

# **Una Arquitectura de Sistemas de Información para vigilancia en salud pública en zonas rurales**



## **Anexo A Análisis y Diseño de la herramienta para Vigilancia Epidemiológica-SIVE**

**Ing. Diego Mauricio López Gutiérrez**

Director  
Dr. Ing. Álvaro Rendón Gallón

*Universidad del Cauca*

Maestría en Ingeniería, Área Telemática  
**Facultad de Ingeniería Electrónica y Telecomunicaciones**  
**Departamento de Telemática**  
Línea de investigación en Ingeniería de Sistemas Telemáticos  
Aplicaciones soportadas en Internet

Popayán, Diciembre de 2005



# Índice General

<b>A.1</b>	<b>Introducción.....</b>	<b>5</b>
<b>A.2</b>	<b>Modelo de análisis .....</b>	<b>5</b>
A.2.1	Realización de casos de uso. ....	5
<b>A.3</b>	<b>Modelo de Diseño .....</b>	<b>17</b>
A.3.1	Diagramas de clases de diseño .....	17
A.3.2	Diagrama general de clases de Diseño .....	24
A.3.3	Diagrama de paquetes de diseño .....	25
A.3.4	Diagrama de implantación .....	25
<b>A.4</b>	<b>Modelo de Implementación .....</b>	<b>26</b>
A.4.1	Descripción de los Componentes .....	26
<b>A.5</b>	<b>Referencias.....</b>	<b>34</b>

# Índice de Figuras

<i>Figura A. 1 Diagrama de clases Caso de uso Presentar formulario.</i>	6
<i>Figura A. 2 .Diagrama de secuencia caso de uso Presentar formulario.</i>	6
<i>Figura A. 3 . Diagrama de clases caso de uso Registrar evento epidemiológico.</i>	7
<i>Figura A.4. Diagrama de secuencia caso de uso Registrar evento epidemiológico.</i>	7
<i>Figura A. 5. Diagrama de clases casos de uso Buscar Registros Enviados.</i>	8
<i>Figura A. 6. Diagrama de secuencia caso de uso Buscar Registros Enviados</i>	9
<i>Figura A. 7. Diagrama de clases caso de uso Actualizar Base De Datos.</i>	10
<i>Figura A. 8. Diagrama de secuencia caso de uso Actualizar Base De Datos.</i>	11
<i>Figura A. 9. Diagrama de clases caso de uso Consultar Consolidado SIVIGILA.</i>	12
<i>Figura A. 10. Diagrama de secuencia caso de uso Consultar Consolidado SIVIGILA.</i>	12
<i>Figura A. 11. Diagrama de clases caso de uso enviar formulario.</i>	13
<i>Figura A. 12. Diagrama de secuencia caso de uso enviar formulario.</i>	14
<i>Figura A. 13. Diagrama de clases caso de uso Crear nuevo formulario.</i>	15
<i>Figura A. 14. Diagrama de secuencia caso de uso Crear nuevo formulario.</i>	16
<i>Figura A. 15. Diagrama de clases caso de uso identificarse en el sistema.</i>	16
<i>Figura A. 16. Diagrama de secuencia caso de uso identificarse en el sistema.</i>	17
<i>Figura A. 17. Diagrama de clases de diseño Caso de uso Presentar formulario.</i>	17
<i>Figura A. 18. Diagrama de clases de diseño Caso de uso Registrar evento epidemiológico.</i>	18
<i>Figura A. 19. Diagrama de clases de diseño Caso de uso Buscar registros enviados.</i>	19
<i>Figura A. 20. Diagrama de clases de diseño Caso de uso Actualizar base de datos</i>	20
<i>Figura A. 21. Diagrama de clases de diseño Caso de uso Consultar consolidado SIVIGILA.</i>	21
<i>Figura A. 22. Diagrama de clases de diseño Caso de uso enviar formularios.</i>	21
<i>Figura A. 23. Diagrama de clases de diseño Caso de uso crear nuevo formulario.</i>	22
<i>Figura A. 24. Diagrama de clases de diseño Caso de uso identificarse en el sistema.</i>	23
<i>Figura A. 25. Diagrama general de clases de diseño</i>	24
<i>Figura A. 26. Diagrama general de paquetes de diseño</i>	25
<i>Figura A. 27. Diagrama de Implantación</i>	26
<i>Figura A.28. Diagrama de componentes de la aplicación</i>	27
<i>Figura A. 29. Componente de recolección de información</i>	28
<i>Figura A. 30. Componente de Notificación de eventos</i>	29
<i>Figura A. 31. Componente de consolidación de información.</i>	30
<i>Figura A. 32. Componente XML de conversión.</i>	31
<i>Figura A. 33. Componente de sincronización.</i>	32
<i>Figura A. 34. Componente de consolidación</i>	33
<i>Figura A. 35. Componente Gestor de Metadatos.</i>	34

## **A.1 Introducción**

Este anexo contiene la documentación generada en el desarrollo de la aplicación para vigilancia epidemiológica utilizada como entorno de validación de la arquitectura propuesta en esta tesis de maestría.

Esta aplicación ha sido desarrollada en múltiples fases. Una primera versión fue desarrollada para el proyecto de grado de la facultad de ingeniería electrónica y telecomunicaciones denominado: “Modelo para la definición, recolección y entrega de información en sistemas de información de vigilancia epidemiológica”[1]. Posteriormente este prototipo fue mejorado y sirvió como base para la herramienta de recolección de información epidemiológica en el proyecto SIGSA: Sistema de Información Geográfica para Salud financiado por la Dirección Departamental de Salud del Cauca. Posteriormente, fue rediseñada en el marco del proyecto EIA-SVSP “Entorno Integrado de Información y Aprendizaje para el Sistema de Vigilancia en Salud Pública del Departamento del Cauca” financiado por la Vicerrectoría de Investigaciones de la Universidad del Cauca. Finalmente, la versión más reciente de la aplicación ha sido desarrollada y está siendo implantada en el marco del proyecto EHAS-@LIS, financiado por la Unión Europea.

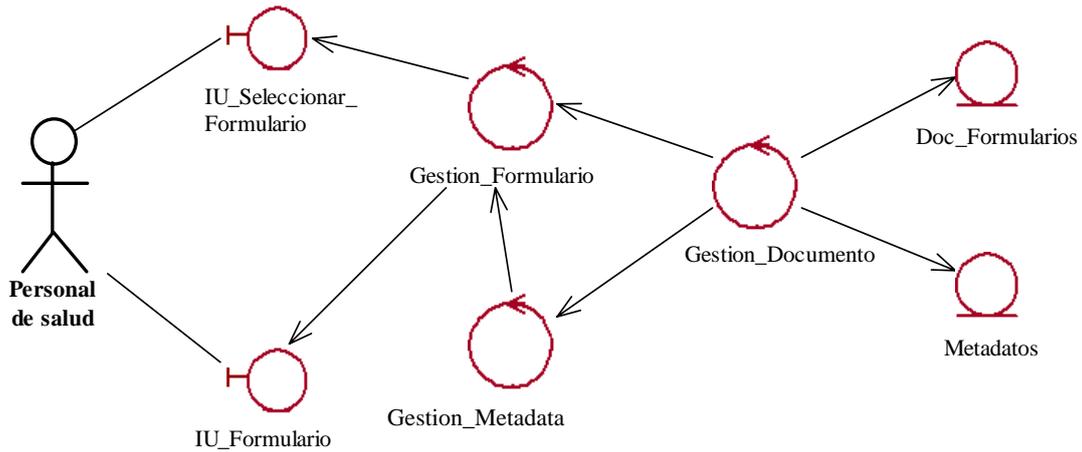
La documentación que se presenta en este anexo corresponde a la documentación generada en el marco del proyecto EIA-SVSP y se organiza según los flujos de trabajo de RUP descritos en la metodología del Capítulo 3. El modelo de la organización y de Casos de Uso de la aplicación no se incluye en este anexo debido a que se desarrolló en el Capítulo 5 de esta tesis. A continuación se presenta el modelo de análisis del sistema, seguido del modelo de diseño y finalmente el modelo de implementación.

## **A.2 Modelo de análisis**

### **A.2.1 Realización de casos de uso.**

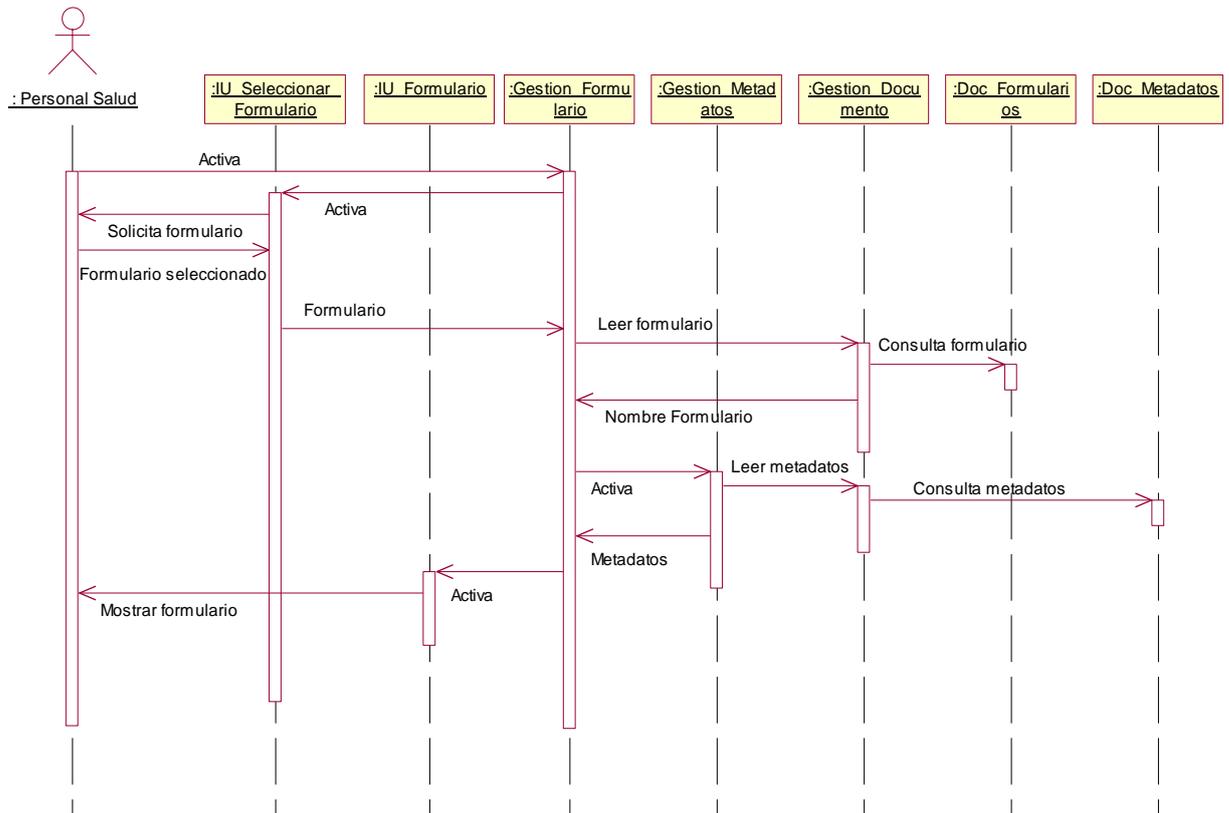
#### **A.2.1.1 Caso de uso Presentar formulario.**

**Diagrama de clases de análisis.**



**Figura A. 1** Diagrama de clases Caso de uso Presentar formulario.

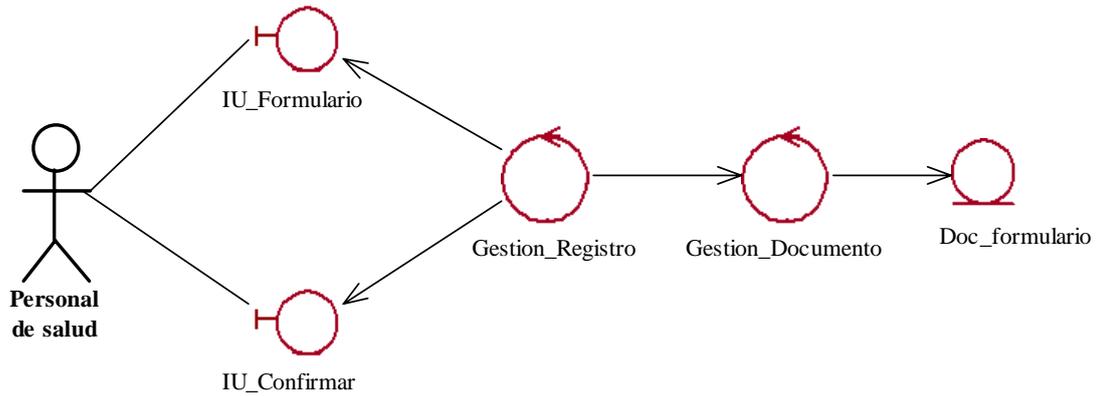
**Diagrama de secuencia.**



**Figura A. 2** .Diagrama de secuencia caso de uso Presentar formulario.

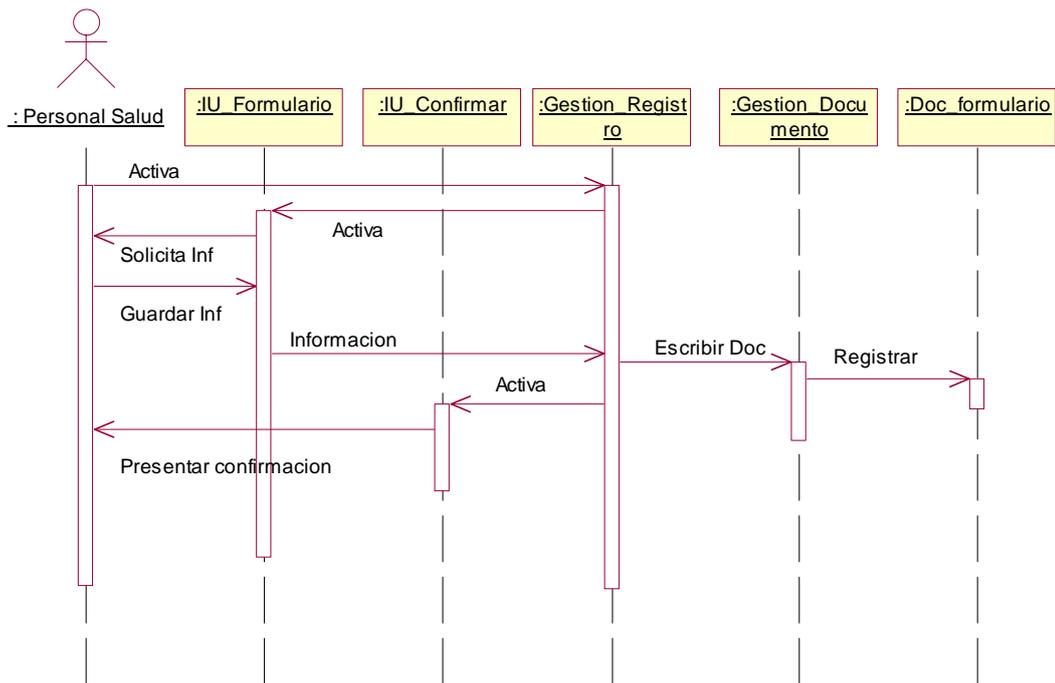
**A.2.1.2 Caso de uso Registrar evento epidemiológico.**

