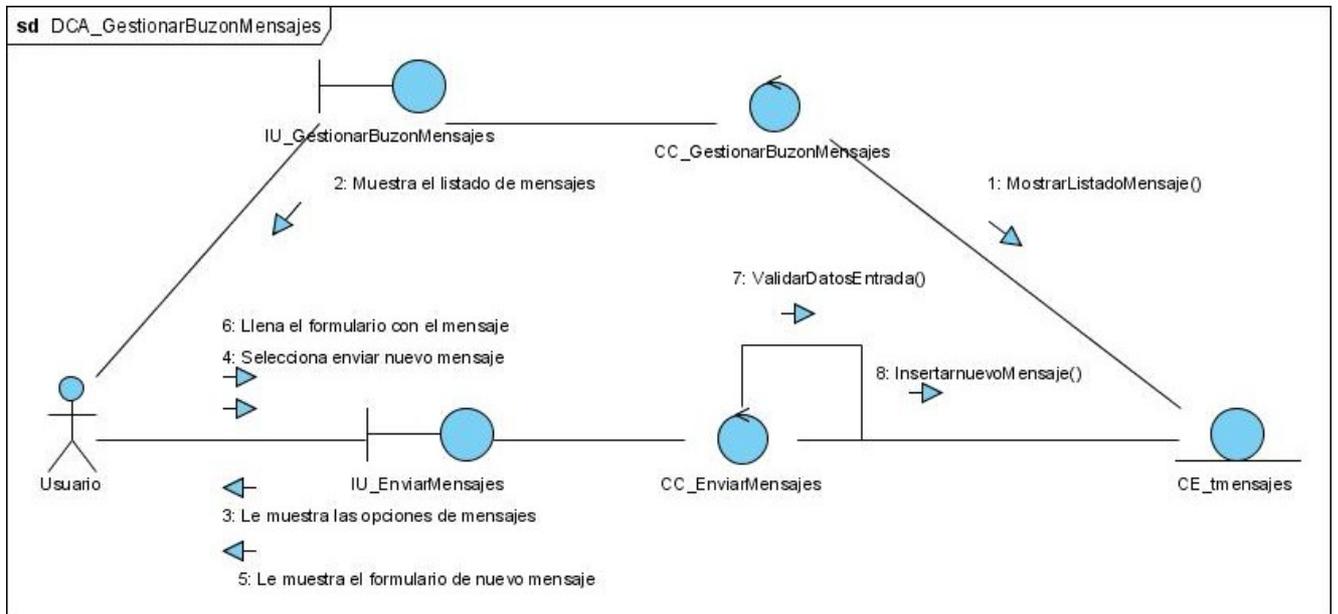


**Figura 3 Diagrama de colaboración de ModificarDatosUsuarios**

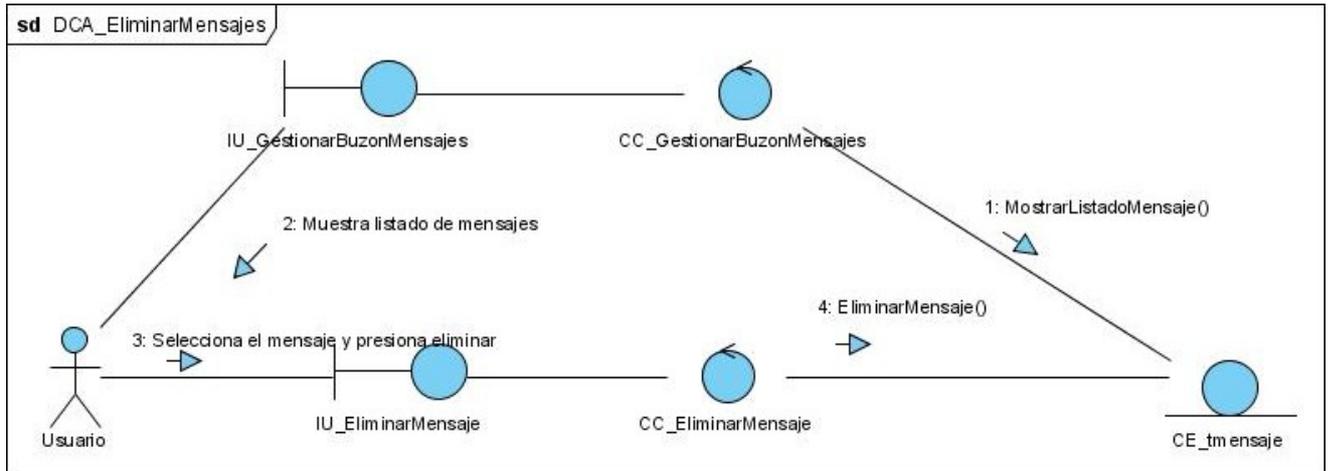


**Figura 4 Diagrama de colaboración del escenario Modificar Permisos**

Diagramas de componentes del caso de uso GestionarBuzonMensajes

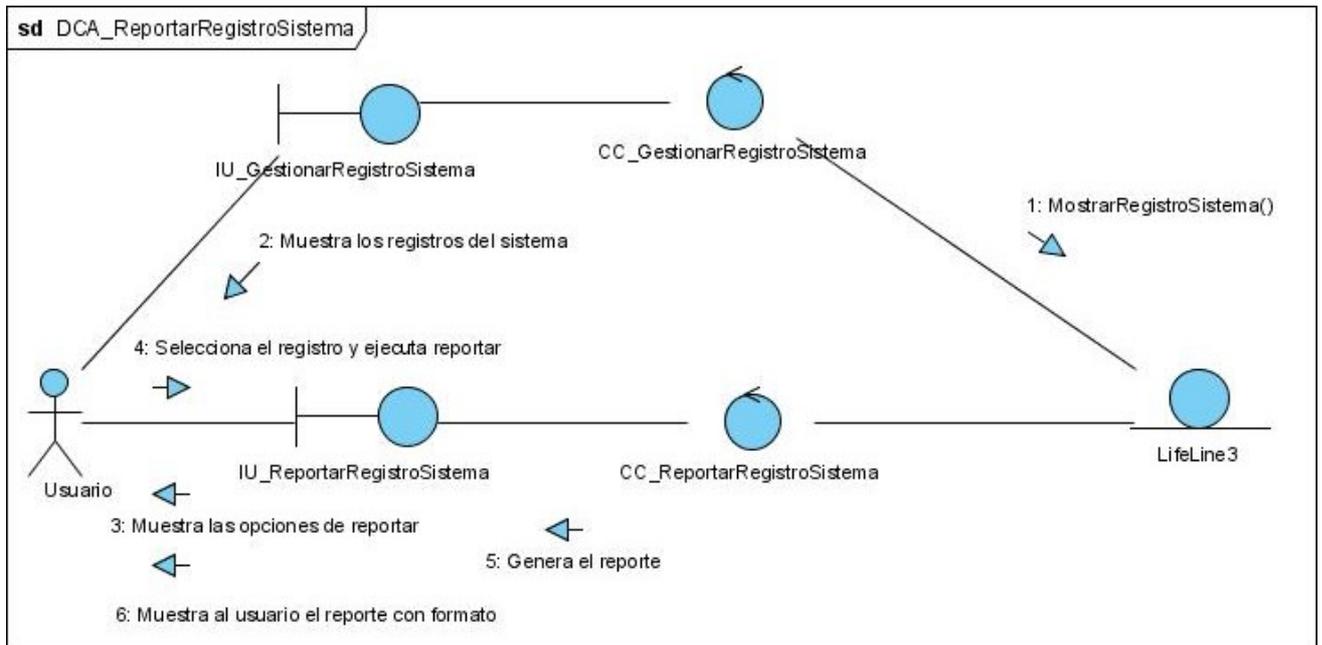


**Figura 5 Diagrama de colaboración del escenario Enviar Mensaje**

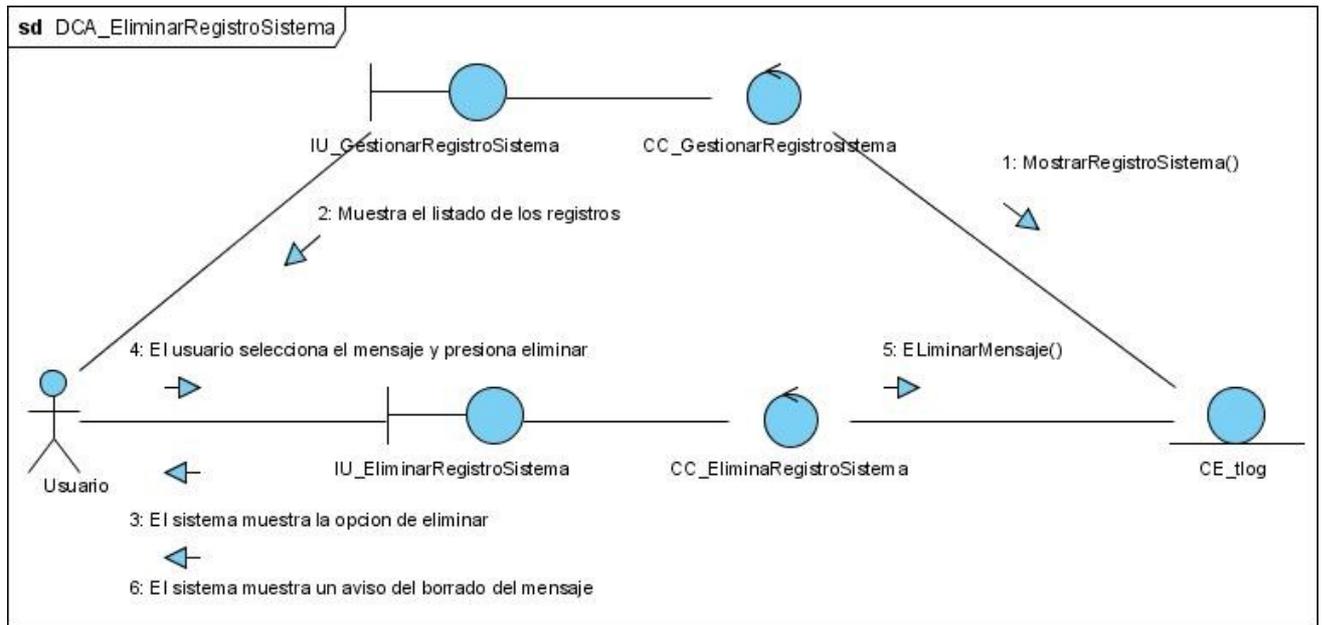


**Figura 6 Diagrama de colaboración del escenario Eliminar Mensaje.**

Diagramas de componentes del caso de uso GestionarRegistroSistema:

