

MEMORIA DEL PROYECTO

SGAPA

Sistema de Gestión Administrativa

de Peticiones y Actividades

SGAPA-MDP-01-v1.0 – 18/06/2007

Consultor

Oscar Escudero Sánchez

Estudiante

Lorenzo García Celada

ÍNDICE

1	Introducción	5
2	Documentos asociados	5
3	Descripción, objetivos y planificación del proyecto	5
3.1	Antecedentes del Proyecto	5
3.1.1	Descripción del dominio del problema	5
3.1.2	Situación inicial de los sistemas	6
3.1.3	Motivación del cambio	6
3.2	Objetivos del proyecto	7
3.2.1	Metas técnicas	7
3.2.2	Metas académicas	8
3.3	Enfoque y metodología	8
3.3.1	Análisis inicial de riesgos	8
3.3.2	Metodología	9
3.4	Planificación del proyecto	9
3.5	Productos generados	10
4	Especificación y análisis de requisitos	10
4.1	Panorámica general	10
4.2	Catálogo de requerimientos del cliente	11
4.3	Actores del sistema	12
4.4	Requisitos del sistema	12
4.4.1	De almacenamiento de la información	12
4.4.2	Funcionales	13
4.4.3	No funcionales	17
4.5	Modelos del Análisis	19
4.5.1	Modelo de objetos del dominio	19
4.5.2	Modelo de casos de uso	19
5	Diseño del Sistema	19
5.1	Objetivos del diseño	19
5.2	Arquitectura del sistema	20
5.2.1	Panorámica general	20
5.2.2	Descomposición en subsistemas funcionales	21
5.2.3	Descomposición en capas	21
5.3	Correspondencia entre hardware y software	22
5.4	Control de acceso y seguridad	23
5.5	Control global del sistema	23
5.6	Administración de datos persistentes	23
5.6.1	Gestión de la persistencia	23
5.6.2	Identificación de objetos	23
5.6.3	Eliminación de la herencia	24
5.6.4	Tratamiento de las relaciones	24

5.6.5	Esquema y tablas de la base de datos	24
5.6.6	Gestión de la concurrencia.....	24
5.7	Arranque y parada del sistema.....	25
5.8	Interfaz de usuario.....	25
5.8.1	Principios de diseño	25
5.8.2	Disposición general del interfaz	25
5.9	Subsistemas.....	26
5.9.1	Subsistema de registro y menú inicial	26
5.9.2	Subsistema de gestión de peticiones	27
5.9.3	Subsistema de gestión de actividades	27
5.9.4	Subsistema de generación de informes.....	27
5.9.5	Subsistema de administración.....	28
5.9.6	Subsistema de gestión de la persistencia	28
5.9.7	Subsistema de gestión de la concurrencia	28
6	Implementación	29
6.1	Aplicación del paradigma MVC.....	29
6.2	Estructura de los módulos del sistema.....	29
6.3	Implementación de la estructura de capas	30
6.3.1	La capa Cliente	30
6.3.2	La capa Web.....	30
6.3.3	La capa de Negocio	30
6.3.4	La capa de Persistencia.....	31
6.4	Implementación de los subsistemas	33
6.4.1	Subsistema de registro y menú inicial	33
6.4.2	Subsistema de gestión de peticiones.....	34
6.4.3	Subsistema de gestión de actividades	34
6.4.4	Subsistema de generación de informes.....	35
6.4.5	Subsistema de administración.....	37
6.4.6	Subsistema de gestión de la persistencia	37
6.4.7	Gestión de bloqueos	37
7	Instalación y uso del sistema	39
7.1	Inventario de la entrega.....	39
7.2	Instalación y configuración del sistema.....	39
7.2.1	Entorno de ejecución	40
7.2.2	Aplicación SGAPA	41
7.3	Guía de uso.....	41
8	Conclusiones	42
8.1	Sobre el Plan de Proyecto.....	42
8.2	Sobre la metodología utilizada	42
8.3	Sobre la consecución de objetivos	43
8.3.1	Objetivos técnicos	43
8.3.2	Objetivos académicos	43

8.4	Líneas futuras de actuación	45
8.4.1	Desarrollo posterior del sistema SGAPA	45
8.4.2	Estudios en el área Web y JEE.....	47
ANEXO A:	Glosario de términos	48
ANEXO B:	Bibliografía.....	50
ANEXO C:	Diagrama de Gant del proyecto.....	51
ANEXO D:	Diagrama de entidades del dominio	52
ANEXO E:	Diagramas de Casos de Uso.....	53
ANEXO F:	Esquema de la estructura en capas del sistema	55
ANEXO G:	Esquema de la base de datos	56
ANEXO H:	Secuencia de acceso a los servicios de Negocio	57

1 Introducción

Este documento contiene la memoria de los trabajos desarrollados por su autor para la especificación, diseño, construcción y pruebas de un sistema informático basado en tecnologías JEE objeto de su Trabajo Fin de Carrera.

2 Documentos asociados

El presente documento se relaciona con los siguientes:

- SGAPA-PDP-01 Plan de proyecto
- SGAPA-EAR-01 Especificación y Análisis de Requisitos
- SGAPA-DDS-01 Diseño del Sistema
- SGAPA-GDU-01 Guía de Uso
- SGAPA-IDE-01 Informe de entrega
- SGAPA-PVP-01 Presentación virtual del proyecto

3 Descripción, objetivos y planificación del proyecto

El proyecto objeto del Trabajo Fin de Carrera del autor de este informe está basado en una situación real de su vida profesional. El problema que se expone a continuación existe y, en la fecha de edición de este documento, permanece sin resolver.

El sistema propuesto sería una solución válida y la parte del mismo desarrollada en el TFC utilizable en la práctica.

3.1 Antecedentes del Proyecto

3.1.1 Descripción del dominio del problema

La empresa **CLIENTE** tiene contratados con la empresa **PROVEEDOR**, en la que trabaja el autor de esta memoria, varios servicios de desarrollo, mantenimiento y soporte de aplicaciones en régimen de externalización.

Para solicitar trabajos al PROVEEDOR, la empresa CLIENTE emite **peticiones** de servicio. Cada petición pertenece a un **proyecto**. Un proyecto puede constar de una o más peticiones.

En una misma petición pueden trabajar uno o más **servicios** del PROVEEDOR. Cada **trabajador** del PROVEEDOR involucrado con el CLIENTE puede estar asignado a uno o más servicios.

En cada petición se define una serie de **hitos**. Un hito se caracteriza por una fecha en la que se deben entregar al CLIENTE un paquete de **productos** (software o documentación)

El PROVEEDOR designa a uno de los trabajadores como **responsable** de todos los servicios ante el CLIENTE.

El responsable de los servicios debe entregar al CLIENTE, a mes vencido, la información siguiente:

- Informe del número de horas dedicadas durante el mes anterior a las peticiones del CLIENTE por parte del conjunto de los trabajadores asignados al servicio, desglosadas por proyecto y petición.
- Informe de los productos entregados en el mes anterior y de la previsión de productos a entregar en el mes siguiente, desglosados por proyecto y tipo de producto.

Aunque no se entregue con carácter habitual, el responsable de los servicios deberá estar preparado para proporcionar al CLIENTE, cuando éste lo solicite, la información adicional siguiente:

- Informe detallado del número de horas dedicadas durante el mes anterior a una determinada petición del CLIENTE, desglosadas por tipo de **actividad** (análisis, diseño, desarrollo, pruebas, formación de usuarios, etc.) y por trabajador.

Para obtener la información necesaria para la elaboración de los informes mensuales, el responsable solicita a cada trabajador un informe con el número de horas que el trabajador ha dedicado a cada una de las peticiones.

Los trabajadores asignados a los servicios pueden trabajar en las dependencias propias del CLIENTE o en las del PROVEEDOR y utilizan equipos y aplicaciones proporcionados por ambas organizaciones. En todos los casos, los equipos utilizados por los trabajadores se encuentran conectados, en forma local o remota, a la red de datos del CLIENTE.

Por este motivo, los equipos y aplicaciones utilizados por los servicios deben ser compatibles con las plataformas tecnológicas establecidas por el CLIENTE y ser aprobados por éste.

3.1.2 Situación inicial de los sistemas

En el momento del lanzamiento del presente proyecto, la organización del PROVEEDOR carecía de un sistema específico para la gestión de las tareas administrativas que son de su responsabilidad y que se han descrito en el epígrafe anterior. Esta sigue siendo la situación a la fecha.

Las tareas necesarias para la recopilación y elaboración de los datos incluidos en los informes se desarrollan manualmente mediante el uso de diversas hojas de cálculo MS Excel.

3.1.3 Motivación del cambio

Se ha comprobado que la realización de los trabajos necesarios para la preparación de los informes mensuales ordinarios ocupa una jornada y media de trabajo al mes del responsable de los servicios. El mantenimiento de la hoja MS Excel de control de hitos requiere de una media de 30 minutos diarios del responsable y aproximadamente una hora semanal de cada trabajador.

Se estima, pues, que se dedica un total de entre 40 y 50 horas mensuales a estas tareas administrativas.

La introducción del sistema propuesto podría rebajar en unas 20 horas mensuales esta dedicación con el consiguiente aumento de la productividad de los servicios.

Por otra parte, el tratamiento manual de los datos que se realiza actualmente es muy propenso a errores, por lo que su automatización redundará en un importante incremento de la calidad del servicio.

3.2 Objetivos del proyecto

3.2.1 Metas técnicas

El proyecto se propone como principales metas técnicas las siguientes:

- Análisis y diseño de un sistema de gestión administrativa de las peticiones recibidas del CLIENTE y de las actividades desempeñadas por parte de los trabajadores asignados a los servicios del PROVEEDOR para atenderlas. El sistema propuesto se denominará ***Sistema de Gestión Administrativa de Peticiones y Actividades*** y se identificará en el resto del documento por sus siglas [SGAPA](#).
- La implementación de la funcionalidad de SGAPA relacionada con la gestión de las peticiones.

Las funciones principales asignadas al sistema SGAPA completo son las siguientes:

- Gestión de peticiones de servicio
 - Alta y mantenimiento de las peticiones de servicio recibidas, incluyendo la gestión de las fechas previstas y reales de los hitos correspondientes.
 - Generación del informe mensual de los productos entregados y de la previsión de productos a entregar.
- Gestión de actividades
 - Imputación por parte de los trabajadores de las actividades realizadas para cumplimentar las peticiones y del tiempo que han requerido (partes de trabajo diario)
 - Validación de los partes de trabajo por los responsables de los servicios.
 - Generación del informe mensual del número de horas dedicadas.
 - Generación de informes de dedicación detallados.
- Administración de usuarios y perfiles, servicios, tipos de actividad, tipos de producto, etc.

El sistema debe poseer las características adicionales siguientes:

- Sus funciones deberán ser accesibles mediante un interfaz de usuario basado en navegador Web.
- Los usuarios deberán identificarse mediante un código y una clave.
- Las funciones que así se determinen serán accesibles únicamente a los usuarios que posean el perfil adecuado.

- El sistema debe contar con los mecanismos que sean necesarios para soportar su uso concurrente por parte de múltiples usuarios, garantizando la integridad de los datos y su correcta actualización.

3.2.2 Metas académicas

Es de destacar que, al comienzo de este proyecto, el autor poseía una amplia experiencia profesional en el análisis, diseño e implementación de soluciones informáticas en entornos microinformáticos y cliente / servidor, pero no así en entornos Web.

Hasta dicho momento, había trabajado varios años en proyectos desarrollados con la herramienta de gestión documental Documentum que proporciona un marco de trabajo (WDK) basado en componentes sobre arquitectura J2EE. También había participado como analista funcional y jefe de proyecto en algún desarrollo con J2EE, pero nunca había diseñado o implementado personalmente soluciones con dichas tecnologías.

En consecuencia, desde el punto de vista académico, me planteaba alcanzar con este proyecto los siguientes objetivos adicionales:

- Profundizar en las particularidades del desarrollo de soluciones informáticas para empresas en un entorno Web.
- Introducirme en algunas de las principales tecnologías que se encuadran bajo la denominación JEE (anteriormente J2EE)
- Profundizar en el uso del paradigma de diseño MVC, ya utilizado por el autor en otros contextos tecnológicos.
- Profundizar en el uso de patrones de diseño orientados a objeto.
- Introducirme en el uso de las transformaciones XSLT para la generación de HTML a partir de contenidos XML.

3.3 Enfoque y metodología

3.3.1 Análisis inicial de riesgos

Teniendo en cuenta lo expuesto en el punto anterior y otras consideraciones, en la fase inicial del proyecto se procedió a realizar un análisis inicial de riesgos que se identificaron como más significativos los siguientes:

- La falta de experiencia del autor en el diseño e implementación de aplicaciones Web en general y, en particular, con las tecnologías JEE constituía el mayor riesgo del proyecto SGAPA.
- El alcance previsto para el sistema SGAPA completo se estimaba como excesivo considerando el tiempo disponible para su desarrollo.

Para mitigar estos riesgos, se plantearon las acciones y estrategias siguientes:

- Descomponer el sistema global en los dos módulos siguientes:
 - Módulo de gestión de peticiones de servicio.

- Módulo de gestión de la actividad de los trabajadores.
- Considerar como objetivo del TFC el análisis y diseño del sistema global y la implementación del módulo de peticiones de servicio.
- Desarrollar el proyecto en forma iterativa e incremental, comenzando con la construcción, en una fase del proyecto tan temprana fuera sea posible, de un prototipo mínimo que implemente un caso de uso representativo.
- Utilizar aquellas tecnologías JEE de las que se dispusiese, o se pudiera encontrar con facilidad, información abundante y, en particular, ejemplos de uso completos.

3.3.2 Metodología

Como consecuencia de los resultados del análisis de riesgos y en base a mi experiencia en el desarrollo de proyectos con alto riesgo técnico, se optó por un método de trabajo con las siguientes características:

- Desarrollo iterativo e incremental tanto del software en sí como de su documentación técnica.
- Dirigido por casos de uso en todas sus fases.
- Centrado en la arquitectura.

Estos son los atributos que definen a la metodología UP (Unified Process) que, de forma adaptada a un proyecto de pequeño volumen, es la que se ha seguido.

3.4 Planificación del proyecto

En aplicación de la metodología adoptada, el tiempo disponible para la realización del proyecto se distribuyó en seis iteraciones o ciclos según se resume en la tabla siguiente:

Hito	Actividades	Productos
26/03/07 Ciclo 1 / Interno	<ul style="list-style-type: none"> • Análisis de requisitos • Establecimiento del entorno técnico de desarrollo 	<ul style="list-style-type: none"> • Documento de especificación del sistema
16/04/07 Ciclo 2 / PEC2	<ul style="list-style-type: none"> • Diseño arquitectónico • Desarrollo del primer prototipo 	<ul style="list-style-type: none"> • Documento de diseño de la arquitectura del sistema • Primer prototipo
07/05/07 Ciclo 3 / Interno	<ul style="list-style-type: none"> • Desarrollo de la solución 	<ul style="list-style-type: none"> • Segundo prototipo
21/05/07 Ciclo 4 / PEC3	<ul style="list-style-type: none"> • Desarrollo de la solución • Elaboración del primer borrador de la guía de usuario 	<ul style="list-style-type: none"> • Borrador de la guía de usuario • Tercer prototipo
04/06/07 Ciclo 5 / Interno	<ul style="list-style-type: none"> • Pruebas de aceptación • Elaboración del primer borrador de la memoria 	<ul style="list-style-type: none"> • Versión 1.0 del sistema • Documento de pruebas de aceptación • Borrador inicial de la memoria

18/06/07 Ciclo 6 / Final	<ul style="list-style-type: none"> • Elaboración de la memoria • Elaboración de la presentación virtual 	<ul style="list-style-type: none"> • Memoria del TFC • Presentación virtual del TFC
---------------------------------	---	---

La entregas formales al Consultor, de los productos generados durante los ciclos 1, 3 y 5 (internos) se realizó junto con los de los ciclos 2, 4 y 6, respectivamente, coincidiendo con las fechas de finalización de las PEC 2 y 3 y la entrega final.

En el [Anexo C](#) se incluye un diagrama de Gant del proyecto.

3.5 Productos generados

Los productos desarrollados durante el proyecto han sido los siguientes:

- Plan del proyecto
- Documento de especificación del sistema
- Documento de diseño de la arquitectura del sistema
- Software correspondiente al módulo de Gestión de Peticiones de Servicio de la Aplicación SGAPA (incluyendo fuentes y ejecutables)
- Informe de la entrega
- Documento de resultado de las pruebas de sistema.
- Memoria del TFC
- Presentación virtual del TFC

4 Especificación y análisis de requisitos

4.1 Panorámica general

El sistema que se propone - SGAPA - debe dar soporte a la gestión administrativa de las peticiones recibidas del CLIENTE y de las actividades desempeñadas por parte de los trabajadores asignados a los servicios del PROVEEDOR para atenderlas. Los principales requisitos de usuario que debe satisfacer el sistema se especifican en el [Catálogo de requerimientos del cliente](#), más adelante.

