Ingeniería Técnica Informática de Gestión

TRABAJO FINAL DE CARRERA – INGENIERIA DEL SOFTWARE

Memoria del proyecto GAS

Estudiante: Lola Rosado Pérez

Consultor: Juan José Cuadrado Gallego

<<Página en blanco>>

Universitat
Oberta
de Catalunya

A mis padres por su infinita generosidad. A mis hijos Enrique y Erica por su paciencia y comprensión. A mi compañero Erich, sin cuyo apoyo y cariño constantes no hubiera llegado hasta aquí.

Universitat Oberta de Catalunya

Agradecimientos

Este documento es el resultado de un esfuerzo no sólo de un trimestre sino de varios años cuando decidí retomar mis estudios de Ingeniería Técnica en la UOC.

En estos años he tenido la satisfacción de encontrarme en el camino con un número de personas estupendas, desde compañeros de la UOC, consultores y tutores a quién que evidentemente no conozco en persona. Sería muy largo nombraros a todos, pero vosotros sí sabéis quién sois; os llevo en mi corazón.

Estoy feliz y orgullosa de pertenecer a esta gran Universidad Abierta de Cataluña, porque realmente está abierta al espíritu del buen saber, del conocimiento y de la comunicación. No tengo nada más que buenas palabras para esta Universidad, que para mí, hace honor a su nombre.

A todos ellos, a todos vosotros, no puedo sin emocionarme expresaros mi más humilde gratitud.

Por otro lado, sería injusto no mencionar a mis compañeros de trabajo: Rafa Troncoso, Abelardo García, Juan López y en especial a José Granados, por su constante generosidad al ayudarme con sus explicaciones cuando les atacaba con mil y una dudas. Y cómo no, a mi supervisor Bob Dighello por permitir que este TFC lo haya podido desarrollar en el departamento. A todos vosotros, gracias.

A mis compañeras de trabajo, por sus constantes ánimos y cafés.

Y por último, a mi familia a la que he robado tantas horas de dedicación: a mis hijos que son pequeños y no entendían por qué su mamá trabajaba tanto delante del ordenador, a mis hermanos por su fé en mí, a mis padres, por tantos fines de semana entreteniendo a mis hijos; a mi marido por esas cenas y cafés que me ponía delante del ordenador. Os recompensaré con creces.

Al Universo, por esta bella oportunidad de cumplir mi sueño de ser Ingeniero. Gracias.

Lola Rosado Mayo 2006

1 Índice

1.1 Índice de Contenidos

1 Índice 5
1.1 Índice de Contenidos 5
1.2 Índice de Figuras 6
1.3 Índice de Tablas 7
2. Cuerpo de la Memoria 8
2.1 Introducción 8
2.1.1 Justificación del TFC y contexto en el que se desarrolla 8
2.1.2 Objetivos 8
2.1.3 Enfoque y Método Seguido 9
2.1.4 Producto Obtenido 9
2.2 El Gestor de Administración de Sistemas (GAS). 10
2.2.1 Descripción 10
2.2.2 Problemática. Situación actual 10
2.2.3 Beneficios de una correcta gestión 11
2.2.4 Necesidades básicas de un Gestor de Administrador de Sistemas 12
3. Modelo de Análisis 12
3.1 Identificación de Actores 12
3.1.1 Diagrama de Actores 12
3.1.2 Descripción Textual de los Actores 12
3.2 Identificación de Casos de Uso 13
3.2.1 Paquetes de Análisis 13
3.2.2 Casos de Uso Subsistema Gestor de Password 14
3.2.3 Casos de Uso Subsistema Gestor de Vulnerabilidad 14
3.2.4 Casos de Uso Subsistema Gestor Administración 15
3.3 Descripción Textual de los Casos de Uso 16
3.3.1 Casos de Uso del Subsistema Gestor de Password 16
3.3.2 Casos de Uso del Subsistema Gestor de Vulnerabilidad 17
3.3.3 Casos de Uso del Subsistema Gestor Administración 18
3.4 Relación de Casos de Uso principales. 22
3.5 Requisitos Hardware y Software 23
4. Introducción a la Fase de Diseño 23
4.1 Organización de alto nivel del Modelo de Diseño 23
4.2 Modelo de Análisis del Diseño. 24
4.2.1 Diagrama de Actividad 24
4.2.2 Diagramas de colaboración y secuencia 25
4.2.3 Identificación de las clases de entidades 33
4.2.4 Identificación de las relaciones entre clases. Herencia. 33
4.2.5 Asociaciones. 34
4.2.6 Especificación de los atributos de las clases de entidades 34
4.2.7 Transformación en Entidades de las Interrelaciones 35
4.2.8 Identificación de las clases frontera, las clases de control y de las operaciones. 37
4.3 Diseño de la persistencia 38
4.3.1 Diagrama estático de análisis. 38
4.3.2 Diagrama de Tabla Relacionales 39
4.3.3 Conclusiones sobre el Análisis realizado 39
4.4 El Diseño Arquitectónico Aplicación GAS 40
4.4.1 Servicios Web. Introducción 41
4.4.2 Arquitectura .NET 43
4.4.3 Diseño Web Aplicación GAS 45
5. Diseño de la Interfaz. 46
5.1 Escenario y Flujo de Interacción. Vocabulario Visual. 46
5.2 Prototipo de Alto Nivel 47

5.3 Conclusiones del Diseño. Fase Final 54
6. Bibliografía 56
7. Glosario 56
8. Anexos 58
8.1 Definición General del Proyecto 58
8.2 Objetivos del proyecto 59
8.2.1 Generales 59
8.2.2 Específicos 59
8.3 Entregables 59
8.4 Recursos e Infraestructura 60
8.4.1 Hardware 60
8.4.2 Software 60
8.5 Estructura del proyecto. Descomposición en actividades 60
8.5.1 Estructura del proyecto 60
8.5.2 Descomposición en actividades 60
8.5.3 Descripción de las actividades 61
8.6 Planificación temporal del proyecto 62
8.6.1 Identificación de Hitos 62
8.6.2 Estimación de tiempos 62
8.7 Diagrama de Gantt 63
9. Conclusiones 63

1.2 Índice de Figuras

Ilustración A: Diagrama de Actores 12
Ilustración B: Paquete de Análisis 14
Ilustración C: Casos de Uso Gestor Password 14
Ilustración D: Casos de Uso Gestor Vulnerabilidad 15
Ilustración E: Casos de Uso Gestor Administración 15
Ilustración F: Modelo de Diseño 23
Ilustración G: Diagrama de Actividad 24
Ilustración H: Diagrama colaboración *"Hacer Login"* 25
Ilustración I: Diagrama colaboración *"Escaneo servidores"* 26
Ilustración J: Diagrama colaboración *"Generación alertas"* 26
Ilustración K: Diagrama colaboración *"Envío Email Administradores"* 27
Ilustración L: Diagrama colaboración *"Alta Tareas Administración"* 27
Ilustración M: Diagrama colaboración *"Gestión de servidores"* 28
Ilustración N: Diagrama de secuencia *"Gestión de servidores"* 28
Ilustración O: Diagrama colaboración *"Alta de Vulnerabilidades"* 29
Ilustración P: Diagrama colaboración *"Alta de servidores"* 29
Ilustración Q: Diagrama colaboración *"Generar Informes"* 30
Ilustración R: Diagrama colaboración *"Alta de Incidencias"* 30
Ilustración S: Diagrama colaboración *"Informe de Servidores"* 31
Ilustración T: Diagrama de secuencia *"Informe de servidores"* 31
Ilustración U: Diagrama de Casos de Uso *"Alta Responsable"* 32
Ilustración V: Diagrama colaboración *"Alta Responsable"* 32
Ilustración W: Diagrama estático de análisis 38
Ilustración X: Diagrama E/R 39
Ilustración Y: Elementos de la Plataforma .NET 41
Ilustración Z: Componentes de la Arquitectura .NET 42
Ilustración AA: Componentes de la Arquitectura .NET 43
Ilustración BB: Modelo de Arquitectura .NET 44
Ilustración CC: Arquitectura .NET para aplicación GAS 45
Ilustración DD: Diagrama Vocabulario Visual 46
Ilustración EE: Formulario de acceso 47

Ilustración FF: Formulario error acceso 48
Ilustración GG: Formulario cambio contraseña 49
Ilustración HH: Formulario Alta tareas 50
Ilustración II: Formulario Alta de Incidencias 51
Ilustración JJ: Formulario Alta de Vulnerabilidades 52
Ilustración KK: Formulario Selección servidor 53
Ilustración LL: Formulario Jefes Departamento 54

1.3 Índice de Tablas

Tabla 1: Casos de Uso “Hacer Login” ID [001] 16
Tabla 2: Casos de Uso “Cambiar Password ID [002] 16
Tabla 3: Casos de Uso “Escaneo de Servidores ID [003] 17
Tabla 4: Casos de Uso “Generación de Alertas” ID [004] 17
Tabla 5: Casos de Uso “Envío Email Administradores” ID [005] 18
Tabla 6: Casos de Uso “Alta Tareas de Administración” ID [006] 18
Tabla 7: Casos de Uso “Gestión de Servidores” ID [007] 19
Tabla 8: Casos de Uso “Alta Vulnerabilidades” ID [008] 20
Tabla 9: Casos de Uso “Alta de Servidores” ID [009] 20
Tabla 10: Casos de Uso “Generar Informes” ID [007] 21
Tabla 11: Casos de Uso “Alta Incidencias” ID [011] 21
Tabla 12: Casos de Uso “Informe Servidores” ID [012] 22
Tabla 13: Relación de Casos de Uso 22
Tabla 14: Casos de Uso “Alta Tareas de Administración” ID [006] 61
Tabla 15: Identificación de Hitos 62
Tabla 16: Estimación de Tiempos 63
Tabla 17: Diagrama de Gantt 63

2. Cuerpo de la Memoria

2.1 Introducción

2.1.1 Justificación del TFC y contexto en el que se desarrolla

El objetivo de este Trabajo Final de Carrera consiste en la realización del desarrollo de un Gestor de Administración de Sistemas (GAS). El desarrollo de este gestor tendrá uso real y vendrá a cubrir una necesidad del departamento de Sistemas Informático del Hospital Americano de la Base Naval de Rota donde trabajo como administradora de sistemas.

En el Hospital gestionamos más de 90 servidores distintos. Al ser un recinto médico y militar, la seguridad en la red y en los sistemas es una gran prioridad.
Cada administrador tiene a su cargo un número de sistemas, con la responsabilidad de actualizar los parches de Microsoft, IAVAS, actualizaciones de software, bases de datos, hardware y demás tareas propias de la administración de un servidor.

En cualquier momento, nuestro supervisor y altas jerarquías pueden pedir la situación de nuestros servidores. Este gestor será la aplicación GUI mediante WEB que permitirá tener almacenado en un repositorio común toda la información de nuestros sistemas informáticos. Podrá ayudar a los administradores a poner a punto los servidores y por otro lado, el jefe de cada departamento podrá saber en cada momento el estatus de cada servidor así como el tiempo medio que emplea el administrador en cada uno de los servidores del que es responsable.

2.1.2 Objetivos

Los objetivos que perseguimos con la realización del presente TFC son los siguientes:

- Hay un gran desconocimiento sobre los tiempos empleados por el administrador en el mantenimiento de los sistemas. GAS va a cubrir una necesidad del Departamento Informático del Hospital Americano de la Base Naval de Rota sobre el tiempo empleado por cada Administrador en cada uno de sus servidores.

- Crear una aplicación que facilite y asegure que el administrador ejecuta todas las tareas de mantenimiento para asegurar que el funcionamiento del servidor sea efectivo y eficiente.

- Se busca maximizar los tiempos de los administradores para que trabajen de forma eficiente.

- Actualmente y dependiendo de cada administrador, los datos de los sistemas se almacenan localmente en el ordenador del administrador responsable y en distintos formatos. La información puede estar almacenada en Excel, en Word o en Access o no se guarda información al respecto. Este Gestor va a

proporcionar un modelo único donde la información estará almacenada en un repositorio común basado en tecnología Web.

- Desarrollar las técnicas de análisis y diseño de aplicaciones siguiendo estándares tanto técnicos como de especificación y notación: metodología RUP, modelo de análisis y diseño utilizando la notación UML 1.5 y una herramienta que la soporte (Rational Rose)
- Realizar el análisis y diseño de una aplicación con las siguientes características:
  - Interfaz de usuario Web.
  - Según los estándares de la programación orientada a objetos.
- Ampliar conocimientos sobre la mejor forma de diseñar una interfaz de usuario que cumpla los requisitos de usabilidad, coherencia, etc.

2.1.3 Enfoque y Método Seguido

He realizado un estudio del flujo de la situación actual ya que soy parte implicada en el proceso. He manteniendo diversas reuniones con los futuros usuarios de la aplicación, los administradores de sistemas para conocer sus necesidades y ver sus puntos de vista, también he mantenido varias reuniones con el responsable de seguridad y con los jefes de departamento.

He elaborado el primer borrador he procedido a generar los actores y casos de uso, presentándoselo al usuario para su visto bueno. Una vez consensuado este primer documento con los usuarios, he desarrollado el Modelo de Análisis y Diseño que componen esta memoria.

2.1.4 Producto Obtenido

En esta Memoria se describe el Modelo de Análisis compuesto por:

- Especificaciones de Requisitos del Gestor GAS
- Funcionalidades del Sistema. Descripción del Gestor.
- Requisitos Hardware y Software
- Identificación de Actores
- Identificación y Relación de Casos de Uso principales
- Documentación textual de Casos de Uso
- Diagramas de Casos de Uso

El modelo se ha realizado utilizando la herramienta de modelado visual Rational Rose. Estos documentos nos sirven de información de entrada para desarrollar el Modelo de Diseño compuesto por:

- Diagrama de Actividad
- Diagrama de colaboración y secuencia
- Identificación de entidades, asociaciones y atributos
- Identificación clases frontera, control
- Diagrama estático de análisis
- Diagrama de Entidad –Relación. Tablas Relacionales
- Estudio de la arquitectura a utilizar

- Prototipo de Alto Nivel

2.2 El Gestor de Administración de Sistemas (GAS).

2.2.1 Descripción

La gestión de los sistemas informáticos se complica cada vez más ya que cada vez va aumentando el número de servidores en el Hospital. Por otro lado, la administración es más complicada debido al control de la seguridad. Este proyecto surge de la necesidad de gestionar estos sistemas de una forma efectiva, segura y fácil para los administradores y técnicos. Y a su vez, satisfacer las necesidades de información de los directivos de más alto nivel y niveles intermedios.

Gracias a GAS los líderes podrán tener una visión clara de los sistemas informáticos a través de los informes que proporciona sin tener necesidad de disponer de conocimientos informáticos. Como resultado los lideres del departamento de IT podrán justificar el aumento o ajuste de los recursos humanos y técnicos en el sector IT del Hospital.

Por otro lado GAS avisará a los administradores mediante alertas de las tareas a realizar y del tiempo estimado por tarea. Se mantendrá un historial/inventario del software y hardware de cada máquina. GAS proporciona una mejora cualitativa en el trabajo del administrador ya que podrá delegar en un sistema seguro y centrarse en el verdadero trabajo de la gestión/administración del sistema liberándole del estrés de tener que controlar las tareas por realizar.

La función intrínseca de este gestor será la recogida, almacenamiento, procesamiento y distribución de la información de los servidores y las estaciones con aplicaciones especificas.

2.2.2 Problemática. Situación actual

Nos encontramos con distintas situaciones:

1. No hay documentación alguna sobre el mantenimiento que el administrador realiza en el servidor.

2. Se desconoce el tiempo empleado en mantenimiento.

3. Los Jefes de departamento desconocen las distintas tareas de mantenimiento realizadas en el servidor así como el tiempo medio empleado.

4. No hay documentación sobre troubleshooting. Cuando aparece un problema no podemos ver como se ha solucionado anteriormente si es que ese problema apareció en ese u otro servidor.

5. Hay un gran número de servidores (más de 90) y máquinas con aplicaciones especificas. Contamos con distintos sistemas operativos como Windows NT, Windows 2000 Server, Linux, MUMPS y XP.

6. El proceso de parches es el siguiente: El oficial de seguridad lanza un fichero batch donde ejecuta el programa hjnetchk. Este programa escanea los servidores y genera un fichero de texto que se importa manualmente a Access donde se genera una tabla.

Esta tabla contiene los campos:

- Nombre del servidor
- Producto (Nombre del Programa)
- Boletín
- Qnumber (Nombre de parche a aplicar)
- Razón
- Status

El responsable de seguridad agrupa los parches por servidor y envía un correo electrónico a los administradores para que se apliquen los respectivos parches de seguridad. Una vez enviado el email ya no hay mas seguimiento por parte de nadie, hasta que se vuelva a generar otro escaneo.

2.2.3 Beneficios de una correcta gestión

- Liberar al administrador del estrés de recordar todas las tareas que afectan a la administración del sistema, ya que GAS tiene un calendario que podrá ser programado por el administrador para que le avise de las tareas a realizar.

- El programa hjnetchk escanea los servidores y alimenta al programa GAS de todas las vulnerabilidades por cada servidor y sistema operativo. GAS avisará de forma automática mediante un controlador de eventos por email al Administrador sobre los parches a instalar. Por otro lado al entrar en GAS habrá una alerta en los servidores que sufran de alguna vulnerabilidad.

- GAS alertará al administrador de los parches a instalar dos formas distintas: a través de un email y mediante una alerta en el programa.

- GAS toma la información de los parches a instalar del fichero de texto generado por hfjetchk de forma automatizada. Evitando al responsable de seguridad la tarea de importar los datos de ACCESS de forma manual.

- GAS ofrece informes para el administrador y jefes de departamentos para llevar un seguimiento del sistema.

- GAS tendrá un repositorio de troubleshooting siendo muy útil debido a la cantidad de sistemas operativos distintos que se mantienen.

- GAS puede contabilizar el tiempo que se dedica por servidor. Excelente herramienta para los jefes de departamentos.

- En nuestra red contamos con servidores con distintos sistemas operativos, esto quiere decir que cada SO tiene vulnerabilidades distintas y los parches de seguridad a aplicar son distintos

2.2.4 Necesidades básicas de un Gestor de Administrador de Sistemas

GAS servirá a los administradores para documentar las tareas realizadas en los servidores y tener un histórico de problemas resueltos, lo que en el argot técnico llamamos troubleshooting.

A los jefes de departamento para ayudar a tomar decisiones sobre la carga de trabajo de sus administradores, optimizar recursos y controlar el aspecto de seguridad.

GAS permite optimizar el mantenimiento de los servidores alargando de esa forma su ciclo de vida.

3. Modelo de Análisis

3.1 Identificación de Actores

3.1.1 Diagrama de Actores

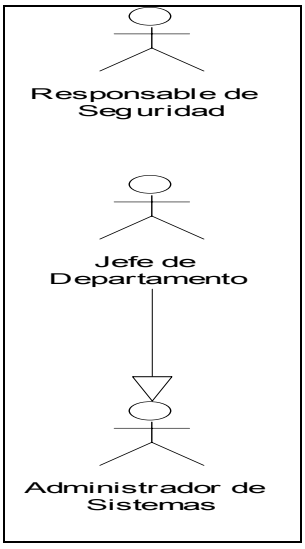


Ilustración A: Diagrama de Actores

3.1.2 Descripción Textual de los Actores

- Responsable de Seguridad: Esta persona se encargar del correcto funcionamiento del programa hjnetchk. Controlando la cantidad de parches que los servidores tienen que tener instalados. No tiene cargo de supervisión.
- Administrador de Sistema: Es el que va a trabajar de forma directa con GAS ya que tendrá a su cargo un número de servidores. Recibirá alerta no solo de los parches a instalar sino también de actualización de antivirus, y demás mantenimiento que el servidor requiera. Es el encargado de documentar la base de datos de GAS.

- Jefe de Departamento: El programa GAS proporcionará al Jefe de Departamento de información para la toma de decisiones. Podrá asignar servidores a los administradores. Incluir tareas e incluso asignar los tiempos a dichas tareas para que el administrador las ejecute.

3.2 Identificación de Casos de Uso

Los casos de Uso son descripciones de la funcionalidad del sistema independiente de la implementación. Permiten definir los límites del sistema y las relaciones entre el sistema y el entorno. Estos Casos de Uso se han determinado observando y precisando, actor por actor, las secuencias de interacción, los escenarios, desde el punto de vista del usuario.