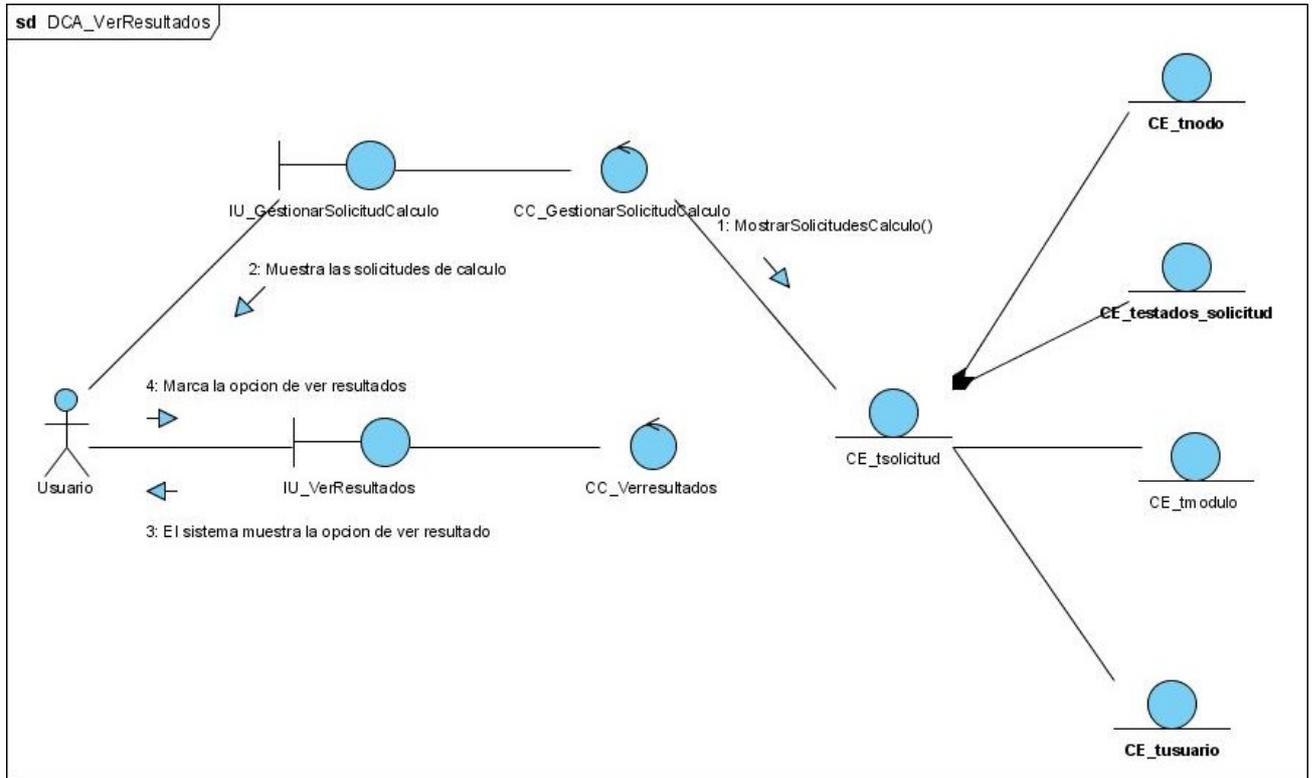


**Figura 7 Diagrama de colaboración del escenario Reportar Registros del Sistema.**

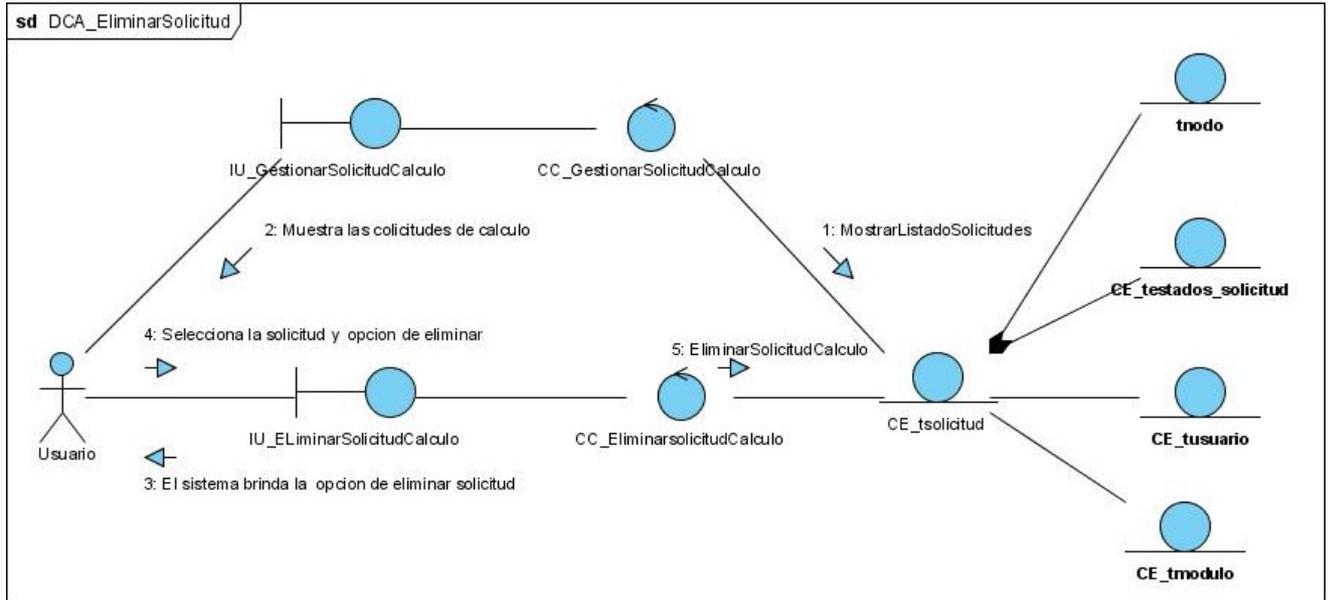


**Figura 8 Diagrama de colaboración del escenario Eliminar Registros del Sistema**

Diagramas de componentes del caso de uso GestionarSolicitudCalculo.



**Figura 9 Diagrama de colaboración del escenario Ver Resultado.**



**Figura 10 Diagrama de colaboración del escenario Eliminar Solicitud de Calculo**

### Anexo III: Diagramas de clases del diseño.

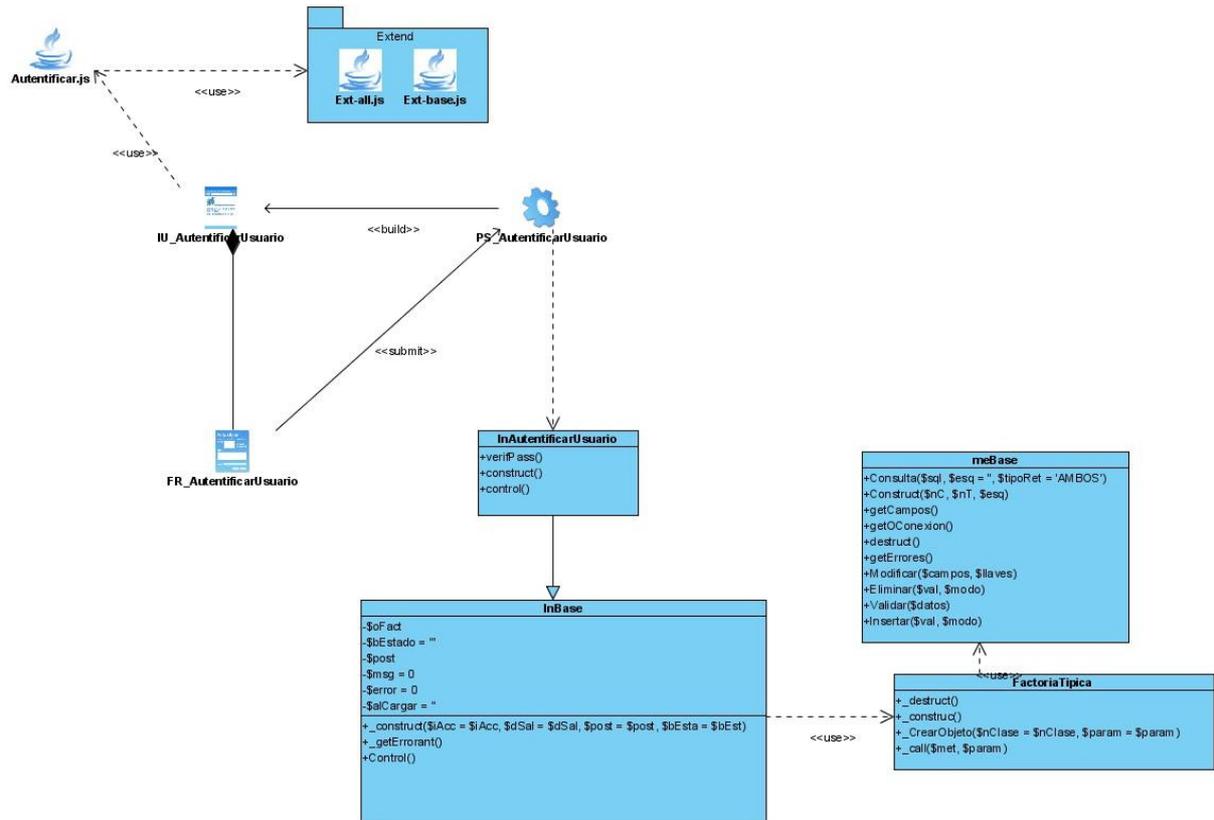
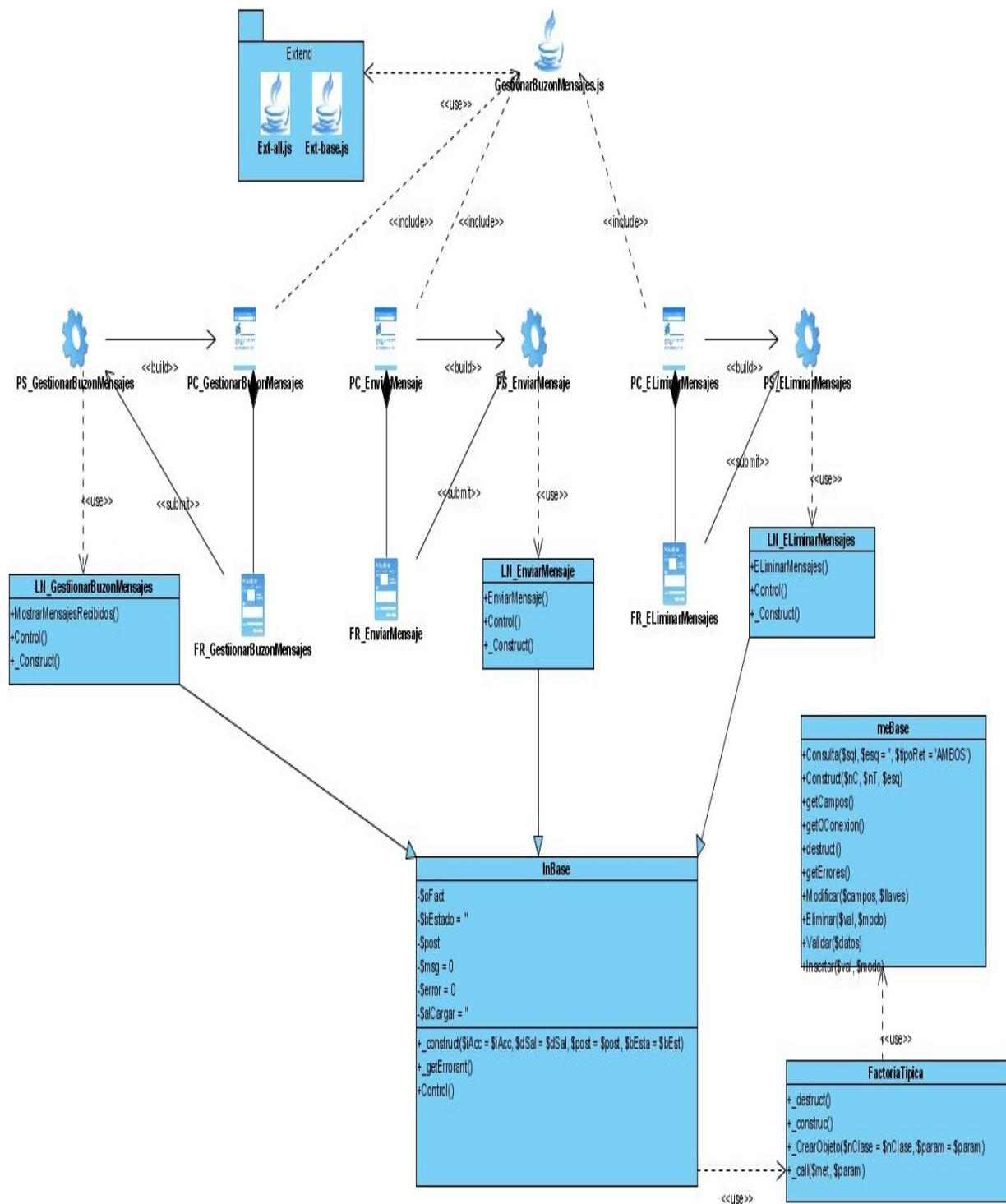