**Diagrama de clases de análisis.**



**Figura A. 3 .** Diagrama de clases caso de uso Registrar evento epidemiológico.

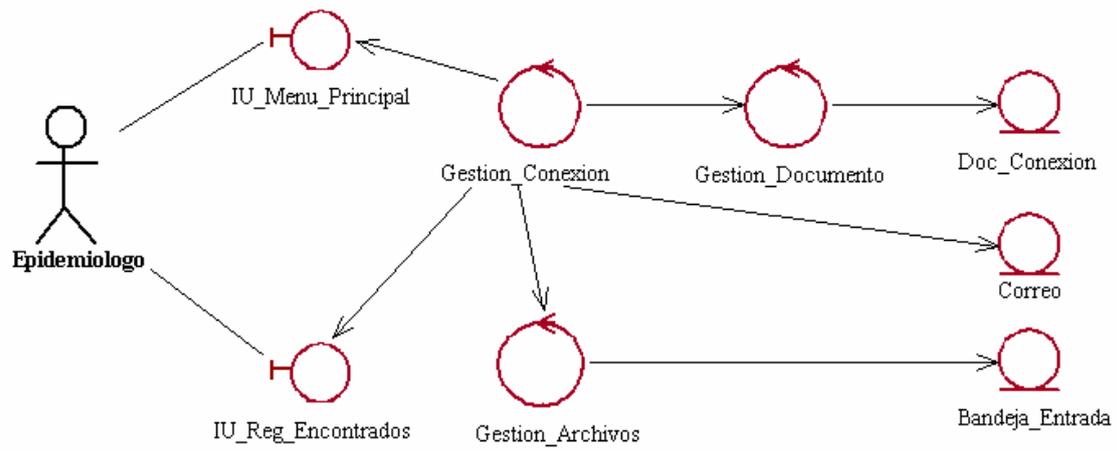
**Diagrama de secuencia.**



**Figura A.4.** Diagrama de secuencia caso de uso Registrar evento epidemiológico.

**A.2.1.3 Caso de uso Buscar Registros Enviados**

**Diagrama de clases de análisis.**



**Figura A. 5.** Diagrama de clases casos de uso Buscar Registros Enviados

## Diagrama de secuencia.

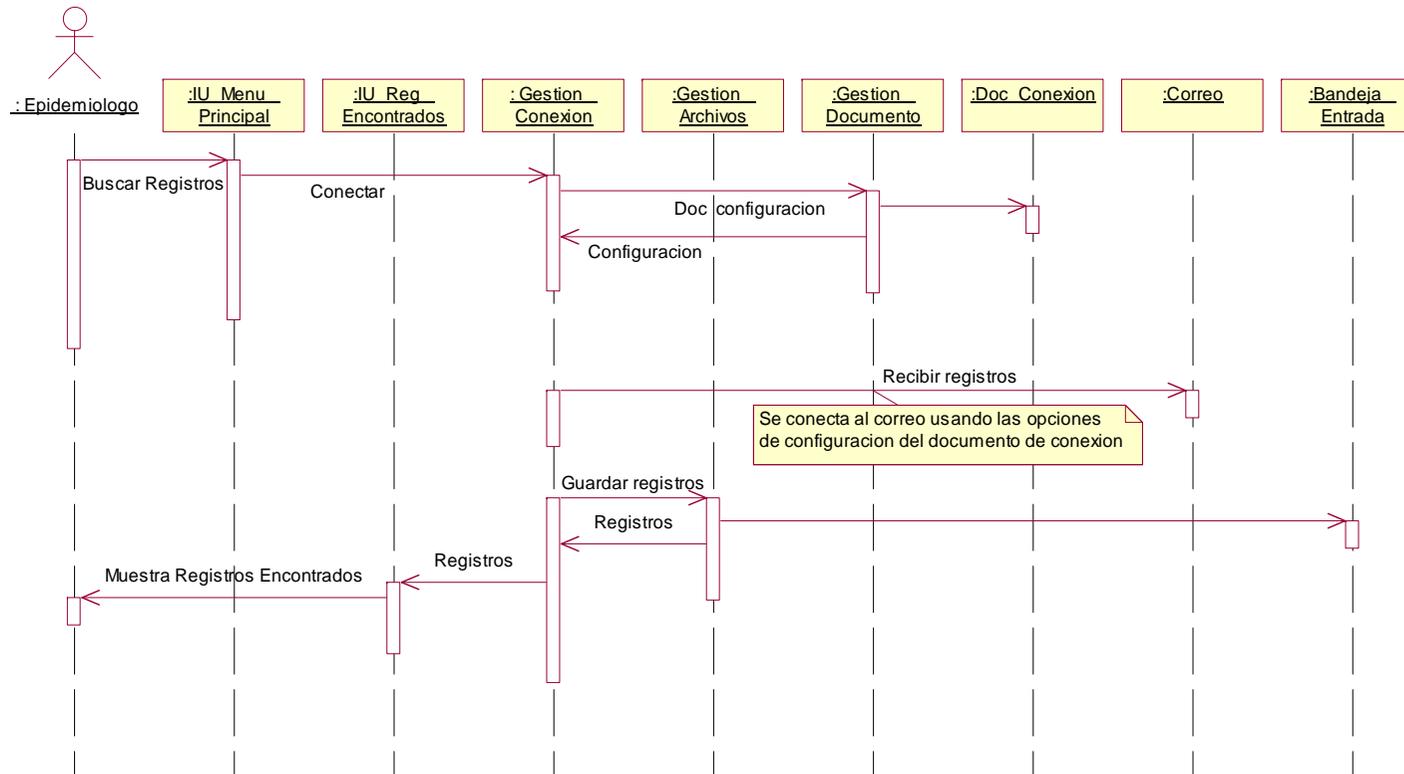
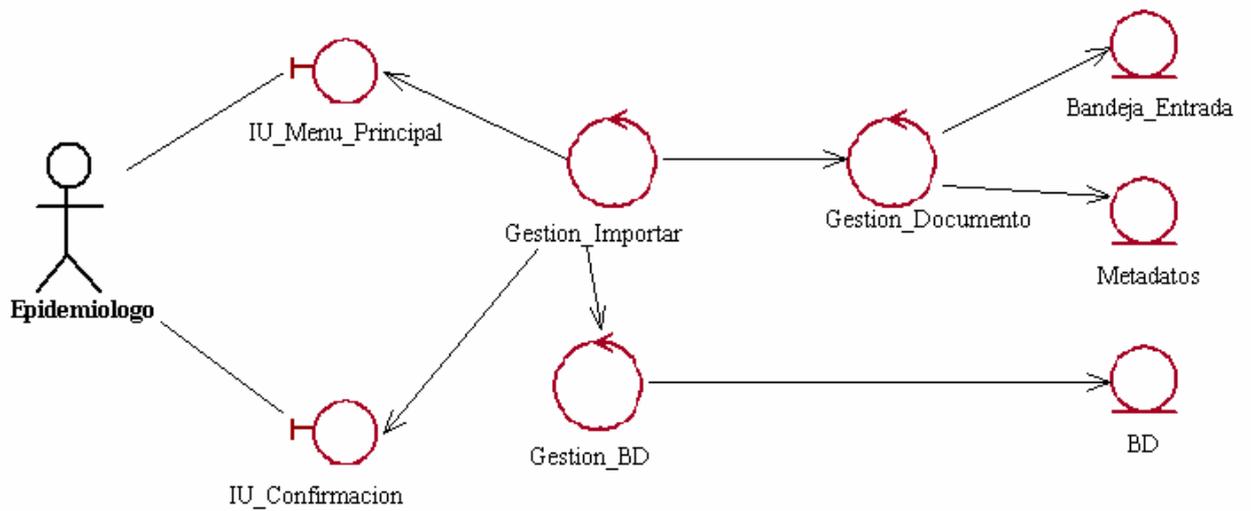


Figura A. 6. Diagrama de secuencia caso de uso Buscar Registros Enviados

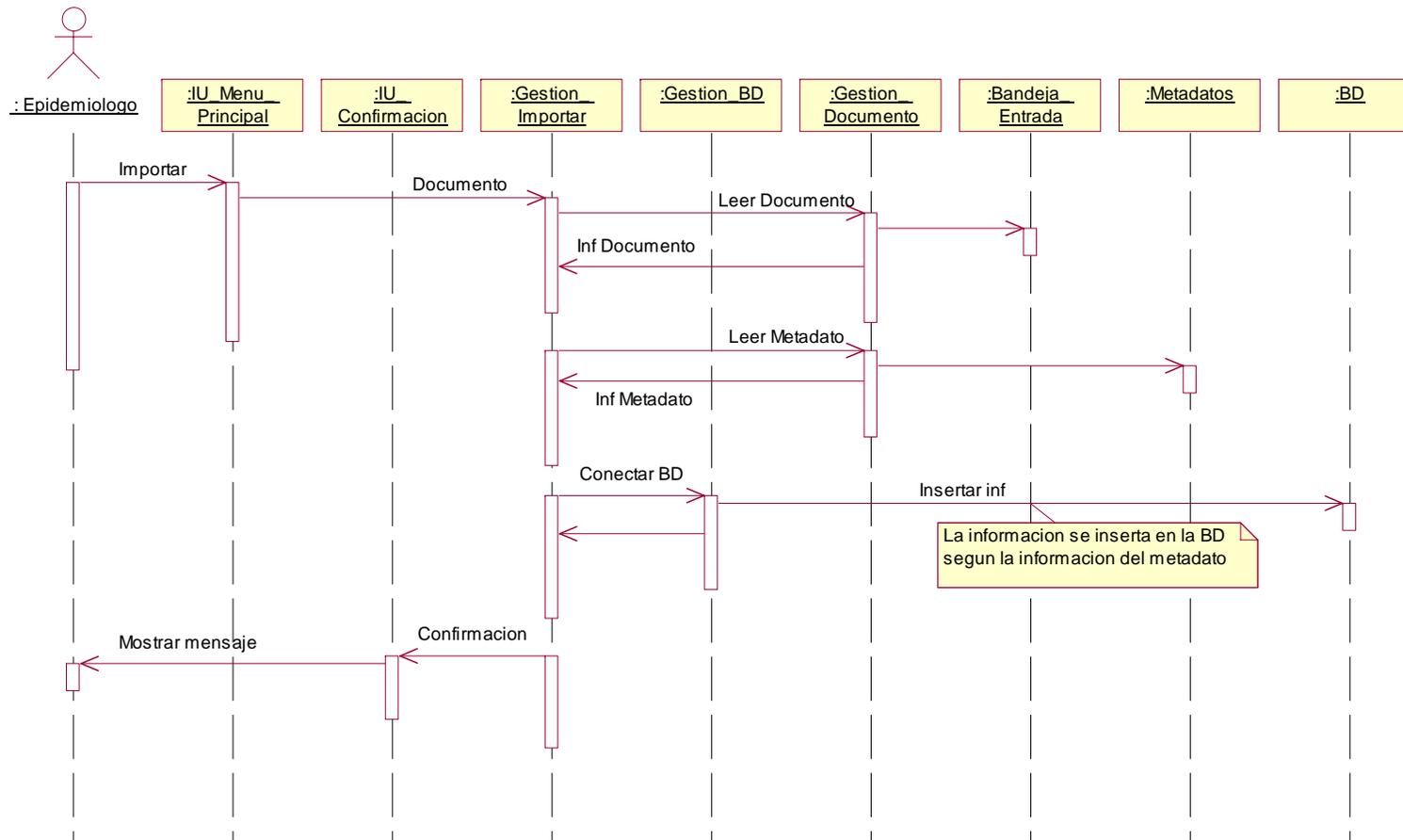
#### A.2.1.4 Caso de uso Actualizar Base De Datos.

Diagrama de clases de análisis.



**Figura A. 7.** Diagrama de clases caso de uso Actualizar Base De Datos.

**Diagrama de secuencia.**



**Figura A. 8.** Diagrama de secuencia caso de uso Actualizar Base De Datos.

### A.2.1.5 Caso de uso Consultar Consolidado SIVIGILA.

Diagrama de clases de análisis.