3.2.1 Paquetes de Análisis

Dentro del paquete de casos de uso que forma el modelo de análisis, se han separado los casos de uso en tres paquetes distintos de la forma:

- Gestor de Password: Los casos de usos que se incluye en este paquete son comunes para todos los actores. Antes de entrar en el sistema y comenzar con la funcionalidad del programa todos los usuarios con independencia del rol que desempeñe debe identificarse con un nombre de usuario y una contraseña. El programa ajustará las vistas según el rol del usuario.
  El Jefe de departamento será el que de de alta y asigne el rol al usuario en el caso de uso *Alta de Usuarios* en el paquete Gestor Administración.

- Gestor de Vulnerabilidades: Este paquete contiene aquellos casos de uso que cubre las funcionalidades relacionadas directamente con el escaneo de los servidores mediante el programa hfjetchk para detectar posibles vulnerabilidades. Estos casos de uso contemplarán las vulnerabilidades así como su resolución. La resolución puede ser mediante la instalación de parches de seguridad o mediante cambios en la configuración del sistema operativo del servidor. Estos cambios son parte de las tareas que el administrador documentara en el Gestor de Administración. El Actor principal y casi único de este gestor será el Responsable de seguridad. El Administrado es solo un receptor de la alerta que GAS le enviará de dos formas: mediante un correo electrónico y mediante una alerta dentro del mismo programa con la vulnerabilidad y la forma de tratarla. Esta alerta continuará hasta que el Administrador documente que ya ha resuelto la vulnerabilidad, con fecha y tiempo empleado.

- Gestor Administración: En este paquete se gestiona todas las tareas propias de la administración de sistemas: El Administrador tendrá un Scheduler donde se va a programar las tareas de mantenimiento a realizar en el servidor con fecha de comienzo, frecuencia y duración. También están documentadas las actualizaciones de software y los incidentes con la respectiva solución y el tiempo empleado.
  El Jefe de departamento podrá realizar informes para conocer el estatus de los servidores y para saber la media del tiempo empleado en el mantenimiento por cada uno de ellos. También dará da alta a los administradores del sistema, asignándole un número de servidores.

El siguiente esquema muestra los Subsistemas a desarrollar en la aplicación.
El subsistema Gestor Password es el de conexión que da acceso al resto de subsistemas. Están todos enlazados.

La división de los subsistemas se ha realizado pensando en el área funcional de cada uno de ellos y en la independencia entre ellos, aunque para que funcione el sistema, todos ellos se tendrán que integrar como un único engranaje.

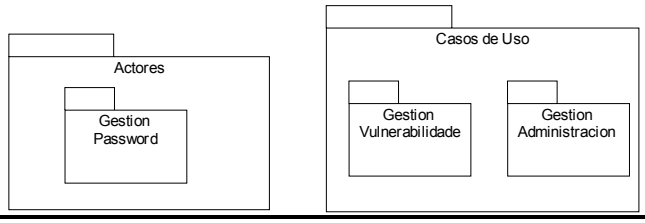


Ilustración B: Paquete de Análisis

3.2.2 Casos de Uso Subsistema Gestor de Password

En este subsistema van a intervenir todos los actores, ya que los usuarios tendrán que identificarse en el sistema para poder trabajar en el. Una vez identificado el usuario según su rol tendrá una vista del programa de forma especifica.

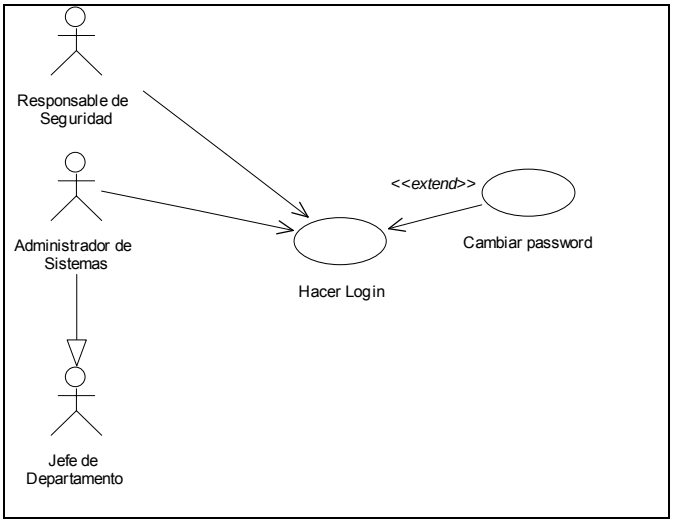


Ilustración C: Casos de Uso Gestor Password

3.2.3 Casos de Uso Subsistema Gestor de Vulnerabilidad

- Responsable de Seguridad: Inicia los Casos de Uso para la gestión de las vulnerabilidades en los servidores.
- Administrador de Sistemas: En este apartado el Administrador es tan sólo el receptor del correo electrónico enviado por GAS. En este subsistema es donde el Administrador recibe la notificación de las vulnerabilidades que el servidor tiene y cómo corregirlas.

El diagrama de los casos de uso es el siguiente:

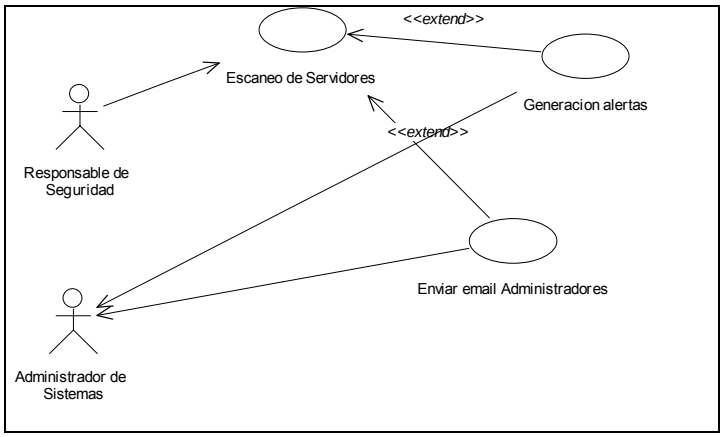


Ilustración D: Casos de Uso Gestor Vulnerabilidad

3.2.4 Casos de Uso Subsistema Gestor Administración

- Administrador de Sistemas: El programa GAS esta especialmente diseñado para liberar al administrador de recordar las tareas a realizar. Este actor en este escenario podrá documentar la información de los servidores y la resolución de problemas. También es en este subsistema donde se documenta cómo se han resuelto las vulnerabilidades en el servidor.
- Jefe de Departamento: El programa GAS será fundamental para los Jefes de Departamentos para obtener información no sólo de los servidores sino de las tareas que se realizan en ellos con los tiempos, de esta forma se podrá maximizar los recursos humanos y técnicos.

El diagrama de los casos de uso es el siguiente:

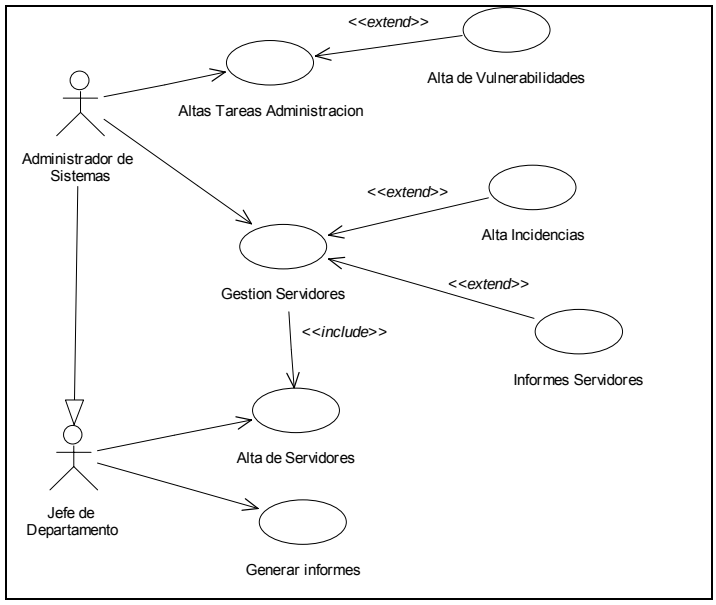


Ilustración E: Casos de Uso Gestor Administración

3.3 Descripción Textual de los Casos de Uso

3.3.1 Casos de Uso del Subsistema Gestor de Password

Tabla 1: Casos de Uso "Hacer Login" ID [001]

Hacer Login	
Identificador. 001	Prioridad[1..10] 10
DESCRIPCION Todos los usuarios del sistema tendrán que identificarse previamente.	
FLUJO BASICO Una vez el sistema detecte el rol, la aplicación nos mostrará la vista correspondiente: 1. - Si el usuario es el Responsable de Seguridad, el sistema iniciará el Gestor Vulnerabilidad 2. - Si el usuario es Administrador, el sistema iniciará el Gestor Administración con restricciones. 3. - Si el usuario es Jefe de Departamento, el sistema iniciará el Gestor Administración sin restricciones.	
FLUJOS ALTERNATIVOS No hay Flujos alternativos. La única forma de entrar en GAS es mediante la identificación del usuario.	
PRECONDICIONES Para acceder al sistema, el usuario debe tener una cuenta abierta por el Jefe de Departamento.	
POSCONDICIONES **N/A**	
PUNTOS DE INCLUSION **N/A**	
PUNTOS DE EXTENSION Este caso de uso extiende al Caso de Uso ***"Cambiar Password" ID[002]***	

Tabla 2: Casos de Uso "Cambiar Password ID [002]

Cambiar Password	
Identificador. 002	Prioridad[1..10] 8
DESCRIPCION Todos los usuarios podrán cambiar sus contraseñas.	
FLUJO BASICO El usuario ha tenido previamente que identificarse en el sistema.	
FLUJOS ALTERNATIVOS No hay flujo alternativo. El Usuario debe haberse identificado con anterioridad.	
PRECONDICIONES El usuario ha tenido que venir del Caso de Uso ***"Hacer Login" [ID. 001]***	
POSCONDICIONES **N/A**	
PUNTOS DE INCLUSION **N/A**	
PUNTOS DE EXTENSION **N/A**	

3.3.2 Casos de Uso del Subsistema Gestor de Vulnerabilidad

Tabla 3: Casos de Uso "Escaneo de Servidores ID [003]

Escaneo de Servidores	
Identificador. 003	Prioridad[1..10] 1
DESCRIPCION Este Caso de Uso es el que inicia toda la Gestión de Vulnerabilidad. Inicia el programa hjnetchk que es el que va a escanear todos los servidores de la red.	
FLUJO BASICO El Responsable de Seguridad inicia y es responsable de este caso de uso.	
FLUJOS ALTERNATIVOS No existe.	
PRECONDICIONES El usuario debe estar en el rol de Responsable de Seguridad. Debe haber hecho el login primero.	
POSCONDICIONES **N/A**	
PUNTOS DE INCLUSION **N/A**	
PUNTOS DE EXTENSION Cuando se escanea los servidores se genera un listado de alertas con las vulnerabilidades y cómo resolverlas. Caso de Uso ***"Generación alertas" [ID 004]*** Por otro lado se envía un email a los usuarios con la misma información. Caso de Uso ***"Enviar Email Administradores" [ID 005]***	

Tabla 4: Casos de Uso "Generación de Alertas" ID [004]

Generación de Alertas	
Identificador. 004	Prioridad[1..10] 10
DESCRIPCION El Administrador del Sistema cuando entra en el programa recibirá unos eventos de alerta sobre los servidores con vulnerabilidades y con la resolución.	
FLUJO BASICO La alerta se genera una vez se ha escaneado los servidores y se ha encontrado vulnerabilidades.	
FLUJOS ALTERNATIVOS	
PRECONDICIONES El sistema debe encontrar vulnerabilidades para generar las alertas.	
POSCONDICIONES Una vez encontrada generada las alertas el sistema notifica al Administrador del Sistema.	
PUNTOS DE INCLUSION **N/A**	
PUNTOS DE EXTENSION **N/A**	

Tabla 5: Casos de Uso "Envío Email Administradores" ID [005]

Envío Email Administradores	
Identificador. 005	Prioridad[1..10] 10
DESCRIPCION Una vez se ha encontrado vulnerabilidades en el servidor, el sistema GAS envía un correo electrónico al Administrador responsable del servidor notificándole la vulnerabilidad y la forma de resolverla.	
FLUJO BASICO El email se genera y envía una vez se ha escaneado los servidores y se ha encontrado vulnerabilidades.	
FLUJOS ALTERNATIVOS **N/A**	
PRECONDICIONES El sistema debe encontrar vulnerabilidades para generar el email.	
POSCONDICIONES **N/A**	
PUNTOS DE INCLUSION **N/A**	
PUNTOS DE EXTENSION **N/A**	

3.3.3 Casos de Uso del Subsistema Gestor Administración

Tabla 6: Casos de Uso "Alta Tareas de Administración" ID [006]

Alta Tareas de Administración	
Identificador. 006	Prioridad[1..10] 10
DESCRIPCION Permite dar de alta las tareas que hay que realizar en un servidor.	
FLUJO BASICO El Administrador inicia este caso de uso. Da de alta las tareas diarias, semanales y mensuales que hay que realizar en los servidores: 1.- Verificación de Backups 2.- Realización de GHOSTs 3.- Instalar y configurar periféricos 4.- Instalación de Sistema Operativos(SO) 5.- Actualización de SO 6.- Manejo de Usuario 7.- Conexiones a Internet 8.- Instalación de Software 9.- Configuración de servicios de ref 10.- Control de la seguridad en la red y en el servidor 11.- Conexión a Internet 12.- Configuración cuentas de email y accesos a Internet 13.- Diagnóstico de Hardware	
FLUJOS ALTERNATIVOS Se puede llegar a este Caso de Uso realizando el acceso con el rol de Jefe de	

Departamento.
PRECONDICIONES El Administrador debe haberse identificado en el sistema GAS.
POSCONDICIONES **N/A**
PUNTOS DE INCLUSION **N/A**
PUNTOS DE EXTENSION El Administrador da de alta las vulnerabilidades encontradas por el Gestor de Vulnerabilidades con la forma de resolverlas y el tiempo empleado. Extiende al Caso de Uso ***"Alta de Vulnerabilidades" ID[008]***

Tabla 7: Casos de Uso "Gestión de Servidores" ID [007]

Gestión Servidores	
Identificador. 007	Prioridad[1..10] 10
DESCRIPCION Una vez el servidor esté dado de alta por el Jefe de Departamentos y asignado al administrador, éste puede mantener la información de las máquinas. Aquí también se incluye toda la gestión sobre las tareas de administración. Este Gestor incluirá un Scheduler donde el Administrador podrá planificar las tareas durante la semana.	
FLUJO BASICO El Administrador inicia este caso de uso: 1. Una vez todas las tareas estén dadas de altas. El Administrador podrá planificar en un calendario cuando las quiere realizar. El scheduler le servirá de recordatorio. 2. El Administrador debe documentar todo el mantenimiento realizado en el servidor en el programa GAS para luego poder realizar los informes.	
FLUJOS ALTERNATIVOS El Jefe de Departamento también puede gestionar los Servidores una vez le da de alta en el Caso de Uso ***"Alta de Servidores" ID[009]***	
PRECONDICIONES El servidor debe estar dado de alta en el sistema por el Jefe de Departamento y asignado a un Administrador.	
POSCONDICIONES **N/A**	
PUNTOS DE INCLUSION Caso de Uso ***"Alta de Servidores" ID[009]***	
PUNTOS DE EXTENSION Caso de Uso ***"Alta de Incidencias" ID[011]*** Caso de Uso ***"Informes Sistemas" ID[012]***	

Tabla 8: Casos de Uso "Alta Vulnerabilidades" ID [008]

Alta Vulnerabilidades	
Identificador. 008	Prioridad[1..10] 10
DESCRIPCION El Administrador da de alta las vulnerabilidades encontradas por el Gestor de Vulnerabilidades con la forma de resolverlas y el tiempo empleado.	
FLUJO BASICO La documentación y el tratamiento de las vulnerabilidades mediante parches u otra modificación de la configuración del sistema forman parte del trabajo de administración del servidor.	
FLUJOS ALTERNATIVOS Se puede llegar a este Caso de Uso realizando el acceso con el rol de Jefe de Departamento.	
PRECONDICIONES El Gesto de Vulnerabilidades deber haber encontrado vulnerabilidades en el sistema y se le ha tenido que notificar previamente al administrador.	
POSCONDICIONES **N/A**	
PUNTOS DE INCLUSION **N/A**	
PUNTOS DE EXTENSION **N/A**	

Tabla 9: Casos de Uso "Alta de Servidores" ID [009]

Alta de Servidores	
Identificador. 009	Prioridad[1..10] 10
DESCRIPCION En este caso de uso el Jefe de Departamento da de alta al servidor y le asigna un administrador.	
FLUJO BASICO El Administrador inicia este caso de uso: 1. El Administrador da de alta las tareas diarias, semanales y mensuales que hay que realizar en los servidores.	
FLUJOS ALTERNATIVOS No hay flujos alternativos. El Jefe de Departamento es el que da de alta el servidor y lo asigna a un administrador.	
PRECONDICIONES **N/A**	
POSCONDICIONES **N/A**	
PUNTOS DE INCLUSION **N/A**	
PUNTOS DE EXTENSION **N/A**	

Tabla 10: Casos de Uso "Generar Informes" ID [007]

Generar Informes	
Identificador. 010	Prioridad[1..10] 7
DESCRIPCION En este Caso de Uso se genera los informes para los Jefes de Departamentos: 1. - Tiempo medio dedicado de mantenimiento por servidor 2. - Tiempo medio de trabajo por Administrador. 3. - Status de los servidores.	
FLUJO BASICO El Jefe de Departamento inicia este caso de uso.	
FLUJOS ALTERNATIVOS Sólo los Jefes de Departamento están autorizados para ver estos tipos de informes.	
PRECONDICIONES Para acceder al sistema se tiene que acceder con el rol de Jefe de Departamento.	
POSCONDICIONES **N/A**	
PUNTOS DE INCLUSION **N/A**	
PUNTOS DE EXTENSION **N/A**	

Tabla 11: Casos de Uso "Alta Incidencias" ID [011]

Alta de Incidencias	
Identificador. 011	Prioridad[1..10] 9
DESCRIPCION El Administrador va a dar de alta las incidencias (troubleshooting) que se produzcan en el sistema.	
FLUJO BASICO Este Caso de Uso se extiende del Caso de Uso ***"Gestión Servidores" ID007***, donde se dará de alta cada vez que ocurra alguna incidencia. Se documenta la fecha de la incidencia, qué lo causó, cómo se solucionó y el tiempo invertido. Se obtiene un repositorio de incidencias por servidor.	
FLUJOS ALTERNATIVOS Se puede llegar a este Caso de Uso desde el Caso de Uso ***"Alta de Servidores" ID[009]*** si se accede a GAS con el rol de Jefe de Departamento.	
PRECONDICIONES El Administrador debe haberse identificado en el sistema GAS.	
POSCONDICIONES **N/A**	
PUNTOS DE INCLUSION **N/A**	
PUNTOS DE EXTENSION **N/A**	

Tabla 12: Casos de Uso "Informe Servidores" ID [012]

Informe Servidores	
Identificador. 012	Prioridad[1..10] 7
DESCRIPCION En este Caso de Uso se generaran los informes para los Administradores responsables de los sistemas. 1.- Listado de servidores que cumplen/incumplen con las normas de seguridad. 2.- Listado de tareas de mantenimiento realizadas/pendientes de realizar en el servidor.	
FLUJO BASICO Este Caso de Uso se extiende del Caso de Uso ***"Gestión Servidores" ID007***	
FLUJOS ALTERNATIVOS Se puede llegar a este Caso de Uso desde el Caso de Uso ***"Alta de Servidores" ID[009]*** si se accede a GAS con el rol de Jefe de Departamento.	
PRECONDICIONES El Administrador debe haberse identificado en el sistema GAS. El Administrador sólo verá los informes de los sistemas que están bajo su responsabilidad.	
POSCONDICIONES **N/A**	
PUNTOS DE INCLUSION **N/A**	
PUNTOS DE EXTENSION **N/A**	

3.4 Relación de Casos de Uso principales.

En la siguiente tabla se muestran los casos de uso del subsistema, con los siguientes datos:

- Código interno Caso de Uso.
- Nombre del Caso de Uso.
- A qué caso de uso extiende.
- A qué caso incluye
- Caso de uso por el que es extendido.