Figura 1 Diagrama de clases del diseño del caso de uso AutenticarUsuarios



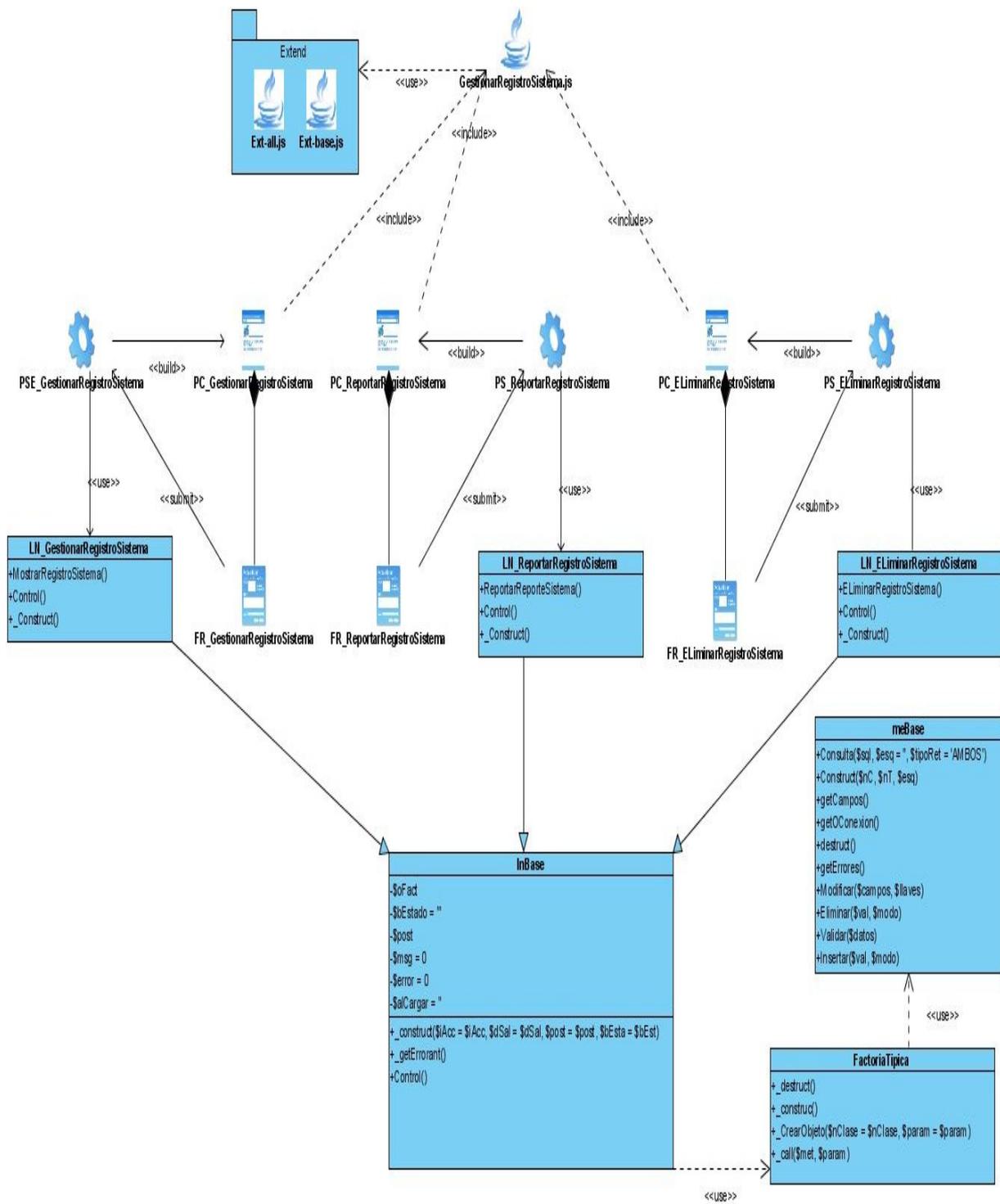


Figura 3 Diagrama de clases del diseño del caso de uso GestionarRegistroSistema

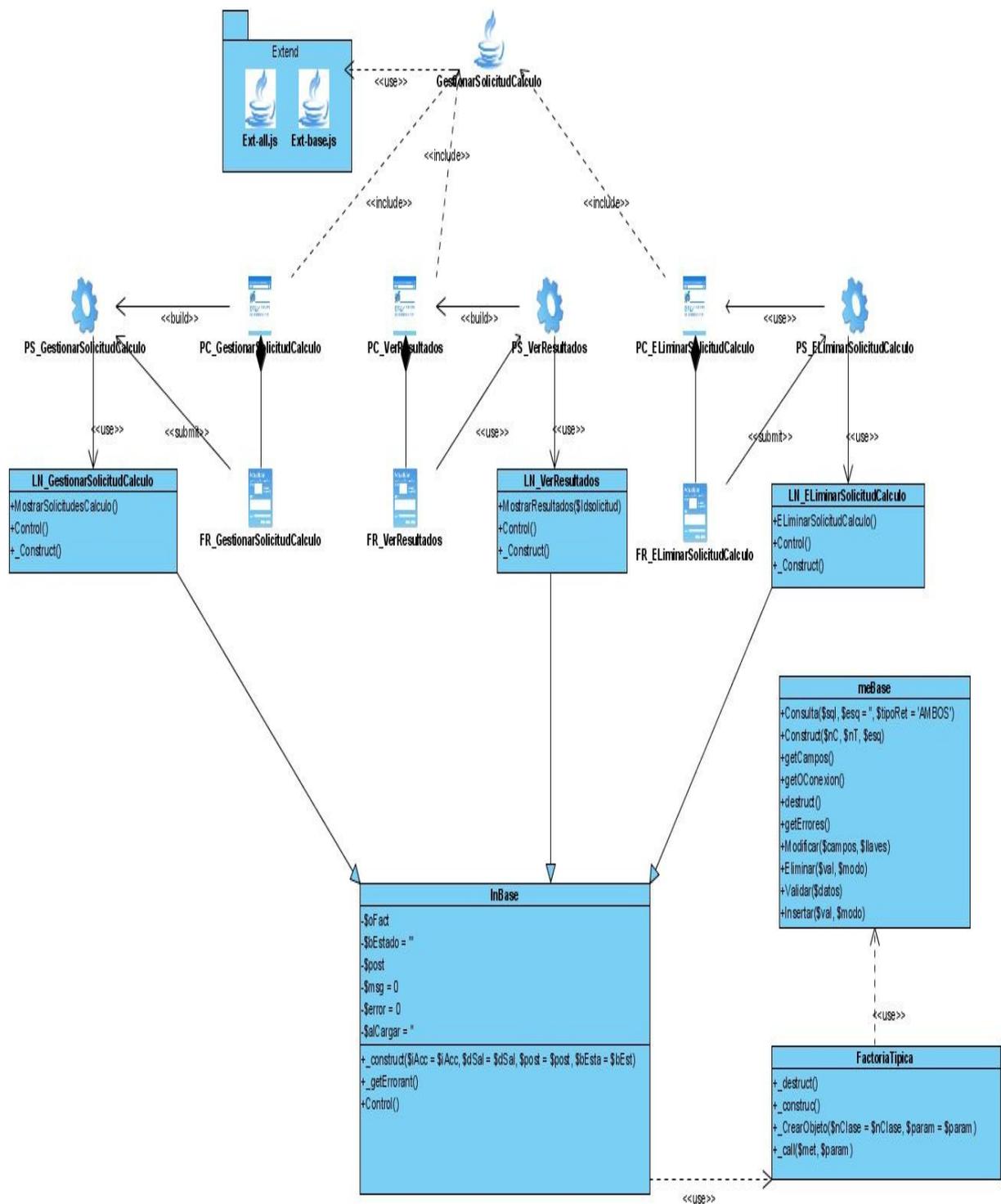


Figura 4 Diagrama de clases del diseño del caso de uso GestionarSolicitudCalculo.

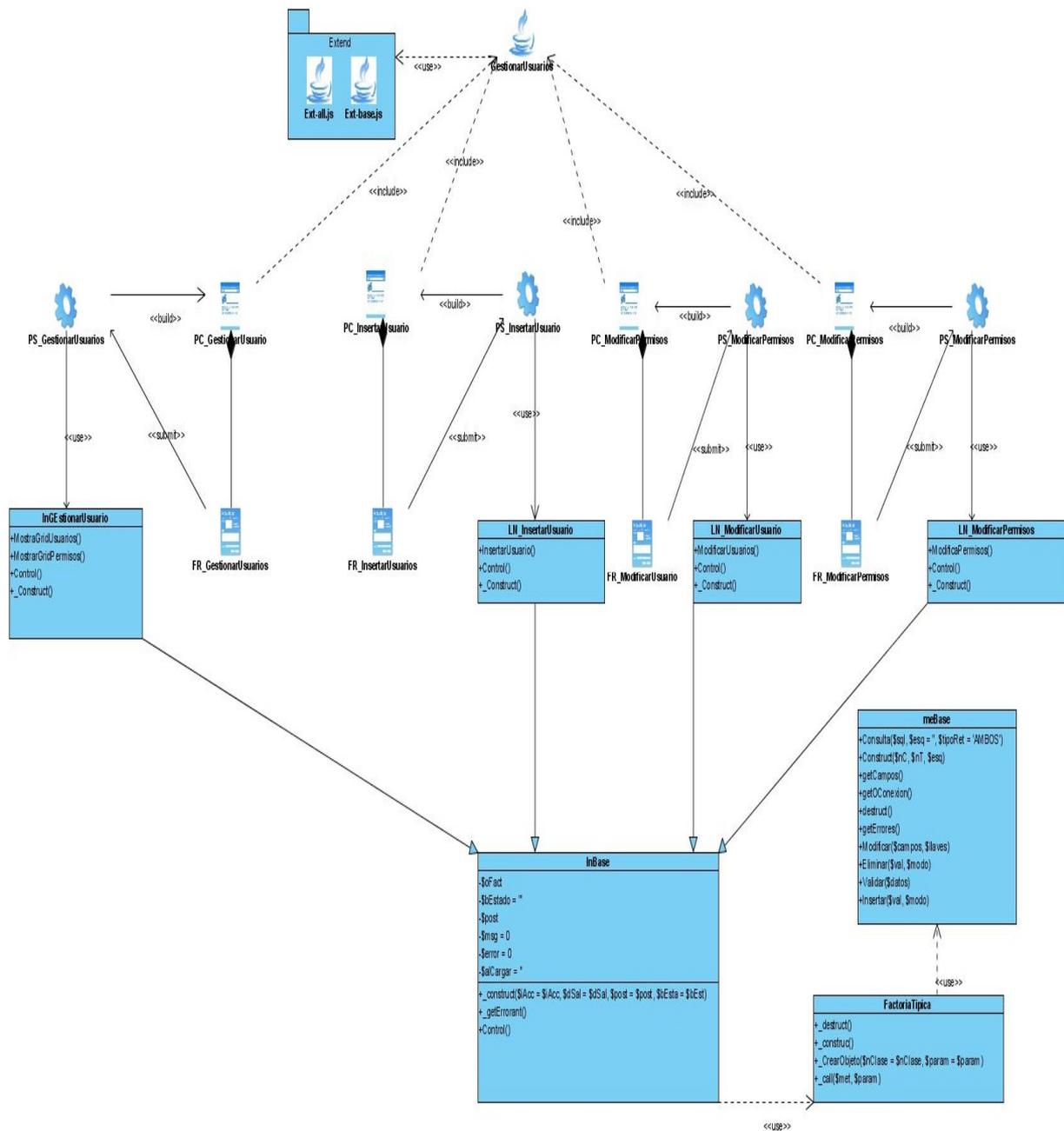


Figura 5 Diagrama de clases del diseño del caso de uso GestionarUsuarios

## Anexo IV: Diagramas de interacción de clase del diseño.

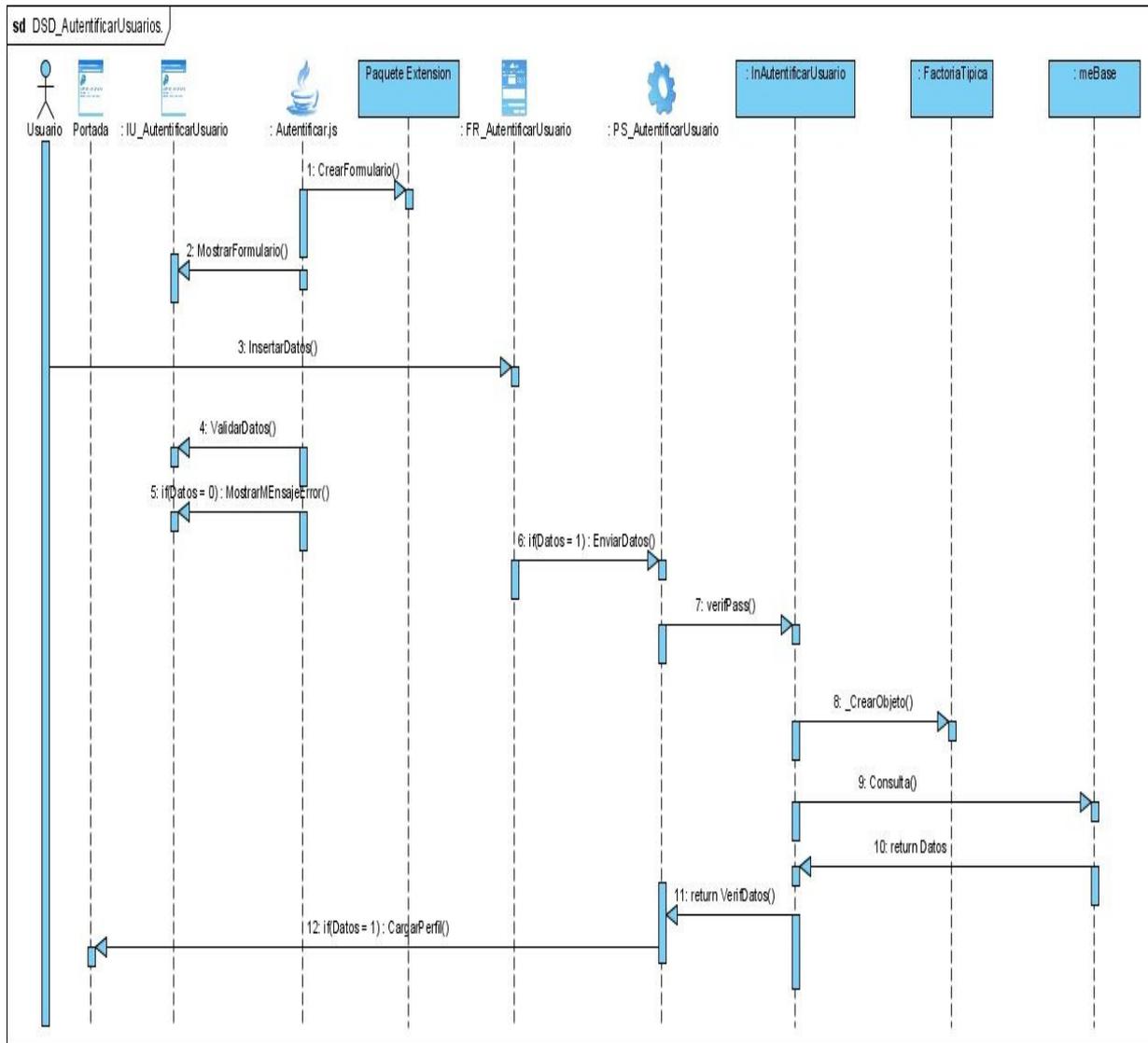


Figura 1 Diagrama de secuencia de clases del diseño del escenario autentificar usuarios

