UNIVERSISTAT OBERTA DE CATALUNYA

ESTUDIOS DE INGENIERIA INFORMÁTICA

PROYECTO FINAL DE CARRERA

APLICACIONES WEB PARA TRABAJO COLABORATIVO

"Análisis de patrones de navegación de los estudiantes dentro del Campus UOC"

MEMORIA DEL PROYECTO

PEC3: **Memoria del proyecto**

Alumno: **David Herrán Castellanos**
Consultor: **Fatos Xhafa**

1. CONTROL DE VERSIONES

Versión	Fecha	Descripción
1.0	20/11/2015	Versión inicial

Contenido

1. CONTROL DE VERSIONES 2

2. RESUMEN 6

3. VALORACIONES Y DECISIONES TOMADAS 6

4. PLAN DE TRABAJO 8

4.1. DESCRIPCION DEL PROYECTO 11

4.1.1. Objetivos del Proyecto *11*

4.1.2. Resultados del Proyecto *11*

4.1.3. Análisis de Riesgos. *12*

4.2. ALCANCE DEL PROYECTO 13

4.2.1. Análisis de ficheros log del Campus Virtual. *13*

4.2.2. Estudio de LOGs de Sistemas en Servidores Apache *14*

4.2.3. Extracción y estructuración de datos de LOGs *14*

4.2.4. Adquisición de conocimiento. *14*

4.2.5. Estudio de mejoras en cuanto a posibilidades que ofrezcan los datos suministrados por los LOGS de navegación de Campus UOC. *15*

4.2.6. Estudio, procesado y exportación de datos de Logs para ser compatibles con herramientas de análisis de datos. Weka. 15

4.2.7. Estudio e implementación de cualquier otra mejora. *15*

4.3. ORGANIZACIÓN DEL PROYECTO 15

4.3.1. Relación de actividades *15*

4.3.2. Calendario de trabajo *18*

4.3.3. Hitos principales y entregables *19*

4.3.4. Equipo de trabajo *19*

4.3.5. Definición de roles *20*

4.3.6. Mecanismos de control *20*

4.4. VALORACION ECONOMICA 21

5. ESPECIFICACION Y ANALISIS 21

5.1. Problemas previos a resolver 22

5.2. Estructura de los logs del Campus Virtual 22

5.3. Patrones de navegación a analizar 26

5.4. Métodos aplicados 27

5.5. Aplicación para el procesamiento de los logs. 28

5.5.1. Aplicación seleccionada *28*

5.5.2. K-means *33*

6. Especificación del proceso de análisis de patrones de navegación 35
6.1. Establecer fuente de datos a procesar 36
6.2. Estructurar datos 36
6.3. Preprocesar datos 38
6.4. Generar fichero intermedio 40
6.4.1. Fichero intermedio Opción 1- Recursos accedidos. 40
6.4.2. Fichero intermedio Opción 2 - Periodos de tiempo de navegación. 41
6.4.3. Fichero intermedio Opción 3 - Respuestas del servidor. 42
6.4.4. Fichero intermedio Opción 4 - Tipos de recursos accedidos. 42
6.4.5. Fichero intermedio Opción 5 - Exploradores web usados. 43
6.4.6. Fichero intermedio Opción 6 - Ubicación desde donde se navega. 43
6.5. Generar fichero final 45
6.5.1. Fichero final Opción 1- Recursos accedidos 45
6.5.2. Fichero final Opción 2- Periodos de tiempo de navegación. 46
6.5.3. Fichero final Opción 3- Respuestas del servidor. 46
6.5.4. Fichero final Opción 4 -Tipos de recursos accedidos. 46
6.5.5. Fichero final Opción 5 - Exploradores web usados. 46
6.5.6. Fichero final Opción 6- Ubicación desde donde se navega. 46
6.6. Analizar datos con Weka 46
7. Estudio funcional de la aplicación 46
7.1. Casos de uso 46
7.2. DISEÑO TÉCNICO 51
7.2.1. INTRODUCCION 51
7.2.2. NECESIDADES DE ARQUITECTURA 51
7.2.3. TECNOLOGIAS 53
7.2.4. DISEÑO DE LA ARQUITECTURA 53
7.2.4.1. Entorno de producción 53
7.2.4.2. Localizador IPs 54
7.2.4.3. Entorno de programación. 58
7.2.5. Modelo de datos. 59
7.2.5.1. IP2Location_db11 59
7.2.5.2. LOG_UOC 62
7.2.5.3. Nivel de sistema de ficheros-Logs de sistema 64
7.3. DISEÑO CODIGO FUENTE 70
7.3.1. WS_IPLocator 71
7.3.2. WS_PFC 79
7.3.3. PFC (Capa de presentación) 93

8. Plan de pruebas 94
8.1. DEFINICION DE PLAN DE PRUEBAS 94
8.2. PRUEBAS DE EXPLOTACIÓN DE LA PLATAFORMA CON DATOS REALES 94
8.3. Estudio experimental 100
8.4. PRUEBA PILOTO 102
8.4.1. Preparación inicial *103*
8.4.2. SimpleKMeans *104*
9. Conclusiones y mejoras 108
***9.1.* Conclusiones** 108
***9.2.* Mejoras** 108
10. Bibliografía 109

2. RESUMEN

El **Proyecto Fin de Carrera** (en adelante PFC), es la última signatura de la Ingeniería Informatica de la Universitat Oberta de Catalunya (en adelante UOC), y en general de cualquier especialidad, sobretodo técnicas, que se curse, que trata de sintetizar todos los conocimiento adquiridos en la carrera.

El presente PFC se enmarca en el área de **Aplicaciones Web para trabajo colaborativo** y tiene por objetivo especificar un proceso para analizar los patrones de navegación de los estudiantes de la UOC en el Campus Virtual. Es decir, se centra en la minería de datos, explotación y estudio de datos registrados en diferentes formatos.

Concretamente, el estudio realizado en el marco de este PFC se centra en la explotación de datos de navegación de los estudiantes del Campus de la UOC, después de una serie de procesos, exportarlos a un formato compatible para poder estudiarlos con la herramienta Weka. "*plataforma de software para el aprendizaje automático y la minería de datos*"

https://es.wikipedia.org/wiki/Weka_(aprendizaje_automático)

En la mayor parte de los Servidores WEB, ciertas acciones que los usuarios realizan en un sitio web, son registradas en los logs del servidor en ficheros de texto; el estudio del comportamiento de los usuarios a través de lo registrado en estos logs mediante herramientas avanzadas de minería de datos combinadas con la inteligencia artificial, permite la identificación de patrones de navegación.

Tomará como referencia datos reales de los logs del Campus Virtual de la UOC y posibilitará el análisis de los patrones de navegación de los estudiantes.

Hay que tener en cuenta diferentes aspectos:

- **El soporte de la información**. La información se almacena en ficheros de texto plano en el FileSystem, que no tienen un acceso tan rápido una base de datos.
- **La estructuración de la información**. Aunque cada línea de apunte de registro del fichero log tiene una serie de campos bajo una estructura, no se puede considerar de ningún modo como datos estructurados que habrá que procesar y estructurar.
- **El volumen de datos registrado** es ingente, llegando a un ratio de más de 13 millones de registros al día. Por este motivo hay que optimizar bien el proceso y tener cuidado de no caer en bucles pseudo-infinitos poniendo límites de principio y fin.

Generando así un fichero con estructura y contenido adecuados para ser procesado por una aplicación de minería de datos. En el caso del presente PFC, la aplicación seleccionada ha sido Weka que, gracias a los algoritmos que incluye, posibilita la obtención de los patrones de navegación buscados.

3. VALORACIONES Y DECISIONES TOMADAS

De todas las ofertas de PFC disponibles, me decante por "**Aplicaciones Web para trabajo colaborativo**" por diferentes motivos.

1. Poseo experiencia en la programación web y web 2.0 y me resultará más sencillo y atractivo este tipo de programación, que otro que sea totalmente nuevo y quizás frustrante al no conseguir los objetivos deseados en el tiempo esperado.