Tabla 13: Relación de Casos de Uso

Codigo	Caso de Uso	Incluye	Extiende a	Es Extendido por
ID001	***Hacer Login***		***ID002***	
ID002	***Cambiar Password***			***ID001***
ID003	***Escaneo de Servidores***		***ID004*** ***ID005***	
ID004	***Generación de Alertas***			***ID003***
ID005	***Envío Email Administradores***			***ID003***
ID006	***Alta Tareas de Administración***		***ID008***	
ID007	***Gestión de Servidores***	***ID009***	***ID011*** ***ID012***	
ID008	***Alta de Vulnerabilidades***			***ID006***
ID009	***Alta de Servidores***			
ID010	***Generar Informes***			
ID011	***Alta de Incidencias***			***ID007***
ID012	***Informe Servidores***			***ID007***

3.5 Requisitos Hardware y Software

La aplicación GAS va a estar disponible tanto para distribuciones Windows 2000 como XP. El software cliente será el navegador del usuario.
El Hospital cuenta con un servidor Oracle dedicado al almacenaje de bases de datos. El software necesario en el área servidor:

- Programación en Visual Basic .NET
- Servidor Web con IIS
- Como SGBD Relacional se utilizará Oracle.
- Desarrollo de páginas web Visual Basic .Net

El Hardware donde se instalará la aplicación será en un Pentium Intel IV.

4. Introducción a la Fase de Diseño

Esta es la última fase del estudio del análisis de la aplicación GAS. En esta Fase se profundiza y revisa el análisis realizado anteriormente en el Modelo de Análisis y se desarrolla el estudio del diseño de la aplicación. Este trabajo se caracteriza por tener un desarrollo integrado, es decir, los resultados de cada fase del proyecto han servido como datos de entrada para la fase siguiente.

4.1 Organización de alto nivel del Modelo de Diseño

La solución de diseño propuesta está formada por 4 paquetes UML, cómo muestra el siguiente diagrama:

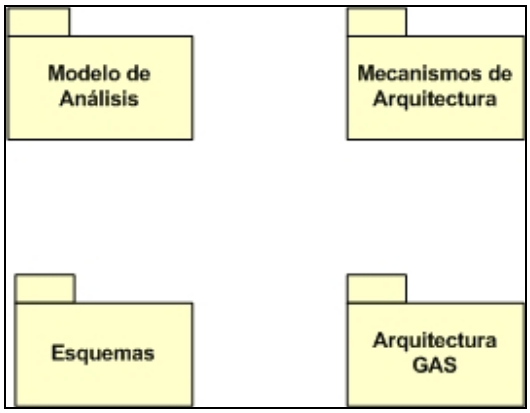


Ilustración F: Modelo de Diseño

- El “Modelo de Análisis del Diseño” contiene todas las realizaciones de los casos de uso a diseñar. Por cada realización de un caso de uso que se desarrolló durante el análisis, se ha creado una nueva realización más detallada, a nivel diseño, conteniendo los diagramas de colaboración, y un diagrama de Actividades y de secuencia de los casos de uso más complicado

- El paquete de “Mecanismos de Arquitectura” contiene los mecanismos definidos para la propuesta de arquitectura que se hace; básicamente se trata de la definición

del mecanismo de persistencia con la particularidad de la utilización de XML para la comunicación.

- El Paquete “Esquemas”, contiene los esquemas físicos de la base de datos, tablas e índices para ellas que darán soporte a la aplicación.

- El paquete “Arquitectura GAS” contiene la definición de la arquitectura.

4.2 Modelo de Análisis del Diseño.

4.2.1 Diagrama de Actividad

Se realiza un Diagrama de Actividad con la intención de comprender cómo se comunica y se relacionan los tres subsistemas dentro de la aplicación GAS:

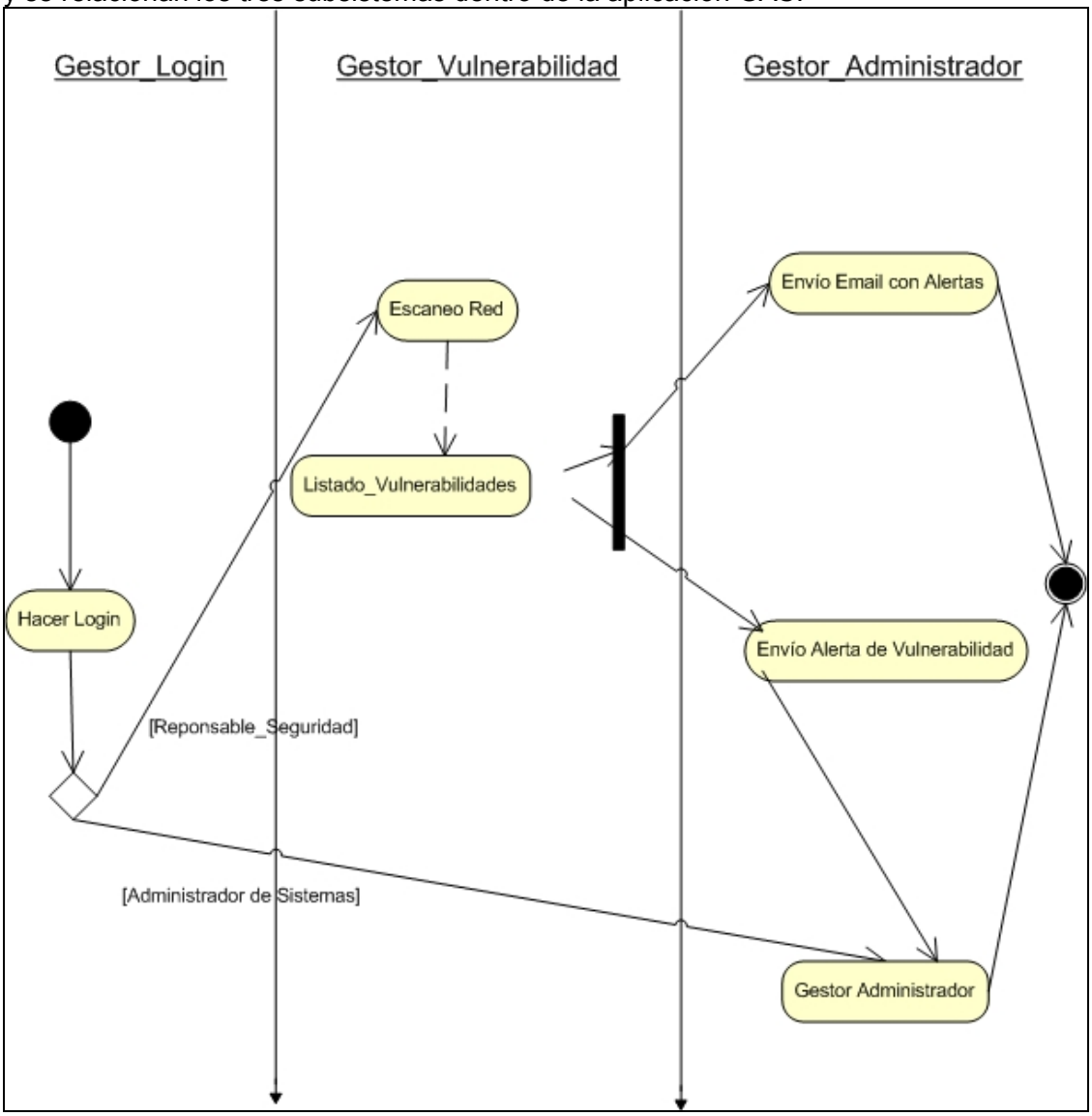


Ilustración G: Diagrama de Actividad

4.2.2 Diagramas de colaboración y secuencia

Especificación de las clases de análisis

Para describir los flujos de trabajo que tienen lugar en el sistema se va a emplear diagramas de interacción. Se han elegido diagramas de colaboración y diagramas de secuencia. Con los diagramas de colaboración podemos ver como colaboran los objetos en un instante dado. Se ha realizado un diagrama de colaboración por cada caso de uso. Con los diagramas de secuencia podemos ver para un determinado flujo de trabajo los objetos que intervienen y los mensajes que se envían entre si recalcando su ordenación temporal. Se ha realizado el diagrama de secuencia de los casos de uso más complicados.

Se ha revisado todos los casos de uso realizados en el modelo anterior. En el subsistema de Administración se ha generado un caso de uso más. Se detallará la nueva estructura dentro del subsistema en cuestión.

- **Subsistema Gestor de Password.**

1. **Caso de uso: *"Hacer login"***

   Este caso de uso extiende al caso de uso *"Cambiar password"*, vamos a incluirlo en el diagrama de colaboración.

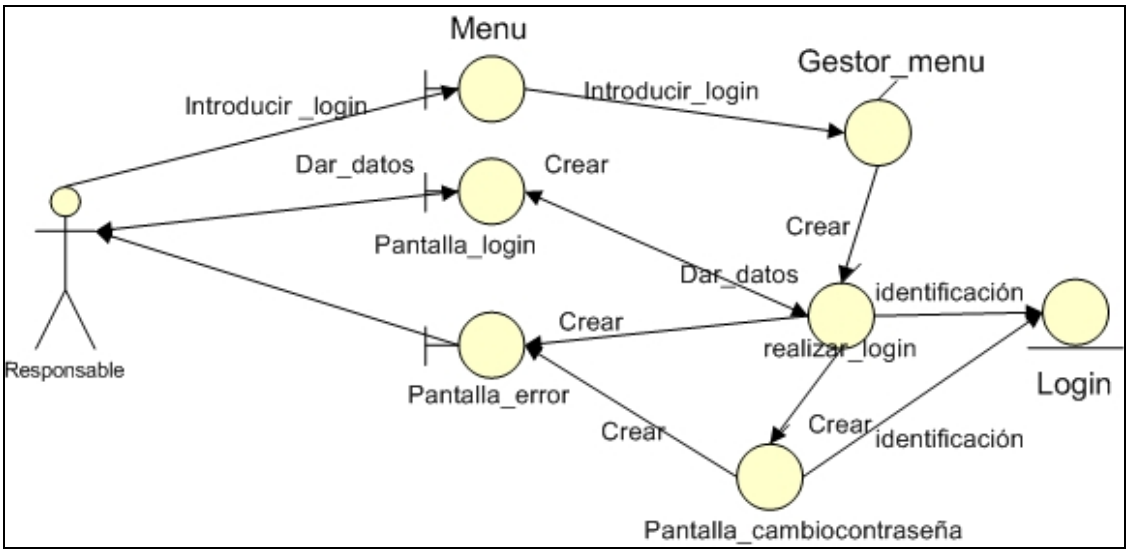


Ilustración H: Diagrama colaboración *"Hacer Login"*

- **Subsistema Gestor de Vulnerabilidad**

2. Escaneo de servidores

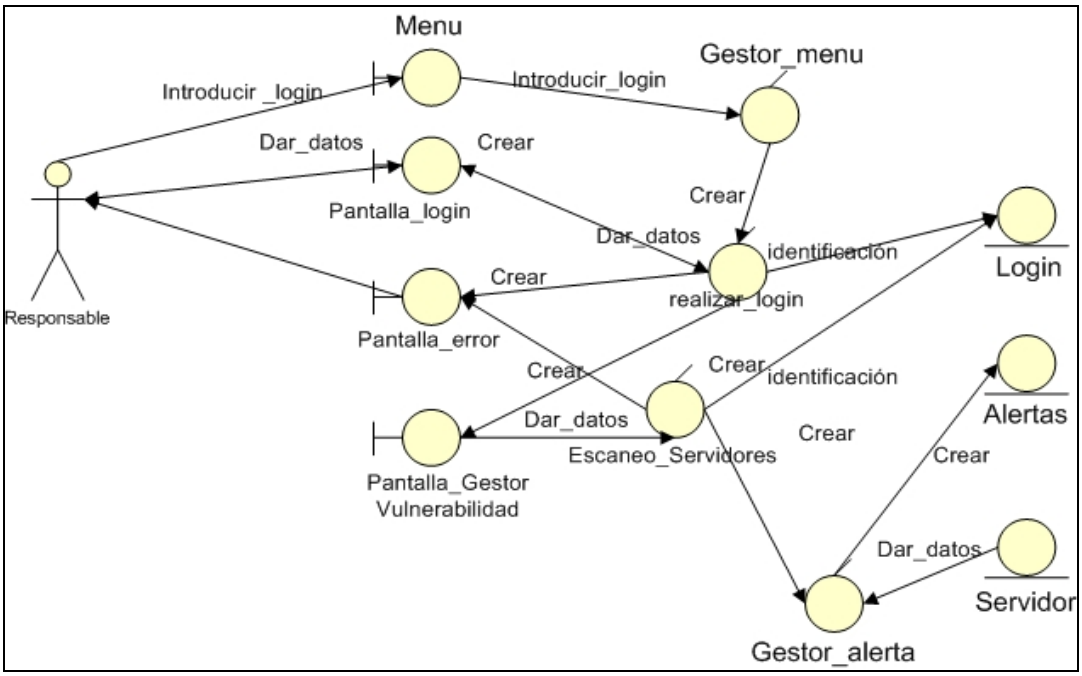


Ilustración I: Diagrama colaboración *"Escaneo servidores"*

3. Caso de uso: "Generación de alertas"

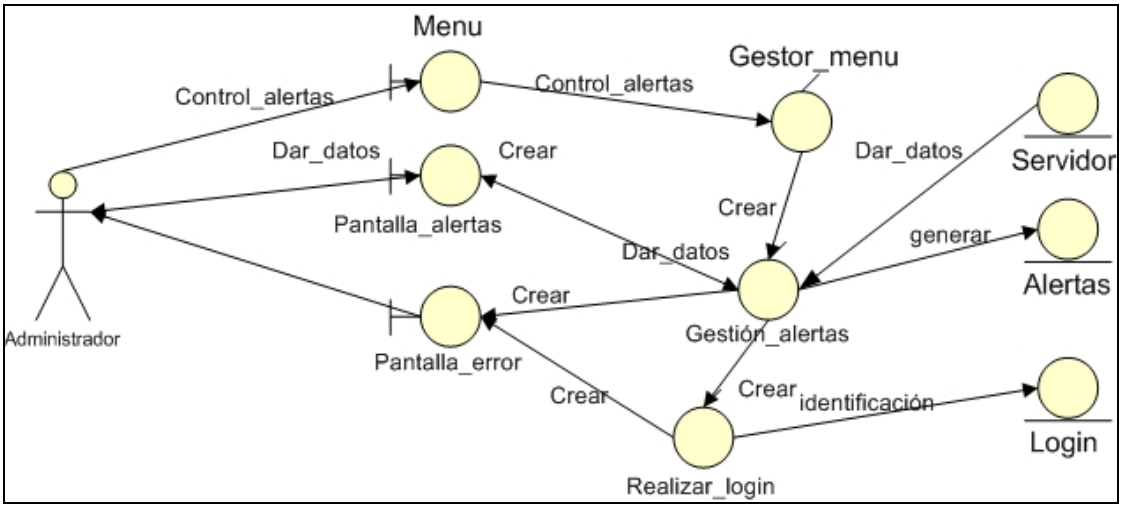


Ilustración J: Diagrama colaboración *"Generación alertas"*

4. Caso de Uso: "Envío Email Administradores"

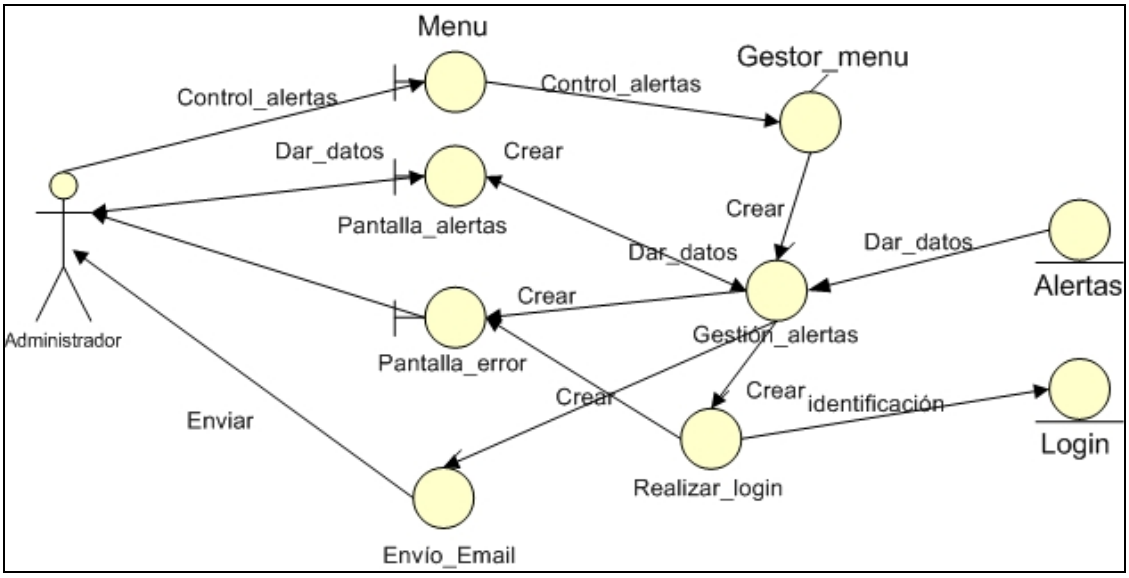


Ilustración K: Diagrama colaboración *"Envío Email Administradores"*

- **Subsistema Gestor Administración**

5. Caso de Uso: "Alta Tareas Administración"

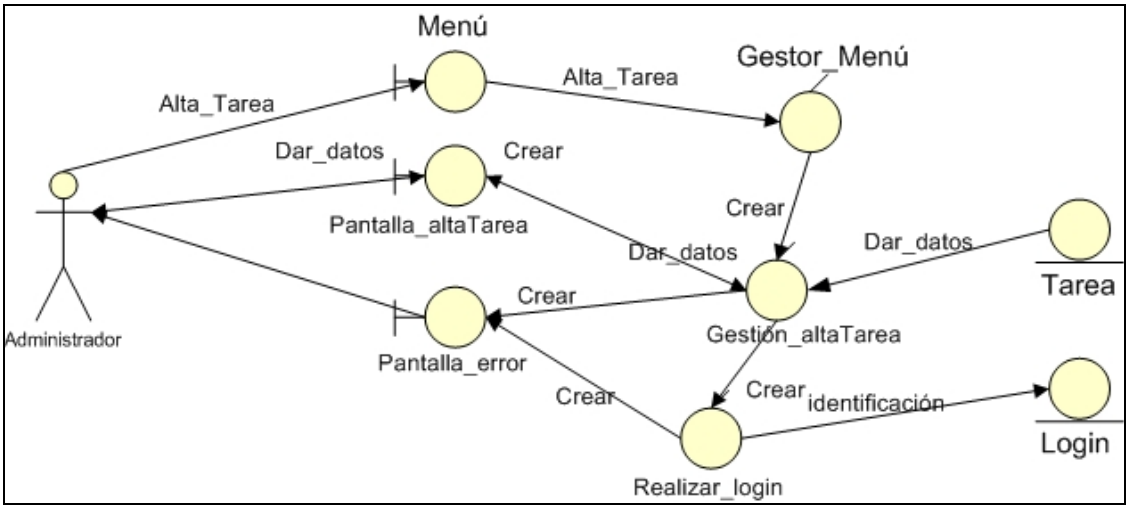


Ilustración L: Diagrama colaboración *"Alta Tareas Administración"*

6. Caso de uso: "Gestión Servidores"

Este caso de uso incluye al caso de uso "Alta de Servidores" y extiende a los casos de uso "Alta de Incidencias" e "Informes Servidores".