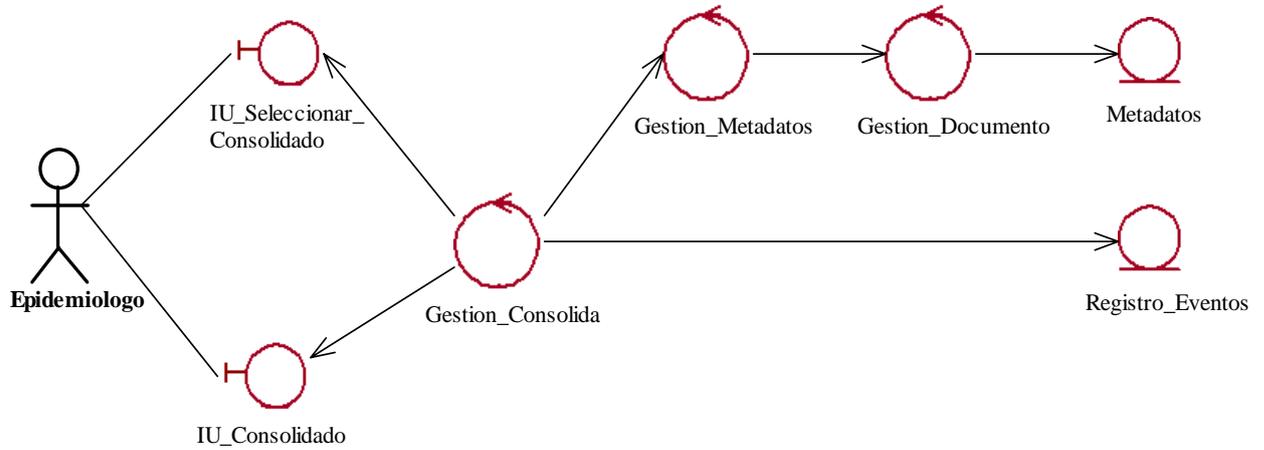


Figura A. 9. Diagrama de clases caso de uso Consultar Consolidado SIVIGILA.

Diagrama de secuencia.

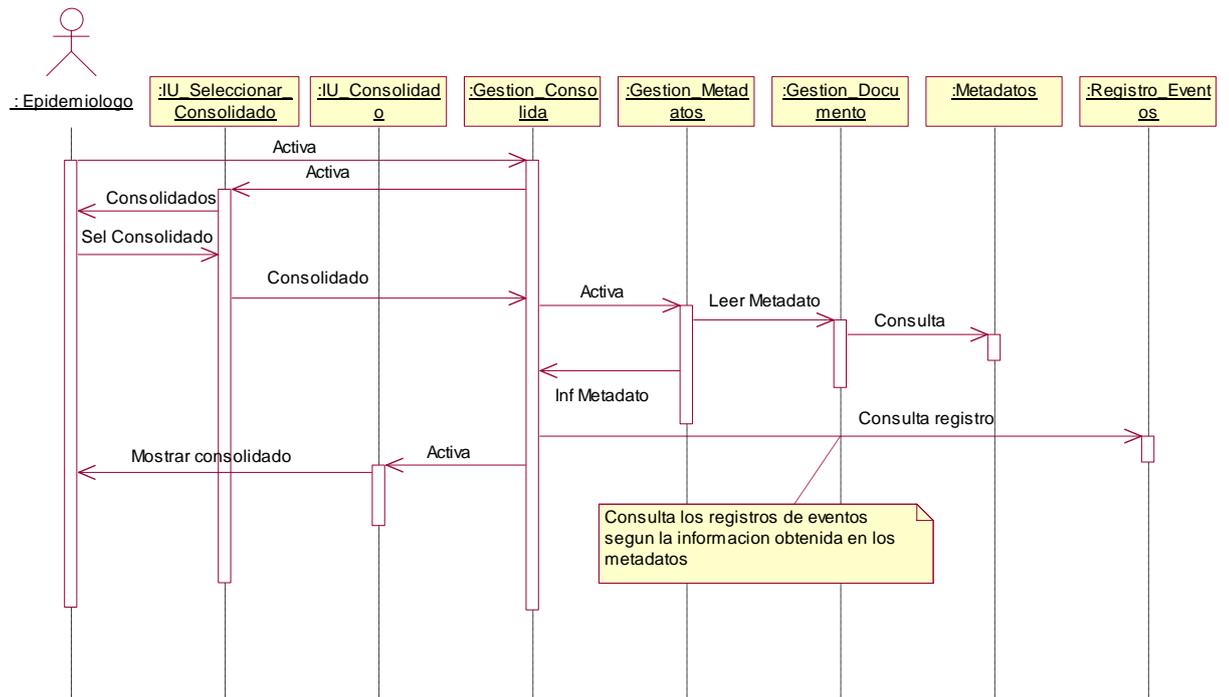
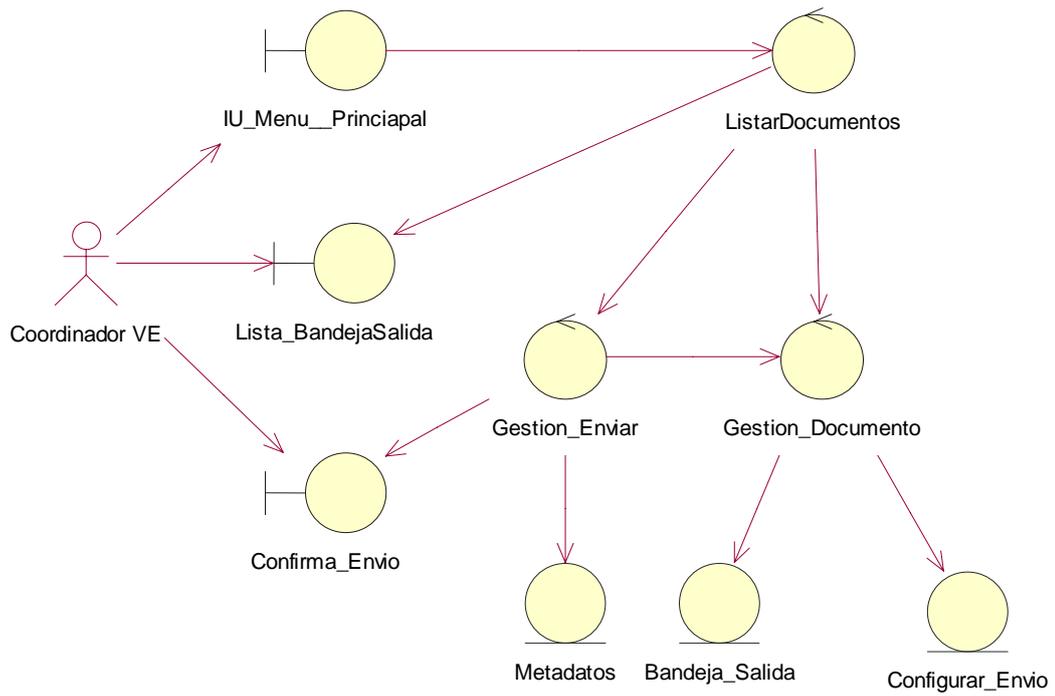


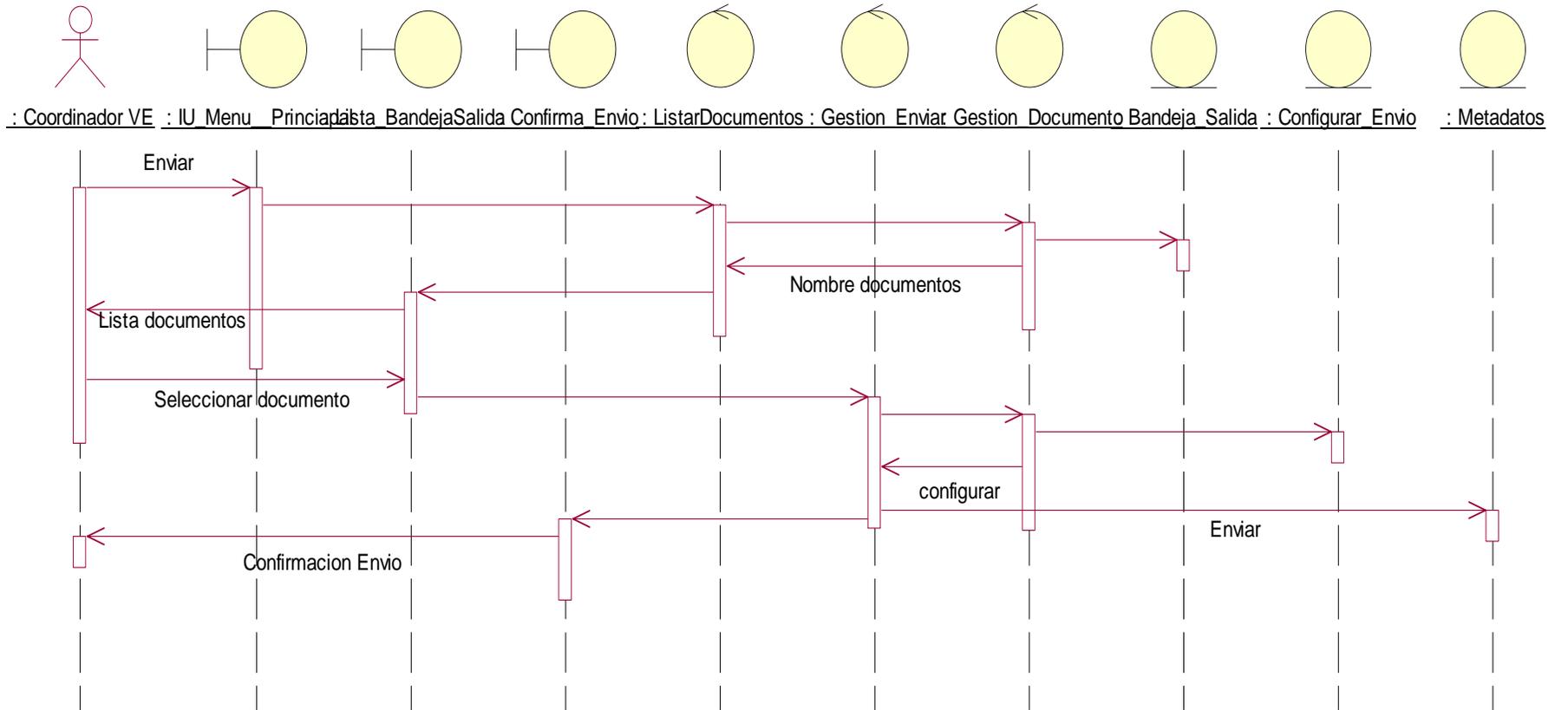
Figura A. 10. Diagrama de secuencia caso de uso Consultar Consolidado SIVIGILA.

### A.2.1.6 Caso de uso enviar formulario.

Diagrama de clases de análisis.



