



Universidad de Valladolid

E.T.S Ingeniería Informática

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

HALP! Herramienta de Análisis Log

Autor:

Gustavo Velasco Sastre

Tutor:

José Manuel Marqués Corral

Resumen:

Este trabajo aborda aspectos relativos al análisis web para modelar y crear un sistema que permita a un administrador web realizar estudios del rendimiento y el comportamiento del servicio que proporciona una página web. Dichos estudios permitirán tomar decisiones sobre su forma, la estructura de su contenido, los elementos alojados en el servidor que la componen y los parámetros del propio servidor que la aloja. Estas decisiones deberán estar encaminadas a obtener mejoras en la usabilidad, navegabilidad, eficiencia y tiempos de respuesta de la web, para poder proveer a sus usuarios de una mejor experiencia que conlleve grandes beneficios a los prestadores del servicio.

Palabras clave: análisis web, página web, archivo log, análisis log, usabilidad web, navegabilidad, acceso web.

Tabla de contenido

Capítulo 1 - Introducción.....	1
1.1 Descripción.....	2
1.2 Objetivos	2
1.3 Organización de la memoria	3
Capítulo 2 - Plan de Desarrollo	5
2.1 Decisiones iniciales.....	6
2.2 Metodología empleada.....	6
2.3 Plan de Proyecto.....	8
2.3.1 Organización del Proyecto.....	8
2.3.2 Plan de Trabajo.....	8
2.3.3 Plan de Seguimiento y Control	14
2.3.4 Gestión de Riesgos.....	15
2.4 Evaluación del Plan.....	17
Capítulo 3 - Entorno.....	19
3.1 Introducción.....	20
3.2 Herramientas utilizadas.....	20
3.2.1 Configuración de Eclipse	21
3.3 Lenguaje Python	22
3.3.1 Mod_Python.....	22
3.3.2 Bibliotecas.....	23
Capítulo 4 - Análisis Web	25
4.1 Introducción.....	26
4.2 Definición.....	26
4.3 Recopilación de datos.....	26
4.3.1 Origen de los datos.....	27
4.3.2 Fuentes de datos	27
4.4 Los datos.....	30
4.5 Los datos en los archivos log.....	30
4.5.1 Contenido.....	30
4.5.2 Estándares de Formato	31
4.6 Conceptos derivados de los datos.....	33
4.6.1 Proyecto web	34
4.6.2 Log	34
4.6.3 Visita	35
4.6.4 Hit.....	35

4.6.5 Visitante	37
4.6.6 Servidor	37
4.6.7 Página	38
4.7 Presentación de los resultados	38
Capítulo 5 - Requisitos	41
5.1 Visión General	42
5.2 Requisitos funcionales	42
5.3 Requisitos no funcionales	45
5.3.1 Ubicación y Portabilidad.....	45
5.3.2 Facilidad de uso	45
5.3.3 Datos	45
5.3.4 Documentación	45
5.3.5 Restricciones.....	46
Capítulo 6 - Análisis.....	47
6.1 Actores y Roles	48
6.2 Casos de Uso	48
6.2.1 Nuevo Proyecto.....	48
6.2.2 Eliminar Proyecto	49
6.2.3 Seleccionar Proyecto	49
6.2.4 Modificar Proyecto	50
6.2.5 Insertar Archivo Log	50
6.2.6 Eliminar Archivo Log	51
6.2.7 Ver Estadística	51
6.3 Modelo Estático	52
6.4 Modelo Dinámico.....	55
6.4.1 Nuevo Proyecto.....	55
6.4.2 Eliminar Proyecto	56
6.4.3 Seleccionar Proyecto	56
6.4.4 Modificar Proyecto	57
6.4.5 Ver Estadística	57
6.4.6 Insertar Archivo Log	58
6.4.7 Eliminar Archivo Log	59
Capítulo 7 - Diseño	61
7.1 Arquitectura del sistema	62
7.2 Diseño Estático	63
7.2.1 Paquete Presentación.....	63
7.2.2 Paquete Lógica de Negocio.....	85

7.2.3 Paquete Persistencia	104
7.3 Diseño Dinámico	110
Capítulo 8 - Pruebas	113
8.1 Introducción	114
8.2 Pruebas y Resultados	114
8.2.1 Nuevo Proyecto	115
8.2.2 Eliminar Proyecto	115
8.2.3 Seleccionar Proyecto	115
8.2.4 Modificar Proyecto	116
8.2.5 Insertar Archivo Log	116
8.2.6 Eliminar Archivo Log	117
8.2.7 Ver Estadística	118
Capítulo 9 - Conclusiones y Extensiones	123
9.1 Introducción	124
9.2 Conclusiones	124
9.3 Futuras líneas de trabajo	125
9.3.1 Entrada de Datos	125
9.3.2 Salida de Información	125
Apéndice A - Contenido del CD	127
A.1 Introducción	128
A.2 Documentación	128
A.3 Software	128
Apéndice B - Bibliografía	131

Tabla de Ilustraciones

Ilustración 2-1 Flujos de trabajo por fase	7
Ilustración 6-1 Casos de Uso	48
Ilustración 6-2 Diagrama Entidad-Relación.....	53
Ilustración 6-3 Diagrama de clases de análisis	54
Ilustración 6-4 Diagrama de secuencia de análisis Nuevo Proyecto	55
Ilustración 6-5 Diagrama de secuencia de análisis Eliminar Proyecto.....	56
Ilustración 6-6 Diagrama de secuencia de análisis Seleccionar Proyecto.....	56
Ilustración 6-7 Diagrama de secuencia de análisis Modificar Proyecto.....	57
Ilustración 6-8 Diagrama de secuencia de análisis Ver Estadística	57
Ilustración 6-9 Diagrama de secuencia de análisis Insertar Archivo Log	58
Ilustración 6-10 Diagrama de secuencia de análisis Eliminar Archivo Log.....	59
Ilustración 7-1 Arquitectura del sistema	62
Ilustración 7-2 Paquete Vista Web - Gestión	64
Ilustración 7-3 Paquete Vista Web - Estadísticas	65
Ilustración 7-4 Paquete Modelo Entrada	66
Ilustración 7-5 Paquete Formularios Gestión	70
Ilustración 7-6 Paquete Formularios Estadísticas	75
Ilustración 7-7 Paquete Control Entrada	78
Ilustración 7-8 Paquete Control Entrada Modelo	78
Ilustración 7-9 Paquete Control Entrada Gestión.....	81
Ilustración 7-10 Paquete Control Entrada Estadísticas.....	83
Ilustración 7-11 Paquete Modelo Aplicación	86
Ilustración 7-12 Paquete Control Aplicación	90
Ilustración 7-13 Paquete Control Gestión	90
Ilustración 7-14 Paquete Lectura	94
Ilustración 7-15 Paquete Estadística.....	102
Ilustración 7-16 Paquete Persistencia.....	105
Ilustración 7-17 Paquete Base de Datos.....	105
Ilustración 7-18 Paquete Archivos.....	107
Ilustración 7-19 Diagrama de Secuencia Insertar Archivo Log - Formato de Archivo	111
Ilustración 7-20 Diagrama de Secuencia Insertar Archivo Log - Comprobar datos.....	112
Ilustración A-1 Árbol de carpetas	128

Capítulo 1 - Introducción

1.1 Descripción

En el mundo actual, los avances logrados en el campo de las comunicaciones han permitido extender de manera notable el uso de Internet. Todos los días se desarrollan miles de sitios web de acceso público. Estos portales ofrecen a sus usuarios numerosos servicios que pueden ir desde la más simple información a la realización de compras y transacciones bancarias, la búsqueda de empleo o estudios, la interacción en las diferentes redes sociales, etc...

Sin embargo, dichos sitios web no son elementos estáticos. Dejando a un lado la información que proporcionan, que cambia constantemente; el modo en el que se proporciona y la forma en la que se demanda también es variante. Debido a este hecho, un administrador de un sistema web encargado del mantenimiento de varios de estos portales necesita de un sistema que realice un análisis individual de cada uno de ellos, que le permita valorar su comportamiento y rendimiento para tomar decisiones sobre su forma, la estructura de su contenido, los elementos alojados en el servidor que lo componen y los parámetros del propio servidor que lo aloja. Estas decisiones deberán estar encaminadas a obtener mejoras en la usabilidad, navegabilidad, eficiencia y tiempos de respuesta de la web, para poder proveer a sus usuarios de una mejor experiencia que conlleve grandes beneficios a los prestadores del servicio de cada web.

Este trabajo aborda los aspectos relativos al análisis web, centrándose en la realización del mismo a través del procesamiento de los archivos log que se generan en los servidores y la obtención de diferentes tipos de estadísticas. Describe el proceso de modelado y creación de un sistema que aproveche dicho procesamiento de archivos log para mostrar estadísticas que resulten útiles para los administradores web. Y finaliza exponiendo algunas conclusiones y posibles extensiones para la herramienta.

1.2 Objetivos

El objetivo principal del trabajo es el desarrollo de un sistema de análisis de rendimiento y planificación web dirigido a administradores encargados del mantenimiento de diversos proyectos de mejora de un sitio web. Dicho sistema ha de ser capaz de ofrecer sus servicios de forma interactiva a través de la red proporcionando las siguientes funcionalidades:

- Descripción de diferentes proyectos de mejora de una manera simple.
- Introducción de datos analizables en cada proyecto de mejora.
- Realización de diferentes análisis individuales de cada uno de los proyectos.
- Obtención de diversas estadísticas de cada proyecto.

Para realizar este objetivo principal, se plantean una serie de objetivos secundarios, derivados de un breve análisis del mismo. Dichos objetivos son:

- Estudiar el análisis web para encontrar las mejores fuentes de datos.
- Analizar los diferentes archivos log de servidor para resolver de qué datos disponemos.
- Investigar las diversas estadísticas que se pueden obtener con los datos disponibles.
- Implementar los algoritmos adecuados para la obtención de dichas estadísticas.

Y, en general, como trabajo fin de grado, se proponen como objetivos:

- Aplicar, a modo de resumen y de forma global, todo los conocimientos adquiridos durante el grado.
- Profundizar en el estudio del análisis y evaluación de sistemas (asignatura cursada en el grado), aplicados a servidores web.

1.3 Organización de la memoria

Esta memoria presenta un estudio sobre los diferentes aspectos que confrontan el análisis web y detalla el desarrollo de una herramienta que facilita alguna de las tareas relacionadas con dicho análisis descritas en el estudio.

La estructura de la memoria es la siguiente:

Capítulo 1 - Introducción. Es el capítulo actual. En él se hace una breve presentación del tema que se aborda en el trabajo, junto con los objetivos del mismo, y se describe el contenido y estructura de la memoria.

Capítulo 2 - Plan de Desarrollo. En este capítulo se exponen las decisiones generales tomadas antes de empezar, enumerando objetivos, prioridades, suposiciones, dependencias, restricciones y posibles riesgos e identificando tareas y la duración de las mismas.

Capítulo 3 - Entorno. En este capítulo se detallan las herramientas utilizadas para la realización del trabajo y el entorno necesario para la ejecución de la herramienta.

Capítulo 4 - Análisis Web. En este capítulo se define el análisis web. Se enumeran sus tipos, sus fuentes y los diferentes datos que se pueden obtener para realizarlo. Se concretan los conceptos que derivan de dichos datos y como realizar el análisis basado en los ficheros Log.

Capítulo 5 - Requisitos. En este capítulo se enumeran los requisitos funcionales y no funcionales del sistema a realizar.

Capítulo 6 - Análisis. En este capítulo se identifican los actores, los roles y los casos de uso y se presenta el primer modelo del sistema.

Capítulo 7 - Diseño. En este capítulo se desarrolla el diseño del sistema, describiendo su estructura, clases y funcionamiento con diferentes diagramas.

Capítulo 8 - Pruebas. En este capítulo se exponen las pruebas realizadas al sistema con sus objetivos y resultados.

Capítulo 9 - Conclusiones y Extensiones. En este capítulo se hace una revisión de los objetivos, explicando los alcanzados y las posibles líneas futuras del sistema.

Apéndice A : Contenido del CD. En este apéndice se muestra el contenido del CD. El cual incluye, entre otras cosas, un manual de instalación y otro de usuario del sistema.

Apéndice B : Bibliografía. En este apéndice se lista la bibliografía utilizada durante el desarrollo del trabajo.

Capítulo 2 - Plan de Desarrollo

2.1 Decisiones iniciales

En este apartado se exponen brevemente el conjunto de decisiones preliminares y primeros pasos que se toman durante la fase inicial del proyecto.

Dado que este trabajo queda perfectamente enmarcado en el desarrollo de software tradicional y se lleva a cabo en un contexto educativo, se considera que UPEDU es la metodología más adecuada para realizarlo gracias a las ventajas de planificación, estabilidad, continuidad y seguridad que proporciona.

Partiendo del objetivo, que como ya se explicó con anterioridad, es el desarrollo de un sistema de análisis web, se comenzará realizando un estudio exhaustivo de dicho análisis. Se enumerarán los tipos y las distintas fuentes de datos; poniendo especial énfasis en los archivos log de servidor, que son la base que utilizará el sistema para realizar su cometido.

Todo este esfuerzo inicial será de gran utilidad en el momento de abordar el desarrollo del proyecto. La parte esencial del estudio se presenta en el Capítulo 4 y es imprescindible para la comprensión de los siguientes capítulos.

Pero, previo a este análisis en profundidad del problema a resolver, se expondrán las herramientas disponibles para encontrar la solución más adecuada, modular y adaptable que se utilizarán posteriormente.

A continuación, se detallan brevemente los aspectos más importantes de la metodología escogida y como se aplica durante el desarrollo del proyecto.

2.2 Metodología empleada

UPEDU (Unified Process for EDUcation) es un marco de desarrollo de software cuyo objetivo es asegurar la producción de software de alta calidad que satisfaga las necesidades de los usuarios finales en un tiempo predecible y dentro de un contexto educativo. En cierto modo es el tradicional UP (Unified Process), proceso iterativo e incremental, pero dirigido a la iniciación y aprendizaje del propio UP.

Un proyecto UPEDU organiza el trabajo y las iteraciones en cuatro fases:

- **Inicio:** visión aproximada, análisis del negocio, alcance y estimaciones.
- **Elaboración:** visión refinada, identificación de la mayoría de los requisitos, implementación iterativa del núcleo central de la arquitectura, resolución de riesgos altos.
- **Construcción:** implementación iterativa del resto de requisitos de menor riesgo y de otros elementos, preparación para la entrega y el despliegue.
- **Transición:** Pruebas definitivas, entrega y despliegue.

En cada una de dichas fases se pueden efectuar uno o varios de los siguientes flujos de trabajo:

- **Requisitos:** Análisis y comprensión del negocio junto con la recogida de los requisitos para la aplicación, los cuales a su vez se pueden dividir en requisitos funcionales y no funcionales.
- **Análisis y Diseño:** Todos los aspectos del análisis y el diseño, incluyendo la arquitectura global, objetos, bases de datos, red, etc...
- **Implementación:** Construcción propiamente dicha del sistema mediante programación o codificación automática a partir de diagramas.

- **Pruebas:** Comprobación realizada sobre cada una de las partes del sistema para asegurar su correcto funcionamiento.
- **Despliegue:** Puesta en marcha del sistema en el entorno real de trabajo.
- **Apoyo:** Gestión de Configuraciones, Gestión del Proyecto y Entorno...

Al ser UPEDU un marco orientado a la educación la importancia de ciertos flujos de trabajo muy relevantes en otros contextos queda ligeramente atenuada, como puede ser la de los flujos de apoyo, o hace que se produzca una fusión entre flujos, como el modelado del negocio que se incluye dentro de los requisitos.

En la Ilustración 2-1 se puede observar una medida aproximada de la presencia de los flujos de trabajo en cada una de las fases en un proyecto como el desarrollado en este trabajo:



Ilustración 2-1 Flujos de trabajo por fase

Finalmente UPEDU indican algunas normas y principios que también se seguirán durante la realización de este proyecto:

- Partiendo de los requisitos se irán construyendo una serie de modelos, al principio más sencillos y cada vez más refinados y detallados, hasta llegar a un modelo solución.
- El método de trabajo será continuo e incremental. Para ello, se procurará que cada uno de los modelos aporten funcionalidad final al sistema. Todos los esfuerzos, por tanto, irán dirigidos hacia la implementación final.
- Se considerará imprescindible que cada uno de los componentes software y los servicios desarrollados tengan la mayor independencia posible para, mediante el aumento de la cohesión y la disminución del acoplamiento, favorecer su reutilización dentro de la arquitectura.

2.3 Plan de Proyecto

En este apartado se describe el plan general que se seguirá para llevar a cabo el proyecto. Se definen cómo deben realizarse todas las fases del proyecto, las actividades y el tiempo de duración de las mismas, así como los recursos disponibles, los hitos y los riesgos a considerar.

2.3.1 Organización del Proyecto

2.3.1.1 Recursos Humanos

Al tratarse de un Trabajo Fin de Grado el único personal disponible presente durante todo su desarrollo, es:

- El tutor del proyecto, José Manuel Marqués Corral, profesor del departamento de Informática de la Universidad de Valladolid, que desempeña el rol compartido de cliente y usuario de la aplicación permitiendo la obtención de los requisitos y funcionalidades deseadas para la herramienta a construir, así como las revisiones parciales y final de dicha herramienta. Simultáneamente, también compartirá el rol de jefe de proyecto, realizando un seguimiento continuo del trabajo y proponiendo sugerencias sobre el mismo. Todo ello gracias a una serie de reuniones celebradas desde el inicio del proyecto.
- El alumno encargado de la realización del proyecto, Gustavo Velasco Sastre, alumno del Curso de Adaptación al Grado de Informática de la Universidad de Valladolid, que comparte el rol de cliente y jefe de proyecto mencionados anteriormente y asume los demás roles necesarios para la consecución del mismo: analista, diseñador, programador y responsable de pruebas.

2.3.1.2 Recursos Materiales

Durante el desarrollo del trabajo se han utilizado los siguientes recursos materiales:

- **Recursos Hardware:** Dos ordenadores, uno de escritorio para la implementación y pruebas y otro portátil para la realización de documentos y la presentación. Ambos tienen un procesador Pentium i-7 920 a 2,66 GHz con 6 GB de RAM y un sistema operativo Windows 7 Profesional de 64 bits versión 6.1 sobre el que se ha montado una máquina virtual con un Linux Ubuntu versión 12.04.4 LTS, un servidor Apache 2 que hace de servidor de la aplicación y un cliente/servidor de base de datos MySQL.
- **Recursos Software:** Eclipse SDK-4.2.2 como editor para el desarrollo en Python, Adobe Dreamweaver CS 5.5 como editor web, Microsoft Word 2010 para la realización de la documentación y Astah Professional 6.7.0 para la realización de diagramas. Además de estas herramientas ha sido necesario la instalación de otros softwares para el perfecto funcionamiento y desarrollo del trabajo que se detallan en el Capítulo 3.
- **Otros Recursos:** Despacho del tutor del proyecto en la Facultad de Nuevas Tecnologías de la Universidad de Valladolid, para la realización de reuniones que han facilitado la recogida de requisitos y el seguimiento del proyecto, así como otras instalaciones del mismo para su desarrollo y presentación.

2.3.2 Plan de Trabajo

2.3.2.1 Fases

Siguiendo la metodología propuesta, el proyecto se organiza en cuatro fases. La Tabla 2-1 describe dichas fases con sus iteraciones y el tiempo que se empleará en cada una de ellas.

Fase	Nº de Iteraciones	Tiempo
Inicio	1	10 días
Elaboración	1	30 días
Construcción	2	50 días
Transición	1	10 días

Tabla 2-1 Fases del proyecto

Al realizar la planificación se ha considerado que se trabajan 4 horas al día. Por tanto como la duración total del proyecto es de 100 días, es decir, aproximadamente 4 meses y medio naturales, el número de horas que se dedicarán a él es de 400.

2.3.2.2 Planes de Fases

Cada una de las iteraciones de las fases en las que se divide el proyecto termina con la consecución de un hito del mismo. En la Tabla 2-2 se describen los diferentes hitos del proyecto junto con su correspondiente iteración y la duración de la misma.

Fase	Iteración	Tiempo	Hito
Inicio	1	12 días	Plan de desarrollo y documento de riesgos.
Elaboración	1	28 días	Modelo de análisis, diseño y datos inicial.
Construcción	1	25 días	Versión alfa del software con gestión de proyectos, estadísticas de geolocalización y estadísticas de perfil de navegación.
	2	25 días	Versión beta del software con estadísticas de perfiles temporales y estadísticas de rendimiento y carga del servidor.
Transición	1	10 días	Versión final del software probada con manual de instalación, manual de uso y memoria del trabajo realizado.

Tabla 2-2 Iteraciones por fase

A continuación se describen las diferentes tareas realizadas en cada una de las fases para la consecución de los mencionados hitos. Cabe destacar que todas estas tareas se llevan a cabo de forma secuencial y que no se realiza una calendarización de las mismas, puesto que al compatibilizar la realización del trabajo con otros proyectos y asignaturas cuya carga es difícil de prever, la cantidad de trabajo que se dedica al mismo puede ser variable en el tiempo, con periodos de inactividad y periodos que sobre pasen las horas diarias estimadas; aun así a cada tarea se le dedica el tiempo aquí estipulado y la finalización será antes de la fecha de presentación en el curso en el que se desarrolla.

2.3.2.2.1 Fase de Inicio

La fase de inicio es el comienzo del desarrollo del software en la que entre las dos partes implicadas en el proyecto se deciden los principios del sistema, el tiempo necesario para realizarlo y la forma más adecuada de hacerlo.

Los objetivos principales de esta fase son: establecer el alcance del software, incluyendo una visión del mismo con los criterios para la aceptación de su funcionamiento; enumerar los casos de uso críticos del sistema, con los escenarios principales; proponer una arquitectura candidata para su desarrollo; estimar el tiempo necesario y los riesgos potenciales; y preparar el entorno de apoyo para el proyecto.

Para la conclusión de todos estos objetivos se realizan las tareas señaladas en la Tabla 2-3.

Tarea	Duración
Formulación del Alcance	12 horas
Elaboración de la Planificación	16 horas
Identificación de Riesgos	6 horas
Propuesta de Arquitectura	5 horas
Preparación del Entrono	5 horas
Preparación de la siguiente fase	4 horas

Tabla 2-3 Tareas de la fase de inicio

2.3.2.2.2 Fase de Elaboración

La fase de elaboración abarca la mayor parte del análisis y el diseño de la aplicación. Se centra en conseguir una arquitectura del sistema sólida a partir de los requisitos más importantes y un análisis adecuado de los casos de uso.

La estabilidad de la arquitectura es esencial para mitigar riesgos durante el diseño y la implementación por lo que en esta fase es necesario realizar las tareas enumeradas en la Tabla 2-4.

Tarea	Duración
Análisis en Profundidad	36 horas
Recogida de Requisitos	12 horas
Especificación de Casos de Uso	12 horas
Definición de la Arquitectura	8 horas
Descripción del Modelo de Datos	8 horas
Elaboración del Diagrama de Clases	12 horas
Realización de Diagramas de Secuencia	12 horas
Consolidación del Entrono	8 horas
Preparación de la siguiente fase	4 horas

Tabla 2-4 Tareas de la fase de elaboración

2.3.2.2.3 Fase de Construcción

La fase de construcción se ha dividido en dos iteraciones que se centrarán en diferentes partes funcionales del sistema. De esta forma será más sencillo aclarar los requisitos restantes, terminal el diseño y completar el desarrollo del sistema basándose en la arquitectura propuesta.

2.3.2.2.3.1 Iteración 1

En esta iteración se tiene como objetivo la construcción de la parte del sistema que se refiere a la gestión de proyectos, la que realiza las estadísticas de geolocalización y la que calcula los perfiles de navegación. Para ello se realizan las tareas indicadas en la Tabla 2-5.

Tarea	Duración
Recogida de Requisitos Específicos	4 horas
Revisión de Casos de Uso	4 horas
Consolidación del Diagrama de Clases	6 horas
Fijación de los Diagramas de Secuencia	6 horas
Implementación del Módulo de Gestión	16 horas
Pruebas del Módulo de Gestión	3 horas
Implementación del Módulo de Estadísticas de Geolocalización	20 horas
Pruebas del Módulo de Estadísticas de Geolocalización	5 horas
Implementación del Módulo de Estadísticas de Perfil de Navegación	24 horas
Pruebas del Módulo de Estadísticas de Perfil de Navegación	8 horas
Preparación de la siguiente iteración	4 horas

Tabla 2-5 Tareas de la fase de construcción - Iteración1

2.3.2.2.3.2 Iteración 2

En esta nueva iteración se tiene como objetivo la construcción de la parte del sistema que se refiere a la elaboración de estadísticas de perfiles temporales y la que calcula el perfil de rendimiento del servidor. Para ello se realizan las tareas detalladas en la Tabla 2-6.

Tarea	Duración
Recogida de Requisitos Específicos	6 horas
Revisión de Casos de Uso	6 horas
Consolidación del Diagrama de Clases	8 horas
Fijación de los Diagramas de Secuencia	8 horas
Implementación del Módulo de Estadísticas de Perfiles Temporales	24 horas
Pruebas del Módulo de Estadísticas de Perfiles Temporales	8 horas
Implementación del Módulo de Estadísticas de Rendimiento y Carga	28 horas
Pruebas del Módulo de Estadísticas de Rendimiento y Carga	8 horas
Preparación de la siguiente fase	4 horas

Tabla 2-6 Tareas de la fase de construcción - Iteración 2

2.3.2.2.4 Fase de Transición

En la fase de transición se realizan las pruebas globales, se pone a punto el sistema y se entrega junto a los manuales y la memoria del trabajo. En esta fase el objetivo es la satisfacción del cliente, que en este caso, en un trabajo fin de grado, se traduce en la entrega de acuerdo a la especificación del proyecto y según la normativa de la universidad. Para ello se realizan las siguientes tareas:

Tarea	Duración
Integración de Módulos	8 horas
Pruebas Globales	8 horas
Manual de Instalación	8 horas
Manual de Usuario	8 horas
Recopilación de la Memoria	7 horas
Entrega y Presentación	1 hora

Tabla 2-7 Tareas de la fase de transición

2.3.2.3 Tareas de Apoyo

Durante todas las fases del proyecto se realizan una serie de tareas simultáneamente a las tareas programadas que permiten un control y seguimiento de las mismas y que se realizarán de forma conjunta entre el tutor y el alumno. El tiempo dedicado a dichas tareas está incluido dentro de la planificación anterior computando como si el seguimiento y control fuese parte de cada tarea del proyecto. Las tareas de apoyo se encuentran debidamente descritas en el apartado 2.3.3 Plan de Seguimiento y Control.

2.3.2.4 Entregas

En este apartado se indican y describen cada uno de los artefactos entregables que serán generados durante el proyecto. Es preciso destacar que, de acuerdo a la filosofía de UPEDU que propone un proceso iterativo e incremental, todos los artefactos son objeto de modificaciones a lo largo de dicho proceso, con lo cual, solo a su término se obtiene una versión definitiva y completa de cada uno. Sin embargo, en la conclusión de cada iteración se ha de conseguir cierto grado de completitud y estabilidad de dichos artefactos para su entrega.

También es necesario indicar que aunque el proceso seguido tanto organizativamente como temporalmente se ajusta a UPEDU, la documentación entregada, aunque basada en sus estándares, puede que no los siga estrictamente para una mayor adecuación al trabajo y un menor coste temporal.

2.3.2.4.1 Fase de Inicio

- **Plan de Desarrollo:** Descripción del plan general que se seguirá para llevar a cabo el proyecto. Incluye cómo deben realizarse todas las fases del proyecto, las actividades y el tiempo de duración de las mismas, así como los recursos disponibles, los hitos y las entregas.
- **Lista de Riesgos:** Listado de los riesgos conocidos y vigentes en el proyecto, establecidos en orden decreciente de importancia y con las acciones de contingencia necesarias para su mitigación.

2.3.2.4.2 Fase de Elaboración

- **Documento de Análisis:** Visión general del sistema desde la perspectiva del cliente, especificando características del mismo y del modelado del negocio.
- **Lista de Requisitos:** Listado de los requisitos funcionales y no funcionales del sistema que constituye una base de acuerdo con el cliente en cuanto a los mismos.
- **Documento de Casos de Uso:** Representación y descripción de las principales funcionalidades del sistema y su actores.
- **Modelo de Datos:** Descripción de las entidades que componen el sistema y sus relaciones.
- **Documento de Arquitectura:** Presentación de la arquitectura del sistema.
- **Diagrama de Clases:** Diagrama y especificación de las clases del sistema.
- **Diagramas de Secuencia:** Representación de la interacción entre los diferentes objetos del sistema basada en los casos de uso.
- **Documento de Entorno:** Descripción de las herramientas y tecnologías empleadas durante el desarrollo del proyecto.
- **Lista de Pruebas:** Listado de pruebas que se realizan sobre el sistema con sus condiciones de ejecución, entradas, pasos a seguir y resultados esperados y obtenidos.

2.3.2.4.3 Fase de Construcción - Iteración 1

- **Versión alfa del software:** Sistema a entregar con las funcionalidades de gestión de proyectos, realización de estadísticas de geolocalización y de perfil de navegación.

2.3.2.4.4 Fase de Construcción - Iteración 2

- **Versión beta del software:** Sistema a entregar con las funcionalidades de realización de estadísticas perfiles temporales y de rendimiento de servidor.

2.3.2.4.5 Fase de Transición

- **Manual de Instalación:** Instrucciones para realizar la instalación del sistema.
- **Manual de Usuario:** Ayuda al usuario para la utilización de la herramienta describiendo de forma simple cómo realizar cada una de las funcionalidades disponibles.
- **Memoria del Trabajo:** Compendio de todos los documentos entregados donde se describe el trabajo realizado.
- **Sistema Completo:** Herramienta producto del desarrollo.

2.3.3 Plan de Seguimiento y Control

2.3.3.1 Control de Requisitos

Los requisitos del sistema están especificados en el documento pertinente. Cada requisito está debidamente clasificado según su tipología y tiene una serie de atributos tales como número de requisito, nombre, descripción y dependencias.

Este documento es revisado en cada una de las reuniones que se mantienen durante el desarrollo del proyecto y aunque no se lleva un control exhaustivo se tiene en cuenta el número de cambios de dicho documento para evaluar tanto la eficiencia de recogida de requisitos como el carácter cambiante de los mismos durante el proyecto.

Los requisitos finalmente están expuestos en el Capítulo 5 de esta memoria.

2.3.3.2 Control de Calidad

Cada uno de los entregables producidos será debidamente revisado y probado por ambos miembros del proyecto para asegurar cierto nivel de calidad tanto en el proceso como en el producto final.

2.3.3.3 Control de Gestión de Riesgos

Durante la fase de inicio se elabora un documento con una lista de riesgos asociados al proyecto y sus respectivos planes de contingencia para poder mitigarlos y/o controlarlos. Dicha lista será evaluada al menos una vez en cada iteración de cada fase del proyecto.

Los riesgos finalmente están expuestos en el apartado 2.3.4 Gestión de Riesgos de esta memoria.

2.3.3.4 Gestión de Configuraciones

Se realizará una gestión de configuración para llevar un registro de los artefactos generados y sus versiones. Para los documentos consistirá en crear una carpeta para cada documento y guardar en la misma cada una de las versiones generadas del mismo. Para el código se utilizará una carpeta para cada módulo con una subcarpeta que también indique la fecha de la versión. Se realizará una copia incremental de todas estas carpetas una vez por semana y se compartirá la última versión de los documentos a través de la herramienta DropBox.

2.3.4 Gestión de Riesgos

Un riesgo es una medida de la probabilidad y pérdida de que se produzca un resultado inadecuado que afecte al proceso o al producto final a entregar. Por ello, se realiza un pequeño estudio de los posibles riesgos que pueden afectar al desarrollo del proyecto, que aunque en este caso, por estar en un entorno docente, no van a repercutir en ningún coste de tipo económico, si pueden provocar fallos en la planificación que generen retrasos.

A continuación se procede a la identificación de cada uno de los riesgos que pueden darse durante el presente proyecto, junto con su alcance, valoración, descripción, disparador del plan de acción y el propio plan de acción o contingencia que permita evitarlos o, en el peor de los casos, mitigar su efecto.

RISK-01	Inexperiencia del jefe de proyecto		
Alcance	Todo el proyecto		
Probabilidad	Media	Impacto	Alto
Descripción	La falta de experiencia del jefe de proyecto puede llevar a una inadecuada planificación y/o ejecución del proyecto.		
Disparador	El tutor del proyecto encuentra fallos en la planificación o las tareas a realizar comienzan a retrasarse sin motivo aparente.		
Plan de Acción	El tutor del proyecto aportará la experiencia y sugerencias necesarias para llevar a cabo el proyecto lo mejor posible.		

Tabla 2-8 Riesgo - Inexperiencia del jefe de proyecto

RISK-02	Requisitos no identificados claramente		
Alcance	Todo el proyecto		
Probabilidad	Baja	Impacto	Alto
Descripción	Los requisitos no siempre tienen una misma interpretación, depende de la persona. La inexperiencia y las suposiciones pueden llevar a ambigüedades.		
Disparador	Duda sobre la aplicación y/o el alcance de un requisito o diferencias de criterio sobre una funcionalidad.		
Plan de Acción	Se consultará cualquier duda sobre cualquier requisito cuyo alcance o definición pueda tener una interpretación ambigua.		

Tabla 2-9 Riesgo - Requisitos no identificados claramente

RISK-03	Arquitectura, modelo o diseño erróneos		
Alcance	Fase de Elaboración, Construcción y Transición		
Probabilidad	Baja	Impacto	Alto
Descripción	Las diversas técnicas para el cálculo de estadísticas en el análisis web pueden no ajustarse bien a la arquitectura, el modelo de datos o el diseño propuestos.		

RISK-03	Arquitectura, modelo o diseño erróneos
Disparador	El cálculo de una estadística se complica durante el diseño o su implementación comienza a retrasarse.
Plan de Acción	Revisar los requisitos referentes a dicha estadística, buscar nueva información sobre como poder resolver el cálculo y generalizar todo lo posible el mismo.

Tabla 2-10 Riesgo - Arquitectura, modelo o diseño erróneos

RISK-04	Inexperiencia en la tecnología y herramientas a emplear		
Alcance	Fase de Elaboración, Construcción y Transición		
Probabilidad	Media	Impacto	Medio
Descripción	Debido al lenguaje de programación escogido se pueden no poseer los conocimientos adecuados para el desarrollo.		
Disparador	Se visualiza un problema a abordar en el futuro del que posiblemente no se tenga el conocimiento de una solución.		
Plan de Acción	Realizar una labor de búsqueda de bibliografía adecuada, actualizada y de calidad que facilite la labor de aprendizaje y permita adquirir el conocimiento para hallar la solución.		

Tabla 2-11 Riesgo - Inexperiencia en la tecnología y herramientas a emplear

RISK-05	Falta de actividad o disponibilidad temporal		
Alcance	Todo el proyecto		
Probabilidad	Alta	Impacto	Bajo
Descripción	Debido la necesidad por parte de ambos integrantes del proyecto de compatibilizar la realización del trabajo con otros proyectos y asignaturas cuya carga es difícil de prever, pueden darse periodos de inactividad o falta de disponibilidad.		
Disparador	Previsión de una temporada con alta carga de trabajo, u otra actividad, externa al proyecto.		
Plan de Acción	Comunicar al otro integrante del proyecto cualquier eventualidad de este tipo y planificar una menor carga de trabajo durante ese periodo y recuperarla cuando este termine.		