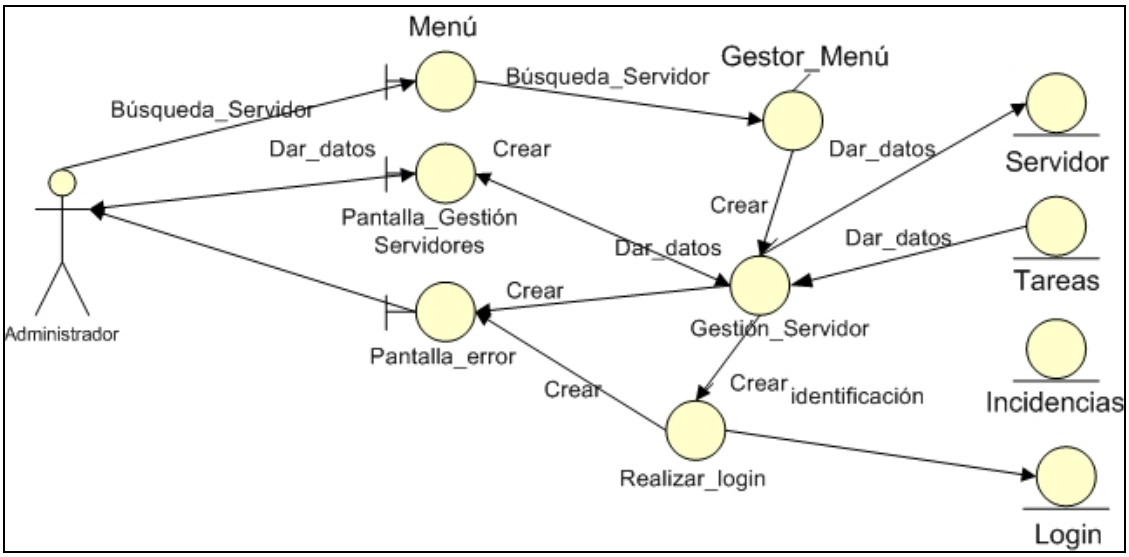


Ilustración M: Diagrama colaboración *"Gestión de servidores"*

Diagrama de Secuencia

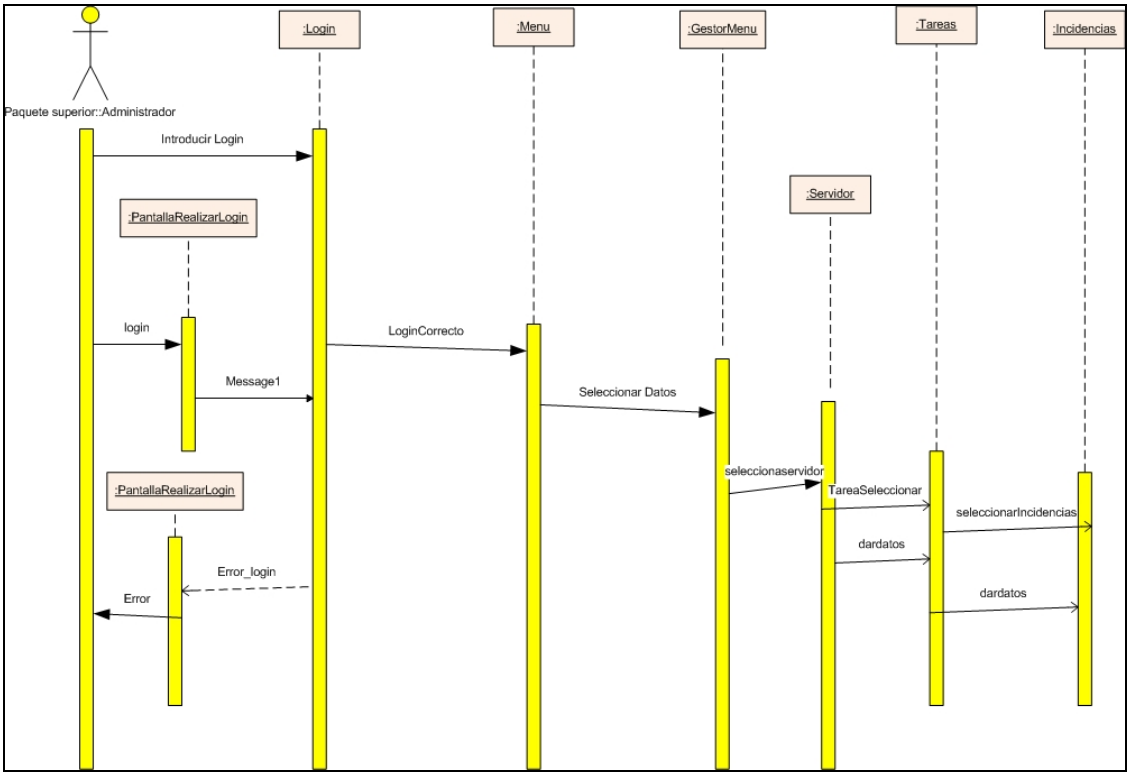


Ilustración N: Diagrama de secuencia *"Gestión de servidores"*

7. Caso de Uso: "Alta de Vulnerabilidades"

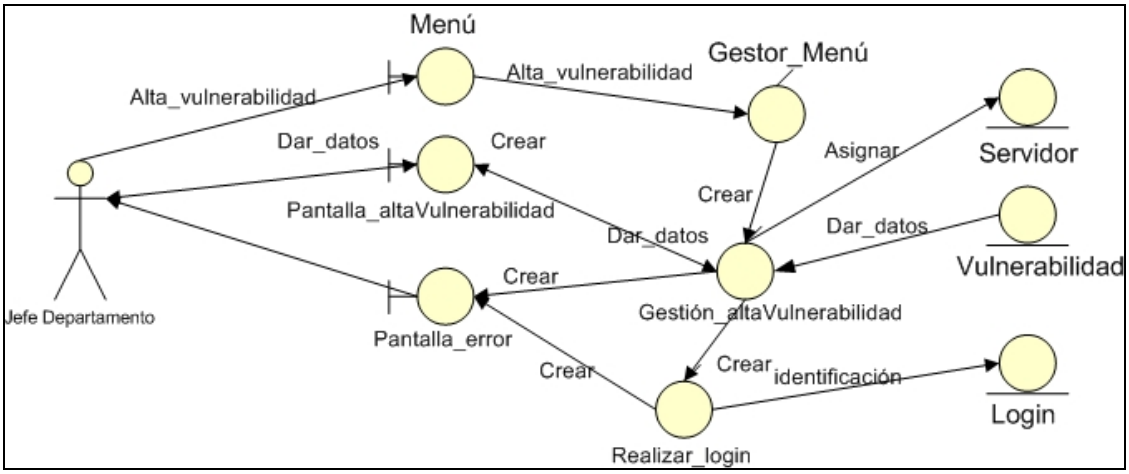


Ilustración O: Diagrama colaboración *"Alta de Vulnerabilidades"*

8. Caso de Uso: "Alta de Servidores"

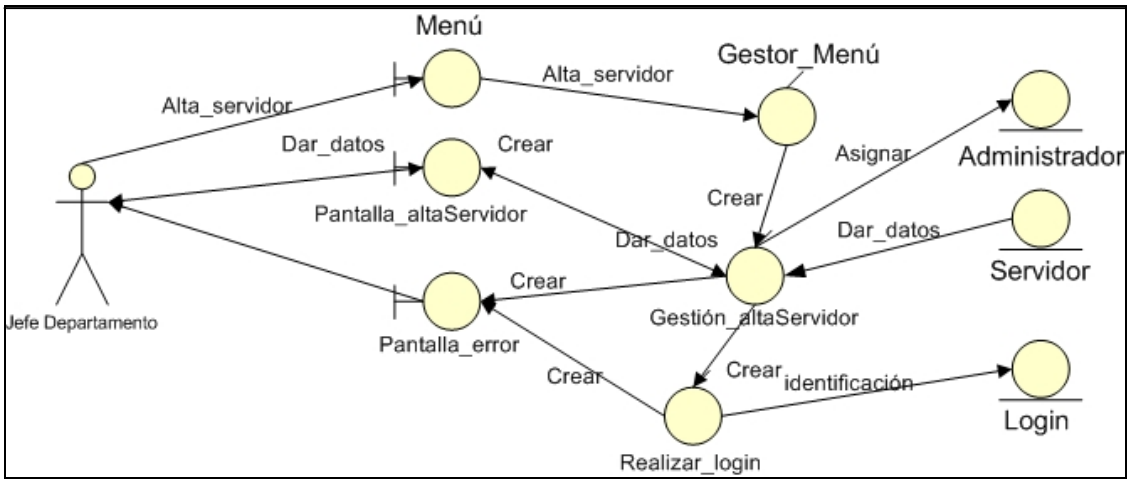


Ilustración P: Diagrama colaboración *"Alta de servidores"*

9. Caso de Uso: "Generar Informes"

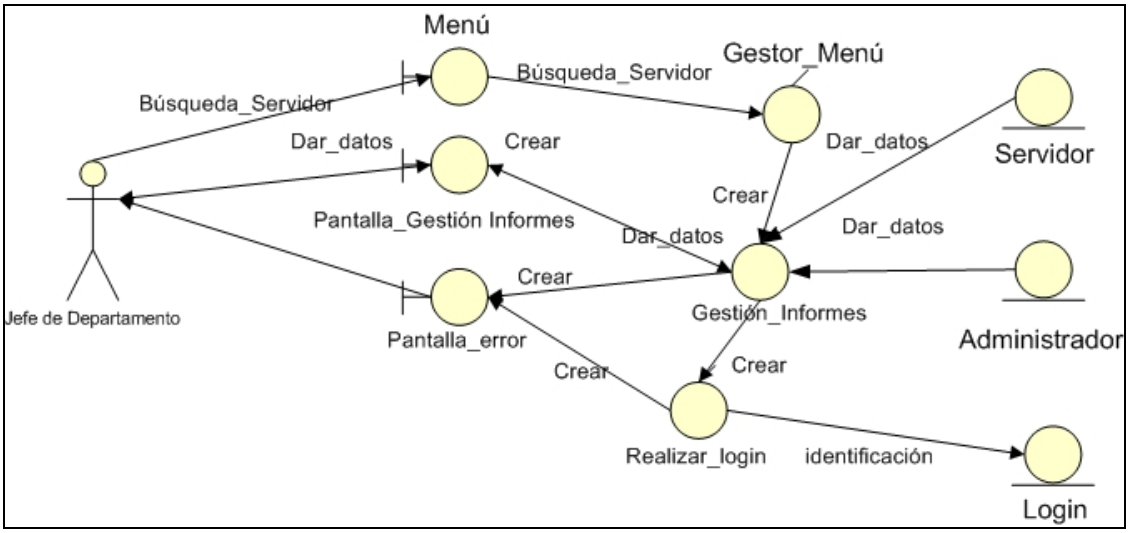


Ilustración Q: Diagrama colaboración *"Generar Informes"*

10. Caso de Uso: "Alta de Incidencias"

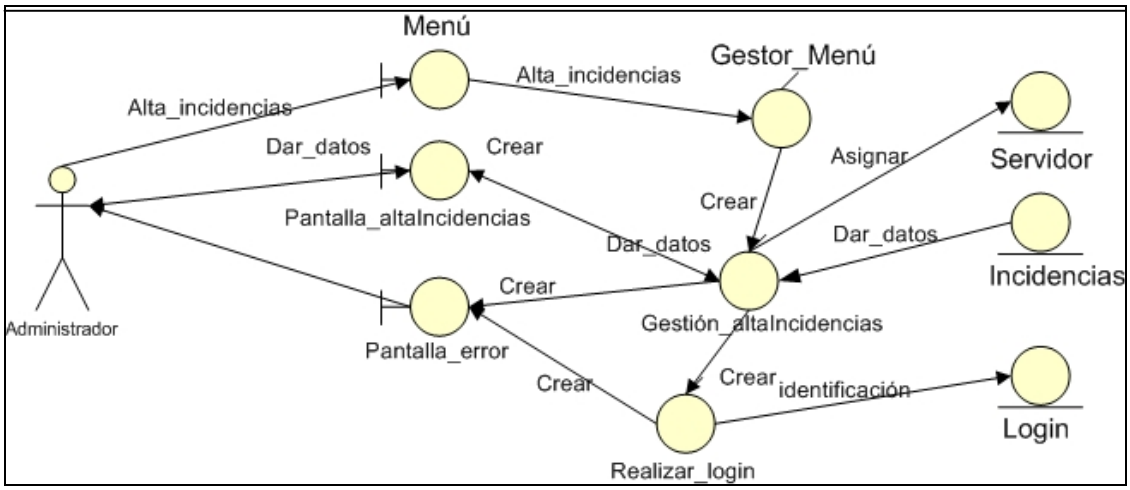


Ilustración R: Diagrama colaboración *"Alta de Incidencias"*

11. Caso de Uso: "Informe Servidores"

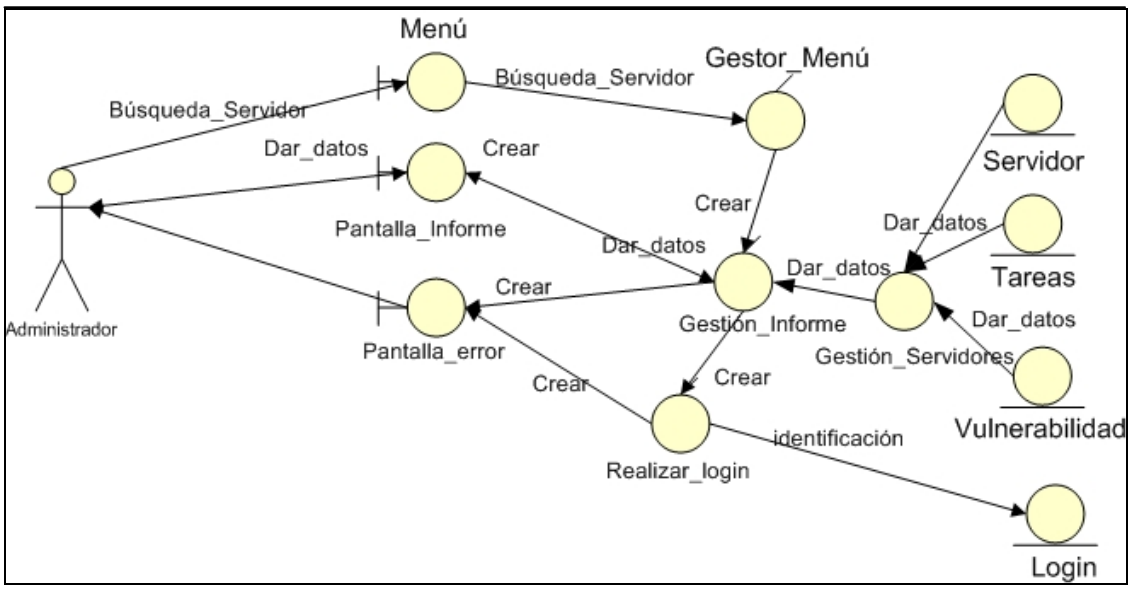


Ilustración S: Diagrama colaboración *"Informe de Servidores"*

Diagrama de secuencia

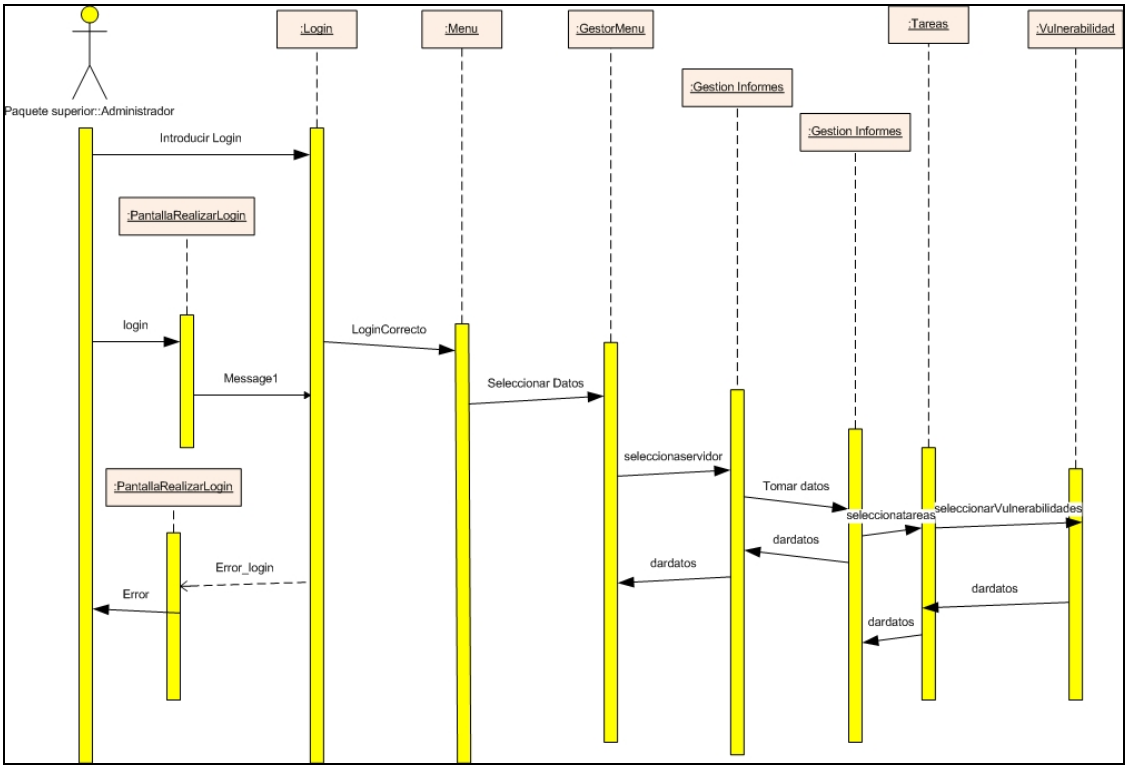


Ilustración T: Diagrama de secuencia *"Informe de servidores"*

12. Caso de Uso: "Alta Responsables"

Este caso de uso es nuevo y surge a partir de la revisión del modelo de casos de uso.

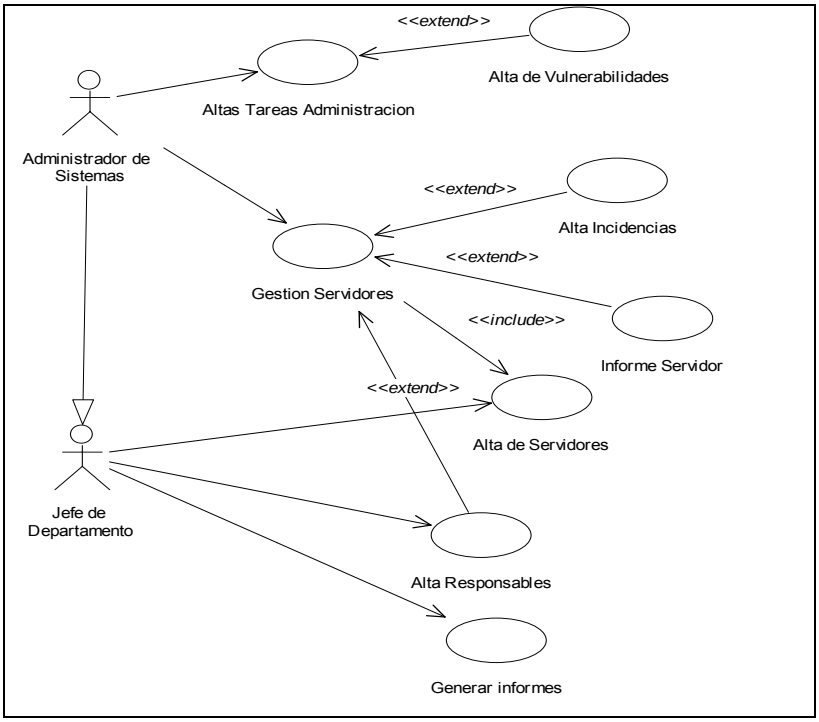


Ilustración U: Diagrama de Casos de Uso *"Alta Responsable"*

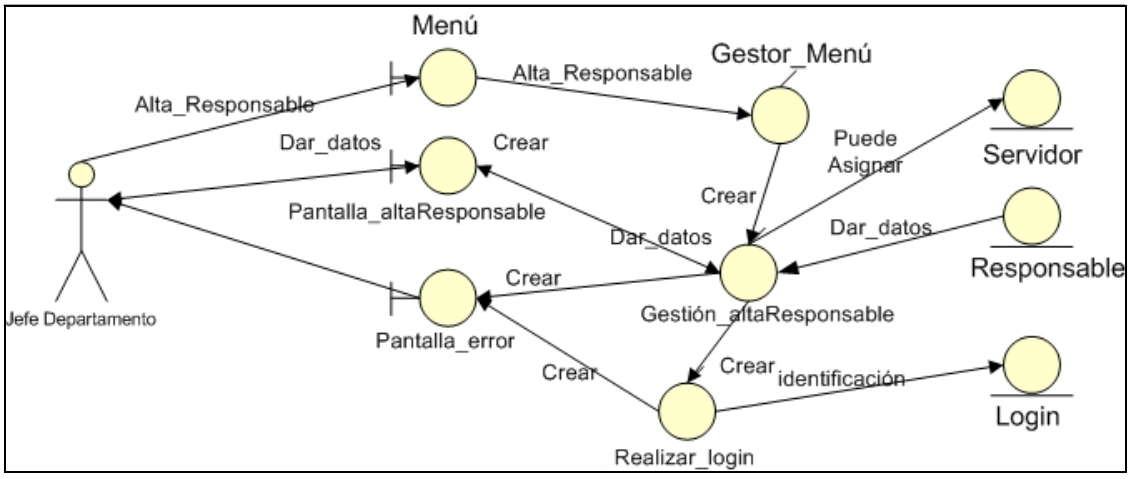


Ilustración V: Diagrama colaboración *"Alta Responsable"*

4.2.3 Identificación de las clases de entidades

Empezaremos por identificar las clases de entidades a partir de los casos de uso. Para cada caso de uso se indican las clases que se encuentran; cuando una clase ya se había identificado en un caso de uso anterior, su nombre va seguido de un asterisco, mientras que las clases que son dudosas van seguidas de un interrogante.