Para facilitar su análisis, diseño y desarrollo, el sistema se considera formado, desde una perspectiva funcional, por un módulo de Gestión de Peticiones y por un módulo de Gestión de Actividades. Las funciones que deberán proporcionar cada uno de los módulos, expresadas en forma de Casos de Uso, son las siguientes:

- **Módulo 01: Gestión de Peticiones**
 - [CU_0101: Alta de Proyecto](#)
 - [CU_0102: Baja de Proyecto](#)
 - [CU_0103: Consulta de Proyecto](#)

- [CU_0104: Edición de Proyecto](#)
- [CU_0105: Alta de Petición](#)
- [CU_0106: Baja de Petición](#)
- [CU_0107: Consulta de Petición](#)
- [CU_0108: Edición de Petición](#)
- [CU_0109: Alta de Hito](#)
- [CU_0110: Baja de Hito](#)
- [CU_0111: Consulta de Hito](#)
- [CU_0112: Edición de Hito](#)
- [CU_0113: Generación del Informe de Entregas](#)

- **Módulo 02: Gestión de Actividades**

- [CU_0201: Alta de Trabajador](#)
- [CU_0202: Baja de Trabajador](#)
- [CU_0203: Edición de Trabajador](#)
- [CU_0204: Alta de Servicio](#)
- [CU_0205: Baja de Servicio](#)
- [CU_0206: Consulta de Servicio](#)
- [CU_0207: Edición de Servicio](#)
- [CU_0208: Asignación de Trabajadores a Servicios](#)
- [CU_0209: Gestión del Parte de Actividades](#)
- [CU_0210: Validación de Partes de Actividad](#)
- [CU_0211: Generación del Informe Mensual de Dedicación](#)
- [CU_0212: Generación del Informe Detallado de Dedicación](#)
- [CU_0213: Gestión de Periodos](#)
- [CU_0214: Alta de Cuenta interna](#)
- [CU_0215: Baja de Cuenta interna](#)
- [CU_0216: Consulta de Cuenta interna](#)
- [CU_0217: Edición de Cuenta interna](#)

4.2 Catálogo de requerimientos del cliente

El sistema a desarrollar deberá satisfacer los requerimientos del CLIENTE siguientes:

Código	Requerimiento	Tipo
RU_001	El sistema dará soporte al alta y mantenimiento de las peticiones de servicio recibidas, incluyendo la gestión de las fechas previstas y reales de los hitos correspondientes.	FUN
RU_002	El sistema dará soporte a la imputación por parte de los trabajadores de las actividades realizadas para cumplimentar las peticiones y del tiempo que han requerido (partes de trabajo diario)	FUN
RU_003	El sistema dará soporte a la validación de los partes de trabajo por los responsables de los servicios.	FUN
RU_004	El sistema dará soporte a la generación de los informes mensuales de dedicación y de entregas.	FUN
RU_005	El sistema dará soporte a la generación de informes de dedicación detallados.	FUN
RU_006	El sistema dará soporte a la administración de usuarios, servicios, proyectos y otras entidades auxiliares.	FUN

RU_007	Las funciones del sistema directamente accesibles para los usuarios deberán serlo a través de un interfaz de usuario basado en navegador Web.	NO FUN
RU_008	Para acceder a las funciones proporcionadas por el sistema, los usuarios deberán identificarse mediante un código y una clave.	NO FUN
RU_009	Las funciones del sistema que así se determine serán accesibles únicamente a los usuarios que posean el perfil adecuado.	NO FUN
RU_010	El sistema contará con los mecanismos que sean necesarios para soportar su uso concurrente por parte de múltiples usuarios, garantizando la integridad de los datos y su correcta actualización.	NO FUN
RU_011	El sistema deberá ser compatible con un servidor estándar de Serv-lets Java.	NO FUN

4.3 Actores del sistema

Los agentes (actores) que interactuarán con el sistema son los siguientes:

Código	Denominación	Descripción	Amplía a
ACT_001	Trabajador	Representa a cualquier usuario autorizado para utilizar el sistema.	No aplica.
ACT_002	Responsable	Representa a un usuario Responsable de Servicio.	Trabajador
ACT_003	Administrador	Representa a un usuario con capacidad para administrar el sistema.	Responsable

4.4 Requisitos del sistema

4.4.1 De almacenamiento de la información

El sistema gestiona el almacenamiento persistente de las entidades que se especifican a continuación. Puede encontrarse una descripción detallada de los atributos de cada entidad en la sección de *Requisitos de almacenamiento de la información* del documento *SGAPA-EAR-01 Especificación y análisis de requisitos*.

Código	Denominación	Descripción
RAI_001	Proyecto	Agrupar los atributos que definen un Proyecto del CLIENTE.
RAI_002	Cuenta	Agrupar los atributos que definen a una Cuenta en la que se cargan las Actividades realizadas por los trabajadores.
RAI_003	Petición	Es un subtipo de Cuenta. Agrupa los atributos de una Cuenta de tipo Petición que son específicos de dicho tipo. Una Petición se encuentra siempre asociada a un Proyecto.
RAI_004	Cuenta interna	Es un subtipo de Cuenta. Agrupa los atributos de una Cuenta de tipo Cuenta interna que son específicos de dicho tipo.

Código	Denominación	Descripción
RAI_005	Hito	Agrupar los atributos de un Hito. Cada Hito es de un Tipo de Hito determinado y produce un producto o paquete de productos de un Tipo de Producto dado. Un Hito se encuentra siempre asociado a una Cuenta de tipo Petición.
RAI_006	Tipo de hito	Define un tipo de Hito.
RAI_007	Tipo de Producto	Define un tipo de Producto
RAI_008	Servicio	Agrupar los atributos que definen un Servicio que el PROVEEDOR presta al CLIENTE.
RAI_009	Trabajador	Agrupar los atributos que definen a un Trabajador del PROVEEDOR. Un trabajador está asociado a uno o más Servicios y realiza una o más Actividades.
RAI_010	Actividad	Agrupar los atributos de una Actividad que realiza un Trabajador en un día concreto y que se carga a una Cuenta.
RAI_011	Tipo de actividad	Agrupar los atributos que definen un tipo de Actividad.
RAI_012	Periodo	Agrupar los atributos que definen un periodo contable.

Las relaciones entre las entidades se muestran en la sección de [Modelo de objetos del dominio](#) en la forma de un Diagrama de Clases UML.

4.4.2 Funcionales

Las interacciones de los actores con el sistema, expresadas en la forma de Casos de Uso resumidos, son las que se relacionan a continuación. Puede encontrarse una descripción detallada de cada uno de los casos de uso en el apartado de *Requisitos funcionales* del documento *SGAPA-EAR-01 Especificación y Análisis de Requisitos*.

Código	Denominación y descripción
CU_0001	Registro y acceso al menú inicial El sistema solicita al usuario su identificador y clave de acceso. El usuario introduce las credenciales y el sistema las valida. Si las credenciales no son válidas, el sistema vuelve a solicitarlas. Si son válidas, el sistema configura los menús de la aplicación e función del perfil del usuario y muestra la ventana de bienvenida de usuario conectado.
CU_0002	Finalizar sesión El sistema termina la sesión del usuario y presenta la pantalla de bienvenida del sistema.
CU_0101	Alta de Proyecto El usuario introduce los datos correspondientes a un nuevo Proyecto en el formulario que se le presenta. El sistema valida los datos y almacena la nueva entidad. Si los datos no son válidos, el sistema muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.

Código	Denominación y descripción
CU_0102	<p>Baja de Proyecto</p> <p>El usuario selecciona el Proyecto a eliminar. El sistema solicita al usuario confirmación de la orden y, si la obtiene, verifica que no tenga relaciones con otras entidades que impidan su baja y procede a la eliminación. Si existen dichas relaciones, el sistema muestra al usuario un mensaje indicando el error y la baja no se produce.</p>
CU_0103	<p>Consulta de Proyecto</p> <p>El usuario selecciona el Proyecto a consultar. El sistema muestra una ficha con los datos de la entidad seleccionada.</p>
CU_0104	<p>Edición de Proyecto</p> <p>El usuario selecciona el Proyecto a editar. El sistema muestra un formulario relleno con los datos de la entidad seleccionada y el usuario realiza las modificaciones pertinentes en su contenido. El sistema valida los datos y almacena la nueva entidad. Si los datos no son válidos, muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p>
CU_0105	<p>Alta de Petición</p> <p>El usuario introduce los datos correspondientes a una nueva Petición en el formulario que se le presenta. El sistema valida los datos y almacena la nueva entidad. Si los datos no son válidos, el sistema muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p>
CU_0106	<p>Baja de Petición</p> <p>El usuario selecciona la Petición a eliminar. El sistema solicita al usuario confirmación de la orden y, si la obtiene, verifica que no tenga relaciones con otras entidades que impidan su baja y procede a la eliminación. Si existen dichas relaciones, el sistema muestra al usuario un mensaje indicando el error y la baja no se produce.</p>
CU_0107	<p>Consulta de Petición</p> <p>El usuario selecciona la Petición a consultar. El sistema muestra una ficha con los datos de la entidad seleccionada.</p>
CU_0108	<p>Edición de Petición</p> <p>El usuario selecciona la Petición a editar. El sistema muestra un formulario relleno con los datos de la entidad seleccionada y el usuario realiza las modificaciones pertinentes en su contenido. El sistema valida los datos y almacena la nueva entidad. Si los datos no son válidos, muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p>
CU_0109	<p>Alta de Hito</p> <p>El usuario introduce los datos correspondientes a un nuevo Hito en el formulario que se le presenta. El sistema valida los datos y almacena la nueva entidad. Si los datos no son válidos, el sistema muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p>
CU_0110	<p>Baja de Hito</p> <p>El usuario selecciona el Hito a eliminar. El sistema solicita al usuario confirmación de la orden y, si la obtiene, procede a la eliminación.</p>
CU_0111	<p>Consulta de Hito</p> <p>El usuario selecciona el Hito a consultar. El sistema muestra una ficha con los datos de la entidad seleccionada.</p>

Código	Denominación y descripción
CU_0112	<p>Edición de Hito</p> <p>El usuario selecciona el Hito a editar. El sistema muestra un formulario relleno con los datos de la entidad seleccionada y el usuario realiza las modificaciones pertinentes en su contenido. El sistema valida los datos y almacena la nueva entidad. Si los datos no son válidos, muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p>
CU_0113	<p>Generación del informe de entregas</p> <p>El usuario indica las fechas que delimitan los datos a incluir en el informe en el formulario que le presenta el sistema. Este valida las fechas y genera y presenta por pantalla el informe. Si las fechas no son válidas, muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p> <p>Opcionalmente, el usuario puede solicitar la descarga del contenido del informe a un archivo local, en cuyo caso, el sistema le presentará un diálogo para que indique el nombre que desea para el archivo y el directorio local donde debe crearse. El sistema generará un archivo de texto con el contenido del informe.</p>
CU_0201	<p>Alta de Trabajador</p> <p>El usuario introduce los datos correspondientes a un nuevo Trabajador en el formulario que se le presenta. El sistema valida los datos y almacena la nueva entidad. Si los datos no son válidos, el sistema muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p>
CU_0202	<p>Baja de Trabajador</p> <p>El usuario selecciona el Trabajador a eliminar. El sistema solicita al usuario confirmación de la orden y, si la obtiene, verifica que no tenga relaciones con otras entidades que impidan su baja y procede a la eliminación. Si existen dichas relaciones, el sistema muestra al usuario un mensaje indicando el error y la baja no se produce.</p>
CU_0203	<p>Edición de Trabajador</p> <p>El usuario selecciona el Trabajador a editar. El sistema muestra un formulario relleno con los datos de la entidad seleccionada y el usuario realiza las modificaciones pertinentes en su contenido. El sistema valida los datos y almacena la nueva entidad. Si los datos no son válidos, muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p>
CU_0204	<p>Alta de Servicio</p> <p>El usuario introduce los datos correspondientes a un nuevo Servicio en el formulario que se le presenta. El sistema valida los datos y almacena la nueva entidad. Si los datos no son válidos, el sistema muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p>
CU_0205	<p>Baja de Servicio</p> <p>El usuario selecciona el Servicio a eliminar. El sistema solicita al usuario confirmación de la orden y, si la obtiene, verifica que no tenga relaciones con otras entidades que impidan su baja y procede a la eliminación. Si existen dichas relaciones, el sistema muestra al usuario un mensaje indicando el error y la baja no se produce.</p>
CU_0206	<p>Consulta de Servicio</p> <p>El usuario selecciona el Servicio a consultar. El sistema muestra una ficha con los datos de la entidad seleccionada.</p>

Código	Denominación y descripción
CU_0207	<p>Edición de Servicio</p> <p>El usuario selecciona el Servicio a editar. El sistema muestra un formulario relleno con los datos de la entidad seleccionada y el usuario realiza las modificaciones pertinentes en su contenido. El sistema valida los datos y almacena la nueva entidad. Si los datos no son válidos, muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p>
CU_0208	<p>Asignación de Trabajadores a Servicios</p> <p>El sistema muestra una relación de los Servicios registrados y el usuario selecciona uno de ellos. El sistema muestra una relación de los Trabajadores registrados en la que los ya asignados al servicio seleccionado aparecen marcados.</p> <p>El usuario añade o elimina marcas de asignación en la relación de Trabajadores y el sistema actualiza los datos de asignación de Trabajadores a Servicios en consecuencia.</p>
CU_0209	<p>Gestión del Parte de Actividades</p> <p>El sistema muestra una relación de las Actividades del usuario pertenecientes a la fecha que este le indica. El usuario añade nuevas Actividades, elimina las existentes o modifica sus datos.</p> <p>El sistema valida los datos y actualiza las Actividades. Si no se supera la validación, el sistema muestra al usuario un mensaje de error y no se realiza la actualización.</p>
CU_0210	<p>Validación de partes de Actividad</p> <p>El sistema muestra una relación de los Trabajadores registrados y el usuario selecciona uno de ellos. El sistema muestra las horas totales correspondientes a las Actividades del Trabajador registradas en el periodo activo y acumuladas por Cuenta. Si está conforme, el usuario valida los datos y el sistema registra la validación.</p> <p>Opcionalmente, el usuario solicita supervisar los Partes del Trabajador. El sistema muestra la relación de Actividades del Trabajador en la fecha que el usuario le indica. El usuario puede seleccionar una Actividad y el sistema le muestra sus detalles.</p>
CU_0211	<p>Generación del Informe Mensual de Dedicación</p> <p>El usuario indica el Servicio las fechas que delimitan los datos a incluir en el informe en el formulario que le presenta el sistema. Este valida los parámetros y genera y presenta por pantalla el informe. Si los parámetros no son válidos, muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p> <p>Opcionalmente, el usuario puede solicitar la descarga del contenido del informe a un archivo local, en cuyo caso, el sistema le presentará un diálogo para que indique el nombre que desea para el archivo y el directorio local donde debe crearse. El sistema generará un archivo de texto con el contenido del informe.</p>

Código	Denominación y descripción
CU_0212	<p>Generación del Informe Detallado de Dedicación</p> <p>El usuario indica el Servicio, la Petición y las fechas que delimitan los datos a incluir en el informe en el formulario que le presenta el sistema. Este valida los parámetros y genera y presenta por pantalla el informe. Si los parámetros no son válidos, muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p> <p>Opcionalmente, el usuario puede solicitar la descarga del contenido del informe a un archivo local, en cuyo caso, el sistema le presentará un diálogo para que indique el nombre que desea para el archivo y el directorio local donde debe crearse. El sistema generará un archivo de texto con el contenido del informe.</p>
CU_0213	<p>Gestión de Periodos</p> <p>El sistema muestra los datos del Periodo actual y el usuario introduce una fecha de cierre. El sistema asigna el estado de cerrado al periodo y crea un nuevo periodo actual cuya fecha de inicio sea la de cierre del anterior más un día.</p> <p>Opcionalmente, el usuario podrá reabrir el periodo anterior con lo que el sistema eliminará el actual.</p>
CU_0214	<p>Alta de Cuenta Interna</p> <p>El usuario introduce los datos correspondientes a una nueva Cuenta Interna en el formulario que se le presenta. El sistema valida los datos y almacena la nueva entidad. Si los datos no son válidos, el sistema muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p>
CU_0215	<p>Baja de Cuenta Interna</p> <p>El usuario selecciona la Cuenta Interna a eliminar. El sistema solicita al usuario confirmación de la orden y, si la obtiene, verifica que no tenga relaciones con otras entidades que impidan su baja y procede a la eliminación. Si existen dichas relaciones, el sistema muestra al usuario un mensaje indicando el error y la baja no se produce.</p>
CU_0216	<p>Consulta de Cuenta Interna</p> <p>El usuario selecciona la Cuenta Interna a consultar. El sistema muestra una ficha con los datos de la entidad seleccionada.</p>
CU_0217	<p>Edición de Cuenta Interna</p> <p>El usuario selecciona la Cuenta Interna a editar. El sistema muestra un formulario relleno con los datos de la entidad seleccionada y el usuario realiza las modificaciones pertinentes en su contenido. El sistema valida los datos y almacena la nueva entidad. Si los datos no son válidos, muestra al usuario de nuevo el formulario junto con un mensaje indicando el error.</p>