Tabla 2-12 Riesgo - Falta de actividad o disponibilidad temporal

RISK-06	Fallos en la integración de diferentes módulos		
Alcance	Fase de Transición		
Probabilidad	Baja	Impacto	Bajo

RISK-06	Fallos en la integración de diferentes módulos
Descripción	Debido a diferentes interpretaciones del diseño en cada momento o pequeñas desviaciones en la realización del modelo físico de datos pueden darse algunas incompatibilidades al integrar los diversos módulos.
Disparador	Utilización de artefactos (módulos, clases u objetos) que se desarrollen en otras partes del diseño pero cuya funcionalidad no se pueda desenvolver de la misma forma.
Plan de Acción	Definir bien el diseño haciéndolo lo menos ambiguo posible y utilizar las mismas técnicas de trabajo y llegado el caso renombrar y generalizar.

Tabla 2-13 Riesgo - Fallos en la integración de diferentes módulos

Aunque en cada fase e iteración es posible que la probabilidad de un riesgo y su impacto sean diferentes, en este caso, se ha realizado una estimación conjunta; y para ello se han utilizado los valores de probabilidad indicados en la Tabla 2-14 y los de impacto señalados en la Tabla 2-15.

Valor de Probabilidad	Rango de Porcentaje
Alta	[67% , 100%]
Media	[33%, 67%)
Baja	[0%, 33%)

Tabla 2-14 Valores de probabilidad

Valor de Impacto	Afecto
Alto	Fracaso en los objetivos del proyecto
Medio	Fracaso con degradación del proceso y del sistema final
Bajo	Fracaso de grado menor

Tabla 2-15 Valores de impacto

2.4 Evaluación del Plan

El plan de proyecto que se obtuvo de la planificación se quedó ligeramente corto debido a los grandes avances de los últimos años en el análisis web y su multitud de interpretaciones que alargaron el estudio necesario para establecer el modelado del negocio, aun siendo a priori una tarea para la que se reservaron 36 horas, tiempo que se pensó suficiente.

Unido a la anterior, también surgieron los siguientes problemas:

- **Falta de actividad o disponibilidad temporal:** Las fases de elaboración y construcción se solaparon con periodos vacacionales y de exámenes donde existe una insalvable incompatibilidad de horarios entre el tutor y el alumno.

- **Inexperiencia en la tecnología y herramientas a emplear:** Desconocimiento del entorno necesario para el lenguaje de programación y notables diferencias con la configuración de dicho entorno con respecto a otros lenguajes.
- **Arquitectura, modelo o diseño erróneo e inexperiencia en la tecnología y herramientas a emplear:** Tratando de minimizar al máximo errores que otras herramientas parecidas a la propuesta cometen, como incluir accesos repetidos, inicialmente se propuso un modelo de datos poco adecuado que desembocó en tiempos excesivamente altos durante la inserción de los mismos. Posteriormente, se amplió el modelo de datos, se utilizaron otros métodos de inserción y se relajaron algunas especificaciones para poder hacer un software adecuado, útil y aprovechable.
- **Inexperiencia en la tecnología y herramientas a emplear:** Durante la fase de construcción se trataron de incluir librerías incompatibles con el método de programación utilizado, que no tenían los permisos adecuados en el servidor o que no poseían las clases requeridas. Esto fue debido al desconocimiento del lenguaje de programación, la poca documentación sobre mod_pyton y sus considerables diferencias con otros lenguajes.

En el momento en el que se predijo que se podrían dar las dificultades descritas, se llevaron a cabo los planes de acción correspondientes. Dichos planes redujeron el impacto sobre la planificación, pero en algunas ocasiones el tiempo del proyecto se vio afectado.

En la Tabla 2-16 se muestra el tiempo estimado durante la planificación frente al tiempo dedicado realmente en cada fase:

Fase	Iteración	Tiempo Estimado	Tiempo Dedicado
Inicio	1	12 días	12 días
Elaboración	1	28 días	30 días
Construcción	1	25 días	30 días
	2	25 días	25 días
Transición	1	10 días	11 días

Tabla 2-16 Duración del proyecto

Como se puede observar, debido a los problemas surgidos, el tiempo total dedicado al proyecto fue de 108 días (432 horas) frente a los 100 días (400 horas) planificados inicialmente.

Capítulo 3 - Entorno

3.1 Introducción

En este capítulo se describe el conjunto de herramientas, módulos y bibliotecas utilizadas para el desarrollo del trabajo.

Dado que este trabajo consiste en la realización de un sistema que ha de ser capaz de ofrecer sus servicios de forma interactiva a través de la red, la forma más adecuada de abordarlo es la realización de un sitio web. Lo cual hace necesario un conjunto de herramientas que permitan dicho desarrollo, además de un entorno para la realización de pruebas.

Para la elaboración del sitio web es necesario un editor de páginas web; para codificar la funcionalidad adicional, un editor que soporte el lenguaje escogido; para la documentación, un editor de textos; y para las pruebas, un servidor web configurado de la forma adecuada y, al menos, un navegador con el que acceder a la web.

A continuación, se detalla el software utilizado, las razones o motivos por los que ha sido elegido y, en su caso, la configuración establecida.

3.2 Herramientas utilizadas

Para el desarrollo del trabajo, como se comentó en el Capítulo 2, se utilizan dos ordenadores muy similares en los que está instalado un sistema operativo Windows 7 Profesional de 64 bits versión 6.1.

Sobre este sistema operativo, para realizar la página web, se utiliza el editor Adobe Dreamweaver CS 5.5 v11.5 por su extraordinaria facilidad a la hora de desarrollar páginas web gracias al autocompletado de código HTML, CSS y JavaScript y al reconocimiento de archivos cargados en cada página.

Para escribir la documentación y esta memoria se utiliza Microsoft Office 2010. En concreto, se usa el editor de textos Microsoft Word 2010 v14.0 por su facilidad a la hora de crear diferentes estilos de páginas, párrafos y tablas, su capacidad de dividir el documento en secciones y de crear índices tanto de contenidos como de imágenes y tablas.

Los diagramas incluidos en los documentos y en esta memoria se realizan en Astah Professional 6.7.0 por su simplicidad, disponibilidad, facilidad de uso y adecuación al estándar UML.

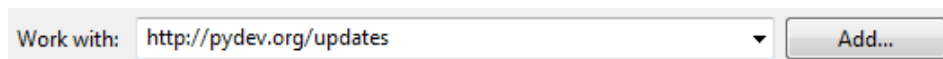
En cuanto al entorno para la realización de pruebas, es necesario instalar un servidor que aloje el sistema como si fuese el entorno de producción en el que se utilizará. Para simular peticiones a dicho sistema como si se realizasen desde otro ordenador a través de la red, se crea una máquina virtual con ayuda del programa VMware Player v6.0. En dicha máquina virtual, con un sistema operativo Linux Ubuntu versión 12.04.4 LTS de 64 bits, se instala un servidor Apache 2.2.22 y una base de datos MySQL v5.5.32. Para el acceso al sitio se instala fuera de la máquina virtual el navegador Google Chrome v30.0.

El lenguaje elegido para la codificación de la funcionalidad del sistema es Python versión 2.7.3 y para integrarlo con la web se elige utilizar el módulo de Apache mod_pyton. Tras esta decisión se prueba con diversos editores intentando encontrar uno que sea compatible con el módulo y que a la vez permita editar HTML. Se encontraron diversos sistemas, como Komodo Edit, que ofrecen compatibilidades con otros módulos o entornos de trabajo, como Django o Pyramid, pero ninguno se integra correctamente con mod_pyton, ni permite la edición en código HTML y Python con capacidades como el autocompletado y la realización de sugerencias. Por ello, se opta por seguir con la edición de la web utilizando Adobe Dreamweaver y para la de código Python se escoge utilizar Eclipse Juno SDK v4.2.2 configurándolo previamente.

3.2.1 Configuración de Eclipse



Para poder editar código Python en Eclipse Juno SDK V4.2.2 es necesario utilizar el complemento PyDev. Para ello lo primero que se debe hacer es descargarlo e instalarlo desde el propio editor siguiendo estos pasos:

- En el menú de Eclipse se escoge *Help* y después *Instal New Software*.
- En el cuadro de texto *Work with:* se escribe <http://pydev.org/updates>.



Work with:

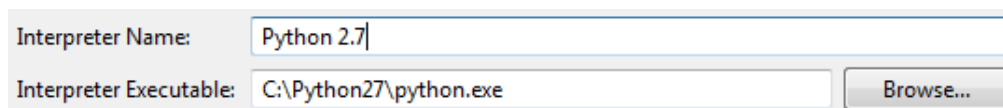
- Unos segundos después aparecen dos opciones, se selecciona *PyDev* y se pulsa el botón *Next*.

Name	Version
<input checked="" type="checkbox"/>  PyDev	
<input type="checkbox"/>  PyDev Mylyn Integration (optional)	

- Automáticamente se calculan las dependencias y se muestra en pantalla todos los módulos necesarios.
- Se sigue el proceso aceptando las condiciones de uso y los certificados necesarios hasta terminar la instalación.
- Por último, se reinicia el editor.






Una vez concluida la instalación, es necesario realizar su configuración siguiendo estos pasos:

- En el menú de Eclipse se escoge *Window* y después *Preferences*.
- En el cuadro lateral se escoge *PyDev*, a continuación *Interpreter - Python* y se pulsa el botón *New*.
- Los cuadros de texto que aparecen en la nueva ventana se rellenan con el nombre del nuevo intérprete y la carpeta donde se encuentra el ejecutable del mismo. A continuación, se pulsa el botón *OK*.



Interpreter Name:
 Interpreter Executable:

- Se sigue el proceso aceptando las nuevas entradas que se añadirán al registro o al path del sistema hasta terminar la instalación.

-  C:\Python27
-  C:\Python27\DLLs
-  C:\Python27\lib
-  C:\Python27\lib\plat-win
-  C:\Python27\lib\site-packages

- Por último, se reinicia el editor.

Una vez realizado esto, se puede comenzar un nuevo proyecto en eclipse de tipo PyDev que permite la codificación en lenguaje Python.

3.3 Lenguaje Python

Python es un lenguaje de programación de propósito general con una sintaxis limpia que permite escribir códigos muy legibles. Soporta múltiples paradigmas de programación, incluyendo la orientación a objetos, la programación imperativa y la programación funcional o procedural. Es un lenguaje interpretado de tipado dinámico y multiplataforma con un sistema de gestión automática de memoria y una biblioteca estándar muy amplia. Posee una licencia de código abierto, denominada *Python Software Foundation License* que es compatible con la licencia pública general de GNU a partir de la versión 2.1.1.

Python se escoge como lenguaje de programación para este proyecto por diversas razones. La primera es que permite la programación web siendo interpretado en el servidor donde se puede ejecutar toda la lógica de negocio y después enviar los resultados al cliente. Otra de las razones es su sencillez, potencia y amplia biblioteca estándar que permiten realizar fácilmente cálculos estadísticos esenciales para el sistema a desarrollar. Por último, al ser un lenguaje maduro está sobradamente probado y se pueden encontrar multitud de bibliotecas que permiten por ejemplo la conexión, si fuese necesaria, con otros lenguajes estadísticos más potentes, como R, o la realización de gráficas, muy útiles para exponer resultados estadísticos.

No obstante, la creación de sitios web con Python no es una tarea simple si no se utiliza alguno de los muchos marcos y herramientas que permiten hacerlo de una forma rápida y con un resultado robusto. Algunos de los métodos utilizados para combinar Python con un servidor web para crear contenido dinámico son:

- **CGI (Common Gateway Interface):** es la interfaz más conocida, antigua y compatible con casi todos los servidores. Es muy sencilla pero en cada petición al servidor se instancia un nuevo intérprete por lo que resulta demasiado lenta.
- **Mod_python:** es un módulo que incrusta el intérprete en el proceso de Apache, acelerando las solicitudes. Permite escribir el código Python embebido en código HTML de la misma forma en que se hace con PHP pero de una manera más potente que ofrece un mayor acceso al interior del servidor Apache. Sin embargo, está vinculado con la versión específica de libpython, teniendo que recompilar al cambiar de versión; y está ligado con el servidor Apache, no siendo muy portable a otros servidores.
- **FastCGI y SCGI:** es una interfaz que en vez de incrustar el intérprete en el servidor, crea procesos que se ejecutan en segundo plano y que incluso permiten ejecutar programas escritos en otros lenguajes que tengan una biblioteca que se encargue de la comunicación con el propio servidor.
- **Mod_wsgi:** es un módulo que proporciona una interfaz que intenta desligarse del código de bajo nivel. Es compatible con dos modos: el modo integrado, en el que se incluye dentro del proceso de Apache, y el modo demonio, que es más parecido a FastCGI aunque controla los procesos por sí mismo, lo que permite una administración más fácil.
- **WSGI (Web Server Gateway Interface):** es una interfaz de alto nivel ideal para desarrolladores web que se puedan basar en marcos ya programados. Tiene una capa de middleware con muchas funciones de uso habitual ya programadas y por debajo es compatible con todas las interfaces anteriores proporcionando una interfaz unificada.

Para la realización de este trabajo se escoge mod_python ya que se considera un módulo con una interfaz simple y de un nivel adecuado para la resolución del mismo.

3.3.1 Mod_Python

Mod_Python es un módulo de servidor Apache desarrollado por la *Apache Software Foundation* que, como se ha comentado, incrusta el intérprete Python dentro del proceso del propio servidor acelerando así las solicitudes, ya que no es necesario instanciar de nuevo al intérprete en cada una de ellas.

Una de las razones por las que se eligió este módulo fue su parecido al módulo `mod_php` del lenguaje PHP, que permite un estilo de trabajo muy similar con el que el desarrollador está más familiarizado. Este estilo consiste en incrustar el código mezclándolo con el código HTML. No obstante, no fue esa la única razón, ya que `mod_python` permite mucho más. Con `mod_python` se tiene acceso a la parte interna del servidor Apache, lo que podría permitir realizar estadísticas web de las páginas alojadas en él sin necesidad de introducir los archivos log, pensando así en una futura ampliación del sistema propuesto para este trabajo.

Aunque `mod_pyton` no está exento de inconvenientes. A diferencia del intérprete de PHP, el intérprete de Python utiliza el almacenamiento en caché al ejecutar archivos, por lo que los cambios en uno de ellos requieren que el servidor web se reinicie, lo que hace un poco más pasada la programación y prueba del sistema.

Otro de sus mayores problemas es que al iniciar procesos hijos para servir las peticiones que llegan al servidor, necesita cargar todo el intérprete Python de nuevo, aunque no lo necesite. Además de estos problemas están los anteriormente mencionados, su vinculación a la versión de la biblioteca `libpython`, que obliga a recompilar al cambiar, y que está ligado con el servidor Apache, no siendo muy portable a otros servidores.

Sin embargo, aunque `mod_python` dejó de desarrollarse en 2010, se ha vuelto a retomar el proyecto en septiembre de 2013 intentando solventar todos estos problemas y, esta vez, al margen de la *Apache Software Foundation*.

3.3.2 Bibliotecas

Mucha de la funcionalidad necesaria para realizar el sistema propuesto se encuentra en la biblioteca estándar de Python. Sin embargo, para determinadas tareas será necesario contar con algunas bibliotecas que ya tengan solucionados algunos de los problemas a resolver.

3.3.2.1 Biblioteca estándar

La biblioteca estándar de Python contiene un gran número de módulos. Algunos de estos módulos están escritos en C y dan acceso a la funcionalidad del sistema, como el módulo para la entrada/salida de archivos. Otros módulos están escritos en Python y proporcionan soluciones estandarizadas para muchos problemas.

Como ayuda para la realización de la herramienta se han utilizado los siguientes módulos de esta biblioteca:

- **Os:** permite la utilización de las funciones del sistema operativo de una forma transparente y portable. Es útil tanto para leer y escribir ficheros como para crear archivos y manipular rutas.
- **Re:** proporciona operaciones con expresiones regulares, útiles para reconocer y segmentar cadenas de caracteres.
- **Math:** incluye todas las operaciones matemáticas básicas.
- **Datetime:** proporciona la funcionalidad necesaria para la manipulación de fechas y horas, pudiendo realizar tanto cambios de formato como operaciones simples sobre ellas.
- **Multiprocessing:** permite la creación y el dialogo entre procesos ofreciendo la posibilidad de utilizar concurrencia y aprovechar varios procesadores, locales o remotos, simultáneamente.

3.3.2.2 Otras Bibliotecas

Aparte de la biblioteca estándar de Python, se han instalado y utilizado las siguientes bibliotecas:

- **python-mysqldb:** proporciona una interfaz para trabajar con bases de datos relacionales mediante el acceso a un gestor MySQL utilizando un cursor.
- **python-geoip:** suministra el acceso a determinadas bases de datos incluyendo información sobre la posible geolocalización de direcciones IP, útil para la realización de estadísticas geográficas.
- **python-numpy:** añade funcionalidad para trabajar con grandes vectores y matrices multi-dimensionales, incluyendo operaciones matemáticas de alto nivel adecuadas para realizar estadísticas.
- **hcluster:** permite el cálculo de matrices de distancias entre observaciones y conglomerados jerárquicos a partir de dichas matrices de distancias.

La Biblioteca de acceso a la base de datos es necesaria para la correcta ejecución de toda la herramienta. Sin embargo, el resto de bibliotecas sólo son utilizadas por algunos módulos para realizar determinadas estadísticas. Pero si no se desearan dichas estadísticas, se podría ejecutar el software eliminando los módulos correspondientes, sin tener que instalar ni utilizar las bibliotecas que importan dichos módulos. Sin embargo, son recomendables por su utilidad para futuros usos y posibles ampliaciones.

Capítulo 4 - Análisis Web

4.1 Introducción

Los avances logrados en los últimos años en el campo de las comunicaciones han permitido extender de manera notable el uso de Internet convirtiéndolo en una enorme fuente de datos. Estos datos provienen del contenido web, representado por miles de millones de páginas de acceso público. Dichas páginas ofrecen a sus visitantes numerosos servicios que pueden ir desde la simple información a la realización de compras y transacciones bancarias, la búsqueda de empleo o estudios, la interacción en las diferentes redes sociales, etc.

Pero éstos no son los únicos datos provenientes de Internet. La utilización de los servicios genera registros en el servidor en el que está alojado dicho servicio. Estos registros contienen un gran número de datos que se van recolectando diariamente. Dichos datos no sólo sirven para medir el tráfico web si no que, analizados adecuadamente, consiguen proporcionar una valiosísima información para los prestadores de los mencionados servicios, pudiéndose realizar estudios de popularidad y/o mercado.

4.2 Definición

El análisis web es un proceso que incluye un conjunto de técnicas para la medición, recopilación, análisis y presentación de datos cualitativos y cuantitativos de Internet con el propósito de entender, optimizar e impulsar la mejora continua del uso de una página web.

Existen esencialmente dos puntos de vista para realizar el análisis web, el punto de vista interno y el externo.

El punto de vista externo (off-site) realiza un análisis del entorno del sitio y el efecto que éste tiene en dicho entorno. Para realizarlo, no es necesario ser el propietario o la persona encargada de mantener el sitio web, pues los datos requeridos, como indica el nombre, son externos al mismo. Su estudio incluye la medición de su audiencia potencial (oportunidad), su participación de voz (visibilidad), y su repercusión (comentarios) en el conjunto de Internet.

Sin embargo, el punto de vista interno (on-site) propone un análisis utilizando la información guardada en el servidor y generada como consecuencia de la utilización del servicio ofertado a los visitantes. La recogida de datos está más orientada al rendimiento y la eficiencia, centrándose en los visitantes como posibles clientes y siendo utilizada para estudios tanto de facilidad de uso como de mercado. Se realiza mediante la medición de rutas de navegación a través del sitio, tiempos de respuesta, mapas de calor de cada página, etc.

4.3 Recopilación de datos

Previo al análisis y presentación de los datos es necesario una extracción y recopilación de la información más relevante de entre toda la disponible. De ello se ocupa una disciplina llamada minería web (Web Mining). Dentro de ella existen diferentes partes que se centran en diversos componentes como la minería de contenido, la de estructura y la de uso.

La minería de contenido web (Web Content Mining) se centra en la recopilación de la información en bruto de la página. Recoge datos sobre el texto de la propia web, no simplemente palabras, sino también etiquetas, y las utiliza para la categorización y clasificación de páginas web.

La minería de estructura web (Web Structure Mining), como su nombre indica, se encarga de la estructura del sitio recogiendo datos con el fin de realizar un mapa web mediante los enlaces entre páginas. También es utilizada para la categorización y clasificación basada en dichos enlaces.

Por último, la minería de uso web (Web Usage Mining) es la parte de minería web cuyos datos de origen se encuentran en los archivos de registro del servidor. Dichos datos son principalmente archivos de texto a modo de bitácora que se recogen cuando los visitantes acceden al servidor. Típicamente se utiliza para el modelado de los visitantes y su comportamiento.

4.3.1 Origen de los datos

Los datos que se recogen para el análisis web pueden provenir de tres lugares: el servidor, el lugar de acceso o el cliente.

El servidor es la fuente de datos que se utiliza más comúnmente. En él se recogen grandes cantidades de información a través de sus archivos de registro y sus bases de datos. Estos suelen contener datos del cliente, como la dirección IP; de la petición, como la fecha y la hora de la solicitud, del propio servidor como su nombre; y otros datos.

El lugar de acceso lo proporciona el proveedor de servicios de Internet (ISP). Estos proveedores habitualmente añaden a su servicio un servidor proxy para que el cliente note una mejora de la velocidad de navegación gracias al almacenamiento en caché de páginas web. La recogida de datos de navegación en ellos es similar a la de cualquier otro servidor, salvo que pueden recoger datos de grupos de visitantes que accedan a grandes grupos de servidores.

El cliente es del que se requiere la información. Sus datos se pueden conseguir mediante el uso de JavaScript, Applets o, incluso, mediante la instalación de software que los envíen directamente al servidor. Estos métodos proporcionan información muy detallada del comportamiento real de los visitantes, sin embargo, se necesita su cooperación y plantean muchas cuestiones relacionadas con las leyes de privacidad.

4.3.2 Fuentes de datos

Para un buen análisis web se han de recoger los datos necesarios de la forma más exacta y precisa posible existiendo diversas fuentes de recogida.

4.3.2.1 Logs del servidor

Los logs del servidor son archivos de registro que recogen información a modo de bitácora en los servidores. Estos los crean y actualizan automáticamente registrando en ellos su actividad.

Requieren de recursos humanos y tecnológicos para ser interpretados. El principal problema a abordar es la identificación de visitantes y visitas. En ellos, se registran con la máxima precisión todos los archivos que se envían y a donde se envían permitiendo saber si estos han sido descargados satisfactoriamente. Sin embargo, cada dirección IP es interpretada como un visitante, aunque detrás de ella pueda haber varios, y no recogen mucha información relevante sobre sus características personales. Además, son sensibles a los robots que utilizan los buscadores web para luego indexar las páginas.

Respecto a las visitas, es necesario poder conocer los caminos que siguieron los visitantes durante la navegación a través del sitio. La tarea es bastante difícil y generalmente depende del tipo de información disponible; además, si el visitante utiliza servidores proxy que tengan páginas en cache, las peticiones pueden no llegar al servidor original, y por tanto, no registrarse en el log.

Muchas herramientas utilizan este sistema de recogida como base para el análisis como [Deep Log Analyzer](#), [AWStats](#), [W3Perl](#), [BBClone](#), [Visitors](#), etc.

4.3.2.2 Cookies

Una cookie es un pequeño conjunto de datos que a través del navegador web, por petición del servidor, se almacena en el disco duro del visitante cuando visita una página web.

Cuando un visitante accede a una página web, el servidor manda una cookie al navegador. Cada vez que el visitante acceda a la misma página web o a otra página del mismo dominio, el navegador devolverá al servidor la cookie enviada por él anteriormente. De esta forma el servidor reconoce al visitante como un visitante anterior.

Aunque las cookies fueron creadas principalmente para permitir almacenar información sobre el estado de la navegación de los visitantes, más tarde, empezaron a usarse para conocer los movimientos que estos realizaban a través de Internet permitiendo realizar históricos de sus visitas. Sin embargo, es imposible identificar las rutas de navegación exactas si el visitante utiliza el botón de retroceso del navegador.

Otra desventaja es que los propios visitantes pueden borrar las cookies almacenadas en su ordenador fácilmente y muchos realizan esta operación con frecuencia. Además, algunos navegadores, programas de seguridad y firewalls pueden bloquearlas y existe una legislación sobre su uso que obliga a informar de él dentro de los sitios web.

No obstante, grandes compañías ofrecen servicios de análisis basados en este sistema como Yahoo y Google. Otros también muy conocidos son [ClickTale](#) y [Woopra](#).

4.3.2.3 Analizadores de paquetes TCP/IP

Además de los registros y las cookies, también se puede seguir el comportamiento de los visitantes en el servidor por medio de analizadores de paquetes TCP/IP. Estos proporcionan algunas ventajas como la recogida de datos en tiempo real o la fusión de la información proveniente de diferentes servidores; pero plantean problemas de escalabilidad en servidores con alto tráfico y no pueden acceder a paquetes encriptados utilizados, por ejemplo, en las transacciones comerciales seguras. Estas limitaciones resultan ser muy graves, sobretodo en cuanto al seguimiento de visitas, por lo que su utilización es escasa.

Uno de los analizadores de paquetes más conocidos es [Wireshark](#) utilizado para realizar análisis y solucionar problemas en redes de comunicaciones, desarrollar software y protocolos, y como herramienta didáctica.

4.3.2.4 Tags

Los Tags son pequeñas porciones de código dentro de las webs que permiten el registro de diferentes eventos dentro del sitio. Se ejecutan en cliente y son muy fáciles y cómodos de implementar, sin embargo, ha de hacerse en todas las páginas del sitio para que sea útil.

La gran mayoría se realizan en código JavaScript, que hoy en día es soportado por prácticamente todos los navegadores, incluso en dispositivos móviles como tabletas y teléfonos; aunque, hay muchos visitantes que tienen desactivado su uso y cada vez proliferan más los programas y complementos que lo deshabilitan para proteger la privacidad.

No son muy precisos, pues, a veces, pueden no llegar a cargarse bien o hacerlo sin que el visitante haya visto la página realmente. Se suelen combinar con las cookies o utilizarse para actualizarlas.

4.3.2.5 Logs de ISP

Los logs de ISP son los archivos de registro que mantienen los proveedores de servicios de Internet y que recogen la información de numerosos usuarios. Gracias a ellos, se puede seguir el rastro de los visitantes fuera del sitio web, proporcionando otras informaciones como desde qué páginas llegan al sitio y a cuales visitan después de estar en él. Y aunque, la reconstrucción de las visitas sigue siendo difícil, si no hay ningún otro almacenamiento entre ellos y sus clientes, esta resulta más sencilla que con los logs de servidor.

Con muy pocos logs, se puede obtener mucha información. Por ejemplo, permiten ver como se utilizan los buscadores, que palabras clave utilizan los visitantes para encontrar el sitio y cuáles de ellas les funcionan mejor a la competencia. Sin embargo, no proporcionan datos demográficos y estos dependen de la configuración que haya realizado el proveedor.

Las herramientas que utilizan este sistema de recogida suelen orientarse a realizar análisis de mercado lo cual las hace realmente útiles para algunas empresas. Dos de las más conocidas son [Compete](#) y [Experian Hitwise](#).

4.3.2.6 Paneles

Un panel es un software que se instala en el ordenador del visitante y que recoge directamente las visitas que éste realiza.

Los paneles tienen numerosas desventajas. Para empezar, necesitan de la colaboración del visitante pues este ha de instalarlo e introducir sus datos. No son muy precisos, ya que la información que recogen está sesgada y puede no ser muy representativa del conjunto general, pues su muestra está restringida a aquellas personas dispuestas a instalárselos. Tardan días en dar resultados e incluso pueden no darlos, ya que han de esperar a que los visitantes que lo tengan instalado entren en el sitio y a que se realice el envío y actualización de la información, lo cual, no suele ser inmediato.

A pesar de sus desventajas, este sistema es el único que permiten obtener información demográfica exacta tanto de los visitantes del sitio como de los de la competencia. Por ello, existen algunas herramientas en el mercado como [Nielsen](#) y [Comscore](#) que se basan en él.

4.3.2.7 Barras de los navegadores

Las barras de los navegadores son complementos software que se instalan en el ordenador del visitante formando parte del navegador.

Aunque comparten con los paneles la desventaja de tener que ser instaladas por los visitantes, estas son más aceptadas que los anteriores debido a que brindan cierta funcionalidad extra, normalmente se ofrecen en la instalación de otros programas y no recogen tantos datos personales. Estas características hacen que no se obtenga tanta información demográfica como con los paneles, pero a cambio, se obtiene una muestra más amplia de datos, aunque sigue ligeramente sesgada con ciertos segmentos de población demasiado representados.

Multitud de servicios web ofrecen su propia barra que utilizan para recoger información de sus visitantes, sobre todo los buscadores y las redes sociales. Una de las compañías más conocidas y reputadas que, entre otras fuentes, utiliza su barra de navegador para hacer sus análisis es [Alexa](#).

4.4 Los datos

Las diversas fuentes de recogida de información contienen una inmensa diversidad y cantidad de datos. Por ejemplo, como se ha comentado anteriormente, en los logs de servidor se pueden encontrar datos sobre la petición, como la fecha y hora de su realización o el tiempo que llevó realizarla; sobre el cliente, como su dirección IP; sobre el servidor, como el nombre del servicio, el nombre del servidor, la dirección IP o el puerto; sobre el envío del servidor al cliente, como el estado o el número de bytes enviados; sobre el envío del cliente al servidor, como el método de respuesta, el archivo requerido, la consulta realizada, el número de bytes recibido, la versión del protocolo, el nombre del host o el contenido de la cookie enviada o recibida; y otros sobre datos remotos o aplicaciones específicas.

Sin embargo, los logs de servidor no permiten la recogida de otros tipos de datos que pueden ser relevantes como la edad de los visitantes, sus aficiones, sus gustos, sus inclinaciones políticas, sindicales o religiosas, etc. que pueden ser obtenidos con los paneles; o las mencionadas páginas desde las que se llega al sitio y las que se visitan después de estar en él, que pueden ser almacenadas por los logs de ISP.

Aun así, los logs de servidor son muy exactos y son una de las fuentes de recogida de información más utilizadas y disponibles. En cuanto a su construcción, obtención y análisis, son la fuente que requiere un menor esfuerzo y coste por parte de los prestadores de servicios. Además, contienen una gran cantidad y diversidad de datos que, bien analizados, pueden revelar información muy relevante sobre multitud de aspectos diferentes. Por todas estas razones, esta será la fuente de datos a utilizar en el desarrollo del proyecto.

4.5 Los datos en los archivos log

Los logs de servidor son una buena fuente de información para el análisis web, ya que pueden contener muchos datos de diferentes tipos. Sin embargo, tal cantidad de información requiere de una herramienta para su tratamiento. Y para la construcción de dicha herramienta es necesario un estudio previo de todos estos tipos de datos y la forma en la que se encuentran almacenados.

En primer lugar hay considerar qué datos son relevantes, ya que dependiendo del objetivo que se persiga unos son más importantes que otros, por lo que es preciso definir qué información es necesaria recoger y para qué interesa. La información no sirve de nada si luego su análisis no conduce a una toma de decisiones.

En segundo lugar, es necesario estudiar cómo se presenta esta información en los logs, de qué tipos disponemos y qué configuraciones nos permiten los diferentes servidores para obtener los datos necesarios.

4.5.1 Contenido

Los logs de servidor recogen los datos de las peticiones que llegan al propio servidor. En ellos pueden almacenarse una gran multitud de estos datos que permite realizar numerosas estadísticas; pero solo algunos son útiles para efectuar las que proporcionan información realmente relevante.

Los datos de una petición se pueden clasificar según su utilidad en esenciales, adicionales o superfluos; y dependiendo de nuestros objetivos y metas, un determinado tipo de datos puede estar dentro de una categoría o de otra.

Los datos esenciales serán aquellos que proporcionan la información mínima e imprescindible para realizar las estadísticas indispensables que ayuden a cumplir el objetivo. Los datos adicionales permiten perfilar algunos resultados y realizar estadísticas adicionales que faciliten la consecución de dicho objetivo. El resto comprenderán el conjunto de los datos superfluos.

En el caso del análisis web que se propone en este trabajo, los datos han de ser escogidos de forma que a la hora de relacionarlos y analizar dichas relaciones se cumpla con el objetivo, que como se comentó, es la mejora continua de la página web. Este objetivo tiene dos metas principales, mejorar la experiencia que el visitante tiene al visitar el sitio y maximizar el beneficio que se obtiene como producto de la prestación del servicio que oferta la web.

Teniendo en cuenta este objetivo y las dos metas a conseguir, se pueden agrupar los datos de una petición presentes en un log de la siguiente forma:

- **Datos esenciales:** fecha y hora de la petición, visitante, página visitada, estado del servidor, bytes transmitidos al cliente y tiempo transcurrido en el envío.
- **Datos adicionales:** puerto de acceso, método de envío, versión del protocolo o entidad (navegador, aplicación o proxy) que realiza la petición.
- **Dato superfluos:** RFC de la petición, sub-estado del servidor, bytes recibidos del cliente, etc.

La fecha y hora de una petición es uno de los datos más importantes; sirve principalmente para analizar la web en segmentos de tiempo concretos, ver tendencias a través del tiempo y discriminar entre datos antiguos y actuales, pudiendo comparar como cambia el comportamiento ante pequeñas variaciones en la web.

El visitante es el centro del análisis ya que está presente en las dos metas que llevan al objetivo principal. Es necesario distinguir que peticiones realiza cada uno descubriendo los diversos perfiles.

La página solicitada en cada petición permite saber que requiere el visitante y como lo busca, por eso es un dato esencial y tanto el estado del servidor, los bytes transmitidos y el tiempo transcurrido proporcionan informaciones técnicas importantes que permiten realizar configuraciones en el servidor que ayudan a la consecución del objetivo.

4.5.2 Estándares de Formato

Los diversos tipos de servidores web permiten recoger distintas clases de datos en los logs de formas diferentes, por lo que cada uno de estos archivos puede tener una estructura o formato distinto.

El análisis web que se pretende realizar con la herramienta propuesta tiene como base estos archivos desde los que se extraen los datos a analizar, por lo que es necesario estudiar los diferentes tipos existentes y conocer su estructura previamente.

La gran mayoría de los servidores permiten configurar qué datos recoger, en qué tipo de log y con qué estructura almacenarlos. A continuación, se revisan las opciones que proporcionan los dos tipos de servidores más extendidos del mercado, IIS y Apache HTTP.

4.5.2.1 IIS

Internet Information Services (IIS) es un conjunto de servicios que se pueden instalar o son parte adicional de algunos sistemas operativos Microsoft Windows. Proporciona la funcionalidad de un servidor web incluyendo la generación de archivos de registro o logs.

Durante la instalación y en la configuración de este paquete de servicios se ponen a disposición del usuario las siguientes opciones para el formato de la recogida de datos:

- **W3C Extended Log File:** es el formato por defecto, está basado en texto ASCII y permite personalizar los campos recogidos para cada sitio web guardando la información en archivos distintos. Dichos campos están separados por espacios y la hora que se registra en ellos es la hora universal coordinada (UTC).
- **W3C Centralized Logging:** este formato es similar al anterior pero recoge toda la información de todos los sitios web en un solo archivo y aunque los campos son configurables son los mismos para todos los sitios.
- **NCSA Common Log File Format:** es el formato de archivo de registro común de National Center for Supercomputing Applications (NCSA). Basado en texto ASCII, es un formato fijo que no permite la personalización de campos y que recoge la información de cada sitio web en archivos distintos. Los campos están separados por comas y la hora que se registra es la hora local teniendo como referencia la hora universal coordinada (Hora UTC).
- **IIS Log File Format:** es el formato de archivo de registro de Microsoft IIS. Basado en texto ASCII, es un formato fijo que no permite personalizar los campos que se registran y que recoge la información de cada sitio web en archivos distintos. Los campos están separados por comas y la hora que se registra es la hora local del servidor.
- **ODBC Logging:** este formato recoge la información de cada sitio web por separado en una base de datos con acceso ODBC que se ha de especificar.
- **Centralized Binary Logging:** este formato binario está diseñado para recoger la máxima información posible de cada sitio web por separado.