Identificaremos las entidades por cada caso de uso de cada subsistema:

- **Casos de Uso Subsistema Gestor de Password**

1. Caso de Uso “Hacer Login”. Clases: Responsable
2. Caso de Uso “Cambiar Password”. Clases: Responsable*

- **Casos de Uso Subsistema Gestor de Vulnerabilidad**

3. Caso de Uso “Escaneo de Servidores”. Clases: Servidores
4. Caso de Uso “Generación de Alertas”. Clases: Servidores*, Vulnerabilidad
5. Caso de Uso “Envío Email Administradores”. Clases: Servidores*, Administrador*, Vulnerabilidad*

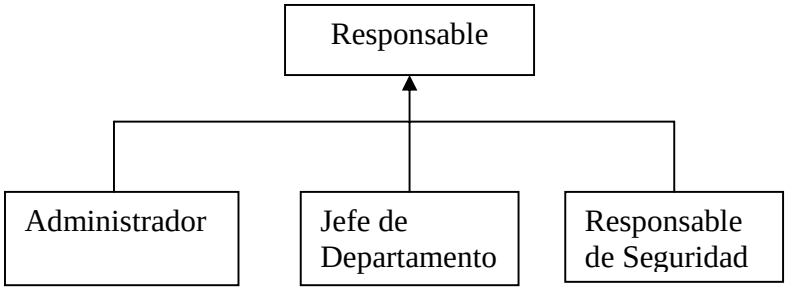
- **Casos de Uso Subsistema Gestor de Administración**

6. Caso de Uso “Alta Tareas de Administración”. Clases: Tareas
7. Caso de Uso “Gestión Servidores”. Clases: Servidores*, Tareas*, Incidencias
8. Caso de Uso “Alta de Vulnerabilidades”. Clases: Vulnerabilidad*
9. Caso de Uso “Alta de Servidores”. Clases: Servidores*, Administrador*
10. Caso de Uso “Generar Informes”. Clases: Tareas*, Vulnerabilidad*, Administrador*, Servidor*
11. Caso de Uso “Alta de Incidencias”. Clases: Incidencias*, Administrador*, Servidor*
12. Caso de Uso “Informe de Servidores”. Clases: Tareas*, Vulnerabilidad*, Administrador*, Servidor*, Incidencias*

4.2.4 Identificación de las relaciones entre clases. Herencia.

En un primer análisis identificamos tres actores. Mediante los casos de uso realizados hemos identificado una superclase que hemos llamado Responsable, en la que va a tener todos los datos de los usuarios. De esta clase heredan la clase Administrador, Jefe de Departamento, Responsable de Seguridad.

El diagrama sería:



4.2.5 Asociaciones.

El tema de la herencia visto en el apartado anterior se ha resuelto de la siguiente forma:

1.- Creamos una entidad llamada Rol donde vamos a guardar el rol de cada responsable, se crea una tabla intermedia donde relacionamos la entidad responsable con el rol bien sea Administrador, Jefe de Departamento o Responsable de Seguridad. Un responsable puede tener más de un rol.

2.- El Responsable se convierte en Administrador cuando se le adjudica un servidor por parte del Jefe de departamento. Un Administrador puede ser responsable de varios servidores, para ello creamos la tabla intermedia Respon_Servidor.

3.- El Jefe de Departamento tiene a su cargo a varios Responsables y tiene adjudicado un departamento; la tabla Respon_Departamento va a relacionar estas dos entidades.

El Responsable de Seguridad será un administrador más que puede o no tener asignado algún servidor. Tiene la tarea de poner en funcionamiento los programas de escaneo de servidores bien htfcneck o Resina y de generar los informes o alertas de vulnerabilidades.

4.2.6 Especificación de los atributos de las clases de entidades

Una vez estudiada y resuelta la herencia y las asociaciones entre las distintas clases entidades, pasamos a presentar los atributos de las clases de entidades. Los campos que componen la clave primaria identificada para cada entidad se han subrayado.
Se han especificado los atributos básicos e imprescindibles para cada entidad, pudiendo gracias a que el diseño de bases de datos es independiente al diseño de la aplicación poder incluir más campos si fuesen necesarios.

- **Responsable**

  NIF(string),Nombre(text),Apellidos (text), Email(txt), Fecha_ant(fecha), nom_usuario(text), clave(text)

  Aquí se almacena los datos de los usuarios del sistema.

- **Servidor**

Nombre (text), Server_IP (string), Server_Admin(text), Server_SO(text), Server_disco(text)

Se guardarán los datos de los Servidores habidos en la red.

- **Incidencias**

Inc_ID, Inc_Nombre(string), INC_Descripcion(string)

Se especifica el listado de incidencias que puede tener un servidor.

- **Jefe_Departamento**

Jef_DNI(num), Nombre_depart(text)

Aquí se asigna al responsable jefe de departamento

- **Vulnerabilidad**

Vuln_ID(num), Vul_Nombre(text), Vuln_Descrip(memo), Vul_Fix(text), Vuln_parche(text)

Se almacenan las vulnerabilidades como resultado del programa de escaneo.

- **Tareas**

Tarea_ID(num), Tarea_Nombre(text), Tarea_Descrip(memo), Tarea_tiempoest(hora)

En esta tabla se documenta todas las tareas a realizar por los administradores en los servidores.

- **Rol**

Rol_ID(num), Rol_Nombre(string)

Listados de Rol, en principio sólo hay tres, al realizar el diseño de esta forma, permitimos que la estructura sea flexible en el caso que se quiera incluir o modificar los roles

4.2.7 Transformación en Entidades de las Interrelaciones

A continuación detallamos los campos de las tablas a las que se traducen las interrelaciones identificadas en el diseño conceptual y que se traducen en entidades débiles. Las referencias entre tablas por medios de claves foráneas. Los campos que forman parte de la clave primaria de las tablas se presentan subrayados.

- **Admin. Server**

AdminDni(num), Server_Name(text)

En esta tabla intermedia es donde el Jefe de Departamento va a asignar el servidor a un administrador. Gracias a esta tabla intermedia, un administrador puede ser responsable de varios servidores.

- **Respon_Departamento**

DepartDNIJefe(num), RespoDNI(num)

Aquí se asignará a los responsables un Jefe de Departamento. Un Jefe de departmento tendrá a su cargo varios responsables.

- **Serv_Inc**

INC_ID(num), SERVER_Nombre(string), Inc_Fecha(fecha), Inc_Tiempo(hora), Inc_Comment(memo)

Un servidor puede sufrir en su ciclo de vida una, muchas o ninguna incidencia. Aquí se documenta el tipo de incidencia, cómo se ha resuelto y el tiempo empleado.

- **Rol_Asig**

Rol_ID(num), RolDniRespo(num)

Un responsable tendrá asignado un rol, mediante la tabla intermedia, un responsable podrá tener distintos roles, por ejemplo el responsable de seguridad podrá ser administrador.

- **Serv_Vuln**

Vuln_ID(num), Serv_Nombre(string), Vuln_Fecha(fecha), Vuln_Tiempo(hora), Vuln_Comment(memo)

Aquí se documentará las vulnerabilidades que pueda tener un servidor, cómo se han resuelto con el tiempo estimado de resolución.

- **Serv_Tarea**

SERV_Nombre(string), Tarea_ID(num), Tarea_Fecha(fecha), Tarea_Tiempo(hora), Tarea_Comment(memo)

En esta tabla se documenta las tareas que se han realizado en el servidor, la fecha y el tiempo que se emplea en el mantenimiento del servidor.

4.2.8 Identificación de las clases frontera, las clases de control y de las operaciones.

De la descripción textual de los casos de uso salen las operaciones de las clases fronteras y también, de manera más indirecta, la de las clases de control;

Las clases fronteras son las pantallas en nuestro caso:

Menú
Pantalla_login
Pantalla_error
Pantalla_cambiocontraseña
Pantalla_Menú
Pantalla_Vulnerabilidad
Pantalla_alertas
Pantalla_altaTareas
Pantalla_GestiónServidores
Pantalla_AltaVulnerabilidad
Pantalla_AltaServidores
Pantalla_AltaIncidencias
Pantalla_GestiónInformes
Pantalla_Informe
Pantalla_AltaResponsable

Las clases de control son las operaciones:

Gestor_Menú
Realizar_login
Escaneo_servidores
Gestor_alerta
Envío_email
Gestión_AltaTarea
Gestión_Servidor
Gestión_AltaVulnerabilidad
Gestión_AltaServidor
Gestión_AltaIncidencias
Gestión_Informes
Gestión_AltaResponsable

4.3 Diseño de la persistencia

Con el estudio del análisis anterior de los casos de uso, de la resolución de la herencia y de los distintos tipos de clases, especialmente de las entidades y sus atributos podemos realizar el diseño de persistencia de la Aplicación GAS.
Veamos los diagramas a continuación:

4.3.1 Diagrama estático de análisis.

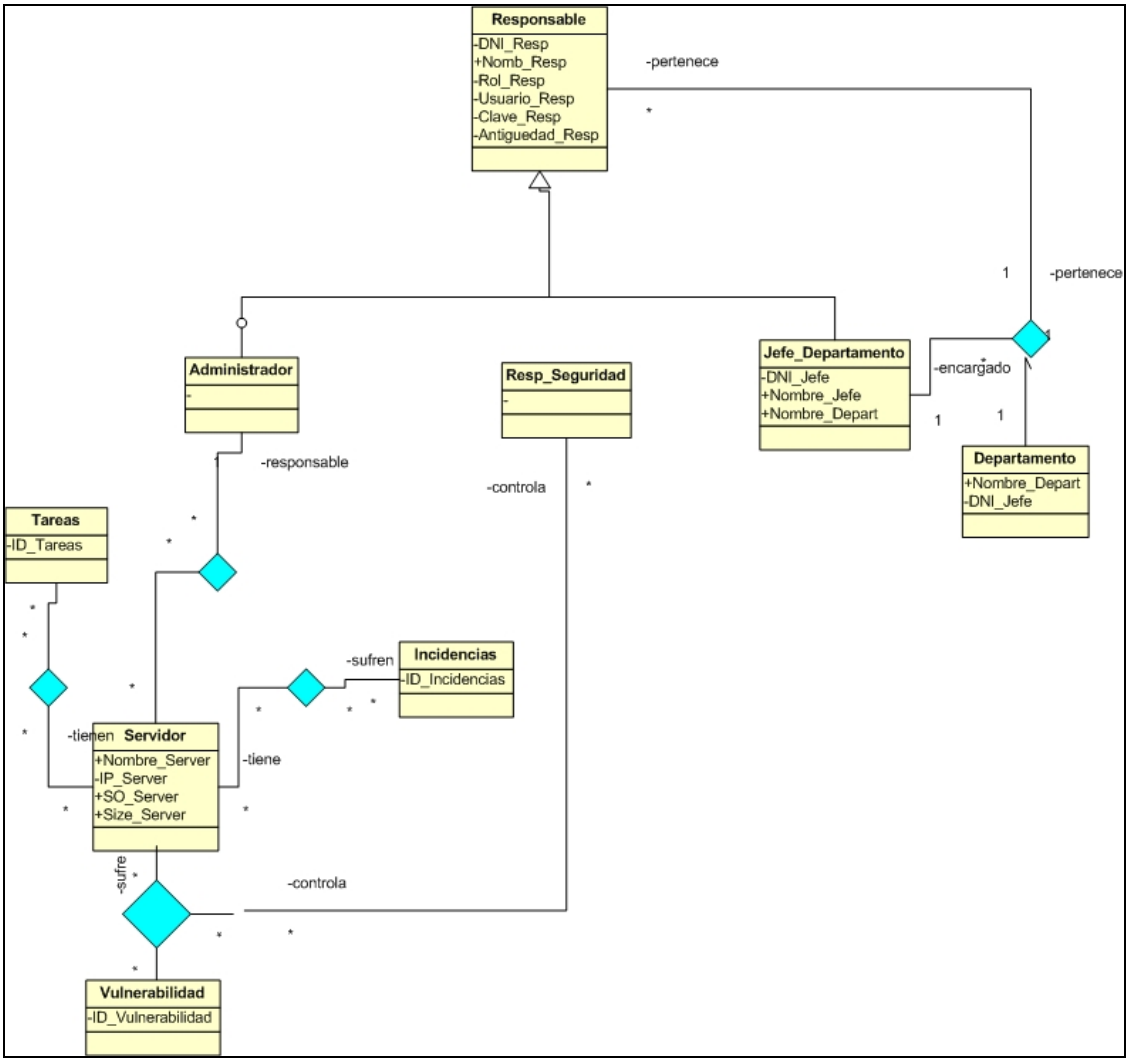


Ilustración W: Diagrama estático de análisis

4.3.2 Diagrama de Tabla Relacionales

A continuación se muestra las tablas relacionales producto de la transformación del modelo E-R al modelo relacional. El modelo lo he realizado con Ms. Access para verificar y comprobar que todas las relaciones entre entidades (en este caso ya tablas) funcionen perfectamente. En negrita tenemos las claves primarias. Vemos claramente las relaciones entre entidades.

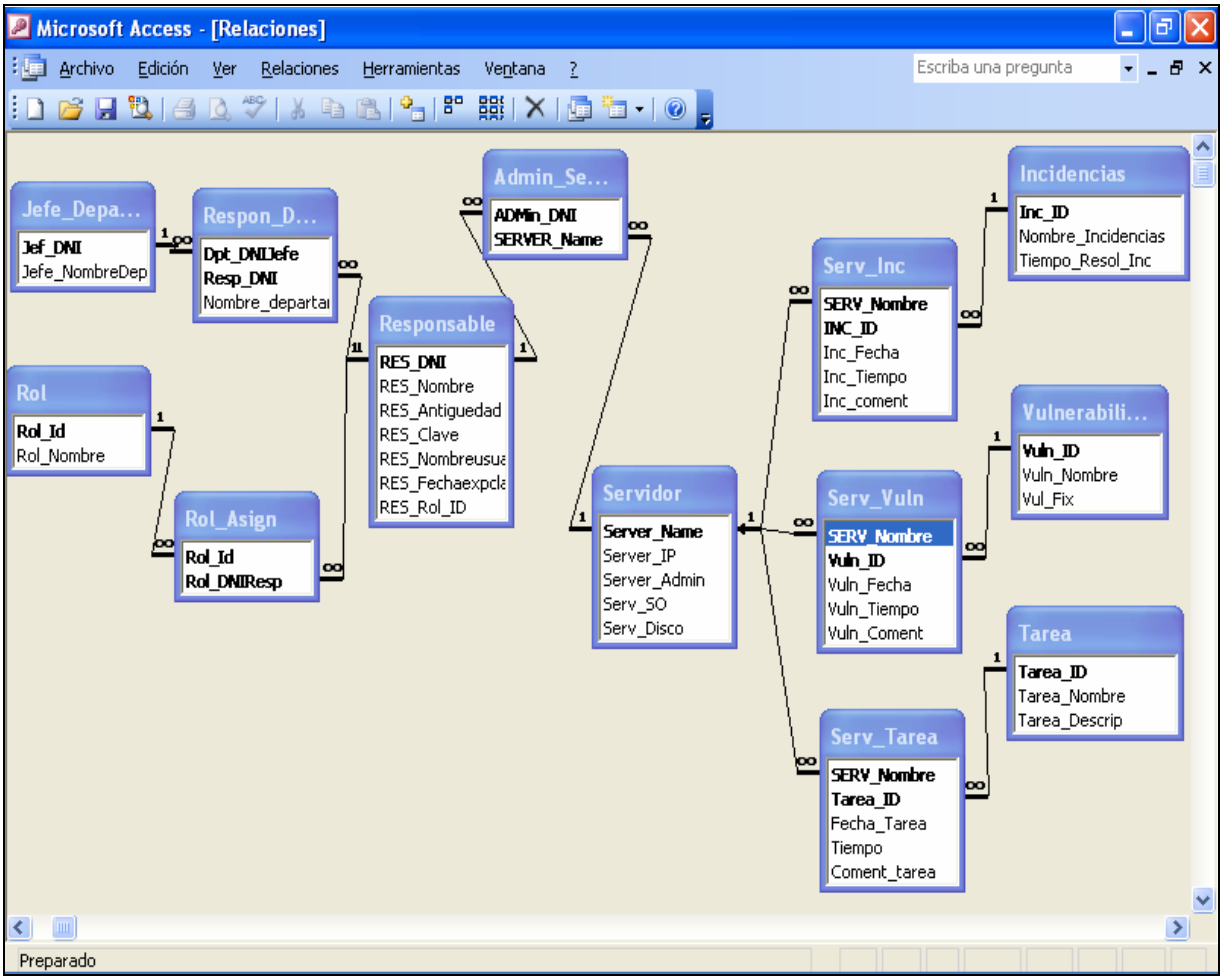


Ilustración X: Diagrama E/R

4.3.3 Conclusiones sobre el Análisis realizado

He querido asegurarme que lo escrito en el análisis refleje fielmente las necesidades del departamento al margen de pensar en la interfaz de la aplicación. Para ello, el desarrollo de las técnicas UML han sido muy útiles para llegar a una comunicación efectiva con el usuario. He intentando que los requisitos funcionales no vayan vinculados a los requisitos de usabilidad o estética de la interfaz de usuario.

Por otro lado, he querido evitar en el análisis y desarrollo de los casos de uso el llegar a una excesiva descomposición o a una excesiva abstracción todo ello sin perder de vista la frontera del sistema con sus subsistemas.

4.4 El Diseño Arquitectónico Aplicación GAS

La arquitectura seleccionada es la Plataforma. NET. El motivo es que todas las estaciones clientes tienen como sistema operativo Microsoft. La empresa (Hospital Americano Base Naval de Rota) tiene un contrato a nivel gubernamental con Microsoft. La aplicación estará desarrollada para acceder desde un navegador.

Esto son los pilares de esta nueva plataforma:

• Integración: Proporciona mecanismos para que una empresa pueda ofrecer servicios a otras empresas o clientes de una forma sencilla y rápida. En general, este tipo de servicios se suelen denominar B2B: Business to Business y B2C: Business to Client.

• Nuevos dispositivos: La forma más común de acceso a Internet hasta ahora ha sido el ordenador personal con sus limitaciones de movilidad. Pero recientemente han ido apareciendo una serie de dispositivos que permiten el acceso a servicios Internet de forma rápida y directa, como por ejemplo agendas electrónicas, teléfonos móviles, WebTV, videoconsolas, etc. Esto supone un cambio radical en la forma de acceder a este tipo de servicios.

Con estos objetivos, Microsoft .NET es una plataforma para construir, ejecutar y experimentar la tercera generación de aplicaciones distribuidas, que consiste en los siguientes elementos:

• Un modelo de programación pasado en XML.

• Un conjunto de servicios Web XML, como Microsoft.NET My Services para facilitar a los desarrolladores integrar estos servicios.

• Un conjunto de servidores que permiten ejecutar estos servicios (como .NET Enterprise Servers).

• Software en el cliente para poder utilizar estos servicios (como Windows XP, agendas electrónicas, etc.)

• Herramientas para el desarrollo como Visual Studio.NET.

A continuación vemos en la ilustración lo detallado anteriormente:

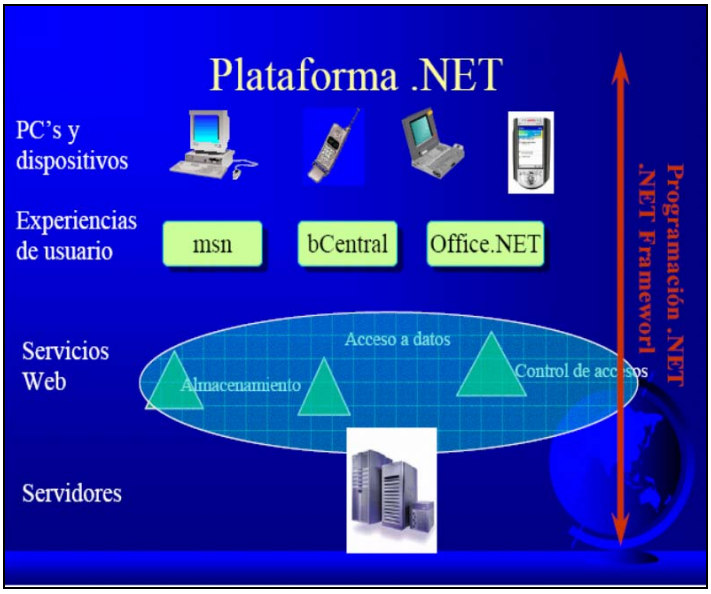


Ilustración Y: Elementos de la Plataforma .NET

Una parte importante de esta plataforma es el software de los dispositivos clientes y servidores. Para los dispositivos clientes, Microsoft planea integrar .NET en cualquier dispositivo imaginable, como PCs con Windows, agendas electrónicas con Pocket PC, teléfonos móviles, etc. Esto supone que el Hospital pueda usar estos servicios sin estar limitados al PC.