2. Tampoco hay que obviar que este tipo de educación a distancia es adecuado para alumnos con muchas obligaciones y poco tiempo, entre los que me encuentro. De este modo, la curva de aprendizaje no me resultará tan pronunciada.
3. Uno de los puntos débiles en mi formación es la profundización en el campo de la minería de datos y la interpretación y explotación de los mismos a través de herramientas complejas. En este caso a través de ciertas áreas de la Inteligencia Artificial.
   Es por eso que en este aspecto me atraía y suponía un reto para mí el hecho de aprovechar la oportunidad para formarme en este tema.
4. Aunque ya ahondaré más en este tema en concreto, debido a que se trata de un Proyecto abierto con cierto grado de autonomía, lo he desarrollado intentando:
   a. Usar lenguajes de programación vigentes. No obsoletos. Es por eso que aunque en un principio y por dominar el lenguaje había previsto desarrollar el PFC en ASP y Visual Basic 6, me decante finalmente por el entorno de .Net desarrollándolo en Visual Basic.Net y ASP.Net.
   b. Si bien, al tener en cuenta el punto anterior, con respecto a la no obsolescencia de los lenguajes y herramientas, me quedaría con dos alternativas. Mundo .Net o mundo J2ee. En este último caso, siempre he visto más difícil de configurar y hacer que funcione correctamente el entorno de desarrollo y debugger de Java, con eclipse, ant y jboss que con Visual Studio, en donde es sumamente fácil de configurar.

   c. Otro objetivo del Proyecto es facilitar la interoperabilidad entre sistemas usando estándares abiertos y conocidos. Es por ello que se ha implementado usando Web services:
      i. Servicios web
      ii. XML
      iii. SOAP
      iv. WSDL
      v. HTTP

      Esto trae como ventajas:

      i. Aportan interoperabilidad entre aplicaciones de software independientemente de sus propiedades o de las plataformas sobre las que se instalen.
      ii. Los servicios Web fomentan los estándares y protocolos basados en texto, que hacen más fácil acceder a su contenido y entender su funcionamiento.

iii. Permiten que servicios y software de diferentes compañías ubicadas en diferentes lugares geográficos puedan ser combinados fácilmente para proveer servicios integrados.

d. Además se ha intentado aplicar los conocimientos adquiridos en la carrera como son:
   i. Paradigmas de Sistemas Distribuidos.
      1. Estándares abiertos
      2. Arquitectura en tres capas.
      3. Arquitectura en diferentes servidores
   ii. Buenas prácticas en la programación.
      1. Programación Orientada objetos
      2. Abstracción de módulos
      3. Nomenclatura "CamelCase"
      4. Poco acoplamiento, mucha cohesión entre módulos
      5. ….

http://wiki.inf.utfsm.cl/index.php?title=Buenas_Practicas_de_Programación

   iii. Registro de LOGs para mejorar el mantenimiento correctivo.
   iv. Fácil parametrización

e. Ya que, se debían procesar todos los datos de los ficheros LOG de un servidor Apache del Campus de la UOC, los cuales están en un soporte no optimo como un fichero de texto en el Sistema de Ficheros. Mi intención era hacerlo solo una vez, esto es. Que se leyera del txt la primera vez y en los posteriores usos poder explotar los datos accediendo a soportes más óptimos como una base de datos. Así, se consolidará la información leída en BD para facilitar su estudio.

4. PLAN DE TRABAJO

Se describe a continuación el plan de trabajo, y que se desarrolló en la PEC1, el cual a pesar de haber sufrido algunos cambios tecnológicos que pasaremos a detallar a continuación, a grandes rasgos permanece invariante.
Los cambios han sido los siguientes:

1. Dentro del estudio tecnológico, en el punto 6.2 se indicaba que se desplegaría el PFC en dos máquinas virtuales, emuladas bajo Virtual Box. Se ha decidido finalmente usar VMWare por disponer de más experiencia y conocimientos en esta tecnología. De todas formas, este aspecto en concreto no aporta ninguna variación al objetivo del mismo.
2. Punto 6.3. Respecto a la capa de presentación, como tecnología de desarrollo se pretendía usar ASP clásico. Está tecnología está completamente desfasada y descatalogada. Se ha reconsiderado esta opción. Se ha decidido usar una más actual, ASP.NET, aun a pesar de que no domino de la misma forma esta última.

3. Punto 6.4. La capa de negocio se planifico, en primera instancia desarrollarla con Visual Basic 6 para construir con librerías Dlls contenidas en contenedores de COM+, pero esta tecnología:
   - No cumple con estándares abiertos definidos en el punto anterior
   - Se trata de una tecnología obsoleta
   - El entorno de desarrollo no es óptimo.

   Es por ello que se ha cambiado el plan anterior y se decidió optar por la construcción de Web Services desarrollados en Visual Basic .Net que fuesen atacados por una interfaz de usuario cualquiera (en este caso una web en ASP.Net).
4. Punto 6.5. Con respecto al motor de datos, en un principio se determinó usar un motor SQL Server Express por varios motivos:
   - Licencia libre.
   - Gestion fácil
   - La comunicación .Net-SQL Server es más rápida que con otro motor, ya que, al ser ambas de Microsoft, están más optimizadas.

   Pero a la hora de la implantación tuve problemas técnicos diversos y me fue imposible desplegarlo, por ello se ha preparado el PFC para que sea compatible con dos motores de BD, SQL Server y MySQL. Únicamente habrá que definir ambas cadenas de conexión, MySQL y SQL server, y un parámetro, "TipoBD" en Web.config e indicar "SQL Server" o "MySQL".

Además de estos cambios, se han introducido algunas mejoras que aunque describiremos posteriormente más detalladamente y que añade nuevas funcionalidades no pedidas expresamente en el texto del PFC, describiremos brevemente. Estas son:

1. La ya descrita en el punto 4 anterior. Permitir que el PFC funcione atacando **a varios motores de bases de datos** fácilmente.
   Se ha implementado la aplicación para que el PFC funcione con MySQL y SQL Server siendo el funcionamiento totalmente transparente para el usuario. El Administrador del sistema lo único que tiene que tener en cuenta es indicar en la parametrización "Web.config"
   - Cadenas de Conexión a los dos motores de BD.

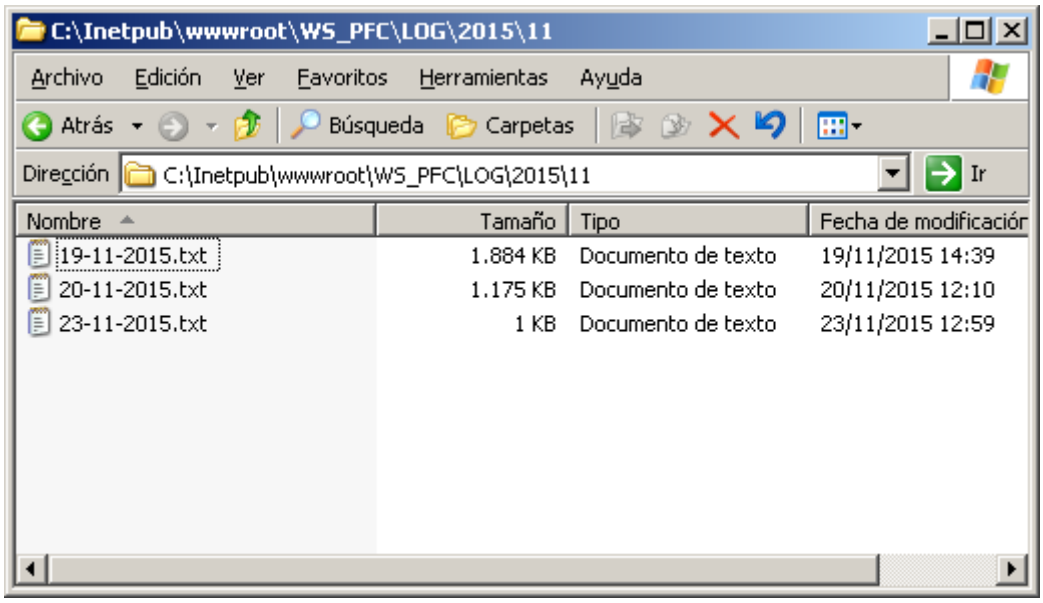
```
<connectionStrings>
    <add name="MySQLConnectionString"
connectionString="Server=localhost;Port=3306;Database=ip2location;Uid=Conexion;Pwd=xx;"/>
    <add name="SQLServerConnectionString" connectionString="Data Source= localhost;Initial
Catalog=IPLocator;Persist Security Info=False;MultipleActiveResultSets=False;User
ID=BBDD\Administrador; Password = xx" providerName="System.Data.SqlClient"/>
 </connectionStrings>
```

   - Tipo de base de datos.

```
<add key="TipoBD" value="MYSQL"/>
```

2. **Exportación de datos a XML**. Siguiendo con la política de fomentar los estándares abiertos, se ha añadido como mejora que las exportaciones de los ficheros intermedios se puedan hacer, ademas de en formato CSV, en XML.

3. **Módulo de LOGS de sistema**. Todas las aplicaciones en entornos profesionales tienen un módulo de logs que registra las acciones del usuario y del sistema con más o menos detalle en función de una parametrización, guardando en ficheros de texto nombrados con la fecha en que se realiza el apunte, una línea por cada acción.
   Esto tiene dos objetivos claros, por un lado poder depurar la aplicación adecuadamente y por otro tener una auditoria de que, como y cuando determinado usuarios realiza ciertas acciones.
   Se grabarán los logs bajo la siguiente estructura de ficheros. Dentro de la carpeta del PFC y del IPLocator se creará una carpeta LOG, por debajo de ella una con el año, por debajo una con el mes y ahí se ubicará el fichero de log dd-mm-aaaa.txt