Entre todas estas opciones el formato más idóneo para el análisis web es W3C Extended Log File, por dos razones principales. En primer lugar incluye la información de cada sitio web en un archivo diferente, lo que permite realizar análisis individualizados de cada uno de los sitios. Y en segundo lugar, es configurable, lo que permite recoger todos los datos que sean necesarios.

Además, W3C Extended Log File es un formato estandarizado. Los archivos en este formato cuentan con una cabecera que permite identificar como estarán estructurados los datos en las peticiones que registran. Esta cabecera contiene los nombres de los tipos de datos y el orden en el que se encuentran. Por convenio, los nombres de los datos están formados por un prefijo que hace referencia al lugar con el que está relacionado el dato y un identificador que representa el dato. El prefijo puede ser uno de los siguientes:

- **c-:** si denotan una característica del cliente,
- **s-:** si denotan una característica del servidor,
- **r-:** si son datos remotos,
- **cs-:** si se refieren a un dato que parten del cliente y llega al servidor,
- **sc-:** si se refieren a un dato que parte del servidor y llega al cliente,
- **rs-:** si se refieren a un dato que parten de un servidor remoto y llega al servidor,
- **sr-:** si se refieren a un dato que parten del servidor un y llega un servidor remoto,
- **x-:** siendo el identificador de una aplicación específica.

Por las razones expuestas y ya que se trata de un formato estandarizado, para este trabajo, se tendrán en cuenta dos estilos de formato W3C Extended Log File que puede generar este tipo de servidor.

Las peticiones del primero de ellos, que a la hora de referirnos a él será llamado IIS Mínimo, contendrán los datos esenciales y adicionales con la siguiente estructura:

```
s-date s-time s-ip cs-method cs-uri-stem s-port cs-username c-ip cs-version cs(User-Agent) sc-status sc-substatus sc-bytes cs-bytes time-taken
```

Las peticiones del segundo, que a partir de aquí será llamado IIS Extendido, además de los anteriores datos, contendrán algunos superfluos con la siguiente estructura:

```
s-date s-time s-sitename s-computername s-ip cs-method cs-uri-stem cs-uri-query s-port cs-username c-ip cs-version cs(User-Agent) cs(Cookie) cs(Referer) cs-host sc-status sc-substatus sc-win32-status sc-bytes cs-bytes time-taken
```

4.5.2.2 Apache HTTP Server

Apache HTTP Server es un servidor web de código abierto que se puede añadir como servicio a plataformas Microsoft Windows, Unix o Macintosh. Inicialmente permite la recogida de archivos de registro o logs mediante uno de los dos formatos siguientes:

- **Common Log Format:** es el formato por defecto, está basado en texto ASCII y contienen información muy básica. Corresponde con la opción NCSA Common Log File Format de IIS.
- **Combined Log Format:** este formato es similar al anterior pero recoge dos datos más: la entidad que realiza la petición y el campo cookie.

Ninguno de estos dos formatos contiene los datos descritos como esenciales para el análisis web. Sin embargo, tras la instalación se puede configurar el servidor para que proporcione archivos donde se registren.

Por ello, para este trabajo se tendrán en cuenta dos formatos que tienen como base los dos que puede generar este tipo de servidor.

El primero de ellos tiene como base el Common Log Format y a sus peticiones se le añade el tiempo transcurrido en el envío para que contengan todos los datos esenciales y alguno superfluo. A la hora de referirnos a él será llamado Apache Common y, expresada con la nomenclatura W3C, la estructura de sus peticiones es la siguiente:

```
c-ip c-RFC c-user-id [s-date s-time s-timezone] "cs-method cs-uri-stem cs-version" sc-status sc-bytes time-taken
```

El segundo de ellos tiene como base el Combined Log Format al que también se le añade en sus peticiones el tiempo transcurrido en el envío para que contenga todos los datos esenciales, adicionales y alguno superfluo. A la hora de referirnos a él será llamado Apache Combined y, expresada con la nomenclatura W3C, la estructura de sus peticiones es la siguiente:

```
c-ip c-RFC c-user-id [s-date s-time s-timezone] "cs-method cs-uri-stem cs-version" sc-status sc-bytes "cs_referer" "cs_agent" time-taken
```

4.6 Conceptos derivados de los datos

Procesando los datos que se encuentran en un archivo log junto con algunos otros, como el nombre del proyecto o la organización a la que pertenece el sitio, se pueden definir una serie de conceptos derivados que permiten realizar el análisis de la web. A continuación se expondrán dichos conceptos junto con sus propiedades y algunas consideraciones a tener en cuenta a la hora de extraerlos de un archivo log.

4.6.1 Proyecto web

Un proyecto web es un conjunto de páginas web que ofrecen un servicio proporcionado por una organización y que se alojan en un servidor web. Este conjunto de páginas no se ha de entender como un sitio web estático o puntual cuyos accesos se analizan sin más, si no que dicho análisis debe aportar datos que permitan realizar pequeños cambios dirigidos a una mejor prestación del servicio y que mantienen el proyecto vivo a lo largo del tiempo.

En este contexto, un proyecto web tiene esencialmente las siguientes características:

- **Nombre:** identifica al proyecto y lo diferencia de los demás posibles proyectos que se estén realizando y analizando para un mismo cliente.
- **Organización:** nombre de la entidad, empresa o cliente para el que se desarrolla el proyecto.
- **Descripción:** definición breve del proyecto que describe su objetivo y que es posible que incluya otros datos relevantes y que ayuden al usuario a identificarlo y distinguirlo de los demás.

Estas características permanecen constantes durante todo el desarrollo del proyecto y su análisis. Pueden sufrir ligeros cambios, como el nombre de la organización, mientras que la finalidad de la web no se vea comprometida, lo que repercutiría sustancialmente en las posibles estadísticas llegando a resultar inservibles para su análisis.

4.6.2 Log

Como se ha mencionado, un log de servidor es un archivo que registra automáticamente a modo de bitácora la actividad relativa a un sitio web alojado en un servidor. Por lo tanto, cada uno de estos archivos pertenece a un único proyecto web.

Estos archivos de datos están compuestos por un conjunto de peticiones, o hits, que reflejan en estricto orden la mencionada actividad del servidor producida por los visitantes entre dos fechas determinadas. Además tienen las siguientes propiedades:

- **Nombre:** identifica al archivo y lo diferencia de los demás posibles archivos que registran la actividad de un determinado proyecto.
- **Proyecto:** proyecto web al que pertenece el archivo.
- **Tipo:** tipo de log al que pertenece el archivo de datos.
- **Servidor:** servidor web en el que se ha registrado el archivo.
- **Fecha:** fecha en la que el log es registrado como un archivo de datos del proyecto.
- **Fecha de inicio:** fecha de la primera petición, o hit, que compone el archivo.
- **Fecha de fin:** fecha de la última petición, o hit, que compone el archivo.
- **Número de visitas:** número de vistas activas en el sitio web durante el periodo de actividad recogido por el archivo de datos.
- **Número de hits:** número de peticiones que se ha producido en el periodo de actividad recogido por el archivo de datos.

Es posible que dentro de un servidor se recojan al mismo tiempo diferentes logs que almacenen la actividad de diferentes tipos de peticiones o de diferentes elementos accedidos. Por lo que, aunque no es común, puede ocurrir que varios logs se refieran a los mismos tiempos de actividad o a tiempos de actividad solapados. En estos casos, las peticiones a cada elemento en cada instante por cada visitante son únicas y las estadísticas que se puedan realizar sobre ellas deben considerarlas de esta forma para que reflejen la verdadera actividad acontecida.

4.6.3 Visita

Una visita es un conjunto de hits realizados a un proyecto web por un único visitante durante un periodo de actividad continuado.

Se considera que una visita comienza con el primer hit que realiza el visitante y continua activa mientras el periodo entre una petición y la siguiente sea menor a 30 minutos. Esto permite que una misma visita pueda estar compuesta por hits almacenados en logs cuyo registro de tiempo de actividad sea consecutivo, lo que, en la práctica, dificultaría el cálculo de las mismas. Por ello, aunque un log no se considera un conjunto de visitas, se determina que la primera visita de un visitante comienza en el primer acceso de dicho visitante registrado en el archivo y su última visita termina con el último hit incluido en el log.

Teniendo esto en cuenta, las visitas se caracterizan por:

- **Visitante:** identifica a quién ha realizado la visita.
- **Proyecto:** proyecto web que ha sido visitado.
- **Fecha de inicio:** fecha de la primera petición, o hit, que compone la visita.
- **Fecha de fin:** fecha de la última petición, o hit, que compone la visita.
- **Página de entrada:** nombre de la primera página accedida en la visita
- **Página de salida:** nombre de la última página accedida en la visita.

Por último, cabe destacar que una visita puede contener diferentes tipos de peticiones y que tanto la página de entrada como la de salida serán páginas web sólo si en dicha visita existen páginas web que hayan sido accedidas, en caso contrario podrán ser archivos de cualquier otro tipo.

4.6.4 Hit

Un hit es una petición de acceso a un archivo del sitio web que compone el proyecto web y que ha sido registrado en un log formando parte de dicho archivo de datos.

Como se ha visto anteriormente un hit pueden tener multitud de propiedades. Sin embargo, como se explicó, para la correcta realización de las estadísticas se ha considerado que obligatoriamente, y como mínimo, ha de tener las siguientes:

- **Proyecto:** identifica el proyecto web al que se ha realizado la petición.
- **Fecha y Hora:** fecha y la hora a la que se ha realizado el hit.
- **Visitante:** identifica quién ha realizado la petición.
- **Página de origen:** nombre de la página visitada anteriormente al hit.

- **Página de destino:** nombre del archivo accedido con el hit.
- **Estado:** código de estado que el servidor envía al visitante que ha realizado la petición de acceso según el protocolo de transferencia de hipertexto (HTTP).
- **Servidor:** identifica al servidor al que se ha enviado la petición.
- **Bytes transmitidos:** número de bytes que el servidor ha enviado al cliente como consecuencia de la petición de acceso al archivo.
- **Tiempo:** segundos transcurridos para el envío del archivo de la petición de acceso.

Respecto a estas propiedades, hay que mencionar que la página de origen de todos los hits no se puede conocer hasta el acceso a la primera página web del sitio, ya que no se registra en el log, por lo que, a efectos prácticos, se considera una página común que representa el mundo exterior al sitio web del proyecto.

Adicionalmente, los hits, tienen las siguientes propiedades:

- **Puerto:** número de puerto al que se ha realizado la petición.
- **Método de envío:** forma de envío con el que se ha pedido el archivo.
- **Versión de Protocolo:** protocolo y versión de la transferencia con la que se realiza el acceso.
- **Agente:** entidad (navegador, aplicación o proxy) utilizada para realizar la petición.
- **Referencia:** página principal o sitio web que contiene el archivo demandado.

Y, aunque pertenecen a las características de un hit, los archivos de datos solo ocasionalmente registran las siguientes propiedades:

- **Cadena de consulta:** parte del URI del archivo accedido que contiene los valores para construir dinámicamente el objeto pedido.
- **Host:** dominio o sitio web que contiene el archivo demandado.
- **Sub-Estado:** código de sub-estado que el servidor envía al visitante que ha realizado la petición de acceso.
- **Estado Win32:** código de estado de Windows que el servidor envía al visitante que ha realizado la petición de acceso.
- **Cookie:** conjunto de datos asociados a la petición.
- **RFC:** documento donde se encuentra la especificación de la norma de la petición.
- **Bytes recibidos:** número de bytes que forman la petición de acceso.

En cuanto a los hits, hay que mencionar que el registro de la una petición sobre un archivo a una hora en una fecha por un visitante sobre un proyecto es único, así que se consideran solo una vez las peticiones con idénticos valores para todas estas propiedades.

4.6.5 Visitante

Un visitante es un usuario del servicio que proporciona el sitio web y que realiza al menos una petición de acceso a uno de los archivos del mismo, reflejando así el hit sobre el archivo log y generando una visita al proyecto.

No se recogen muchas propiedades de él directamente en el log, pero gran parte del análisis que se realiza esta dedicado a conocer el valor de sus características. Los únicos datos que se podrían conocer a priori son:

- **Dirección IP:** etiqueta que identifica la interfaz del visitante.
- **Nombre:** identifica al visitante que ha ingresado en el sitio web.

En la práctica, los diferentes tipos de servidores web, por defecto, no recogen el nombre del visitante, ni si quiera las raras veces en las que éste está disponible, ya que su registro en un log es ineficiente. Por ello, como se ha mencionado, se considera que cada dirección IP es un solo visitante aunque detrás de ella pueda haber varios.

Por otro lado, también se presupone que dichos visitantes se encuentran en un determinado lugar. Así que, aunque no sea una información proporcionada por el propio log, a posteriori, basadas en el valor de la IP, se pueden conocer las siguientes características geográficas:

- **Ciudad:** nombre de la población desde la que el visitante ha ingresado en el sitio.
- **Región:** nombre de la región a la que pertenece la población anterior.
- **País:** nombre del país al que pertenece la región anterior.

Aunque cabe destacar que estos últimos datos pueden no estar disponibles, ser difíciles de averiguar o no resultar exactos, puesto que una dirección IP indica el lugar desde el que se realiza la petición que no necesariamente coincide con el origen de la misma ya que este, por ejemplo, puede ser detrás de un servidor proxy.

4.6.6 Servidor

El servidor es el lugar en el que se encuentra alojado el sitio web, al que va dirigido la petición de acceso de un visitante y en el que se escribe el log en el que quedan registrados los hits.

Al ser el servidor el que proporciona el archivo con los datos a analizar, en cada uno de estos archivos el servidor es único y posee las siguientes propiedades:

- **Dirección IP:** etiqueta que identifica la interfaz del servidor.
- **Nombre:** secuencia de caracteres que identifica al servidor.
- **Proyecto:** identifica el proyecto que está alojado.

Cabe mencionar que en este trabajo no se tendrán en cuenta sistemas distribuidos en los que la actividad sobre un proyecto podría estar repartida entre varios servidores a la vez.

4.6.7 Página

Una página es un documento o archivo con código interpretable por un servidor que devuelve un objeto generalmente HTML o XHTML que puede ser enviado utilizando el protocolo de transferencia de hipertexto a través de la red e interpretado por un navegador.

Las páginas tienen las siguientes características

- **Nombre:** identifica a la página dentro de un proyecto.
- **Proyecto:** identifica el proyecto web al que pertenece la página.

Parte de los resultados del análisis web van dirigidos a realizar cambios tanto en el conjunto de páginas que forman el proyecto web como en la estructura de su contenido.

4.7 Presentación de los resultados

Como resultado de la recolección y el procesado de los datos anteriores se obtiene una estadística. Según la RAE, una estadística es un conjunto de datos cuantitativos relacionados.

Las estadísticas pueden describir las características de los datos o poner de manifiesto asociaciones entre datos que no están directamente relacionados o que no son directamente visibles tal y como se obtienen. Las primeras se las denomina estadísticas descriptivas y a las segundas estadísticas inferenciales. Pero ambos tipos tienen que tener un sentido con respecto al significado intrínseco de los datos sobre los que se realizan y ser elegidas para una finalidad concreta, si no, su análisis no conllevaría ninguna conclusión lógica.

Por tanto, para realizar una estadística es necesario un conjunto de datos y una finalidad. Ambas cosas permiten realizar otro tipo de clasificación.

4.7.1.1 Estadísticas según su finalidad:

- **Medir la capacidad del servidor:** dirigidas al administrador del servidor para que mantenga un acceso rápido al servicio y un tiempo de respuesta bajo con un número de personas acorde a su demanda. Las estadísticas de este tipo permiten, entre otras cosas, comprobar el comportamiento en picos de carga o realizar simulaciones ante la previsión de un aumento de la demanda.
- **Diseñar el sitio web:** dirigidas al diseñador web para conseguir un acceso eficiente a la información por parte de los visitantes. Las estadísticas de este tipo permiten, entre otras cosas, acortar las visitas gracias a la redistribución de la información de manera que disminuya la longitud del camino recorrido para el acceso al servicio deseado por los visitantes.
- **Marketing del servicio:** dirigidas al departamento comercial para conseguir un acceso rentable al servicio de una manera satisfactoria por parte del cliente. Las estadísticas de este tipo permiten, entre otras cosas, realizar estudios de mercado y así ofrecer un servicio adecuado al tipo de personas que lo demanda y que reporte mayores ganancias para la empresa en menor tiempo.

4.7.1.2 Estadísticas según el conjunto de datos:

- **Temporales:** sobre datos recogidos a lo largo del tiempo cuyo significado intrínseco se relaciona con entidades temporales (horas, días, semanas, etc.). Los logs de servidor permiten realizar estas estadísticas adecuadamente ya que proporcionan la fecha y hora de todos los eventos.

- **Geográficas:** sobre datos cuyo significado intrínseco se relaciona con entidades espaciales (ciudades, regiones, países, etc.). Este tipo de estadísticas no se pueden realizar directamente con los datos de los logs de servidor. Entre estos solo se encuentra la dirección IP desde la que se realiza la petición de acceso, pero no el lugar desde donde se efectúa, por lo que es necesario ayudarse de tablas que relacionen dicha dirección IP con alguna entidad espacial, las cuales pueden no ser muy precisas.
- **Caracterización de Carga:** sobre datos cuyo significado intrínseco se relaciona con parámetros del servidor o del servicio (tiempo de acceso, número de accesos, bytes transmitidos, tiempo de respuesta, etc.). Los datos que proporcionan los logs de servidor son idóneos para realizar estas estadísticas ya que es el propio servidor el que registra los necesarios directamente. Las estadísticas de este tipo pueden ser meramente descriptivas o ayudar a desarrollar distintos modelos de carga según su finalidad.
- **Perfil de Navegación:** sobre datos cuyo significado intrínseco se relaciona con el recorrido del visitante a través del sitio web (páginas de entradas, páginas de salida, probabilidades de acceso a páginas, etc.). Este tipo de estadísticas pueden realizarse con los datos proporcionados por los logs de servidor. Sin embargo, no resultan del todo precisas debido a las consideraciones que hay que realizar, explicadas anteriormente, sobre los visitantes, las direcciones IP y las visitas. En este grupo de estadísticas cobra una especial relevancia las que proporcionan información sobre la primera y última página visitada en cada visita por los diferentes visitantes o las que permiten saber cuál es la probabilidad de transición entre una determinada página y las demás.

Capítulo 5 - Requisitos

5.1 Visión General

El sistema consistirá en una herramienta interactiva que permita realizar estudios del rendimiento y la planificación de sitios web. Estará dirigido a administradores encargados del mantenimiento de proyectos de mejora de dichos sitios.

Para ello, la herramienta posibilitará a un administrador el describir los diferentes proyectos en los que esté involucrado de una manera simple, y facilitará la introducción de datos en cada uno de ellos mediante el procesado de archivos log de servidor en los formatos considerados en el apartado 4.5.2.

Gracias al procesado de datos, el sistema proporcionará al administrador diversas estadísticas, así como los datos derivados de las mismas, que faciliten la realización de los diferentes análisis web de cada uno de los sitios.

5.2 Requisitos funcionales

RF-001 Añadir Proyectos Web

Descripción: El sistema debe permitir dar de alta nuevos proyectos web.

Justificación: Es parte del propósito principal del sistema.

RF-002 Modificar Proyectos Web

Descripción: El sistema debe permitir modificar las características de un proyecto web anteriormente introducido en él.

Justificación: Las características de un proyecto web pueden cambiar con el tiempo.

Dependencia: RF-001

RF-003 Eliminar Proyectos Web

Descripción: El sistema debe permitir eliminar proyectos web previamente introducidos en él.

Justificación: El usuario puede desear abandonar un proyecto web o la gestión y administración del mismo.

Dependencia: RF-001

RF-004 Mostrar Estadísticas

Descripción: El sistema debe mostrar diferentes estadísticas referentes a los accesos a la web de un determinado proyecto.

Justificación: Es parte del propósito general del sistema.

Dependencia: RF-001

RF-005 Introducir Datos Web

Descripción: El sistema debe permitir introducir información sobre los accesos a la web de un determinado proyecto desde archivos log.

Justificación: Es necesario para que el sistema pueda realizar estadísticas basadas en dichos accesos.

Dependencia: RF-001, RF-004

RF-006 Eliminar Datos Web

Descripción: El sistema debe permitir eliminar información sobre los accesos a la web de un determinado proyecto.

Justificación: El usuario puede equivocarse al introducir datos en un proyecto, considerar que algunos datos son obsoletos o necesitar espacio para nuevos datos.

Dependencia: RF-001, RF-005

RF-007 Mostrar Estadísticas Temporales

Descripción: El sistema debe mostrar estadísticas basadas en los accesos a la web que se hayan introducido en un determinado proyecto, entre dos fechas señaladas por el usuario y según las siguientes agrupaciones de tiempo: horas, días de la semana, días del mes, meses y años.

Justificación: Este tipo de estadísticas son de mucha utilidad para el usuario.

Dependencia: RF-001, RF-004, RF-005

RF-008 Mostrar Estadísticas Geográficas

Descripción: El sistema debe mostrar estadísticas basadas en los accesos a la web que se hayan introducido en un determinado proyecto, entre dos fechas señaladas por el usuario y según las siguientes agrupaciones geográficas: ciudad, región y país.

Justificación: Este tipo de estadísticas son de mucha utilidad para el usuario.

Dependencia: RF-001, RF-004, RF-005

RF-009 Mostrar Estadísticas de Entradas y Salidas

Descripción: El sistema debe mostrar estadísticas, basadas en los accesos que se hayan introducido en un determinado proyecto, sobre la primera página visitada y la última entre dos fechas señaladas por el usuario.

Justificación: Este tipo de estadísticas son de mucha utilidad para el usuario.

Dependencia: RF-001, RF-004, RF-005

RF-010 Mostrar Estadísticas de Perfil de Navegación

Descripción: El sistema debe mostrar estadísticas, basadas en los accesos que se hayan introducido en un determinado proyecto, sobre los porcentajes de transición entre páginas entre dos fechas señaladas por el usuario.

Justificación: Este tipo de estadísticas son de mucha utilidad para el usuario.

Dependencia: RF-001, RF-004, RF-005

RF-011 Mostrar Estadísticas de Carga Descriptiva

Descripción: El sistema debe mostrar estadísticas descriptivas, basadas en los datos que se hayan introducido en un determinado proyecto, sobre el número de accesos, el tamaño y el tiempo de cada página entre dos fechas señaladas por el usuario.

Justificación: Este tipo de estadísticas son de mucha utilidad para el usuario.

Dependencia: RF-001, RF-004, RF-005

RF-012 Mostrar Estadísticas de Modelo de Carga

Descripción: El sistema debe mostrar modelos de carga, basados en los datos que se hayan introducido en un determinado proyecto pertenecientes a un intervalo entre dos fechas señaladas por el usuario; según una agrupación apoyada en el número de accesos, el tamaño y el tiempo de carga de cada página, calculados utilizando un método, una métrica y una distancia de corte indicados por el usuario.

Justificación: Este tipo de estadísticas son de mucha utilidad para el usuario.

Dependencia: RF-001, RF-004, RF-005

RF-013 Proporcionar Datos estadísticos

Descripción: El sistema por cada estadística que muestre debe ofrecer al usuario todos los datos derivados de la misma, incluidos los que no muestre directamente.

Justificación: Es parte del propósito principal del sistema proporcionar datos al usuario para que este pueda analizarlos en profundidad.

Dependencia: RF-001, RF-004, RF-005

RF-014 Proporcionar información sobre las estadísticas mostradas

Descripción: Si la estadística mostrada por el sistema proporciona información sobre una única parte de los datos, en vez de su totalidad, deberá informar sobre el porcentaje de datos que está considerando. Igualmente deberá comunicar las posibles opciones tenidas en cuenta o si se ha considerado algún tipo de sesgo.

Justificación: Es necesario para la correcta interpretación de los resultados.

Dependencia: RF-001, RF-004, RF-005

5.3 Requisitos no funcionales

5.3.1 Ubicación y Portabilidad

RNF-001 Cliente-Servidor de dos niveles

Descripción: La herramienta debe poder estar separada de la información sobre los proyectos y accesos a las diferentes webs.

Justificación: Eficiencia, movilidad, fiabilidad y seguridad del sistema.

5.3.2 Facilidad de uso

RNF-002 Acceso remoto

Descripción: Poder acceder a la herramienta remotamente desde cualquier dispositivo mediante una interfaz de usuario basada en web.

Justificación: Accesibilidad del sistema.

5.3.3 Datos

RNF-003 Formato Archivos Log

Descripción: Los archivos log desde los que se recogen los datos de acceso a los proyectos web deben estar en alguno de los formatos tenidos en cuenta para este trabajo en el punto 4.5.2.

Justificación: Son los formatos más extendidos con los datos necesarios para casi todo tipo de estadísticas.

Dependencia: RF-005

RNF-004 Datos de Salida

Descripción: Los datos derivados de las estadísticas que se deben proporcionar se darán en un formato estandarizado y acorde con su significado, como archivos .CSV o .DOT.

Justificación: Se debe favorecer la utilización de los resultados por otros sistemas.

Dependencia: RF-013

5.3.4 Documentación

RNF-005 Manual de usuario

Descripción: Con la herramienta se debe hacer entrega de un manual de usuario.

Justificación: Ayuda al usuario a familiarizarse con el sistema.

RNF-006 Manual de configuración e instalación

Descripción: Con la herramienta se debe hacer entrega de un manual de configuración e instalación.

Justificación: El usuario puede querer reinstalar, reconfigurar o instalar y configurar la herramienta en otro lugar y requiere saber cómo hacerlo.

5.3.5 Restricciones

RNF-007 Plataforma Apache

Descripción: La herramienta debe desplegarse sobre un servidor HTTP Apache.

Justificación: Es la plataforma que posee el usuario en sus instalaciones.

RNF-008 Lenguaje Python

Descripción: La base de la herramienta debe programarse en lenguaje Python.

Justificación: Algunas funcionalidades y posibles ampliaciones de la herramienta necesitan de un lenguaje potente que realice fácilmente cálculos matemáticos complejos.

Capítulo 6 - Análisis

6.1 Actores y Roles

Durante la descripción del problema a resolver se han nombrado muchos tipos de personas: visitante, administrador del servicio, diseñador web, comerciales... Y como consecuencia de la solución pueden verse beneficiados muchos otros: clientes, inversores, promotores... Sin embargo, aunque los resultados de las estadísticas pueden ir dirigidos a analistas comerciales, departamento de marketing, analistas de sistemas...; para el uso de la solución, la única figura realmente presente, en contacto con en el problema y, por tanto, el actor, es aquel que juega el único rol que se podría denominar como pre-analista.

El pre-analista es una persona encargada de realizar la recopilación y discriminación de datos aportando las primeras estadísticas resultado de un pre-análisis individual de cada uno de los proyectos web que tiene encomendados. Es el responsable de retroalimentar a todos los posibles analistas que deben tomar decisiones tanto de la estructura funcional de la web como de su estructura física en el servidor dentro del proyecto de mejora de la web.

En proyectos pequeños este rol lo puede jugar el propio administrador web que es el mismo que deberá tomar esas decisiones y aplicarlas en busca de su objetivo, mejorar la web, lo que proveerá a los visitantes de una mejor experiencia y conllevará grandes beneficios para sus clientes.

6.2 Casos de Uso

Tras analizar los requisitos funcionales de la aplicación se han identificado los casos de uso que se representan en la Ilustración 6-1 y que se detallan a continuación.

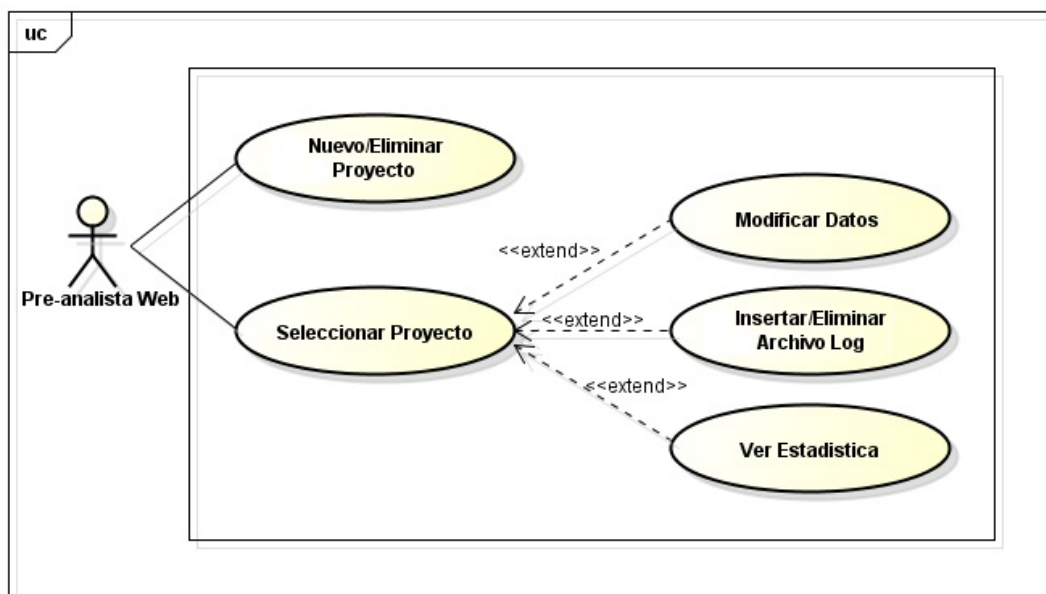


Ilustración 6-1 Casos de Uso

6.2.1 Nuevo Proyecto

- Actor: Pre-analista web
- Código: CU-001
- Inicio: Cuando el pre-analista desea crear un nuevo proyecto web en el sistema.
- Precondición:
 - El pre-analista web ha iniciado el sistema.

- Flujo de eventos:
 1. El pre-analista web selecciona crear un nuevo proyecto.
 2. El sistema pide los datos del nuevo proyecto al pre-analista web.
 3. El pre-analista web introduce los datos del nuevo proyecto.
 4. El sistema recoge y verifica los datos del proyecto.
 - 4.1. Si los datos son correctos, el sistema pasa al paso 5.
 - 4.2. Si los datos no son correctos, el sistema pasa al paso 2.
 5. El sistema guarda los datos del nuevo proyecto.
 6. Fin del caso de uso.
- Postcondición:
 - Se ha creado un nuevo proyecto web.
 - El proyecto web creado queda seleccionado.
- Requisitos Relacionados: RF-001

6.2.2 Eliminar Proyecto

- Actor: Pre-analista web
- Código: CU-002
- Inicio: Cuando el pre-analista web desea eliminar un proyecto web del sistema.
- Precondición:
 - El pre-analista web ha iniciado el sistema.
 - En el sistema hay al menos un proyecto web eliminable.
- Flujo de eventos:
 1. El pre-analista web selecciona eliminar proyecto.
 2. El sistema muestra los diferentes proyectos webs eliminables.
 3. El pre-analista web escoge uno o más proyectos.
 4. El sistema elimina toda la información del proyecto o los proyectos elegidos.
 5. Fin del caso de uso.
- Postcondición:
 - Los proyectos web escogidos y toda la información recogida sobre ellos queda eliminada del sistema.
- Requisitos Relacionados: RF-003

6.2.3 Seleccionar Proyecto

- Actor: Pre-analista web
- Código: CU-003
- Inicio: Cuando el usuario desea seleccionar un proyecto web del sistema.
- Precondición:
 - El pre-analista web ha iniciado el sistema.
 - En el sistema hay al menos un proyecto web seleccionable.
- Flujo de eventos:
 1. El pre-analista web selecciona abrir proyecto.
 2. El sistema muestra los diferentes proyectos webs seleccionables.
 3. El pre-analista web escoge un proyecto.

- 4. El sistema recoge los datos del proyecto elegido.
- 5. Fin del caso de uso.
- Postcondición:
 - El proyecto web escogido queda seleccionado.
- Requisitos Relacionados: RF-002, RF-004, RF-005, RF-006.

6.2.4 Modificar Proyecto

- Actor: Pre-analista web
- Código: CU-004
- Inicio: Cuando el pre-analista web desea modificar los datos de un proyecto web.
- Extiende: Seleccionar Proyecto.
- Precondición:
 - En el sistema hay un proyecto web seleccionado.
- Flujo de eventos:
 1. El pre-analista web selecciona modificar datos del proyecto.
 2. El sistema muestra los datos del proyecto web.
 3. El pre-analista web modifica los datos del proyecto.
 4. El sistema recoge y verifica los nuevos datos del proyecto web.
 - 4.1. Si el pre-analista decide cancelar la modificación, el sistema pasa al paso 6.
 - 4.2. Si los datos son correctos, el sistema pasa al paso 5.
 - 4.3. Si los datos no son correctos, pasa al paso 2.
 5. El sistema actualiza los datos del proyecto web.
 6. Fin del caso de uso.
- Postcondición:
 - El proyecto web seleccionado tiene los datos deseados.
- Requisitos Relacionados: RF-002

6.2.5 Insertar Archivo Log

- Actor: Pre-analista web
- Código: CU-005.
- Inicio: Cuando el pre-analista web desea insertar un log con información sobre accesos a la web.
- Extiende: Seleccionar Proyecto.
- Precondición:
 - Hay un proyecto web seleccionado en el sistema.
- Flujo de eventos:
 1. El pre-analista web selecciona insertar log.
 2. El sistema pide al pre-analista web el log con los accesos a la web.
 3. El pre-analista web introduce el log.
 4. El sistema recoge y verifica el log y los datos de los accesos que contiene.
 - 4.1. Si el log y sus datos son correctos, el sistema pasa al paso 5.
 - 4.2. Si el log o sus datos no son correctos, el sistema pasa al paso 2.
 5. El sistema guarda los datos que no se hubiesen recogido anteriormente.

6. Fin del caso de uso.

- Postcondición:
 - El proyecto seleccionado puede tener asociado un nuevo log con nuevos datos de acceso.
- Requisitos Relacionados: RF-005

6.2.6 Eliminar Archivo Log

- Actor: Pre-analista web
- Código: CU-006
- Inicio: Cuando el pre-analista desea eliminar un log y sus datos de un proyecto.
- Extiende: Seleccionar Proyecto.
- Precondición:
 - Hay un proyecto web seleccionado en el sistema.
 - El proyecto web seleccionado tiene asociados logs y datos.
- Flujo de eventos:
 1. El pre-analista web selecciona eliminar log.
 2. El sistema muestra los logs eliminables asociados al proyecto.
 3. El pre-analista web escoge uno o más logs asociados al proyecto.
 4. El sistema elimina todos los logs escogidos y sus datos asociados.
 5. Fin del caso de uso.
- Postcondición:
 - Los logs escogidos y su información asociada quedan eliminados del sistema.
- Requisitos Relacionados: RF-006

6.2.7 Ver Estadística

- Actor: Pre-analista web
- Código: CU-006
- Inicio: Cuando el pre-analista web desea ver una estadística.
- Extiende: Seleccionar Proyecto.
- Precondición:
 - Hay un proyecto web seleccionado en el sistema.
- Flujo de eventos:
 1. El pre-analista web selecciona ver estadística.
 2. El sistema muestra las diferentes estadísticas disponibles.
 3. El pre-analista web elige la estadística.
 4. El sistema muestra las opciones de la estadística.
 5. El pre-analista web selecciona las opciones de la estadística.
 6. El sistema calcula los datos pertinentes y muestra la estadística.
 7. Fin del caso de uso.
- Postcondición:
 - Se ha mostrado una estadística al pre-analista web.
- Requisitos Relacionados: RF-004

6.3 Modelo Estático

El modelo estático muestra la estructura estática de los objetos que intervienen en el análisis web y la organiza en segmentos manejables describiendo dichos objetos y sus relaciones entre sí. Lo más importante dentro del modelado de estos objetos es su descripción al más alto nivel dentro del sistema como entidades conectadas mediante asociaciones.