4.4.1 Servicios Web. Introducción

Simplificando, un servicio Web es un programa que se puede acceder a través de Internet utilizando protocolos estándar. Estos servicios se ejecutarán en un servidor Web, no en los PCs, permitiendo que los dispositivos que los utilicen sean más simples (simplemente se necesita un navegador Web) . Para implementar un servicio Web es necesario resolver varios problemas:

- **Representación de los datos.** Para poder compartir datos entre distintas organizaciones se necesita un estándar de representación de datos. Este estándar es XML.
- **Utilización del servicio**. Se necesita un protocolo para definir cómo acceder y utilizar el servicio. Para ello se utiliza SOAP.
- **Definición del servicio**. Dado un servicio, para poder utilizarlo se necesita saber qué operaciones ofrece y cómo utilizarlas. Para esto, se utiliza el protocolo WSDL

Todo estos mecanismos son los que utiliza la plataforma .NET para implementar sus servicios Web, que en muchos casos denomina "Servicios Web XML".

Los servicios Web son componentes débilmente acoplados, lo cual quiere decir que se puede modificar la implementación en cada lado de conexión (sin modificar el interfaz) y el conjunto seguirá funcionando.

Estándares: Dado que el objetivo de los servicios Web es la integración entre empresas es necesario que se definan una serie de estándares para que esta comunicación se realice sin problemas. Para ello el consorcio W3C está realizando un proceso de estandarización de los protocolos que utiliza .NET, detallado a continuación:

- XML Extensible Markup Language, XML es un metalenguaje de marcas que permite definir cómo es la información que se transmite. Esto permite una comunicación de datos entre distintos sistemas.
- SOAP Single Object Access Protocol. Este protocolo define como un cliente se comunica con un servicio usando HTTP y XML como mecanismo de intercambio de información.
- WSDL Web Service Description Language. Este protocolo basado en XML ha sido desarrollado conjuntamente por Microsoft e IBM. WSDL es un lenguaje en formato XML que define las operaciones que proporciona un servicio.
- UDDI Universal Description, Discovery and Integration. El UDDI es un directorio universal de Servicios Web, basado en XML que permite publicar, localizar y utilizar los servicios Web.

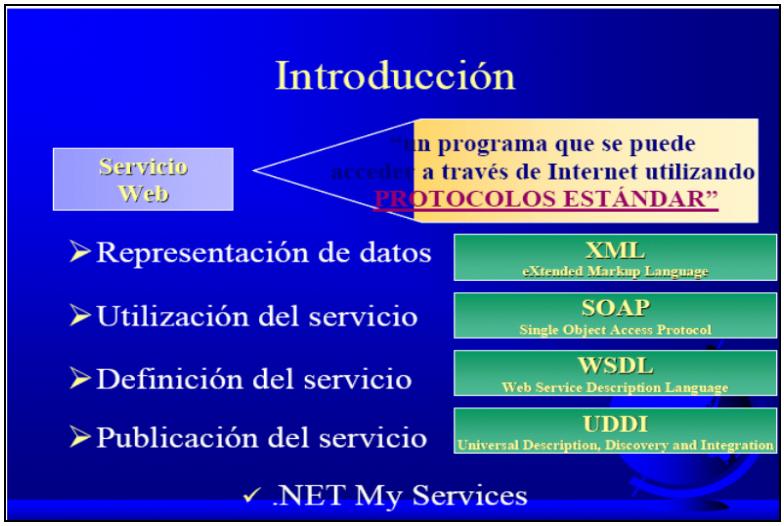


Ilustración Z: Componentes de la Arquitectura .NET

4.4.2 Arquitectura .NET

Una definición general de la arquitectura .NET podría ser la siguiente: "Una plataforma independiente del lenguaje para el desarrollo de servicios Web".

La arquitectura .NET (.NET Framework) es el modelo de programación de la plataforma .NET para construir y ejecutar los servicios .NET. El objetivo de esta arquitectura es reducir la complejidad en el desarrollo de este tipo de aplicaciones, permitiendo a los desarrolladores centrarse en escribir la lógica específica del servicio a desarrollar.
Esta arquitectura está compuesta por librerías y un ejecutivo tal como muestra la figura:

Ilustración AA: Componentes de la Arquitectura .NET

El ejecutivo del lenguaje común (CLR: Common Languaje Runtime) es un soporte que permite ejecutar los servicios .NET en cualquier máquina que lo disponga. Esta basado en la idea de Java, que también tiene un módulo de ejecución independiente del sistema operativo donde se vaya a ejecutar. Este ejecutivo es multilenguaje, esto es, no está limitado a un único lenguaje. Esto permite al desarrollador utilizar una amplia variedad de lenguajes como C++, Visual Basic y C#. Las librerías básicas proporcionan una serie de funcionalidades que son necesarias a la hora de desarrollar los servicios Web. Las clases básicas gestionan las operaciones más básicas como las comunicaciones, entrada/salida, seguridad, etc. Las clases XML y de datos gestionan el acceso a base de datos y la gestión de datos en XML. El objetivo de las librerías Servicios Web XML es la de dar soporte para el desarrollo de aplicaciones distribuidas que ofrezcan servicios XML a otras entidades. Las Web forms permiten desarrollar la parte gráfica de una aplicación para la Web, mientras las Windows Forms están orientadas a implementar la parte gráfica de las aplicaciones clásicas para Windows.

La aplicación GAS al ser una aplicación en Web estará desarrollada en Web Forms.

En la figura 3 se muestra el modelo de ejecución de los programas .NET muy parecida a la de Java.

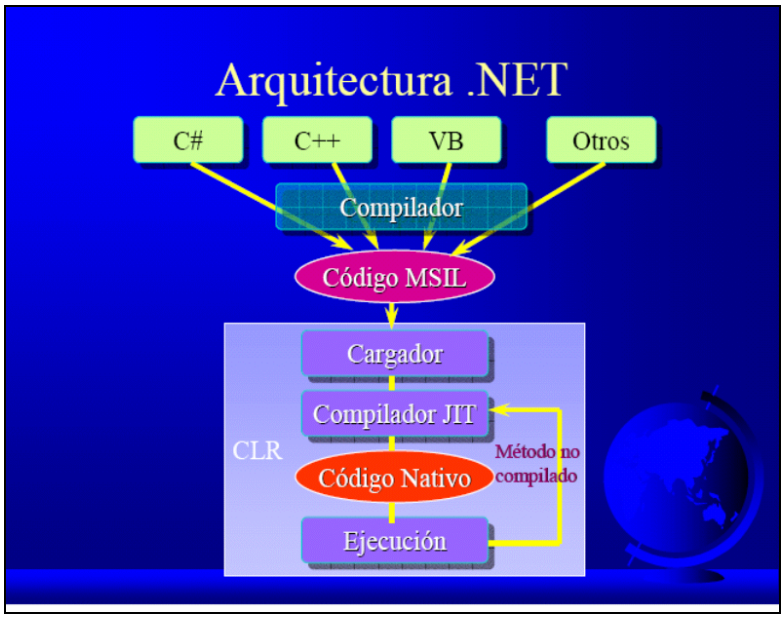


Ilustración BB: Modelo de Arquitectura .NET

Los compiladores producen código MSIL (MicroSoft Intermediate Language), que es un lenguaje intermedio que se puede ejecutar en la máquina virtual. Este código no es interpretado por el ejecutivo, sino que es compilado de nuevo en tiempo de ejecución (JIT: Just in Time) al código nativo de la máquina. Este código compilado no se ejecuta independientemente sino dentro de este ejecutivo. Esto se denomina código manejado, lo cual permite que el ejecutivo controle ciertos aspectos de la aplicación que ejecuta como son seguridad, gestión de memoria, compartición de datos, etc.

4.4.3 Diseño Web Aplicación GAS

El sistema que se quiere diseñar se perfila como un sistema web en el que se permite que el usuario invoque la lógica del negocio y por consiguiente, cambie el estado de negocio del servidor. Esta definición implica que por lo menos existen tres elementos arquitectónicos en el sistema web:

- Navegado cliente.
- Servidor Web y servidor de Aplicaciones y componentes
- Servidor de Base de Datos.

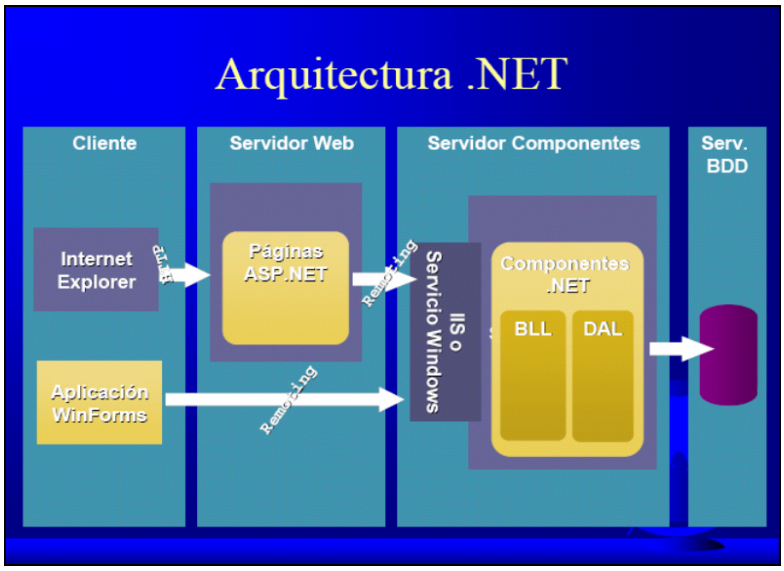


Ilustración CC: Arquitectura .NET para aplicación GAS

En nuestro caso el servidor Web y de aplicaciones va a ser un mismo servidor.

Existen tres patrones comunes que expresan un esquema de organización estructural de un sistema Web:

- Cliente Web ligero: Se requiere que el cliente tenga un navegador Web estándar capaz de mostrar formularios. Toda la lógica del negocio se ejecuta en el servidor.
- Cliente Web pesado: Una cantidad significativa de la lógica de negocio se ejecuta en la máquina del cliente.
- Entrega vía Web: Además de usar http como medio de comunicación entre el cliente y el servidor, se emplean otros protocolos como IIOP y DCOM, para dar soporte al sistema de objetos distribuidos. El navegador Web actúa principalmente como entrega y dispositivo contenedor del sistema de objetos distribuidos.

5. Diseño de la Interfaz.

A partir de los escenarios anteriormente descritos, podemos generar una estructura de la aplicación y los flujos de interacción que se producen. He querido hacer un flujo de interacción general y lo he realizado gráficamente mediante Microsoft Visio 2003.

Estos flujos de interacción proporcionan un vocabulario visual para describir, en un nivel alto, la estructura y/o flujo de la experiencia de usuario de un sitio Web.
Adjunto el link del documento que explica los distintos iconos:
http://www.jjg.net/ia/visvocab/files/garrett_ia_cheatsheet.pdf

5.1 Escenario y Flujo de Interacción. Vocabulario Visual.

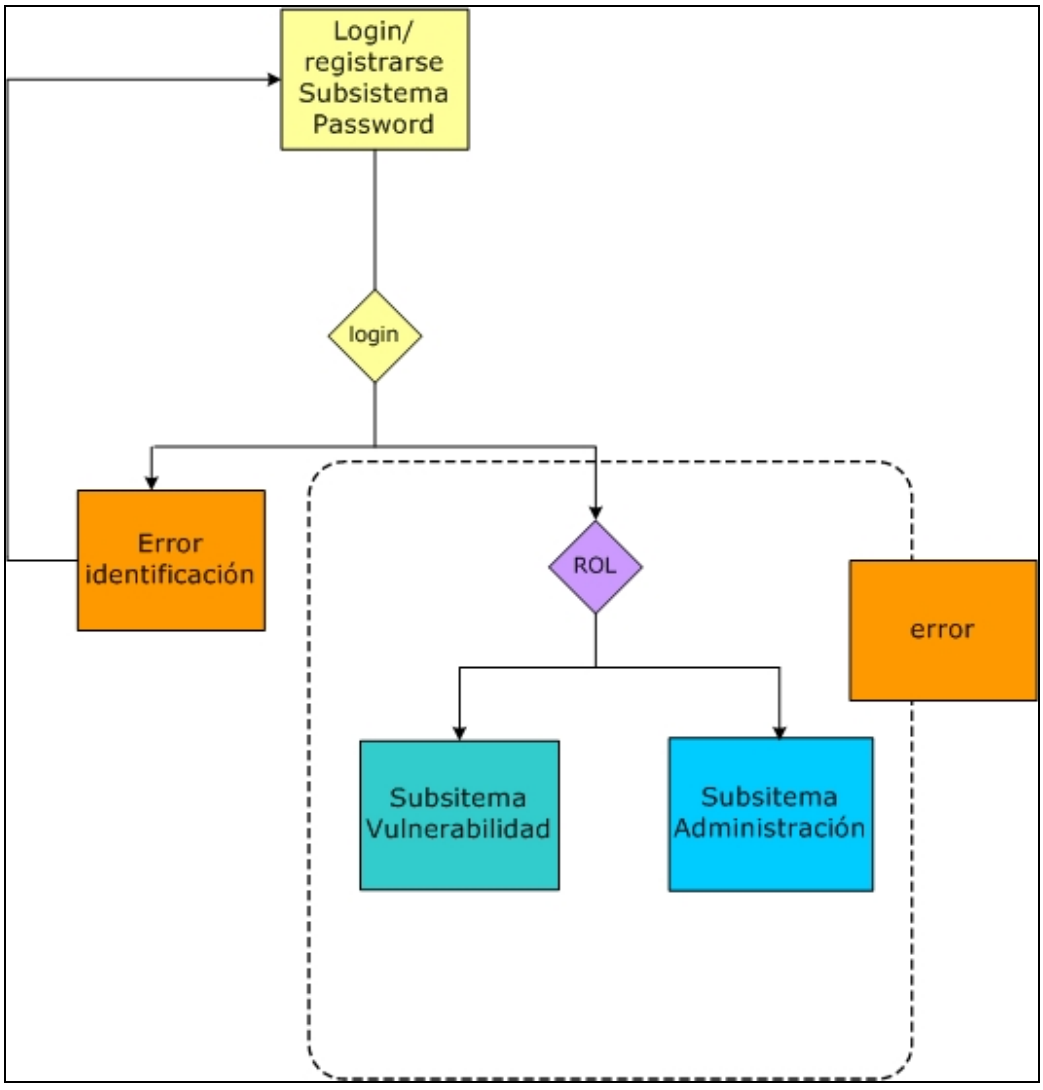


Ilustración DD: Diagrama Vocabulario Visual

5.2 Prototipo de Alto Nivel

Aunque esta parte no es obligatoria, he querido incluir algunas imágenes de las pantallas diseñadas, para mostrar los formularios en un entorno Web. El desarrollo de la aplicación se ha realizado con **Visual Basic .NET** con Web Forms explicado anteriormente y he utilizado localhost como servidor Web para el prototipo