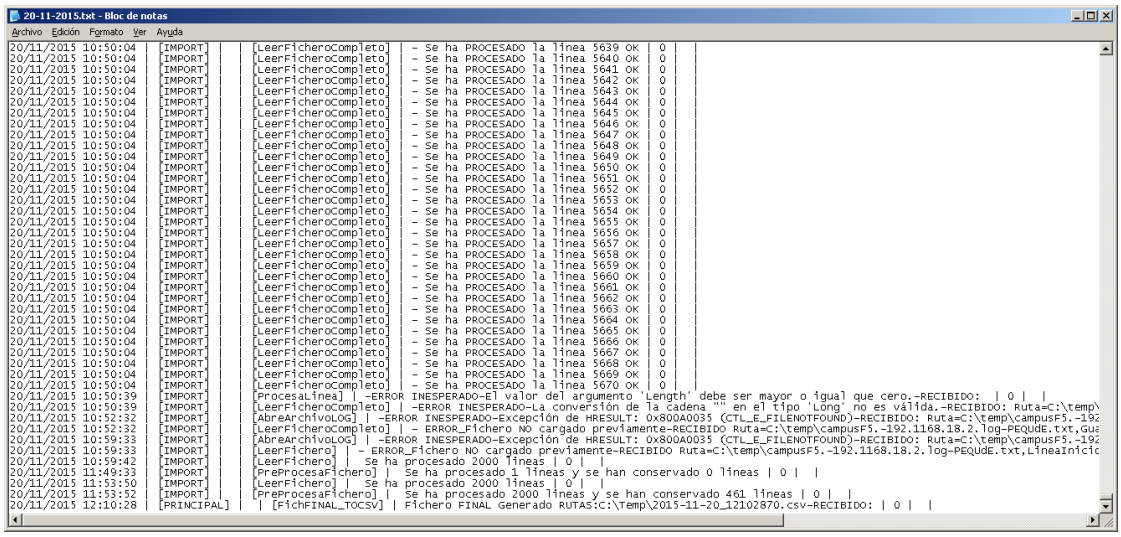


   Habrá un parámetro en Web.config, "NivelLOGS" que indica el modo en el que se encuentra la ejecución y en función del valor que se le indique, la aplicación grabará más o menos LOGs de sistema. En todo el código fuente del PFC hay repartidos Checkpoints de registro de LOG que graban valores clave del punto de la ejecución en que se encuentra o resúmenes de ejecución.

```
<add key="NivelLOGS" value="1"/>
<!--
  0 ->  SIN LOG
  1 ->  LOG POR DEFECTO
  2 ->  LOG DEPURACION
-->
```

   Los campos que se graban por cada apunte son:

- Fecha-Hora. En que se realiza el apunte.
- Modulo.
- Procedimiento.
- Descripción de la acción.
- Resultado de la operación.



```
20-11-2015.txt - Bloc de notas
Archivo  Edición  Formato  Ver  Ayuda
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5639 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5640 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5641 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5642 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5643 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5644 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5645 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5646 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5647 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5648 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5649 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5650 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5651 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5652 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5653 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5654 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5655 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5656 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5657 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5658 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5659 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5660 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5661 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5662 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5663 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5664 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5665 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5666 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5667 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5668 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5669 OK | 0 | |
20/11/2015 10:50:04 | [IMPORT] | | [LeerFicheroCompleto] | - Se ha PROCESADO la línea 5670 OK | 0 | |
20/11/2015 10:50:39 | [IMPORT] | | [ProcesaLinea] | -ERROR INESPERADO-El valor del argumento 'Length' debe ser mayor o igual que cero.-RECIBIDO:  | 0 |  |
20/11/2015 10:50:39 | [IMPORT] | | [LeerFicheroCompleto] | -ERROR INESPERADO-La conversión de la cadena "" en el tipo 'Long' no es válida.-RECIBIDO: Ruta=C:\temp\
20/11/2015 10:52:32 | [IMPORT] | | [AbreArchivoLog] | -ERROR INESPERADO-Excepción de HRESULT: 0x800A0035 (CTL_E_FILENOTFOUND)-RECIBIDO: Ruta=C:\temp\campusF5.-192
20/11/2015 10:52:32 | [IMPORT] | | [LeerFicheroCompleto] | - ERROR_Fichero NO cargado previamente-RECIBIDO Ruta=C:\temp\campusF5.-192.1168.18.2.log-PEQUdE.txt,Gua
20/11/2015 10:59:33 | [IMPORT] | | [AbreArchivoLog] | -ERROR INESPERADO-Excepción de HRESULT: 0x800A0035 (CTL_E_FILENOTFOUND)-RECIBIDO: Ruta=C:\temp\campusF5.-192
20/11/2015 10:59:33 | [IMPORT] | | [LeerFichero] | - ERROR_Fichero NO cargado previamente-RECIBIDO Ruta=C:\temp\campusF5.-192.1168.18.2.log-PEQUdE.txt,LineaInicio
20/11/2015 10:59:42 | [IMPORT] | | [LeerFichero] |  Se ha procesado 2000 líneas | 0 |  |
20/11/2015 11:49:33 | [IMPORT] | | [PreProcesaFichero] |  Se ha procesado 1 líneas y se han conservado 0 líneas | 0 |  |
20/11/2015 11:53:50 | [IMPORT] | | [LeerFichero] |  Se ha procesado 2000 líneas | 0 |  |
20/11/2015 11:53:52 | [IMPORT] | | [PreProcesaFichero] |  Se ha procesado 2000 líneas y se han conservado 461 líneas | 0 |  |
20/11/2015 12:10:28 | [PRINCIPAL] | | [FichFINAL_TOCSV] | Fichero FINAL Generado RUTA:C:\Temp\2015-11-20_12102870.csv-RECIBIDO: | 0 |  |
```

4.1. DESCRIPCION DEL PROYECTO

4.1.1. Objetivos del Proyecto

El objetivo principal es definir un aplicativo que procese los datos de navegación de los estudiantes de la UOC contenidos en los LOGs del servidor WEB Apache y los exporte a un formato compatible con herramientas de análisis (Weka) para analizar los patrones de navegación.

Una vez logrado el objetivo principal, en base a los patrones de navegación analizados, se han incorporado mejoras en cuanto a los datos a estudiar. Todos los posibles que permite la información contenida en los LOGs de navegación.

4.1.2. Resultados del Proyecto

La evolución de PFC y grado consecución se podrá determinar en función de unos resultados intermedios tangibles e intangibles, así como de un conjunto de entregables que reflejan la progresión y el conocimiento que se va adquiriendo:

1. **Entregable 1**. **PEC1**. Documento del Plan de Trabajo. Donde se especificará todo el cronograma de hitos, tareas y responsabilidades que darán una directriz clara al PFC.
2. Estudio y conocimiento de sistema de registro de LOGs de navegación de Servidor Apache.
3. **Entregable 2. PEC1.** Análisis y confección de Documento de especificación y análisis: en él se detallarán las conclusiones del análisis realizado sobre los ficheros de log del Campus Virtual de la UOC tomados como muestra, y se especificará el proceso que se seguirá para lograr los objetivos del proyecto
4. Estudio y aprendizaje de tecnologías, herramientas y decisiones de diseño. Se deberá abordar un proceso de estudio que permita tener datos e información de que tecnologías hay disponibles, pros, contras, circunstancias del PFC,… y en funcion de todo esto tomar decisiones de diseño.
5. **Entregable 3**. PEC2. Desarrollo del Proyecto. El objetivo es detallar adecuadamente las tareas que componen el Plan de Proyecto (Diseño funcional, Diseño Tecnico y Plan de Pruebas)

6. **Entregable 4**. **PEC2**. Prototipo del software. Este entregable consistirá en un prototipo del PFC que cumpla con un cierto porcentaje de cumplimiento de funcionalidad del software.
   Circunscrito al caso concreto que nos ocupa, este entregable constó de:
   - Instalación y configuración de software de virtualización de máquinas.
   - Instalación y configuración de Sistemas operativos. Servidor de datos y web.
   - Configuración de servidores de datos, aplicaciones y web.
   - Instalación y configuración de motores de bases de datos.
   - Configuración de instancias de bases de datos e importación de datos.
   - Programación, pruebas y despliegue de WS de localización de IPs.
   - Programación al 80%, pruebas y despliegue de WS del PFC.

   En total se puede decir que en este entregable, el PFC estaba cumplimentado al 95%.
7. Programación y pruebas de 5% restante del PFC (Resto de opciones de explotación de datos y capa de presentación, pagina web en ASP.NET)
8. Despliegue de aplicación en entorno de la UOC.
9. **Entregable 5**. Construcción de manuales de usuario y administrador.
10. **Entregable 6**. Ejecución de Plan de pruebas con datos reales, datos erróneos, datos frontera, tasa de crecimiento de disco,…
11. **Entregable 7**. **Memoria del PFC**. Exposición detallada del trabajo realizado en el PFC y de los resultados alcanzados
12. **Entregable 8. Presentación virtual**: documento de presentación del PFC sintetizando de forma clara y concisa el trabajo realizado y los resultados alcanzados

4.1.3. Análisis de Riesgos.

Durante el desarrollo del PFC se han detectado los siguientes riesgos.

Disponibilidad de RRHH en el desarrollo del PFC	
Id	RI-01
Descripción	El equipo de desarrollo (David Herrán Castellanos) debe compaginar sus obligaciones profesionales y domesticas con el desarrollo del PFC y su disponibilidad puede que no sea la esperada.
Impacto	Plazos de entrega
Probabilidad	Muy Alta
Acciones	1. Realizar una planificación adecuada del proyecto y un plan de trabajo que contemple esta situación. 2. Realizar un seguimiento continuo del proyecto de tal forma que se detecten rápidamente desviaciones en los plazos de desarrollo. 3. Intentar en la medida de lo posible, en determinados periodos, adelantar el trabajo por encima del Plan de Trabajo.

Periodos de inactividad	
Id	RI-02

Descripción	El responsable de ejecución del proyecto es una única persona, por lo que podría haber paradas debido a enfermedad, trabajo o problemas familiares.
Impacto	Plazos de entrega
Probabilidad	Media
Acciones	Intentar en la medida de lo posible, en determinados periodos, adelantar el trabajo por encima del Plan de Trabajo.