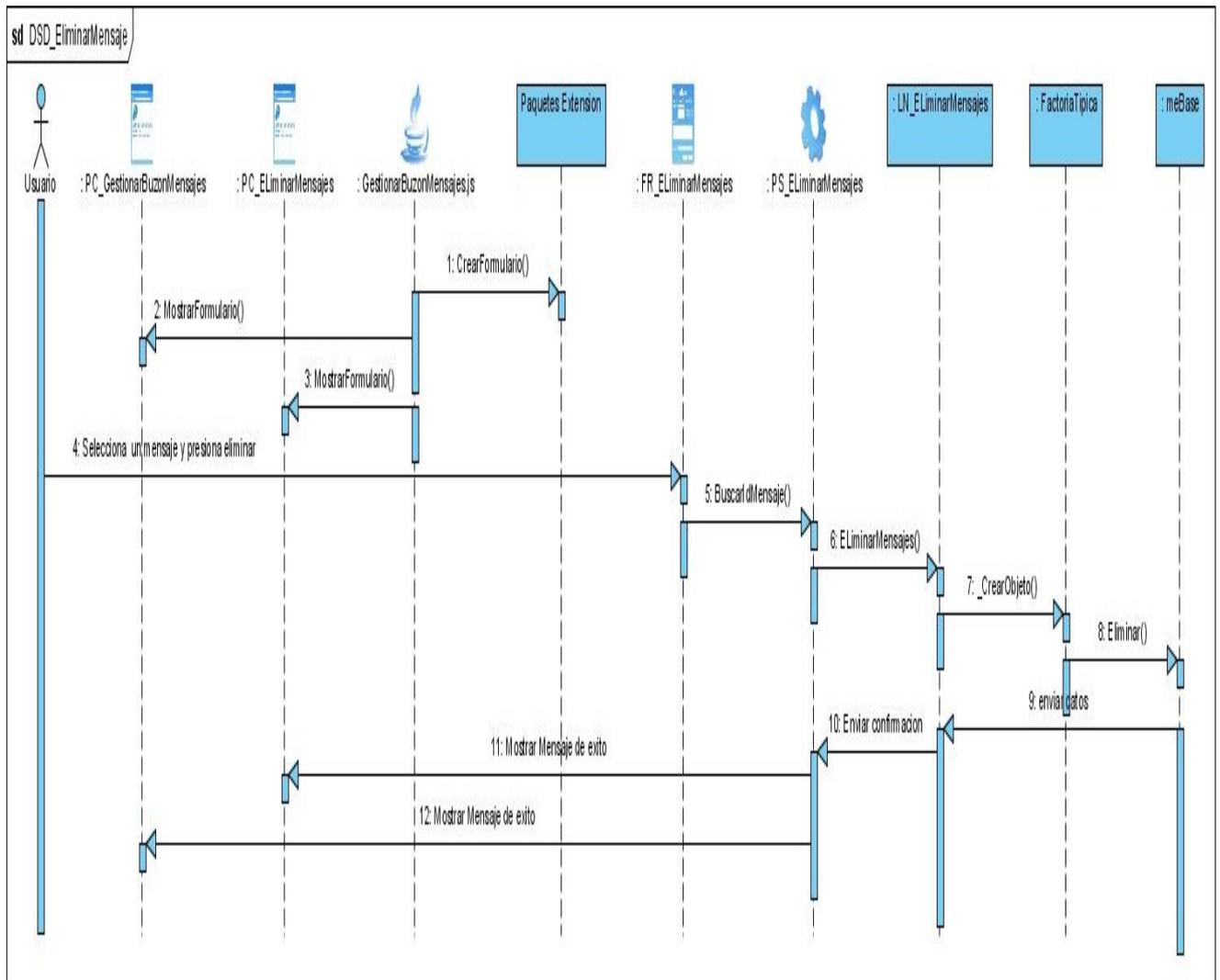
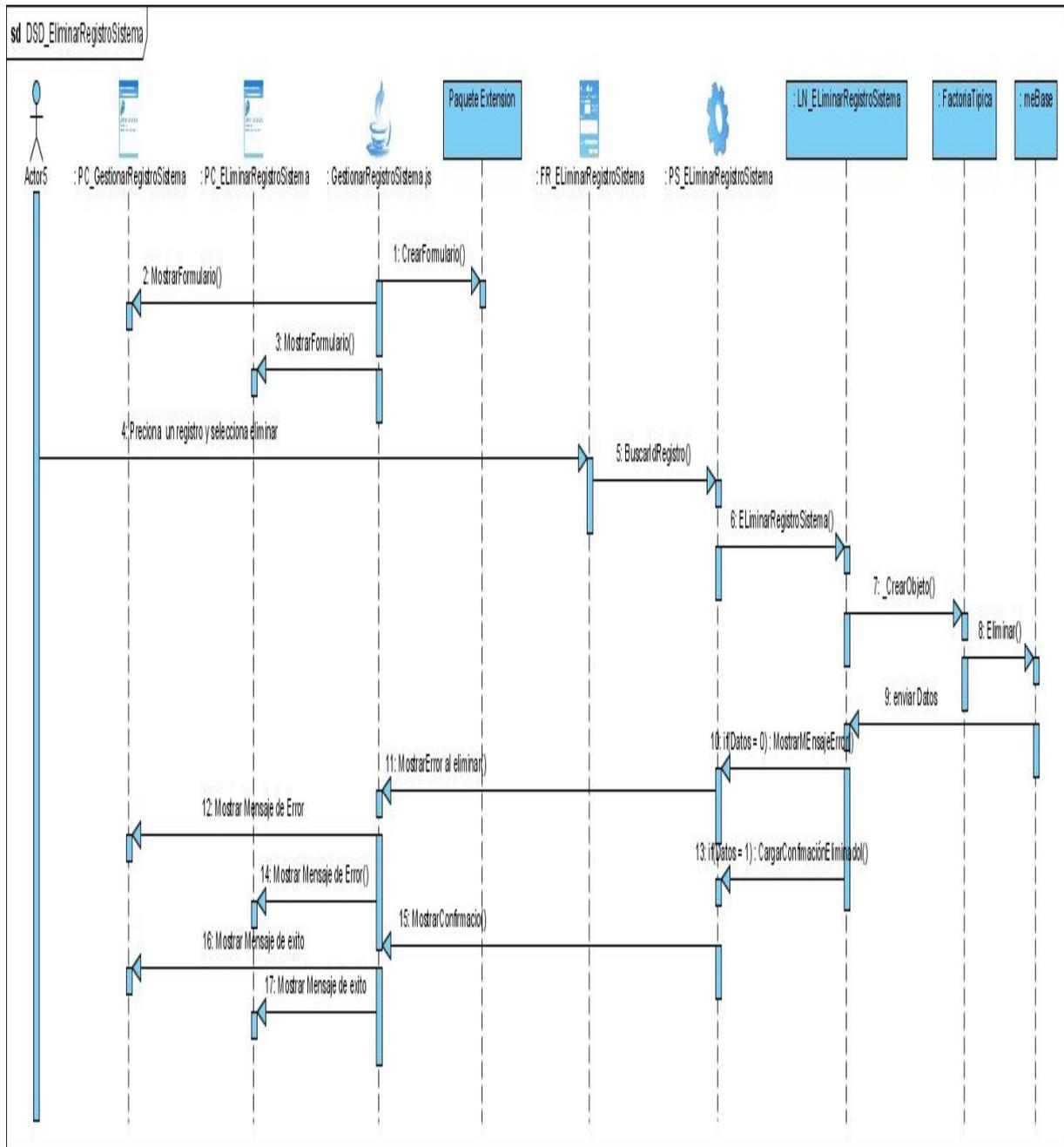


Figura 2 Diagrama de secuencia de clases del diseño del escenario Eliminar Mensaje



**Figura 3 Diagrama de secuencia de clases del diseño del escenario eliminar registro sistema.**

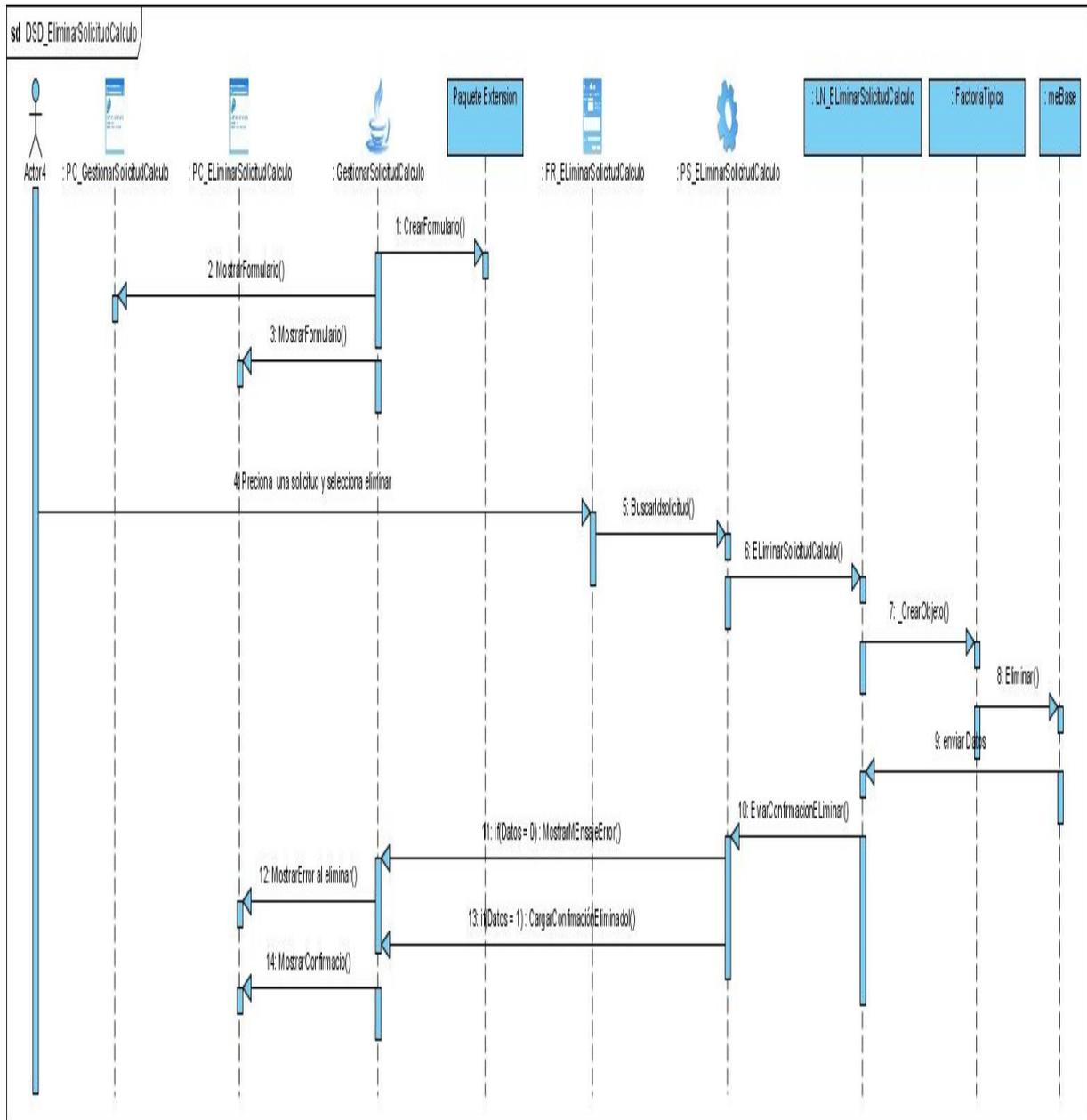


Figura 4 Diagrama de secuencia del diseño del escenario eliminar solicitud de calculo