Ilustración EE: Formulario de acceso

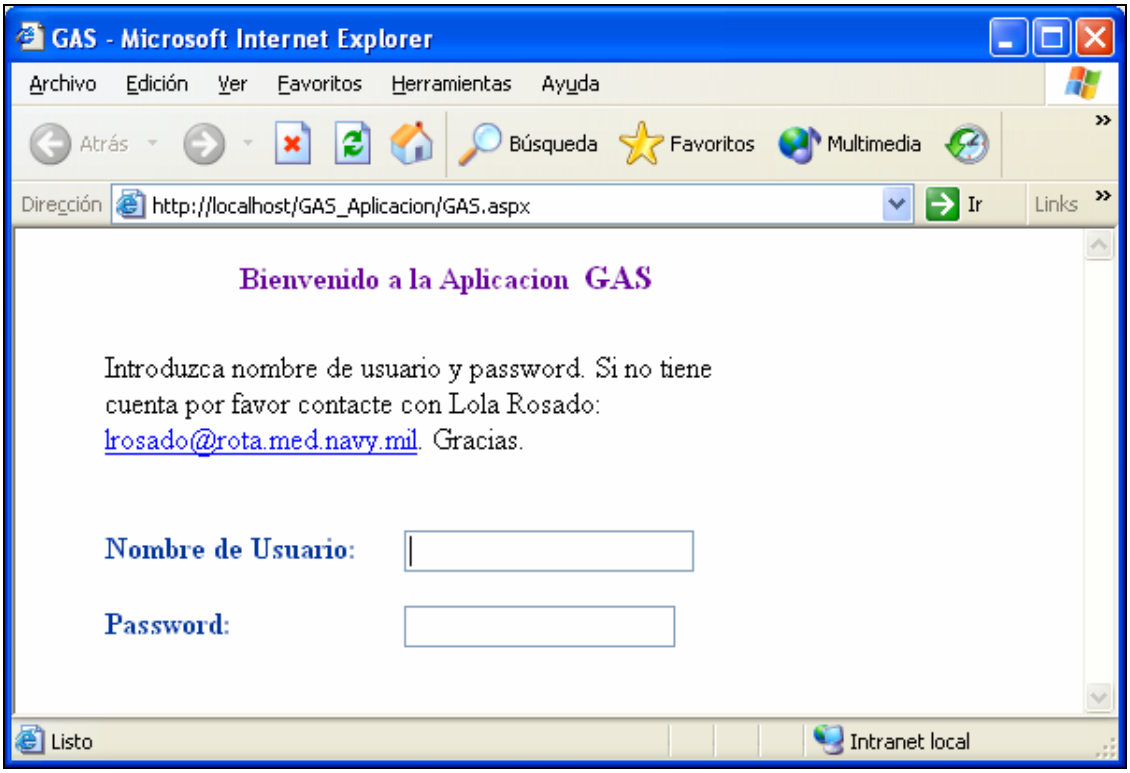


Ilustración FF: Formulario error acceso

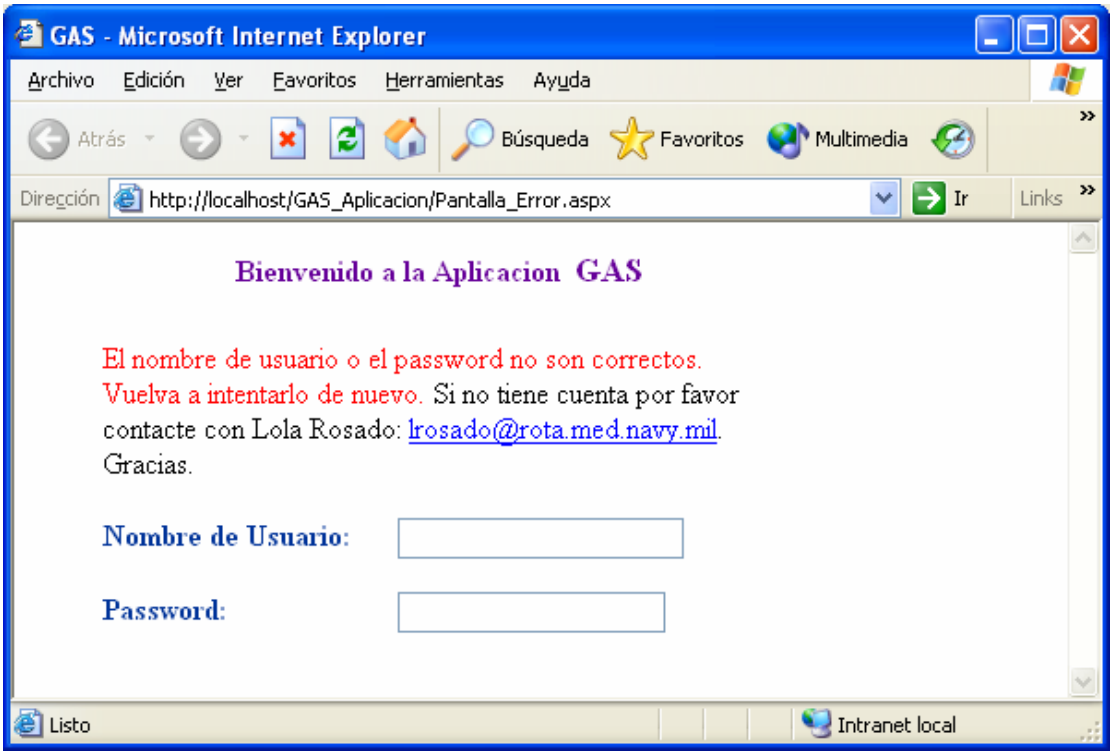


Ilustración GG: Formulario cambio contraseña

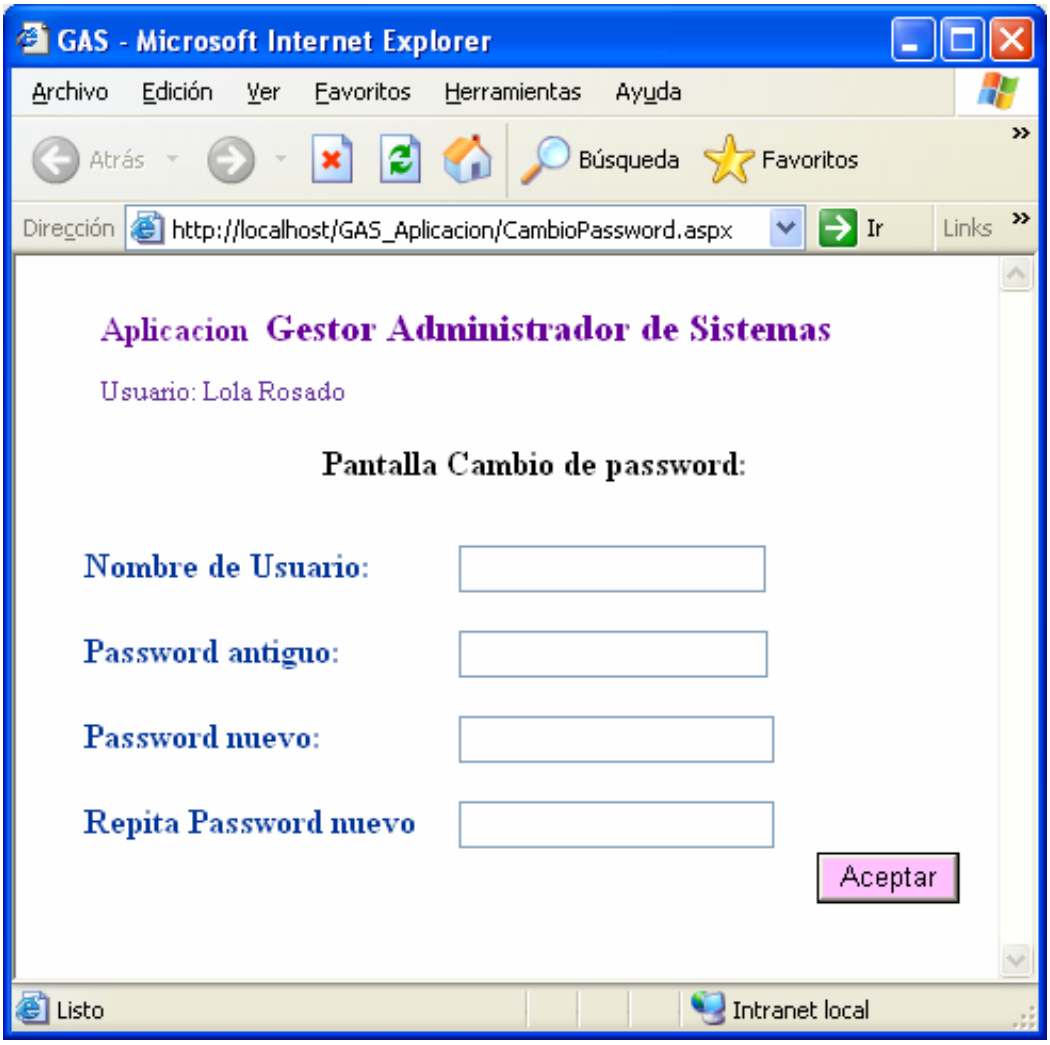


Ilustración HH: Formulario Alta tareas

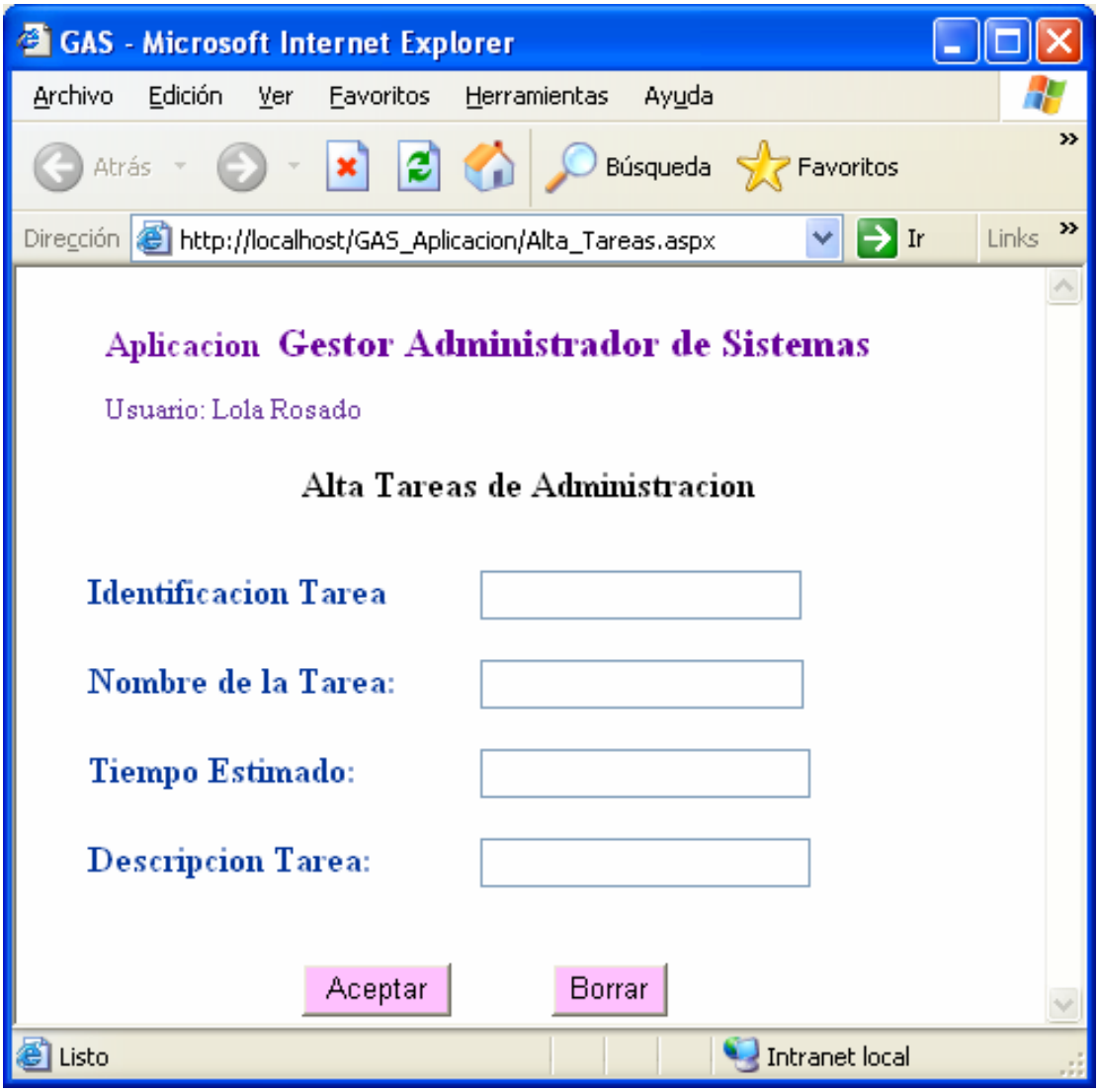


Ilustración II: Formulario Alta de Incidencias

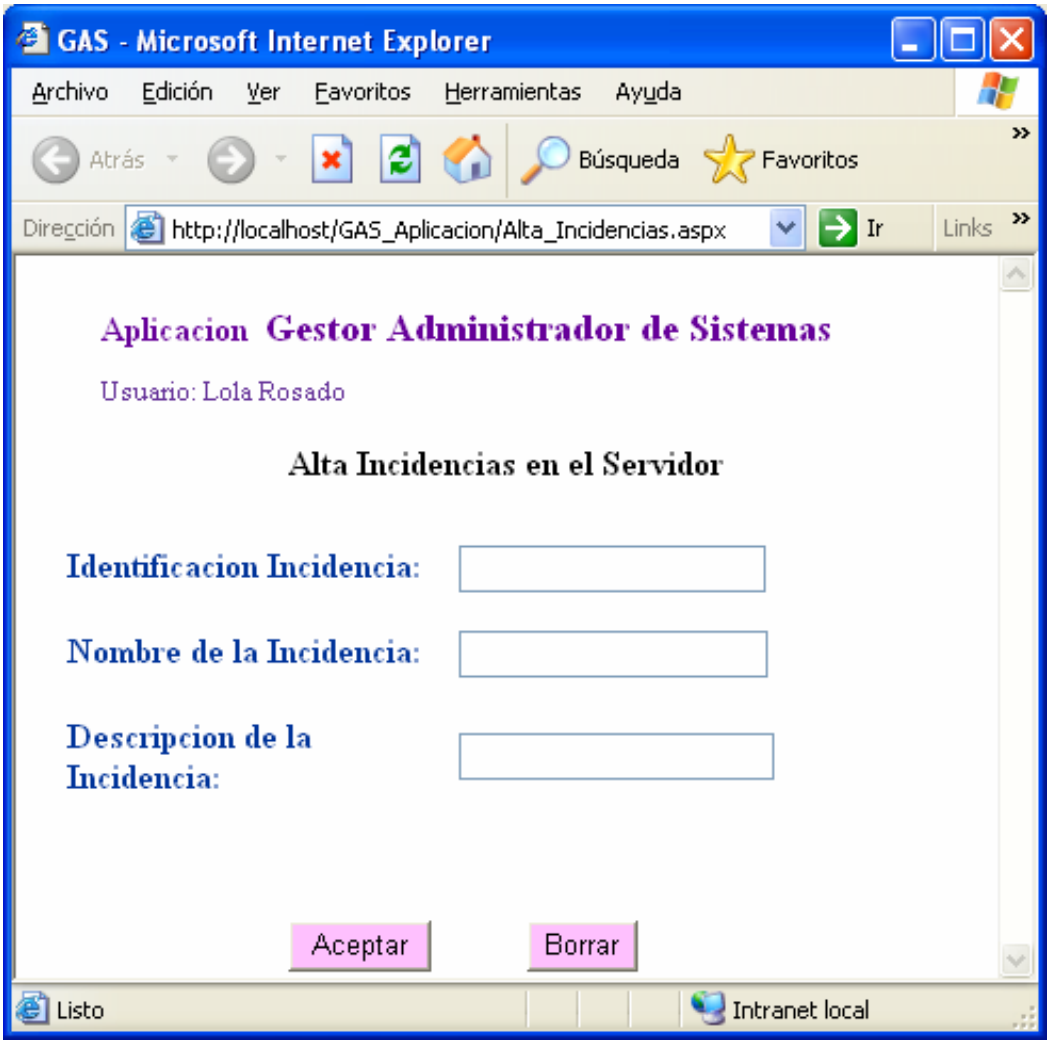


Ilustración JJ: Formulario Alta de Vulnerabilidades

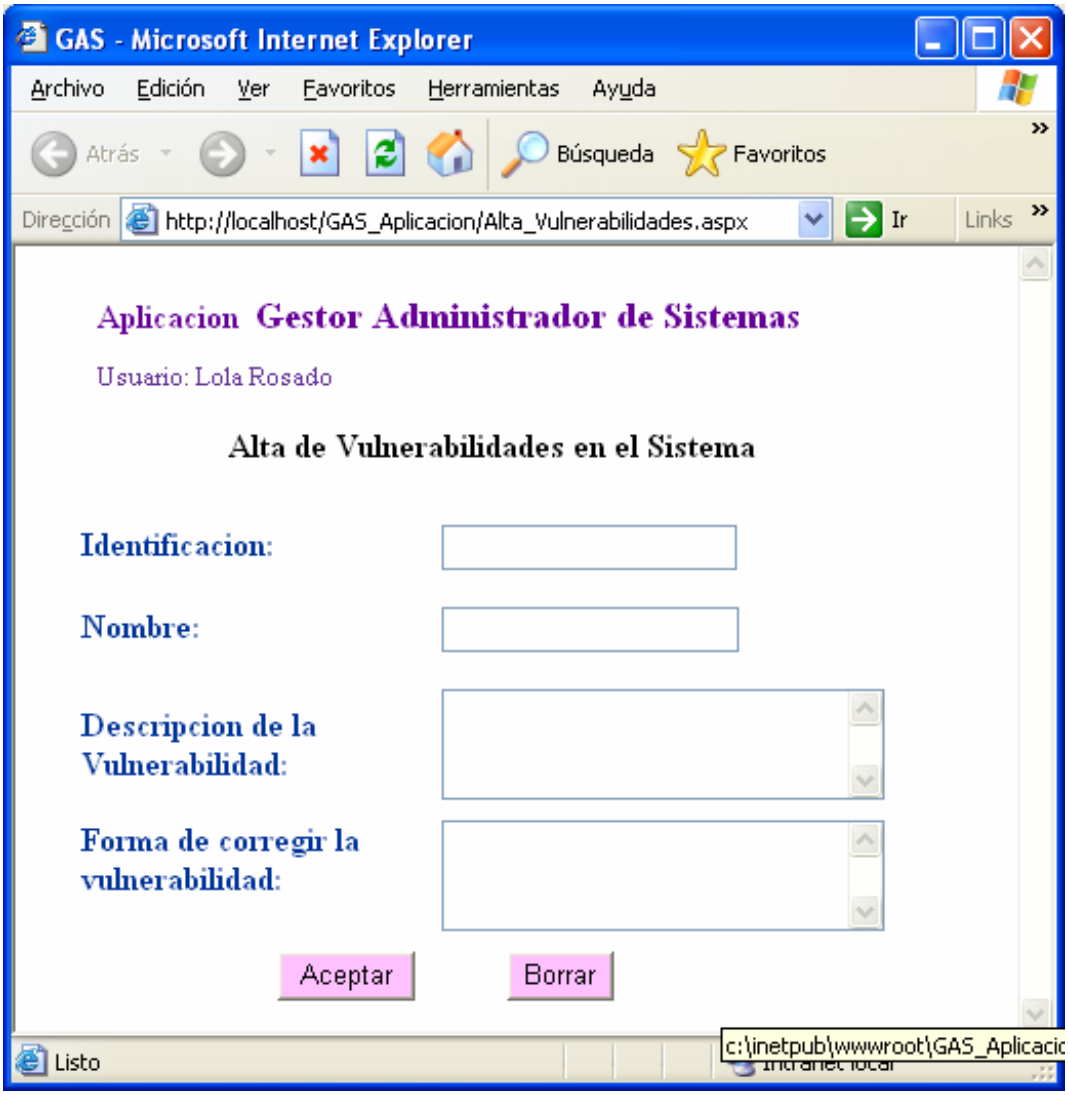


Ilustración KK: Formulario Selección servidor

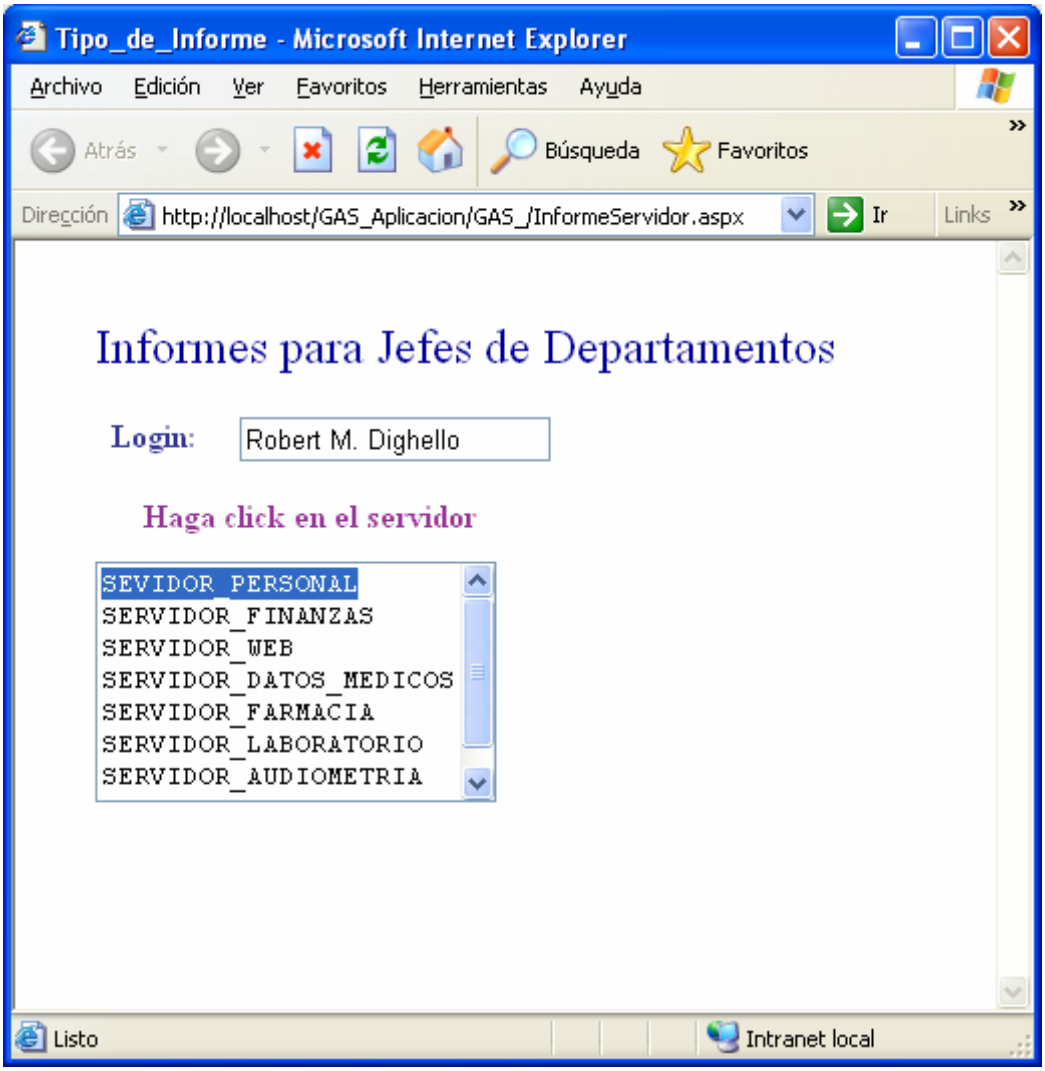
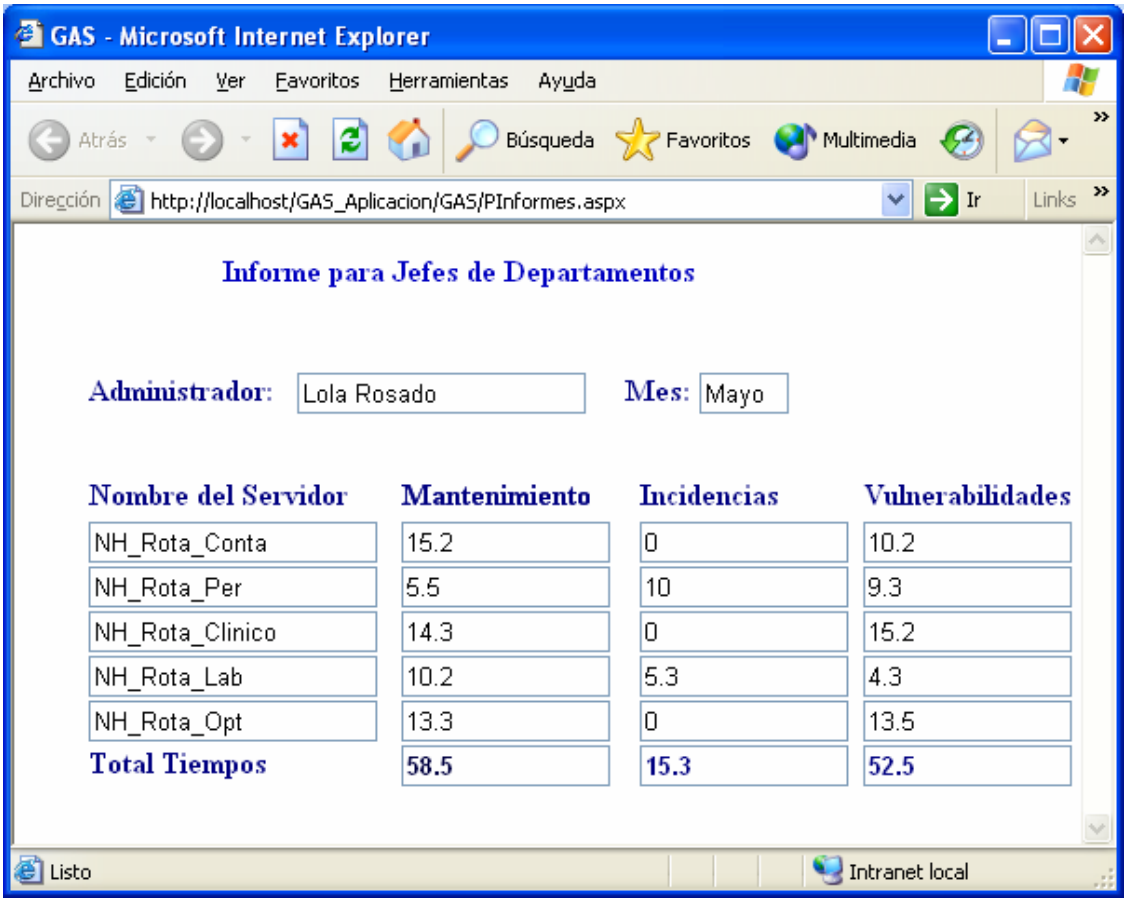


Ilustración LL: Formulario Jefes Departamento

Aquí se relaciona el Jefe de Departamento con el Administrador y todos sus sistemas con los tiempos empleados en mantenimiento, incidencias y control de vulnerabilidades.