Periodos de inactividad	
Id	RI-03
Descripción	El dispositivo donde se están construyendo y almacenando los entregables del proyecto podría resultar dañado y provocar la pérdida de información
Impacto	Perdida de datos
Probabilidad	Baja
Acciones	1. Realizar copias de seguridad al finalizar cada jornada de trabajo en el proyecto. Dichas copias de seguridad deben estar guardadas en una ubicación externa al dispositivo en el que se está realizando el proyecto. 2. Disponer de un dispositivo alternativo en el cual continuar la realización del proyecto en caso de avería del principal.

4.2. ALCANCE DEL PROYECTO

Queda dentro del alcance del Proyecto (PFC)

1. Análisis de ficheros log del Campos Virtual de la UOC.
2. Estudio de LOGs de sistema en servidores Apache e información registrada.
3. Extracción y estructuración de datos de LOGs.
4. Procesamiento de los logs del Campus Virtual, determinación y análisis de patrones de navegación más habituales de los estudiantes
5. Estudio e implementación de mejoras en cuanto a posibilidades que ofrezcan los datos suministrados por los LOGS de navegación de Campus UOC.
6. Estudio, procesado y exportación de datos de Logs para ser compatibles con herramientas de análisis de datos. Weka.
7. Estudio e implementación de cualquier otra mejora.

4.2.1. Análisis de ficheros log del Campus Virtual.

Para poder disponer de datos reales y poder:

- Estudiar datos que ofrece los LOGs de navegación del Campus de la UOC.
- Estudiar la estructura de dichos datos.
- Poder realizar baterías de pruebas preliminares en la depuración del software.

El Consultor suministrará un conjunto de ficheros LOGs de datos reales.

4.2.2. Estudio de LOGs de Sistemas en Servidores Apache

Se deberá estudiar y analizar el sistema de registro de LOGs de los servidores Apache para determina:

1. Formato de información.
2. Que información se registra.
3. Tipos de información. Existen códigos que hay que determinar que significan.

Para ello se cuenta con una WEB de referencia suministrada por el consultor e internet como fuente de datos.

http://support.moonpoint.com/network/web/server/apache/log_format.php

4.2.3. Extracción y estructuración de datos de LOGs.

La información que se registra en los LOGs de navegación de un servidor Apache tiene dos problemas en cuanto a su explotación:

1. No está estructurada. Cada registro o apunte de información es una línea en un fichero de texto, la cual tiene diferentes campos separados entre sí de diversas formas:
    a. Espacio en blanco
    b. Espacio en blanco después de cierre
    c. Guiones
    d. Determinadas posiciones de una cadena de texto.
    e. ….

    Será necesario determinar cada una de ellas y aplicar a cada campo el método correcto para extraer su información y estructurarla.

2. El soporte de la información no es óptimo. Al estar

4.2.4. Adquisición de conocimiento.

El PFC conlleva inherentemente una fase de adquisición de conocimiento:

- Estudio de registro de LOGs de información de Servidores Apache.
- Estudio de tecnologías: Webservices, WSDLs, SOAP, WEB, .Net,..
- Estudio de herramientas: Visual Studio, WebServer IIS, MySql, SQLServer, Windows Server,Weka, …
- Estudio de algoritmos a aplicar.
- Estudio de herramienta de análisis de datos y estructuras de orígenes de datos. WEKA

Este proceso de adquisición del conocimiento se basará fundamentalmente en la localización y lectura de información en Internet dando preferencia a los artículos científicos procedentes de fuentes fiables. Como resultado de este proceso de

adquisición del conocimiento, se recopilarán un conjunto de referencias documentales que serán incluidas en la memoria final del proyecto.

4.2.5. Estudio de mejoras en cuanto a posibilidades que ofrezcan los datos suministrados por los LOGS de navegación de Campus UOC.

Tenemos que ver que posibilidades nos ofrece la información registrada en los LOGS del servidor Apache en cuanto a posibilidades de explotación y análisis de datos.
Se intentará cubrir todo el potencial posible.

4.2.6. Estudio, procesado y exportación de datos de Logs para ser compatibles con herramientas de análisis de datos. Weka.

Se deberá de estudiar que formato de datos necesita el aplicativo WEKA como fuente de datos para poder generar dicho formato en la exportación del CSV del PFC y que se pueda importar por Weka.

4.2.7. Estudio e implementación de cualquier otra mejora.

Además de la funcionalidad solicitada por el Consultor para el PFC, se implementarán ciertas mejoras que encajen dentro del ámbito del Proyecto y del tiempo estipulado para el desarrollo.
En puntos posteriores se detallarán adecuadamente.

4.3. ORGANIZACIÓN DEL PROYECTO

En este punto se va a describir la organización del Proyecto, en cuanto a las partes de que se compone, fechas de inicio de cada hito, equipo de trabajo, cronograma, etc...

4.3.1. Relación de actividades

4.3.1.1. Lanzamiento del Proyecto

Está actividad consiste en:

- Los primeros contactos que se tuvieron con el Consultor, en donde se comenzó a estudiar cada una de las propuestas de PFC ofertadas que tuvieron lugar el 16/09/2015 en el primer semestre del 2015.
- Formalización de la propuesta de PFC al Consultor que se llevó a cabo el 16/06/2015
- Confirmación de PFC asignado por parte del Consultor. 20/09/2015.

4.3.1.2. Planificación del Proyecto y análisis de requisitos

Esta actividad tendrá como resultado la **elaboración del plan del proyecto**, considerado como el primer entregable del proyecto.
La ejecución de esta actividad se dividirá en las siguientes subactividades:

1. Definición y planificación del proyecto

2. Análisis de requisitos

Duración estimada: entre el 21 de septiembre de 2015 y el 12 de octubre de 2015

DEFINICIÓN Y PLANIFICACIÓN DEL PROYECTO
Esta actividad dará comienzo con la realización de la definición y planificación del proyecto y posibilitará la producción de la primera versión del plan del proyecto. En esta primera versión del documento serán incluidos los requisitos conocidos con anterioridad a la ejecución de la siguiente subactividad.

ANÁLISIS DE REQUISITOS
Una vez realizada la definición y planificación del proyecto, se contactará con el Consultor solicitándole la información necesaria que posibilite el análisis de los requisitos del proyecto. La ejecución de esta subactividad producirá una versión ampliada del presente documento

4.3.1.3. Desarrollo del Proyecto

La ejecución de esta actividad se dividirá en las siguientes subactividades que transcurrirán en mayor o menor medida, en paralelo:

1. Adquisición del conocimiento

2. Desarrollo

Duración estimada: entre el 13 de octubre de 2015 y 14 de noviembre de 2015.

ADQUISICIÓN DEL CONOCIMIENTO
Con la ejecución de esta subactividad se pretende alcanzar el conocimiento necesario, ya sea técnico o funcional, que posibilite el correcto desarrollo del proyecto y la consecución de los objetivos definidos.

DESARROLLO
Esta subactividad (cuarto entregable) consistirá en la realización de la especificación y análisis del proyecto de forma detallada, tomando como referencia los requisitos determinados durante la actividad anterior.

Durante esta subactividad se realizará lo siguiente:

- Análisis detallado de la estructura de los logs del Campus Virtual.

- Especificación del proceso para el tratamiento de los logs del Campus Virtual y el análisis de patrones de navegación.

- Realización de una prueba piloto que permita valorar lo realizado hasta la fecha.

La prueba piloto consistirá en una versión de funcionalidades reducidas con respecto a la aplicación final. Consistirá en el procesamiento de ficheros con la aplicación de minería de datos y la determinación y análisis de los patrones de navegación de una de las seis opciones que estarán disponibles en la versión definitiva.

Además del producto software se construirá una seria de informes que documenten el citado producto software (tercer entregable). En concreto la fase de Diseño y que incorporarán:

1. Diagramas de casos de uso
2. Definición de funcionalidades
3. Diagramas de actividad
4. Descripción de mejoras
5. Definición de interfaces
6. Diseño de la capa de datos
7. Diseño de arquitectura tecnológica
8. Plan de pruebas

4.3.1.4. Entrega final

Esta actividad consistirá en la construcción y entrega de los manuales de usuario y administrador, software completo al 100%, memoria del PFC y presentación PPT.

Duración estimada: entre el 15 de noviembre de 2015 y el 21 de diciembre de 2015.