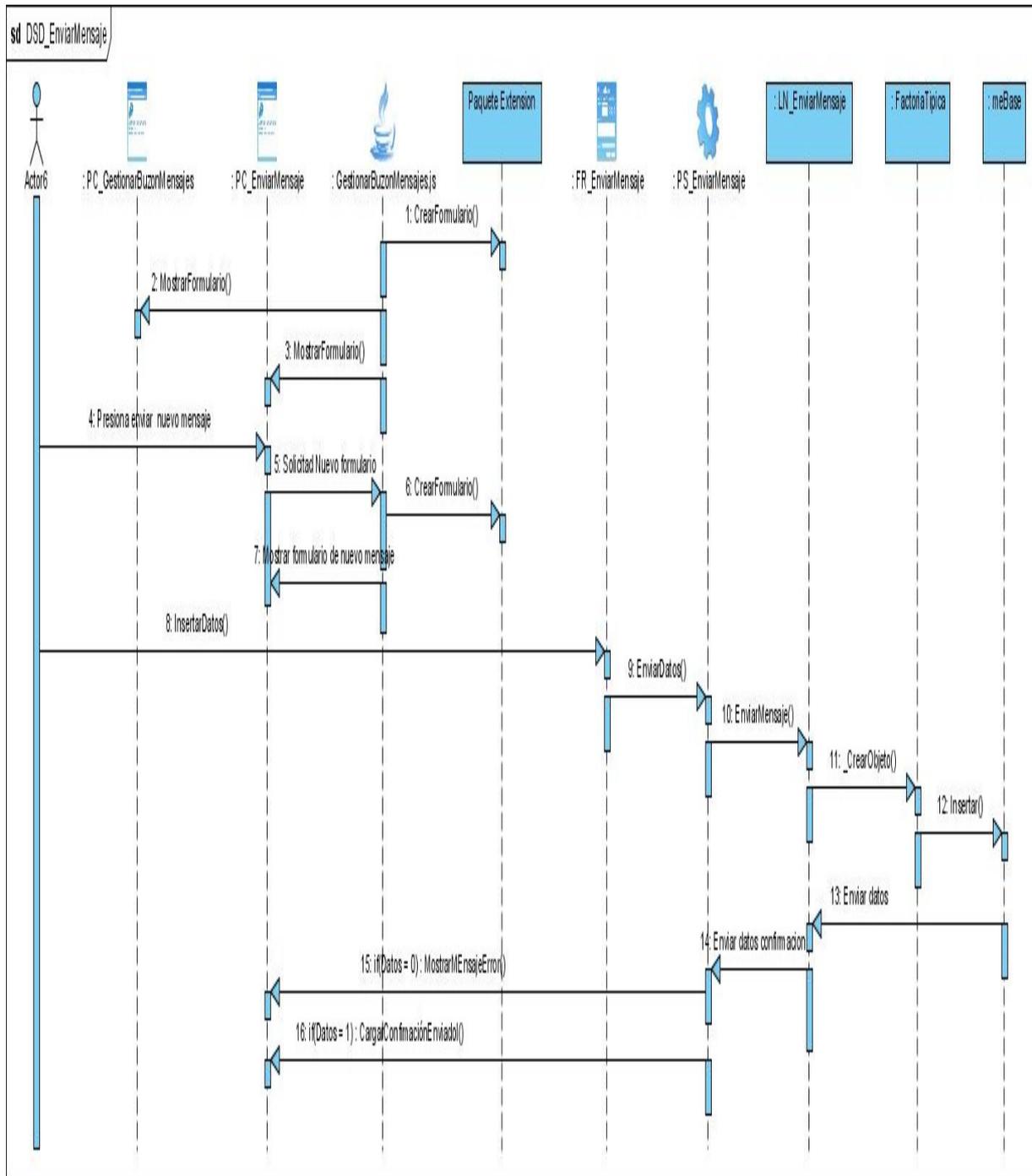
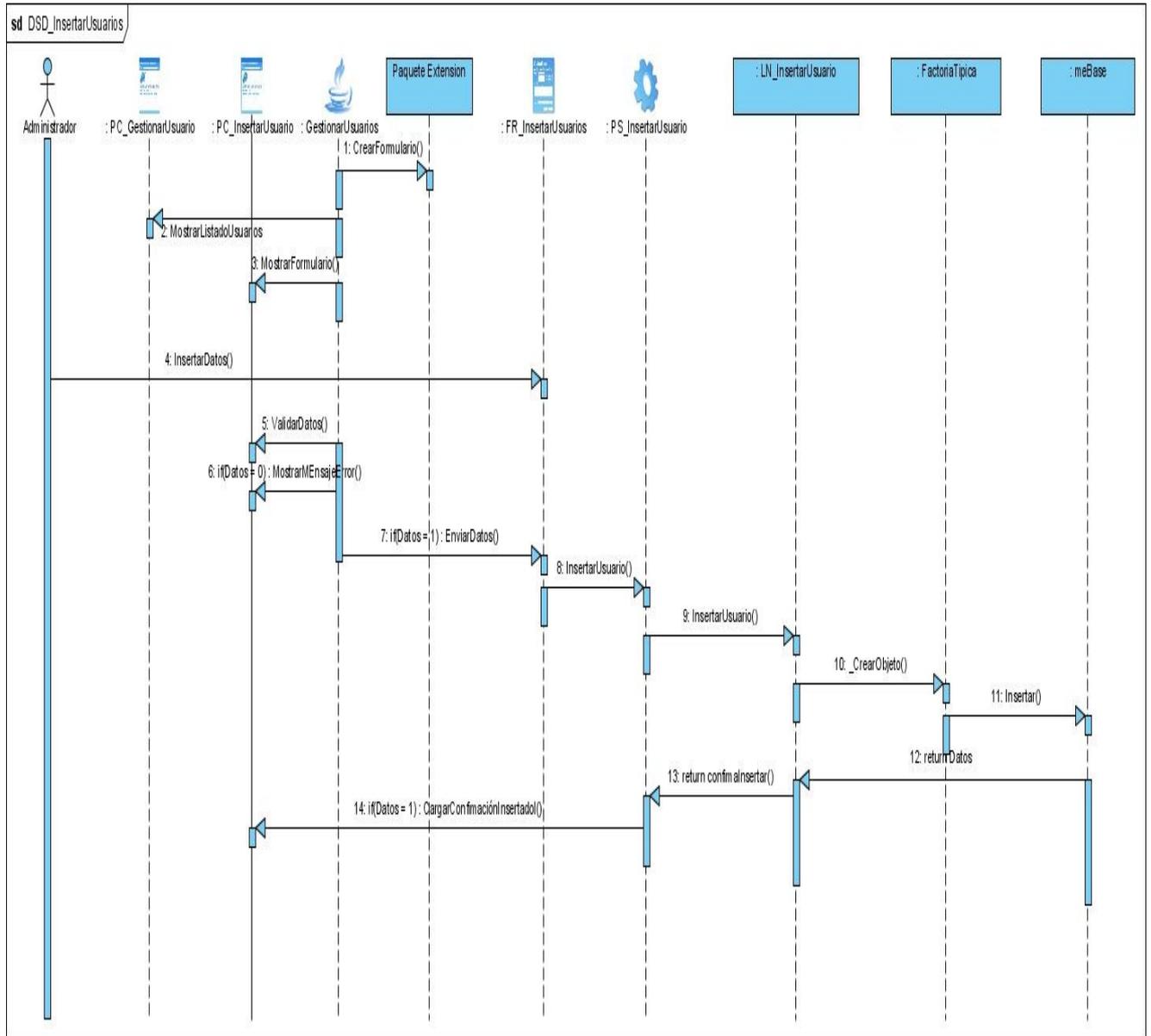
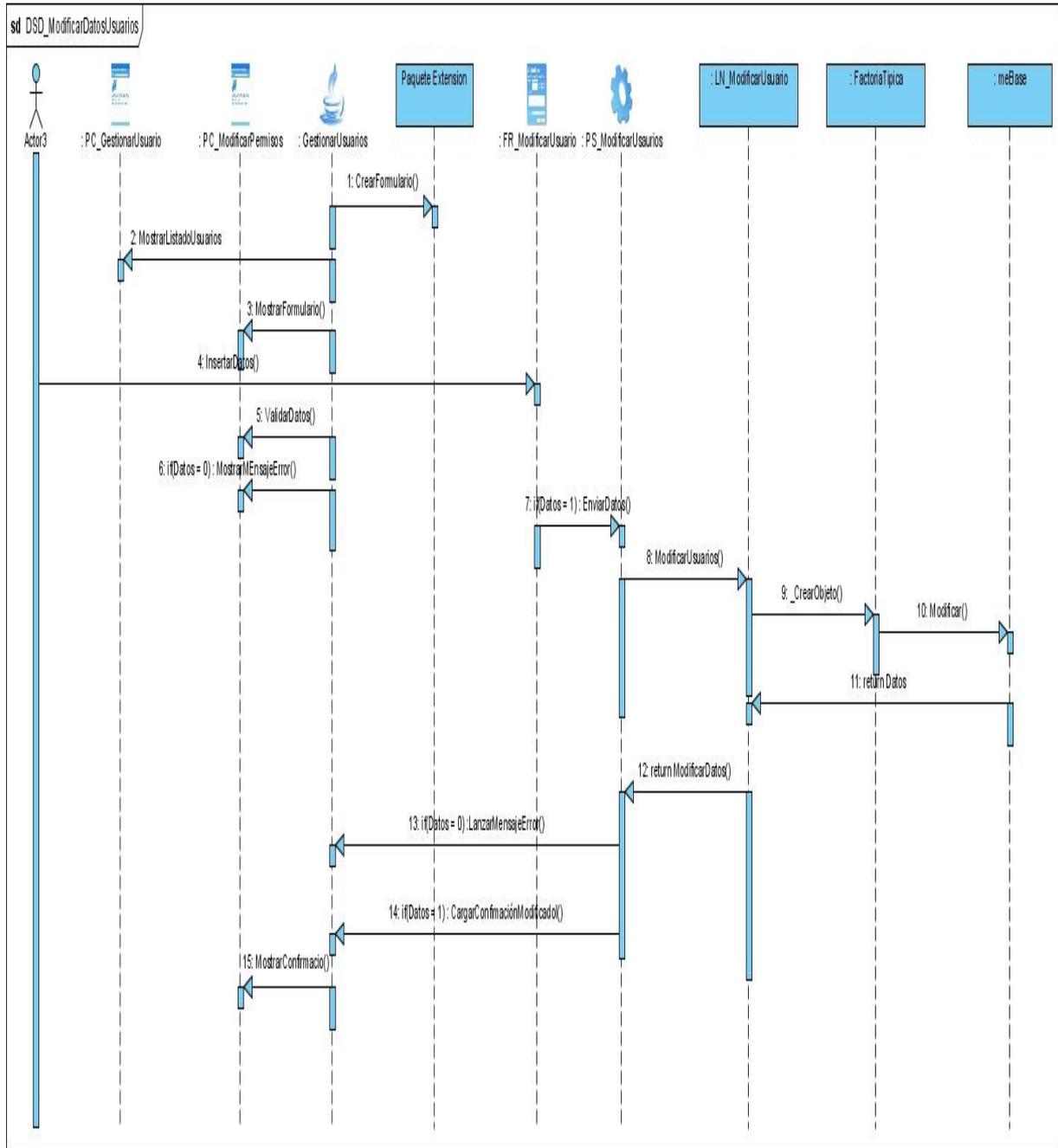


Figura 5 Diagrama de secuencia del diseño del enviar mensaje



**Figura 6 Diagrama de secuencia del diseño del escenario Insertar Usuarios**



**Figura 7 Diagrama de secuencia del diseño del escenario Modificar datos de usuario**

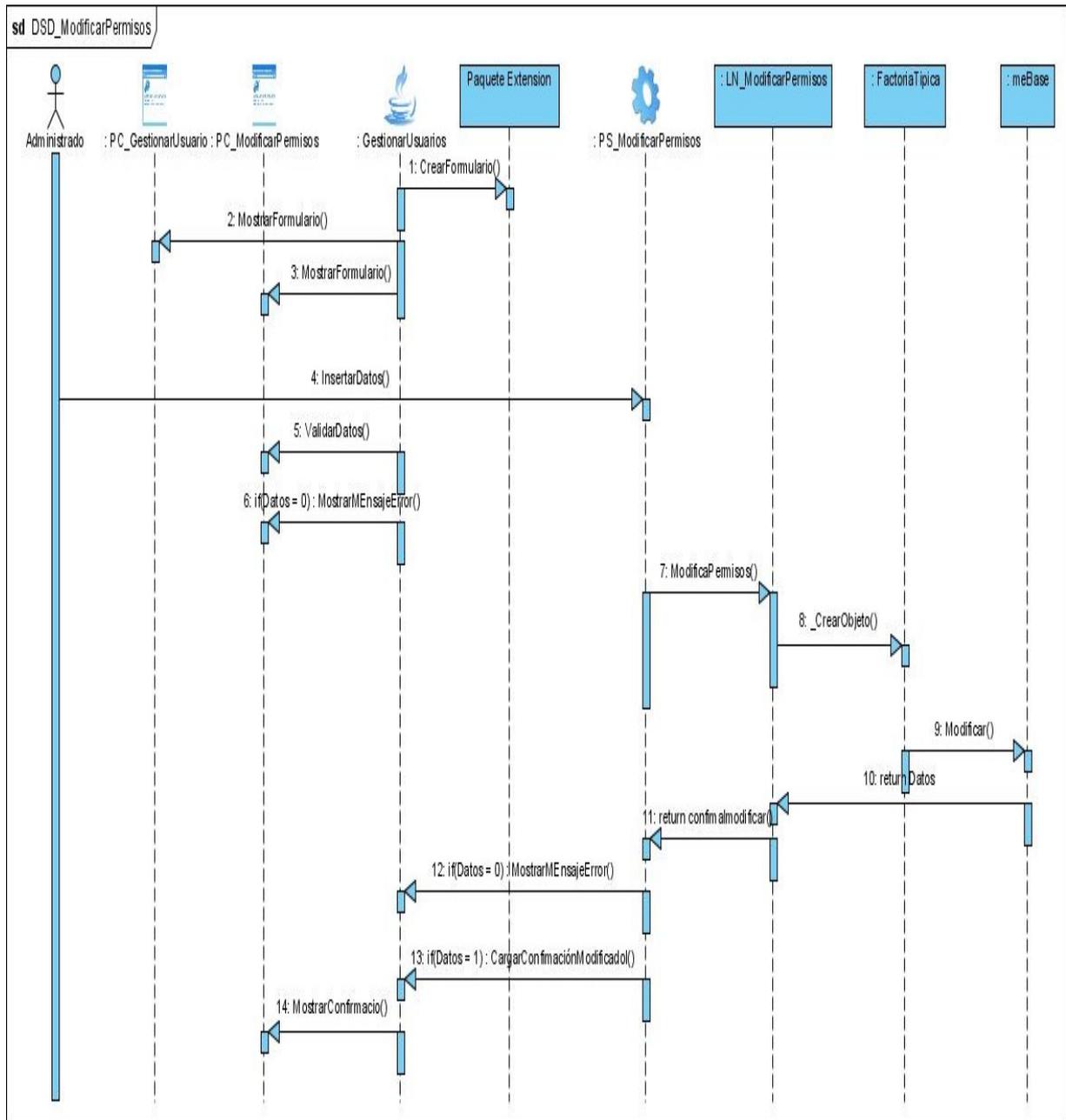
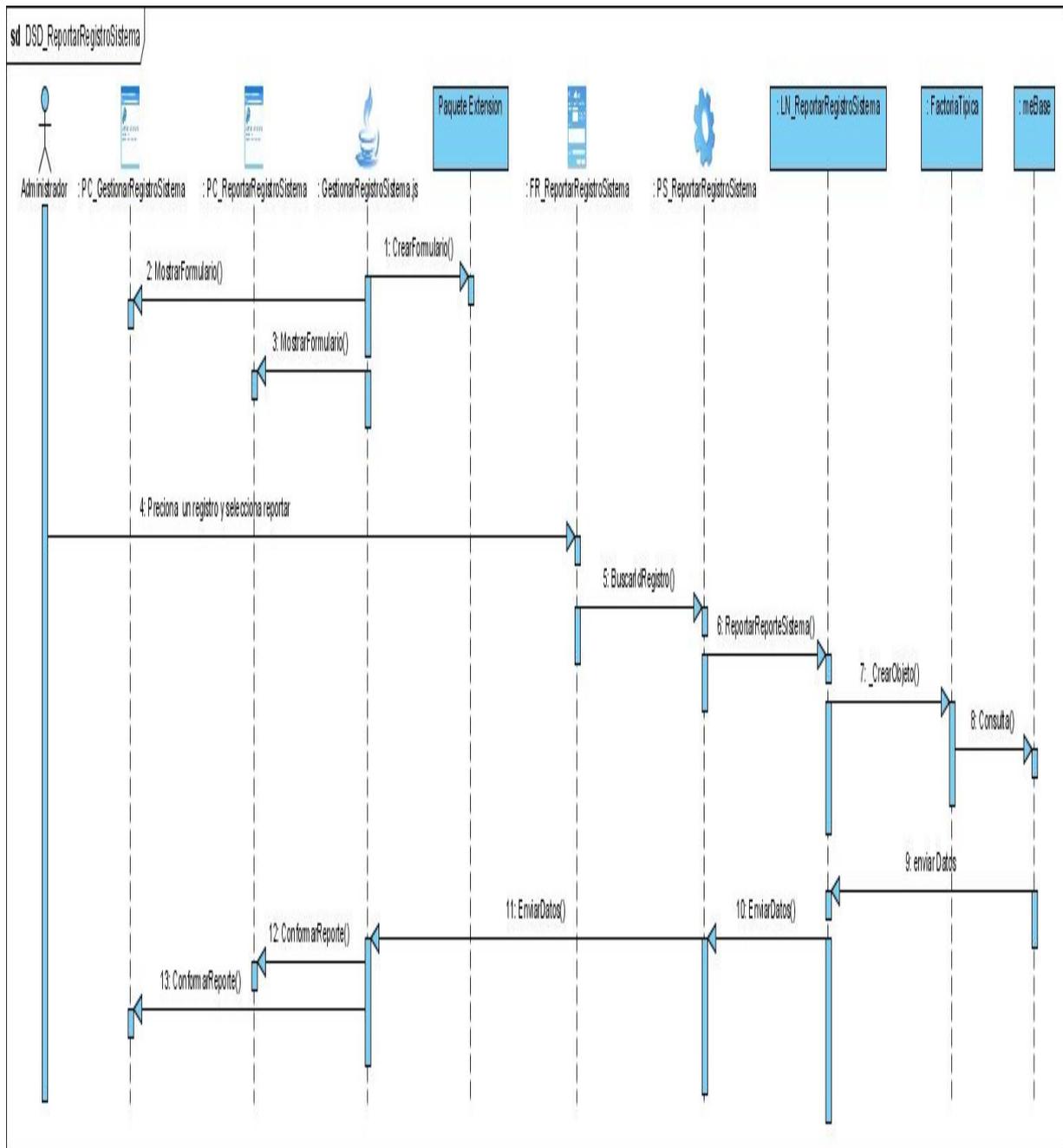