La información del modelo estático proviene de la definición del problema, del conocimiento de expertos acerca del dominio de la aplicación y del conocimiento general del análisis web. Todo esto, es lo que se detalló anteriormente en el Capítulo 5.

El primer paso para construir el modelo estático es analizar los conceptos más importantes de análisis web para identificar entidades y relaciones, y construir una primera representación llamada diagrama entidad-relación. Para resolver la herramienta que se propone realizar en el trabajo, el diagrama entidad-relación es el que se puede observar en la Ilustración 6-2.

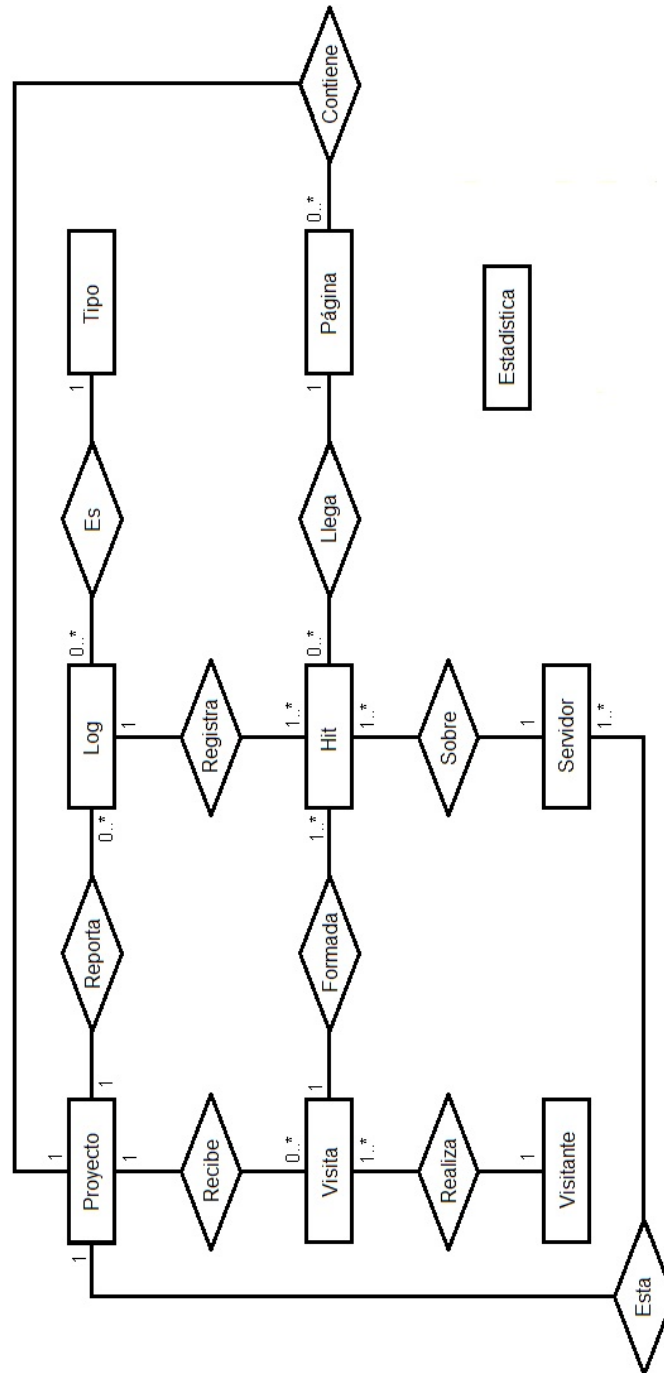


Ilustración 6-2 Diagrama Entidad-Relación

En dicha Ilustración 6-2 se puede observar claramente el modelado de los componentes principales necesarios para el análisis web que se derivan directamente de la descripción dada.

A continuación, utilizando como base el diagrama entidad-relación y gracias al análisis de requisitos y de los casos de uso, se pueden identificar objetos que darán lugar a diversas clases y asociaciones que conformarán un nuevo diagrama.

Posteriormente, este nuevo diagrama junto con el análisis de la dinámica de la herramienta, realizado en el siguiente apartado, será completado con una serie de atributos y operaciones para dichas clases que más tarde, en la etapa de diseño, se irán refinando para llegar a la solución.

El diagrama completo correspondiente, llamado diagrama de clases, se puede ver en la Ilustración 6-3.

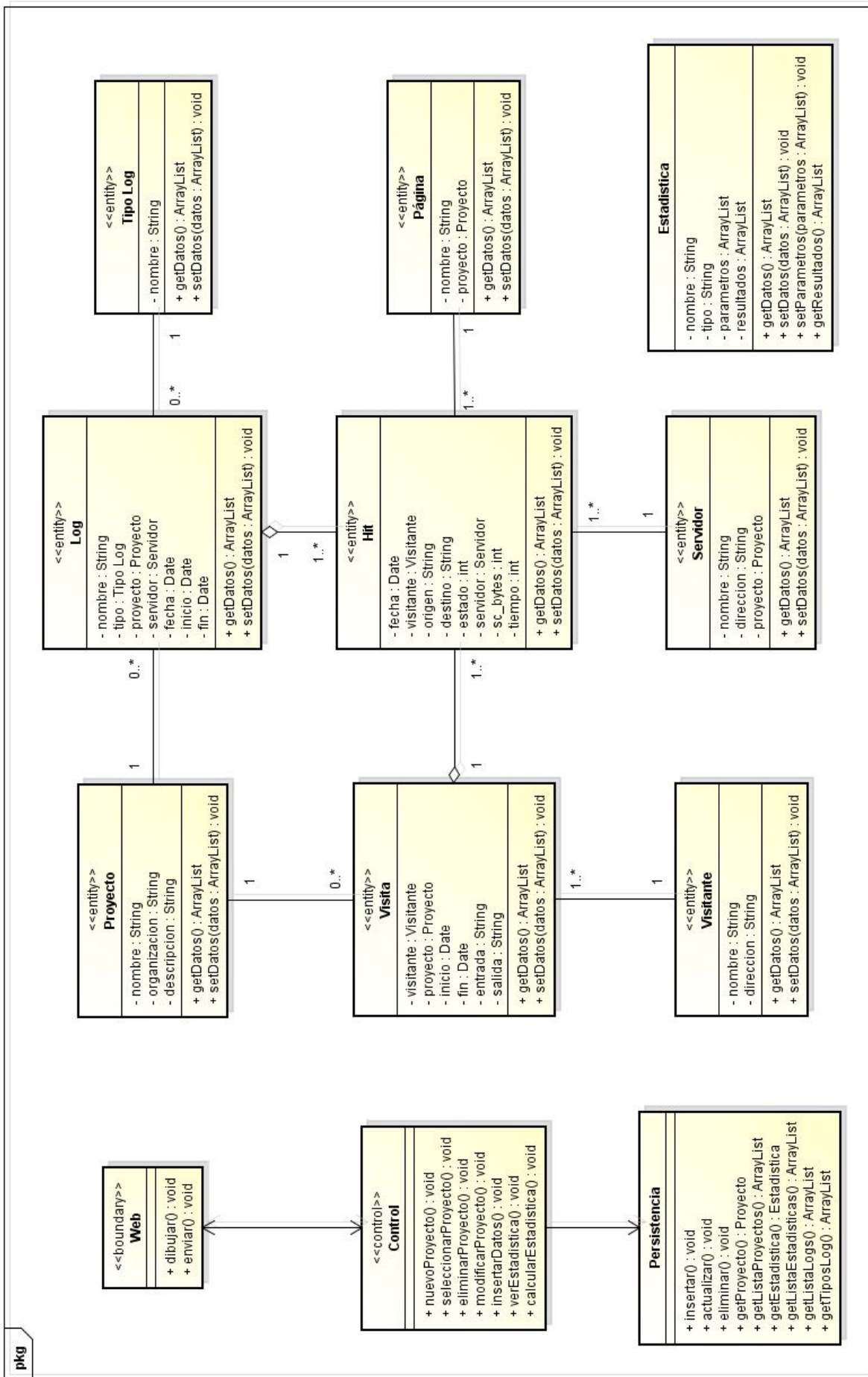


Ilustración 6-3 Diagrama de clases de análisis

6.4 Modelo Dinámico

El modelo dinámico muestra la forma en que el sistema y los objetos de que consta van variando con el tiempo. Para realizarlo se busca como interaccionan los objetos entre sí y que respuestas son visibles externamente. Así se llegan a describir una serie de acciones de carácter general que se han de realizar para poder llevar a cabo la herramienta para el análisis web.

El objetivo de un diagrama de secuencia es destacar la ordenación temporal de los mensajes. Un diagrama de secuencia se construye colocando los objetos que participan en la interacción en la parte superior del diagrama. A continuación, se colocan los mensajes que los objetos envían y reciben a lo largo del tiempo.

Los diagramas de secuencia de la etapa de análisis tratan únicamente de mostrar la idea del funcionamiento de la herramienta siendo mucho más detallados los diagramas de la etapa de diseño.

6.4.1 Nuevo Proyecto

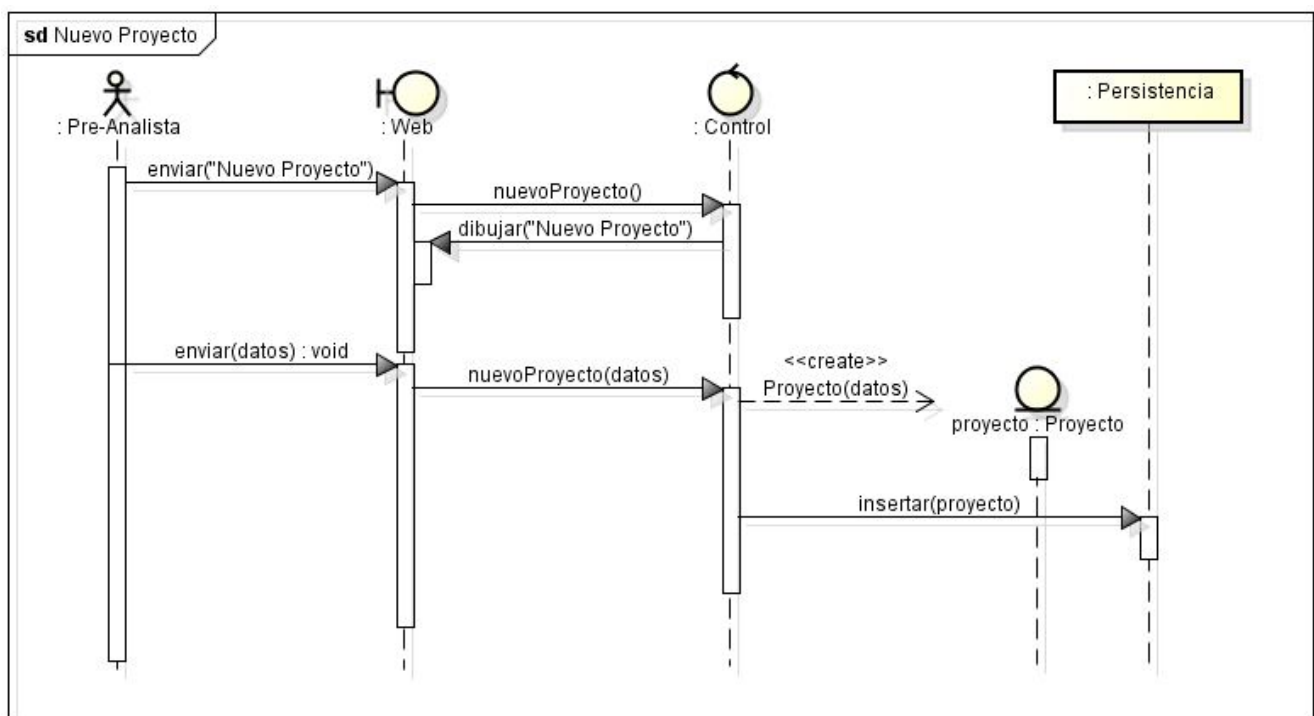


Ilustración 6-4 Diagrama de secuencia de análisis Nuevo Proyecto

6.4.2 Eliminar Proyecto

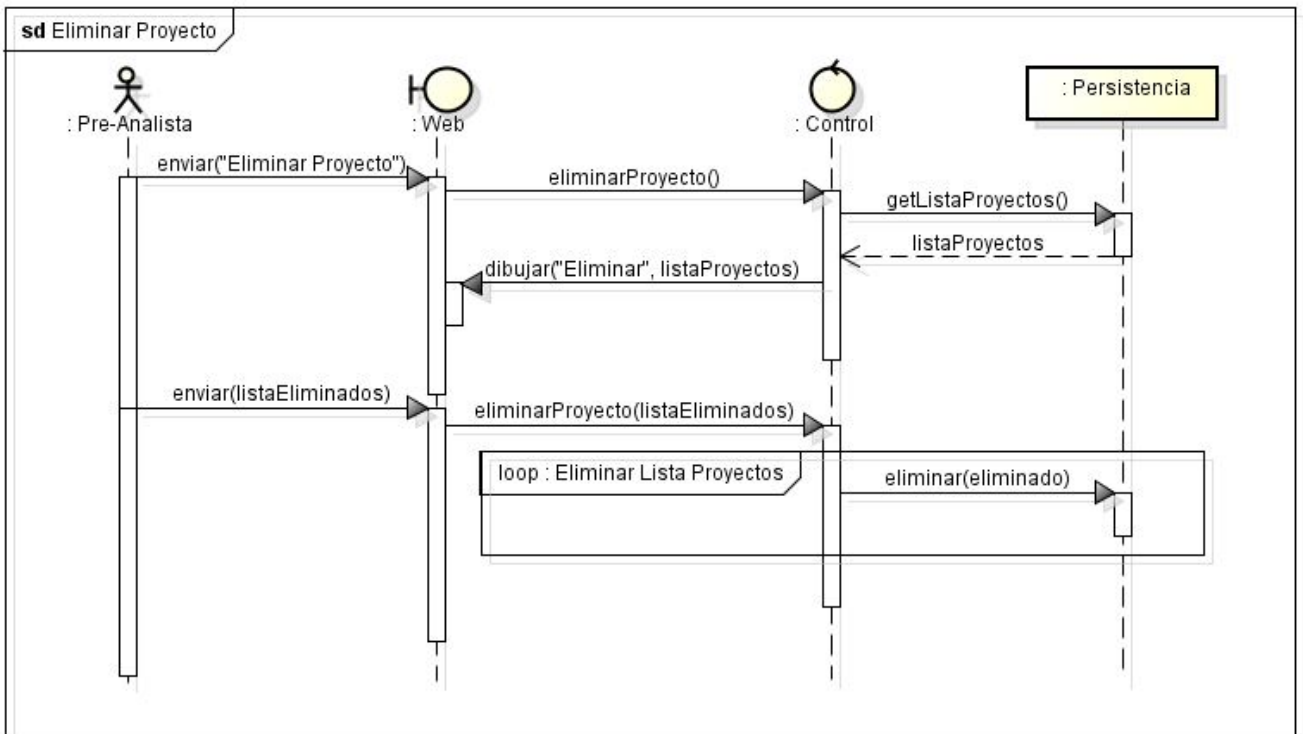


Ilustración 6-5 Diagrama de secuencia de análisis Eliminar Proyecto

6.4.3 Seleccionar Proyecto

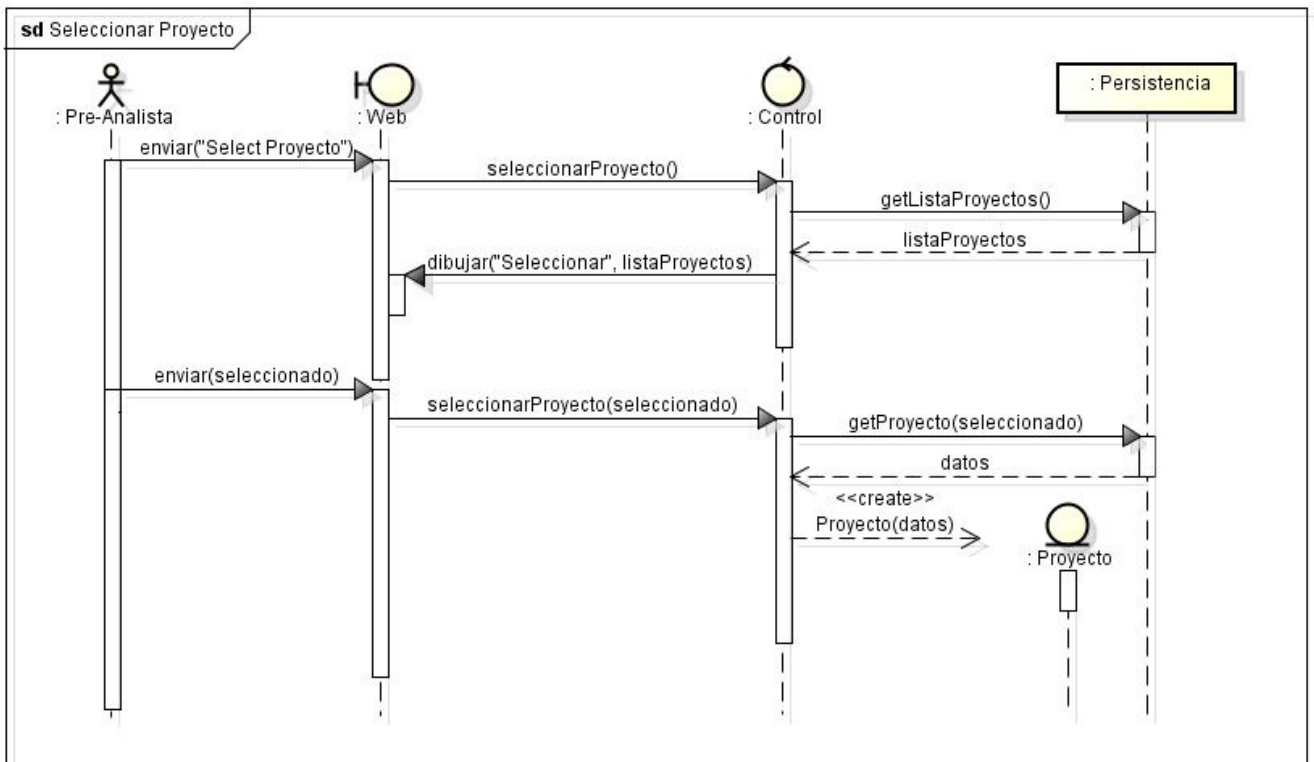


Ilustración 6-6 Diagrama de secuencia de análisis Seleccionar Proyecto

6.4.4 Modificar Proyecto

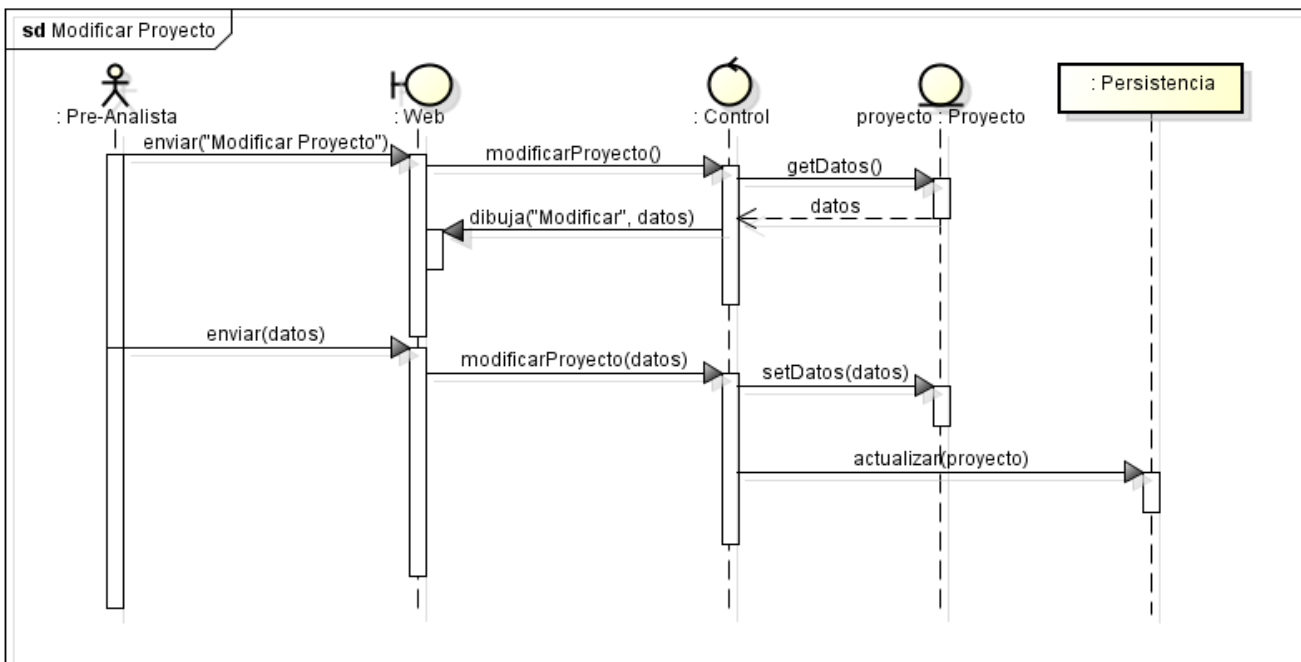


Ilustración 6-7 Diagrama de secuencia de análisis Modificar Proyecto

6.4.5 Ver Estadística

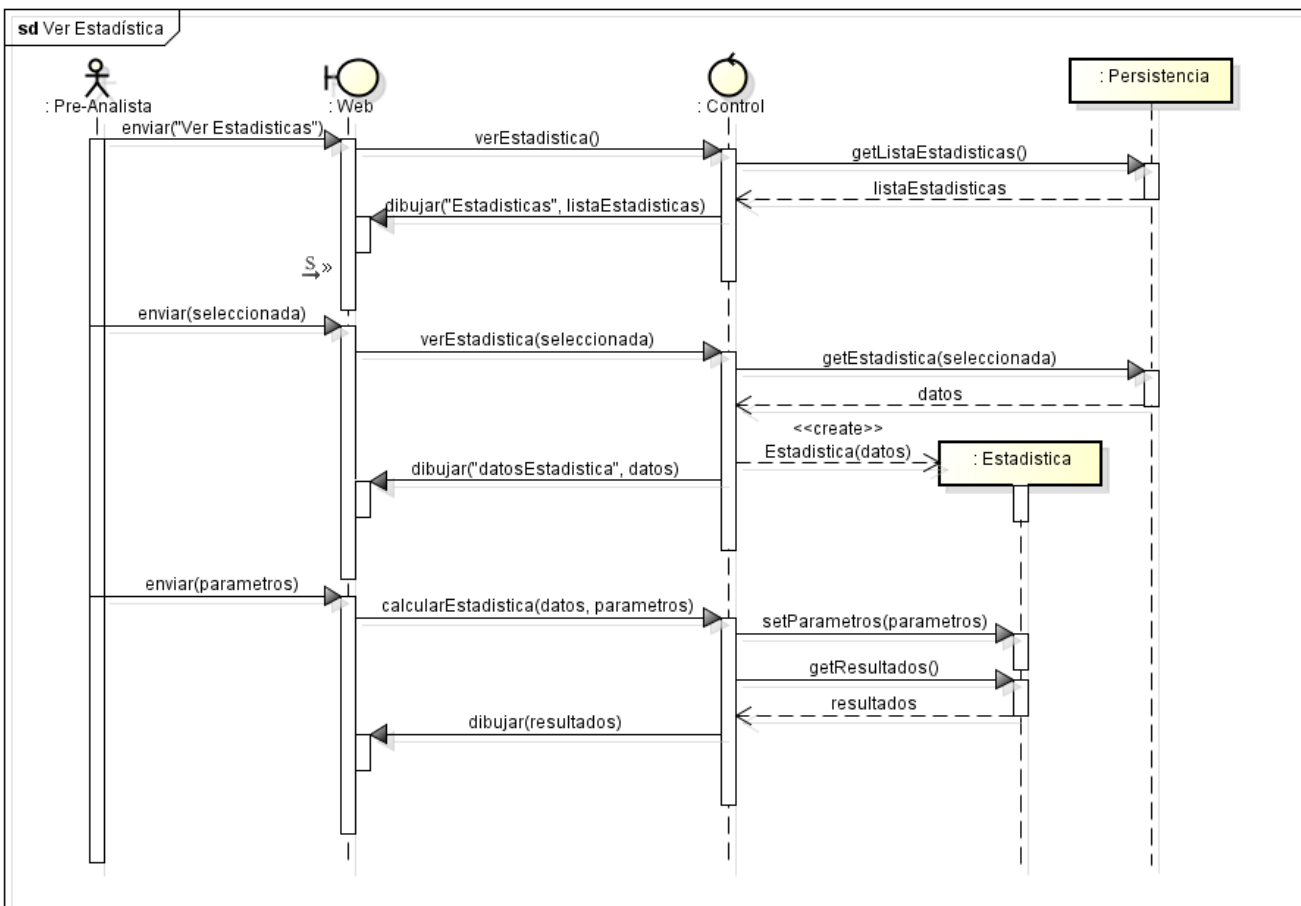


Ilustración 6-8 Diagrama de secuencia de análisis Ver Estadística

6.4.6 Insertar Archivo Log

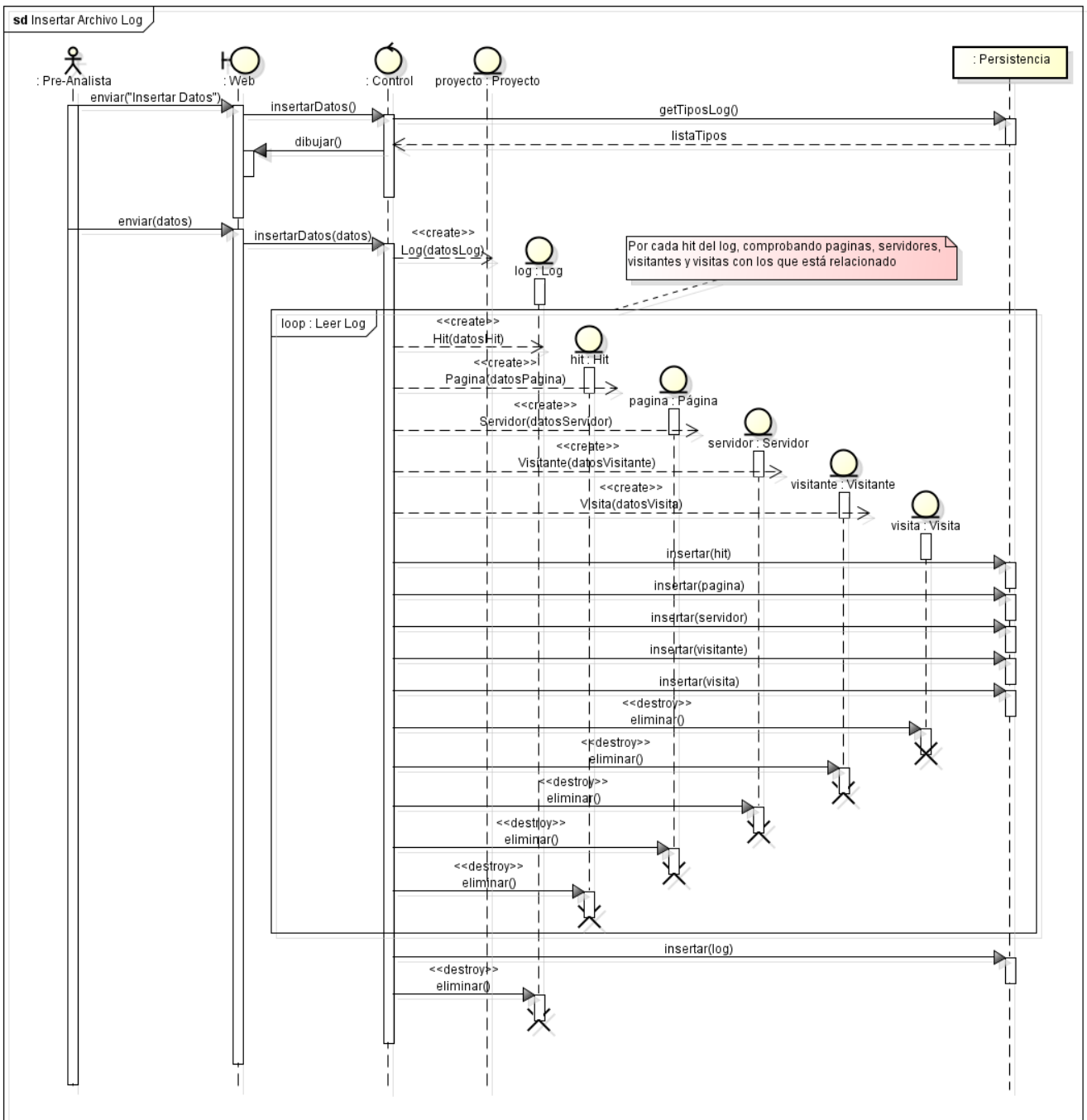


Ilustración 6-9 Diagrama de secuencia de análisis Insertar Archivo Log

6.4.7 Eliminar Archivo Log

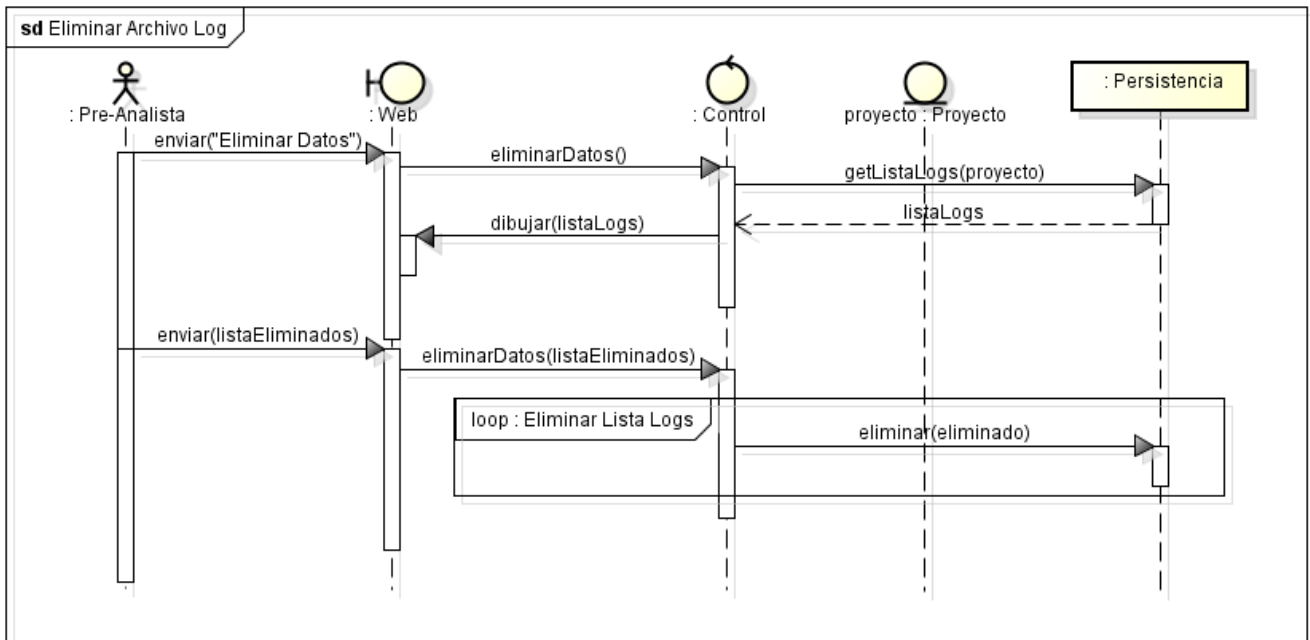


Ilustración 6-10 Diagrama de secuencia de análisis Eliminar Archivo Log

7.1 Arquitectura del sistema

La arquitectura del sistema define de forma global su estructura, funcionamiento e interacción entre partes. Para la resolución de esta herramienta se ha optado por una arquitectura Modelo-Vista-Controlador organizada en capas por varias razones.

En primer lugar, permite visualizar datos con facilidad, incluso si estos cambian en tiempo real. Esto es importante, pues el propósito principal de la herramienta es la visualización de estadísticas basadas en una gran cantidad de datos, y además, una de las posibles líneas futuras es recogerlos directamente desde los archivos log mientras estos se van generando en los diferentes servidores.

En segundo lugar, admite una gran variabilidad a la hora de crear diferentes vistas. Al ser una aplicación que genera datos a partir de otros, estos se podrían querer visualizar de diferentes formas y/o en diferentes formatos, como a través de una aplicación de escritorio o una aplicación móvil.

Y por último, su organización en capas dota a la herramienta de cierta versatilidad; ya que podría permitir su instalación y ejecución en varios servidores; y seguridad, puesto que los datos están totalmente separados de la verdadera herramienta, pudiéndose hacer copias de los mismos y utilizando alguna de dichas copias en caso de borrado de los originales.

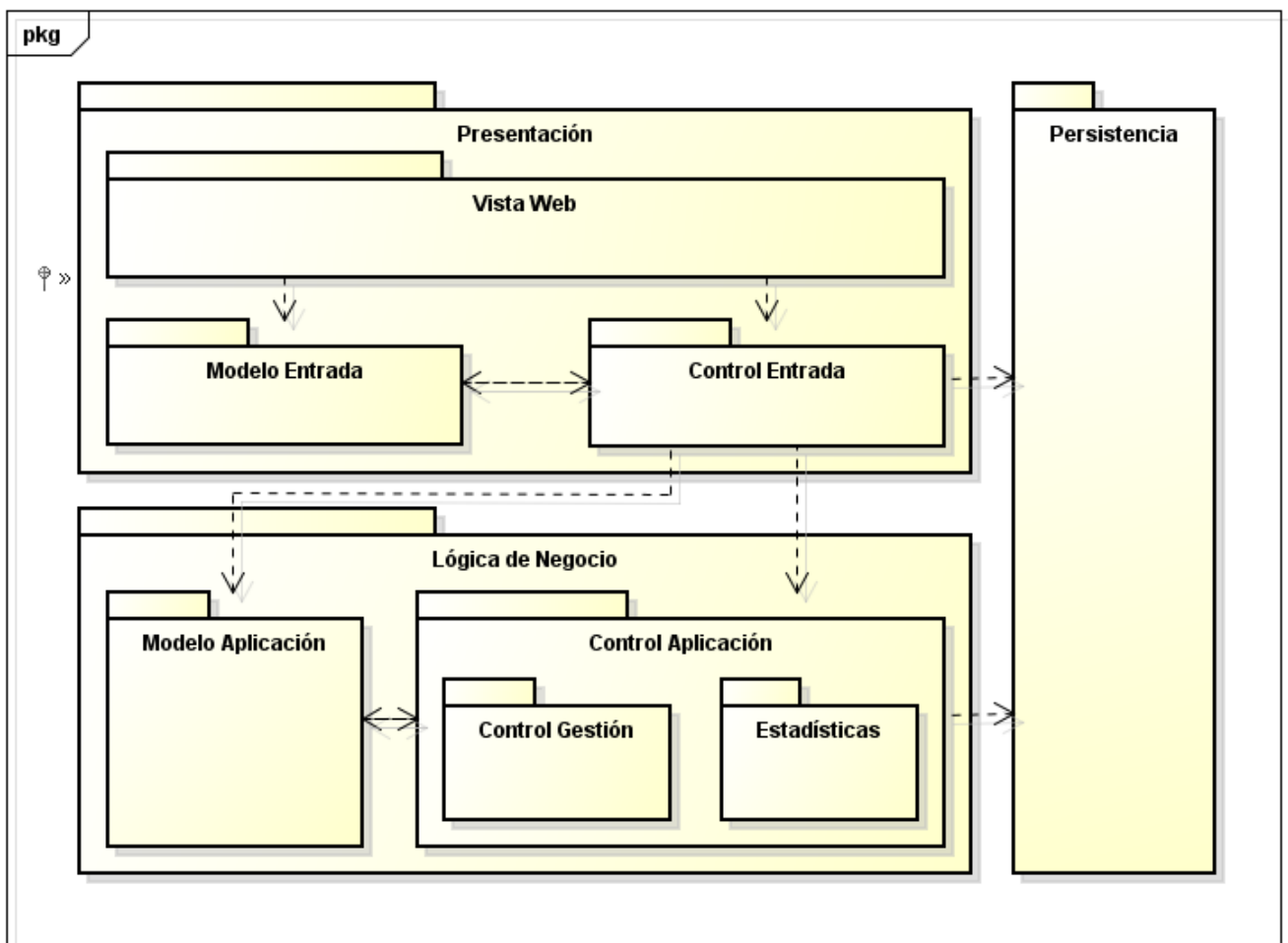


Ilustración 7-1 Arquitectura del sistema

En la Ilustración 7-1 se pueden observar las dos capas horizontales de las que consta el sistema, Presentación y Lógica de Negocio. Sendas capas están encapsuladas en el interior de sus paquetes

homónimos y necesitan de los servicios de un paquete auxiliar, Persistencia, que proporciona los medios necesarios para el almacenamiento de los datos y su recuperación.