**Figura A. 11.** Diagrama de clases caso de uso enviar formulario.



**Figura A. 12.** Diagrama de secuencia caso de uso enviar formulario.

### A.2.1.7 Caso de uso Crear nuevo formulario.

#### Diagrama de clases de análisis.

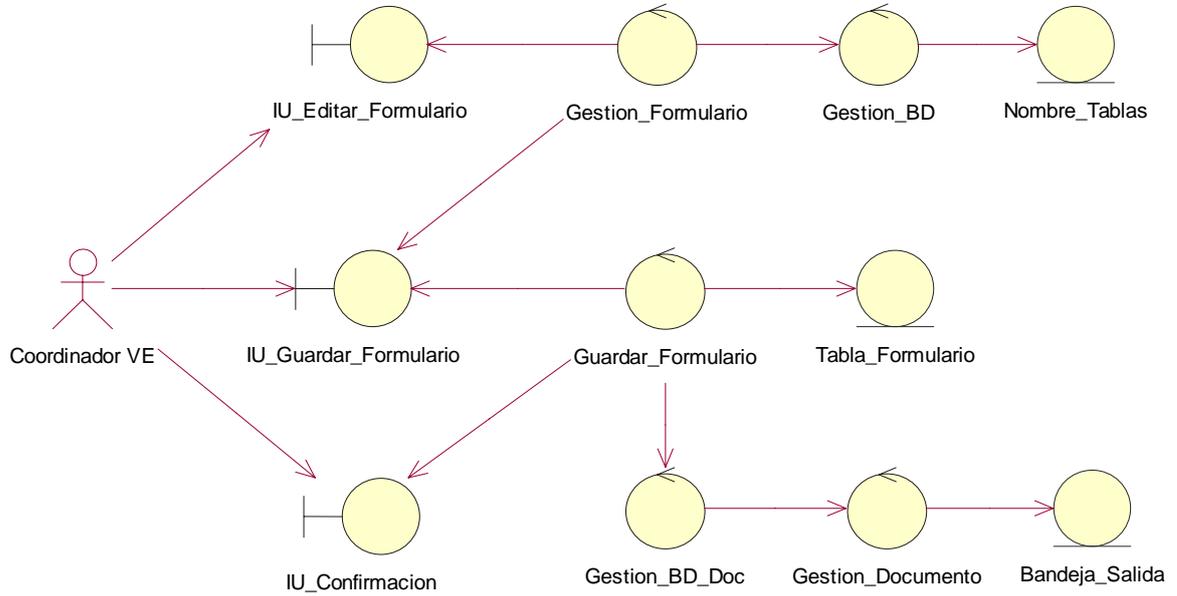
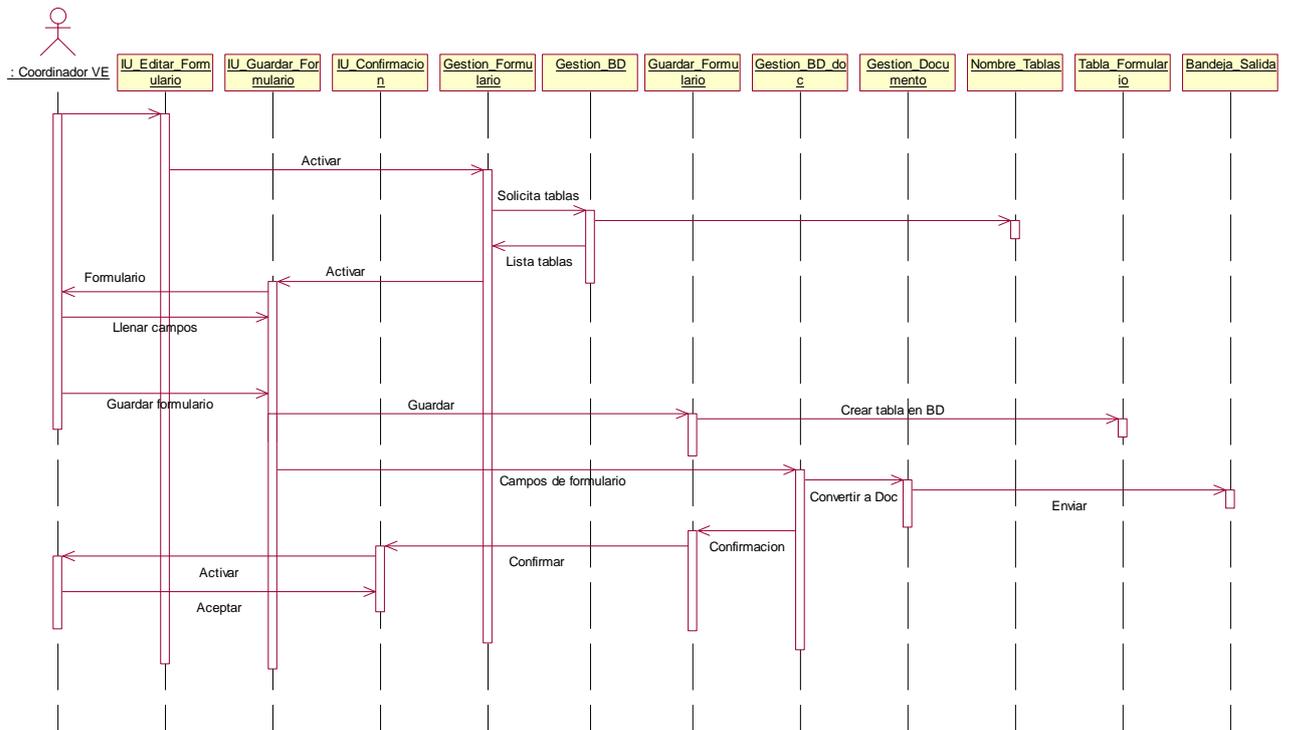


Figura A. 13. Diagrama de clases caso de uso Crear nuevo formulario.

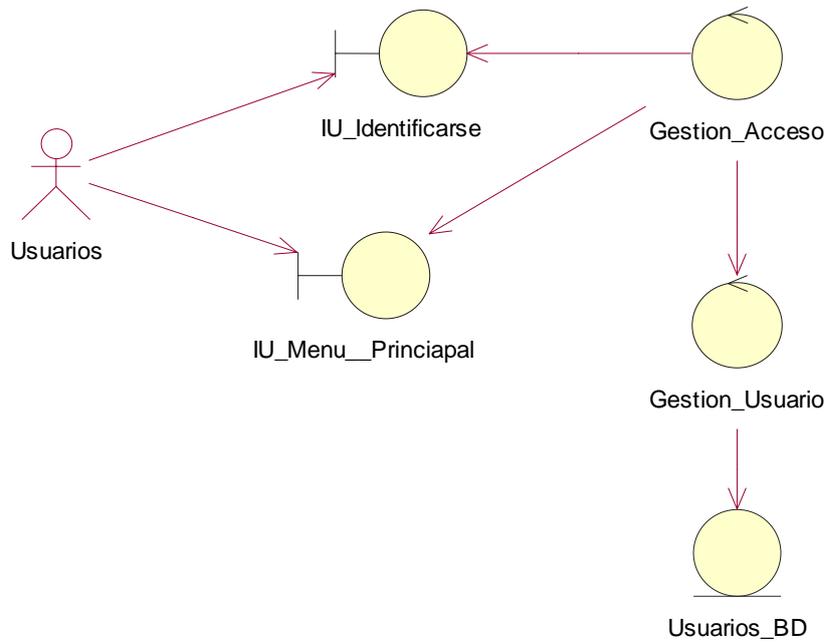
#### Diagrama de secuencia



**Figura A. 14.** Diagrama de secuencia caso de uso Crear nuevo formulario.

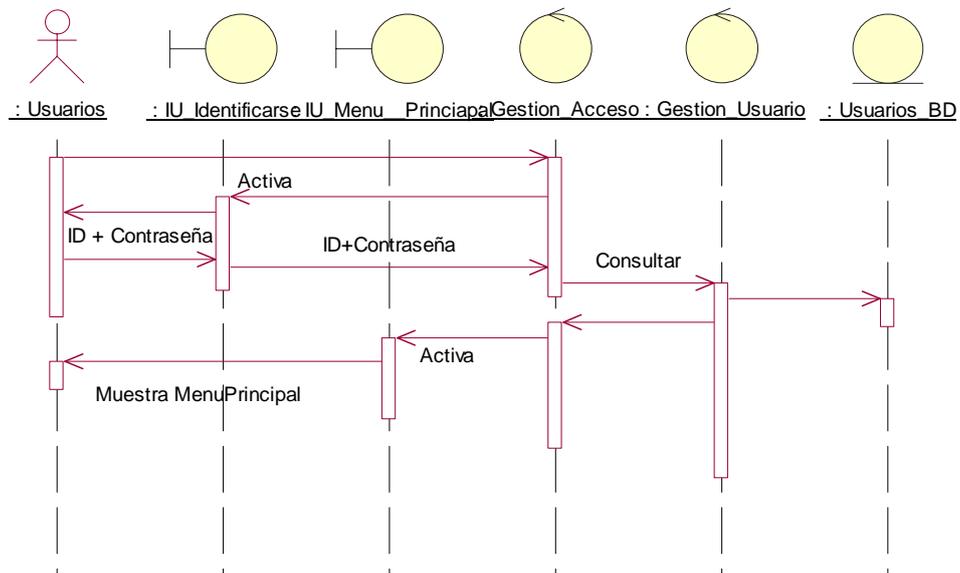
**A.2.1.8 Caso de uso identificarse en el sistema.**

**Diagrama de clases de análisis.**



**Figura A. 15.** Diagrama de clases caso de uso identificarse en el sistema.

**Diagrama de Secuencia.**



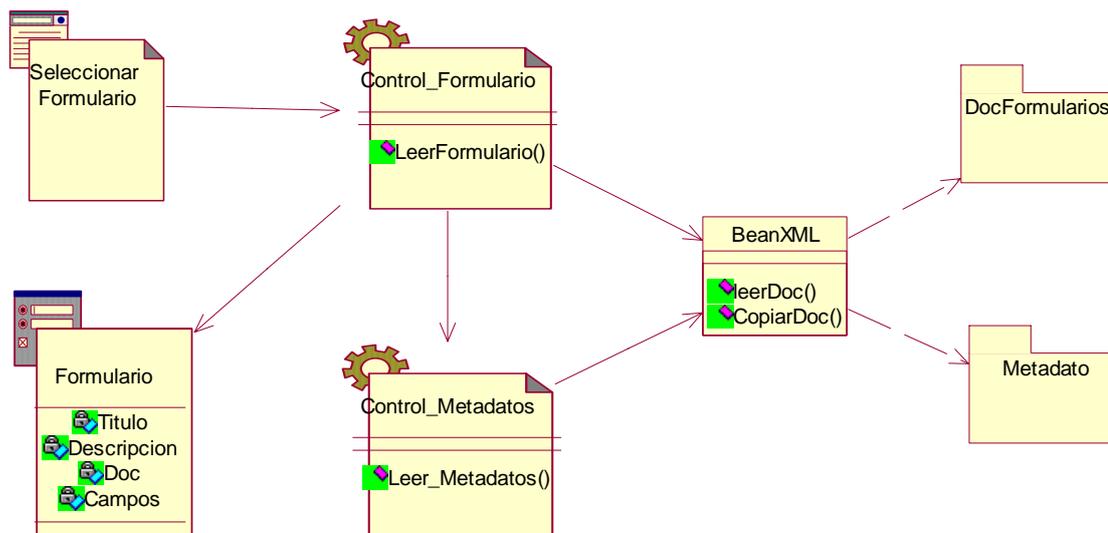
**Figura A. 16.** Diagrama de secuencia caso de uso identificarse en el sistema.

### A.3 Modelo de Diseño

A continuación se presentan las clases de diseño realizadas para cada caso de uso.

#### A.3.1 Diagramas de clases de diseño

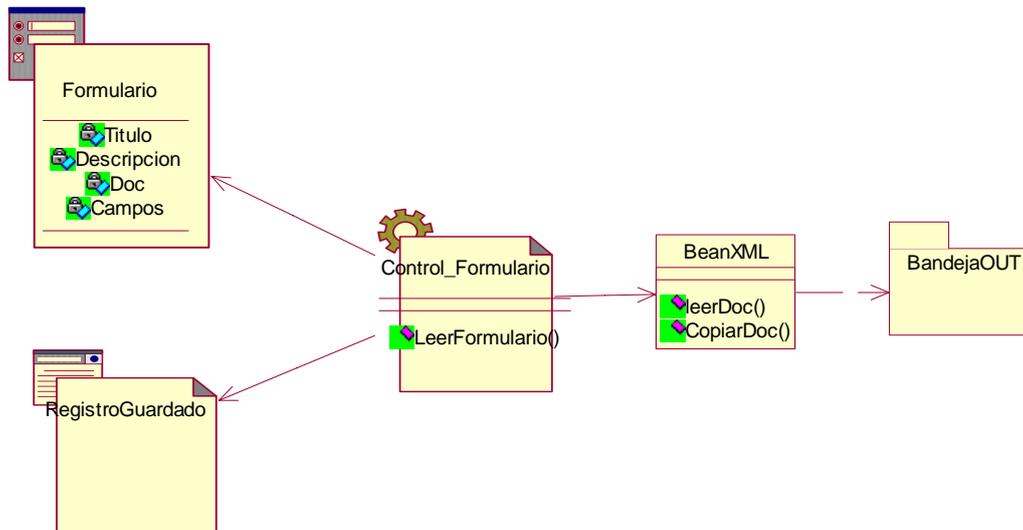
##### A.3.1.1 Caso de uso Presentar formulario.



**Figura A. 17.** Diagrama de clases de diseño Caso de uso Presentar formulario.

Este caso de uso se presenta en la aplicación de escritorio del cliente, genera y despliega un formulario de recolección de información epidemiológica de forma dinámica, inicia cuando se selecciona un formulario a presentar desde la ventana IU\_Seleccionar\_Formulario. Las clases de control vistas en el modelo de análisis son implementadas como Beans en Java, los cuales se encargan de interpretar los documentos de formularios y metadatos para la presentación del formulario. Las clases de entidad: Doc\_Formularios y Metadatos son documentos XML almacenados localmente.

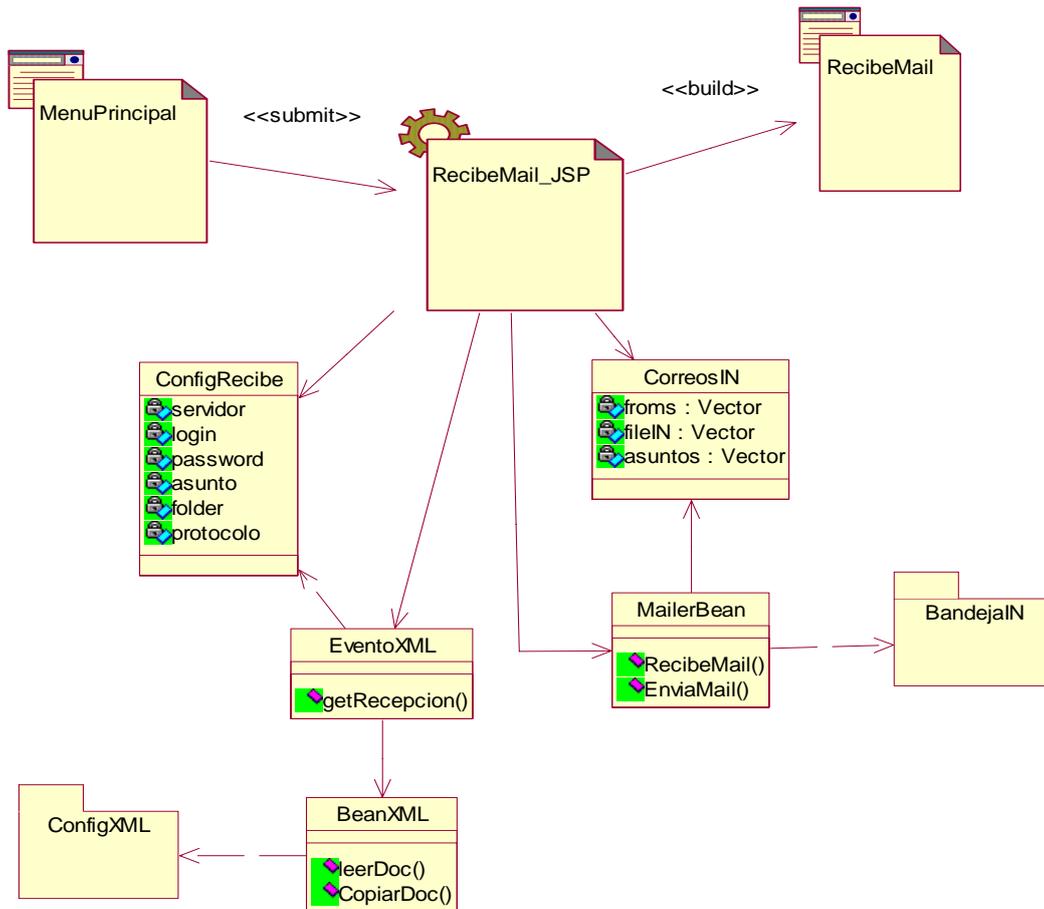
### A.3.1.2 Caso de uso Registrar evento epidemiológico.