Figura 8 Diagrama de secuencia del diseño del escenario modificar permisos



**Figura 9 Diagrama de secuencia del diseño del escenario reportar registro sistema**

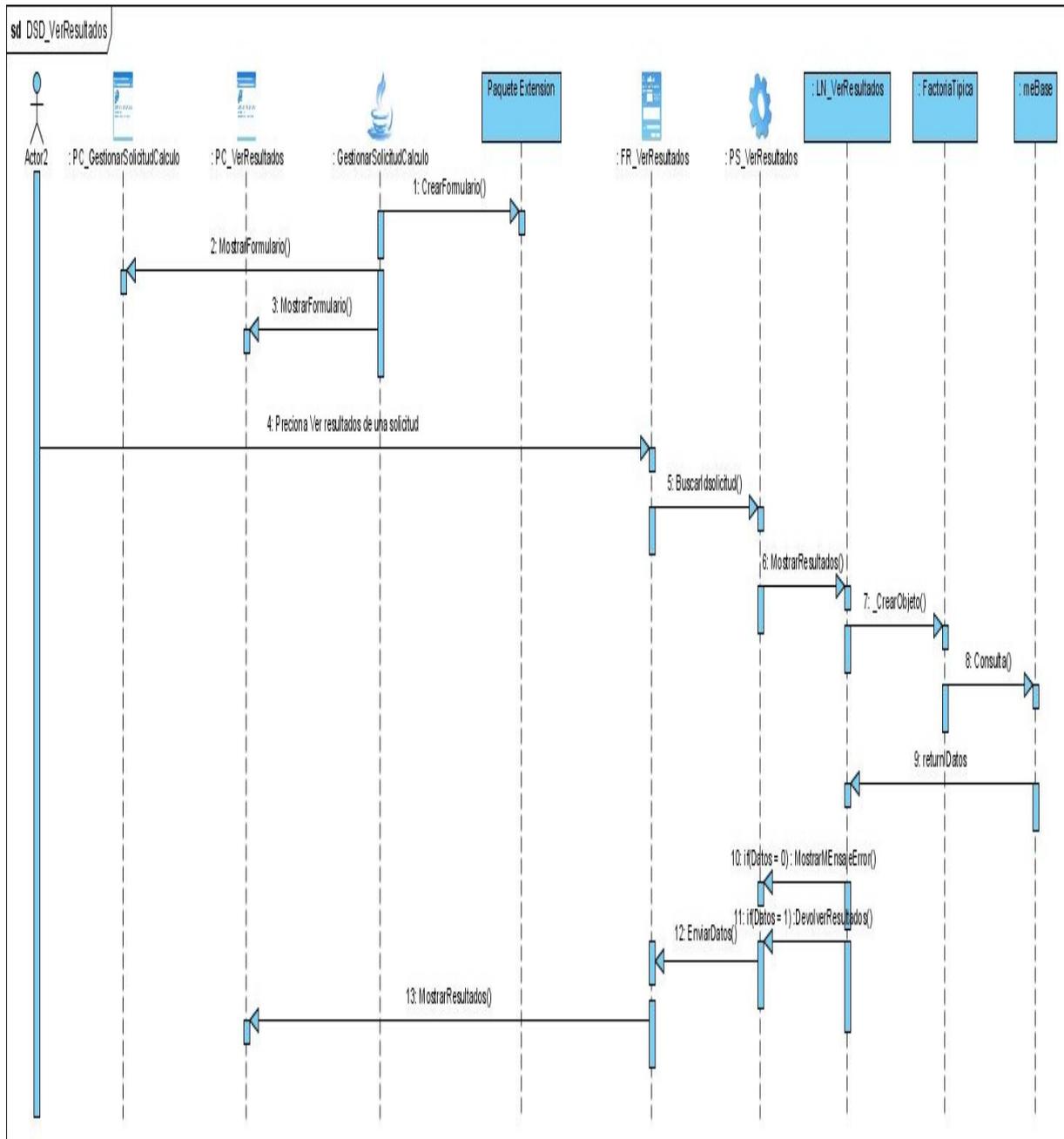


Figura 10 Diagrama de secuencia del diseño del escenario ver resultados.

**Anexo V: Descripción de las tablas del Diagrama Entidad-Relación.**

<b>Nombre de la tabla:</b> tsolicitud		
<b>Descripción:</b> Almacena la información de las solicitudes de cálculo que se realizan.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
idsolicitud	bigint	Identificador de la solicitud.
fecha	date	Fecha en que se realiza la solicitud
complejidad	integer	Valor asociado a la complejidad de la solicitud. Dependiente del volumen de información que procesa la solicitud.
idusuario	integer	Identificador del usuario que realiza la solicitud.
idnodo	integer	Identificador del nodo en el que se ejecuta la subtaska máster de la solicitud.
idmodulo	integer	Identificador asociado al módulo de cálculo al que pertenece la solicitud.
idestado	integer	Identificador asociado al estado en que se encuentra la solicitud.

Cuadro 5.1: Tabla tsolicitud

<b>Nombre de la tabla:</b> testados_solicitud		
<b>Descripción:</b> Almacena los tipos de estados que pueden tener las solicitudes en dependencia del nivel de procesamiento en que se encuentran.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
idestado	bigint	Identificador único de la tabla testados_solicitud.
descripción	varchar(80)	Descripción detallada del estado de la solicitud
nombre_estado	varchar(20)	Nombre del estado de la solicitud

Cuadro 5.2: Tabla testados\_solicitud

<b>Nombre de la tabla:</b> tprestaciones		
<b>Descripción:</b> Almacena las prestaciones que tiene un módulo de cómputo. Representan las funcionalidades de un módulo acoplado a la plataforma.		
<b>Atributo</b>	<b>Tipo</b>	<b>Descripción</b>
idprestacion	integer	Identificador único de la tabla tprestaciones.
nombre	varchar(80)	Descripción detallada del estado de procesamiento de una solicitud
referencia	varchar(80)	Referencia a la página que carga la prestación.
idmodulo	integer	Identificador del módulo al que pertenece la prestación.

Cuadro 5.3: Tabla tprestaciones

<b>Nombre de la tabla:</b> ttipo _ log		
Descripción: Almacena los tipos de registro de información del sistema		
Atributo	Tipo	Descripción
idtipo	integer	Identificador único de la tabla ttipo _ log.
tipo	varchar(30)	Tipo de registro.
descripcion	varchar(80)	Descripción del registro.
Nombre	varchar(40)	Nombre del registro.

Cuadro 5.4: Tabla ttipo\_log

<b>Nombre de la tabla:</b> tlog		
Descripción: Almacena la información de los registros que arrojan los nodos del clúster de procesamiento.		
Atributo	Tipo	Descripción
idlog	bigint	Identificador único de la tabla tlog.
mensaje	varchar(255)	Mensaje asociado al registro.
fecha	date	Fecha en que se emitió.
fechaseg	integer	Fecha en segundos.
descripcion	varchar(255)	Descripción detallada del registro.
ldtipo	Integer	Tipo de registro emitido.
idnodo	Integer	Nodo que lo emitió.

Cuadro 5.5: Tabla tlog

<b>Nombre de la tabla:</b> tnodo		
Descripción: Almacena la información de los nodos de un clúster de procesamiento.		
Atributo	Tipo	Descripción
idnodo	integer	Identificador del nodo.
nombre	varchar(20)	Nombre del nodo.
descripcion	varchar(100)	Información adicional del nodo
rendimiento	double precision	Valor asociado a la capacidad de procesamiento del nodo.

Cuadro 5.6: Tabla tnode

<b>Nombre de la tabla:</b> tusuario		
Descripción: Almacena la información de todos los usuarios del sistema. Estos pueden ser Administradores o Clientes.		
Atributo	Tipo	Descripción
idusuario	integer	Identificador único de la tabla tusuario.
nombre	varchar(30)	Nombre del usuario.
apellidos	varchar(60)	Apellidos del usuario.
contrasena	varchar(25)	Contraseña del usuario para la cuenta en el sistema.
usuario	varchar(30)	Nombre de la cuenta del usuario.
correo	varchar(60)	Cuenta de correo del usuario.
entidad	varchar(15)	Nombre de la entidad a la que pertenece el usuario.
fecha	date	Fecha de inscripción del usuario al sistema.

Cuadro 5.7: Tabla tusuario

<b>Nombre de la tabla:</b> tmensaje		
Descripción: Almacena la información de los mensajes que se envían los usuarios para compartir información de los resultados de las solicitudes que procesan.		
Atributo	Tipo	Descripción
idmensaje	bigint	Identificador único de la tabla tmensaje.
asunto	varchar(100)	Asunto del mensaje.
cuerpo	varchar(255)	Cuerpo del mensaje.
idusuario	integer	Identificador del usuario que envía el mensaje.
idusuariodestino	integer	Identificador del usuario que recibe el mensaje.

Cuadro 5.8: Tabla tmensaje

<b>Nombre de la tabla:</b> tcliente		
Descripción: Almacena la información relacionada con los usuarios que son clientes del sistema.		
Atributo	Tipo	Descripción
idprestacion	integer	Identificador único de la tabla tcliente

Cuadro 5.9: Tabla tcliente

<b>Nombre de la tabla:</b> tadministrador		
Descripción: Almacena la información relacionada con los administradores del sistema.		
Atributo	Tipo	Descripción
idadministrador	integer	Identificador que representa los usuarios del sistema que son administradores.

Cuadro 5.10: Tabla tadministrador

<b>Nombre de la tabla:</b> r_tpermisos_establecidos		
Descripción: Almacena los permisos que concede un administrador a un cliente.		
Atributo	Tipo	Descripción
idcliente	integer	Identificador del usuario cliente que recibe el permiso.
idadministrador	integer	Identificador del usuario administrador que otorga el permiso.
idpermiso	integer	Identificador del permiso otorgado.

Cuadro 5.11: Tabla r\_tpermisos\_establecidos

<b>Nombre de la tabla:</b> tpermiso		
Descripción: Almacena los permisos que se le conceden a los usuarios del sistema.		
Atributo	Tipo	Descripción
idpermiso	integer	Identificador único de la tabla tpermiso.
descripcion	varchar(80)	Descripción detallada del permiso.
nombre	varchar(80)	Nombre del permiso que se concede.

Cuadro 5.12 Tabla tpermiso

<b>Nombre de la tabla:</b> tmodulo_web		
Descripción: Almacena la información de los módulos de aplicación.		
Atributo	Tipo	Descripción
idmodulo	integer	Identificador único de la tabla tmodulo.
pagina_inicio	varchar(80)	Identificador de la página relacionada con el módulo.
idpermiso	varchar(80)	Identificador del permiso asociado a ese módulo.

Cuadro 5.13: Tabla tmodulo\_web

<b>Nombre de la tabla:</b> tmodulo		
Descripción: Almacena la información de los módulos generales que se acoplan a la plataforma. Pueden ser módulos de cálculo y/o modulo de aplicación.		