4.4.3 No funcionales

Otros requisitos del sistema, no funcionales, son los siguientes:

Requisito	Descripción	Categoría
RNF_001	Se proporcionará una Guía Básica de Uso.	Documentación

RNF_002	Las transacciones simples que no impliquen la creación de nuevos registros en la base de datos, su modificación o baja deberán ejecutarse en un tiempo máximo de 2 segundos medidos desde el instante en el que el usuario invoca la acción hasta el momento en el que se recibe confirmación de la acción o el interfaz de usuario vuelve a estar preparado para una nueva transacción.	Consideraciones de desempeño
RNF_003	Las transacciones simples que impliquen la creación de nuevos registros en la base de datos, su modificación o baja deberán ejecutarse en un tiempo máximo de 5 segundos medidos desde el instante en el que el usuario invoca la acción hasta el momento en el que se recibe confirmación de la acción o el interfaz de usuario vuelve a estar preparado para una nueva transacción.	Consideraciones de desempeño
RNF_004	En todas las situaciones en las que se produzca un error recuperable, el sistema deberá informar al usuario del problema, de su origen y de las acciones que el usuario debe realizar para que el sistema se recupere de la situación de error o este no se repita.	Manejo de errores y condiciones extremas
RNF_005	En todas las situaciones en las que se produzca un error irreparable, el sistema deberá informar al usuario del problema y de su origen y anotará en un archivo de bitácora (log) de ejecución información suficiente para el análisis de la situación por parte del personal de mantenimiento. El sistema tomará cuantas medidas sean apropiadas para asegurar la consistencia de la base de datos.	Manejo de errores y condiciones extremas
RNF_006	Se seguirán los estándares de calidad de la empresa PROVEEDOR.	Cuestiones de calidad
RNF_007	El sistema deberá ser ejecutable en un ambiente de oficina estándar tanto en su parte cliente como en su parte servidora.	Ambiente físico
RNF_008	El acceso al sistema estará restringido a los usuarios validados mediante identificador y clave.	Seguridad
RDD_001	Tanto la parte cliente como la parte servidora del sistema deberán ser ejecutables sobre equipos PC compatibles con arquitectura Intel estándar.	Plataforma física
RDD_002	La parte servidora del sistema deberá ser ejecutable en un servidor estándar de Servlets JEE. Se tomará como referencia el servidor Tomcat en su versión 4.1.	Plataforma lógica

RDD_003	La parte cliente del sistema deberá ser ejecutable en un navegador Web estándar, Se tomará como referencia el navegador Microsoft Internet Explorer en su versión 7.0.	Plataforma lógica
RDD_004	Se tomará como referencia de base de datos el sistema MySQL 5.0.37 Community Edition. Si fuera necesario, se podrá utilizar cualquier otro sistema de gestión de bases de datos relacional compatible con la arquitectura JEE.	Plataforma lógica
RDD_005	No se almacenará en el sistema información de carácter personal que requiera la aplicación de la Ley Orgánica de Protección de Datos.	Consideraciones normativas

4.5 Modelos del Análisis

4.5.1 Modelo de objetos del dominio

Se muestra en el [Anexo D](#) con el formato de diagrama UML.

4.5.2 Modelo de casos de uso

Se muestra en el [Anexo E](#) con el formato de diagrama UML.

5 Diseño del Sistema

5.1 Objetivos del diseño

Los objetivos de diseño planteados para el sistema SGAPA son los siguientes:

- El sistema dará satisfacción a los requisitos funcionales que, expresados en la forma de Casos de Uso, se especifican en el documento SGAPA-EAR-01 Especificación y Análisis de Requisitos.
- El sistema revestirá la forma de una aplicación Web compatible con los estándares JEE y con el paradigma MVC (Modelo, Vista, Controlador).
- Tanto la parte cliente como la parte Web del sistema deben ser ejecutables sobre equipos PC compatibles con arquitectura Intel estándar.
- La parte cliente de la aplicación debe ser ejecutable en un navegador Web estándar. Se tomará como referencia el navegador Microsoft Internet Explorer en su versión 7.0.
- La parte Web de la aplicación debe ser ejecutable en un servidor estándar de Servlets JEE. Se tomará como referencia el servidor Tomcat en su versión 4.1.
- La gestión de datos persistentes se realizará mediante el uso de un Sistema de Gestión de Bases de Datos Relacionales estándar. Se tomará como referencia de base de datos el sistema MySQL 5.0.37 Community Edition.

- El esquema de base de datos utilizado para el soporte de la persistencia de los objetos del Modelo deberá realizarse de forma que sea explotable de forma autónoma y externa al sistema SGAPA.
- En todas las situaciones en las que se produzca un error recuperable, el sistema deberá informar al usuario del problema, de su origen y de las acciones que el usuario debe realizar para que el sistema se recupere de la situación de error o este no se repita.
- En todas las situaciones en las que se produzca un error irrecuperable, el sistema deberá informar al usuario del problema y de su origen y anotará en un archivo de bitácora (log) de ejecución información suficiente para el análisis de la situación por parte del personal de mantenimiento.
- El sistema tomará cuantas medidas sean apropiadas para asegurar la consistencia de la base de datos.
- Las transacciones simples que no impliquen la creación de nuevos registros en la base de datos, su modificación o baja deberán ejecutarse en un tiempo máximo de 2 segundos. Las transacciones simples que impliquen la creación de nuevos registros en la base de datos, su modificación o baja deberán ejecutarse en un tiempo máximo de 5 segundos.

Los tiempos se medirán en un sistema en el que se encuentren instalados todos los componentes de la aplicación (navegador, servidor Web, servidor de base de datos y software de aplicación) de forma que puedan despreciarse los tiempos de retardo y de latencia debidos a los accesos a red.

Los tiempos se medirán desde el instante en el que el usuario invoca la acción hasta el momento en el que se recibe confirmación de la acción o el interfaz de usuario vuelve a estar preparado para una nueva transacción.

- El acceso al sistema estará restringido a los usuarios validados mediante identificador y clave. Se controlará e impedirá el acceso a los recursos de la aplicación mediante URL a usuarios no autorizados.
- El sistema contará con mecanismos para soportar su uso concurrente por parte de múltiples usuarios, garantizando la integridad de los datos y su correcta actualización.
- La arquitectura del sistema se realizará de forma que la mayor parte posible de sus componentes puedan utilizarse en un nuevo sistema de tipo cliente / servidor.

5.2 Arquitectura del sistema

5.2.1 Panorámica general

Para satisfacer los objetivos de diseño indicados, el sistema SGAPA se diseña como una aplicación Web a utilizar en un entorno de intranet utilizando los paradigmas y las tecnologías siguientes:

- Paradigma MVC (Modelo, Vista, Controlador) con

- JavaBeans y clases Java estándar para el Modelo.
- Java Server Pages y etiquetas personalizadas para la Vista.
- Struts Actions y componentes Java para el Controlador.
- Marco de persistencia basado en el patrón DAO / JDBC.
- Servidor de Aplicaciones Tomcat.

5.2.2 Descomposición en subsistemas funcionales

Para facilitar el uso del sistema, su desarrollo y su codificación e implantación por fases, el sistema se descompone, desde una perspectiva funcional, en subsistemas.

5.2.2.1 Subsistemas de usuario

- [Subsistema de registro y menú inicial](#)
- [Subsistema de gestión de peticiones](#)
- [Subsistema de gestión de actividades](#)
- [Subsistema de generación de informes](#)
- [Subsistema de administración](#)

5.2.2.2 Subsistemas de apoyo

Los subsistemas anteriores, con los que los usuarios tendrán una interfaz directa, serán complementados por los subsistemas de apoyo siguientes:

- [Subsistema de gestión de la persistencia](#)
- [Subsistema de gestión de la concurrencia](#)

5.2.3 Descomposición en capas

El sistema se estructura mediante una arquitectura en capas, tal y como se muestra en el diagrama que se incluye en el [Anexo F](#) y se describe a continuación.

5.2.3.1 Capa Cliente

La capa cliente está implementada por un navegador Web estándar. Se toma como referencia el navegador Microsoft Internet Explorer en su versión 7.0.

Dado que se utilizan funciones JavaScript para mostrar diálogos de confirmación y de advertencia, el navegador utilizado debe tener activada esta opción.

5.2.3.2 Capa Web

Se implementa utilizando páginas JSP y Clases de tipo Action del marco Struts. Las páginas JSP constituyen la parte Vista y las clases Struts Action la parte Controlador del patrón MVC correspondiente a la capa Web.

El acceso desde esta capa a los recursos de la capa de Negocio se realiza a través del interfaz INegocioSGAPA proporcionado por esta última. Las clases Struts Action utilizan esta interfaz para acceder al Modelo y a la lógica del Negocio, delegando la acción a clases controladoras del nivel inferior.

5.2.3.3 Capa de Negocio

Encapsula los datos y la lógica de negocio del Modelo de SGAPA mediante clases Java estándar. Implementa el interfaz INegocioSGAPA y define y utiliza, a su vez el interfaz IPersistenciaSGAPA para acceder a los servicios de la capa de Persistencia.

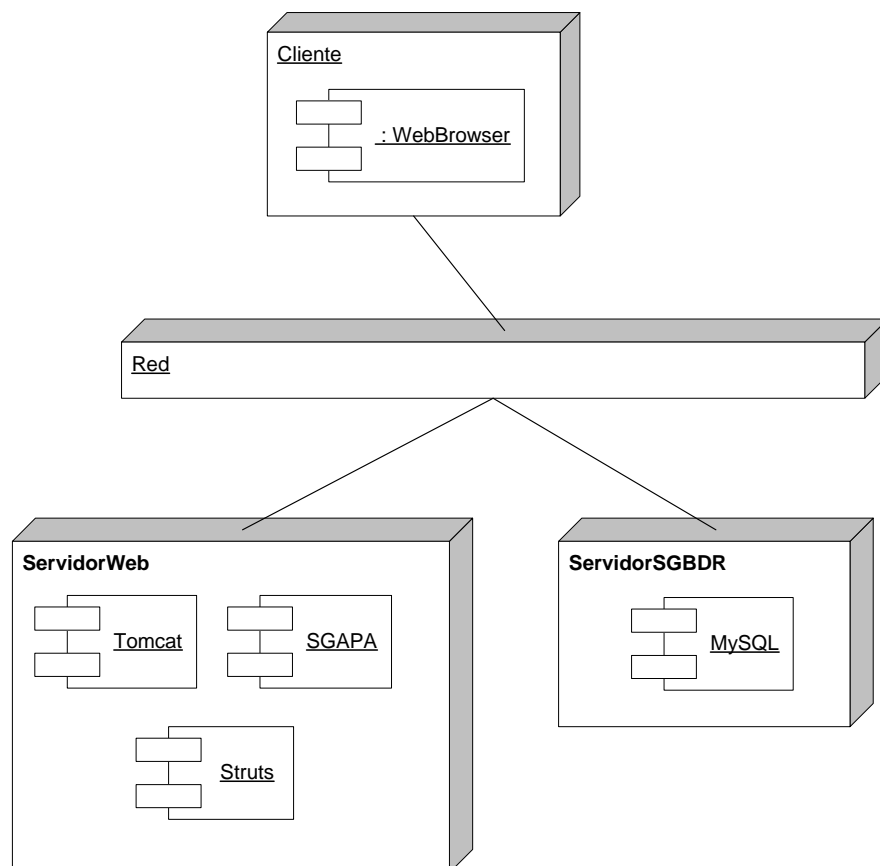
Se codifica de forma que, si fuera necesario, pueda ejecutarse sobre una Máquina Virtual Java estándar independiente de cualquier tipo de servidor Web, con el objeto de que pueda formar parte de un eventual sistema cliente servidor.

5.2.3.4 Capa de Persistencia

Importa e implementa el interfaz IPersistenciaSGAPA proporcionando a la capa de Negocio los servicios de persistencia para las clases de dicha clase que lo precisen. Utiliza clases Java de tipo DAO que acceden a la base de datos vía JDBC.

5.3 Correspondencia entre hardware y software

El despliegue de los módulos que componen el sistema SGAPA se realizará, conceptualmente, en la forma que se indica en el diagrama UML de implementación siguiente:



5.4 Control de acceso y seguridad

El acceso al sistema está restringido, de forma que solamente puedan utilizarlo los usuarios autorizados. Los usuarios se identifican mediante el uso de credenciales formadas por un identificador de usuario y una clave de acceso.

La acreditación se realiza mediante el uso de un formulario, página JSP, específico, lo que se indica en la sección correspondiente del archivo web.xml de la aplicación.

La autorización para el acceso a los diversos recursos de la aplicación se controla mediante el uso de roles de usuarios. Se utiliza una parcela (Realm) de seguridad basada en acceso a base de datos vía JDB, para lo que se especifica en el archivo sever.xml del servidor de aplicaciones Tomcat el uso de las tablas TRABAJADORES y ROLES con este fin.

Las restricciones de acceso a los procesos por parte de los usuarios, en función de su perfil, se especifica en *el Anexo C – Permisos* del documento *SGAPA-DDS-01 – Diseño del sistema*.

5.5 Control global del sistema

El sistema está bajo el control global del servidor Web y, en particular, del Servlet controlador de Struts que se encarga de determinar las clases Action Struts a invocar en cada momento como respuesta a las peticiones del usuario.

Las mencionadas peticiones de acción de los usuarios pueden corresponder a la invocación directa de una URL desde el navegador o, lo que es más habitual, a la invocación de enlaces contenidos en páginas JSP de la aplicación.

5.6 Administración de datos persistentes

5.6.1 Gestión de la persistencia

Se utiliza un Sistema de Gestión de Bases de Datos Relacionales, SGBDR, para soportar la persistencia de aquellas entidades del dominio de la aplicación SGAPA que lo requieren.

Las funciones de acceso al SGBDR para la inserción, búsqueda, modificación y eliminación de los registros que materializan la persistencia de los objetos del modelo de la aplicación se agrupan en un módulo de servicio. Este módulo se describe en la sección [Subsistema de gestión de la persistencia](#).

Se toma como referencia para el desarrollo y pruebas de la fase inicial del proyecto SGAPA el SGBDR MySQL 5.0.37 Community Edition.

5.6.2 Identificación de objetos

Con el objeto de facilitar la posibilidad de realizar consultas a la base de datos externas y ajenas al sistema SGAPA, el esquema de la base de datos relacional se ha diseñado de forma que sea lo más independiente posible del modelo de objetos de la aplicación. En particular, las claves primarias de las tablas de la base de datos están, en general, constituidas por columnas, o combinaciones de columnas, de valores significativos, no arbitrarios.

Así, por ejemplo, una instancia de Proyecto se materializa en forma persistente en una fila de la tabla PROYECTOS. La clave primaria de esta tabla está constituida por la columna ID_PROYECTO cuyo contenido es el código identificador del Proyecto.

Las relaciones entre tablas se materializan mediante el uso de referencias de claves foráneas en la forma que se detalla más adelante.

5.6.3 Eliminación de la herencia

El Modelo del dominio de la aplicación SGAPA contiene una única jerarquía de clases: la constituida por la clase base Cuenta y sus subclases Petición y Cuenta interna.

Debe tenerse en cuenta que, mientras que las instancias de la entidad Actividad pueden tener relación con cualquier tipo de Cuenta (Petición o Cuenta interna), los Hitos solo se relacionan con las Cuentas de tipo Petición.

Considerando lo anterior, de las diferentes formas de eliminación de la herencia disponibles, en este caso se ha optado por incluir en el esquema de la base de datos una tabla por cada clase aplicando el patrón de diseño denominado por [Fowler](#) como Class Table Inheritance. Esta opción permite utilizar las facilidades proporcionadas por el SGBDR para garantizar la integridad referencial entre los objetos Actividad y Cuenta, por una parte, y Petición e Hito, por otra.