**Figura A. 18.** Diagrama de clases de diseño Caso de uso Registrar evento epidemiológico.

Este caso de uso se presenta en la aplicación de escritorio del cliente, guarda la información recolectada por los formularios en la bandeja de salida del cliente para ser enviada por correo electrónico, inicia cuando se presiona el botón Guardar de IU\_Formulario. Las clases de control vistas en el modelo de análisis son implementadas como Beans en Java, los cuales se encargan de transformar los datos ingresados en documentos XML en la bandeja de salida. La clase de entidad: Doc\_Formulario es un documento XML en la bandeja de salida del cliente.

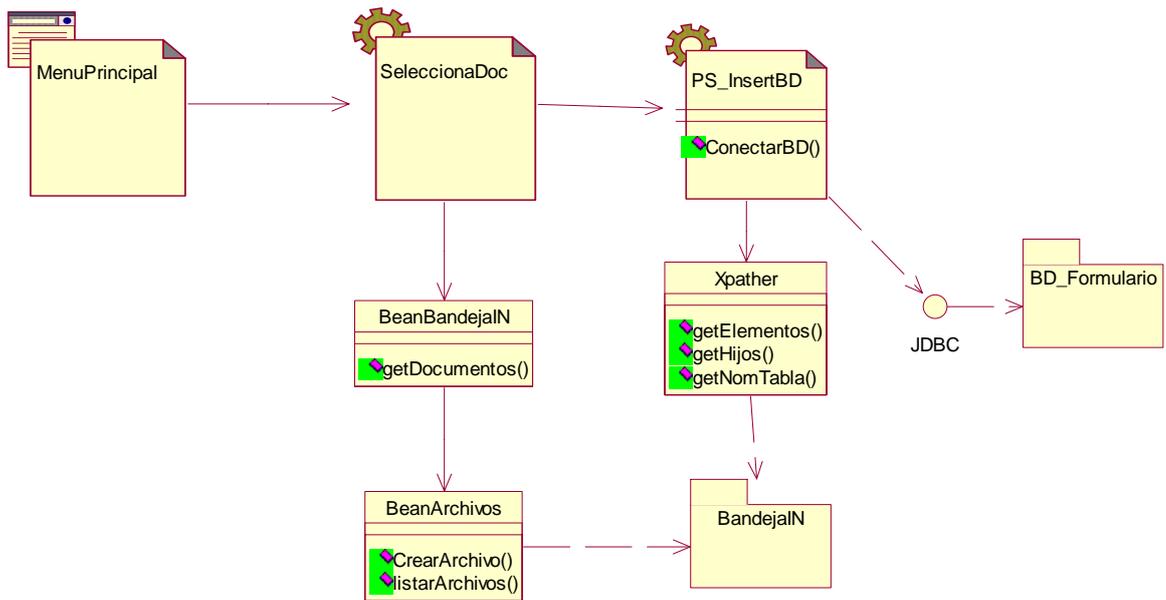
### A.3.1.3 Caso de uso Buscar Registros Enviados



**Figura A. 19.** Diagrama de clases de diseño Caso de uso Buscar registros enviados

Este caso de uso se presenta en la aplicación Web del servidor, se conecta al servidor de correo para buscar nuevos registros enviados desde la aplicación cliente y guardarlos como documentos XML en la bandeja de entrada de la aplicación servidor, inicia cuando se selecciona la opción buscar registros desde el Menu\_Principal. Las clases de control vistas en el modelo de análisis son implementadas como paginas JSP, Servlets y Beans en Java, los cuales se encargan de interpretar el documento XML de configuración de conexión y transformar los correos entrantes en documentos XML en la bandeja de entrada. Las clases de entidad: ConfigXML y BandejaIN son documentos XML almacenados en el servidor.

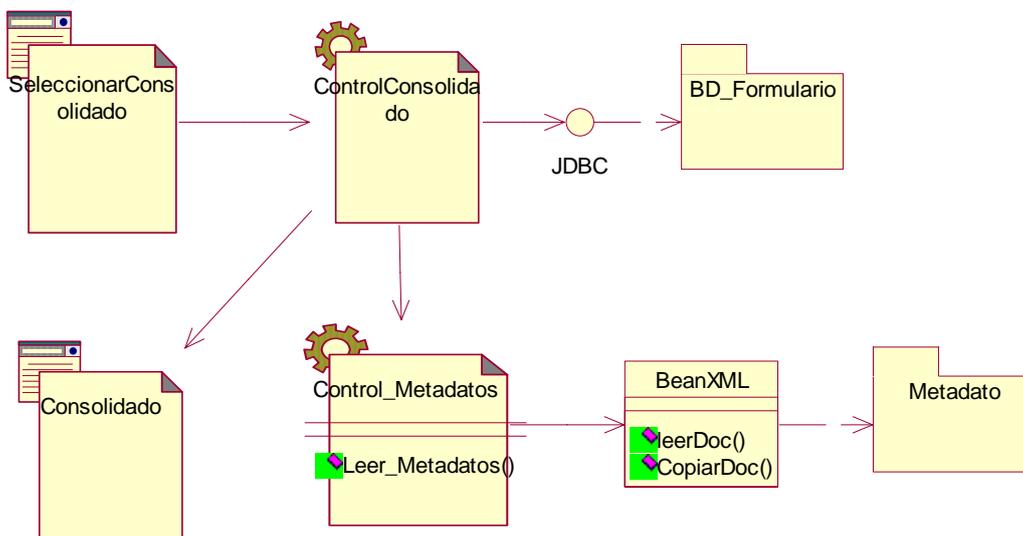
### A.3.1.4 Caso de uso Actualizar Base De Datos.



**Figura A. 20.** Diagrama de clases de diseño Caso de uso Actualizar base de datos

Este caso de uso se presenta en la aplicación Web del servidor, busca en la bandeja de entrada nuevos registros y los almacena en la tabla respectiva de su base de datos, inicia cuando se selecciona la opción Importar registro en el Menu\_Principal. Las clases de control vistas en el modelo de análisis son implementadas como Servlets y Beans en Java, los cuales se encargan de interpretar el documento XML con la información recolectada y el documento XML de metadatos y almacenarlos en la base de datos. Las clases de entidad: Bandeja\_Entrada y Metadatos son documentos XML almacenados en el servidor.

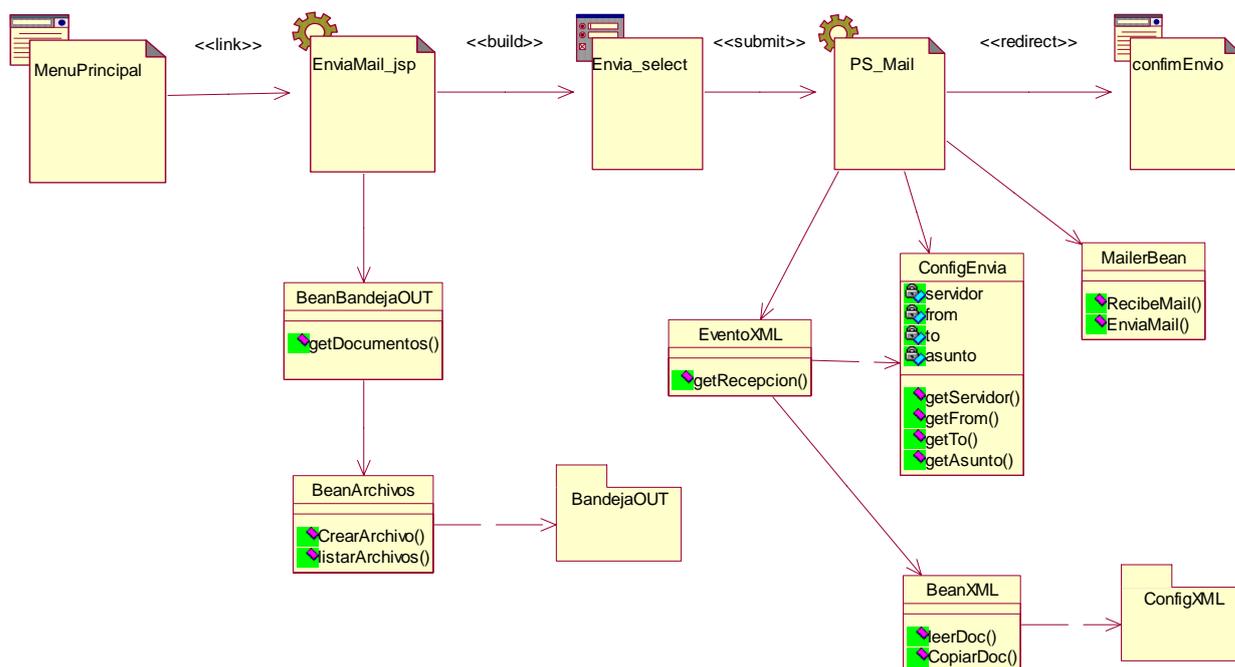
### A.3.1.5 Caso de uso Consultar Consolidado SIVIGILA.



**Figura A. 21.** Diagrama de clases de diseño Caso de uso Consultar consolidado SIVIGILA.

Este diagrama muestra como se implementan las clases de análisis. Se inicia cuando se selecciona un consolidado desde la ventana Seleccionar\_Consolidado. Las clases de control ControlConsolidado y ControlMetadatos son implementadas como Servlets y Beans en Java, los cuales se encargan de interpretar el documento XML de metadatos e ingresar la información a la base de datos a través de la Clase BD\_Formulario. La clase de entidad: Metadatos son documentos XML almacenados en el servidor.

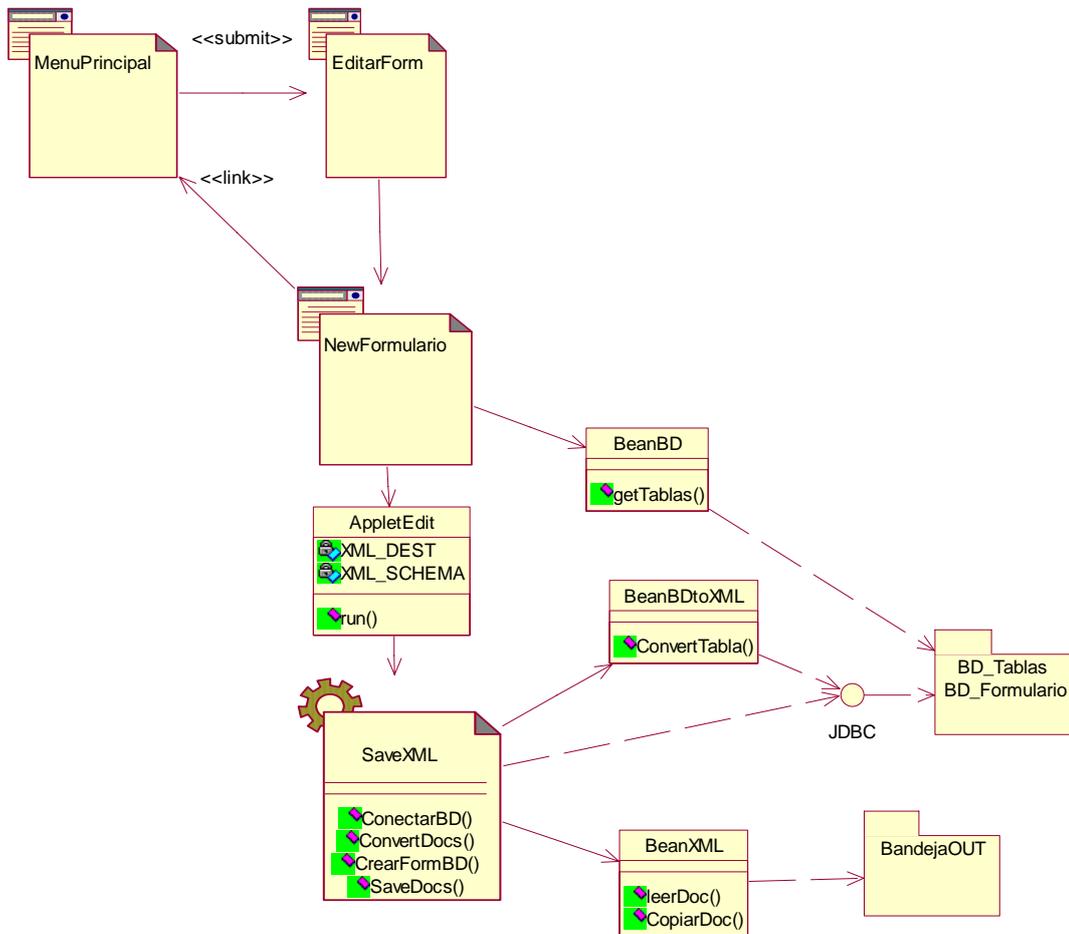
**A.3.1.6 Caso de uso enviar formulario.**



**Figura A. 22.** Diagrama de clases de diseño Caso de uso enviar formularios..

Este caso de uso se presenta en la aplicación Web del servidor, busca nuevos formularios en la bandeja de salida y los envía por correo electrónico a la aplicación cliente según la configuración del documento de conexión, inicia cuando se selecciona enviar formulario de IU\_Menu\_Principal. Las clases de control vistas en el modelo de análisis son implementadas como Servlets y Beans en Java, los cuales se encargan de obtener el nombre de los documentos a enviar en la bandeja de salida e interpretar el documento XML de configuración de conexión para correo electrónico. Las clases de entidad: Metadatos y Configurar\_Envio son documentos XML almacenados en el servidor.