5.3 *Conclusiones del Diseño. Fase Final*

Como punto final quedaría hacer un estudio de la usabilidad de la aplicación. La evaluación heurística es el método más popular para evaluar la usabilidad ya que se basa en evaluar un conjunto de criterios.

Se ha realizado el estudio del diseño de la aplicación GAS junto con el prototipo de las pantallas siguiendo los criterios recomendados por Jakob Nielsen:
http://www.useit.com/papers/heuristic/heuristic_list.html

- Visibilidad del estado del sistema:

El sistema debería mantener siempre informados a los usuarios de lo que ocurre, a través de una realimentación apropiada en un plazo razonable.
En nuestro caso, desde la primera pantalla podemos ver todas las nuevas tareas, desde el tablón de noticias, foro, sugerencias y listados de impuestos.

- Encaje entre sistema y mundo real:

El sistema debería hablar el idioma de los usuarios, con términos, expresiones y conceptos familiares para el usuario, en lugar de con términos orientados al sistema. Sigue las convenciones del mundo real, presentando la información en un orden lógico y natural. El lenguaje es llano y claro, no técnico.

- Libertad y control del usuario:

Los usuarios a menudo eligen funciones del sistema por error y necesitarán una "salida de emergencia" claramente indicada para salir del estado no deseado sin tener que pasar por un diálogo prolongado. Proporcionar soporte de deshacer y rehacer.

- Consistencia y estándares:

Los usuarios no deben tener que preguntarse si diferentes palabras, situaciones o acciones significan lo mismo. Seguir las convenciones de la plataforma.

- Prevención de errores:

Aún mejor que unos buenos mensajes de error, es un diseño cuidadoso que impide que los problemas lleguen a ocurrir. Cómo se ha visto en el flujo interactivo se controla los errores.

- Reconocimiento en lugar de memoria:

Hacer visibles objetos, acciones y opciones. El usuario no debería tener que recordar información de una parte de un diálogo a otra. Las instrucciones para el uso del sistema deberían ser visibles o fácilmente accesibles cada vez que sea adecuado. Se intenta que el sistema sea consistente.

- Flexibilidad y eficiencia de uso:

Los atajos -- invisibles al usuario novato -- a menudo pueden acelerar la interacción para el usuario experto de modo que el sistema puede adecuarse tanto a usuarios experimentados como noveles. Permite a los usuarios personalizar o adaptar acciones frecuentes.

- Estética y diseño minimalista:

Los diálogo no deberían contener información irrelevante o rara vez necesaria. Cada unidad extra de información de un diálogo compite con las unidades relevantes de información y reduce su visibilidad relativa.

- Ayuda a los usuarios a reconocer, diagnosticar y recuperarse de los errores:

Los mensajes de error deberían expresarse en lenguaje llano (sin códigos), describir con precisión el problema y sugerir de forma constructiva una solución.

- Ayuda y documentación:

Aunque es mejor si el sistema puede usarse sin documentación, puede ser necesario proporcionar ayuda y documentación. Estas informaciones deberían ser fáciles de buscar, deberían centrarse en las tareas del usuario, detallar los pasos concretos que es necesario realizar y no ser demasiado voluminosas.

6. Bibliografía

Además de la bibliografía incluida en la PEC2 para la realización de esta PEC3 se ha estudiado los documentos siguientes:

1. Introducción a Microsoft .Net
http://www.disca.upv.es/enheror/pdf/ActaNET.PDF

2. Modelo Heurístico
http://www.useit.com/papers/heuristic/heuristic_list.html

3. Diseño Visual. Modelo Garret.
http://www.jjg.net/ia/visvocab/files/garrett_ia_cheatsheet.pdf

4. Centro de Arquitectura .Net
http://www.microsoft.com/spanish/msdn/arquitectura/default.asp

5. Mi segunda práctica de IHO para la última parte de este trabajo.

6. El trabajo Presa cedido por el consultor.

7. Material de UML de Ingeniería del Software de la UOC

8. Material de Administración de Sistemas Operativos de la UOC

9. Material de Técnicas de Desarrollo de Software de la UOC

10. Material de Gestión de Organizaciones y Proyectos Informáticos de la UOC

11. Herramienta hfnetchk
http://www.microsoft.com/latam/technet/seguridad/herramientas/hfnetchk.asp

12. Trabajo Presa.doc cedido por el consultor de la asignatura

7. Glosario

Además del Glosario realizado en la PEC2, incluyo aquí los nuevos términos que aparecen en esta PEC3.

Atributo de una entidad: Propiedad que interesa de una entidad

Base De datos: Es un conjunto estructurado de datos que representa, entre otros, entidades y sus interrelaciones, con integración y compartimiento de datos.

Diseño Conceptual: Etapa del diseño de una base de datos que obtiene una estructura de la información futura base de datos independiente de las tecnologías que se quiera utilizar.

Entidad: Objeto del mundo real que podemos distinguir del resto de los objetos y del cual nos interesan algunas propiedades.

Interrelación: Asociación entre entidades.

Entidad débil: Entidad cuyos atributos no la identifican completamente, sino que sólo la identifican de forma parcial.

Usabilidad: Medida en la cual un producto puede ser usado por usuarios específicos para conseguir objetivos específicos con efectividad, eficiencia y satisfacción en un contexto de uso especificado.

XML: (eXtensive Markup Language) Lenguaje desarrollado por el W3 Consortium para permitir la descripción de información contenida en el WWW a través de estándares y formatos comunes, de manera que tanto los usuarios de Internet como programas específicos (agentes) puedan buscar, comparar y compartir información en la red. El formato de XML es muy parecido al del HTML aunque no es una extensión ni un componente de éste.

HTML: (Hypertext Markup Language). Lenguaje de "etiquetas" en el que se asigna formato a las páginas de Web y se distribuye la información.

Vulnerabilidad*:* Es un fallo en el sistema que puede ser aprovechado por una amenaza.

Parches de Seguridad*:* Programa que se instala en el servidor para resolver la amenaza o fallo en el sistema para solventar la vulnerabilidad.

Email*:* Correo electrónico.

GUI*:* Graphical Users Interfaces. Es una interface gráfica para el usuario.

Dashboard*:* Pantalla con la información.

Schedule*:* Calendario donde se va a programar las tareas de mantenimiento a realizar en el servidor con fecha de comienzo, frecuencia y duración.

Troubleshooting: Problemas técnicos en el servidor. Fallos técnicos a resolver por el administrador del sistema.

GHOST: Software utilizado por los administradores para realizar copias de seguridad completas del sistema.

Sistema: Servidor.

8. Anexos

8.1 Definición General del Proyecto

El objetivo del presente Trabajo Final de Carrera consiste en la realización del desarrollo de un Gestor de Administración de Sistemas (GAS). El desarrollo de este gestor tendrá uso real y vendrá a cubrir una necesidad del departamento de Sistemas Informático del Hospital Americano de la Base Naval de Rota donde trabajo como administradora de sistemas.

En el Hospital gestionamos un total de más de 90 servidores distintos. Al ser un recinto medico y militar, la seguridad en la red y en los sistemas es una alta prioridad.
Cada administrador tenemos a nuestro cargo un número de sistemas, donde nuestra responsabilidad es actualizar los parches de Microsoft, IAVAS, actualizaciones de software, bases de datos, hardware y demás tarea propias de la administración de un servidor.

En cualquier momento, nuestro supervisor y altas jerarquías pueden pedir la situación de nuestros servidores. Este gestor será la aplicación GUI mediante WEB que permitirá tener almacenado en un repositorio común toda la información de nuestros sistemas informáticos. Podrá ayudar a los administradores a poner a punto nuestros servidores y por otro lado, el jefe de cada departamento podrá saber en cada momento el estatus de cada sistema. Este Gestor también podrá informar a los supervisores y justificar los administradores del tiempo empleado por tarea en cada servidor. Habrá un repositorio de tareas con el tiempo que se tarda en cada caso.

8.2 Objetivos del proyecto

8.2.1 Generales

- Actualmente y dependiendo de cada administrador, los datos de los sistemas se encuentran a nivel local en el ordenador del administrador responsable y en distintos formatos. Algunos guardan la información en Excel, en Word o en Access o no hay información almacenada. Este Gestor va a proporcionar un modelo único donde la información estará almacenada en un repositorio común basado en tecnología Web.

8.2.2 Específicos

- Realizar un trabajo de fin de proyecto que me permita obtener el apto y dar por finalizados mis estudios en la carrera.
- Aprovechar el desarrollo del proyecto para conocer profundamente el ciclo de vida del Proceso Unificado de Desarrollo y manejar el lenguaje UML perfectamente.
- Resolver una carencia que actualmente tiene mi departamento de informática y yo personalmente como administradora de sistemas, ya que no tenemos un repositorio donde almacenar y posteriormente gestionar la información de nuestros servidores.

8.3 Entregables

Los entregables en este Trabajo Final de Carrera han sido los siguientes:

- El Plan de Trabajo, que recoge la planificación y estimación de las tareas necesarias para llevar a cabo los objetivos previstos. PEC 1
- Fase de Especificación de requisitos y análisis del Sistema GAS. PEC2.
- Fase de Diseño del Sistema GAS. PEC 3
- La Memoria, que es actual documento que sintetiza el trabajo realizado y muestra que se han alcanzado los objetivos propuestos. Incorpora toda la información relevante para comprender el problema planteado, la metodología utilizada para su resolución y detalla la solución elaborada.

- La Presentación Virtual, que resumirá de forma clara y concisa el trabajo realizado y los resultados obtenidos

8.4 Recursos e Infraestructura

En este apartado se detallan los recursos necesarios para llevar a cabo el proyecto.

8.4.1 Hardware

Se utilizarán en este proyecto dos ordenadores PC compatibles:

- Equipo de sobremesa: DELL Intel (R) Pentium; CPU 2.40 GHz. MS Windows 2000 SP4
- Ordenador portátil: DELL Intel (R) 82845 Processor to AGP Controller. 1 GB de memoria RAM con SO Windows XP Professional Edition.

8.4.2 Software

- Diagramas UML: Rational Rose, MS Visio 2003 y Poseidón
- Elaboración de documentos: MS Office (MS Word, Excel, PowerPoint)
- Elaboración del Modelo Entidad-Relación: Ms. Access.
- Planificación y seguimiento del proyecto: Ms. Excel

8.5 Estructura del proyecto. Descomposición en actividades

8.5.1 Estructura del proyecto

Teniendo en cuenta que el proyecto se puede definir con bastante precisión y dados los plazos previstos, utilizaremos una metodología de desarrollo en cascada.

8.5.2 Descomposición en actividades

Las actividades que compondrán el proyecto serán las siguientes:

Descomposición Estructural de Actividades (WBS)	
NIVEL 1	NIVEL 2
Definición del TFC. PEC1	
	Identificación del TFC
	Elaboración del Plan del Trabajo. Entrega
PEC2. Análisis de la Aplicación GAS	
	Especificación de requisitos.
	Funcionalidades del sistema GAS
	Requisitos Hardware y Software
	Identificación de Actores
	Identificación de Casos de Usos principales
	Documentación textual de Casos de Uso
	Diagramas de Casos de Uso
PEC3. Diseño de la Aplicación GAS	
	Identificación de Clases entidades
	Diagrama de Entidades

	Diseño Casos de Uso
	Identificación de clases gestores, frontera, entidades
	Diagramas de clases gestores, frontera, entidades
	Diagramas de colaboración
	Diseño interfaz Usuario
Memoria. Entrega Final. Presentación	
	Revisión de las entregas parciales.
	Elaboración del documento Memoria
	Elaboración de la Presentación Virtual

Tabla 14: Casos de Uso "Alta Tareas de Administración" ID [006]

8.5.3 Descripción de las actividades

Las principales actividades que componen el proyecto son las siguientes:

- **Planificación TFC**

Esta tarea se realiza en la entrega del Plan de Trabajo. Siguiendo las metodologías de la planificación aprendidas en las distintas asignaturas de la carrera, se determinan las tareas necesarias para alcanzar los objetivos del proyecto y se estima el esfuerzo y los recursos necesarios para llevarlas a cabo.

- **Fase de Especificación de Requisitos y Análisis de la aplicación GAS**

Aquí es donde se inicia las entrevistas con los futuros usuarios de la aplicación. Se identifica:

- Requisitos de Hardware y Software para la aplicación
- Identificación de Actores
- Identificación de Casos de usos
- Documentación textual de los Casos de Uso
- Diagramas de Casos de Uso

- **Fase de Diseño de la aplicación GAS**

En esta sección es donde se realiza el Análisis y diseño de la arquitectura de la aplicación:

- Identificación de Clases
- Diagrama de Entidades (Diseño Bases de Datos)
- Diseño Casos de Uso
- Identificación y Diagramas de clases gestores, fronteras y entidades
- Diagramas de Colaboración

- Diseño interfaz usuario

- **Elaboración Memoria y Presentación Virtual del Proyecto GAS**

Esta tarea recoge la realización y entrega de los documentos finales de este Trabajo Final de Carrera. Se prevé una revisión de los documentos entregados con anterioridad a fin de asegurar la robustez del trabajo.

8.6 Planificación temporal del proyecto

La fecha inicial del proyecto es el 2-03-2006, mientras que la fecha límite es el 16-06-2006. Disponemos de 107 días naturales. Esta es la única asignatura que me queda para la terminación de la carrera y mi dedicación ha sido a tiempo completo

8.6.1 Identificación de Hitos

FECHA	HITO DE CONTROL
13/03/2006	Entrega Plan de Trabajo. PEC1
21/04/2006	Entrega del análisis de la aplicación. PEC2
29/05/2006	Entrega del diseño de la aplicación. PEC3
16/06/2006	Memoria y Presentación.

Tabla 15: Identificación de Hitos

8.6.2 Estimación de tiempos

Nombre de tarea	Duración
- Trabajo Fin de Carrera. GAS.	**107 días**
- Definición del TFC. PEC1	**12 días**
Identificación del TFC	2 días
Elaboración del Plan del Trabajo. Entrega	10 días
- PEC2. FASE DE ESPECIFICACION DE REQUISITOS Y ANALISIS	**39 días**
Especificaciones de Requisitos del sistema GAS	9 días
Funcionalidades del Sistema (General)	2 días
Requisitos Hardware y Software	1 días
Identificación de Actores	2 días
Identificación y Relación de Casos de Uso principales	7 días
Documentación textual de Casos de Uso	9 días
Diagramas de Casos de Uso	9dias
- PEC3. FASE DE DISENO.	**38 días**
Identificación de Clases	5 días
Diagrama de entidades	3 días
Diseño Casos de uso.	7 días
Identificación clases gestores, fronteras y entidades	5 días
Diagramas de clases gestores, entidades y fronteras y entidades	4 días
Diagramas de colaboración	4 días
Diseño interfaz usuario	10 días

- Memoria y presentación. Entrega final	**18 días**
Revisión de las entregas parciales del proyecto	2 días
Elaboración del documento de Memoria del Proyecto	7 días
Elaboración de la Presentación Virtual del proyecto	9 días

Tabla 16: Estimación de Tiempos

8.7 Diagrama de Gantt

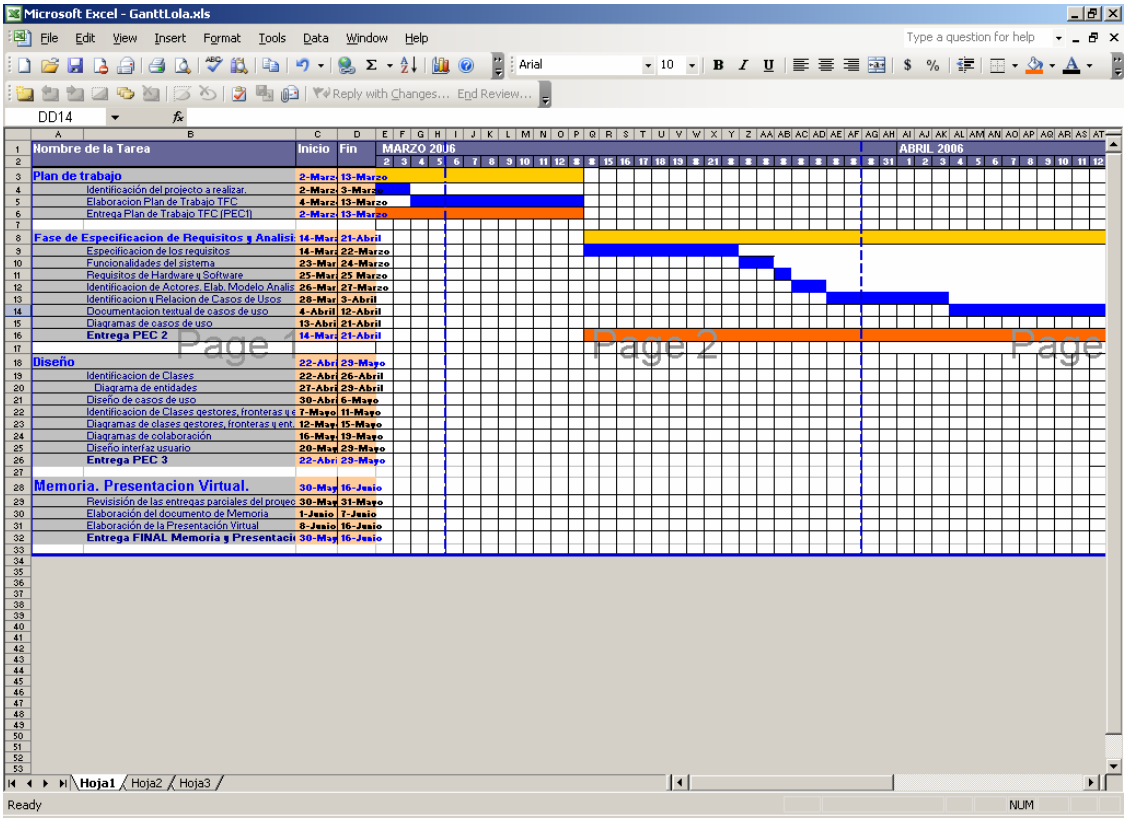


Tabla 17: Diagrama de Gantt

9. Conclusiones

Una vez finalizado todo el proceso de desarrollo de acuerdo con la planificación realizada, podemos verificar que los objetivos que identificamos al inicio y que se detallan en el apartado 2.1.2 se han cumplido. Hemos realizado el análisis y diseño de una aplicación siguiendo estándares tantos técnicos como de especificación.

También se han cumplido los objetivos de más largo alcance que hemos detallado al comienzo de este documento y que justifican la realización de este trabajo. El proceso de desarrollo que detalla esta memoria, y la calidad de los productos obtenidos,

reconocida a lo largo de la evaluación continua, evidencian que he aplicado de forma rigurosa y metódica los conocimientos, metodologías, técnicas y habilidades que he adquirido a lo largo de mis estudios de informática, en particular en el área de ingeniería del software en la que este trabajo se centra.

Este trabajo ha servido también para demostrar mi capacidad de afrontar un proyecto de cierta complejidad, haciendo un uso integral de mis conocimientos y habilidades para llevar a cabo con éxito y de forma independiente actividades relacionadas con la planificación, análisis, y diseño.

Por último, sólo me queda señalar que la realización de este trabajo ha constituido una experiencia enriquecedora que me permitirá desempeñar mejor mi rol como administradora de sistemas ya que este proyecto será de gran utilidad en mi trabajo diario.