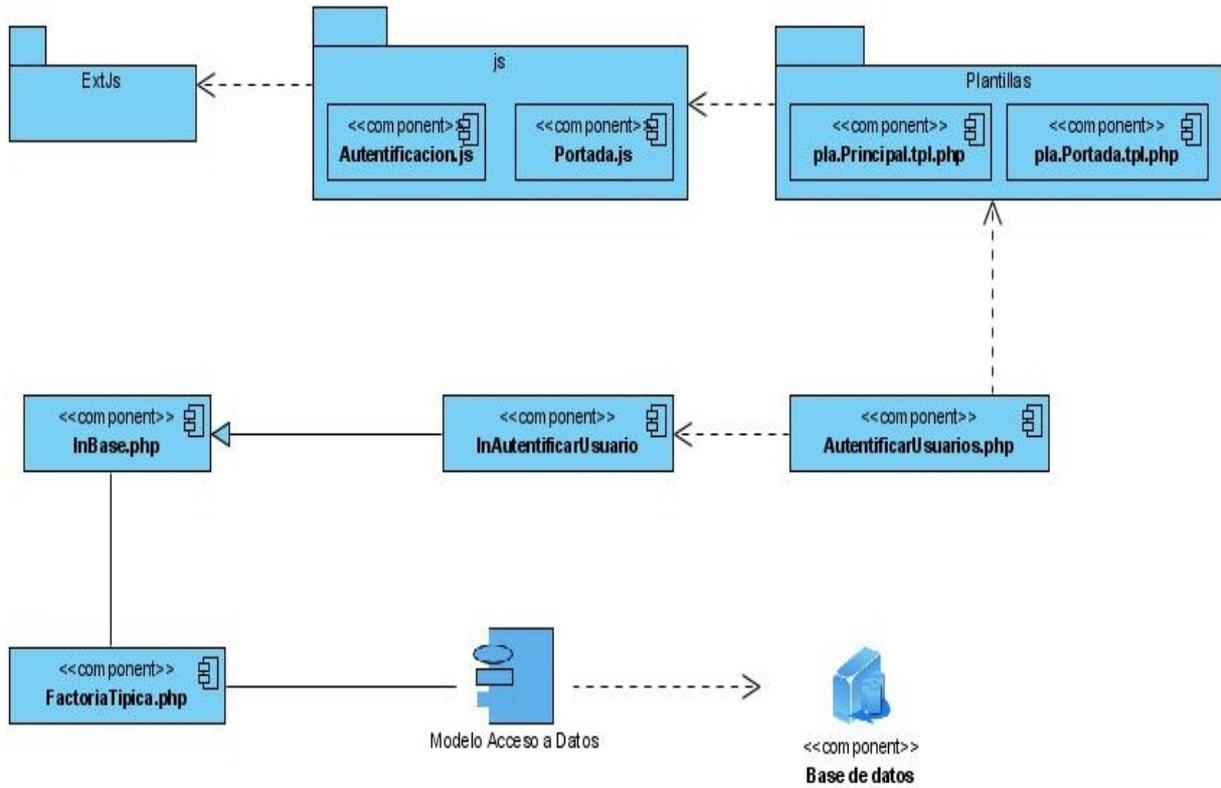
Atributo	Tipo	Descripción
idmodulo	integer	Identificador único de los módulos del sistema.
descripción	varchar (255)	Descripción de los diferentes módulos.
nombre	varchar (20)	Nombre de los módulos de la plataforma.
alias	varchar (20)	Es el alias que se le pone al modulo

Cuadro 5.14: Tabla tmodulo

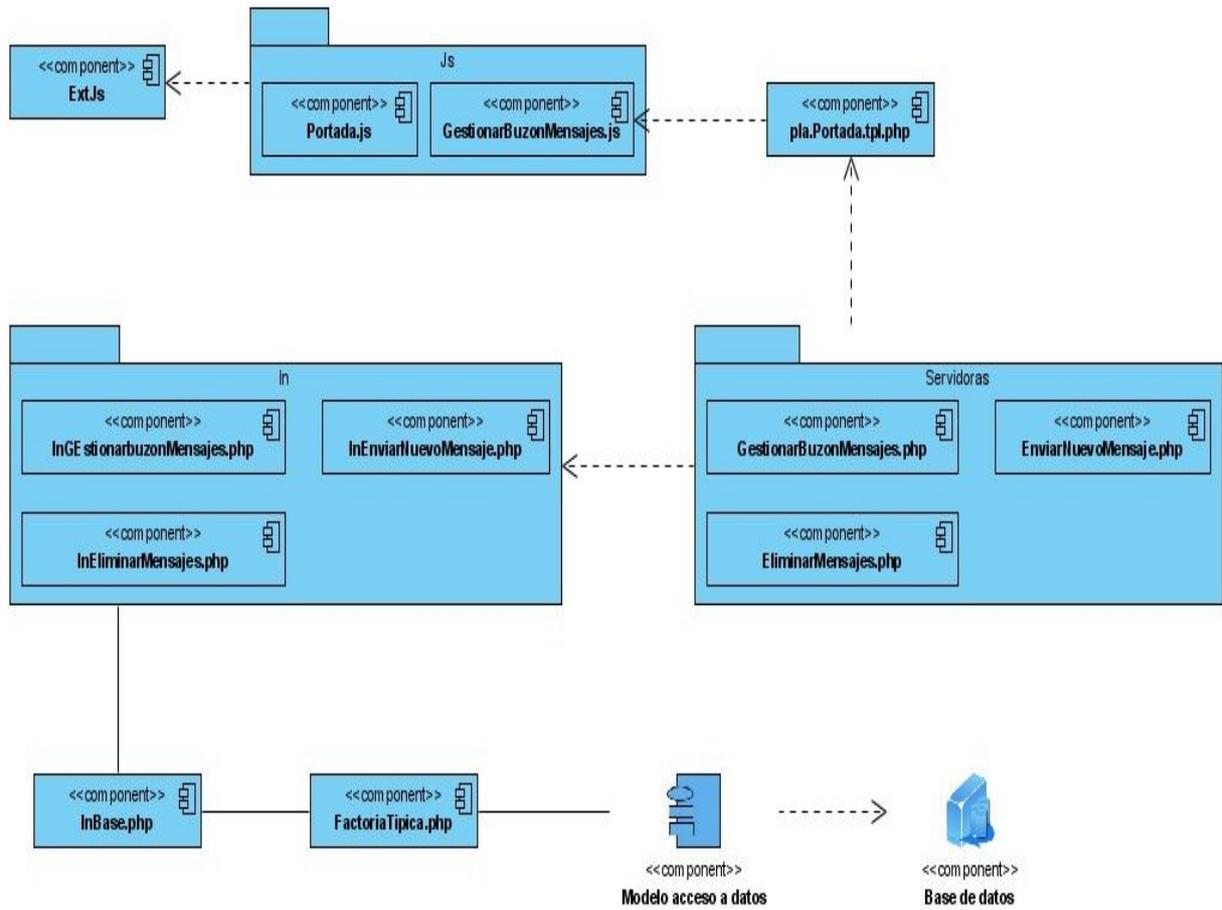
Nombre de la tabla: tprestaciones		
Descripción: Almacena las prestaciones que tiene un módulo de cómputo. Representan las funcionalidades de un módulo acoplado a la plataforma.		
Atributo	Tipo	Descripción
idprestacion	integer	Identificador único de la tabla tprestaciones
nombre	varchar(80)	Descripción detallada del estado de procesamiento de una solicitud
referencia	varchar(80)	Referencia a la página que carga la prestación.
idmodulo	integer	Identificador del módulo al que pertenece la prestación.

Cuadro 3.15: Tabla tprestaciones

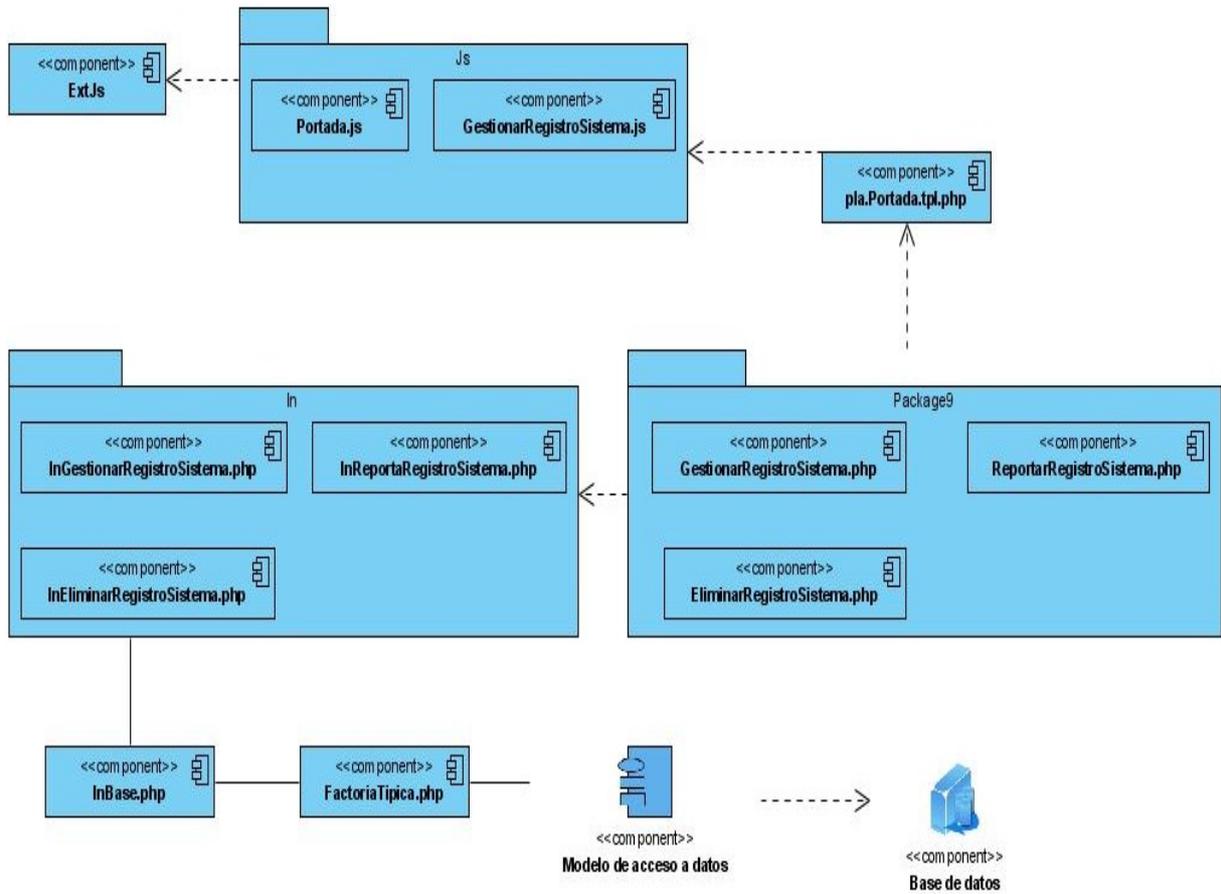
**Anexo VI: Diagramas de componentes.**



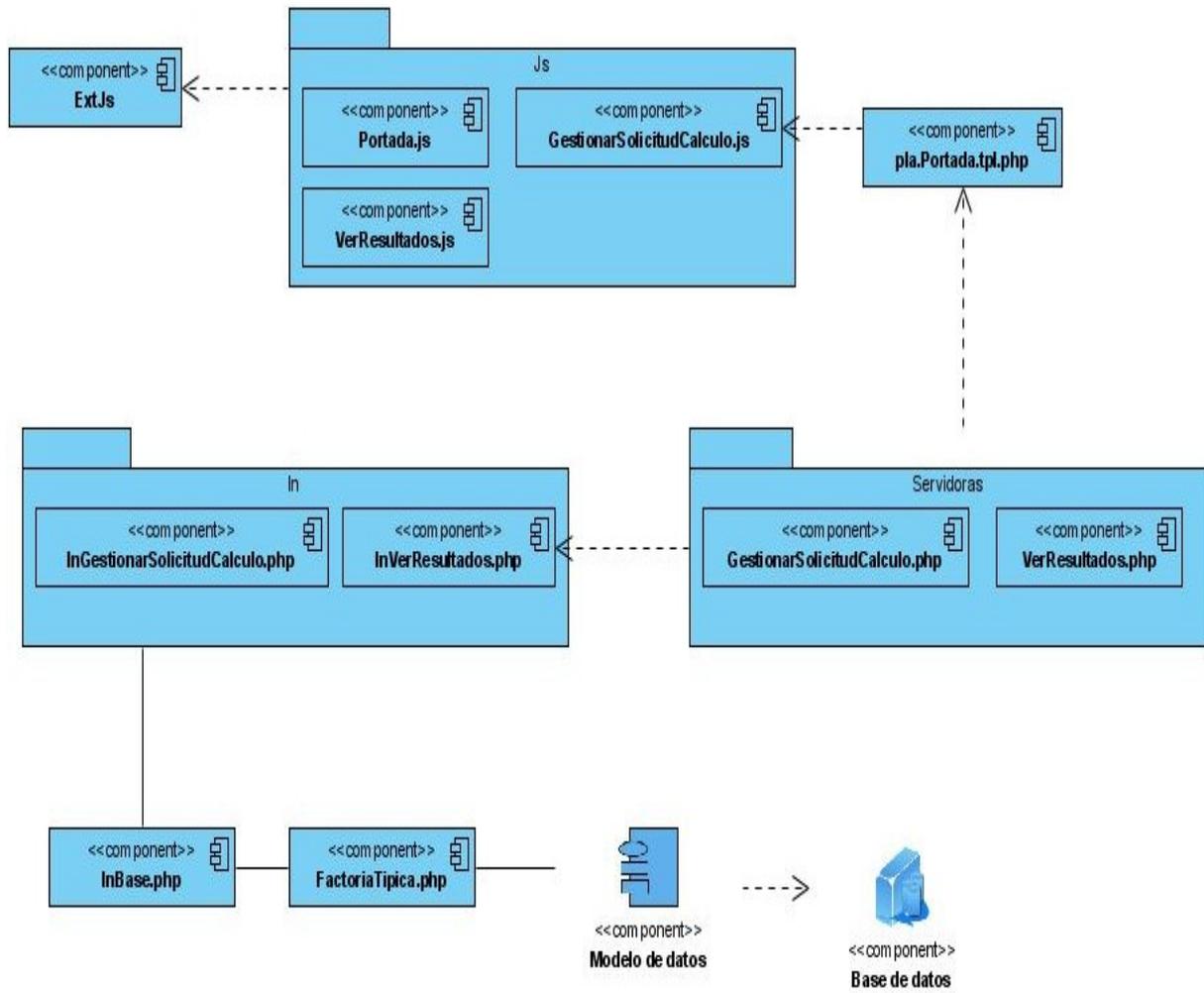
**Figura 1 Diagrama de componentes del caso de uso AutenticarUsuarios.**