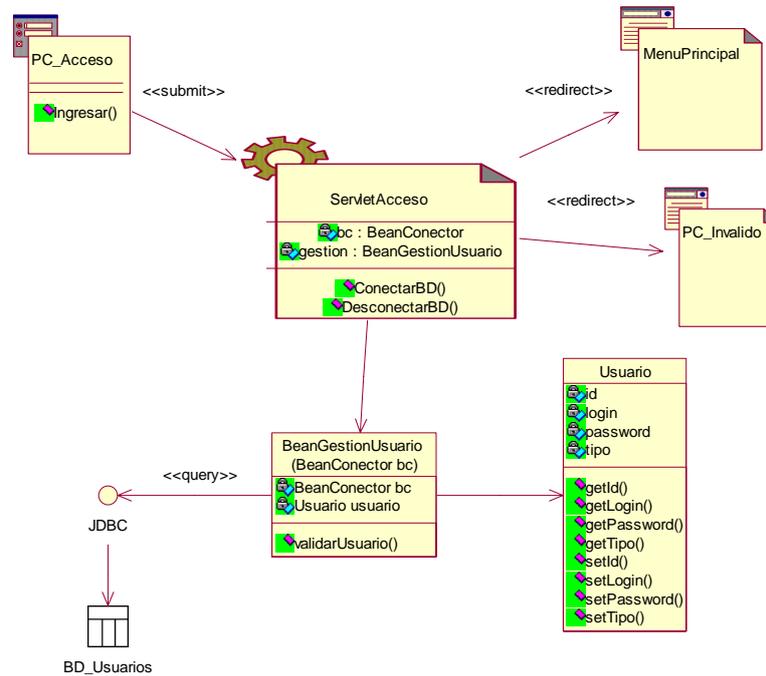
### A.3.1.7 Caso de uso Crear nuevo formulario.



**Figura A. 23.** Diagrama de clases de diseño Caso de uso crear nuevo formulario.

Este caso de uso se presenta en la aplicación Web del servidor, crea en la base de datos y en la bandeja de salida un nuevo formulario de recolección de información, inicia cuando se selecciona Editar Formulario de IU\_EditarFormulario. Las clases de control vistas en el modelo de análisis son implementadas como Servlets y Beans en Java, los cuales se encargan de obtener información de la base de datos, crear un nuevo formulario y generar un nuevo documento XML para el formulario. Las clases de entidad: Nombre\_Tablas y Tabla\_Formulario son obtenidas de la base de datos.

### A.3.1.8 Caso de uso identificarse en el sistema.



**Figura A. 24.** Diagrama de clases de diseño Caso de uso identificarse en el sistema.

Este caso de uso se presenta en la aplicación de escritorio del cliente y la aplicación Web del servidor, permite a los usuarios del sistema identificarse y tener acceso a diferentes funciones de la aplicación, inicia cuando se ejecuta la aplicación o cuando se accede a la página de inicio. Las clases de control vistas en el modelo de análisis son implementadas como Servlets y Beans en Java en la aplicación Web y como Beans en la aplicación cliente, estos se encargan de obtener información de los usuarios en la base de datos y en documentos XML, validar el acceso y generar logs de errores. La clase de entidad: Usuario es un documento XML en la aplicación cliente y una tabla de la base de datos en la aplicación Web.

### A.3.2 Diagrama general de clases de Diseño

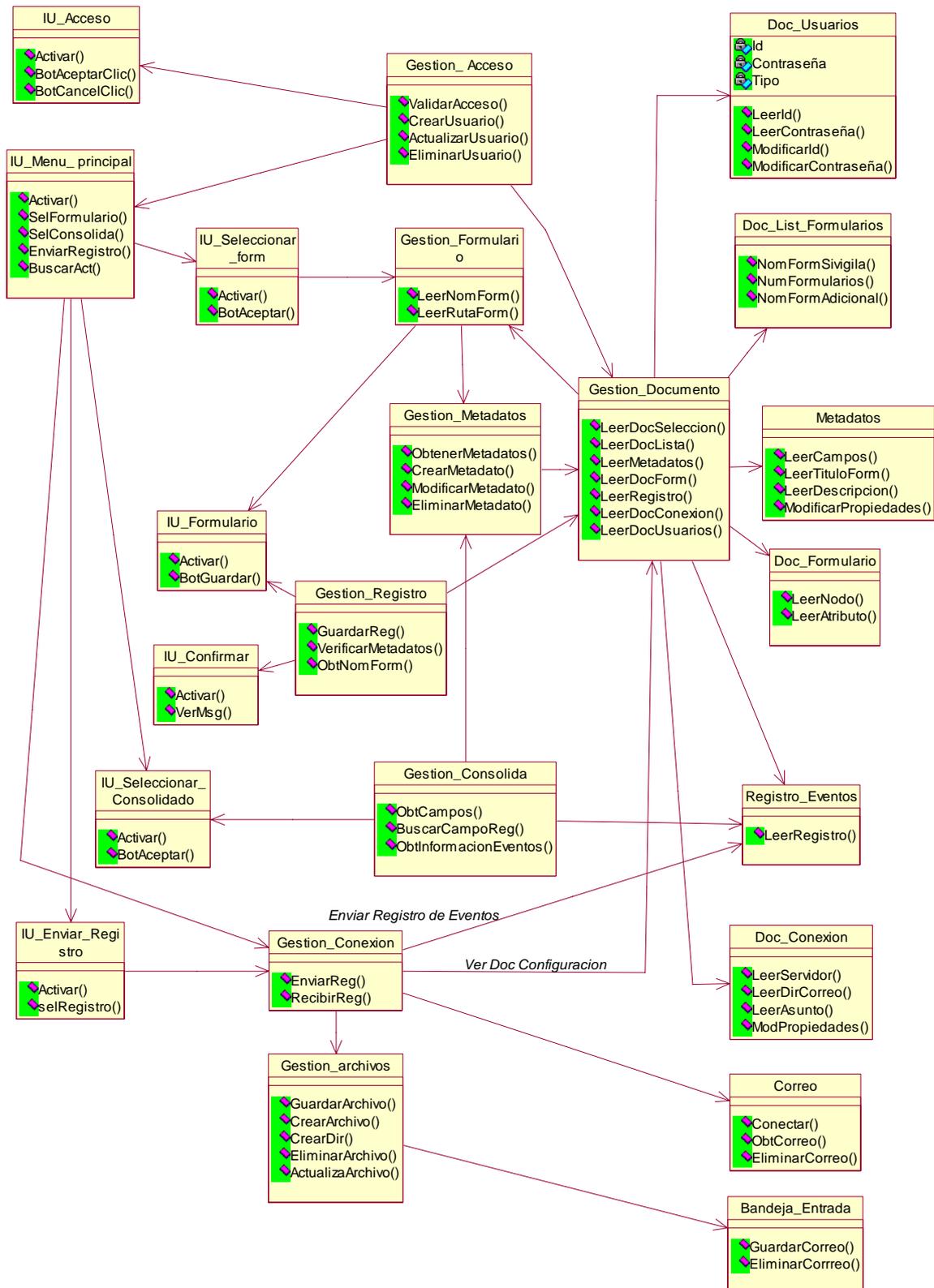


Figura A. 25. Diagrama general de clases de diseño

### A.3.3 Diagrama de paquetes de diseño

Los principales paquetes diseñados para la aplicación y que agrupan las clases de diseño son Herramienta Cliente, Documentos XML, Herramienta Servidor y Bases de Datos. Los paquetes del entorno de programación son principalmente Java y Javamail como API para el manejo de correo electrónico, como base de datos se tienen documentos XML, manejados por medio del API de JDOM. La plataforma sobre la cual se implementa es Windows o Linux, usando como protocolo transporte SMTP. Para el desarrollo de la herramienta Web Servidor se usan paquetes de JSP, HTML, Servlets y JDBC como API de conexión a la base de datos Firebird, para el transporte de información también se usa el API de Javamail, esta herramienta esta disponible al usuario a través de TCP/IP y HTTP.

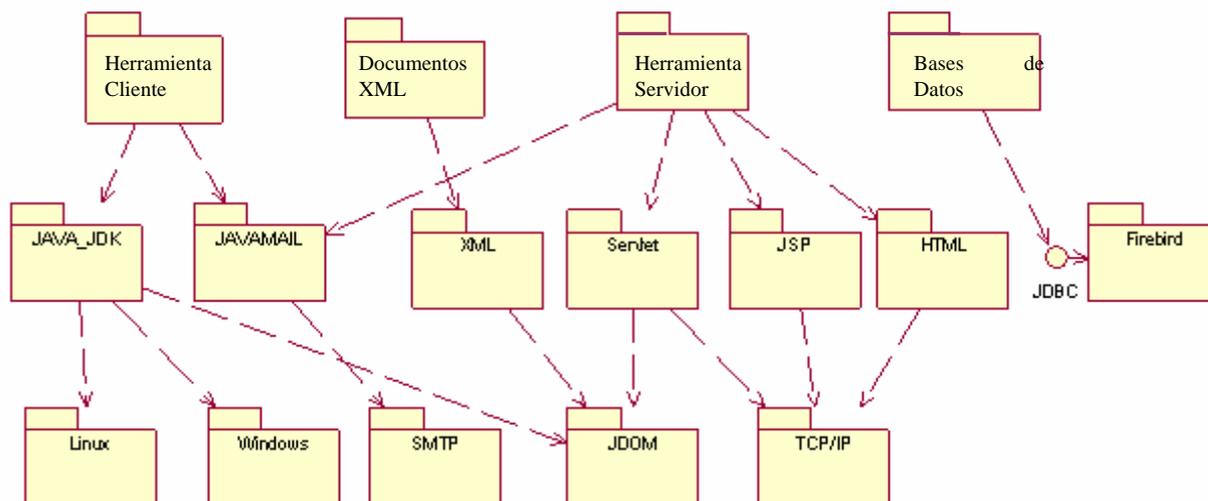


Figura A. 26. Diagrama general de paquetes de diseño

### A.3.4 Diagrama de implantación

Para la herramienta Cliente (PC\_Cliente) se requiere de un computador con una configuración hardware básica, no se necesita gran capacidad de memoria o procesamiento ya que se trata de una aplicación liviana, la información es almacenada en documentos XML, los cuales son accedidos a través del API de JDOM. Los documentos XML aparecen tanto en el PC cliente como en el PC Servidor porque se están actualizando constantemente de un lado a otro. Esto se demuestra mediante una relación de dependencia con el estereotipo <<becomes>>. La aplicación cliente corre sobre cualquier plataforma ya sea Windows o Linux , sólo se requiere instalar la máquina virtual de Java. Para la comunicación con la aplicación Web (PC\_Servidor) se utiliza SMTP.

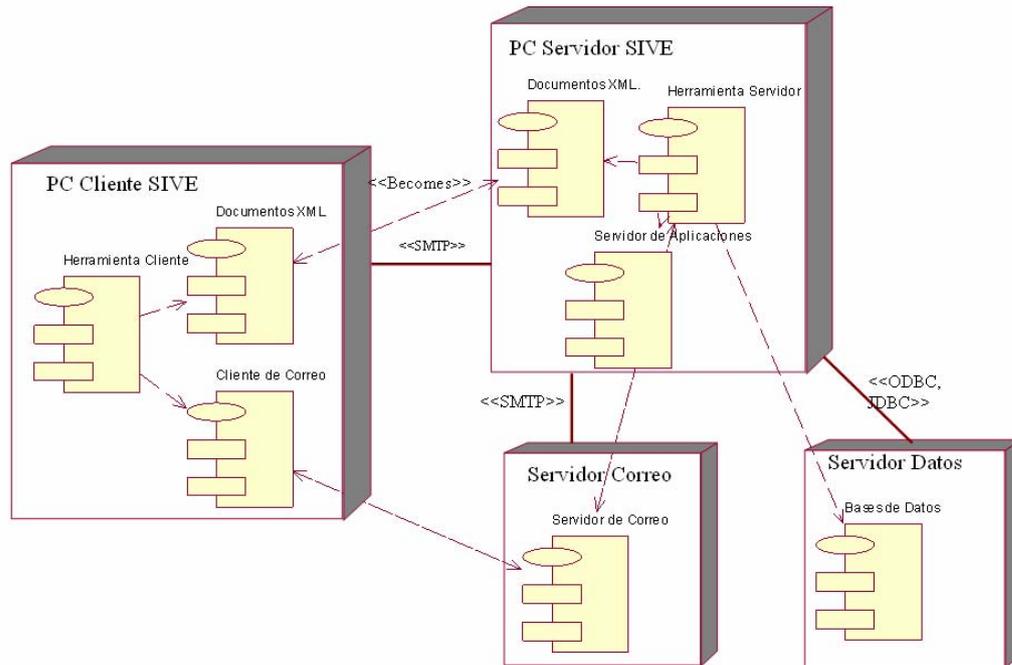


Figura A. 27. Diagrama de Implantación

Para la Herramienta Servidor (PC\_Servidor) se requiere de un computador con una configuración hardware de servidor ya que sobre este se instalan el servidor de aplicaciones que controla los Servlets y JSPs además del manejo de documentos XML. En este mismo equipo o en un computador diferente está el Servidor de correo electrónico, el cual se encarga de comunicarse con los clientes. Así mismo podemos tener implementada la base de datos en el mismo equipo u otro diferente como se representa en la Figura.