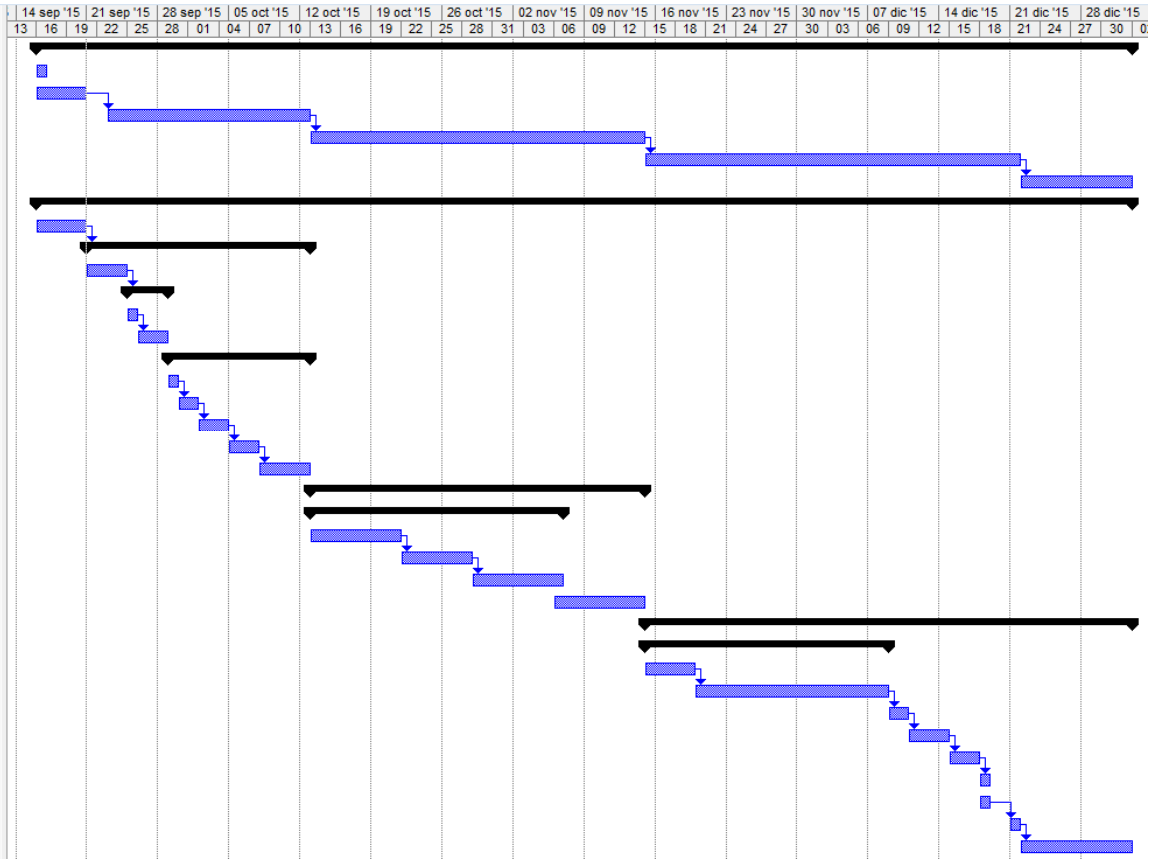
4.3.1.5. Defensa virtual

Con esta actividad se dará por finalizada la ejecución del PFC realizando la defensa virtual del mismo.

Duración estimada: entre el 22 de diciembre de 2015 y el 1 de enero de 2016.

4.3.2. Calendario de trabajo

	Nombre de tarea	Duración	Comienzo	Fin
1	**PFC- Hitos**	**81 días**	**mié 16/09/15**	**vie 01/01/16**
2	Inicio del Proyecto	1 día	mié 16/09/15	mié 16/09/15
3	Formalización de propuesta	3 días	mié 16/09/15	dom 20/09/15
4	Planificación y análisis de requisitos. PEC1	16 días	mié 23/09/15	lun 12/10/15
5	Desarrollo. PEC2	24 días	mar 13/10/15	sáb 14/11/15
6	Entrega final. PEC3	27 días	dom 15/11/15	lun 21/12/15
7	Defensa virtual	9 días	mar 22/12/15	vie 01/01/16
8	**PFC**	**81 días?**	**mié 16/09/15**	**vie 01/01/16**
9	Formalizar propuesta de Proyecto (Tarea 1)	3 días	mié 16/09/15	dom 20/09/15
10	**Planificación y análisis de requisitos. PEC1**	**18 días**	**lun 21/09/15**	**lun 12/10/15**
11	Estudio del material previo obtenido (Tarea 2.1)	4 días	lun 21/09/15	jue 24/09/15
12	**Elaboración de Planificación y Análisis de r**	**3 días**	**vie 25/09/15**	**lun 28/09/15**
13	Entrevista con el usuario/consultor (Tarea 3.	1 día	vie 25/09/15	vie 25/09/15
14	Consolidar la información (Tarea 3.2)	2 días	sáb 26/09/15	lun 28/09/15
15	**Planificación del Proyecto (Tarea 4)**	**11 días**	**mar 29/09/15**	**lun 12/10/15**
16	Descripción del proyecto (Tarea 4.1)	1 día	mar 29/09/15	mar 29/09/15
17	Definición de tareas (Tarea 4.2)	2 días	mié 30/09/15	jue 01/10/15
18	Definición de entregables (Tarea 4.3)	2 días	vie 02/10/15	dom 04/10/15
19	Planificación del proyecto (Tarea 4.4)	3 días	lun 05/10/15	mié 07/10/15
20	Definición de Requisitos (Tarea 4.5)	3 días	jue 08/10/15	lun 12/10/15
21	**Desarrollo. PEC2**	**24 días**	**mar 13/10/15**	**sáb 14/11/15**
22	**Diseño de la solucion (Tarea 6)**	**19 días**	**mar 13/10/15**	**vie 06/11/15**
23	Diseño funcional (Tarea 6.1)	7 días	mar 13/10/15	mié 21/10/15
24	Diseño técnico (Tarea 6.2)	5 días	jue 22/10/15	mié 28/10/15
25	Diseño tecnologico (Tarea 6.3)	7 días	jue 29/10/15	vie 06/11/15
26	Diseño de Plan de pruebas (Tarea 7)	6 días	vie 06/11/15	sáb 14/11/15
27	**Implementacion del codigo. PEC3**	**36 días?**	**dom 15/11/15**	**vie 01/01/16**
28	**Implementación (Tarea 8)**	**18 días**	**dom 15/11/15**	**mar 08/12/15**
29	Preparación del entorno de trabajo (Tarea 8.	5 días	dom 15/11/15	jue 19/11/15
30	Implementación del código (Tarea 8.2)	13 días	vie 20/11/15	mar 08/12/15
31	Ejecución del plan de pruebas (Tarea 9)	2 días	mié 09/12/15	jue 10/12/15
32	Despliegue. (Tarea 10)	2 días	vie 11/12/15	lun 14/12/15
33	Manuales de usuario. (Tarea 11)	3 días	mar 15/12/15	jue 17/12/15
34	Memoria final (tarea 12)	1 día?	vie 18/12/15	vie 18/12/15
35	Presentacion (Tarea 13)	1 día?	vie 18/12/15	vie 18/12/15
36	Entrega final (Tarea 14)	1 día	lun 21/12/15	lun 21/12/15
37	Defensa virtual (Tarea 15)	9 días	mar 22/12/15	vie 01/01/16



4.3.3. Hitos principales y entregables

En la siguiente tabla muestra los hitos más importantes del proyecto:

Hito	Fecha prevista	
	Fecha inicio	**Fecha fin**
Inicio del proyecto	16/09/2015	16/09/2015
Se estudian las propuestas de PFC valorándose diferentes criterios como son: - Experiencia previa del equipo de trabajo en el campo a estudia, tanto a nivel funcional como a nivel técnico. - Gustos personales del equipo de trabajo en el área propuesta - Grado en que el campo seleccionado pueda ser un reto útil al equipo de trabajo orientado a su carrera profesional.		
Formalización de propuesta	16/09/2015	20/09/2015
Se realiza la propuesta y se espera la aceptación o no aceptación de la lista de PFCs solicitados		
Planificación y análisis de requisitos. PEC1	21/09/2015	12/10/2015
Se deberá hacer un análisis de los requisitos del usuario. En nuestro caso, para cumplimentar este punto habrá que documentarse de los requisitos del usuario por dos fuentes de datos, entrevistas o preguntas al consultor, y a través de internet. Y desarrollar: - **Plan de Proyecto (entregable 1)**en el cual, en función del trabajo planificado en el punto anterior y la disponibilidad del equipo de trabajo, definir un cronograma de actividades y fechas de cumplimiento. - **Documento de especificación y análisis. (Entregable 2)**. Donde se detallarán las conclusiones de la fase de análisis.		
Desarrollo. PEC2	13/10/2015	14/11/2015
Esta fase corresponde principalmente al Diseño de un Proyecto software. Por ello constará de - Adquisición de conocimientos. Sobretodo técnicos, y en algún grado funcional. - **Desarrollo del Proyecto (Entregable 3)** Diseño y construcción de diagramas de diseño. (Casos de uso, diagramas de actividad, de secuencia,..) - **Construcción del prototipo del Proyecto (Entregable 4)**. Se trata del PFC construido a un determinado porcentaje de uso.		
Entrega final. PEC3	15/11/2015	21/12/2015
Se deberá de haber realizado el PFC en su totalidad tanto a nivel de construcción del software, como a nivel de documentaciones. - **Construcción de manuales de usuario y administrador. (Entregable 5).** Se trata de unas guías de uso desde el punto de vista del usuario y del administrador que faciliten la utilización del aplicativo. - **Implementación del software en su totalidad. (Entregable 6)**. Habrá que seguir con el desarrollo del prototipo de la etapa anterior hasta implementarlo en su totalidad - **Memoria del PFC. (Entregable 7).** Se trata de un resumen de todo lo tratado en el PFC.		
Defensa virtual	22/12/2015	01/01/2016
La defensa virtual del PFC consistirá en responder a todas las cuestiones acerca del PFC que se pudiesen dar		

4.3.4. Equipo de trabajo

El equipo de trabajo se compone de cuatro personas:

- ***David Herrán Castellanos***. Director de Proyecto, Analista, Diseñador y programador.
- ***Fatos Xhafa***. Consultor del Proyecto que en un entorno real realizará el rol de usuario de la aplicación y será la persona que facilitará requisitos, materiales y validará entregables.
- ***Santi Caballé Llobet***. Consultor de Proyecto nivel 2.
- ***David Gañan Jiménez***. Responsable de sistemas. Será el encargado de facilitar plataforma hardware y software base para el correcto funcionamiento del PFC.