Por otro lado, cabe destacar que debido a las dos funcionalidades principales de la herramienta, la gestión de proyectos y la realización de estadísticas, existe otra organización vertical además de la mencionada en capas, aunque menos visible. Está subyace al analizar los paquetes control principales, ya que se puede concluir que cada uno de ellos está dividido en otros dos dedicado a una de estas dos funcionalidades; como se puede ver en el paquete Control Aplicación en la Ilustración 7-1.

7.2 Diseño Estático

En este apartado se desglosan los diferentes módulos de los que consta el sistema, explicando sus correspondientes responsabilidades, poniendo especial atención a las diversas soluciones propuestas y relacionando estas con los patrones de diseño aplicados. También se describirán las clases más relevantes enumerando sus características principales, sin embargo, al realizar dicha descripción no se han mencionado los métodos que devuelven o dan valor a las propiedades o atributos.

7.2.1 Paquete Presentación

El paquete Presentación realiza las funciones del componente Vista dentro de la arquitectura propuesta; gracias a él, el usuario puede interactuar con el sistema. Es el responsable de presentar al usuario toda la funcionalidad disponible con sus opciones; recoger y transmitir las acciones que el usuario desee realizar junto con los datos necesarios para llevarlas a cabo; y presentar los resultados de dichas acciones.

A su vez este paquete también está estructurado con una arquitectura Modelo-Vista-Controlador dividida en dos capas; una formada por el paquete Vista Web y otra compuesta por los paquetes Modelo Entrada y Control Entrada.

7.2.1.1 Paquete Vista Web

El paquete Vista Web realiza las funciones del componente Vista dentro de la arquitectura que estructura el paquete Presentación; implementa la interfaz de usuario constituyendo un marco visual con el que este puede interactuar.

Está compuesto principalmente por páginas web .PSP que se puede dividir conceptualmente en dos conjuntos: las páginas dedicadas a la gestión de proyectos y logs y las que organizan y muestran estadísticas.

En la Ilustración 7-2 se pueden observar los elementos y la estructura de las páginas dedicadas a la gestión y en la Ilustración 7-3 de las dedicadas a las estadísticas. Cabe destacar, que los enlaces se han representado con una flecha discontinua, que muchos de ellos se calculan en tiempo de ejecución y que, para simplificar, sólo se han representado algunas de las composiciones de páginas, pero las demás se intuyen; por ejemplo, todas las páginas de estadísticas cuyo nombre comienza por el prefijo est_ están compuestas por estadística_cabecera.psp, algo a tener en cuenta si se desea ampliar el sistema añadiendo nuevas estadísticas.

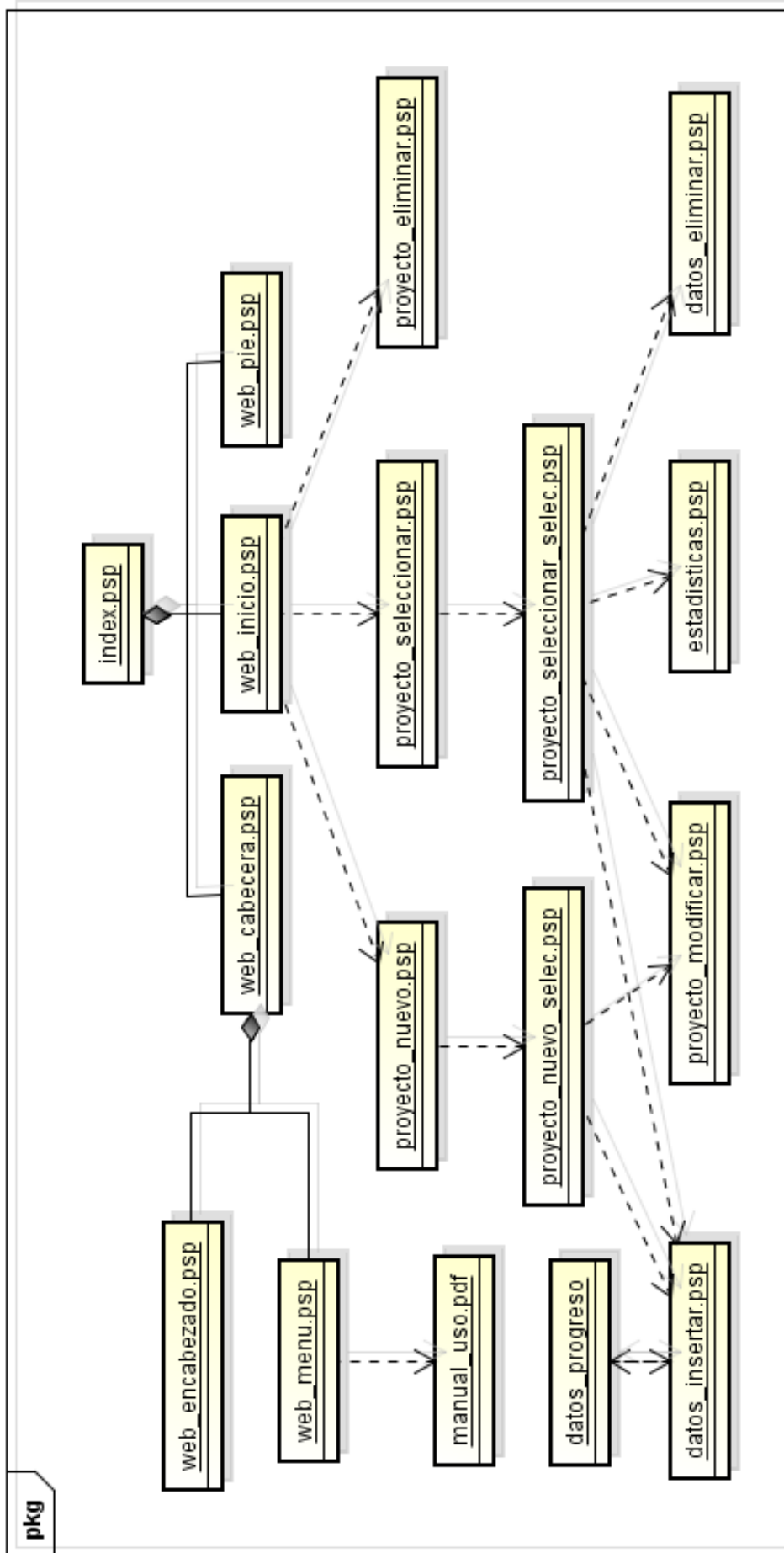


Ilustración 7-2 Paquete Vista Web - Gestión

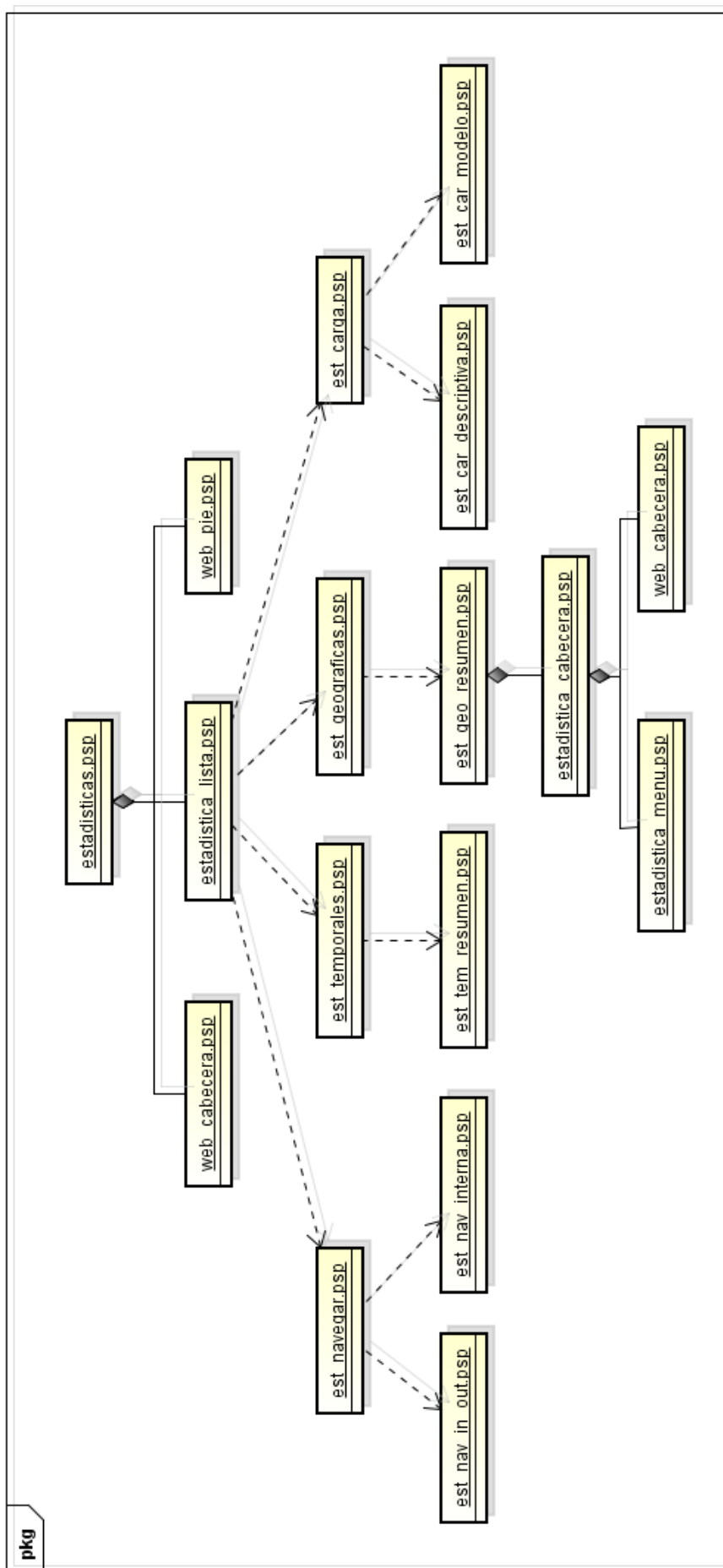


Ilustración 7-3 Paquete Vista Web - Estadísticas

7.2.1.2 Paquete Modelo Entrada

El paquete Modelo Entrada realiza las funciones del componente Modelo dentro de la arquitectura que estructura el paquete Presentación; contiene los datos que se van a presentar en la vista o los que se recogen de ella. Como se puede ver en la Ilustración 7-4 está formado por una serie de clases que se corresponden con componentes conceptúeles de recogida y presentación de datos o selección de acciones, como menú, formulario, seleccionable... y dos paquetes, Formularios Gestión y Formularios Estadísticas.

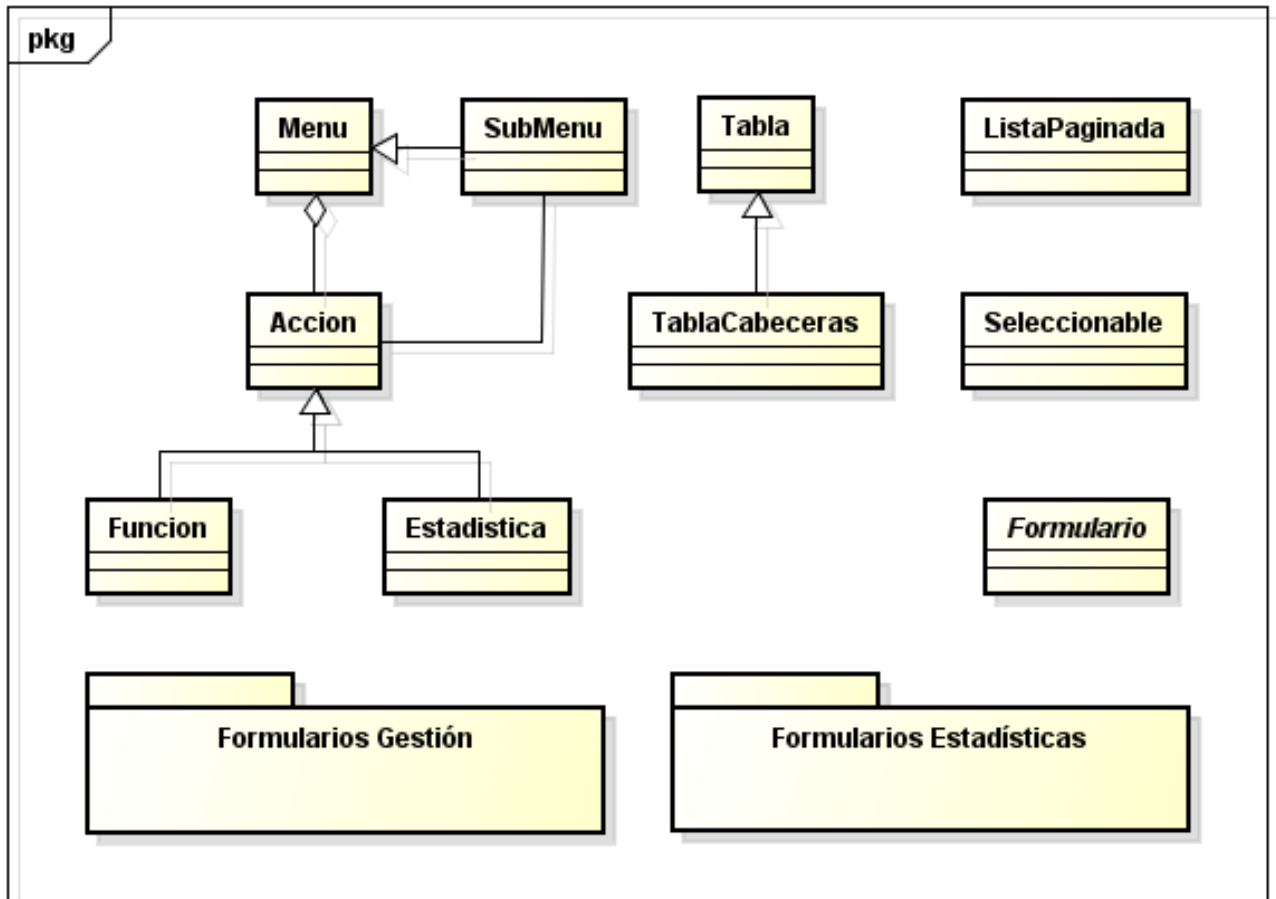


Ilustración 7-4 Paquete Modelo Entrada

7.2.1.2.1 Descripción de clases

Menu	
Propiedades	
- acciones[]	Lista de acciones del menú.
Métodos	
+ getAccion (indice)	Devuelve la acción del menú que se encuentra en la posición indicada por parámetro índice + 1.
+ getNumAcciones	Devuelve el número de acciones del menú.
+ insertarAccion (accion)	Añade la acción pasada como parámetro al final del menú.
+ borrarAcciones	Vacía el menú de acciones.
Responsabilidades	
<ul style="list-style-type: none"> • Contener una lista de acciones (funciones o estadísticas). 	

Accion	
Propiedades	
- id	Identificador de la acción.
- nombre	Nombre de la Acción.
- url	URL al que dirigirse para realizar la acción.
- icono	Ruta del archivo que contiene la imagen que representa la acción.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de una acción. 	

Funcion	
Propiedades	
- seleccion	Indica si la función se ha de aplicar sobre un proyecto seleccionado o no.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de una función. 	

Estadística	
Propiedades	
- tipo	Número que identifica el tipo de estadística.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de una estadística. 	

TablaCabeceras	
Propiedades	
- primfilcab[]	Contenido de la primera fila.
- primcolcab[]	Contenido de la primera columna.
Métodos	
+ TienePrimFilCab	Devuelve valor Verdadero si la tabla tiene encabezado fila.
+ TienePrimColCab	Devuelve valor Verdadero si la tabla tiene encabezado columna.
Responsabilidades	
<ul style="list-style-type: none"> • Contener una tabla bidimensional de objetos con posibles encabezados de fila y columna. 	

Seleccionable	
Propiedades	
- lista[]	Lista de objetos seleccionables.
- seleccionado	Posición del elemento seleccionado.
Métodos	
+ insertar (objeto)	Añade el objeto pasado como parámetro al final de la lista de objetos seleccionables.
+ numObjetos	Devuelve el número de objetos seleccionables.
+ vaciar	Elimina todos los objetos seleccionables.
Responsabilidades	
<ul style="list-style-type: none"> • Contener una lista de objetos seleccionables. 	

ListaPaginada	
Propiedades	
- lista[]	Lista de objetos.
- objetospagina	Número de objetos de cada página o trozo en el que se divide la lista
Métodos	
+ getListaPagina (num_pagina)	Devuelve la lista de objetos que pertenecen al trozo o página correspondiente al valor del parámetro num_página.
+ numPaginas	Devuelve el número de trozos o páginas en las que está dividida la lista.
+ numObjetos	Devuelve el número de objetos total.
+ vaciar	Elimina todos los objetos de la lista.
Responsabilidades	
<ul style="list-style-type: none"> • Contener una lista de objetos troceada en páginas, todas con un mismo número de objetos determinado, excepto la última que puede contener un número igual o menor. 	

7.2.1.2.2 Paquete Formularios Gestión

El paquete Formularios Gestión engloba el conjunto de clases que constituye el modelo de datos necesario para la introducción de los datos imprescindibles para realizar las tareas de gestión de proyectos y archivos log. En la Ilustración 7-5 se pueden observar sus elementos y su estructura.

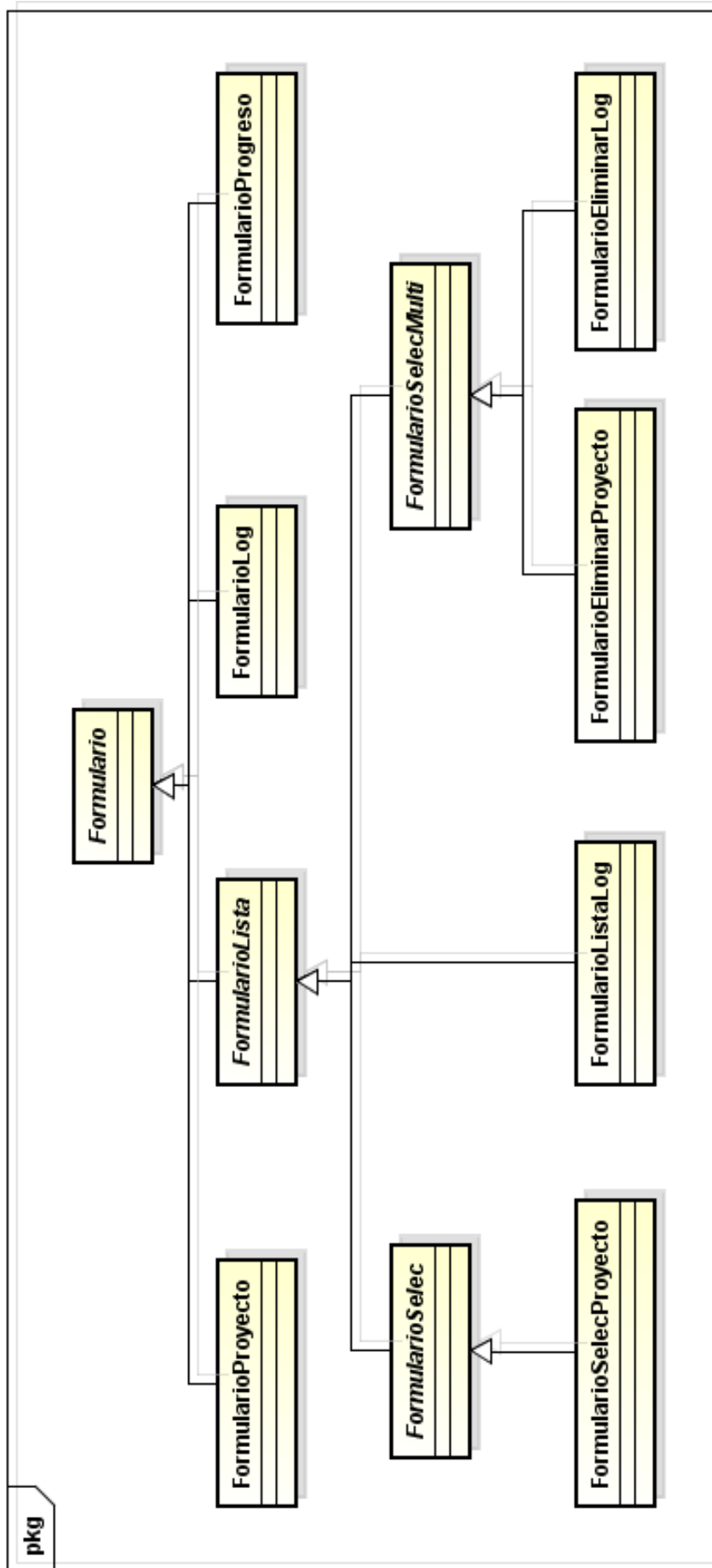


Ilustración 7-5 Paquete Formularios Gestión

7.2.1.2.2.1 Descripción de clases

FormularioProyecto	
Propiedades	
- nombre	Campo con el nombre del proyecto.
- organización	Campo con el nombre de la organización a la que pertenece el proyecto.
- descripción	Campo con texto que describe el proyecto.
Métodos	
+ getDatos	Devuelve un array con los valores de los campos del formulario.
+ recogerDatos (form)	Recoge del parámetro form los valores de los campos del formulario.
+ cogerDatos (datos)	Introduce los valores pasados como parámetro en los campos del formulario.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de los campos de un formulario con los datos de un proyecto. • Recoger la información de dichos campos. 	

FormularioProgreso	
Propiedades	
- porcentaje	Campo con el porcentaje del progreso.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información del formulario que indica el porcentaje realizado de una acción del sistema. 	

FormularioLog	
Propiedades	
- archivo	Campo con el archivo log.
- solapar	Campo con el indicador que permite introducir un log con fechas de hits entre las fechas de otros logs presentes en el sistema.
- tipos	Campo con el seleccionable del tipo de log.
Métodos	
+ getTipo	Devuelve el tipo de log seleccionado.
+ recogerDatos (form)	Recoge del parámetro form los valores de los campos del formulario.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de los campos de un formulario para introducir los datos de un archivo log en un proyecto del sistema. • Recoger la información de dichos campos. 	

FormularioLista	
Propiedades	
- lista[]	Lista de objetos de la página actual.
- pagina	Campo con el número de la página actual.
- anterior	Indica si está habilitada la opción de pasar a la página anterior.
- siguiente	Indica si está habilitada la opción de pasar a la página siguiente.
- objetosPagina	Campo con el número de objetos por página.
Métodos	
+ numObjetos	Devuelve el número total de objetos de la lista (página actual).
+ siguientePágina	Incrementa el valor de la página actual en 1 y habilita la opción de pasar a la página anterior.
+ anteriorPágina	Disminuye el valor de la página actual en 1 y habilita la opción de pasar a la página siguiente.
+ habilitarAnterior	Habilita la opción de pasar a la página anterior.

<i>FormularioLista</i>	
+ deshabilitarAnterior	Deshabilita la opción de pasar a la página anterior.
+ habilitarSiguiente	Habilita la opción de pasar a la página siguiente.
+ deshabilitarSiguiente	Deshabilita la opción de pasar a la página siguiente.
- recogerPagina (form)	Recoge del parámetro form los valores de los campos pagina y objetosPagina.
Responsabilidades	
<ul style="list-style-type: none"> • Base para formularios que contengan una página con un listado de objetos. 	

<i>FormularioListaLog</i>	
Métodos	
+ recogerDatos (form)	Recoge del parámetro form, la página actual y el número de objetos por página.
Responsabilidades	
<ul style="list-style-type: none"> • Contener una página con una lista de logs. • Permitir el cambio de página y su actualización recogiendo el número de página y el número de objetos por página. 	

<i>FormularioSelec</i>	
Propiedades	
- seleccionado	Número identificativo del objeto seleccionado.
Métodos	
+ recogerElDato (form, nombre)	Recoge del parámetro form el valor del campo cuyo nombre es el valor del parámetro nombre y lo deja en la propiedad seleccionado.
Responsabilidades	
<ul style="list-style-type: none"> • Base para formularios que contengan una página con un listado de objetos y permitan la selección de uno de ellos. 	

FormularioSelecProyecto	
Métodos	
+ recogerDatos (form)	Recoge del parámetro form, el identificador del proyecto seleccionado.
Responsabilidades	
<ul style="list-style-type: none"> • Contener una página con una lista de proyectos. • Recoger el proyecto seleccionado por el usuario de dicha lista. 	

FormularioSelecMulti	
Propiedades	
- seleccionados[]	Lista de números identificativos de los objetos seleccionados.
Métodos	
+ recogerLosDatos (form, numero, prefijo)	Recoge del parámetro form, los valores de los campos cuyo nombre empieza por el valor del parámetro nombre y termina por un número que va de 0 al valor del parámetro numero menos 1, y los deja en la propiedad seleccionados.
Responsabilidades	
<ul style="list-style-type: none"> • Base para formularios que contengan una página con un listado de objetos y que permitan la selección de varios de ellos. 	

FormularioEliminarProyecto	
Métodos	
+ recogerDatos (form)	Recoge del parámetro form, los identificadores de los proyectos seleccionados.
Responsabilidades	
<ul style="list-style-type: none"> • Contener una página con una lista de proyectos. • Recoger los proyectos seleccionados por el usuario de dicha lista. 	

FormularioEliminarLog	
Métodos	
+ recogerDatos (form)	Recoge del parámetro form, los identificadores de los logs seleccionados.
Responsabilidades	
<ul style="list-style-type: none"> • Contener una página con una lista de logs. • Recoger los logs seleccionados por el usuario de dicha lista. 	

7.2.1.2.3 Paquete Formularios Estadísticas

El paquete Formularios Estadísticas engloba el conjunto de clases que constituye el modelo de datos necesario para la introducción de los datos imprescindibles para realizar las estadísticas sobre los proyectos. Si se desea ampliar la herramienta con más estadísticas este sería el lugar idóneo para incluir las clases que representen a los formularios que recojan los datos para dichas estadísticas. En la Ilustración 7-6 se pueden observar sus elementos y su estructura.

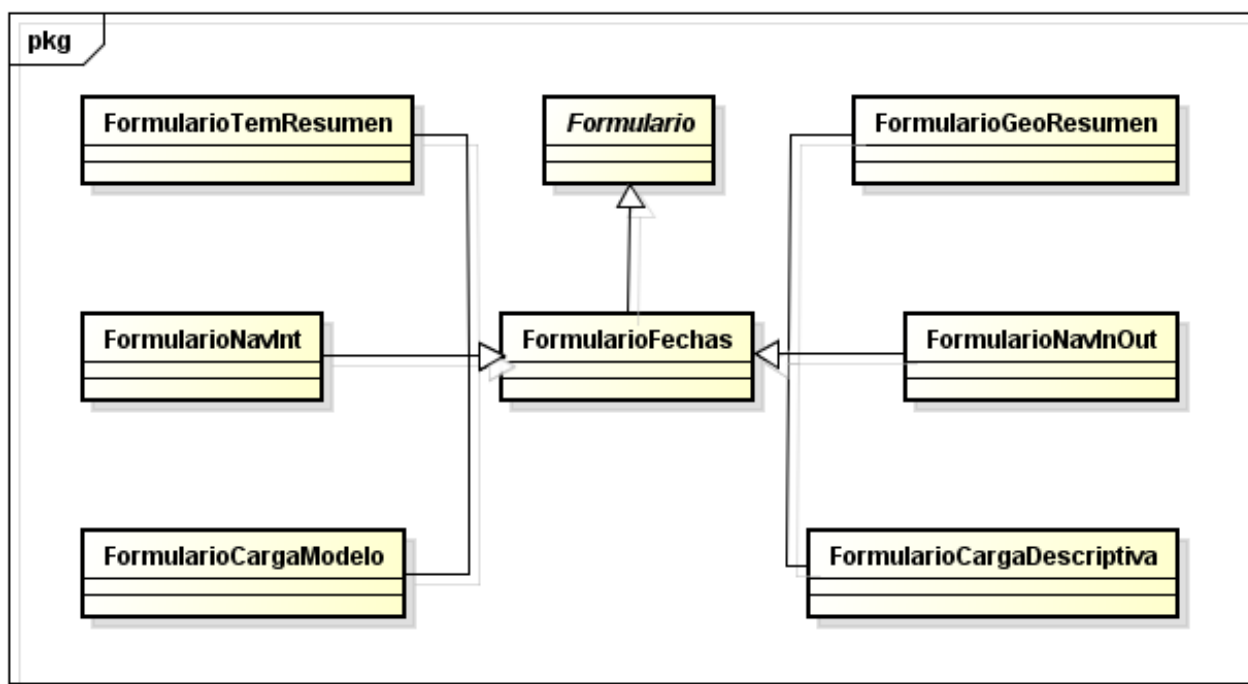


Ilustración 7-6 Paquete Formularios Estadísticas

7.2.1.2.3.1 Descripción de clases

FormularioFechas	
Propiedades	
- fechaIni	Campo con la fecha inicial.
- fechaFin	Campo con la fecha final.
- fechaIniLimit	Campo con la fecha inicial determinada.
- fechaFinLimit	Campo con la fecha final determinada.
- opcionfecha	Campo que indica la opción de quedarse con las fechas determinadas o recoger las introducidas.
Métodos	
+ recogerDatos (form)	Recoge del parámetro form los valores de las fechas introducidas por el usuario o de las determinadas por defecto según el valor recogido en el campo opcionfechas.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de un formulario que permita escoger entre unas fechas determinadas u otras introducidas por el usuario. • Recoger la opción entre dichas fechas y, en el caso de ser las introducidas por el usuario, las propias fechas. 	

FormularioTemResumen	
Propiedades	
- agrupaciones	Campo con el seleccionable del tipo de agrupación temporal necesaria para realizar la estadística.
- tabla	Contenido del resultado de los cálculos de la estadística.
- archivoCSV	Nombre del archivo con el contenido de los cálculos de la estadística.
Métodos	
+ getAgrupacion	Devuelve el tipo de agrupación seleccionada.
+ recogerDatos (form)	Recoge del parámetro form los valores de las fechas y el tipo de agrupación.

FormularioTemResumen	
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de un formulario que permita escoger unas fechas y un valor de agrupación temporal. • Recoger dichas fechas y dicho valor de agrupación. • Contener la información y el nombre del archivo con el resultado de la estadística realizada con los valores recogidos. 	

FormularioGeoResumen	
Propiedades	
- agrupaciones	Campo con el seleccionable del tipo de agrupación geográfica necesaria para realizar la estadística.
- tabla	Contenido del resultado de los cálculos de la estadística.
- archivoCSV	Nombre del archivo con el contenido de los cálculos de la estadística.
Métodos	
+ getAgrupacion	Devuelve el tipo de agrupación seleccionada.
+ recogerDatos (form)	Recoge del parámetro form los valores de las fechas y el tipo de agrupación.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de un formulario que permita escoger unas fechas y un valor de agrupación geográfica. • Recoger dichas fechas y dicho valor de agrupación. • Contener la información y el nombre del archivo con el resultado de la estadística realizada con los valores recogidos. 	

7.2.1.3 Paquete Control Entrada

El paquete Control Entrada realiza las funciones del componente Control dentro de la arquitectura que estructura el paquete Presentación. Su labor consiste en controlar Vista Web, recogiendo y validando los datos introducidos por el usuario, gracias a Modelo Entrada, y adaptándolos y pasándoselos a la capa inferior. También recolecta los datos que le proporcionan Lógica de Negocio y Persistencia, adaptándolos a Modelo Entrada, para presentarlos a través de Vista Web. Como se puede ver en la Ilustración 7-7, este paquete también está formado por otros tres, Control Entrada Modelo, Control Entrada Gestión y Control Entrada Estadísticas.

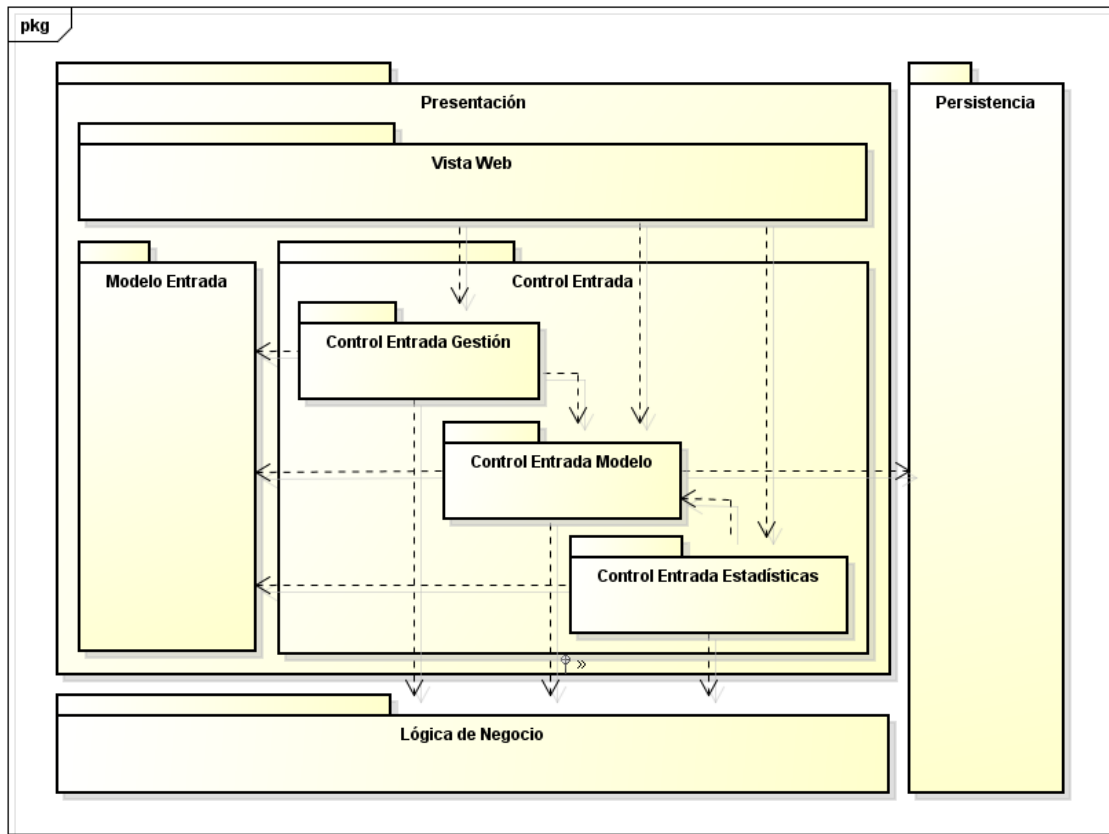


Ilustración 7-7 Paquete Control Entrada

7.2.1.3.1 Paquete Control Entrada Modelo

El paquete Control Entrada Modelo engloba el conjunto de clases necesarias para la adaptación de los datos entre los paquetes Lógica de Negocio o Persistencia y el paquete Modelo Entrada. En la Ilustración 7-8 se pueden observar sus elementos y su estructura.