#### A.4 Modelo de Implementación

En UML el modelo de implementación se representa normalmente mediante diagramas de Componentes y Diagramas de Implantación describiendo detalles de la implementación del sistema: archivos ejecutables, de configuración, librerías, etc.

En este proyecto no se describe el modelo de implementación con diagramas UML. Sin embargo, para tener una visión general de la estructura e implementación de la aplicación final, a continuación se describen los componentes de la aplicación de manera informal de acuerdo a los seis componentes básicos definidos en la Figura A 26 Diagrama de Implantación: Herramienta Cliente, Documentos XML, Cliente Correo, Herramienta Servidor, Documentos XML Servidor, Servidor de Correo y Bases de Datos.

##### A.4.1 Descripción de los Componentes

La Figura A.27 representa el diagrama de componentes de la aplicación. La información es recolectada por la Herramienta Cliente en los puestos de salud o los

Hospitales Municipales y enviada por correo electrónico a la Herramienta Servidor ubicada en la Unidad Notificadora del municipio para obtener los consolidados por municipio y ser analizados por el personal de salud encargado y los COVE (Comités de Vigilancia Epidemiológica). El sistema cuenta con una herramienta gestor de formularios o metadatos, basada en un editor XML, esta herramienta se integró a la aplicación Web, para que el personal de salud encargado pueda crear y modificar sus propios formularios de recolección de información epidemiológica definiendo únicamente el nombre de los campos y el tipo de datos a recolectar ya que la estructura de estos se encuentra predefinida en el editor XML por medio de documentos XMLSchema. El documento XML generado por el gestor de formularios es enviado a través de la red de comunicaciones, por medio de correo electrónico, a la herramienta cliente en cada puesto de salud del municipio para generar dinámicamente el formulario de recolección. A continuación cada uno de los componentes es descrito en detalle.

**¡Error! No se pueden crear objetos modificando códigos de campo.**

**Figura A.28.** Diagrama de componentes de la aplicación

#### **A.4.1.1 Componente de recolección de información.**

Este componente cumple la función de recolección de información epidemiológica y de pacientes registrados por medio de los siguientes formularios:

- **Formulario pacientes:** Es un formulario que recoge la información de los pacientes que han presentado algún caso de notificación obligatoria, con el fin de tener un registro de los datos personales de cada paciente.
- **Formularios SIVIGILA:** Son aquellos formularios que siguen el formato del Sistema de Vigilancia Epidemiológica de Colombia, SIVIGILA. Los formatos son : Enfermedades de notificación obligatoria, Muerte Materno Perinatal y EDA-Enfermedades Diarreicas Agudas/IRA-Enfermedades Respiratorias Agudas.
- **Formularios adicional:** Son aquellos formularios que pueden ser editados desde el Servidor para la recolección de información epidemiológica adicional requerida por personal de salud local o los COVE.

La herramienta Cliente esta desarrollada en Java y utiliza documentos XML como repositorio de datos y como fuente de datos. La función de *repositorio* es para el almacenamiento de la información recolectada, la función de *fuentes* se realiza cuando el sistema requiere información para la presentación de los formularios, la cual es enviada desde el servidor en documentos XML para la actualización de los formularios, por ejemplo Enfermedades de Notificación Obligatoria (ENO), Municipios, IPS. Para los formularios adicionales se utilizan XmlSchemas como metadatos (estructura de información).

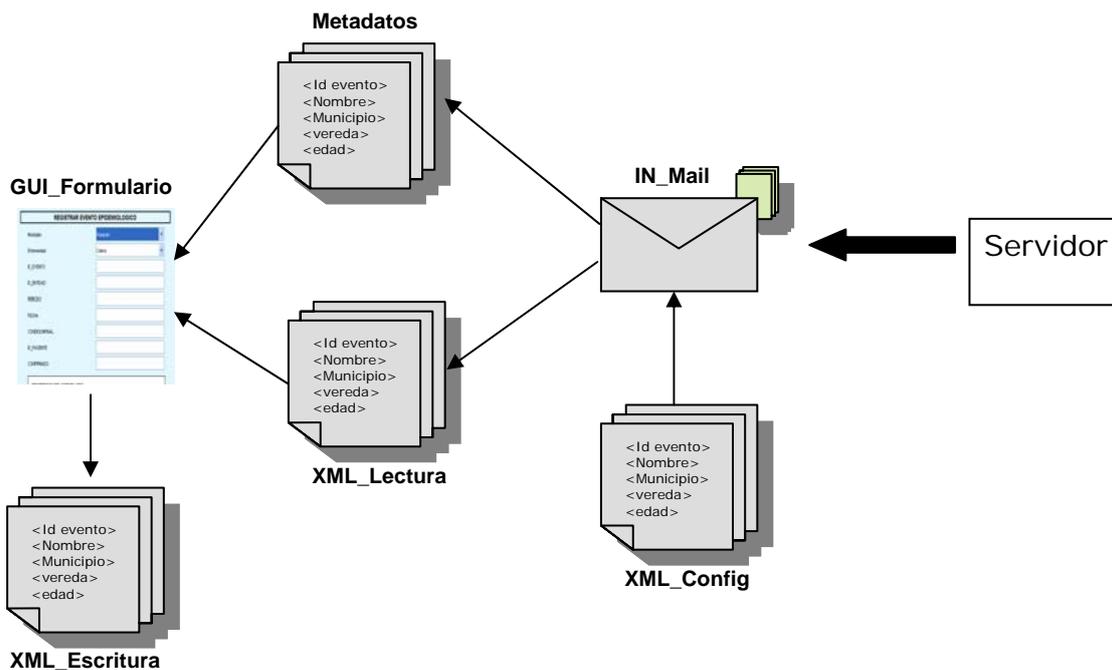


Figura A. 29. Componente de recolección de información

**GUI\_Formularios:** Son las interfaces que presentan los formularios para recolección de información, a través de ella el usuario introduce la información epidemiológica o de pacientes y la guarda localmente. Están implementados en Swing sobre Java y controlados por JavaBeans.

**XML\_Lectura:** La información que se presenta como menús de selección en los GUI\_Formularios es obtenida de documentos XML llamados de *lectura*. Los cuales son enviados por correo electrónico desde el Servidor y se actualizan automáticamente. Los XML\_Lectura están controlados por JavaBeans que utilizan los API de JDOM para manejo de documentos XML.

**XML\_Escritura:** La información recolectada en el GUI\_Formulario es guardada localmente como documentos XML llamados de *escritura*. Los XML\_Escritura están controlados por JavaBeans que utilizan los API de JDOM para manejo de documentos XML.

**Metadatos:** Los GUI\_Formularios son presentados al usuario según una estructura definida previamente en documentos XmlSchemas llamados Metadatos. Los Metadatos contienen información sobre nombre de los campos en el formulario, propiedades de los campos (selección o texto), número de campos. Estos Metadatos son definidos por el Coordinador de VE en el Servidor y enviados al Cliente por correo electrónico. Los formularios Adicionales son los únicos que pueden ser modificados, por lo tanto sus Metadatos son dinámicos (se crean en el Servidor y se envían al Cliente para su actualización).

**IN\_Mail:** Los documentos XML\_Lectura y los XmlSchemas son enviados desde el Servidor por correo electrónico y recibidos por el Cliente. Este proceso se realiza automáticamente, el Cliente se conecta con un servidor de correo predeterminado y

busca en él correos por asunto, en el caso de encontrar un correo con un asunto definido y realizar la correspondiente comprobación copia los documentos en una carpeta local para ser accedidos por el sistema Cliente. Se utiliza el API de JavaMail para el procesamiento y manejo del correo electrónico.

**XML\_Config:** Documento XML que contiene las opciones de configuración para la conexión, envío y recepción de documentos XML a través del servicio de correo electrónico.

#### A.4.1.2 Componente de Notificación de eventos.

Este componente cumple la función de Notificación o envío de la información epidemiológica recolectada por el componente de recolección.

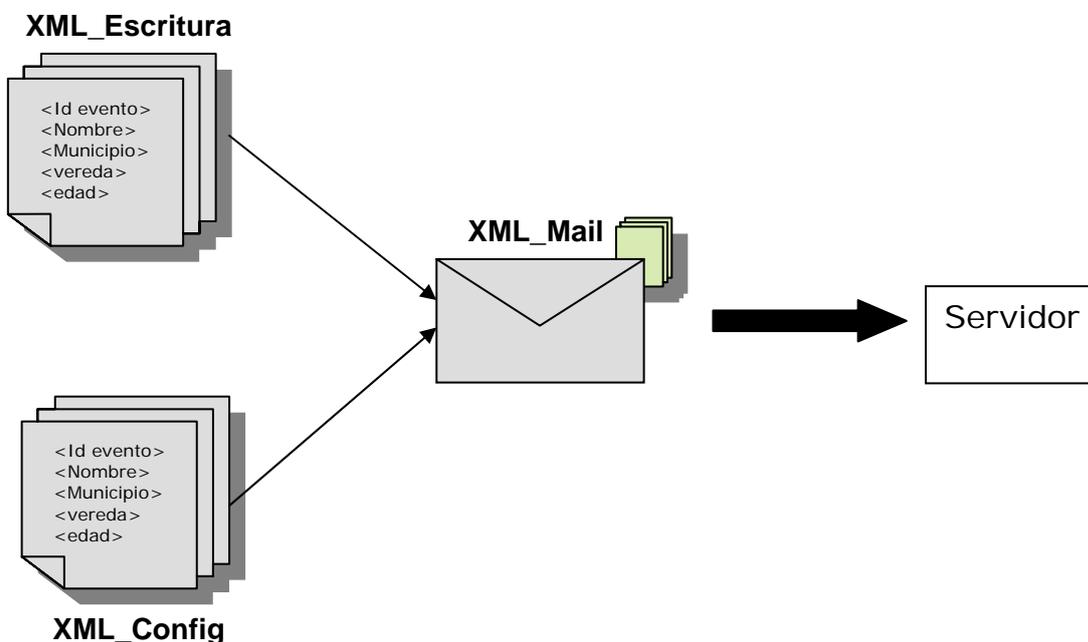


Figura A. 30. Componente de Notificación de eventos

**XML\_Mail:** Cuando el usuario lo requiera puede enviar la información epidemiológica recolectada al Servidor por medio de correo electrónico. Los documentos XML\_Escritura se envían como archivos adjuntos de un email, el correo de origen y destino estarán dados por un servidor de correo predefinido para esta tarea, el asunto y cuerpo del mensaje serán siempre los mismos ya que estos permitirán identificar al Servidor de que documento se trata y realizar la respectiva función. Esta información esta almacenada en un documento XML\_Config el cual es procesado por JavaBeans que utilizan el API JavaMail para manejo de correo electrónico. Los documentos son enviados solamente una vez y se guarda una copia de respaldo localmente, la información no se repite, en cada envío se adjunta solo la información recolectada desde el ultimo envío. Este proceso se realiza automáticamente por el sistema, de esta forma el usuario no requiere de conocimientos en correo electrónico para su utilización.

### A.4.1.3 Componente de consolidación de información.

Este componente cumple la función de consolidar la información en reportes epidemiológicos para ser analizados por el personal de salud local o por los COVE.

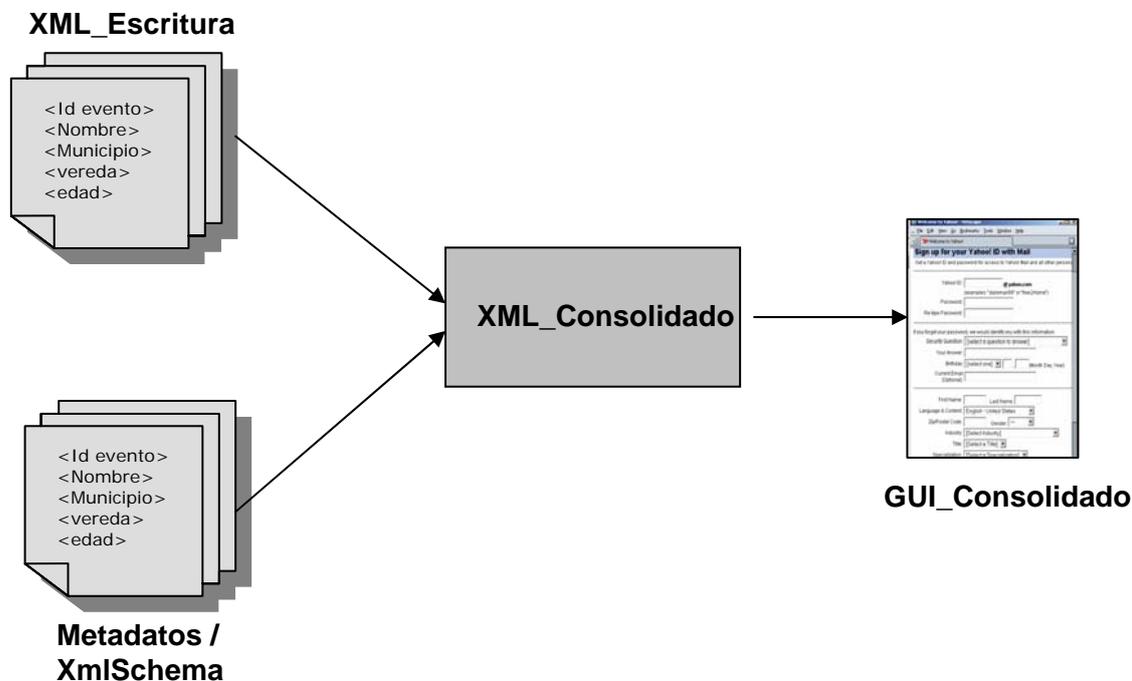


Figura A. 31. Componente de consolidación de información.