**Figura 2 Diagrama de componentes del caso de uso GestionarBuzonMensajes**



**Figura 3 Diagrama de componentes para el caso de uso GestionarRegistroSistema.**



**Figura 4 Diagrama de componentes para el caso de uso GestionarSolicitudCalculo**

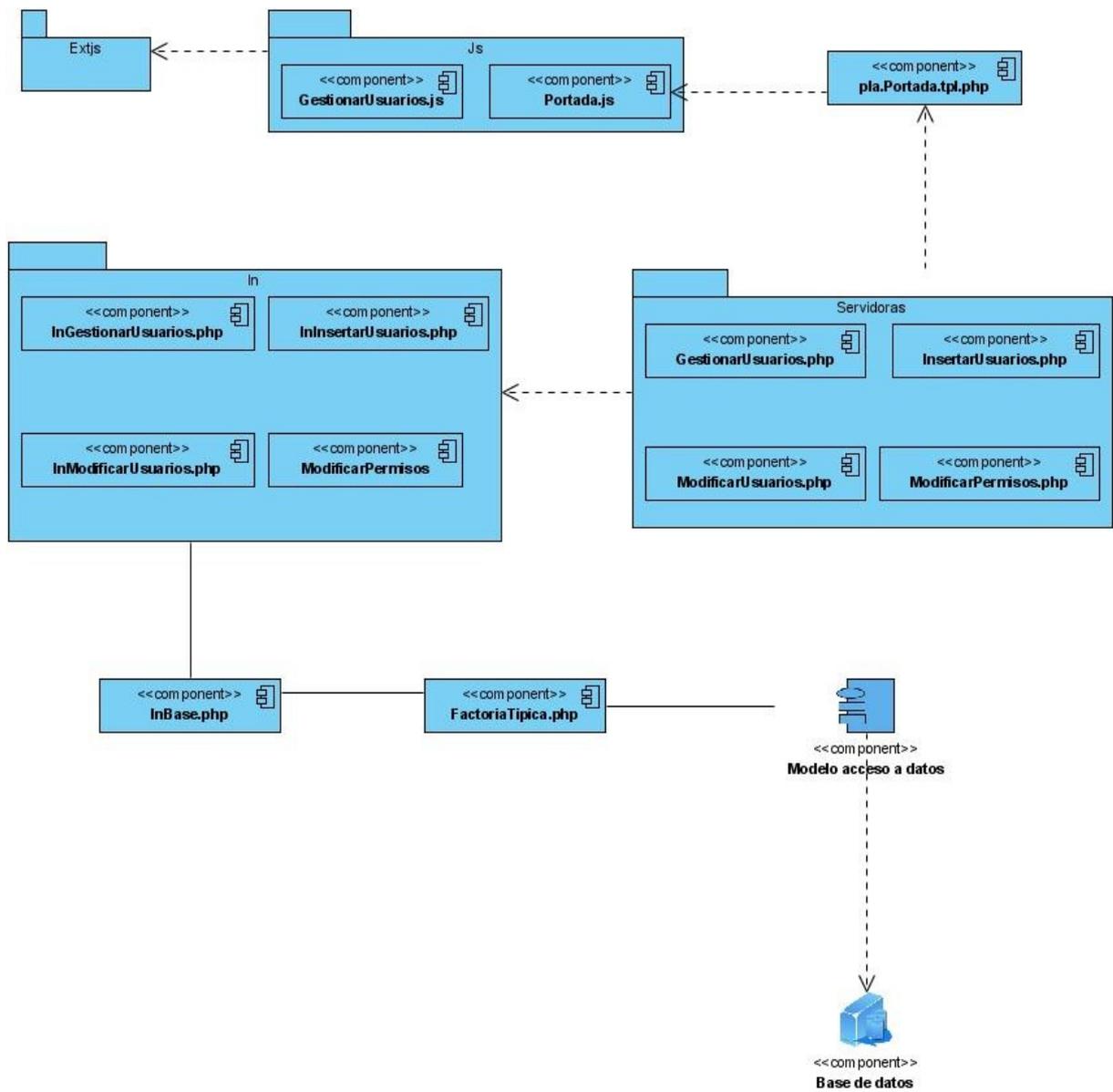


Figura 5 Diagrama de componentes para el caso de uso GestionarUsuarios

## ***Anexo VII: Descripción de los casos de prueba.***

Descripción general: El usuario queda autenticado.

**CP1: Nombre del CU AutenticarUsuarios.**

**CP1.1: Autenticar Usuario.**

**Tabla 2 Caso de Prueba para Autenticar Usuarios**

<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El usuario entra el usuario y la contraseña y presiona aceptar.	El usuario queda autenticado y el sistema le muestra su perfil dependiendo de los permisos que tiene.	El usuario y contraseña que entra son correctos.
El usuario entra el usuario y la contraseña y presiona aceptar.	El sistema muestra un mensaje en la interfaz indicándole que el usuario o la contraseña no corresponden con el correcto.	El usuario o la contraseña no coinciden con los correctos.
El usuario no completa los campos correspondientes y presiona aceptar.	El sistema muestra un mensaje de error solicitando el usuario o la contraseña para entrar al sistema.	El usuario no debe haber introducido el usuario o la contraseña dejando en blanco algún espacio o ambos del formulario.

Descripción general: Gestión de los usuarios y los permisos del sistema.

**CP2: Nombre del CU GestionarUsuarios.**

**CP2.1: Insertar Usuario.**

**Tabla 2 Caso de Prueba para Insertar Usuarios**

<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El administrador entra todos los datos del formulario y presiona enviar.	El usuario nuevo queda añadido y sus datos guardados en la base de datos.	Los datos de entrada del formulario son correctos.

El usuario entra todos los datos del formulario y presiona aceptar	El sistema muestra un mensaje en la interfaz indicándole que la entrada en algún campo esta incorrecta.	Los datos tienen errores de formato o de validación.
El usuario no completa los campos correspondientes y presiona aceptar.	El sistema muestra un mensaje de error solicitando al usuario que llene los campos del formulario.	El usuario no debe haber introducido algún campo dejando en blanco algún espacio del formulario.

Descripción general: Gestión de los usuarios y los permisos del sistema.

**CP2: Nombre del CU GestionarUsuarios.**

**CP2.2: Modificar Usuario.**

**Tabla 2 Caso de Prueba para Modificar Usuario**

<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El administrador entra todos los datos del formulario y presiona enviar.	El sistema muestra un mensaje en la interfaz indicándole que no ha introducido ningún cambio o modificación en el usuario forzándolo a hacerla o a dejar la interfaz de modificación.	Los datos de entrada del formulario son los mismos del que antes existía y no existe modificación alguna.
El usuario entra todos los datos del formulario y presiona aceptar	El sistema muestra un mensaje en la interfaz indicándole que la entrada en algún campo esta incorrecta.	Los datos tienen errores de formato o de validación.
El usuario no completa los campos correspondientes y presiona aceptar.	El sistema muestra un mensaje de error solicitando al usuario que llene los campos del formulario.	El usuario no debe haber introducido algún campo dejando en blanco algún espacio del formulario.

Descripción general: Gestión del Buzón de mensajes del sistema.

**CP3: Nombre del CU GestionarBuzonMensajes.**

**CP3.1: Enviar Nuevo Mensaje.**

**Tabla 3 Caso de Prueba para Enviar Nuevo Mensaje.**

<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El usuario entra todos los datos de la interfaz de nuevo mensaje y presiona enviar.	El sistema muestra un mensaje en la interfaz indicándole que el mensaje ha sido enviado correctamente	Los datos de entrada del formulario están correctamente escritos.
El usuario entra todos los datos de la interfaz de nuevo mensaje y presiona enviar.	El sistema muestra un mensaje en la interfaz indicándole que la entrada en algún campo esta incorrecta.	Los datos tienen errores de formato o de validación.
El usuario selecciona enviar.	El sistema muestra un mensaje de error solicitando al usuario que llene los campos del formulario.	El usuario no debe haber introducido algún campo dejando en blanco algún espacio del formulario que es obligatorio.

Descripción general: Gestión del Buzón de mensajes del sistema.

**CP3: Nombre del CU GestionarBuzonMensajes.**

**CP3.2: Eliminar Mensaje.**

**Tabla 4 Caso de Prueba para Eliminar Mensaje.**

<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El usuario selecciona un mensaje y presiona eliminar.	El sistema muestra un mensaje en la interfaz indicándole que el mensaje ha sido eliminado correctamente	El usuario tiene seleccionado un mensaje o varios.

El usuario selecciona eliminar	El sistema muestra un mensaje en la interfaz indicándole que debe seleccionar un mensaje al menos	El usuario no tiene ningún mensaje previamente seleccionado.
--------------------------------	---	--

Descripción general: Gestión de los registros del sistema.

**CP4: Nombre del CU GestionarRegistroSistema.**

**CP4.1: Reportar Registro Sistema.**

**Tabla 5 Caso de Prueba para Reportar Registro Sistema.**

<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El usuario selecciona uno o más registros y presiona reportar.	El sistema muestra un mensaje en la interfaz indicándole que el mensaje ha sido reportado correctamente	El usuario tiene seleccionado uno varios registros previamente.
El usuario selecciona reportar.	El sistema muestra un mensaje en la interfaz indicándole que debe seleccionar un registro al menos	El usuario no tiene ningún registro previamente seleccionado.

Descripción general: Gestión de los registros del sistema.

**CP4: Nombre del CU GestionarRegistroSistema.**

**CP4.2: Eliminar Registro Sistema.**

**Tabla 5 Caso de Prueba para Eliminar Registro Sistema.**

<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El usuario selecciona uno o más registros y presiona eliminar	El sistema muestra un mensaje en la interfaz indicándole que el mensaje ha sido eliminado correctamente	El usuario tiene seleccionado uno varios registros previamente.
El usuario selecciona eliminar.	El sistema muestra un mensaje en la interfaz indicándole que debe seleccionar un registro al menos	El usuario no tiene ningún registro previamente seleccionado.

Descripción general: Gestión de las solicitudes de cálculo.

**CP5: Nombre del CU GestionarSolicitudCalculo.**

**CP5.1: Ver Resultados.**

**Tabla 6 Caso de Prueba para Ver Resultados.**

<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El usuario selecciona una solicitud de cálculo y presiona ver resultados.	El sistema muestra la interfaz correspondiente.	El usuario tiene seleccionada una solicitud de cálculo previamente.
El usuario selecciona una solicitud de cálculo y presiona ver resultados.	El sistema muestra un mensaje en la interfaz indicándole que la solicitud de cálculo esta todavía sin procesar o en procesamiento.	El usuario tiene seleccionada una solicitud de cálculo que esta en procesamiento o sin procesar.
El usuario presiona ver resultados.	El sistema muestra un mensaje en la interfaz indicándole que debe seleccionar una solicitud.	El usuario no tiene seleccionada una solicitud de cálculo previamente.

Descripción general: Gestión de las solicitudes de cálculo.

**CP5: Nombre del CU GestionarSolicitudCalculo.**

**CP5.2: Eliminar Solicitud Cálculo.**

**Tabla 6 Caso de Prueba para Eliminar Solicitud Cálculo**

<b>Entrada</b>	<b>Resultados</b>	<b>Condiciones</b>
El usuario selecciona una solicitud de cálculo y presiona eliminar.	El sistema muestra una confirmación de que fue eliminada correctamente.	El usuario tiene seleccionada una solicitud de cálculo previamente.
El usuario presiona eliminar.	El sistema muestra un mensaje en la interfaz indicándole que debe seleccionar una solicitud.	El usuario no tiene seleccionada una solicitud de cálculo previamente.

# Glosario de Términos:

**APXS:** Es un script escrito en Perl que ayuda a compilar el código fuente de algunos módulos para convertirlos en Objetos Dinámicos Compartidos.

**BSD:** Berkeley Software Distribution, de la Universidad de California en Berkeley, autora de una versión de UNIX que lleva su mismo nombre.

**GNU/Linux:** Nombre por el que se conoce al sistema operativo formado por el conjunto de utilidades de sistema GNU y el núcleo Linux más otras aplicaciones.

**CGI:** Acrónimo de Common Gateway Interface. Interface de Acceso Común. Programas usados para hacer llamadas a rutinas o controlar otros programas o bases de datos.

**Perl:** Lenguaje de programación utilizado en el World Wide Web a través de un CGI, principalmente para realizar consultas a bases de datos como Oracle.

**ActiveX:** Lenguaje desarrollado por Microsoft para la elaboración de aplicaciones exportables a la red y capaces de operar sobre C.

**JSON:** Acrónimo de "JavaScript Object Notation", es un formato ligero para el intercambio de datos. JSON es un subconjunto de la notación literal.

**Applets:** Aplicación escrita en JAVA y compilada.

**PDA:** Híbrido entre los ordenadores portátiles y las agendas electrónicas que, con la incorporación de Internet, podrían plagar los bolsillos de la gente.

**JScript:** Implementación de JavaScript por parte de Microsoft, compatible con el JavaScript de Netscape Navigator 2.0.

**IDE:** Integrated Drive Electronics. Interfaz de transmisión de datos en bus dentro de equipos informáticos.

**SQL:** Structured Query Language. Es un lenguaje de programación interactivo y estandarizado para extraer información y actualizar una base de datos. ...