4.3.5. Definición de roles

Hay que tener en cuenta que esta no será la asignación de roles en un entorno real, aquí nos encontramos en un entorno académico en el que prácticamente todos los roles recaen en el alumno. Si nos encontrásemos en un entorno profesional, el trabajo estaría más distribuido.

Así pues, David Herrán Castellanos asumirá los roles de (jefe de proyecto, analista...).

El Consultor tendrá el rol de cliente por tratarse de un 'emisor' de requisitos y ser responsable de la aceptación final de lo realizado. A la par, el Consultor asumirá responsabilidades de control y seguimiento del proyecto.

Quedando la distribución de roles queda de la siguiente manera:

En la siguiente tabla muestra los hitos más importantes del proyecto:

Rol	Persona asignada
Cliente	Fatos Xhafa
Control y seguimiento	Fatos Xhafa, Santi Caballé Llobet
Validación de resultados	Fatos Xhafa, Santi Caballé Llobet
Jefe de Proyecto	David Herrán Castellanos
Analista de Proyecto	David Herrán Castellanos
Diseñador de Proyecto	David Herrán Castellanos
Programador	David Herrán Castellanos
Responsable de Pruebas	David Herrán Castellanos
Responsable de Sistemas	David Gañan Jiménez

4.3.6. Mecanismos de control

Se considera importante establecer mecanismos de control que detecten desviaciones respecto al Plan de Proyecto a tenor de los riesgos definidos en el punto 4.1.3 o de otros nuevos que pudieran darse a lo largo del desarrollo del Proyecto.

Sirva como propuesta los siguientes:

1. Establecer semanalmente un control que confronte las fechas de realización y el grado de cumplimiento determinado en el Plan de Proyecto con las reales que se están cumpliendo.
2. Será obligatorio para la validación del PFC, el envío y visto bueno de los entregables definidos en el punto anterior dentro del plazo indicado.

4.4. VALORACION ECONOMICA

Dado el entorno académico en el que nos encontramos, asumimos la valoración económica del Proyecto como nula. Aun así se puede diferenciar:

Concepto		Importe
COSTES DE PERSONAL		
Jefe de Proyecto		0 €
Analista de Proyecto		0 €
Diseñador de Proyecto		0 €
Programador		0 €
Responsable de Pruebas		0 €
Responsable de Sistemas		0 €
TOTAL		**0 €**
COSTES DE INFRAESTRUCTURA		
Adquisición de Hardware	*Asumido por UOC*	0 €
Licencia emulador de hardware	*VMWare. Ver. de evaluación*	0 €
Licencia software base (Sistema operativo)	*Windows Server. Ver de evaluación*	0 €
Licencia software base (Servidor web)	*IIS. Ver de evaluación*	0 €
Licencia motor de base de datos	*MySQL. Licencia freeware*	0 €
Licencia Entorno de desarrollo	*Visual Studio. Ver. evaluación*	0 €
TOTAL		**0 €**

5. ESPECIFICACION Y ANALISIS

En este apartado se detallan las conclusiones del análisis de requisitos que se ha nutrido de las diferentes fuentes de información de que disponemos:

- Entrevistas con el cliente. Consultor.
- Análisis de información de LOGs de navegación de Campus Virtual de la UOC
- Consulta de información y documentación a través de Internet y otras fuentes.

Además de esto, se describe el proceso que se seguirá durante el desarrollo del proyecto para llegar al objetivo final.

5.1. Problemas previos a resolver

Durante la fase de análisis del Proyecto el analista y Jefe de Proyecto se han encontrado con los siguientes problemas previos que habrán de ser resueltos:

1. **Cantidad ingente de datos**. El volumen de datos registrado es ingente, llegando a un ratio de más de 13 millones de registros al día, con un espacio en disco de 6GB. Por este motivo hay que optimizar bien el proceso y tener cuidado de no caer en bucles pseudo-infinitos. Por este motivo se establecerá el mecanismo de poner límites de principio y fin al proceso, pudiendo de esta forma analizar datos de la línea x a la línea y del fichero LOG. Otra opción es que sea el propio usuario el que establezca subconjuntos de datos de estos LOGs e introduzca ficheros de entrada de datos más pequeños.

2. **El soporte de la información**. La información de los Logs de navegación de los usuarios del Campus Virtual de la UOC es generada por un servidor Web Apache y se almacena en ficheros de texto plano en el FileSystem, este soporte, evidentemente, no es óptimo para la explotación de la información, no tienen un acceso tan rápido una base de datos. Es por ello que se deberá generar operaciones específicas de extracción y estructuración de la información.

3. En el caso del **contenido**, si bien los ficheros de log siguen una estructura definida, esta estructura no permite extraer de forma directa los patrones de navegación cuyo análisis es el objeto del presente PFC; por ello será necesario realizar un **preprocesamiento** de dichos ficheros de log. Es por ello que se ha considerado dentro del alcance del presente PFC, la construcción de una aplicación que posibilite la realización de este preprocesamiento.

4. **La estructuración de la información**. Aunque cada línea de apunte de registro del fichero log tiene una serie de campos bajo una estructura, no se puede considerar de ningún modo como datos estructurados que habrá que procesar y estructurar.

5.2. Estructura de los logs del Campus Virtual

Los ficheros de log del Campus Virtual de la UOC, generados por el servidor Web Apache, están formados por millones de líneas soportados en ficheros de texto, uno por día, donde cada una de ellas representa la operación efectuada por un determinado estudiante dentro del Campus Virtual. Entre otra información, estas líneas registran la dirección IP del dispositivo desde el que se ha accedido y no el nombre de usuario del estudiante, lo que asegura una mayor privacidad.

```
campusF5.-192.1168.18.2.log-PEQUE.txt
1 [10/Mar/2012:00:07:34 +0100] 85.61.166.47 "GET
/rb/inici/user_modul/library/2066261.js?features=library:setprefs:dynamic-heights=778e8632a0e43c08cb5f0e0c71e29955a7ded842e233b00e3d1745c96c43c225b1fda17a420d4f92c30f80d52aee9b4413f81a1f7f8af773866c55e16e1d9473 HTTP/1.1" 200 "
http://cv.uoc.edu/webapps/widgetsUOC/widgetsRssServlet?up_items=5&up_numDias=120&up_title=Mis%20tareas&up_maximized=true&up_novaFinestra=false&up_tipusBustia=old&up_sortable=false&up_rssUrlServiceAPI=http%253A%252F%252Fcv.uoc.edu%2F
2 [10/Mar/2012:00:07:34 +0100] 192.168.1.150 "GET /webapps/campusGateway/agents/e6fa4d2d4fbd7af06c107f1822cc7aeb2652db8cc443070b1f5444d25264c8b1d825058e198bd7f591c3e8a909b39bb6ef1f51da7d6433b06be3884f082e42e4/profiles.xml
HTTP/1.1" 200 "-" "-" 967 11
3 [10/Mar/2012:00:07:34 +0100] 79.150.91.165 "POST /webapps/webUtil/servlet/InvitationChatServlet HTTP/1.1" 200 "
http://cv.uoc.edu/cgi-bin/titol?is=1&w=ws=m4429af02266fa557720708f5ac39b60fd79814861d76445761f41293f877f65c4641d4272b8424beb8ce1b8fa6000b862b9a78e55c5fd1634da8a5705a63371" "Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1;
Trident/4.0; GTB7.3; .NET CLR 2.0.50727)" 3 1
4 [10/Mar/2012:00:07:32 +0100] 83.36.108.225 "GET /UOC/mc-icons/fotos/btwose.jpg HTTP/1.1" 200 "
http://cv.uoc.edu/webapps/classroom/student.do?nav=participants-aula&domainId=346798&domainCode=112_16_615_01&preview=0&idLang=ca&s=d5284e5d0de216eed1d8a0ff913dec3aee22e3f4c37dd361f04b23794e9d2d11f5cb93643715dc08843553ad16f530e266
acc0bcfe285a63ec3c4f1a152ec0c2" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; Trident/4.0; FunWebProducts; SIMBAR={94BE3635-C0DE-4F5D-9A0A-F5548B9482A0}; GTB7.3; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; .NET CLR
3.5.30729; InfoPath.2; OfficeLiveConnector.1.5; OfficeLivePatch.1.3; .NET CLR 3.0.30729; .NET4.0C; AskTbORJ/5.14.1.20007)" 4240 1620
5 [10/Mar/2012:00:07:34 +0100] 88.17.55.151 "POST /rb/inici/tips/random?format=text HTTP/1.1" 302 "
http://cv.uoc.edu/rb/inici/grid?s=5fa4d2d4fbd7af06c107f1822cc7aeb2652db8cc443070b1f5444d25264c8b1d825058e198bd7f591c3e8a909b39bb6ef1f51da7d6433b06be3884f082e42e4" "Mozilla/4.0" 110 112
6 [10/Mar/2012:00:07:34 +0100] 95.61.32.133 "GET
/rb/inici/user_modul/library/1338391.js?features=library:setprefs:dynamic-heights=36fc742de330c2a2808e2e3bc89f3af282e7a9e83e9206d87e9b2af4231ad32d4c06c7502ffa99169e7b7c2c49b4457b8cbe35c602a2b0ef837dca77e1a3a0ad HTTP/1.1" 200 "
http://cv.uoc.edu/webapps/widgetsUOC/widgetsFotoServlet?up_title=Avui%2520destaquem&up_maximized=true&up_rssUrlServiceAPI=http%253A%252F%252Fcv.uoc.edu%252Fwebapps%252FDat2RSS%252FdatFotoPerfilServlet&up_ck=ad&up_target=imatgeDia
Estudiant.jpg&lib=/rb/inici/javascripts/prototype.js,/rb/inici/javascripts/effects.js,/rb/inici/javascripts/application.js,/rb/inici/javascripts/prefs.js,%2Frb%2Finici%2Fuser_modul%2Flibrary%2F1338391.js%3Ffeatures%3Dlibrary%3Ase
tprefs%3Adynamic-heights%26fromCampus=true&lang=ca&country=ES&color=&userType=UOC-ESTUDIANT-6-ass=36fc742de330c2a2808e2e3bc89f3af282e7a9e83e9206d87e9b2af4231ad32d4c06c7502ffa99169e7b7c2c49b4457b8cbe35c602a2b0ef837dca77e1a3a0ad"
"Mozilla/5.0 (X11; U; linux x86_64; es-ES; rv:1.9.2.10) Gecko/20100914 SUSE/3.6.10-0.3.1 Firefox/3.6.10" 2351 21
7 [10/Mar/2012:00:07:34 +0100] 192.168.1.146 "GET /webapps/campusGateway/agents/5cd9f2b3fff08e9cd7f288461ef9071079d6b20b3d0c7c0ebcaeb77ec8c04db5260e517dbaf3f2c27ee9698cc900f9c76a750cafea8fefcf8d93051afcca9c70/profiles/current.xml
HTTP/1.1" 200 "-" "-" 478 12
```