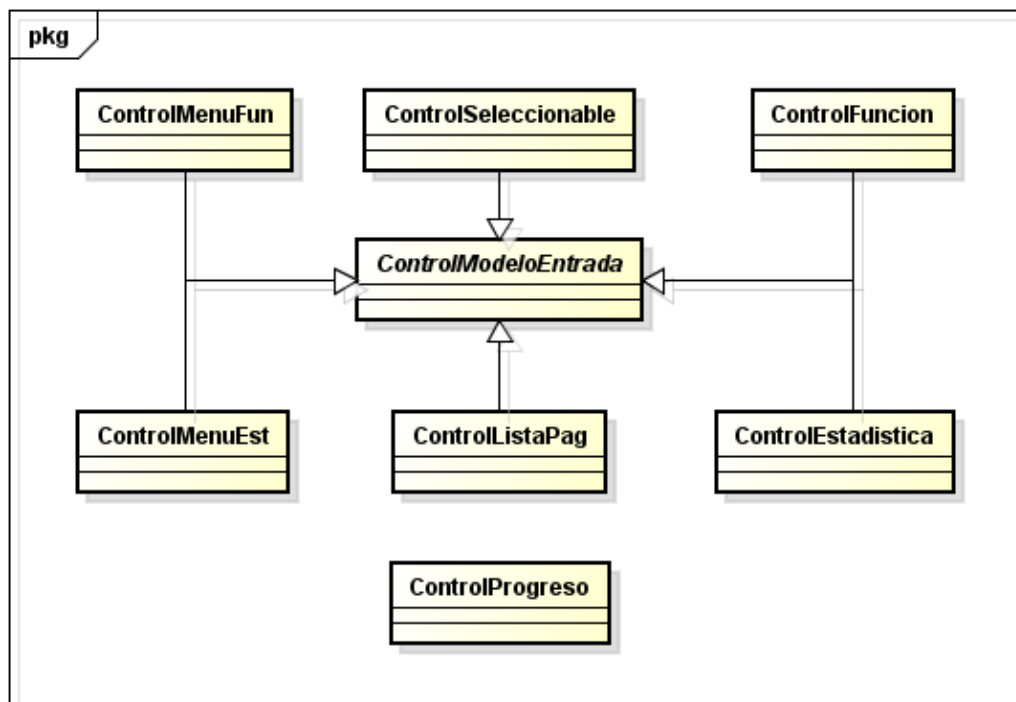


Ilustración 7-8 Paquete Control Entrada Modelo

7.2.1.3.1.1 Descripción de clases

<i>ControlModeloEntrada</i>	
Métodos	
+ cargarDatos (objeto, valor)	Carga los elementos del parámetro valor, o que cumplan con el parámetro valor, en el parámetro objeto.
Responsabilidades	
<ul style="list-style-type: none"> • Base para la carga de datos en los modelos del paquete Modelo Entrada desde el paquete Persistencia o desde el paquete Lógica de Negocio. 	

<i>ControlMenuFun</i>	
Métodos	
+ cargarDatos (menu, selec)	Carga en el parámetro menu, las funciones que necesitan un proyecto seleccionado para poder ejecutarse o las que no lo necesitan según el parámetro selec.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la carga de funciones sobre un menú. 	

<i>ControlMenuEst</i>	
Métodos	
+ cargarDatos (submenu, tipo)	Carga en el parámetro submenu, las estadísticas del tipo cuyo identificador sea el parámetro tipo.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la carga de estadísticas sobre un submenú. 	

<i>ControlFuncion</i>	
Métodos	
+ cargarDatos (id, funcion)	Carga en el parámetro funcion, los datos de la función cuyo identificador sea el parámetro id.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la carga de los datos de una función. 	

ControlEstadistica	
Métodos	
+ cargarDatos (id, estadistica)	Carga en el parámetro estadistica, los datos de la estadística cuyo identificador sea el parámetro id.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la carga de los datos de una estadística. 	

ControlListaPag	
Métodos	
+ cargarDatos (listPag, listObj)	Carga en el parámetro listPag, los objetos del parámetro listObj.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la carga de los objetos de una Lista de Objetos en una Lista Paginada. 	

ControlSeleccionable	
Métodos	
+ cargarDatos (selec, listObj)	Carga en la lista del parámetro selec, los objetos del parámetro listObj.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la carga de los objetos de una Lista de Objetos en la lista de un seleccionable. 	

7.2.1.3.2 Paquete Control Entrada Gestión

El paquete Control Entrada Gestión engloba el conjunto de clases necesarias para el control de la recogida y validación de datos para realizar las tareas de gestión de proyectos y archivos log. En la Ilustración 7-9 se pueden observar sus elementos y su estructura.

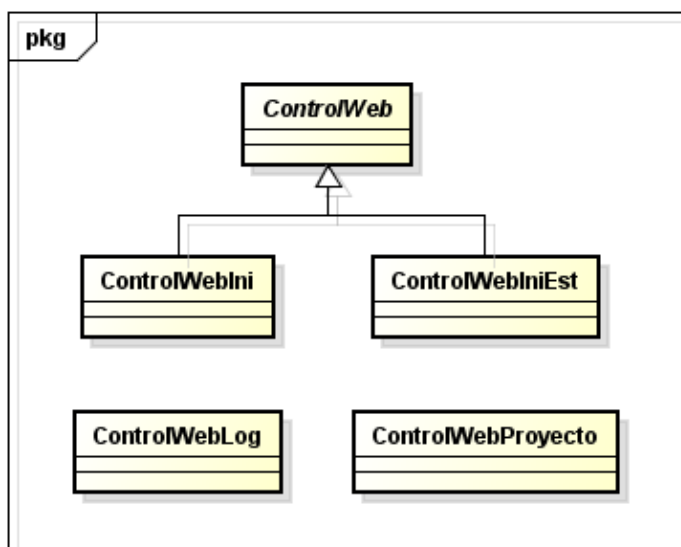


Ilustración 7-9 Paquete Control Entrada Gestión

7.2.1.3.2.1 Descripción de clases

<i>ControlWeb</i>	
Métodos	
+ escribirMenu (menu, valor)	Escribe el parámetro menu sobre una web según el parámetro valor.
Responsabilidades	
<ul style="list-style-type: none"> • Base para el control de la escritura de los menús de una web. 	

<i>ControlWebIni</i>	
Métodos	
+ escribirMenu (menu, seleccion)	Escribe el parámetro menú sobre una web con las funciones adecuadas según el parámetro seleccion.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar de la escritura del menú de funciones de la web. 	

<i>ControlWebIniEst</i>	
Métodos	
+ escribirMenu (menu, tipo)	Escribir el parámetro menú sobre una web con las estadísticas adecuadas según el parámetro tipo.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar de la escritura del menú de estadísticas de la web. 	

ControlWebProyecto	
Métodos	
+ insertar (form, formProyect, sesion)	Controla la recogida y validación de datos para la inserción de un nuevo proyecto.
+ seleccionar (form, formProyect, sesion)	Controla la recogida y validación de datos para la selección de un proyecto.
+ modificar (form, formProyect, sesion)	Controla la recogida y validación de datos para la modificación del proyecto en sesion.
+ eliminar (form, formProyect, sesion)	Controla la recogida y validación de datos para la eliminación de varios proyectos.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la entrada de datos para la gestión de proyectos. 	

ControlWebLog	
Métodos	
+ insertar (form, formLog, sesion)	Controla la recogida y validación de datos para la inserción de un nuevo log en el proyecto en sesion.
+ guardar (form, formLog, sesion)	Controla la recogida y validación de datos para la inserción de los datos de un nuevo archivo log, así como la visualización del progreso de la acción.
+ listar (form, formLog, sesion)	Controla la visualización de los archivos log del proyecto en sesion.
+ eliminar (form, formLog, sesion)	Controla la recogida y validación de datos para la eliminación de varios archivos log del proyecto en sesion.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la entrada de datos para la gestión de los logs de un proyecto. 	

7.2.1.3.3 Paquete Control Entrada Estadísticas

El paquete Control Entrada Estadísticas engloba el conjunto de clases necesarias para el control de la recogida y validación de datos para realizar las estadísticas de los proyectos y sus archivos log. Además, estas mismas clases, se encargan de recibir de Lógica de Negocio los datos del resultado de dichas estadísticas y de adaptarlos a Modelo Entrada. En la Ilustración 7-10 se pueden observar sus elementos y su estructura.

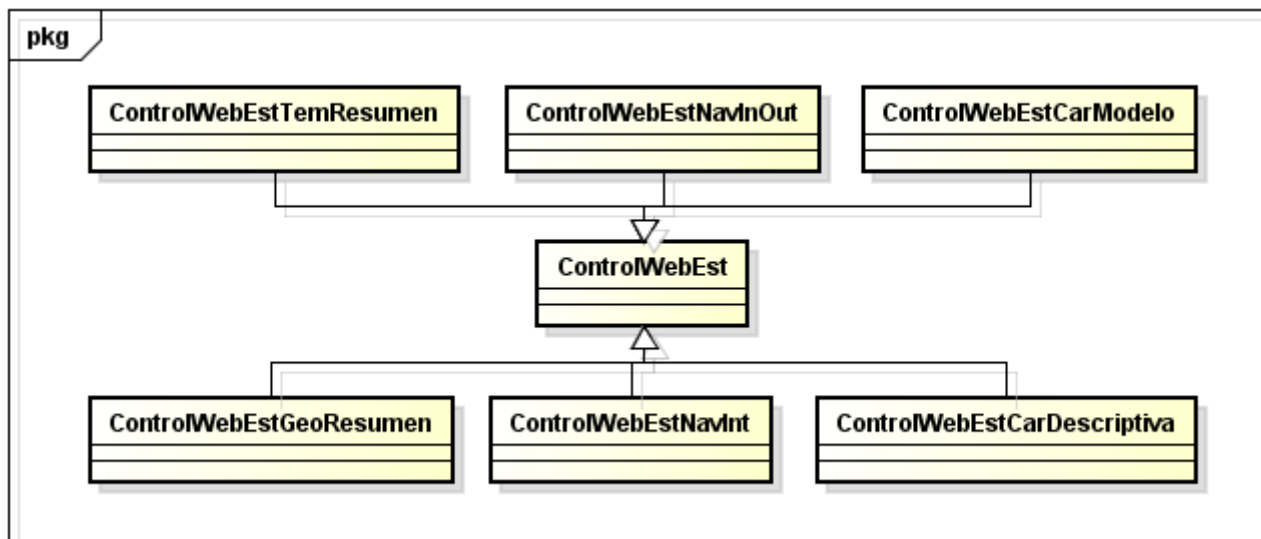


Ilustración 7-10 Paquete Control Entrada Estadísticas

7.2.1.3.3.1 Descripción de clases

<i>ControlWebEst</i>	
Métodos	
+ estadística (form, formEst, sesion)	Controla la recogida y validación de datos para la realización de una estadística y su posterior visualización.
Responsabilidades	
<ul style="list-style-type: none"> • Base para el control de la entrada y visualización de una estadística. 	

ControlWebEstTemResumen	
Métodos	
+ estadística (form, formEst, sesion)	Controla la recogida y validación de datos para la realización de una estadística de tipo resumen temporal y su posterior visualización.
Responsabilidades	
<ul style="list-style-type: none"> • Base para el control de la entrada y visualización de una estadística de tipo resumen temporal. 	

ControlWebEstGeoResumen	
Métodos	
+ estadística (form, formEst, sesion)	Controla la recogida y validación de datos para la realización de una estadística de tipo resumen geográfico y su posterior visualización.
Responsabilidades	
<ul style="list-style-type: none"> • Base para el control de la entrada y visualización de una estadística de tipo resumen geográfico. 	

ControlWebNavInOut	
Métodos	
+ estadística (form, formEst, sesion)	Controla la recogida y validación de datos para la realización de una estadística que muestra las páginas de salida y entrada, y su posterior visualización.
Responsabilidades	
<ul style="list-style-type: none"> • Base para el control de la entrada y visualización de una estadística que muestra las páginas de salida y entrada. 	

ControlWebEstNavInt	
Métodos	
+ estadística (form, formEst, sesion)	Controla la recogida y validación de datos para la realización de una estadística que muestra las probabilidades de transición entre páginas, y su posterior visualización.
Responsabilidades	
<ul style="list-style-type: none"> • Base para el control de la entrada y visualización de una estadística que muestra las probabilidades de transición entre páginas. 	

ControlWebEstCarModelo	
Métodos	
+ estadística (form, formEst, sesion)	Controla la recogida y validación de datos para la realización de una estadística que muestra un modelo de carga, y su posterior visualización.
Responsabilidades	
<ul style="list-style-type: none"> • Base para el control de la entrada y visualización de una estadística que muestra un modelos de carga. 	

ControlWebEstCarDescriptiva	
Métodos	
+ estadística (form, formEst, sesion)	Controla la recogida y validación de datos para la realización de una estadística que muestra la descripción de la carga del servidor, y su posterior visualización.
Responsabilidades	
<ul style="list-style-type: none"> • Base para el control de la entrada y visualización de una estadística que muestra la descripción de la carga del servidor. 	

7.2.2 Paquete Lógica de Negocio

El paquete Lógica de Negocio encapsula la funcionalidad propia de la aplicación desligándola de la recogida y visualización de datos. Se encarga de realizar las acciones necesarias, y de contener y transformar los datos precisos, para dar respuesta a las operaciones que el usuario ordene ejecutar a través de la capa Presentación.

Este paquete forma la segunda capa en la organización horizontal del sistema y está dividido en otros dos Modelo Aplicación y Control Aplicación.

7.2.2.1 Paquete Modelo Aplicación

El paquete Modelo Aplicación realiza la función del componente Modelo dentro de la arquitectura principal propuesta para el sistema; contener la información necesaria para la realización de las operaciones. Parte de esta información provendrá del paquete Presentación y será preparada, por Control Aplicación, para ser introducirla en la base de datos utilizando el paquete Persistencia. Y otra parte, será recogida mediante dicho paquete Persistencia, transformada por Control Aplicación y llevada al paquete Modelo Entrada a través de Control Entrada para ser finalmente presentada al usuario mediante el paquete Vista Web. En la Ilustración 7-11 se pueden observar sus elementos y su estructura.

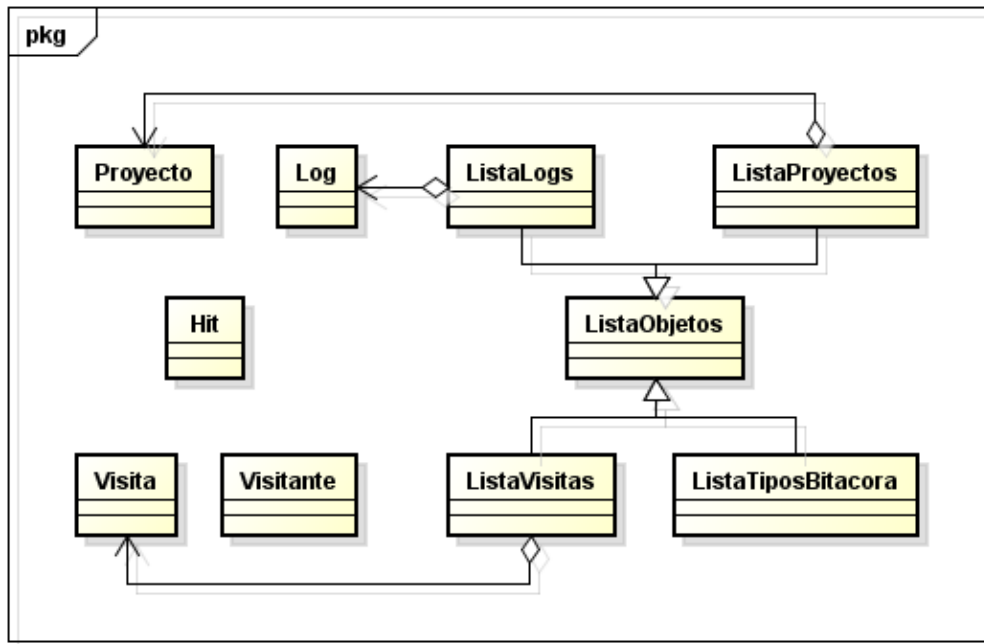


Ilustración 7-11 Paquete Modelo Aplicación

7.2.2.1.1 Descripción de clases

Proyecto	
Propiedades	
- id	Identificador del proyecto.
- nombre	Nombre del proyecto.
- organizacion	Organización a la que pertenece el proyecto.
- descripcion	Definición breve del proyecto.
Responsabilidades	
•	Contener la información de un proyecto.

Log	
Propiedades	
- id	Identificador del archivo log.
- nombre	Nombre del archivo log.
- proyecto	Proyecto al que pertenece el archivo log.
- fecha	Fecha en la que se introduce el archivo log.
- inicio	Fecha del primer hit del archivo log.
- fin	Fecha del último hit del archivo log.

Log	
- visitas	Número de visitas registradas en el archivo log.
- hits	Número de hits registrados en el archivo log.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de las propiedades de un archivo log. 	

Hit	
Propiedades	
- proyecto	Identificador del proyecto al que pertenece el hit.
- s_fecha_hora	Fecha y hora de la realización de la petición.
- s_ip	Dirección IP a la que se realiza la petición.
- cs_metodo	Forma de envío que se solicita en la petición.
- cs_uri_sistema_salida	Nombre de la página visitada anteriormente.
- cs_uri_sistema_llegada	Nombre del archivo que se solicita.
- cs_uri_query	Cadena de datos para construir dinámicamente el archivo solicitado.
- s_puerto	Número del puerto al que se realiza la petición.
- c_ip	Dirección IP desde la que se realiza la petición.
- c_RFC	Nombre del documento que especifica la norma con la que se realiza la petición.
- cs_version	Protocolo y versión de la transferencia del hit.
- cs_agente	Entidad utilizada para realizar la petición.
- cs_cookie	Cadena de datos asociados a la petición.
- cs_referencia	Página web que contiene el archivo demandado.
- cs_host	Dominio web que contiene el archivo demandado.
- sc_estado	Código del estado del envío de la petición.
- sc_subestado	Código del sub-estado del envío de la petición.
- sc_estadoWin32	Código del estado de Windows del envío de la petición.

Hit	
- sc_bytes	Número de bytes enviados del servidor al cliente.
- cs_bytes	Número de bytes enviados del cliente al servidor.
- sc_time	Tiempo de transmisión del archivo demandado.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de un hit. 	

Visita	
Propiedades	
- proyecto	Identificador del proyecto que se visita.
- ip	Dirección IP desde la que se realiza la visita.
- inicio	Fecha y hora de la primera petición de la visita.
- fin	Fecha y hora de la última petición de la visita.
- entrada	Nombre de la primera página visitada.
- salida	Nombre de la última página visitada.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de una visita. 	

Visitante	
Propiedades	
- proyecto	Identificador del proyecto al que visita.
- c_ip	Dirección IP del visitante.
- c_id_usuario	Identificador del visitante.
- cs_nombre_usuario	Nombre de usuario del visitante.
- pais	Nombre del país del visitante.
- región	Nombre de la región del visitante.
- ciudad	Nombre de la ciudad del visitante.
- latitud	Latitud, parte de las coordenadas del visitante.
- longitud	Longitud, parte de las coordenadas del visitante.

Visitante	
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de un visitante. 	

ListaObjetos	
Propiedades	
- lista[]	Lista de objetos.
Métodos	
+ getObjeto (índice)	Devuelve el objeto que ocupa la posición que indica el parámetro índice en la lista.
+ numero	Devuelve el número de objetos de la lista.
+ insertar (objeto)	Inserta el objeto pasado como parámetro al final de la lista.
+ modificar (índice, objeto)	Cambia el objeto que ocupa la posición que indica el parámetro índice en la lista por el objeto pasado como parámetro.
+ elimina (índice)	Elimina el objeto que ocupa la posición que indica el parámetro índice de la lista.
+ buscar (clave)	Devuelve la posición que ocupa en la lista el objeto cuyo clave es igual al parámetro clave.
+ escribir (archivo)	Escribe cada objeto de la lista en el archivo pasado como parámetro.
+ vaciar	Elimina todos los objetos de la lista.
Responsabilidades	
<ul style="list-style-type: none"> • Base para contener la información de una lista de objetos. 	

7.2.2.2 Paquete Control Aplicación

El paquete Control Aplicación realiza la función del componente Control dentro de la arquitectura principal propuesta para el sistema; controlar y hacer efectivas las operaciones demandadas por el usuario. La información que se recoge a través del paquete Presentación, es procesada y transformada en este paquete, con ayuda del paquete Modelo Aplicación, para ser devuelta como solución del problema al paquete Persistencia o guardada para posteriores cálculos a través del paquete Persistencia. Como se puede observar en la este paquete está compuesto por otros dos, separando los dos propósitos principales del sistema, el paquete Control Gestión y el paquete Estadísticas.

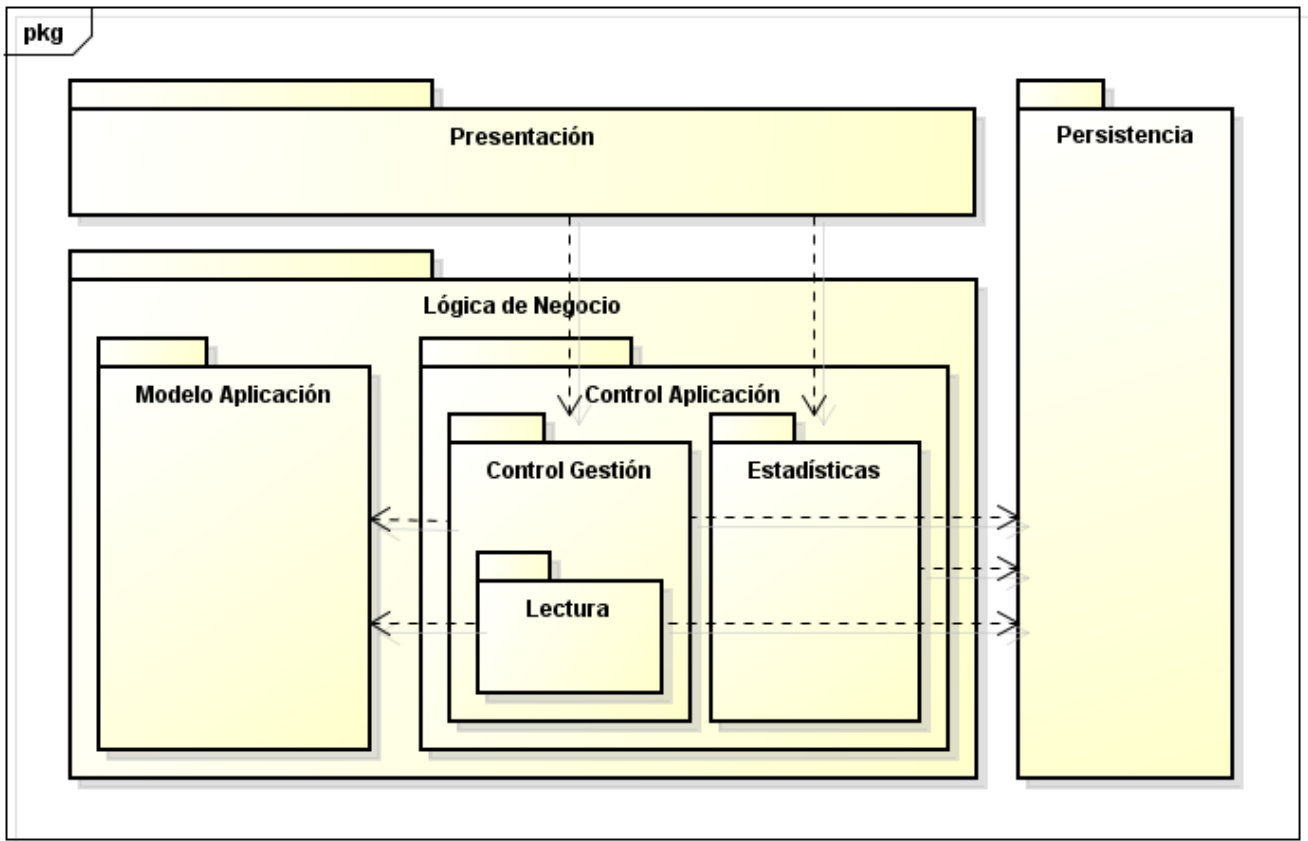


Ilustración 7-12 Paquete Control Aplicación

7.2.2.2.1 Paquete Control Gestión

El paquete Control Gestión se encarga de controlar y realizar las operaciones concernientes a la gestión de proyectos y logs. Permite la creación, modificación y eliminación de proyectos, así como su selección para la introducción de datos desde un archivo log, necesarios para la posterior realización de estadísticas. Dicha introducción de datos aunque gobernada desde el paquete Control Gestión es posible gracias a la ayuda de otro paquete llamado Lectura, que se encuentra en su interior. Esto junto al resto de su estructura se puede observar en la Ilustración 7-13.

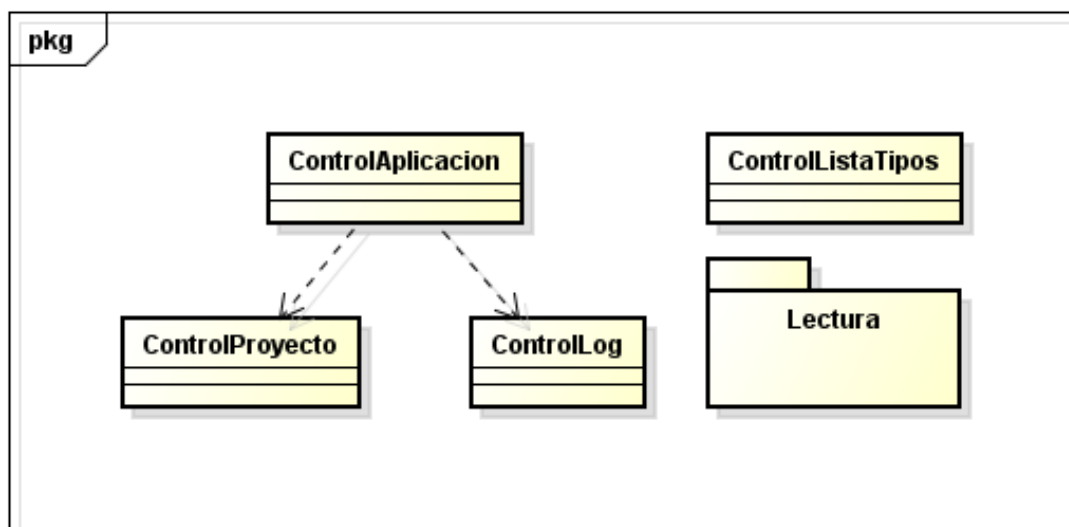


Ilustración 7-13 Paquete Control Gestión

7.2.2.2.1.1 Descripción de clases

ControlListaTipos	
Métodos	
+ cargarDatos (ListTipo)	Carga los datos de los posibles tipos de archivos log que puede leer el sistema en la lista pasada como parámetro.
Responsabilidades	
<ul style="list-style-type: none"> • Carga los datos de los posibles tipos de archivos log. 	

ControlAplicacion	
Métodos	
+ insertar (proyecto)	Controla la creación de un nuevo proyecto con los datos del parámetro introducido.
+ modificar (proyecto)	Controla la modificación de un proyecto anteriormente creado con los nuevos datos del parámetro introducido.
+ eliminar (listIdProy)	Controla la eliminación de los proyectos cuyo identificador esté en la lista pasada como parámetro.
+ comprobarDatos (archivo, tipo, solapar, idProy)	Comprueba si el archivo pasado como parámetro es del tipo pasado como parámetro y si se puede introducir en el proyecto cuyo identificador se pasa como parámetro en función de sus características, de las características de los logs ya introducidos y del parámetro solapar.
+ insertarDatos (archivo, tipo, idProy)	Controla la inserción de datos en el proyecto cuya id se pasa como parámetro, desde el archivo log pasado como parámetro, si este es del tipo pasado como parámetro.
+ eliminarDatos (listNLog, idProy)	Controla la eliminación de datos del proyecto cuyo identificador se pasa como parámetro, que se encuentren entre las fechas del conjunto de archivos log pertenecientes a dicho proyecto cuyos nombres se pasan en una lista como parámetro.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la inserción, modificación y eliminación de proyectos. • Controlar la inserción y eliminación de archivos log en un proyecto. 	

ControlProyecto	
Métodos	
+ cargarDatos (idProyecto, proyecto)	Introduce los datos del proyecto cuyo identificador se pasa como parámetro en el parámetro proyecto.
+ getIdProyecto (proyecto)	Devuelve el identificador del proyecto cuyo nombre y organización coincide con los del proyecto pasado como parámetro.
+ numProyectos (proyecto)	Devuelve el número de proyectos cuyo nombre y organización coinciden con los del proyecto pasado como parámetro.
+ numProyectosDistintos (proyecto)	Devuelve el número de proyectos cuyo nombre y organización coinciden con los del proyecto pasado como parámetro, pero que tienen diferente identificador.
+ listaProyectos (listProy)	Introduce en el parámetro la lista de proyectos del sistema con sus características.
+ insertar (proyecto)	Inserta un nuevo proyecto en el sistema con los datos del proyecto pasado como parámetro.
+ modificar (proyecto)	Modifica con los datos del proyecto pasado como parámetro los datos del proyecto del sistema cuyo identificador coincide con el de dicho parámetro.
+ borrar (idProyecto)	Elimina del sistema el proyecto cuyo identificador se pasa como parámetro y todos los datos asociados al mismo.
Responsabilidades	
<ul style="list-style-type: none"> • Hacer efectivas las operaciones de gestión de proyectos. • Contener métodos que ayuden al control de dichas operaciones. 	

ControlLog	
Métodos	
+ cargarDatos (nombre, idProyecto, log)	Introduce los datos del log del proyecto cuyo identificador se pasa como parámetro, y cuyo nombre coincide con el que se pasa como parámetro en el parámetro log.

ControlLog	
+ numeroLogs (log)	Devuelve el número de logs cuyo nombre y proyecto coinciden con los del log pasado como parámetro.
+ numeroLogsFecha (log)	Devuelve el número de logs cuyo intervalo de fechas [inicio, fin] se solapa con el intervalo de fechas del log pasado como parámetro.
+ logsFechasLimite (idProyecto)	Devuelve una lista de 2 logs. En primer lugar el log con los datos más antiguos y en segundo lugar el de los datos más recientes del proyecto cuyo identificador se pasa por parámetro.
+ listaLogs (listLog, idProy)	Introduce en el primer parámetro la lista de logs del proyecto cuyo identificador se pasa como parámetro.
+ insertar (log)	Inserta un nuevo log en el sistema con los datos del log pasado como parámetro.
+ modificar (log)	Modifica con los datos del log pasado como parámetro los datos del log del sistema cuyo nombre e identificador de proyecto coincide con el de dicho parámetro.
+ borrar (nombre, idProyecto)	Elimina todos los logs y datos del proyecto cuyo identificador se pasa como parámetro que estén en el intervalo de fechas del log cuyo nombre se pasa como parámetro.
+ borrarFechas (inicio, fin, idProyecto)	Elimina todos los logs junto con los datos del proyecto cuyo identificador se pasa como parámetro que estén en el intervalo de fechas que se pasa como parámetro.
Responsabilidades	
<ul style="list-style-type: none"> • Hacer efectivas las operaciones de gestión de los archivos log de un proyecto. • Contener métodos que ayuden al control de dichas operaciones. 	

7.2.2.2.1.2 Paquete Lectura

El paquete Lectura, como se ha comentado, contiene las clases necesarias para la lectura e inserción de los datos de los archivos log. Como se puede observar en la Ilustración 7-14, está compuesto por una serie de clases organizadas según el patrón Método Factoría, cuyo fin es la creación parametrizada de un objeto de la clase Lector. Éste, a su vez, mediante un nuevo conjunto de clases, también estructurado según el patrón Método Factoría, crea internamente un objeto de la clase bitácora para poder llevar a cabo su cometido, la lectura del archivo log.

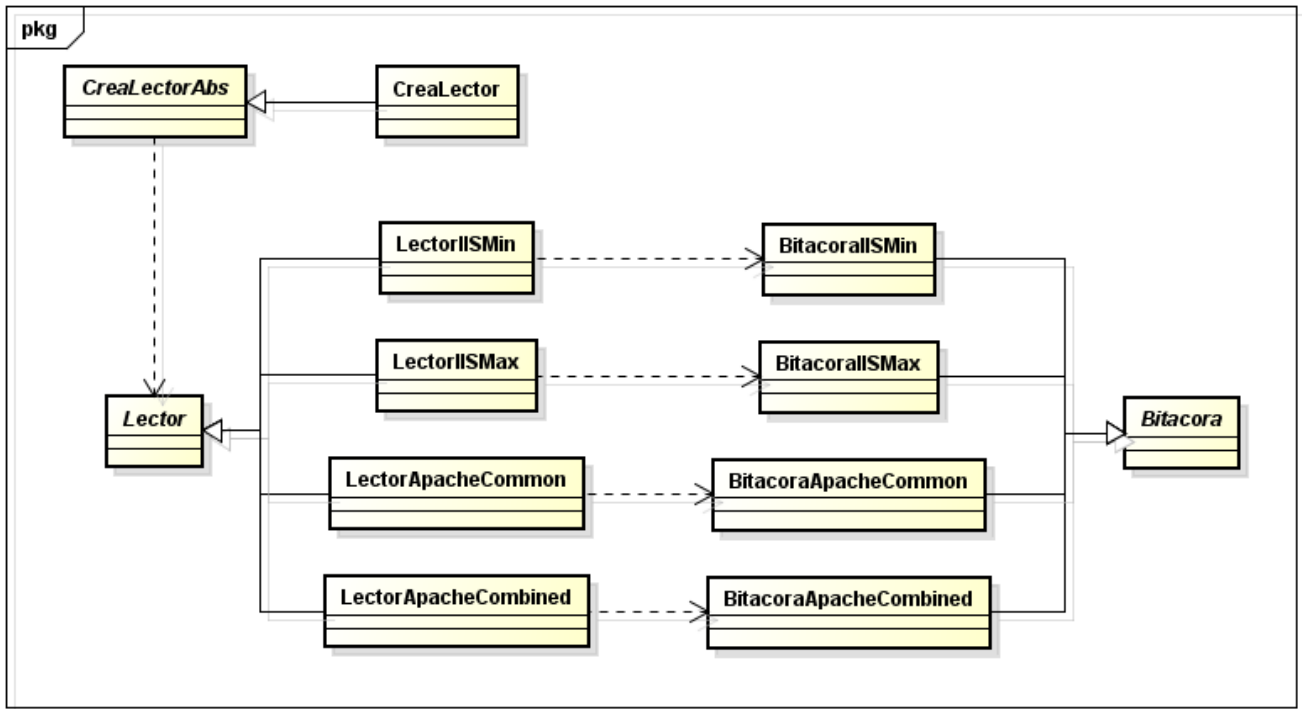


Ilustración 7-14 Paquete Lectura

7.2.2.2.1.2.1 Descripción de clases

<i>CreaLectorAbs</i>	
Métodos	
+ creaLector (tipo)	Devuelve un objeto de la clase Lector según el tipo pasado como parámetro.
Responsabilidades	
<ul style="list-style-type: none"> • Base para el control de la creación de objetos de la clase Lector. 	