**XML\_Consolidado:** Los consolidados o reportes epidemiológicos son presentados en el Cliente siguiendo los formatos del SIVIGILA, para esto se utilizan los Metadatos definidos en XmlSchemas, los cuales definen los campos que serán incluidos en los consolidados y la información es obtenida de los XML\_Escritura almacenados localmente en el Cliente. Debido a que esta fase utiliza los Metadatos para la creación de los consolidados, la información de SIVIGILA y de los formularios adicionales creados dinámicamente se consolidan de la misma forma. También es posible imprimir los reportes consolidados.

**GUI\_Consolidado:** Son las interfases que presentan los reportes consolidados en el Cliente, a través de ellos el usuario puede desplegar la información epidemiológica recolectada en un periodo de tiempo determinado para su análisis. Están implementados en Swing sobre Java y controlados por JavaBeans.

### A.4.1.4 Componente XML de conversión.

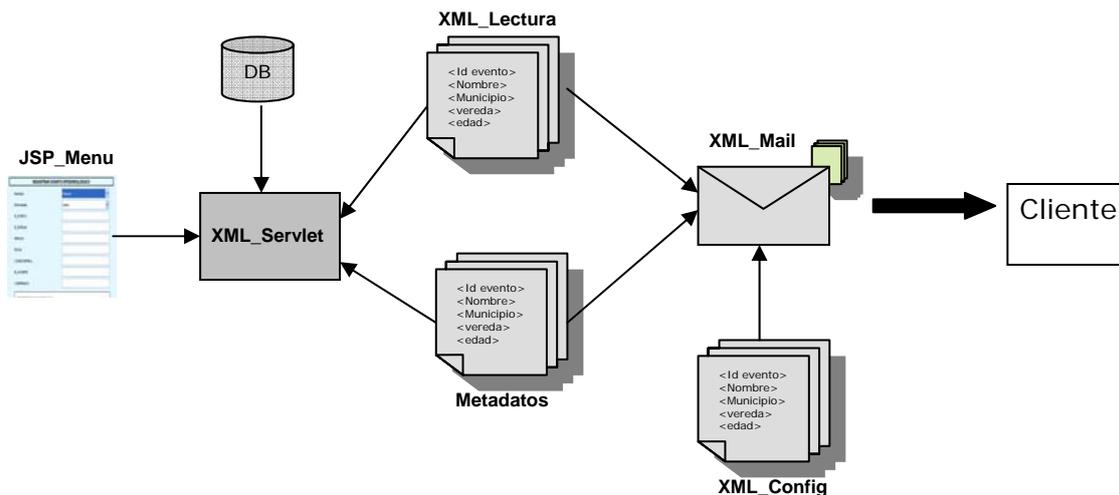


Figura A. 32. Componente XML de conversión.

**JSP\_Menu:** Interfaz de usuario que permite seleccionar la función que el usuario realiza, busca el Servlet que cumple esta función y lo inicializa. Esta realizado con JSP y código HTML.

**XML\_Servlet:** En la herramienta Servidor tenemos la base de datos central del sistema con su gestor, en nuestro caso, FireBird, ya que la información que enviamos al Cliente esta en documentos XML necesitamos realizar la correspondiente conversión a XML. De esta forma la información almacenada en nuestra base de datos será mapeada a documentos XML\_Lectura para ser utilizados posteriormente por el Cliente. De la misma forma convertimos esta información a Metadatos o XmlSchemas los cuales se utilizaran para crear los formularios y los consolidados en el Cliente (Componente de Recolección y Componente de Consolidación). Este componente se desarrolla totalmente con tecnologías J2EE como Servlets y JSP, además utiliza las API de JDOM para manejo de documentos XML.

**XML\_Mail:** Similar al Cliente (Componente de Notificación), cuando el usuario lo requiera puede enviar la información convertida en los XML\_Lectura como documentos adjuntos de un correo electrónico. Estos documentos XML\_Lectura se envían como archivos adjuntos de un email, el correo de origen y destino estarán dados por un servidor de correo predefinido para esta tarea, el asunto y cuerpo del mensaje serán siempre los mismos ya que estos permitirán identificar al Servidor de que documento se trata y realizar la respectiva función. Esta información esta almacenada en un documento **XML\_Config** el cual es procesado por JavaBeans que utilizan el API JavaMail para manejo de correo electrónico. Este proceso se realiza automáticamente por el sistema, de esta forma el usuario no requiere de conocimientos en correo electrónico para su utilización.

#### A.4.1.5 Componente de sincronización.

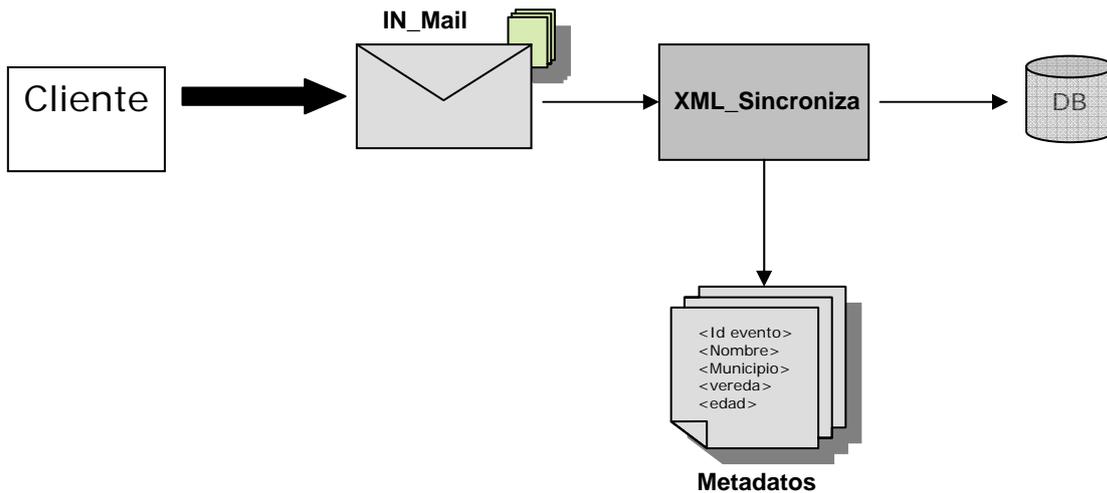


Figura A. 33. Componente de sincronización.

**IN\_Mail:** Similar al Cliente (Componente de recolección), los documentos XML\_Escritura son enviados desde el Cliente por medio de correo electrónico, automáticamente el Servidor revisa en su bandeja de entrada los correos con un asunto predefinido, si los encuentra guarda los documentos XML adjuntos en una carpeta local, de esta forma se puede realizar alguna función de depuración para verificar que la información que se va a insertar a la base de datos no se encuentre duplicada o con errores.

**XML\_Sincroniza:** Los documentos XML\_Escritura del Cliente son leídos por el Servidor e insertados a la base de datos, en este proceso se incluyen algunas reglas para evitar errores en la sincronización de la información como la implementación de claves primarias y la comprobación de documentos XML bien formados por medio de los XmlSchemas creados para los Metadatos. Para este componente se utilizan los API de JDOM para manejo de documentos XML y JavaBeans manejados por Servlets para procesamiento de la información en la base de datos.

#### A.4.1.6 Componente de consolidación.

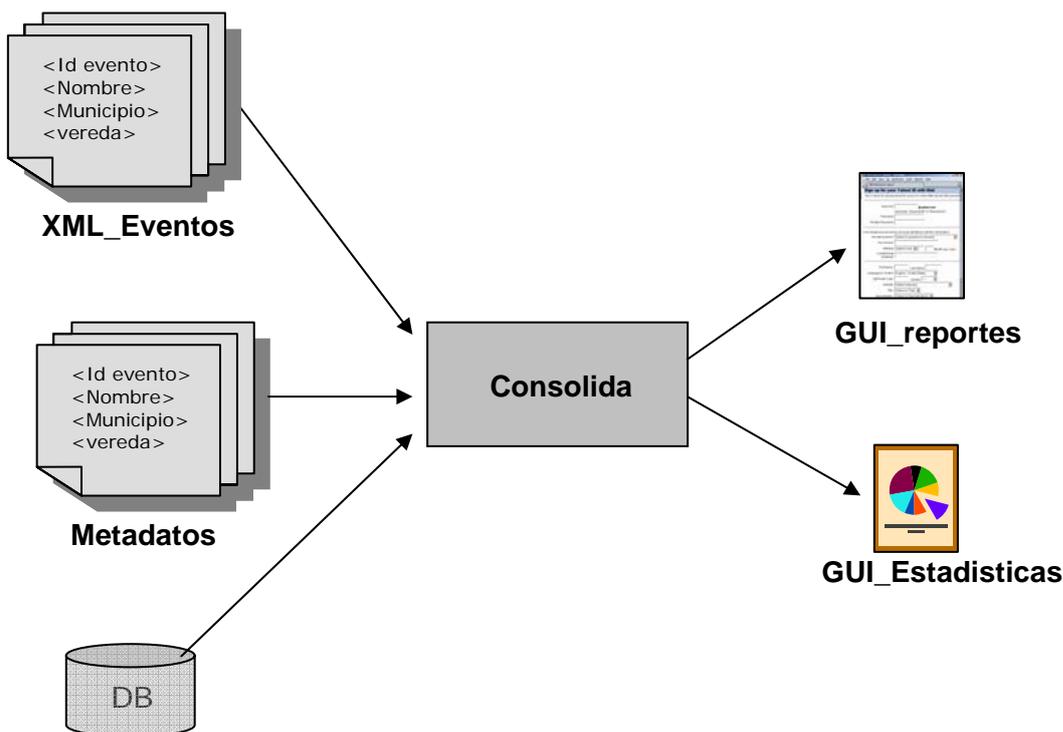


Figura A. 34. Componente de consolidación

**Consolida:** Al igual que el Cliente (Componente de Consolidación), el Servidor tiene la posibilidad de generar reportes consolidados con la información almacenada en los documentos XML (**XML\_Eventos**) y la base de datos. Aunque la información es la misma podemos reutilizar el componente de Consolidación del Cliente para generar los reportes y con la información de la base de datos generamos graficas estadísticas como tasas de mortalidad y morbilidad.

**GUI\_Reportes:** Es la interfaz que nos permite visualizar los reportes consolidados en el Servidor para su respectivo análisis por parte de la DDSC y los Coordinadores de VE. Estos reportes pueden ser impresos.

**GUI\_Estadisticas:** Es la interfaz que nos permite visualizar los datos epidemiológicos en forma de graficas estadísticas.

#### A.4.1.7 Componente Gestor de Metadatos.

Este componente es el que se encarga de crear los *Metadatos* o documentos de esquema en XmlSchemas, los cuales serán utilizados en el sistema para:

- Definir la estructura y propiedades de los formularios que aparecerán en el Cliente. (Nombre de campos, cajas de texto, de selección).
- Definir la estructura de los reportes consolidados en el Cliente y el Servidor. (Campos a desplegar, tipos de datos).
- Verificar que los documentos XML están bien formados antes de insertarlos a la base de datos del Servidor.

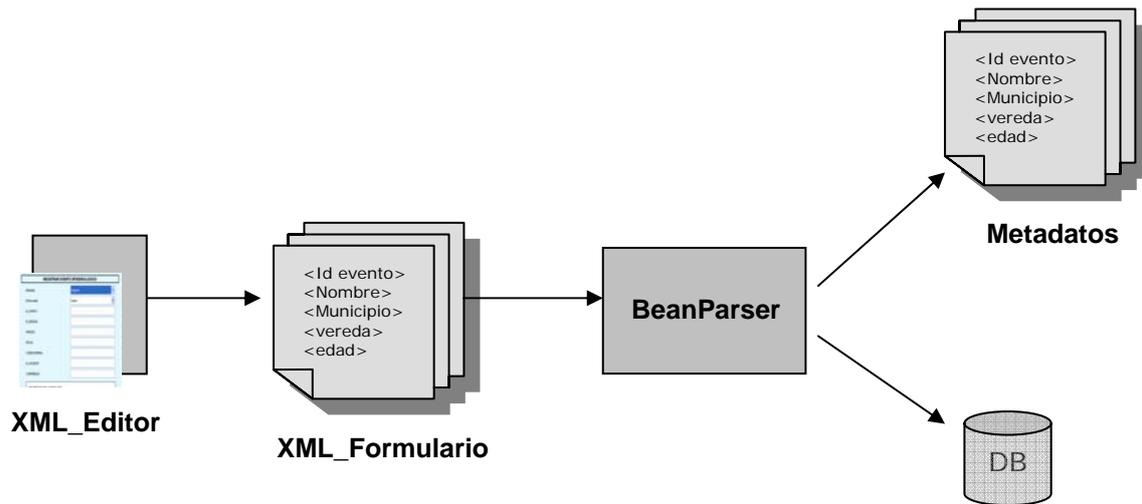


Figura A. 35. Componente Gestor de Metadatos.

**XML\_Editor:** Es una aplicación o herramienta funcional que le permite al usuario editar o crear formularios adicionales de recolección de información epidemiológica de una forma amigable y sencilla en formato XML sin necesidad de saber sobre este lenguaje de marcado. Esta puede ser una de las herramienta de libre distribución que se utilizan dentro del proyecto para la edición de cursos en XML, solo que en este caso editamos formularios con propiedades definidas.

**XML\_Formulario:** Son los formularios en XML, el formato de estos documentos depende de la herramienta que utilizemos como XML\_Editor.

**BeanParser:** Los XML\_Formularios deben ser interpretados como Metadatos (XmlSchemas) y convertidos en una tabla en la base de datos del Servidor. Este componente es un JavaBeans que utiliza las API de JDOM para manejo de documentos XML.

## A.5 Referencias

- [1] Figueroa J. A, Vidal M. Modelo para la definición, recolección y entrega de información en un sistema de información para vigilancia epidemiológica. Trabajo de Grado. Facultad de Ingeniería Electrónica y Telecomunicaciones. Universidad del Cauca, Popayán, 2003.