Se muestra un ejemplo de los 7 primeros registros (acciones realizadas por usuarios) de un fichero LOG de ejemplo

ANALISIS DE DATOS DE ENTRADA

En este apartado deberemos analizar los datos de entrada de los que se nutrirá nuestra aplicación (PFC). Debiendo responder a las siguientes preguntas:

1. Formato de información.
2. Que información se registra.
3. Tipos de información. Existen códigos que hay que determinar que significan.

Los diferentes apuntes o registros (acciones de cada usuario) almacenados en los LOGs, son generados por el servidor web del Sitio Web de la UOC desplegado en Apache, que almacena las acciones de los usuarios en filas (una por acción) de un fichero de texto correspondiente al día en que se realiza el apunte (al día siguiente se genera otro fichero).

Los datos generados en cada línea o acción, si bien no están estructurados, siguen una pauta que será necesario estudiar y analizar y que definiremos en el siguiente punto.

Una vez hecho esto, habrá que aplicar técnicas de minería de datos que permitan estructurar primero y posteriormente analizar dichos datos.

Pasaremos ahora a definir la pauta que siguen los registros de los ficheros LOGs.

Como referencia se ha usado este enlace.
http://support.moonpoint.com/network/web/server/apache/log_format.php

5.2.1. Estructura de la línea

Se trata de un fichero en formato texto con n líneas (dependiendo del tráfico de servidor WEB, normalmente puede generar millones de líneas) y cada vez que un usuario realiza una acción en la página WEB (en este caso la de la UOC) se grabará una línea con los siguientes datos.

Es la siguiente:

[<Fecha y hora registro>] <Dirección IP> "<método petición> <ruta recurso solicitado> <protocolo>" <código respuesta> "<ruta origen>" "<navegador>" <n. º bytes obtenido> <p1>

DESCRIPCIÓN DE LOS ELEMENTOS DE UNA LÍNEA

Cada una de las líneas representa la solicitud de acceso a un recurso realizada desde una determinada dirección IP y está formada por los siguientes elementos [16] separados por espacios:

- **fecha y hora registro**: Momento temporal en el que se ha registrado la operación. Su formato es el siguiente: <día>/<mes>/<año>:<hora>:<minuto>:<segundo> <huso horario>. Dónde:

- **día**: Día de registro de la operación en formato dd.
- **mes**: Mes de registro de la operación en formato mmm (los meses se representan con tres letras; por ejemplo marzo se representa como 'Mar').
- **año**: Año de registro de la operación en formato aaaa (cuatro dígitos de año).
- **hora**: Hora de registro de la operación en formato hh.
- **minuto**: Minuto de registro de la operación en formato MM.
- **segundo**: Segundo de registro de la operación en formato ss.
- **huso horario**: Modificador horario según el Tiempo Medio de Greenwich (GMT).

o **dirección IP**: Dirección IP del dispositivo desde el que se ha realizado la acción registrada en el log.

o **método petición**: Cada una de las líneas de los logs del Campus Virtual lleva asociado un tipo de operación identificado por el método de petición utilizado (POST, GET, HEAD...). Se hablará de los métodos de petición un poco más adelante en este mismo documento.

o **ruta recurso solicitado**: Ruta del Campus Virtual a la que se ha accedido, acompañada de los parámetros necesarios para realizar la operación demandada.

o **protocolo**: Protocolo utilizado para acceder al recurso. Por ejemplo: HTTP/1.1.

o **código respuesta**: Código de respuesta obtenido tras el acceso al recurso.

o **ruta origen**: Dirección desde la que se ha realizado la solicitud del recurso.

o **navegador**: Información identificativa del navegador desde el que se ha accedido al recurso.

o **n.º bytes obtenido**: total de bytes recibidos al acceder al recurso. Por ejemplo, en caso de haber accedido correctamente a un fichero, el valor que aquí figuraría sería el tamaño del mismo.

o **p1**: el log del Campus Virtual de la UOC dispone de un último parámetro que no ha conseguido identificarse. No obstante, dado que su relevancia es nula para el presente estudio, únicamente vamos a limitarnos a señalar su existencia.