5.6.4 Tratamiento de las relaciones

Las relaciones 1 a N se manejan introduciendo en las filas de las tablas del lado N de la relación una columna cuyo contenido sea una referencia, clave foránea, al registro de la tabla del lado 1 de la relación.

El Modelo del dominio de la aplicación SGAPA contiene un único caso de relación N a M: la establecida entre las clases Servicio y Trabajador. Esta relación se maneja mediante la introducción de la tabla ASIGNACIONES. Cada fila de esta tabla contendrá sendas referencias, claves foráneas, a los registros de SERVICIOS y TRABAJADORES afectados.

5.6.5 Esquema y tablas de la base de datos

El esquema de la base de datos se muestra en el [Anexo G](#) de este documento.

Puede encontrarse una especificación detallada de cada una de las tablas en el *Anexo B – Base de datos* del documento *SGAPA-DDS-01 – Diseño del sistema*.

5.6.6 Gestión de la concurrencia

Se ha adoptado una estrategia de bloqueo pesimista mediante la cual se impide el acceso a las funciones del sistema que puedan implicar la modificación de datos persistentes cuando se encuentre ya en curso una operación del mismo tipo.

El mecanismo para la implementación de esta estrategia se basa en la utilización de un sistema de semáforos, independiente de la base de datos relacional, que registra las concesiones de bloqueo realizadas, con indicación del identificador del recurso bloqueado y del identificador del usuario al que se concede el bloqueo.

Este mecanismo se describe en la sección de [Gestión de bloqueos](#), más adelante.

5.7 Arranque y parada del sistema

El arranque y la parada de la aplicación se encuentran bajo el control del servidor Web. No es necesario, en general, realizar acciones específicas para el arranque o parada del sistema.

No obstante, dado que, con el objeto de evitar los tiempos asociados al establecimiento y cierre de las conexiones con la base de datos, se utiliza una agrupación (pool) de conexiones que será utilizada por todos los procesos que necesiten acceder a la base de datos, durante el arranque de la aplicación deberá ejecutarse el Servlet responsable de la inicialización de dicha agrupación.

Con este fin, el Servlet DBInitServlet se declara en el archivo web.xml de la aplicación de forma que sea invocado por Tomcat durante el arranque.

La creación y gestión de la agrupación de conexiones se describe más adelante con detalle en la sección de [Agrupación de conexiones](#).

5.8 Interfaz de usuario

En el diseño de la interfaz de usuario se ha dado la máxima prioridad a la facilidad de uso de forma que la curva de aprendizaje del usuario sea lo más rápida posible.

5.8.1 Principios de diseño

Con este objetivo se han aplicado los principios de diseño siguientes:

1. Principio de agrupamiento. El espacio visible se organiza en bloques claramente separados en los que se sitúan controles similares y, cuando se considera apropiado, con un título para cada bloque.
2. Principio de visibilidad y utilidad. Los controles usados con mayor frecuencia se disponen de forma que sean siempre visibles y de fácil acceso.
3. Principio de la consistencia inteligente. Se utiliza una distribución de la información similar para funciones similares.
4. Principio de la economía en el diseño. Se omite cualquier elemento que no aporte información.
5. Principio del color como suplemento. El color se utiliza exclusivamente para aportar énfasis a la información sin convertirse en el único medio para comunicarla.
6. Principio de reducción del desorden. Se utiliza un único tipo de letra y con un máximo de tres tamaños distintos de carácter.

5.8.2 Disposición general del interfaz

Las pantallas que componen el interfaz de usuario presentan una estructura de tres paneles.

El primero, situado en la parte superior, incluye el logotipo e identifica la aplicación en uso y la versión de la misma.

El segundo panel, situado a la izquierda, contiene el menú general de la aplicación.

El tercero, situado a la derecha, se dedica al área principal de trabajo. Está, a su vez, dividido verticalmente en dos secciones. La superior contiene el submenú de opciones propias de la funcionalidad concreta que se está ejecutando. La superior se destina a la presentación y edición de datos.

Menú General [Atrás](#) [Editar](#) [Añadir Petición](#) [Baja](#)

[Peticiones](#)
[Actividades](#)
[Informes](#)
[Administración](#)

Consulta de Proyecto

Código de proyecto: PRO_001
Descripción: Proyecto de prueba 1
Activo: S

Lista de Peticiones

Código	Descripción
PET 01	Peticion 1 de Proyecto de prueba 1
PET 02	Peticion 2 de Proyecto de prueba 1
PET 03	Peticion 3 de Proyecto de prueba 1
PET INT	Test de coincidencia PET_INT

5.9 Subsistemas

A continuación se hace una relación de los subsistemas que componen la solución SGAPA, indicando para cada uno de ellos los casos de uso que implementan, los procesos adicionales los que dan soporte y sus interfaces con otros subsistemas. En la sección *Subsistemas* del documento *SGAPA-DDS-01 – Diseño del sistema* se puede encontrar una extensa y detallada descripción de los mismos.

Puede encontrarse una descripción de la arquitectura adoptada para cada subsistema en la sección de [Implementación](#) de este documento.

Es de destacar que, debido a la particular importancia que tiene la navegación entre ventanas en el diseño de las aplicaciones Web y a la dificultad que representa su especificación por medio de descripciones textuales, en la elaboración de dicho documento se recurrió al uso del [lenguaje visual de especificación de la interacción](#) propuesto por Jesse James Garrett.

El autor de esta memoria considera de gran utilidad el uso de este lenguaje, de forma que lo ha incorporado a su ámbito profesional encontrando un alto grado de aceptación por parte de los desarrolladores y otros usuarios de la documentación de diseño.

5.9.1 Subsistema de registro y menú inicial

El subsistema de registro y menú inicial implementa los Casos de Uso siguientes:

- CU_0001: Registro y acceso al menú inicial
- CU_0002: Finalizar sesión

Adicionalmente, da soporte a los procesos siguientes:

- Reconexión del usuario al caducar la sesión.

El subsistema no presenta un interfaz explícito para el acceso a sus servicios por parte de otros subsistemas y utiliza el interfaz IPersistenciaSGAPA para acceder a los servicios proporcionados por el subsistema de Persistencia.

5.9.2 Subsistema de gestión de peticiones

El subsistema de gestión de peticiones implementa los Casos de Uso siguientes:

- CU_0101: Alta de Proyecto
- CU_0102: Baja de Proyecto
- CU_0103: Consulta de Proyecto
- CU_0104: Edición de Proyecto
- CU_0105: Alta de Petición
- CU_0106: Baja de Petición
- CU_0107: Consulta de Petición
- CU_0108: Edición de Petición
- CU_0109: Alta de Hito
- CU_0110: Baja de Hito
- CU_0111: Consulta de Hito
- CU_0112: Edición de Hito

El subsistema implementa una parte del interfaz INegocioSGAPA y utiliza el interfaz IPersistenciaSGAPA para acceder a los servicios proporcionados por el subsistema de Persistencia.

5.9.3 Subsistema de gestión de actividades

El subsistema de gestión de actividades implementa los Casos de Uso siguientes:

- CU_0205: Gestión del Parte de Actividades
- CU_0306: Validación de Partes de Actividad

El subsistema implementa una parte del interfaz INegocioSGAPA y utiliza el interfaz IPersistenciaSGAPA para acceder a los servicios proporcionados por el subsistema de Persistencia.

5.9.4 Subsistema de generación de informes

El subsistema de gestión de informes implementa los Casos de Uso siguientes:

- CU_0113: Generación del Informe de Entregas
- CU_0207: Generación del Informe Mensual de Dedicación
- CU_0208: Generación del Informe Detallado de Dedicación

El subsistema implementa una parte del interfaz INegocioSGAPA y utiliza el interfaz IPersistenciaSGAPA para acceder a los servicios proporcionados por el subsistema de Persistencia.

5.9.5 Subsistema de administración

El subsistema de administración implementa los Casos de Uso siguientes:

- CU_0105: Alta de Servicio
- CU_0106: Baja de Servicio
- CU_0107: Consulta de Servicio
- CU_0108: Edición de Servicio
- CU_0201: Alta de Trabajador
- CU_0202: Baja de Trabajador
- CU_0203: Edición de Trabajador
- CU_0204: Asignación de Trabajadores a Servicios
- CU_0209: Gestión de Periodos
- CU_0210: Alta de Cuenta interna
- CU_0211: Baja de Cuenta interna
- CU_0212: Consulta de Cuenta interna
- CU_0213: Edición de Cuenta interna

El subsistema implementa una parte del interfaz INegocioSGAPA y utiliza el interfaz IPersistenciaSGAPA para acceder a los servicios proporcionados por el subsistema de Persistencia.

5.9.6 Subsistema de gestión de la persistencia

El subsistema de gestión de la persistencia proporciona los servicios siguientes:

- Alta, búsqueda, consulta, modificación y baja de entidades persistentes.
- Alta, búsqueda, consulta, modificación y baja de credenciales de usuarios.
- Adquisición, consulta y liberación de bloqueos de acceso a entidades.

El subsistema implementa el interfaz IPersistenciaSGAPA para el acceso a sus servicios por parte de otros subsistemas y no utiliza interfaces para acceder a los servicios proporcionados por otros subsistemas.

5.9.7 Subsistema de gestión de la concurrencia

Aunque en el diseño se consideró que la gestión de la concurrencia en el acceso a los recursos era competencia del subsistema de gestión de la persistencia, durante el desarrollo se concluyó que esta responsabilidad debía asignarse a un subsistema propio.

El subsistema de gestión de la concurrencia proporciona los servicios siguientes:

- Bloqueo y liberación de recursos
- Obtención una relación de los recursos bloqueados

El subsistema implementa una parte del interfaz INegocioSGAPA y no utiliza interfaces para acceder a los servicios proporcionados por otros subsistemas.

6 Implementación

6.1 Aplicación del paradigma MVC

El sistema ESGAPA 1.0 es una aplicación Web desarrollada con tecnologías JEE siguiendo el paradigma de diseño conocido como MVC (Modelo, Vista, Controlador) con una arquitectura de capas: Cliente, Web, Negocio y Persistencia.

La parte Vista del paradigma se ha implementado mediante el uso de Java Server Pages (JSP) codificadas con el auxilio de etiquetas personalizadas proporcionadas por el marco de trabajo Struts que forma parte de la solución.

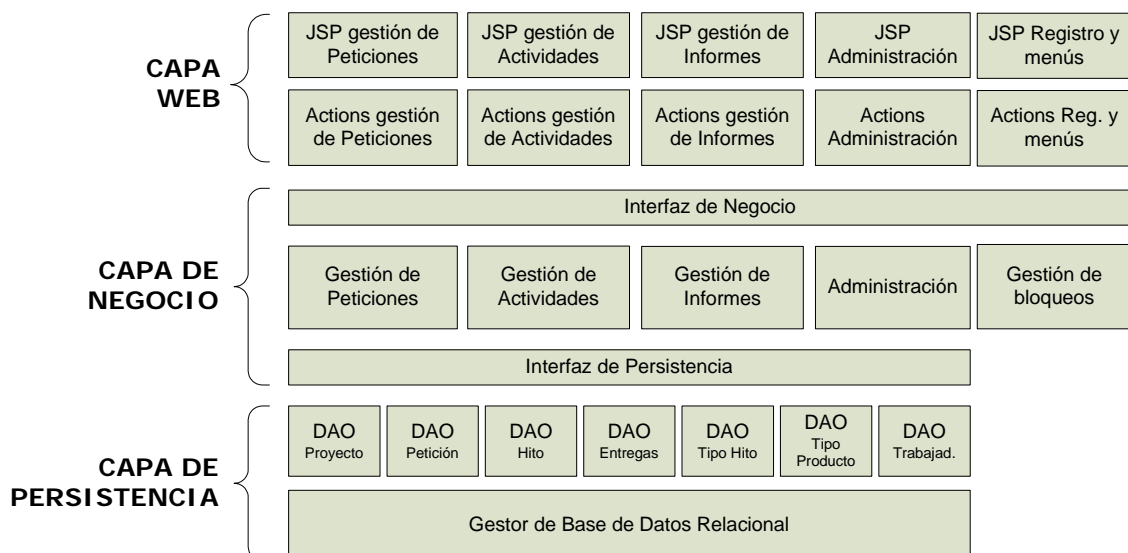
La parte correspondiente al Controlador se ha implementado utilizando clases Java derivadas de la clase Action de Struts, situadas en la capa Web, que para realizar sus funciones recurren a los servicios proporcionados, a través del interfaz INegocioSGAPA, por otras clases Java situadas en la capa de Negocio.

El Modelo está constituido por clases Java pertenecientes a la capa de Negocio. La comunicación de datos entre las clases controladoras y las JSP se realiza por medio de objetos de tipo ActionForms de Struts. En ningún caso la Vista accede directamente al Modelo.

Las comunicaciones de datos entre las clases controladoras y las correspondientes al Modelo se realizan mediante objetos de tipo Bean pertenecientes al Negocio. Las clases controladoras se ocupan de transferir los datos necesarios en cada caso entre los bean de la capa de Negocio y las ActionForm de la capa web.

6.2 Estructura de los módulos del sistema

Los diferentes módulos de software que componen el sistema SGAPA se agrupan en subsistemas distribuidos en capas según lo especificado en el diseño y se muestra en la figura siguiente.



6.3 Implementación de la estructura de capas

6.3.1 La capa Cliente

La capa Cliente es realmente externa al sistema ya que a éste se accede mediante el uso de un navegador Web estándar vía protocolo HTTP.

Para el desarrollo se ha tomado como referencia el navegador Microsoft Internet Explorer en su versión 7.0 y también se ha probado con la versión 6.0 de dicho software.

Se hace uso de sencillas rutinas JavaScript para pedir al usuario confirmación de las acciones irreversibles (bajas) y mostrar algunos mensajes de advertencia. Por lo tanto, para un correcto funcionamiento de la aplicación, el navegador utilizado debe tener habilitado el uso de JavaScript.

6.3.2 La capa Web

La capa Web está constituida por páginas JSP y clases de tipo Action y ActionForm de Struts.

Las páginas JSP constituyen la parte Vista del patrón MVC, mientras que las clases Struts Action representan la parte Controlador del patrón MVC correspondiente a la capa Web. Las clases ActionForm permiten la transferencia de datos entre las clases Action y las páginas JSP.

El acceso desde esta capa a los recursos de la capa de Negocio se realiza a través del interfaz INegocioSGAPA proporcionado por esta última. Las clases Struts Action utilizarán esta interfaz para acceder al Modelo y a la lógica del Negocio delegando la acción a clases controladoras del nivel inferior.

Para facilitar el proceso de desarrollo, hacer más legibles las clases controladoras y facilitar su mantenimiento, se ha introducido una clase SGAPAAction de la que heredan todas ellas.

La clase SGAPAAction incluye los métodos necesarios para transferir información entre las clases Bean de la capa de Negocio y las ActionForm de la capa Web. También proporciona métodos para la validación y manipulación de fechas, así como de un punto de acceso único a los servicios proporcionados por la capa de Negocio.

6.3.3 La capa de Negocio

La capa de Negocio encapsula los datos y la lógica de negocio del Modelo de SGAPA mediante clases Java estándar.

Esta capa se ha codificado forma que, si fuera necesario, pueda ejecutarse sobre una Máquina Virtual Java estándar independiente de cualquier tipo de servidor Web, con el objeto de que pueda formar parte de un eventual sistema cliente servidor puro o mixto cliente / servidor / Web.

6.3.3.1 Acceso a los servicios de la capa de Negocio

Vista desde la capa Web, la capa de Negocio proporciona un conjunto de servicios a los que se accede a través del interfaz INegocioSGAPA en aplicación del patrón de diseño [SERVICE LAYER](#). Este interfaz está implementado por la clase [FACHADA](#) de la capa de negocio ServicioNegocioSGAPA.

La clase FACHADA proporciona un punto de acceso único a los servicios de la capa de negocio del sistema SGAPA y encamina las llamadas a las clases controladoras competentes en cada caso. La clase se declara con visibilidad de paquete, de forma que su método constructor solamente pueda invocarse desde clases pertenecientes a su paquete. Esto fuerza a las clases clientes a utilizar exclusivamente la clase implementadora del interfaz INegocioSGAPA. Las clases de la capa Web clientes de INegocioSGAPA pueden obtener una referencia a dicha clase implementadora invocando el método estático `getInterface()` de la clase auxiliar ServiciosSGAPA.

Mediante el mecanismo descrito se consigue la total separación entre la definición del interfaz y su implementación según el patrón [BRIDGE](#).

Para ocultar esta relativa complejidad, la clase SGAPAAction, de la que heredan todas las sub-clases clientes del interfaz en aplicación del patrón [LAYER SUPERTIPE](#), incorpora un atributo protegido que se inicializa en el momento de su construcción con una referencia a la clase FACHADA.