<i>Lector</i>	
Métodos	
+ leer (archivo, idProyecto)	Controla el proceso de lectura del archivo log que se pasa como parámetro para incluir sus datos en el proyecto cuyo identificador coincide con el parámetro pasado.
+ leerFechas (archivo)	Devuelve una lista con dos fechas, la fecha de inicio y la fecha de fin del archivo log pasado como parámetro.
- leerBitacora (archivo, idProyecto, bitacora)	Lee el archivo log que se pasa como parámetro para incluir sus datos en el proyecto cuyo identificador coincide con el parámetro pasado, utilizando como base para las líneas el último parámetro.

<i>Lector</i>	
<p>- leerFechasBitacora (archivo, bitacora)</p>	<p>Devuelve una lista con dos fechas, la fecha de inicio y la fecha de fin del archivo log pasado como parámetro utilizando como base para las líneas el último parámetro.</p>
<p>- procesarPagina (listVisit, bitacora, idProyecto, listCerrada)</p>	<p>Procesa para el proyecto cuyo identificador se pasa como parámetro, una línea de un archivo log perteneciente a una página web pasada como parámetro en bitácora, modificando la lista de visitas abiertas y de visitas cerradas pasadas como parámetro.</p>
<p>- procesarPetición (listVisit, bitacora, idProyecto)</p>	<p>Procesa para el proyecto cuyo identificador se pasa como parámetro, una línea de un archivo log perteneciente a una petición de un archivo que no sea una página web pasada como parámetro en bitácora.</p>
<p>- procesarLog (archivo, idProyecto)</p>	<p>Devuelve un objeto de la clase Log con los datos propios del archivo log pasado como parámetro y el identificador de proyecto pasado como parámetro.</p>
<p>- esPagina (cadena)</p>	<p>Devuelve el valor True si el parámetro termina en htm, html, php, asp, psp, o py, se decir, si es el nombre de una página web.</p>
Responsabilidades	
<ul style="list-style-type: none"> • Base para el control de la lectura de archivos log creando el objeto de la clase Bitacora adecuado. 	

<i>Bitacora</i>	
Propiedades	
<p>- s_fecha_hora</p>	<p>Fecha y hora en una línea de un archivo log.</p>
<p>- c_ip</p>	<p>Dirección IP a la que se realiza la petición en una línea de un archivo log.</p>
<p>- cs_metodo</p>	<p>Forma de envío en una línea de un archivo log.</p>
<p>- cs_uri_sistema</p>	<p>Nombre del archivo que se solicita en una línea de un archivo log.</p>
<p>- cs_version</p>	<p>Protocolo y versión de la transferencia en una línea de un archivo log.</p>
<p>- sc_estado</p>	<p>Código del estado del envío en una línea de un archivo log.</p>

<i>Bitacora</i>	
- sc_bytes	Número de bytes enviados del servidor al cliente en una línea de un archivo log.
- sc_time	Tiempo de transmisión del archivo demandado en una línea de un archivo log.
Métodos	
+ setBitacora (linea)	Recoge del parámetro los datos de las propiedades de la línea de archivo log.
+ esBitacora (linea)	Comprueba si la cadena introducida como parámetro tiene el formato de línea de un tipo de archivo log.
+ esPagina	Comprueba si la clase contiene la información de una página web examinando cs_uri_sistema.
+ toHit (idProyecto)	Devuelve un objeto de la clase Hit con los valores de la línea contenida en el objeto y el identificador de proyecto pasado como parámetro.
Responsabilidades	
<ul style="list-style-type: none"> • Base para contener la información de una línea de un archivo log. • Base para recoger dicha información desde una cadena de caracteres. 	

LectorIISMin	
Métodos	
+ leer (archivo, idProyecto)	Controla el proceso de lectura del archivo log de tipo IIS Mínimo que se pasa como parámetro para incluir sus datos en el proyecto cuyo identificador coincide con el parámetro pasado.
+ leerFechas (archivo)	Devuelve una lista con dos fechas, la fecha de inicio y la fecha de fin del archivo log de tipo IIS Mínimo pasado como parámetro.
- procesarVisitante (bitacora, idProyecto)	Devuelve un objeto de la clase Visitante con los datos resultantes de procesar la línea de archivo log de tipo IIS Mínimo y el identificador de proyecto pasados como parámetros.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la lectura de archivos log en formato IIS Mínimo. 	

LectorIISMax	
Métodos	
+ leer (archivo, idProyecto)	Controla el proceso de lectura del archivo log de tipo IIS Extendido que se pasa como parámetro para incluir sus datos en el proyecto cuyo identificador coincide con el parámetro pasado.
+ leerFechas (archivo)	Devuelve una lista con dos fechas, la fecha de inicio y la fecha de fin del archivo log de tipo IIS Extendido pasado como parámetro.
- procesarVisitante (bitacora, idProyecto)	Devuelve un objeto de la clase Visitante con los datos resultantes de procesar la línea de archivo log de tipo IIS Extendido y el identificador de proyecto pasados como parámetros.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la lectura de archivos log en formato IIS Extendido. 	

LectorApacheCommon	
Métodos	
+ leer (archivo, idProyecto)	Controla el proceso de lectura del archivo log de tipo Apache Common que se pasa como parámetro para incluir sus datos en el proyecto cuyo identificador coincide con el parámetro pasado.
+ leerFechas (archivo)	Devuelve una lista con dos fechas, la fecha de inicio y la fecha de fin del archivo log de tipo Apache Common pasado como parámetro.
- procesarVisitante (bitacora, idProyecto)	Devuelve un objeto de la clase Visitante con los datos resultantes de procesar la línea de archivo log de tipo Apache Common y el identificador de proyecto pasados como parámetros.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la lectura de archivos log en formato Apache Common. 	

LectorApacheCombined	
Métodos	
+ leer (archivo, idProyecto)	Controla el proceso de lectura del archivo log de tipo Apache Combined que se pasa como parámetro para incluir sus datos en el proyecto cuyo identificador coincide con el parámetro pasado.
+ leerFechas (archivo)	Devuelve una lista con dos fechas, la fecha de inicio y la fecha de fin del archivo log de tipo Apache Combined pasado como parámetro.
- procesarVisitante (bitacora, idProyecto)	Devuelve un objeto de la clase Visitante con los datos resultantes de procesar la línea de archivo log de tipo Apache Combined y el identificador de proyecto pasados como parámetros.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la lectura de archivos log en formato Apache Combined. 	

CreaLector	
Métodos	
+ creaLector (tipo)	Devuelve un objeto de la clase Lector según el tipo pasado como parámetro.
Responsabilidades	
<ul style="list-style-type: none"> • Controlar la creación del objeto de la clase Lector adecuado. 	

BitacorallSMin	
Propiedades	
- s_ip	Dirección IP a la que se realiza la petición en una línea de un archivo log.
- c_puerto	Número del puerto al que se realiza la petición en una línea de un archivo log.
- cs_nombre_usuario	Nombre de usuario del visitante en una línea de un archivo log.
- cs_agente	Entidad utilizada para realizar la petición en una línea de un archivo log.
- sc_subestado	Código del sub-estado del envío de la petición en una línea de un archivo log.

BitacorallSMin	
Métodos	
+ setBitacora (línea)	Recoge del parámetro los datos de las propiedades de la línea de archivo log en formato IIS Mínimo.
+ esBitacora (línea)	Comprueba si la cadena introducida como parámetro tiene la estructura de línea de un archivo log en formato IIS Mínimo.
+ toHit (idProyecto)	Devuelve un objeto de la clase Hit con los valores de la línea contenida en el objeto y el identificador de proyecto pasado como parámetro.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de una línea de un archivo log en formato IIS Mínimo. • Recoger dicha información de una cadena de caracteres con dicho formato. 	

BitacorallSMax	
Propiedades	
- s_nombre_sitio	Nombre del sitio web en una línea de un archivo log.
- s_nombre_servidor	Nombre del servidor en una línea de un archivo log.
- s_ip	Dirección IP a la que se realiza la petición en una línea de un archivo log.
- cs_uri_query	Cadena de datos para construir dinámicamente el archivo solicitado en una línea de un archivo log.
- c_puerto	Número del puerto al que se realiza la petición en una línea de un archivo log.
- cs_nombre_usuario	Nombre de usuario del visitante en una línea de un archivo log.
- cs_agente	Entidad utilizada para realizar la petición en una línea de un archivo log.
- cs_cookie	Cadena de datos asociados a la petición en una línea de un archivo log.
- cs_referencia	Página web que contiene el archivo demandado en una línea de un archivo log.
- cs_host	Dominio web que contiene el archivo demandado en una línea de un archivo log.

BitacoraIISMax	
- sc_subestado	Código del sub-estado del envío de la petición en una línea de un archivo log.
- sc_estadoWin32	Código del estado de Windows del envío de la petición en una línea de un archivo log.
- cs_bytes	Número de bytes enviados del cliente al servido en una línea de un archivo log.
Métodos	
+ setBitacora (línea)	Recoge del parámetro los datos de las propiedades de la línea de archivo log en formato IIS Extendido.
+ esBitacora (línea)	Comprueba si la cadena introducida como parámetro tiene la estructura de línea de un archivo log en formato IIS Extendido.
+ toHit (idProyecto)	Devuelve un objeto de la clase Hit con los valores de la línea contenida en el objeto y el identificador de proyecto pasado como parámetro.
Responsabilidades	
<ul style="list-style-type: none"> • Contener la información de una línea de un archivo log en formato IIS Extendido. • Recoger dicha información de una cadena de caracteres con dicho formato. 	

BitacoraApacheCommon	
Propiedades	
- c_RFC	Nombre del documento que especifica la norma con la que se realiza la petición en una línea de un archivo log.
- c_id_usuario	Identificador del visitante en una línea de un archivo log.
Métodos	
+ setBitacora (línea)	Recoge del parámetro los datos de las propiedades de la línea de archivo log en formato Apache Common.
+ esBitacora (línea)	Comprueba si la cadena introducida como parámetro tiene la estructura de línea de un archivo log en formato Apache Common.
+ toHit (idProyecto)	Devuelve un objeto de la clase Hit con los valores de la línea contenida en el objeto y el identificador de proyecto pasado como parámetro.

BitacoraApacheCommon	
Responsabilidades	
•	Contener la información de una línea de un archivo log en formato Apache Common.
•	Recoger dicha información de una cadena de caracteres con dicho formato.

BitacoraApacheCombined	
Propiedades	
- c_RFC	Nombre del documento que especifica la norma con la que se realiza la petición en una línea de un archivo log.
- c_id_usuario	Identificador del visitante en una línea de un archivo log.
- cs_referencia	Página web que contiene el archivo demandado en una línea de un archivo log.
- cs_agente	Entidad utilizada para realizar la petición en una línea de un archivo log.
Métodos	
+ setBitacora (linea)	Recoge del parámetro los datos de las propiedades de la línea de archivo log en formato Apache Combined.
+ esBitacora (linea)	Comprueba si la cadena introducida como parámetro tiene la estructura de línea de un archivo log en formato Apache Combined.
+ toHit (idProyecto)	Devuelve un objeto de la clase Hit con los valores de la línea contenida en el objeto y el identificador de proyecto pasado como parámetro.
Responsabilidades	
•	Contener la información de una línea de un archivo log en formato Apache Combined.
•	Recoger dicha información de una cadena de caracteres con dicho formato.

7.2.2.2.2 Paquete Estadística

El paquete Estadística se encarga de controlar y realizar las operaciones relativas a las estadísticas. Este paquete extrae la información necesaria a través del paquete Persistencia, y junto con las opciones y datos que recibe del paquete Presentación, ejecuta los cálculos pertinentes para devolver los resultados de las estadísticas al propio paquete Presentación o almacenarlos de nuevo a través del paquete Persistencia. Si se desea ampliar la herramienta con más estadísticas este es el paquete indicado para añadir las clases que

realicen los cálculos adecuados para dichas estadísticas. En la Ilustración 7-15 se pueden observar sus elementos y su estructura.

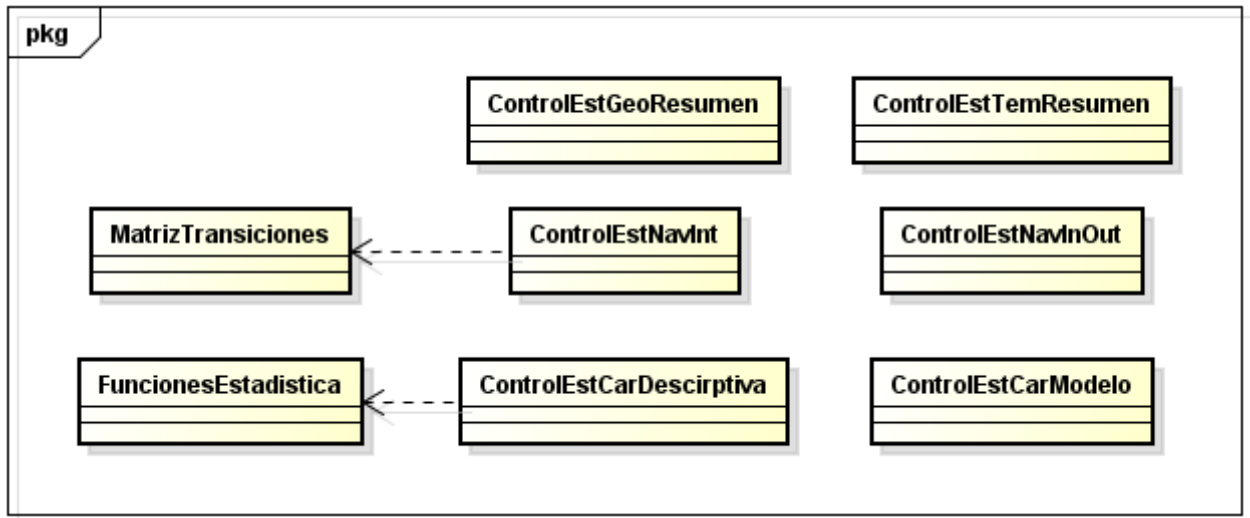


Ilustración 7-15 Paquete Estadística

7.2.2.2.1 Descripción de clases

MatrizTransiciones	
Propiedades	
- salidas[]	Nombre de los orígenes o salidas de las transiciones.
- llegadas[]	Nombre de los destinos o llegadas de las transiciones.
- transición[]	Tabla que guarda el valor de las transiciones como vectores, ordenados según los orígenes, con los valores de las transiciones, ordenados según los destinos.
- numTransiciones	Número de transiciones de la tabla.
Responsabilidades	
•	Contener la información de las probabilidades de transición entre páginas.

FuncionesEstadistica	
Métodos	
- media (a)	Devuelve la media de los valores del vector a.
- percentil (a, per)	Devuelve el percentil indicado por el parámetro per de los valores del vector a.
- varianza (a)	Devuelve la varianza del vector a.
- desviación (a)	Devuelve la desviación típica del vector a.
Responsabilidades	
<ul style="list-style-type: none"> Realizar operaciones estadísticas básicas sobre un vector. 	

ControlEstTemResumen	
Propiedades	
- idProyecto	Identificador del proyecto de la estadística.
- fechaIni	Fecha de inicio de los datos.
- fechaFin	Fecha de fin de los datos.
- agrupacion	Identificador de la agrupación temporal a utilizar.
Métodos	
+ calcularMatriz	Devuelve una matriz organizada y ordenada según agrupación con un resumen temporal de los datos del proyecto entre las fechas indicadas.
+ calcularArchivoCSV	Controla la acción de escribir en un archivo CSV un resumen temporal organizado y ordenado según agrupación de los datos del proyecto entre las fechas indicadas. Devuelve la dirección del archivo.
Responsabilidades	
<ul style="list-style-type: none"> Realizar los cálculos pertinentes para obtener un resumen temporal según una determinada agrupación de los datos de un proyecto específico entre dos fechas concretas. 	

ControlEstGeoResumen	
Propiedades	
- idProyecto	Identificador del proyecto de la estadística.
- fechaIni	Fecha de inicio de los datos.
- fechaFin	Fecha de fin de los datos.
- agrupacion	Identificador de la agrupación geográfica a utilizar.
- numMaximo	Número máximo de filas que incluir en la matriz.
Métodos	
+ calcularMatriz	Devuelve una matriz organizada y ordenada según agrupación con un resumen geográfico de los datos del proyecto entre las fechas indicadas con un número máximo de filas según numMaximo.
+ calcularArchivoCSV	Controla la acción de escribir en un archivo CSV un resumen geográfico organizado y ordenado según agrupación de los datos del proyecto entre las fechas indicadas. Devuelve la dirección del archivo.
Responsabilidades	
<ul style="list-style-type: none"> • Realizar los cálculos pertinentes para obtener un resumen geográfico según una determinada agrupación de los datos de un proyecto específico entre dos fechas concretas. 	

7.2.3 Paquete Persistencia

El paquete Persistencia es un paquete auxiliar que utilizan los dos paquetes principales del sistema para el almacenamiento de datos y su posterior recuperación. Dicho almacenamiento de datos se realiza como consecuencia de dos actividades diferentes y sobre dos medios distintos, por lo que el paquete está dividido en otros dos paquetes, Base de Datos y Archivos, que permiten dicho almacenamiento en sendos medios. Esta división se puede observar en la Ilustración 7-16.

7.2.3.1 Paquete Base de Datos

El paquete Base de Datos es el paquete que permite el almacenamiento y acceso a datos de una base de datos. Es utilizado tanto por el paquete Presentación, para la configuración de menús y opciones, como por el paquete Lógica de Negocio para la gestión y las operaciones estadísticas. Como se puede observar en la Ilustración 7-17, sus clases están estructuradas según el patrón Adapter, lo que permite utilizar cualquier tipo de base de datos; en este caso en concreto, se utiliza una base de datos MySQL.

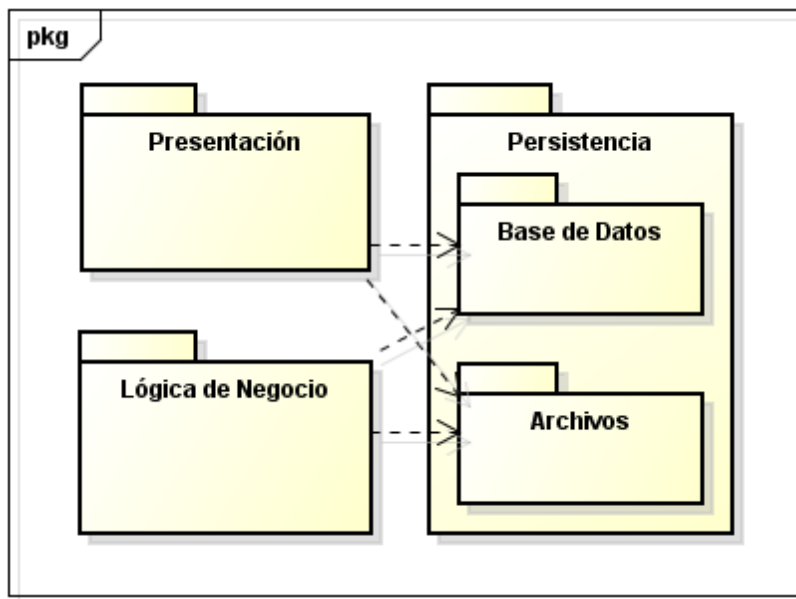


Ilustración 7-16 Paquete Persistencia

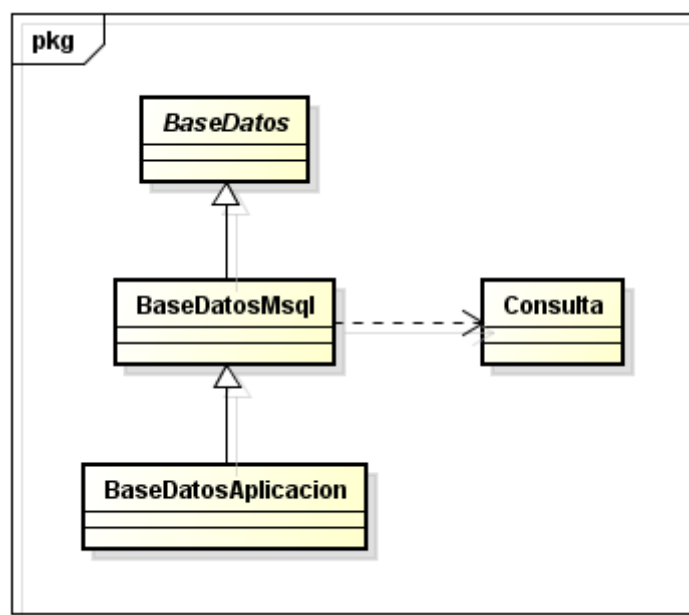


Ilustración 7-17 Paquete Base de Datos

7.2.3.1.1 Descripción de Clases

<i>BaseDatos</i>	
Propiedades	
- enlace	Enlace de conexión a la base de datos.
Métodos	
+ conectar	Conecta a la base de datos.

BaseDatos	
+ sentencia (sentencia)	Ejecuta la sentencia pasada como parámetro en la base de datos.
+ desconectar	Desconecta la base de datos.
Responsabilidades	
<ul style="list-style-type: none"> • Base para la utilización de una base de datos. 	

BaseDatosMySQL	
Propiedades	
- enlace	Enlace de conexión a la base de datos MySQL.
Métodos	
+ conectar (host, usuario, contraseña, base)	Conecta a la base de datos MySQL cuyo nombre es el parámetro base, localizada en el parámetro host, con el usuario y contraseña pasados como parámetro.
+ sentencia (sentencia)	Ejecuta la sentencia MySQL pasada como parámetro en la base de datos.
+ desconectar	Desconecta la base de datos MySQL.
Responsabilidades	
<ul style="list-style-type: none"> • Controla la utilización de una base de datos MySQL. 	

BaseDatosAplicación	
Propiedades	
- host	Localización de la base de datos de la aplicación.
- usuario	Usuario para la base de datos de la aplicación.
- contrasena	Contraseña para el usuario de dicha base de datos.
Métodos	
+ conectar (base)	Conecta la aplicación a la base de datos MySQL de la aplicación cuyo nombre es el parámetro base, con ayuda de las propiedades.
Responsabilidades	
<ul style="list-style-type: none"> • Controla la utilización de la base de datos de la aplicación. 	

Consulta	
Propiedades	
- cursor	Enlace a la respuesta de una sentencia o consulta.
Métodos	
+ numResultados	Devuelve el número de resultados o filas de la sentencia o consulta ejecutada sobre la base de datos.
+ entrada (sentencia)	Devuelve el siguiente resultado o fila de la sentencia o consulta ejecutada sobre la base de datos.
Responsabilidades	
<ul style="list-style-type: none"> Controla el acceso al contenido de una respuesta a una sentencia o consulta realizada sobre una base de datos. 	

7.2.3.2 Paquete Archivos

El paquete Base de Datos es el paquete que permite el almacenamiento y acceso a datos en diferentes archivos. Es utilizado por el paquete Presentación, como memoria de intercambio entre procesos y alternativa de presentación de estadísticas al usuario, y por el paquete Lógica de Negocio, para generar dicha alternativa de presentación. Como se puede observar en la Ilustración 7-18, todas las clases derivan de una misma clase y están estructuradas según el patrón Adapter, lo que permite utilizar cualquier sistema operativo; en este caso en concreto, se utiliza uno basado en UNIX.

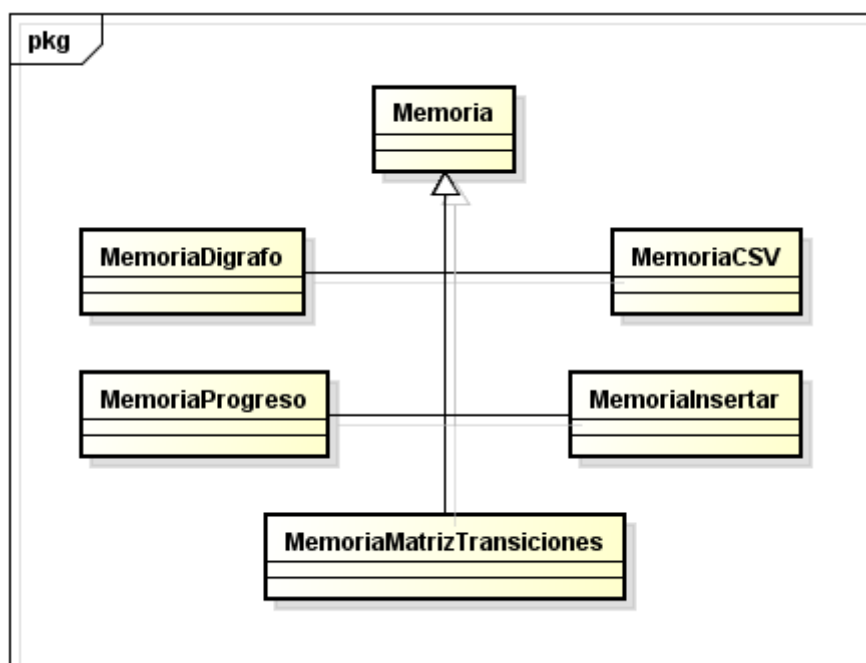


Ilustración 7-18 Paquete Archivos

7.2.3.2.1 Descripción de clases

<i>Memoria</i>	
Propiedades	
- nombre	Nombre del espacio de memoria o archivo.
Métodos	
+ crear	Crea un espacio de memoria o archivo cuyo nombre es la propiedad nombre.
+ eliminar	Elimina un espacio de memoria o archivo cuyo nombre es la propiedad nombre.
Responsabilidades	
<ul style="list-style-type: none"> • Base para la gestión y almacenamiento en espacios de memoria o archivos. 	

MemoriaDigrafo	
Métodos	
+ crear	Crea un archivo cuyo nombre es la propiedad nombre y escribe la cabecera para un dígrafo según el formato DOT.
+ escribirNodo (nodo, características)	Escribe una fila con la descripción de un nodo del dígrafo en el archivo con formato DOT.
+ escribirTransicion (origen, destino, peso)	Escribe una fila con la descripción de una transición entre el nodo del parámetro origen y el del parámetro destino con el peso del parámetro peso del dígrafo en el archivo con formato DOT.
+ cerrar	Escribe el cierre de un dígrafo en el archivo con formato DOT.
Responsabilidades	
<ul style="list-style-type: none"> • Escribir dígrafos en archivos con el formato para grafos DOT. 	

MemoriaCSV	
Métodos	
+ escribir (vector)	Escribe una fila con el contenido del vector pasado como parámetro en el archivo con formato CSV.

MemoriaCSV	
Responsabilidades	
<ul style="list-style-type: none"> • Escribir archivos con el formato CSV. 	

MemoriaInsertar	
Métodos	
+ escribir (vector)	Escribe una fila con el contenido del vector pasado como parámetro en el archivo con el formato adecuado para ser interpretado como los valores a introducir de forma masiva en una base de datos MySQL.
Responsabilidades	
<ul style="list-style-type: none"> • Escribir archivos con el formato adecuado para ser interpretados como valores a introducir de forma masiva en una base de datos MySQL. 	

MemoriaProgreso	
Métodos	
+ crear (maximo, actual)	Crea un archivo cuyo nombre es la propiedad nombre y escribe en él tres enteros: los parámetros actual y maximo, y -1.
+ existe	Comprueba si el archivo cuyo nombre es la propiedad nombre existe.
+ getMáximo	Devuelve el segundo entero del archivo cuyo nombre es la propiedad nombre.
+ getActual	Devuelve el primer entero del archivo cuyo nombre es la propiedad nombre.
+ setActual (actual)	Cambia el valor del primer entero del archivo cuyo nombre es la propiedad nombre por el del parámetro.
+ getTotal	Devuelve el tercer entero del archivo cuyo nombre es la propiedad nombre.
+ setTotal (total)	Cambia el valor del tercer entero del archivo cuyo nombre es la propiedad nombre por el del parámetro.

MemoriaProgreso	
Responsabilidades	
<ul style="list-style-type: none"> • Escribir un archivo con tres enteros que representan el número actual de elementos procesados, el número máximo de elementos procesados y el estado total del proceso. 	

MemoriaTablaTransicion	
Métodos	
+ escribir (matriz)	Escribe la tabla de transiciones pasada como parámetro en el archivo con formato CSV.
+ leer	Devuelve la tabla de transiciones escrita en el archivo.
Responsabilidades	
<ul style="list-style-type: none"> • Escribir tablas de transiciones y las lee de archivos con el formato CSV. 	

7.3 Diseño Dinámico

En este apartado se describe la forma en la que los módulos expuestos en el apartado anterior se relacionan entre sí, describiendo la secuencia temporal que muestra como interaccionan las diferentes clases que los componen y las respuestas que son visibles por el actor al interactuar con el sistema.

Para ello se utilizan diagramas de secuencia que destacan la ordenación temporal de las diferentes llamadas a procedimientos y mensajes que se envían entre las clases al ejecutar las acciones de carácter general representadas en los casos de uso identificados durante la etapa de análisis.

Estos diagramas de secuencia detallan con más precisión la descripción del modelo dinámico realizado anteriormente ayudando a comprender mejor el funcionamiento de la herramienta paso a paso.

A continuación se muestran algunas partes relevantes de alguno de estos diagramas ya que su tamaño no permite visualizarlos correctamente en el formato de esta memoria. Sin embargo se pueden consultar en el CD-ROM adjunto.

En la Ilustración 7-19 se puede observar una parte del diagrama Insertar Archivo Log. En ella se representa cómo se recogen de la base de datos los nombres de los formatos de archivos log que reconoce la herramienta, y se insertan en un objeto seleccionable que se imprimirá en la web para que el usuario pueda escoger cual es el formato de archivo log que va a introducir.

La Ilustración 7-20 representa también un fragmento del diagrama Insertar Archivo Log. En esta ocasión se puede observar la comprobación que se realiza para saber si un archivo log ya está en el sistema y si el intervalo de fechas de sus peticiones se solapa con alguna petición ya incluida en el sistema.

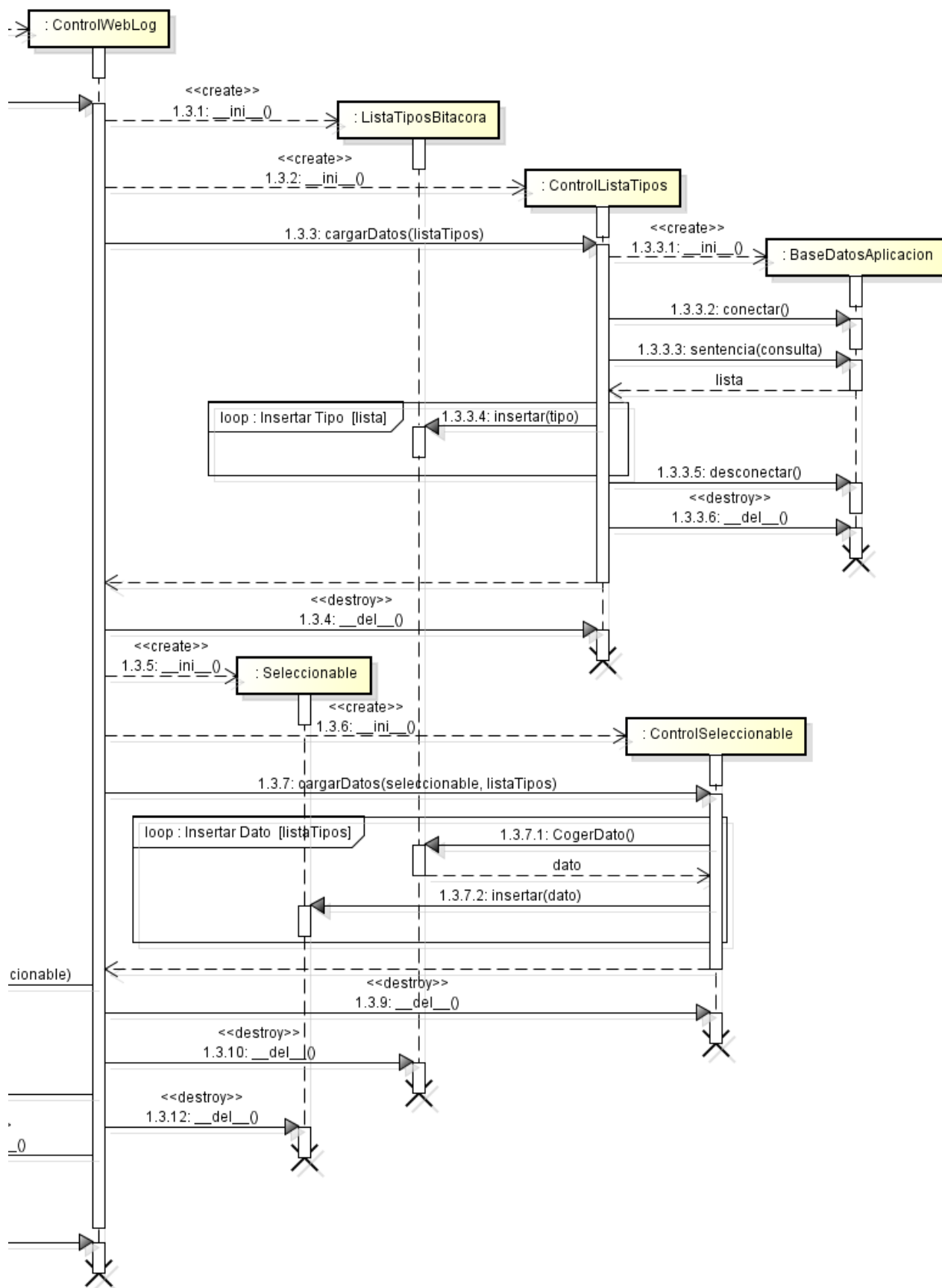


Ilustración 7-19 Diagrama de Secuencia Insertar Archivo Log - Formato de Archivo

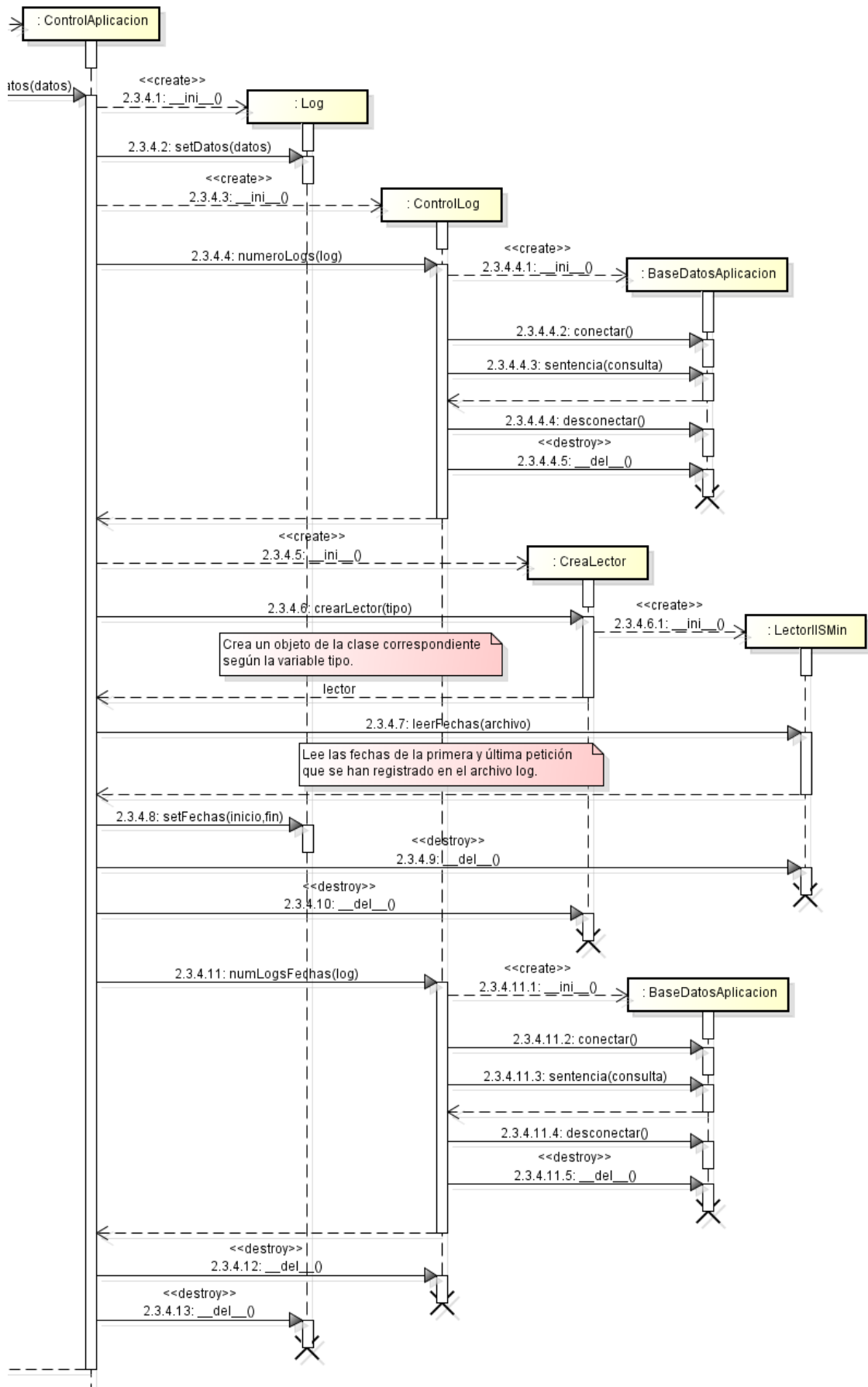


Ilustración 7-20 Diagrama de Secuencia Insertar Archivo Log - Comprobar datos

Capítulo 8 - Pruebas

8.1 Introducción

En este capítulo se recoge el conjunto de casos de prueba realizados sobre la herramienta. El objetivo de dicho conjunto es determinar el cumplimiento de los requisitos del sistema.