En el [Anexo H](#) se incluye un diagrama de secuencia en el que muestra la dinámica descrita.

6.3.3.2 Implementación de los servicios de la capa de Negocio

La clase ServicioNegocioSGAPA implementa el patrón [SINGLETON](#), de forma que solamente exista una instancia de la misma en el ámbito de una Máquina Virtual Java determinada. Consecuentemente, solamente existirá una instancia de las clases controladoras referenciadas por ServicioNegocioSGAPA. Dado que se prevé que esta clase se utilizará en el contexto de un único servidor de aplicaciones Tomcat, las instancias serán únicas para toda la aplicación.

La clase ServicioNegocioSGAPA representa una FACHADA que se limita a encaminar las peticiones a las clases controladoras del nivel de Negocio adecuadas.

En la versión 1.0 de SGAPA existen únicamente tres clases controladoras que corresponden, respectivamente a los subsistemas de Gestión de Peticiones y de Generación de Informes. Estas clases son:

- ControladorGestPet
- ControladorInformes
- ControladorAdministrador

En eventuales sucesivas versiones, la clase ServicioNegocioSGAPA recogerá, en su caso, los servicios proporcionados por otros subsistemas de la aplicación.

6.3.4 La capa de Persistencia

La capa de Persistencia es responsable de proporcionar a la capa de Negocio los servicios de persistencia para las clases de dicha capa que lo precisen. Está constituida por clases Java construidas según el patrón [DAO](#) (Data Access Object).

Se dispone de una clase DAO por cada tipo de entidad del modelo de Negocio en aplicación de los patrones [DATA MAPPER](#) y [CLASS TABLE INHERITANCE](#). En la versión 1.0 de SGAPA estos son:

Para el subsistema de Gestión de Peticiones:

- ProyectoDAO
- PeticionDAO
- HitoDAO
- TipoHitoDAO
- TipoProductoDAO

Para el subsistema de Generación de Informes:

- EntregasDAO

Para el subsistema de Administración

- TipoHitoDAO
- TipoProductoDAO
- TrabajadorDAO

En la presente versión 1.0 de SGAPA, clases DAO interaccionan con la base de datos relacional mediante la utilización de las funciones proporcionadas por JDBC. En eventuales futuras versiones sería posible utilizar un marco del tipo de iBatis o Hibernate sustituyendo las versiones actuales de las clases DAO por otras basadas en la utilización de dichos marcos, sin afectar a la capa de Negocio.

6.3.4.1 Acceso a los servicios de la capa de Persistencia

Vista desde la capa de Negocio, la capa de Persistencia proporciona un conjunto de servicios a los que se accede a través del interfaz IPersistenciaSGAPA. Este interfaz está definida dentro de la capa de Negocio, pero se implementa por la clase FACHADA de la capa de persistencia ServicioPersistenciaSGAPA, en aplicación del patrón [SEPARATED INTERFACE](#).

El mecanismo FACHADA / BRIDGE utilizado, que permite un alto grado de independencia entre ambas capas, es paralelo al ya comentado para la relación entre las capas Web y de Negocio y constituye otro ejemplo de uso del patrón SERVICE LAYER.

6.3.4.2 Agrupación de conexiones

Las clases DAO hacen uso de una agrupación de conexiones (connection pool) de forma que no sea necesario crear y cerrar una conexión con la base de datos en cada transacción.

Se utiliza un servlet específico, DBInitServlet, para la inicialización de la conexión a la base de datos. Este servlet se ejecuta en el arranque de la aplicación, lo que se indica en el archivo de despliegue web.xml.

El servlet de inicialización de la conexión a la base de datos define e inicializa una agrupación (pool) de conexiones, ConnectionPool, que será utilizada por todos los procesos que necesiten acceder a la base de datos. La clase que implementa la agrupación de conexiones se realiza

mediante el patrón SINGLETON para facilitar su localización y uso desde cualquier parte de la aplicación que lo precise.

Las clases ConnectionPool y DBInitServlet se han tomado directamente del texto del libro [Desarrollo Web](#) con JSP de WROX / Anaya Multimedia.

No se ha utilizado la posibilidad alternativa de usar una fuente de datos proporcionada por el contenedor Tomcat. Esta decisión de diseño se justifica por el objetivo de que las capas de Negocio y Persistencia no sean dependientes de un servidor de aplicaciones y puedan, en su caso, ejecutarse de forma independiente sobre una Máquina Virtual Java estándar en una eventual arquitectura cliente / servidor pura o mixta cliente / servidor / Web.

6.4 Implementación de los subsistemas

6.4.1 Subsistema de registro y menú inicial

El subsistema de registro y menú principal se implementa totalmente en la capa Web de la aplicación. Consta de varias páginas JSP y de las clases derivadas de Action (Struts) denominadas PrincipalAction y DesconectarAction.

La validación de las credenciales de los usuarios se delega al servidor de aplicaciones Tomcat mediante el uso de la directiva <login-config> del archivo "web.xml" de la aplicación dentro de la cual se elige el método de autorización basado en formularios.

Todas las acciones de la aplicación, excluyendo la correspondiente a la presentación de la página de bienvenida se declaran como protegidas en "web.xml" incluyéndolas en las adecuadas declaraciones de <security-constrain> lo que fuerza la acción de login una vez que el usuario invoca cualquiera de ellas, incluyendo la página principal de la aplicación.

6.4.1.1 Perfiles y Roles

Se definen tres perfiles de usuario: trabajador, responsable y administrador. Por su parte, para poder utilizar los mecanismos de validación de usuarios proporcionados por la parcela de seguridad de Tomcat, se definen tres roles con las mismas denominaciones. La relación que se establece entre perfiles de usuario y roles es la siguiente:

Perfil de usuario	Roles
Trabajador	trabajador
Responsable	trabajador responsable
Administrador	trabajador responsable administrador

El menú principal de la aplicación se configura en función del rol o roles desempeñados por el usuario mostrando exclusivamente aquellas funciones que deben ser accesibles a cada tipo de usuario.

La página JSP del menú principal utiliza las etiquetas "logic" de Struts para autoconfigurarse en función de los roles de los usuarios. Este mismo principio y mecanismo se utiliza dentro de las páginas JSP de los demás subsistemas para configurar las opciones de menú de los mismos.

Los roles se utilizan asimismo para el control por parte del contenedor Tomcat del acceso a las diversas acciones de la capa Web. Este control se configura mediante el uso de directivas <auth-constrain> en el archivo "web.xml" de la aplicación.

6.4.1.2 Parcela de seguridad

La gestión de los usuarios y de sus roles utiliza una parcela de seguridad del contenedor Tomcat de tipo JDBC. Su configuración se realiza mediante la inclusión de un elemento <Realm> en el archivo de configuración correspondiente de Tomcat como se detalla en el documento *SGAPA-IDE-01 – Informe de entrega*.

6.4.2 Subsistema de gestión de peticiones

La versión 1.0 de SGAPA, conforme a los objetivos técnicos del proyecto, implementa la totalidad del subsistema de Gestión de Peticiones

Los métodos correspondientes a este subsistema que se definen en el interfaz INegocioSGAPA se implementan en la clase ControladorGestPet. Esta clase es la que se ocupa de crear, actualizar y eliminar, con el concurso de la capa de Persistencia, los objetos de Negocio necesarios, como respuesta a las peticiones de las clases controladoras Action de la capa Web.

Dada la sencillez de la lógica del negocio asociada a las transacciones que gestiona el subsistema de gestión de peticiones, se ha optado por organizar dicha lógica según el patrón [Transaction Script](#).

Este patrón, que se adapta perfectamente al modo de trabajo de Struts, consiste en esencia en disponer de un conjunto de procedimientos cada uno de los cuales toma los datos proporcionados por el usuario, los valida, realiza con ellos los cálculos pertinentes, almacena los resultados en el sistema de persistencia e informa al usuario de la conclusión de la acción.

En el caso de SGAPA, cada procedimiento se asocia a una clase de tipo Action de Struts que, a su vez, delega las actividades que dependen de la lógica de negocio a la rutina correspondiente de la clase ControladorGestPet.

6.4.3 Subsistema de gestión de actividades

Conforme a los objetivos técnicos del TFC del autor, el desarrollo del subsistema de gestión de actividades no forma parte del ámbito del proyecto.

Está en el ánimo del autor intentar obtener de su empresa financiación, en la forma de tiempo libre entre proyectos, para el eventual desarrollo de esta parte del sistema SGAPA.

6.4.4 Subsistema de generación de informes

La versión 1.0 de SGAPA implementa la parte del subsistema de Generación de Informes correspondiente al Informe Mensual de Entregas.

Aunque en la versión inicial de SGAPA solamente se emitirán informes de dicho tipo, la arquitectura del subsistema de generación de informes se ha diseñado e implementado de forma que el tratamiento de los diversos tipos de informes sea uniforme.

6.4.4.1 Factoría de informes

Dado que los objetos que representan a los informes son estructuralmente complejos, su creación se confía a un Método Factoría (patrón [FACTORY METHOD](#)). Siguiendo este patrón, se define un método `getNuevoInforme()` en el interfaz `IFactorialInformes` que será el encargado de construir un nuevo informe. La construcción de un tipo concreto de informe depende, en cada caso, de la clase implementadora del interfaz que se utilice.

En el caso del Informe Mensual de Entregas la clase que implementa el método factoría es `FactorialInformeEntregas`. Esta clase posee todo el conocimiento necesario para construir un objeto de tipo `InformeEntregas`.

Para hacer más claro el uso del patrón y facilitar la futura ampliación del sistema, se han implementado también las clases `FactorialInformeActividad` y `FactorialInformeEntregas`, pero sus métodos factoría devuelven actualmente un valor nulo.

6.4.4.2 Estructura de los informes

En el conjunto del sistema SGAPA se considera que un informe consta, en un primer nivel, de una sección de cabecera, una sección de cuerpo y una sección de pie. Cada una de estas secciones puede, a su vez, contener subsecciones.

Las clases que representen a un informe SGAPA, de cualquier tipo, deberán implementar el interfaz `IInforme`. Las secciones de los informes deberán implementar, a su vez, el interfaz `ISeccionInforme`.

Estas dependencias se definen con el objeto de, por una parte, uniformizar la creación de las clases concretas por parte de los métodos factoría. Por otra, como se verá más adelante, configuran para los informes una variante no canónica del patrón [COMPOSITE](#) que posibilita el uso de otros patrones en su tratamiento.

6.4.4.3 Asignación de contenidos mediante Cadena de Responsabilidad

La asignación de contenidos a las secciones y subsecciones de un informe es una tarea compleja que exige, en principio, conocer detalladamente la estructura del mismo. Para soslayar esta complejidad se recurre al uso del patrón [CADENA DE RESPONSABILIDAD](#).

En aplicación del mismo, la clase controladora encargada de la creación del informe obtiene, con el concurso de los servicios de la capa de Persistencia, el conjunto de datos correspondientes al informe. En el caso del Informe Mensual de Entregas este conjunto es una lista de objetos de tipo `Entrega`.

La clase cliente invoca el método `addEntrega()` del objeto de Negocio `InformeEntregas` iniciando de este modo la Cadena de Responsabilidad. En cada paso de la Cadena, los objetos participantes toman del objeto `Entrega` la información que les compete y la pasan a los objetos que dependen de ellos.

6.4.4.4 Uso de XML en la representación de los informes

A la hora de generar una representación del informe que pueda visualizarse en un navegador, se ha optado por la generación de una vista intermedia en formato XML y en su transformación posterior a HTML mediante el uso de una transformación XSLT, en aplicación del patrón [TRANSFORM VIEW](#).

De esta forma se obtienen varias ventajas:

- La vista intermedia en formato XML puede descargarse a un archivo local del usuario de forma que pueda ser utilizada por otras aplicaciones externas al sistema SGAPA.
- El formato XML es idóneo para representar la estructura de objetos compuestos y complejos como son los informes de SGAPA.
- El uso de XML / XSLT permite separar los datos de su representación de forma clara y eficaz.

6.4.4.5 Obtención de una vista XML mediante el patrón Visitante

Para la obtención de la vista XML de la estructura y contenido de los informes se ha recurrido al uso del patrón [VISITOR](#).

En el caso del Informe Mensual de Entregas la clase visitadora es la denominada `Generador-VistaXML`. Cada objeto componente de un informe posee, siguiendo el patrón, un método `accept()` que permite al visitante acceder a su contenido y generar el fragmento correspondiente de la vista XML.

Dado que el formato XML obliga a la existencia de etiquetas de cierre, ha sido necesario modificar ligeramente el patrón visitante para que cada nodo del `COMPOSITE` se visite dos veces.

6.4.4.6 Servlet de descarga a fichero local

La descarga a fichero local se encarga al `Servlet DownloadServlet` dedicado a esta tarea.

Es de destacar que los navegadores de Microsoft presentan un comportamiento no estándar en lo que respecta a la descarga de archivos, lo que da lugar a múltiples problemas de compatibilidad.

`DownloadServlet` se ha probado con las versiones 6.0 y 7.0 de MS Internet Explorer. No es posible garantizar, sin realizar las pruebas pertinentes, su correcto funcionamiento con otros navegadores o versiones.

En la [bibliografía](#) se incluyen algunos foros en los que se puede encontrar información relativa a esta problemática.

6.4.5 Subsistema de administración

La versión 1.0 de SGAPA implementa la parte del subsistema de administración necesaria para el soporte de los restantes subsistemas desarrollados. Los métodos correspondientes a este subsistema que se definen en el interfaz INegocioSGAPA se implementan en la clase ControladorAdministración.

6.4.5.1 Mantenimiento de entidades

La clase ControladorAdministración se ocupa de crear, actualizar y eliminar, con el concurso de la capa de Persistencia, los objetos Tipo de Hito y Tipo de Producto, como respuesta a las peticiones de las clases controladoras Action de la capa Web.

Al igual que en el caso de los demás subsistemas desarrollados, se ha optado por organizar la lógica de negocio según el patrón TRANSACTION SCRIPT. Cada procedimiento se asocia a una clase de tipo Action de Struts que, a su vez, delega las actividades que dependen de la lógica de negocio a la rutina correspondiente de la clase ControladorGestPet.

6.4.5.2 Gestión de Bloqueos

La clase ControladorAdministración proporciona también métodos para obtener una relación de los recursos bloqueados e un instante dado y para forzar su liberación.

Para obtener mayor información, puede consultarse el epígrafe de este documento correspondiente al mecanismo de [gestión de bloqueos](#).

6.4.6 Subsistema de gestión de la persistencia

La versión 1.0 de SGAPA implementa la parte del subsistema de gestión de la persistencia necesaria para el soporte de los restantes subsistemas desarrollados.

Para obtener mayor información, puede consultarse el epígrafe de este documento correspondiente a la [capa de Persistencia](#).

6.4.7 Gestión de bloqueos

6.4.7.1 Descripción del mecanismo de bloqueos

El mecanismo de gestión de bloqueos utilizado en la versión 1.0 de SGAPA se basa en el uso de un Gestor de Bloqueos implementado en la clase denominada GestorDeBloqueos. Esta clase Proporciona métodos para obtener y liberar bloqueos de recursos que se identifican mediante una clave y se asignan a un propietario. Cada bloqueo (lock) puede considerarse como un semáforo binario cuya existencia indica que el recurso nominado está reservado para su propietario designado.

La clase mantiene un registro de objetos de tipo DatosDeBloqueo en un mapa de memoria y está implementada siguiendo el patrón de diseño SINGLETON, de forma que exista una única instancia de la misma. Todos los métodos públicos de la clase se declaran como 'synchronized' para asegurar la integridad del mapa de bloqueos frente al acceso concurrente al mismo. Esta clase puede considerarse un ejemplo de uso del patrón [REGISTRY](#).

6.4.7.2 Uso del mecanismo de bloqueos

Las acciones susceptibles de modificar datos preexistentes, tales como las de edición y eliminación de entidades, deben obtener previamente un bloqueo del recurso correspondiente mediante la invocación del método `obtenerBloqueo()`. En la llamada se identifica el tipo e identificador del recurso a bloquear y el identificador del usuario en cuyo nombre se realiza la acción.

Si la llamada tiene éxito se puede proceder a la acción modificadora de datos. En caso contrario se debe cancelar la acción.

Terminada la acción modificadora de datos, está debe invocar el método adecuado de `GestorDeBloqueos` para la liberación del recurso.

Un recurso puede ser tanto un objeto como un conjunto de ellos en aplicación del patrón [COARSE GRAINED LOCK](#).

6.4.7.3 Cancelación forzada de los bloqueos

Si por cualquier causa (caída del sistema cliente, ausencia prolongada del usuario, error de programación, etc.) no llegara a ejecutarse la invocación al método de liberación del recurso, este se quedaría bloqueado hasta que se reiniciara la aplicación. Por este motivo en el sistema SGAPA se realiza una cancelación forzada de los bloqueos del usuario conectado en las circunstancias siguientes:

- Cuando el usuario se reconecta.
- Cuando el usuario se desconecta.
- Cuando el usuario visita la página principal del sistema.
- Cuando el usuario invoca las acciones de visualización de Proyectos, Peticiones o Hitos.
- Cuando el contenedor invalida la sesión del usuario al exceder el límite de tiempo de inactividad (time-out)
- Liberación forzada por el Administrador.

En general, estas cancelaciones forzadas se implementan insertando llamadas al método `liberarRecursosDelPropietario()` en los lugares apropiados de la aplicación. La cancelación en el caso de invalidación de la sesión se realiza en el método `valueUnbound()` de la clase `UsuarioConectado`.

A este efecto, cuando un usuario se conecta, el sistema crea una instancia de la clase `UsuarioConectado`, que implementa el interfaz `HttpSessionBindingListener`, y la añade al objeto sesión. Cuando Tomcat invalida la sesión, invoca los métodos `valueUnbound()` de todos los objetos asociados a la misma que implementen el mencionado interfaz, causando la ejecución del procedimiento de desbloqueo forzado de los recursos que eventualmente pudiera tener bloqueados el usuario.

6.4.7.4 Limitaciones del mecanismo de bloqueos

El mecanismo de gestión de accesos descrito es muy eficaz y, correctamente utilizado, muy seguro. No obstante, presenta una limitación que debe tenerse en cuenta.

Para que el mecanismo funcione correctamente es absolutamente imprescindible asegurar que exista una y solo una instancia de GestorDeBloqueos, en realidad de su mapa de bloqueos, en toda la aplicación. En principio, esto se consigue mediante el uso del patrón SINGLETON. Sin embargo, hay que tener en cuenta que la especificación de Java solamente permite asegurar que habrá una única instancia de GestorDeBloqueos en una Máquina Virtual Java (JVM) dada.

Si, como se prevé para SGAPA, la aplicación se ejecutará sobre un único servidor de aplicaciones Tomcat, la condición anterior será equivalente a decir que habrá una única instancia del mapa de bloqueos para toda la aplicación. En cambio, si se utiliza un sistema de múltiples servidores de aplicaciones (IBM WebSphere, BEA WebLogic, etc.) para el balanceo de cargas, nos encontraremos con un mapa de bloqueos en cada servidor. Dado que sincronizar estos diversos mapas es una tarea compleja, sería más adecuado utilizar, en estos casos, una versión de LockManager que sustituya el mapa en memoria por una tabla de la base de datos sin modificar su interfaz.

7 Instalación y uso del sistema

7.1 Inventario de la entrega

La entrega correspondiente a la versión 1.0 para producción del sistema SGAPA consta de los elementos siguientes:

- Informe de entrega (SGAPA-IDE-01).
- Archivo WAR de la aplicación.
- Archivo de comandos SQL para la creación de las tablas de la base de datos.
- Archivo de comandos SQL para la asignación de valores iniciales y de pruebas a las tablas de la base de datos.
- Archivo para la configuración de la parcela de seguridad SQL de Tomcat.
- Guía básica de uso (SGAPA-GDU-01).
- Documentación técnica del proyecto (SGAPA-PDP-01, SGAPA-EAR-01, SGAPA-DDS-01 y SGAPA-PDS-01)

7.2 Instalación y configuración del sistema

Las instrucciones detalladas de instalación y configuración se encuentran en el documento *SGAPA-IDE-01 – Informe de entrega*. Se reseñan a continuación algunos puntos de especial interés.

7.2.1 Entorno de ejecución

7.2.1.1 Configuración del equipo servidor

La versión 1.0 del sistema SGAPA se ha instalado y probado en las configuraciones de servidor siguientes:

Configuración 1

- Sistema operativo MS Windows XP SP2
- Servidor de aplicaciones Tomcat 4.1.24 con Java Virtual Machine 1.4.2
- MySQL 5.0.37 Community Edition con conector JDBC para MySQL versión 5.0.5

Configuración 2

- Sistema operativo MS Windows 2000 SP4
- Servidor de aplicaciones Tomcat 5.5.23 con Java Virtual Machine 1.6.0
- MySQL 5.0.41 Community Edition con conector JDBC para MySQL versión 5.0.5

Se recomienda el uso de una de las dos configuraciones de servidor probadas para la instalación de la versión 1.0 del sistema SGAPA.

7.2.1.2 Configuración de las estaciones de trabajo

La versión 1.0 del sistema SGAPA se ha probado en las configuraciones de estación de trabajo siguientes:

Configuración 1

- Sistema operativo MS Windows XP SP2
- Explorador Web MS Internet Explorer 7.0

Configuración 2

- Sistema operativo MS Windows 2000 SP4
- Explorador Web MS Internet Explorer 6.0

Se recomienda el uso de una de las dos configuraciones de estación de trabajo probadas para el uso de la versión 1.0 del sistema SGAPA, especialmente en lo que respecta al explorador Web.

7.2.1.3 Preparación del entorno de ejecución (servidor)

7.2.1.3.1 Obtención de los elementos

Las versiones necesarias de los diversos elementos, según la configuración de ejecución elegida, pueden descargarse de las URL siguientes:

- **Apache Tomcat:** <http://tomcat.apache.org/whichversion.html>
- **Java Virtual Machine:** <http://java.sun.com/>
- **MySQL y conector:** <http://dev.mysql.com/downloads/connector/j/5.0.html>

7.2.1.3.2 Instalación y configuración de los elementos

Todas las aplicaciones disponen de un instalador para Windows. Se recomienda ejecutar los instaladores aceptando todas las opciones por defecto.

Para el correcto funcionamiento de SGAPA es necesario que el usuario "sgapa_user" posea plenos derechos sobre el esquema de base de datos "sgapa". En consecuencia, si la configuración de MySQL utilizada exige la validación de usuarios, deberá crearse el usuario "sgapa_user" con clave "sgapa_user" y otorgarle todos los derechos sobre el esquema de base de datos "sgapa".

7.2.2 Aplicación SGAPA

7.2.2.1 Despliegue de la aplicación sobre Tomcat

Para desplegar SGAPA sobre el servidor Tomcat se recomienda utilizar la aplicación de utilidad Tomcat Manager, que se instala por defecto con dicho software, para desplegar el archivo WAR en el que se entrega la aplicación.

7.2.2.2 Configuración de la parcela de seguridad JDBC

Como se ha descrito con anterioridad, el sistema SGAPA hace uso de una parcela de seguridad del contenedor Tomcat para el control del acceso de usuarios. La configuración de la parcela de seguridad, de tipo JDBC, se realiza mediante la inclusión de un elemento <Realm> en el archivo de configuración server.xml de Tomcat.

El elemento <Realm> deberá situarse dentro del elemento <Context> correspondiente a la aplicación SGAPA. Un lugar adecuado para situar el elemento <Context> dentro del archivo Server.xml es inmediatamente antes del tag </Host>. Con la entrega se proporciona el archivo "server_context_realm_sgapa.txt" con el contenido necesario.

Para el correcto funcionamiento de la parcela de seguridad es necesario que una copia del conector JDBC para MySQL se encuentre en el directorio de Tomcat %CATALINA_HOME%\common\lib

7.2.2.3 Creación de la base de datos

Con la entrega se proporciona el archivo "CrearTablasSGAPA.sql" que contiene los comandos SQL necesarios para la creación de la base de datos "sgapa" y de las tablas que la forman.

En su primera línea se encuentra, comentado, el comando para la creación del usuario "sgapa_user". Si no se ha creado manualmente la cuenta de este usuario durante la configuración de MySQL, deberá "descomentarse" esta línea antes de la ejecución del script.

Una vez ejecutado el script de creación de la base de datos, deberá ejecutarse el que se encuentra contenido en el archivo "InicializarValoresSGAPA.sql" que dotará a las tablas de un contenido inicial adecuado para la realización de las pruebas que se estimen oportunas.

7.3 Guía de uso

Debido a su extensión, la Guía de uso de la versión 1.0 del sistema SGAPA se entrega independientemente en el documento *SGAPA-GDU-01 – Guía de uso*.

8 Conclusiones

8.1 Sobre el Plan de Proyecto

El proyecto se ha desarrollado en la práctica cumpliendo el calendario previsto. Para ello han sido determinantes los factores siguientes:

- El enfoque dirigido a la gestión de los riesgos.
- La adopción de una metodología de desarrollo iterativa e incremental.
- El establecimiento de ciclos de trabajo de duración fija.
- La definición de unos objetivos concretos para cada uno de los ciclos de trabajo.
- Una considerable dedicación al proyecto.
- Una fluida comunicación con el Consultor acordando prioridades y resolviendo dificultades técnicas y organizativas.

Es de destacar que el tiempo que se ha necesitado para la codificación del software ha sido superior al inicialmente estimado. Al comienzo del proyecto, la valoración realizada de la dedicación necesaria para llevarlo a término era de unas 225 horas, un 45% de las mismas dedicadas a la codificación. En su conclusión, el tiempo total empleado en proyecto ha sido aproximadamente de 270 horas de las que un 55% se utilizaron para la codificación, pruebas y resolución de problemas técnicos.

8.2 Sobre la metodología utilizada

Como ya se ha indicado, la [metodología utilizada](#), basada en el [Unified Process](#) de Rational, ha sido un factor decisivo para el desarrollo del proyecto.

Como se indicó en la sección dedicada a los [riesgos del proyecto](#), el desconocimiento inicial de las tecnologías JEE del autor constituía un riesgo tecnológico muy alto que ponía en peligro cierto la consecución de los objetivos del mismo.

Ante esta realidad, se consideró imprescindible abordar el necesario proceso de aprendizaje de las nuevas tecnologías tan pronto como fuera posible y sin esperar a la fase formal de desarrollo.

En esta línea, se planteó el objetivo de disponer de un primer prototipo básico antes del comienzo de la fase formal de diseño. Este prototipo debía implementar, al menos de forma parcial, un Caso de Uso representativo del sistema y utilizar cuantas tecnologías JEE fuera posible de entre las que se preveía usar para el sistema completo.

Aunque la arquitectura del primer prototipo no correspondía a la luego adoptada, su construcción permitió comprobar que se disponía de un entorno de desarrollo utilizable y que se tenía la información necesaria para realizar el proyecto.

La decisión de adelantar el desarrollo del software mediante la adopción de ciclos iterativos y construcción incremental se reveló totalmente acertada ya que, teniendo en cuenta el exceso de tiempo de codificación, pruebas y resolución de problemas sobre el inicialmente previsto,

hubiera sido materialmente imposible completar el proyecto si la codificación hubiera comenzado con posterioridad.

También hay que destacar que el proceso iterativo e incremental no se ha limitado al desarrollo del software, sino que también la documentación de análisis, diseño y la guía de usuario se han visto sometidas al mismo, generándose varias versiones de cada uno de los documentos citados.

8.3 Sobre la consecución de objetivos

8.3.1 Objetivos técnicos

En lo que respecta a los objetivos técnicos planteados al comienzo del trabajo en el Plan de Proyecto es la opinión de este autor que se han cumplido en un alto grado.

El sistema construido implementa los Casos de Uso definidos como necesarios para la Gestión de las Peticiones de Servicio tal y como se especifican en el documento de análisis. Adicionalmente, implementa también aquella funcionalidad de administración del sistema que se ha considerado necesaria para su correcto funcionamiento.

La aplicación supera las pruebas de sistema establecidas, según queda plasmado en el documento correspondiente.

En lo que respecta a los requisitos no funcionales, el sistema entregado los satisface en un grado superior al que suele aceptarse como satisfactorio para una primera versión de producción.

Como toda primera versión, es seguro que la 1.0 para producción de SGAPA contendrá errores que podrán dar lugar a fallos durante las pruebas de aceptación o en su uso cotidiano y que deberán ser eliminados con el oportuno mantenimiento correctivo.

Es de destacar que la documentación técnica de análisis, diseño, instalación y pruebas, así como la guía de uso, se han realizado siguiendo unos estrictos criterios con el objeto de facilitar al máximo tanto las mencionadas tareas de mantenimiento como el futuro desarrollo del sistema.

En un [capítulo posterior](#) se describen las líneas de mejora que el autor considera prioritarias para el desarrollo futuro del sistema SGAPA.

8.3.2 Objetivos académicos

En lo que respecta a la consecución de los [objetivos académicos](#) que me marqué al comienzo del trabajo, hay que destacar que la realización de este TFC me ha proporcionado una inmejorable oportunidad para obtener nuevos y muy valiosos conocimientos en áreas en las que apenas disponía de alguna experiencia.

Como ya he apuntado en otras secciones de esta memoria, aunque en mi vida profesional he mantenido un cierto contacto, que podríamos calificar como periférico, con el mundo Web en general y las tecnologías JEE en particular, nunca había tenido la oportunidad de profundizar en las mismas. Convencido de su importancia para mi formación y futuro profesional, durante

los últimos años he intentado adquirir por cuenta propia estos conocimientos. Pero la relativa complejidad del área y la falta de foco inherente a este tipo de esfuerzos me venían impidiendo alcanzar estas metas.

El inmenso poder de concentración, motivación e impulso que en mi caso ha supuesto la realización de este TFC han marcado la diferencia. Aunque el esfuerzo, en especial las horas de dedicación, ha superado con creces el estimado inicialmente, considero que los resultados son más que satisfactorios y que los conocimientos y experiencia constituyen un sólido cimiento sobre el que construir otros más extensos y avanzados en estas áreas.

En esta línea, quisiera resaltar los resultados alcanzados más significativos desde un punto de vista de adquisición de conocimientos y experiencia.

- Este trabajo me ha permitido profundizar en aspectos que son peculiares del desarrollo de soluciones informáticas para empresas en un entorno Web. Desde el uso de tecnologías básicas como [HTML](#) y [JavaScript](#) que, no obstante su simplicidad, nunca con anterioridad había utilizado personalmente, hasta el manejo de las complejidades de la navegación entre ventanas, propias de las aplicaciones Web, pasando por temas como la validación de usuarios gestionada por el servidor de aplicaciones.

En particular, el desarrollo de SGAPA me ha dado la oportunidad de meditar sobre las similitudes y diferencias entre las aplicaciones cliente / servidor clásicas y las aplicaciones Web y cómo es necesario desarrollar patrones de diseño y codificación adaptados a este paradigma. En este sentido, me ha resultado de gran utilidad el libro de Martin [Fowler](#) referenciado en la bibliografía.

- En lo que respecta a las tecnologías JEE, mi esfuerzo se ha centrado en dos que he considerado esenciales para el proyecto: JSP y Struts.

La realización de la parte Vista del paradigma MVC por medio de JSP supuso un considerable reto al comienzo del proyecto ya que, como ya he comentado, mis conocimientos de HTML eran puramente teóricos. Fue necesario dedicar bastante tiempo al estudio de los ejemplos de interfaces realizados con JSP que pude reunir. En este caso, el libro [Desarrollo Web con JSP](#) proporcionó la mayor parte de la información utilizada.

En el caso de Struts, tanto el mencionado libro Desarrollo Web con JSP como el titulado [Jakarta Struts](#) suministraron los ejemplos y explicaciones necesarios para diseñar el modo de uso de las clases Action necesarias para la parte Controlador de la aplicación y las ActionForm como vehículo de comunicación entre el Controlador y la Vista.

- El desarrollo de SGAPA también me ha aportado una nueva comprensión del paradigma de diseño MVC. Esta forma de estructurar las aplicaciones ya había sido utilizado por el autor en otros contextos tecnológicos desde antiguo. No obstante, la radical separación entre la Vista y el Modelo que propugna el uso de JSP/Struts me ha resultado especialmente atractiva aunque, en ocasiones, complique algo el diseño.

- Desde que en diciembre del 2000 descubrí el libro de "[la banda de los cuatro](#)" quedé cautivado por el concepto de patrón de diseño y me he esforzado en incorporarlo e mis proyectos siempre que he podido.

En SGAPA he tenido la oportunidad de emplear algunos patrones de diseño clásicos como FACADE, BRIDGE, SINGLETON, COMPOSITE y FACTORY METHOD con los que ya había trabajado en el pasado. Pero, sobre todo, me ha proporcionado la ocasión de investigar otros como VISITOR y CHAIN OF RESPONSIBILITY, que han sido de gran utilidad en el diseño y construcción del subsistema de generación de informes y con los que no había tenido oportunidad de trabajar anteriormente, así como con varios de los patrones de alto nivel de [Fowler](#).