Aunque se realizaron pruebas individuales de los procedimientos programados y de las interacciones entre clases, a continuación sólo se describirán las pruebas de interfaz y funcionalidad enfocadas a la localización de errores o fallos en el diseño y ejecución de las acciones principales de la herramienta.

Estas pruebas fueron planificadas antes de comenzar la codificación para que ésta no influyese en la realización de las mismas. Fueron concebidas desde el punto de vista de los casos de uso detectados en la etapa de análisis y efectuadas utilizándolos como guía. Si durante alguna de ellas, se detecta un incumplimiento de los requisitos, la prueba es repetida tras la corrección pertinente, junto con el resto de pruebas a las que pudiesen afectar las modificaciones.

Con el objetivo de facilitar su lectura, las pruebas se agruparon en conjuntos de casos de prueba que se describirán mediante un nombre y su objetivo. Dentro de cada conjunto, en cada uno de los casos de prueba se indica una breve descripción, explicando que se va a realizar; el resultado obtenido; el resultado esperado; la causa del problema encontrado, si es que lo hubiese, y la solución adoptada para resolverlo.

8.2 Pruebas y Resultados

Antes de comenzar a describir cada uno de los casos de prueba conviene destacar algunas cuestiones relevantes sobre los mismos.

En primer lugar, hay que resolver el cálculo de todos los posibles casos de prueba de un proceso. Determinadas funcionalidades, como la introducción de archivos log o la visualización de estadísticas, tienen numerosas variables que dan lugar a cuantiosos casos de prueba, por lo que para reducirlos se ha realizado un diagrama del proceso y utilizado la siguiente fórmula para calcular el número de casos de prueba necesarios:

$$C = E - N + 2$$

en la que C representa el número de casos de prueba necesarios, E el número de vértices del diagrama y N su número de nodos.

En segundo lugar, comentar que los casos de prueba realizados sobre la funcionalidad de visualización de estadísticas se desglosarán y particularizarán con cada una de las estadísticas propuestas, y que para ellas se utilizarán los datos introducidos en las pruebas realizadas para verificar la funcionalidad de insertar archivos log.

En tercer lugar, indicar que se han utilizado archivos de log de todos los formatos admitidos; así como alguno que no cumple las especificaciones, para comprobar que la herramienta lo rechazaba; y que incluyen todas las peculiaridades en sus peticiones previstas durante el análisis.

Y por último, hay que tener en cuenta que si un caso de prueba da error, se dejan de realizar el resto de las pruebas relacionadas. Cuando el error es subsanado, se repite el conjunto de pruebas de la funcionalidad; y si se pasa la prueba que dio el error, se continúa hasta el final. Sin embargo, si este error era susceptible de repetirse en pruebas posteriores, no se dará y al redactar la batería pruebas aquí expuesta, quedará reflejado en los primeros casos, pero no en los posteriores.

8.2.1 Nuevo Proyecto

El **objetivo** de los casos de prueba siguientes es comprobar que el usuario puede crear diferentes proyectos de mejora de la web.

Los **casos de prueba** que se realizarán son:

1. Crear un nuevo proyecto sin ningún otro proyecto en el sistema.
2. Crear un nuevo proyecto con otros proyectos en el sistema.

Los **resultados esperados** son los siguientes:

1. El proyecto se crea satisfactoriamente y se muestran sus características.
2. En el caso de intentar crear más de un proyecto con el mismo nombre para una misma organización, el sistema no lo permitirá, avisando al usuario de ello. En el resto de los casos, el proyecto se crea satisfactoriamente y se muestran sus características.

Los **resultados obtenidos** fueron los siguientes:

1. Correcto
2. Correcto

8.2.2 Eliminar Proyecto

El **objetivo** de los casos de prueba siguientes es comprobar que el usuario puede eliminar del sistema diferentes proyectos de mejora de la web junto con sus datos asociados.

Los **casos de prueba** que se realizarán son:

1. Eliminar un proyecto del sistema.
2. Eliminar varios proyectos del sistema.

Los **resultados esperados** son los siguientes:

1. Si no hay proyectos en el sistema, se le indica al usuario. En caso contrario, el proyecto elegido se elimina satisfactoriamente, se muestra un aviso de la acción realizada y si quedan más proyectos en el sistema, se expondrán adecuadamente, si no, se le indicará al usuario que no hay proyectos.
2. Si no hay proyectos en el sistema, se le indica al usuario. En caso contrario, los proyectos elegidos se eliminan satisfactoriamente, se muestra un aviso de la acción realizada y, si quedan más proyectos en el sistema, se expondrán adecuadamente, si no, se le indicará al usuario que no hay proyectos.

Los **resultados obtenidos** fueron los siguientes:

1. Correcto
2. Correcto

8.2.3 Seleccionar Proyecto

El **objetivo** de los casos de prueba siguientes es comprobar que el usuario puede seleccionar un proyecto de mejora de la web incluido en el sistema.

Los **casos de prueba** que se realizarán son:

1. Seleccionar un proyecto del sistema, desde el inicio.
2. Seleccionar un proyecto del sistema, desde otro proyecto seleccionado.

Los **resultados esperados** son los siguientes:

1. Si no hay ningún proyecto en el sistema, se le indica al usuario. En caso contrario, el proyecto elegido por el usuario se selecciona satisfactoriamente y se muestran sus características.
2. El proyecto elegido se selecciona satisfactoriamente y se muestran sus características.

Los **resultados obtenidos** fueron los siguientes:

1. Correcto
2. Error

Las **causas de los errores** detectados fueron:

2. El sistema no cambia el proyecto seleccionado, hasta que se selecciona uno nuevo. Sin embargo, el usuario desea que el proyecto seleccionado actualmente se deseccione al volver al inicio para seleccionar otro.

Las **soluciones adoptadas** fueron las siguientes:

2. Siempre que se entra en el menú inicio, ya sea al comenzar o al continuar la ejecución después de interactuar con un determinado proyecto, si hay algún proyecto seleccionado se deselecciona.

8.2.4 Modificar Proyecto

El **objetivo** de los casos de prueba siguientes es comprobar que el usuario puede modificar los datos de un proyecto de mejora de la web creado en el sistema.

Los **casos de prueba** que se realizarán son:

1. Modificar un proyecto sin ningún otro proyecto en el sistema.
2. Modificar un proyecto con otros proyectos en el sistema.

Los **resultados esperados** son los siguientes:

1. El proyecto se modifica satisfactoriamente y se muestran sus nuevas características.
2. En el caso de intentar modificar un proyecto introduciendo el mismo nombre y siendo para la misma organización que otro, el sistema no lo permitirá y avisará al usuario de ello. En el resto de los casos, el proyecto se modifica satisfactoriamente y se muestran sus nuevas características.

Los **resultados obtenidos** fueron los siguientes:

1. Correcto
2. Correcto

8.2.5 Insertar Archivo Log

El **objetivo** de los casos de prueba siguientes es comprobar que el usuario puede insertar archivos log con diferentes formatos en uno o varios proyectos de mejora de la web creados previamente en el sistema.

Los **casos de prueba** que se realizarán son:

1. Insertar un archivo log en un proyecto sin ningún otro archivo log en el sistema.
2. Insertar un archivo log en un proyecto sin archivos log, con otros archivos log en otros proyectos del sistema.
3. Insertar un archivo log en un proyecto con varios archivos log.

Los **resultados esperados** son los siguientes:

1. Si el archivo log está en alguno de los formatos admitidos, se inserta en el proyecto bajo los supuestos especificados durante el análisis y se muestran sus características principales.
2. Si el archivo log está en alguno de los formatos admitidos, se inserta en el proyecto bajo los supuestos especificados durante el análisis y se muestran sus características principales.
3. Si el archivo log está en alguno de los formatos admitidos, se inserta en el proyecto bajo los supuestos especificados durante el análisis y se muestran sus características principales junto con las de los demás archivos log introducidos anteriormente en el proyecto.

Los **resultados obtenidos** fueron los siguientes:

1. Error
2. Correcto
3. Error

Las **causas de los errores** detectados fueron:

1. La acción de lectura y procesado del archivo log para la inserción de los datos en el sistema tarda demasiado tiempo.
3. En el caso de que las fechas de las peticiones de un archivo log se solapen con las de otro ya introducido anteriormente en el proyecto, el sistema no permite introducir el archivo. Sin embargo, el usuario desea que la herramienta le admita forzar la introducción del archivo para no perder las posibles peticiones no registradas anteriormente sin que esto perjudique a los análisis que se realicen sobre el proyecto.

Las **soluciones adoptadas** fueron las siguientes:

1. Se cambia el algoritmo del proceso, transformando ligeramente el modelo y rebajando ciertas restricciones de forma que tarde un menor tiempo pero, a la vez, se sigan cumpliendo los requisitos.
3. Se incluye una casilla de verificación que al ser marcada por el usuario le permita forzar la introducción del archivo, aunque las fechas de las peticiones que incluye se solapen con las de otro ya introducido anteriormente en el proyecto, sin duplicar las peticiones ya registradas para que no perjudique a los análisis que se realicen sobre el proyecto.

8.2.6 Eliminar Archivo Log

El **objetivo** de los casos de prueba siguientes es comprobar que el usuario puede eliminar del sistema diferentes archivos log introducidos previamente en un proyecto de mejora de la web junto con sus datos asociados.

Los **casos de prueba** que se realizarán son:

1. Eliminar las peticiones registradas entre las fechas de inicio y fin de un archivo log incluido en un proyecto.
2. Eliminar las peticiones registradas entre dos fechas cuyo intervalo este dentro de las fechas de inicio y fin (sin coincidir con él) de un archivo log incluido en un proyecto.
3. Eliminar las peticiones registradas entre dos fechas cuyo intervalo incluya peticiones añadidas gracias a diferentes archivos log incluidos en un proyecto.

Los **resultados esperados** son los siguientes:

1. Si no hay archivos log incluidos en el proyecto, se le indica al usuario. En caso contrario, todas las peticiones entre las fechas de inicio y fin registradas junto con sus datos asociados incluido el archivo

- log se eliminan satisfactoriamente; se muestra un aviso de la acción realizada y, si quedan mas archivos log asociados al proyecto, se expondrán adecuadamente, si no, se le indicará al usuario.
2. Si no hay archivos log incluidos en el proyecto, se le indica al usuario. En caso contrario, se eliminan satisfactoriamente todas las peticiones entre las fechas registradas junto con sus datos asociados, incluyendo los archivos log que hayan registrado todas sus peticiones dentro del intervalo de fechas, pero sin incluir los que hayan registrado alguna petición fuera del intervalo; se muestra un aviso de la acción realizada y se expondrán adecuadamente los archivos log no eliminados asociados al proyecto.
 3. Si no hay archivos log incluidos en el proyecto, se le indica al usuario. En caso contrario, se eliminan satisfactoriamente todas las peticiones entre las fechas registradas junto con sus datos asociados, incluyendo los archivos log que hayan registrado todas sus peticiones dentro del intervalo de fechas, pero sin incluir los que hayan registrado alguna petición fuera del intervalo; se muestra un aviso de la acción realizada y, si quedan mas archivos log asociados al proyecto, se expondrán adecuadamente, si no, se le indicará al usuario.

Los **resultados obtenidos** fueron los siguientes:

1. Correcto
2. Correcto
3. Correcto

8.2.7 Ver Estadística

Como se ha comentado anteriormente los casos de prueba realizados sobre esta funcionalidad se desglosan con cada una de las estadísticas propuestas. Se ha tomado esta decisión ya que, aunque tienen un modo de proceder parecido, cada una de ellas posee particularidades únicas, sobre todo en la forma de visualizar los resultados, que hace que sean necesarios unos casos de prueba individualizados.

Cabe destacar, que aparte de las estadísticas que se muestran como resultado, en cada una de estas pruebas también se comprueba que los archivos de datos generados sean los correctos.

8.2.7.1 Resumen Temporal

El **objetivo** de los casos de prueba siguientes es comprobar que el usuario puede realizar diversos resúmenes temporales entre distintos intervalos de fechas, con diferentes agrupaciones de tiempo y basados en los accesos a la web que se hayan introducido en un determinado proyecto.

Los **casos de prueba** que se realizarán son:

1. Realizar resumen temporal con los mismos intervalos de fechas y la misma agrupación de tiempo sobre distintos proyectos.
2. Realizar resumen temporal variando los intervalos de fechas y las agrupaciones de tiempo sobre un mismo proyecto.

Los **resultados esperados** son los siguientes:

1. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrará un resumen desglosado en cada una de las unidades de la agrupación de tiempo correspondiente, si no, se informará al usuario de que no hay datos.
2. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrará un resumen desglosado en cada una de las unidades de la agrupación de tiempo correspondiente, si no, se informará al usuario de que no hay datos.

Los **resultados obtenidos** fueron los siguientes:

1. Correcto
2. Correcto

8.2.7.2 Resumen Geográfico

El **objetivo** de los casos de prueba siguientes es comprobar que el usuario puede realizar diversos resúmenes geográficos entre distintos intervalos de fechas, con diferentes agrupaciones geográficas y basados en los accesos a la web que se hayan introducido en un determinado proyecto.

Los **casos de prueba** que se realizarán son:

1. Realizar resúmenes geográficos con los mismos intervalos de fechas y la misma agrupación geográfica sobre distintos proyectos.
2. Realizar resúmenes geográficos variando los intervalos de fechas y las agrupaciones geográficas sobre un mismo proyecto.

Los **resultados esperados** son los siguientes:

1. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrará un resumen desglosado en cada una de las unidades de la agrupación geográfica correspondiente, si no, se informará al usuario de que no hay datos.
2. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrará un resumen desglosado en cada una de las unidades de la agrupación geográfica correspondiente, si no, se informará al usuario de que no hay datos.

Los **resultados obtenidos** fueron los siguientes:

1. Correcto
2. Correcto

8.2.7.3 Entradas y Salidas

El **objetivo** de los casos de prueba siguientes es comprobar que el usuario puede realizar diversas estadísticas entre distintos intervalos de fechas basadas en la primera página visitada y la última según los accesos a la web de un determinado proyecto.

Los **casos de prueba** que se realizarán son:

1. Realizar estadísticas de entradas y salidas con los mismos intervalos de fechas sobre distintos proyectos.
2. Realizar estadísticas de entradas y salidas variando los intervalos de fechas sobre un mismo proyecto.

Los **resultados esperados** son los siguientes:

1. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrarán datos sobre las páginas más accedidas por cada visitante de la web en primer y último lugar, si no, se informará al usuario de que no hay peticiones registradas en dichas fechas.
2. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrarán datos sobre las páginas más accedidas por cada visitante de la web en primer y último lugar, si no, se informará al usuario de que no hay peticiones registradas en dichas fechas.

Los **resultados obtenidos** fueron los siguientes:

1. Correcto
2. Correcto

8.2.7.4 Perfil de Navegación

El **objetivo** de los casos de prueba siguientes es comprobar que el usuario puede realizar diversas estadísticas sobre los porcentajes de transición entre páginas en distintos intervalos de fechas y basadas en los accesos a la web que se hayan introducido en un determinado proyecto.

Los **casos de prueba** que se realizarán son:

1. Realizar estadísticas de perfil de navegación con los mismos intervalos de fechas sobre distintos proyectos.
2. Realizar estadísticas de perfil de navegación variando los intervalos de fechas sobre un mismo proyecto.

Los **resultados esperados** son los siguientes:

1. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrará un árbol cuyos nodos representarán las diferentes páginas; las conexiones, las transiciones entre páginas y el peso de las conexiones, el porcentaje de las transiciones; si no, se informará al usuario de que no hay peticiones registradas en dichas fechas.
2. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrará una tabla con los porcentajes de las transiciones entre las páginas; si no, se informará al usuario de que no hay peticiones registradas en dichas fechas.

Los **resultados obtenidos** fueron los siguientes:

1. Error
2. Correcto

Las **causas de los errores** detectados fueron:

1. La visualización del árbol es demasiado confusa para el usuario por el número de nodos que intervienen o por la desconexión entre los mismos, cuando se opta por representar solo las páginas con más accesos.

Las **soluciones adoptadas** fueron las siguientes:

1. Se opta por cambiar la representación en forma de árbol por una tabla con los porcentajes de las transiciones entre las páginas más accedidas, en la que el color del fondo de cada casilla está degradado según el porcentaje que contiene.

8.2.7.5 Carga Descriptiva

El **objetivo** de los casos de prueba siguientes es comprobar que el usuario puede realizar diversas estadísticas descriptivas sobre el número de accesos, el tamaño y el tiempo de carga cada página en distintos intervalos de fechas y basadas en los accesos a la web que se hayan introducido en un determinado proyecto.

Los **casos de prueba** que se realizarán son:

1. Realizar estadísticas descriptivas sobre el número de accesos variando los intervalos de fechas sobre un mismo proyecto.

2. Realizar estadísticas descriptivas sobre el tamaño de las páginas variando los intervalos de fechas sobre un mismo proyecto.
3. Realizar estadísticas descriptivas sobre el tiempo de carga variando los intervalos de fechas sobre un mismo proyecto.
4. Realizar estadísticas descriptivas sobre los diferentes aspectos con los mismos intervalos de fechas sobre distintos proyectos.

Los **resultados esperados** son los siguientes:

1. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrará una tabla describiendo las diferentes páginas según su número de accesos y los parámetros de orden y restricción escogidos; si no, se informará al usuario de que no hay peticiones registradas en dichas fechas.
2. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrará una tabla describiendo las diferentes páginas según su tamaño y los parámetros de orden y restricción escogidos; si no, se informará al usuario de que no hay peticiones registradas en dichas fechas.
3. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrará una tabla describiendo las diferentes páginas según su tiempo de carga y los parámetros de orden y restricción escogidos; si no, se informará al usuario de que no hay peticiones registradas en dichas fechas.
4. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrará una tabla describiendo las diferentes páginas según los parámetros escogidos; si no, se informará al usuario de que no hay peticiones registradas en dichas fechas.

Los **resultados obtenidos** fueron los siguientes:

1. Error
2. Correcto
3. Correcto
4. Correcto

Las **causas de los errores** detectados fueron:

1. Las medidas de centralización y variación mostradas no son correctas.

Las **soluciones adoptadas** fueron las siguientes:

1. Se opta por realizar los cálculos en un módulo propio en vez de utilizar una biblioteca externa para realizar estos sencillos cálculos.

8.2.7.6 Modelo de Carga

El **objetivo** de los casos de prueba siguientes es comprobar que el usuario puede realizar diversos modelos de carga en distintos intervalos de fechas según una agrupación apoyada en el número de accesos, el tamaño y el tiempo de carga de cada página, utilizando diferentes métodos, métricas y distancias de corte y basados en los accesos a la web que se hayan introducido en un determinado proyecto.

Los **casos de prueba** que se realizarán son:

1. Realizar modelos de carga con los mismos intervalos de fechas y parámetros sobre distintos proyectos.
2. Realizar modelos de carga variando los intervalos de fechas y el resto de parámetros, incluyendo todas las combinaciones posibles de métodos y métricas, sobre un mismo proyecto.

Los **resultados esperados** son los siguientes:

1. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrará un modelo de carga distinto en cada proyecto; si no, se informará al usuario de que no hay peticiones registradas en dichas fechas.
2. Si existen peticiones registradas en el proyecto dentro del intervalo de fechas, se mostrará un modelo de carga en cada prueba acorde con las fechas y parámetros introducidos; si no, se informará al usuario de que no hay peticiones registradas en dichas fechas.

Los **resultados obtenidos** fueron los siguientes:

1. Correcto
2. Correcto

Capítulo 9 - Conclusiones y Extensiones

9.1 Introducción

En este capítulo se realiza un resumen muy breve de los objetivos principales que se propusieron al comienzo del proyecto para, a continuación, realizar un estudio exhaustivo del grado de cumplimiento de dichos objetivos y de la solución obtenida, así como una valoración crítica del desarrollo de la herramienta y la metodología empleada.

Por último, se barajan diversas líneas futuras de trabajo sobre el proyecto actual que permitan mejorar y ampliar la funcionalidad del sistema.

9.2 Conclusiones

El objetivo principal del trabajo era el desarrollo de un sistema de análisis de rendimiento y planificación web dirigido a administradores encargados del mantenimiento de diversos proyectos de mejora del servicio proporcionado por un sitio web. Con dicho sistema se deberían poder describir diferentes proyectos de mejora de una manera simple, introduciendo datos en cada uno de ellos que permitiesen realizar diferentes análisis individuales, obteniendo diversas estadísticas.

Como ha podido verse a lo largo de esta memoria, se ha trabajado teniendo siempre presente este objetivo principal, intentando satisfacerlo de manera íntegra. Para ello, como se propuso en los objetivos, se ha realizado un estudio del análisis web encontrando las mejores fuentes de datos disponibles. Se han analizado los diferentes datos que podemos encontrar y derivar de los archivos log de servidor e investigado las diversas estadísticas que se pueden obtener con dichos datos.

El resultado obtenido es un Analizador Web Log capaz de proporcionar la información necesaria para optimizar el uso de una página web. Dicha herramienta permite tomar decisiones encaminadas a obtener mejoras en la usabilidad, navegabilidad, eficiencia y tiempos de respuesta cada página web, que llevarán a proveer a sus usuarios de una mejor experiencia y conllevará grandes beneficios a los prestadores del servicio que se ofrece en las mismas.

Para el desarrollo de este sistema se ha empleado UPEDU, una metodología que dado el entorno en el que se ha realizado ha resultado la más adecuada. Se podría pensar que para la realización de una herramienta semejante sería más conveniente una metodología ágil que definiese en cada interacción una estadística a realizar y la presentase al cliente al final de la misma para su confirmación. Sin embargo, dado la poca disponibilidad de los participantes y que desde un principio se tenía clara tanto la funcionalidad, como la información que se debía presentar a los usuarios; UPEDU, representando una metodología tradicional, ha permitido definir los límites del sistema desde el principio y mantener su desarrollo dentro de los mismos, indicando las fases por las que se ha ido pasando y los artefactos generados, lo que ha ayudado a su seguimiento. Además, ha logrado consolidar los conocimientos sobre esta metodología que poseía el autor gracias a anteriores proyectos.

La dedicación a tiempo parcial durante el desarrollo de este trabajo fin de grado ha sido también causa de la demora en la entrega del mismo, y se ha convertido en el único punto negativo que ha de ser destacado en estas conclusiones. Esto se ha debido, entre otras causas, a que el autor ha compaginado la realización del mismo con el desarrollo de otros proyectos de índole personal y profesional. Sin embargo, lejos de verse como algo puramente negativo, conviene decir que este hecho ha mejorado la capacidad del autor a la hora de organizar con mayor eficiencia su tiempo y disponibilidad.

Para la codificación del sistema se ha empleado Python, un idioma de programación con diversas variantes y totalmente nuevo para el autor. Tener la oportunidad de trabajar con este idioma, con el que no se ha trabajado durante la carrera, ha permitido al autor no solo ampliar sus conocimientos, si no que también le ha aportado la madurez necesaria a la hora de afrontar situaciones nuevas. Este hecho, junto al resto de

problemas afrontados durante el desarrollo del trabajo, será de gran utilidad durante su futura carrera laboral y profesional permitiéndole adaptarse a nuevos retos y afrontar las dificultades de una manera serena y objetiva.

Por otro lado, la necesidad de aplicar gran parte de los conocimientos adquiridos en la carrera para la resolución de un único problema, ha sido, no solo una manera de recordar y fijar dichos conocimientos, si no, que además, ha permitido al autor autoevaluarse en cada uno de ellos de una manera global, y no solo como conocimientos aislados en sus respectivas asignaturas.

Además, en particular, se ha profundizado en el estudio del análisis y evaluación de sistemas aplicándolo a los servidores y páginas web. Lo que ha permitido al autor ampliar los conocimientos adquiridos en la asignatura cursada durante el grado y valorar la importancia de otros conocimientos, como los estadísticos, dentro de la informática.

Como valoración final, se puede concluir que se han alcanzado los objetivos del trabajo fin de grado, marcados al comienzo, de forma satisfactoria gracias al desarrollo de la herramienta que permite realizar análisis del rendimiento y planificación web a través de archivos log.

9.3 Futuras líneas de trabajo

A continuación se detallan algunas posibles líneas futuras de trabajo sobre el proyecto que podrían añadir mejoras y nuevas funcionalidades. Estas están divididas en dos grupos, las que guardan relación con la entrada de datos y las que se refieren a la salida que proporciona el sistema.

9.3.1 Entrada de Datos

Como se ha visto, la herramienta permite la introducción de datos en los proyectos de mejora gracias a archivos log en diversos formatos. Parece lógico que una futura línea de trabajo sea extender el catálogo de formatos de archivo aceptados por la herramienta. De hecho, como se ha explicado, el diseño de la misma ha sido elaborado de forma que se pueda realizar dicha ampliación fácil y rápidamente.

Este proceso de introducción de datos es lento y tedioso para los usuarios. Por ello, otra mejora podría incluir la posibilidad de incorporar en una sola acción varios archivos log en un determinado proyecto. O, siguiendo en esta misma línea, se podrían insertar varios archivos log, cada uno en su correspondiente proyecto, sin más que indicarlo al comienzo de cada sesión de importación de datos.

Sin embargo, estas extensiones no librarían al usuario de tal indeseable tarea. Lo verdaderamente eficaz sería que el propio sistema recogiese periódica y automáticamente los archivos log sin más que configurar el lugar de donde debe hacerse y la frecuencia para cada proyecto. Incluso una mejora a esto último sería que se recogiesen los datos en tiempo real directamente del servidor sin tener que procesar sus archivos log.

Por último, para tener una mayor cantidad de datos que proporcionasen más información al usuario para mejorar la web, podrían incluirse otro tipo de recogidas, automáticas o no, que permitiesen procesar otras fuentes de datos como logs de ISP, tags o cookies.

9.3.2 Salida de Información

El propósito principal del sistema es dotar al usuario de información relevante para mejorar cada página web de la que esté encargado. Esto se realiza a través de la elaboración de diversas estadísticas según las características de los accesos. Para conseguir más información relevante o mejorar la ya existente, se podría extender la herramienta incluyendo nuevas estadísticas o formas de presentar las estadísticas actuales. En base a lo primero, su diseño ha sido elaborado de forma que se puedan incluir tantas estadísticas como se deseen de manera fácil y rápida, como se ha explicado anteriormente.

A veces, poder analizar y comparar varios sistemas a la vez, o ver las reacciones que provocan cambios diferentes en sistemas parecidos, puede resultar interesante, darnos más información sobre los mismos y llevar a conclusiones más rápidamente que si se analizan por separado. Por lo que, aunque en principio la herramienta no está pensada para ello, podría ampliarse añadiendo una sección en la que se mostrasen estadísticas que comparasen diferentes proyectos.

Sin embargo, por mucha información que tengamos, la más relevante siempre va a ser la más actualizada. Si en la herramienta se incluyera la extensión que recoge los datos en tiempo real directamente del servidor, de la que se ha hablado en el apartado anterior, podría complementarse con estadísticas también en tiempo real que mostrasen dichos datos.

Ya que esta información se requiere para realizar mejoras, una posible extensión del sistema sería que, analizada la situación de la web, la propia herramienta propusiese dichas mejoras. A demás, si incluimos datos en tiempo real y estas recomendaciones fueran aplicables automáticamente; como sucede por ejemplo, con cambios en la configuración del servidor que tiene alojada la web; la herramienta podría incluir una mejora que permitiese aplicarlas bajo consentimiento del usuario o directamente en tiempo real.

Como ayuda para tomar las decisiones necesarias para mejorar la web y como complemento a la extensión que propone dichas mejoras, podría incluirse otra que de cierto modo predijese cual sería el comportamiento de la web aplicando una u otra mejora sobre ella.

Y por último, aunando todas las extensiones, se podría ampliar la herramienta para que propusiese mejoras y las aplicase sobre los servidores teniendo en cuenta todos los proyectos a la vez. Es decir, teniendo en cuenta situaciones como que haya varias webs con proyectos de mejora incluidas en el sistema, alojadas en un mismo servidor; o si pertenecen a una misma organización, buscando el mejor rendimiento entre todas ellas y no sólo por separado.

Apéndice A - Contenido del CD

A.1 Introducción

En este apéndice se detalla el contenido del CD-ROM adjunto a esta memoria, en el que se incluyen todos los ficheros necesarios para la correcta instalación y el adecuado funcionamiento de la herramienta de análisis web desarrollada, así como, la presente documentación en formato PDF.

El árbol de carpetas que aparece en la Ilustración A-1, muestra la distribución de los distintos componentes dentro del CD-ROM. Como puede observarse, existen dos carpetas dentro del medio óptico, sendas carpetas se denominan “Documentación” y “Software”.

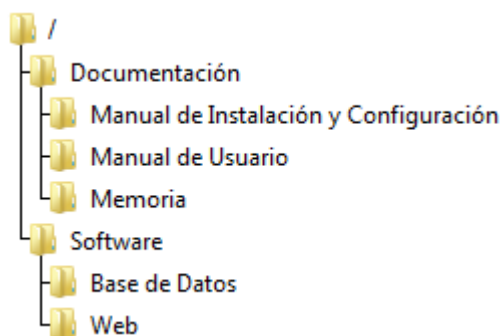


Ilustración A-1 Árbol de carpetas

Por un lado, la carpeta Documentación contiene manuales del sistema realizado, así como una copia de esta memoria y otros archivos ligados a la misma.

Por otro lado, la carpeta Software contiene los archivos necesarios para la correcta instalación de la herramienta.

A.2 Documentación

En este punto, se describe con mayor detalle el contenido de la carpeta Documentación. La cual contiene tres subcarpetas: “Manual de Instalación y Configuración”, “Manual de Usuario” y “Memoria”.

En la subcarpeta “Manual de Instalación y Configuración” se encuentra el documento donde se recogen todos y cada uno de los pasos necesarios para la correcta instalación, y posterior configuración de la herramienta de análisis en un servidor apache sobre un entorno basado en Linux. Se recomienda encarecidamente la lectura de este manual antes de comenzar el proceso de instalación.

En la subcarpeta “Manual de Usuario” se encuentra el documento donde se explica cada una de las funcionalidades y estadísticas que se pueden llevar a cabo con el sistema. Gracias a él se consigue comprender mejor los pasos precisos para realizar cualquier acción o los parámetros necesarios para visualizar cualquier estadística.

Por último, en la subcarpeta “Memoria” se encuentra el documento correspondiente a esta memoria. También hay artefactos adicionales a los que se hace referencia en este documento en una subcarpeta suya llamada “Artefactos”.

A.3 Software

En este punto, se detalla el contenido de la carpeta Software. La cual contiene dos subcarpetas: Web y Base de Datos

En la subcarpeta “Web” se encuentran los archivos web que comprenden la herramienta de análisis web. Todos ellos son necesarios para su correcta instalación y sobre algunos habrá que realizar las modificaciones pertinentes indicadas en el Manual de Instalación y Configuración para que el sistema tenga el funcionamiento adecuado y esperado.

En la subcarpeta “Base de Datos” se encuentra el archivo preciso para crear la base de datos necesaria para el correcto funcionamiento de la herramienta tal y como se detalla Manual de Instalación y Configuración.

Apéndice B - Bibliografía

- Colomé, X. (9 de Mayo de 2011). *¿Qué herramienta de analítica web necesito?* Recuperado el 15 de Diciembre de 2014, de <http://www.top-rankin.com/que-herramienta-de-analitica-web-necesito/>
- Dierbach, C. (2013). *Introduction to computer science using Python : a computational problem-solving focus*. Hoboken, New Jersey, U.S.A.: John Wiley & Sons.
- Facca, F. M. (2005). Mining interesting knowledge from weblogs: a survey. (P. d. Milano, Ed.) *Data & Knowledge Engineering*, 225–241.
- Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (2002). *Patrones de Diseño - Elementos de Software Orientado a Objetos Reutilizable*. Addison Wesley.
- Hill, T., & Lewicki, P. (Noviembre de 2005). *Big Data, Data Mining, Predictive Analytics, Statistics, StatSoft Electronic Textbook*. Recuperado el 15 de Diciembre de 2014, de <http://www.statsoft.com/Textbook>
- Kaushik, A. (2010). *Web Analytics 2.0 (The art of online Accountability & Science of Customer Centricity)*. Indianapolis, Indiana, U.S.A.: Wiley Publishing.
- Larman, C. (2003). *UML y Patrones Una introducción al análisis y diseño orientado a objetos y al proceso unificado*. Prentice Hall, Pearson Education.
- Liu, B. (2011). *Web Data Mining, Exploring Hyperlinks, Contents and Usage Data*. (M. J. Carey, Ed.) Chicago, Illinois, U.S.A.: Data-Centric Systems and Applications, Springer.
- Molero, X., Juiz, C., & Rodeño, M. J. (2010). *Evaluación y Modelado del Rendimiento de los Sistemas Informáticos*. Prentice Hall, Pearson Education.
- Rougier, N., Müller, M., Cimrman, R., & Virtanen, P. (Octubre de 2013). *Python Lecture Notes Científicas - Lecture Notes scipy*. Recuperado el 15 de Diciembre de 2014, de <http://scipy-lectures.github.io/>
- Microsoft Developer Network. (Agosto de 2005). *Log File Formats in IIS*. Recuperado el 15 de Diciembre de 2014, de [http://msdn.microsoft.com/es-es/library/cc785886\(v=ws.10\).aspx](http://msdn.microsoft.com/es-es/library/cc785886(v=ws.10).aspx)
- Real Academia Española. (Marzo de 2012). *Diccionario de la lengua española*. Recuperado el 15 de Diciembre de 2014, de <http://lema.rae.es/drae/?val=estadística>
- The Apache Software Foundation. (21 de Junio de 2014). *Log Files*. Recuperado el 15 de Diciembre de 2014, de Apache HTTP Server Version 2.4: <http://httpd.apache.org/docs/current/logs.html>
- Wikimedia Foundation. (17 de Mayo de 2014). *Análisis web - Wikipedia, la enciclopedia libre*. Recuperado el 15 de Diciembre de 2014, de http://es.wikipedia.org/wiki/An%C3%A1lisis_web