- El uso de las transformaciones XSLT para la generación de HTML a partir de contenidos XML es otro área en la que este TFC me ha permitido trabajar e incorporar un conocimiento y experiencia que considero de gran utilidad profesional. La información necesaria sobre las transformadas la encontré en el libro [XML a través de ejemplos](#), que ya había utilizado anteriormente para estudiar el procesamiento de contenidos XML con DOM y XAS.
- Por último, citaré la experiencia adquirida en el uso de una metodología iterativo / incremental con ciclos de trabajo de duración (casi) fija.

Dado que las metodologías de trabajo siempre me había venido impuestas externamente, bien por las empresas para las que he trabajado, bien por sus clientes, nunca había tenido la oportunidad de experimentar con este tipo de métodos. Y he de destacar que la experiencia ha sido muy positiva comparada con la obtenida en el uso de diversas variantes del ciclo tradicional en cascada que son aún los prevalentes en mi entorno profesional.

8.4 Líneas futuras de actuación

8.4.1 Desarrollo posterior del sistema SGAPA

8.4.1.1 Funcionalidad adicional

Dado que la funcionalidad desarrollada en este Trabajo Fin de Carrera es una parte de la funcionalidad total prevista para el sistema SGAPA, una línea natural de desarrollo para el mismo sería la implementación de los subsistemas y funcionalidades diseñadas pero no implementadas en la actualidad.

Como se ha indicado anteriormente, está en el ánimo del autor presentar a su empresa una maqueta / prototipo basada en el presente TFC y solicitar permiso para completar el desarrollo necesario cuando la carga de trabajo lo permita.

8.4.1.2 Mejoras en el control de accesos

Actualmente se permite que, bajo un mismo conjunto de credenciales, un usuario pueda mantener varias sesiones simultáneas lo que puede provocar errores en el control de la concurrencia. Debe, o bien limitarse el acceso a una sesión por usuario, o bien incluir en la información de bloqueo, junto con los identificadores de recurso y usuario, un identificador adicional de sesión.

8.4.1.3 Incorporación de un registro (log) de actividad y errores

Por falta material de tiempo, no se ha provisto a la versión actual del sistema de un registro (log) de actividad y errores.

La experiencia demuestra que esta característica contribuye de forma decisiva a la calidad de las aplicaciones y facilita notablemente su mantenimiento correctivo. En consecuencia, su incorporación sería una tarea de alta prioridad.

8.4.1.4 Reestructuración del código

El código actual de SGAPA, sobre todo en las subclases de Action, presenta una excesiva redundancia que sería muy conveniente reducir.

Dado que las acciones a ejecutar para el alta, baja, visualización y edición de las diversas entidades gestionadas por el sistema es muy similar esto las hace candidatas al empleo del patrón TEMPLATE METHOD. El esqueleto del algoritmo de cada operación podría definirse en una superclase, SGAPAAction, y delegarse a las clases Action derivadas la implementación de los pasos relevantes de los algoritmos.

Por el contrario, considero que los métodos incluidos actualmente en SGAPAAction para transferir datos entre las clases ActionForm y las clases de entidad del Negocio deberían pasarse a la clase ActionForm correspondiente. Esto se ha hecho así ya para las entidades Tipo de Hito, Tipo de Producto y Trabajador, pero no se ha dispuesto de tiempo suficiente para hacerlo con el resto de las entidades.

8.4.1.5 Uso de un marco de persistencia estándar

De cara a facilitar el mantenimiento evolutivo y correctivo por parte de desarrolladores más familiarizados con el paradigma de orientación a objetos que con los sistemas de gestión de bases de datos relacionales, sería quizá deseable sustituir la capa de persistencia actual por otra desarrollada con un marco de persistencia estándar como Hibernate o iBatis.

Gracias al diseño arquitectónico en capas de SGAPA, esta sustitución podría realizarse, sin afectar al sistema global, haciendo que el nuevo servicio de persistencia implementase el interfaz IPersistenciaSGAPA.

8.4.1.6 Actualización de la versión de Struts

En la fase inicial del proyecto, el autor decidió utilizar el entorno de desarrollo WebSphere Application Developer – WSAD – en su versión 5.1 por estar familiarizado con el mismo al haberlo utilizado en otros proyectos.

Esta decisión tuvo el efecto positivo deseado de evitar la necesidad de invertir un tiempo precioso en el aprendizaje del uso de otro entorno. No obstante lo anterior, la elección de WSAD 5.1 supuso también la adopción de una versión de Struts excesivamente anticuada, la 1.0, que es a la que el entorno da soporte.

Las futuras versiones de SGAPA deberán desarrollarse utilizando un entorno de trabajo actualizado (Eclipse, Rational Application Developer, etc) que soporte una versión de Struts como la 1.3.8.

Para ello será necesario realizar en el código actual de SGAPA al menos los cambios siguientes:

- Se deberá sustituir la tecnología Templates utilizada en la codificación de las JSP, ya no soportada, por la tecnología Tiles.
- Se deberán utilizar clases ActionMessage en lugar de ActionErrors por no estar ya soportadas estas últimas.
- Se deberá reescribir el Servlet de inicialización de la agrupación de conexiones a la base de datos o utilizar una agrupación de conexiones proporcionada por Tomcat ya que la clase GenericDataSource utilizada en el mismo ya no está soportada.

8.4.2 Estudios en el área Web y JEE

Las áreas en las que me propongo continuar investigando en un futuro inmediato son las siguientes:

- Utilización de las librerías Common Controls en las JSP. Estas librerías son las utilizadas preferentemente por la organización CLIENTE para la realización de interfaces en aplicaciones JEE.

No he considerado apropiado su introducción en SGAPA debido a que no son de uso público y carezco de las licencias necesarias para su utilización fuera del CLIENTE.

- Diferencias, ventajas e inconvenientes de Struts 2 frente a Struts 1 u otras tecnologías alternativas como Faces.
- Utilización de Enterprise Java Beans y JBoss.
- Servicios Web, SOAP, XML-RCP, etc.

ANEXO A: Glosario de términos

Término	Definición
SGAPA	Sistema de Gestión Administrativa de Peticiones y Actividades. SGAPA es el sistema informático cuyo desarrollo es el objeto del presente proyecto.
CLIENTE	Empresa que contrata los servicios del PROVEEDOR para el desarrollo, soporte y mantenimiento de soluciones informáticas en régimen de externalización.
PROVEEDOR	Empresa a la que el CLIENTE contrata el desarrollo, soporte y mantenimiento de soluciones informáticas en régimen de externalización. La organización del PROVEEDOR será la usuaria del sistema cuyo desarrollo es el objeto del proyecto SGAPA .
Petición	Solicitud formal de trabajos encargada por el CLIENTE al PROVEEDOR dentro del contexto de un proyecto . El objeto de una petición es la producción de uno o más productos que deberán entregarse en unas fechas pactadas (hitos de la petición)
Proyecto	Proyecto de desarrollo, mantenimiento o soporte de una solución informática propiedad del CLIENTE y que este utiliza para la actividad de su negocio. Un proyecto engloba a una o más peticiones .
Servicio	Agrupación de trabajadores del PROVEEDOR cuya misión es prestar servicios de desarrollo, soporte o mantenimiento de soluciones informáticas para el CLIENTE dentro de un área tecnológica o funcional concreta. Un Servicio está sometido a, y definido por, un contrato de externalización entre el CLIENTE y el PROVEEDOR .
Trabajador	Profesional contratado por la organización PROVEEDORA para prestar servicios de desarrollo, soporte o mantenimiento de soluciones informáticas para el CLIENTE . Un trabajador puede estar asignado a uno o más Servicios . Perfil del usuario que puede acceder a las funcionalidades del sistema que son privativas de este perfil.
Hito	Fecha en la que un Servicio debe hacer entrega al CLIENTE de uno o más productos .
Producto	Entregable cuya producción es uno de los objetos de una petición .
Responsable	Trabajador designado por el PROVEEDOR como interlocutor con el CLIENTE para todo lo relacionado con un Servicio determinado. Perfil del usuario que puede acceder a las funcionalidades del sistema que son privativas de este perfil y del perfil de Trabajador .
Actividad	Trabajos realizados por un Trabajador con el objeto de cumplimentar una petición . Una actividad se caracteriza por haber sido realizada en una determinada fecha y haber ocupado un determinado tiempo de dedicación.
Usuario	Trabajador cuya cuenta en el sistema SGAPA se encuentra activa y que, en consecuencia, puede utilizarlo.

Perfil	Característica de los usuarios del sistema SGAPA que les otorga el derecho de acceder a las funcionalidades del mismo que sean privativas de los usuarios que lo posean.
Administrador	Perfil del usuario que puede acceder a las funcionalidades del sistema que son privativas de este perfil y de los perfiles de Responsable y Trabajador .
Recurso	Termino con el que se designa de forma genérica a cualquier entidad gestionada por el sistema SGAPA (Proyecto, Petición, Hito, etc.)
Bloqueo	Acción por la que el sistema otorga a un usuario derechos exclusivos para la modificación de un recurso . Mientras un usuario posea el bloqueo de un recurso, ningún otro usuario del sistema podrá modificarlo.

ANEXO B: Bibliografía

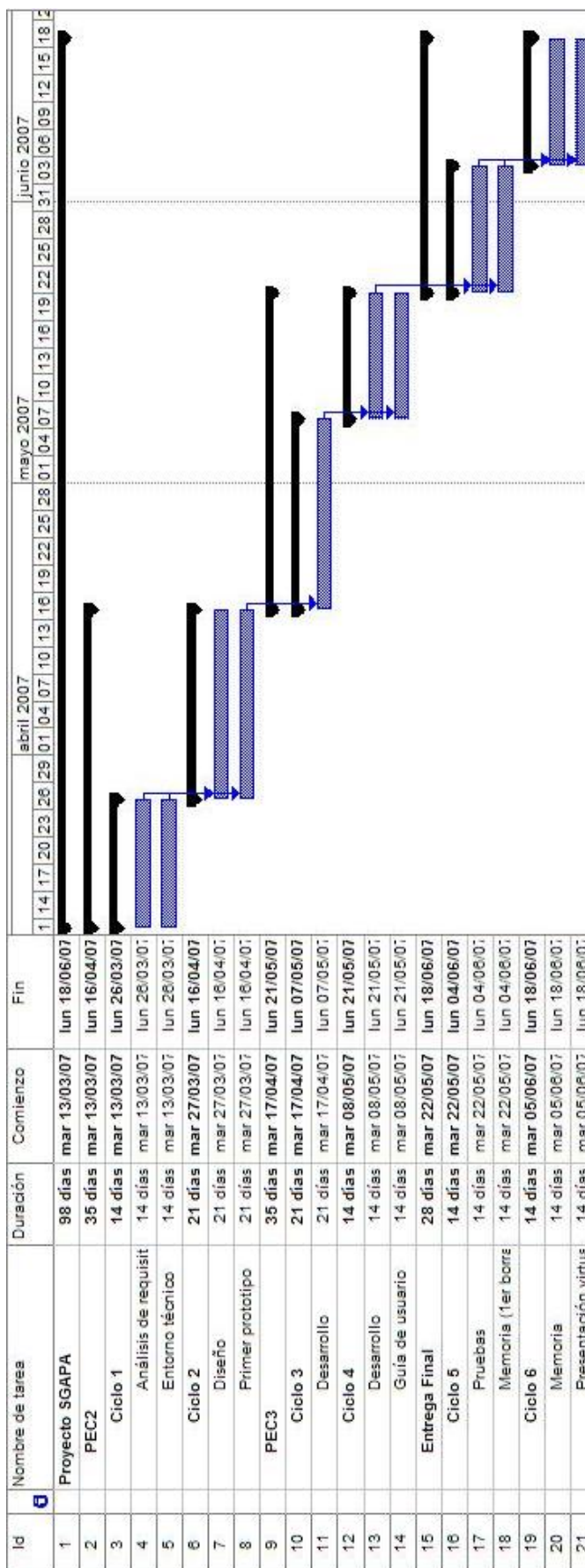
Para el desarrollo del Trabajo Fin de Carrera descrito e esta memoria se ha consultado y utilizado la bibliografía siguiente:

- **Design Patterns – Elements of Reusable Object-Oriented Software.** Erich Gamma, Richard Helm, Ralph Johnson y John Vlissides. Addison Wesley, 1995.
- **El Proceso Unificado de Desarrollo de Software.** Ivar Jacobson, Grady Booch y James Rumbaugh. Addison Wesley, 2000.
- **Fundamentos – Desarrollo Web con JSP.** Jayson Falkner, Ben Galbraith, Roman Irani, Casey Kochmer, Sathya Narayana Panduranga, Krishnaraj Perrumal, John Timney y Meeraj Moidoo Kunnumpurath. WROX / Anaya Multimedia, 2002.
- **HTML Diseño y creación de páginas Web.** Ramón Soria. Ra-Ma, 1997.
- **Ingeniería de software.** Ian Sommerville. Addison Wesley, 2001.
- **Ingeniería de software orientado a objetos.** Bernd Bruegge y Allen H. Dutoit. Prentice Hall, 2002.
- **Jakarta Struts.** Check Cavaness. O'REILLY / Anaya Multimedia, 2005.
- **JavaScript Demistified.** Jim Keogh. Mc Graw Hill / Osborne, 2005.
- **Patterns of Enterprise Application Architecture.** Martin Fowler. Addison-Wesley, 2003.
- **Writing Effective Use Cases.** Alistar Cockburn. Addison Wesley, 2001.
- **XML a través de ejemplos.** Abraham Gutiérrez y Raúl Martínez. Ra-Ma, 2001.

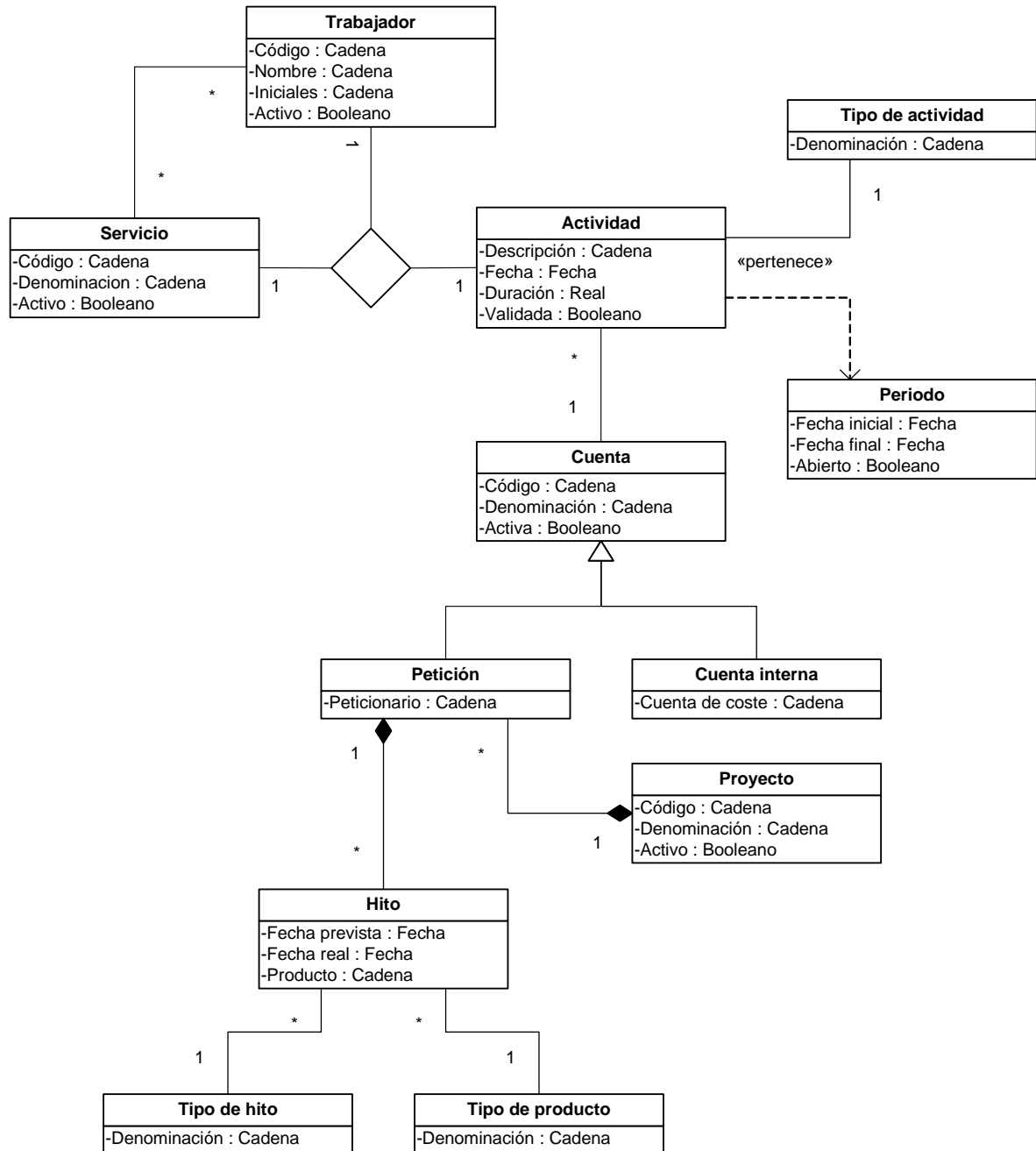
Para obtener una información actualizada, así como ejemplos concretos de uso, se ha recurrido a la búsqueda en Internet resultando de especial utilidad las direcciones siguientes:

- Transformación de XML con XSLT
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=jaxpjsp>
- Seguridad en Tomcat
<http://www.adictosaltrabajo.com/tutoriales/tutoriales.php?pagina=seguridadtomcat>
- Descarga de ficheros
<http://forum.java.sun.com/thread.jspa?threadID=586671&messageID=3023600>
<http://jspwiki.org/wiki/MakingADownloadServlet>
- Vocabulario visual para el diseño de la interacción
<http://www.jjg.net/ia/visvocab/spanish.html>

ANEXO C: Diagrama de Gant del proyecto.

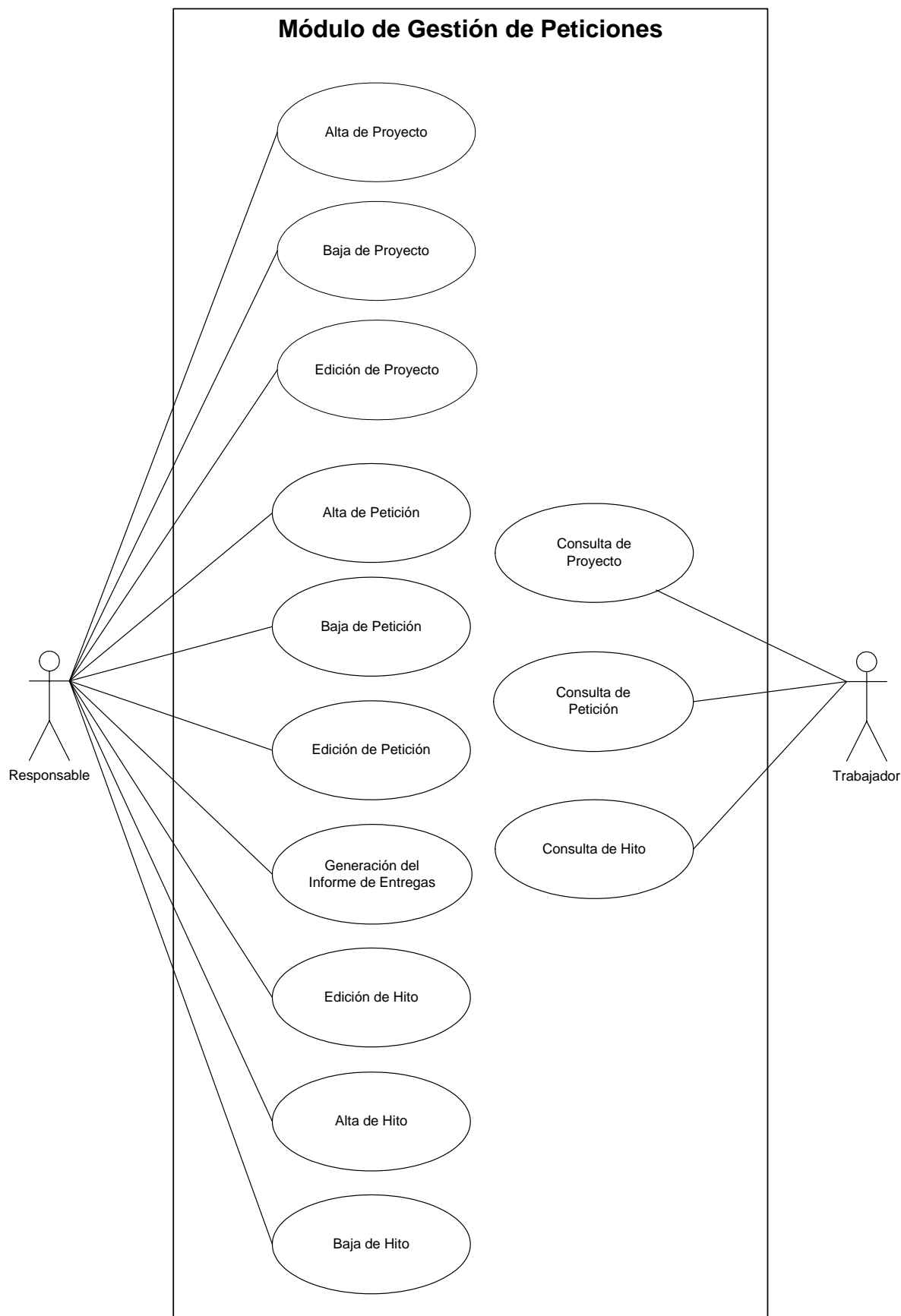


ANEXO D: Diagrama de entidades del dominio

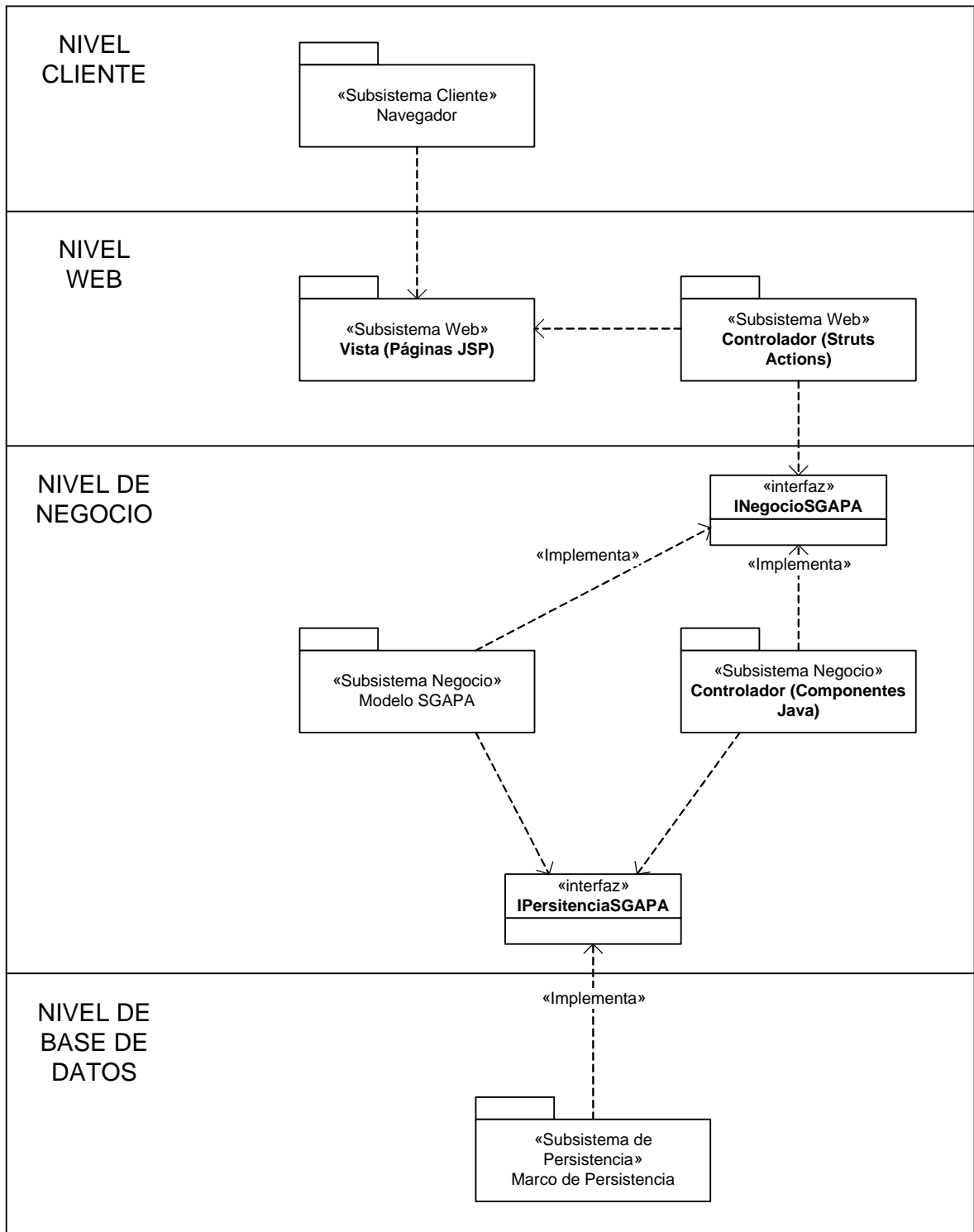


ANEXO E: Diagramas de Casos de Uso

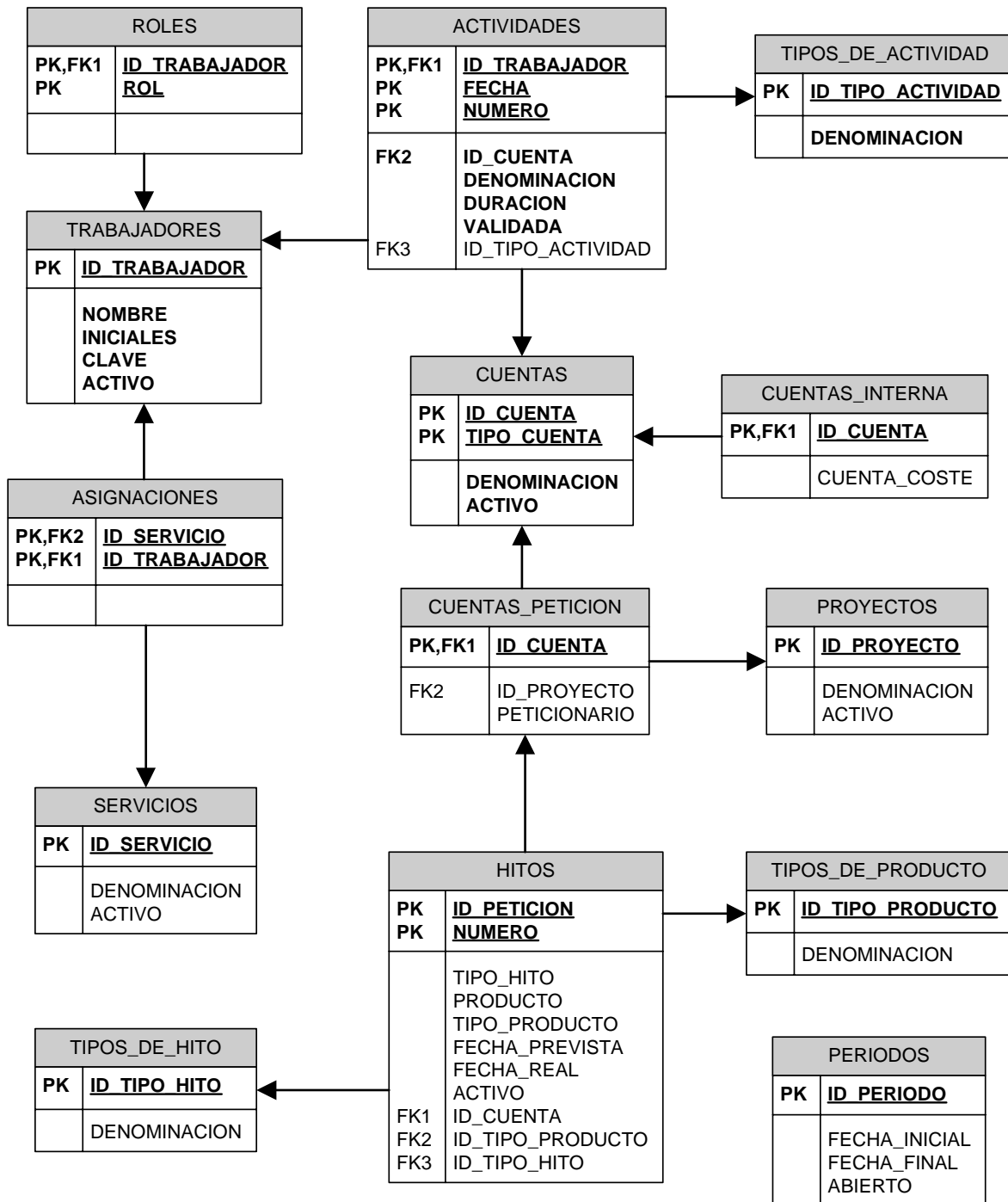




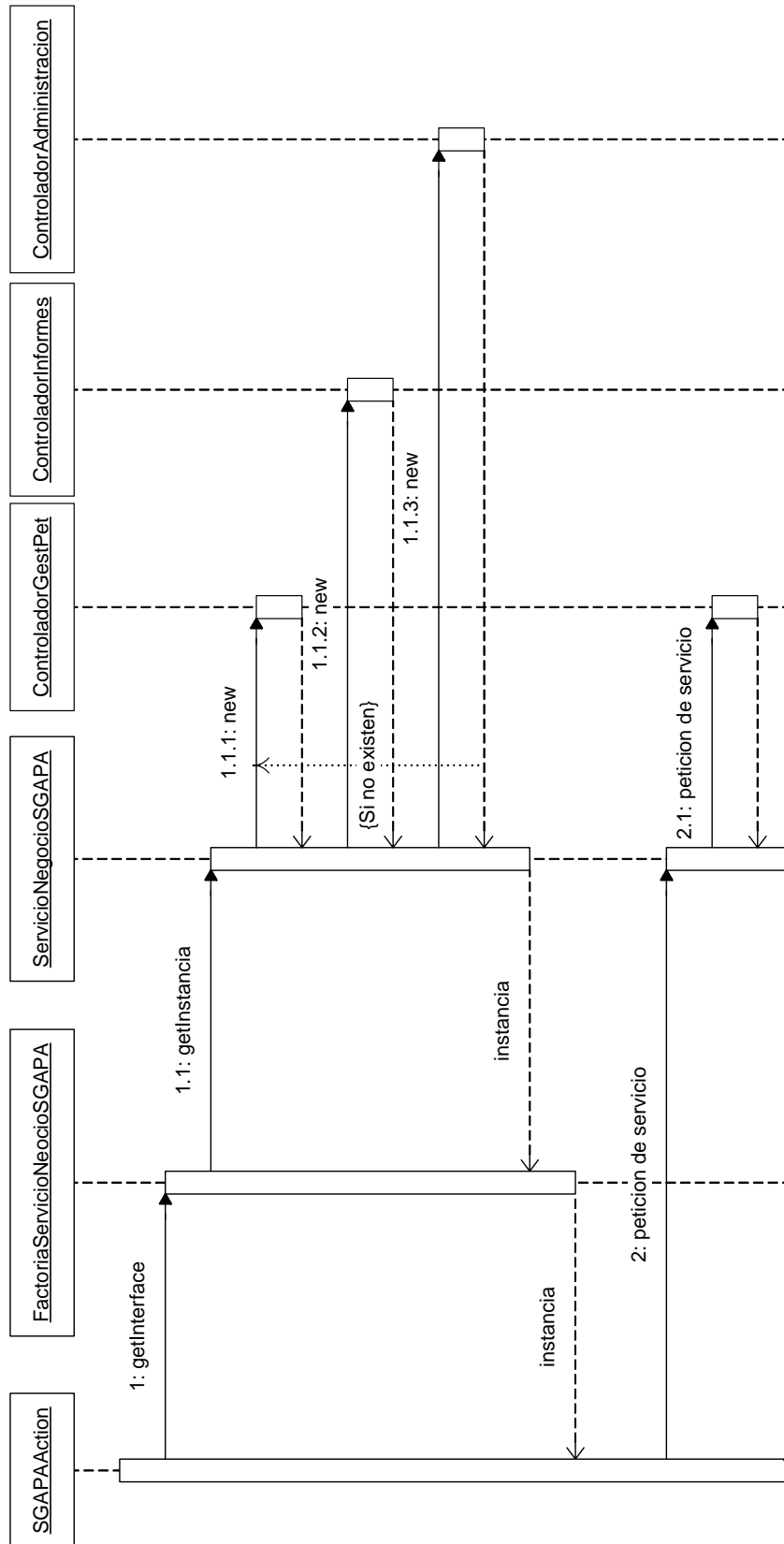
ANEXO F: Esquema de la estructura en capas del sistema



ANEXO G: Esquema de la base de datos



ANEXO H: Secuencia de acceso a los servicios de Negocio





www.uoc.edu

Estudios de Informática y Multimedia y Telecomunicación