Un ejemplo de LOG seria el siguiente:

```
[10/Mar/2012:00:07:34 +0100] 87.217.126.24 "GET /continguts/XW06_72001_01542/web/imgc/nav_on.gif HTTP/1.1" 404 "http://materials.cv.uoc.edu/continguts/XW06_72001_01542/web/main/m0/v5_4.html" "Mozilla/4.0 (compat
[10/Mar/2012:00:07:34 +0100] 188.77.244.239 "GET /webapps/rac/img/logonourac.gif HTTP/1.1" 200 "http://cv.uoc.edu/webapps/rac/listEstudiant.action?s=abd1d5405f20ea32f9539ee1431813581dac1b6bc0c83bdb3dc30250062fde
[10/Mar/2012:00:07:34 +0100] 88.15.228.149 "GET /webapps/widgetsUOC/widgetsBannerServlet?up_title=Publicitat&up_xmlUrlServiceAPI=http%253A%252F%252Fcv.uoc.edu%252FUOC%252Fa%252Finici%252Fbanner%252Fbanner.xml&up
[10/Mar/2012:00:07:34 +0100] 83.60.13.167 "POST /rb/inici/login/check HTTP/1.1" 200 "http://cv.uoc.edu/rb/inici/grid?s=5ae84ab8f4e40c6d3c34bb3d2d46ae235d30e57e0c01e1cf1339d858de71f58181d24e34b968fb277829bd02162b
[10/Mar/2012:00:07:34 +0100] 192.168.1.146 "GET /webapps/campusGateway/agents/347e4a4ae892bde6691c631f5bcfd5edb52f61006c4a285c5dd8796d1db6edc2671d6ed9c0ce0a8006bfb1c882c92fcab7541ce36dac1c3017200cfbca3b0b2e/prof
[10/Mar/2012:00:07:34 +0100] 95.21.65.168 "GET /rb/inici/tips/show/15.text HTTP/1.1" 200 "http://cv.uoc.edu/rb/inici/grid" "Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; Trident/4.0; Mozilla/4.0 (compatible
[10/Mar/2012:00:07:34 +0100] 192.168.1.132 "GET /webapps/campusGateway/agents/628fb7d34fe070dd9e825ae8cc77fb731a0786817b5973d5a5e3540dfa7711acd9d763cf2bcbc22d5ea5c5b3c19c48da193eb383141fa2946367c2cb139331d0/prof
[10/Mar/2012:00:07:34 +0100] 188.77.244.239 "GET /webapps/rac/img/cerrar_01.gif HTTP/1.1" 200 "http://cv.uoc.edu/webapps/rac/listEstudiant.action?s=abd1d5405f20ea32f9539ee1431813581dac1b6bc0c83bdb3dc30250062fde
[10/Mar/2012:00:07:34 +0100] 87.217.126.24 "GET /continguts/XW06_72001_01542/web/imgc/nav_off.gif HTTP/1.1" 404 "http://materials.cv.uoc.edu/continguts/XW06_72001_01542/web/main/m0/v5_4.html" "Mozilla/4.0 (compa
[10/Mar/2012:00:07:34 +0100] 79.157.134.56 "GET /favicon.ico HTTP/1.1" 404 "-" "Mozilla/5.0 (Windows NT 6.0) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.78 Safari/535.11" 1 2
[10/Mar/2012:00:07:34 +0100] 80.103.173.79 "GET /rb/inici/tips/show/6.text HTTP/1.1" 200 "http://cv.uoc.edu/rb/inici/grid" "Mozilla/5.0 (Windows NT 6.1; rv:8.0.1) Gecko/20100101 Firefox/8.0.1" 268 45
[10/Mar/2012:00:07:34 +0100] 88.23.161.52 "GET /favicon.ico HTTP/1.1" 404 "-" "Mozilla/5.0 (Windows NT 6.0) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.78 Safari/535.11" 1 1
[10/Mar/2012:00:07:34 +0100] 80.24.98.230 "GET /UOC/mc-icons/fotos/areverters.jpg HTTP/1.1" 200 "http://cv.uoc.es/cgi-bin/ma_msgPreview?s=7982563e9e907bf3a63cccf9b1a346b7ed0fab035af9e9f8f9de3df6303355cac530134
[10/Mar/2012:00:07:34 +0100] 85.61.166.47 "GET /rb/inici/user_modul/library/2066267.js?features=library:setprefs:dynamic-height&s=778e8632a0e43c08cb5f0e0c71e29955a7ded842e233b00e3d1745c96c43c225b1fda17a420d4f92c
[10/Mar/2012:00:07:34 +0100] 87.217.126.24 "GET /continguts/XW06_72001_01542/imgc/audicion_on.gif HTTP/1.1" 404 "http://materials.cv.uoc.edu/continguts/XW06_72001_01542/web/main/m0/v5_4.html" "Mozilla/4.0 (compa
[10/Mar/2012:00:07:34 +0100] 81.202.216.29 "POST /webapps/webUtils/servlet/InvitationChatServlet HTTP/1.1" 200 "http://cv.uoc.edu/cgi-bin/titol?is=1&w=w&s=fc0f3963a2ec6e333f48312fc8766cfeb1feefc3927a1fcd9c6fd57s
[10/Mar/2012:00:07:33 +0100] 85.60.105.1 "GET /webapps/widgetsUOC/widgetsRssServlet?up_items=10&up_numDias=120&up_title=Missatges&up_maximized=false&up_novaFinestra=false&up_tipusBustia=old&up_sortable=false&up_
[10/Mar/2012:00:07:34 +0100] 190.165.194.64 "GET /cdocent/RSVXJ8LL4YDWNBMA99QG.pdf?ajax=true HTTP/1.1" 206 "-" "Mozilla/5.0 (Windows NT 6.1; rv:10.0.2) Gecko/20100101 Firefox/10.0.2" 16624 1
[10/Mar/2012:00:07:34 +0100] 190.165.194.64 "GET /cdocent/RSVXJ8LL4YDWNBMA99QG.pdf?ajax=true HTTP/1.1" 206 "-" "Mozilla/5.0 (Windows NT 6.1; rv:10.0.2) Gecko/20100101 Firefox/10.0.2" 16624 1
[10/Mar/2012:00:07:34 +0100] 62.57.137.80 "GET /favicon.ico HTTP/1.1" 404 "-" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_8) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.78 Safari/535.11" 1 2
[10/Mar/2012:00:07:34 +0100] 192.168.1.146 "GET /webapps/campusGateway/agents/ac73fc3bcdd281c5839867f0776ca30eae14e2d91370282789bc90b1770b348eb642bece59c68e3d10b2669953521fc88482408f5f238a02446c4705fa609969/prof
[10/Mar/2012:00:07:34 +0100] 87.217.126.24 "GET /continguts/XW06_72001_01542/imgc/audicion_off.gif HTTP/1.1" 404 "http://materials.cv.uoc.edu/continguts/XW06_72001_01542/web/main/m0/v5_4.html" "Mozilla/4.0 (comp
[10/Mar/2012:00:07:34 +0100] 192.168.1.137 "GET /webapps/Agenda/AgendaServlet?operacion=ical&s=0afda5caabb0e01caebd0015adeb960b654d05e1433a8d079f96cc39f67e7fb8b3f8845bc5f8adfe564f421b2bab7e91319e57bd9fc64564bcdc
[10/Mar/2012:00:07:34 +0100] 81.39.251.183 "POST /rb/inici/tips/random?format=text HTTP/1.1" 302 "http://cv.uoc.edu/rb/inici/grid?s=ac73fc3bcdd281c5839867f0776ca30eae14e2d91370282789bc90b1770b348eb642bece59c68e3
[10/Mar/2012:00:07:34 +0100] 88.15.228.149 "GET /webapps/widgetsUOC/widgetsPlainServlet?up_title=Qui%2520h1%2520ha&up_urlServiceAPI=http%253A%252F%252Fcv.uoc.edu%252Fwebapps%252Fcampus2widget%252Fservlet%252FQui
[10/Mar/2012:00:07:34 +0100] 190.165.194.64 "GET /cdocent/RSVXJ8LL4YDWNBMA99QG.pdf?ajax=true HTTP/1.1" 206 "-" "Mozilla/5.0 (Windows NT 6.1; rv:10.0.2) Gecko/20100101 Firefox/10.0.2" 16624 1
[10/Mar/2012:00:07:34 +0100] 85.61.166.47 "GET /rb/inici/user_modul/library/2066268.js?features=library:setprefs:dynamic-height&s=778e8632a0e43c08cb5f0e0c71e29955a7ded842e233b00e3d1745c96c43c225b1fda17a420d4f92c
[10/Mar/2012:00:07:34 +0100] 190.165.194.64 "GET /cdocent/RSVXJ8LL4YDWNBMA99QG.pdf?ajax=true HTTP/1.1" 206 "-" "Mozilla/5.0 (Windows NT 6.1; rv:10.0.2) Gecko/20100101 Firefox/10.0.2" 4336 1
[10/Mar/2012:00:07:34 +0100] 192.168.1.137 "GET /UOC/celler/agenda/agenda_s.ics HTTP/1.1" 200 "-" "Java/1.5.0_15" 9750 2
```

5.2.2. Códigos de respuesta

Es importante señalar que el protocolo considerado a efectos del presente estudio será únicamente HTTP; las líneas de los ficheros log que refieran a otros protocolos serán descartadas y no se considerarán a efectos de la determinación de patrones de navegación.

En el caso del protocolo HTTP, estos códigos de respuesta están normalizados y pueden verse en la tabla de abajo, por lo que es posible determinar cuándo se ha producido un error en el acceso al recurso o cuando este se ha realizado con éxito.

Estos códigos están formados por tres dígitos siendo el primero de ellos el que identifica el tipo de respuesta según los siguientes criterios:

- **1xx**: Informativo. La petición se recibe y sigue el proceso. Esta familia de respuestas indican una respuesta provisional.
- **2xx**: Éxito. La acción requerida por la petición ha sido recibida, entendida y aceptada.
- **3xx**: Redirección. Para completar la petición se han de tomar más acciones.
- **4xx**: Error del cliente. La petición no es sintácticamente correcta y no se puede llevar a cabo.
- **5xx**: Error del servidor. El servidor falla al atender la petición que aparentemente es correcta.

Pero por ser visibles desde el navegador, son especialmente conocidos los siguientes códigos:

Código	Descripción
400	Petición errónea.
401	Acceso no autorizado.
403	Acceso prohibido.
404	Recurso no encontrado.

5.2.3. Métodos de petición

En cada línea de LOG se indica el método bajo el que se ha realizado la solicitud de una determinada operación. Estos son los siguientes:

- **GET**: Se utiliza para recuperar información, generalmente especificando parámetros en la URL, por ejemplo:
  http://midominio.es/miweb.htm?parametro1=valor1& parametro2=valor2

- **POST**: Cuando una petición se realiza usando el método POST, los datos se adjuntan a la petición a modo de objeto. Una de las ventajas de POST sobre GET es que facilita el envío de un mayor volumen de datos al servidor y que los parámetros están ocultos (no se visualizan en la URL).
- **HEAD**: Es equivalente al método GET excepto que el servidor no devolverá contenido, sólo las cabeceras HTTP. Generalmente se utiliza para comprobar si un enlace es válido.
- **OPTIONS**: Permite al cliente conocer las opciones y requisitos asociados con un recurso o las capacidades del servidor.

Aunque el protocolo HTTP considera también los siguientes métodos:

- **PUT**: Permite guardar el contenido de la petición en el servidor bajo la URL de la petición.
- **DELETE**: Método utilizado para que el servidor borre el recurso indicado por la URL de la petición.
- **TRACE**: Se utiliza para determinar si existe el receptor del mensaje enviado y usar la información para hacer un diagnóstico.

5.3. Patrones de navegación a analizar

Cada acción que realiza el usuario en el Campus Virtual de la UOC queda registrada en los ficheros LOGs a través del registro de una serie de información determinada. (ya descrita en el punto 5.2.1) por otra parte, el acceso al Campus Virtual implica el inicio de una sesión de trabajo por parte del usuario que ha accedido, dicha sesión de trabajo es identificada por una secuencia alfanumérica, que se realiza sobre un equipo informático (Tablet, portátil, PC, Smartphone,..) identificado con una Direccion IP.

Bajo este razonamiento anterior es posible vincular la sesión de navegación que realiza un usuario y dispositivo desde donde lo realiza con todos los datos registrados en una línea de LOG.

Así pues el presente PFC hará las trasformaciones necesarias en los ficheros LOGs de navegación a través del Campus Virtual para que se puedan analizar todos los patrones de navegación que sean posibles con los datos de los que disponemos. Estos son:

1. **Recursos accedidos**.

   Este patrón permitirá identificar a aquellos **recursos solicitados de manera más usual** por los usuarios del Campus Virtual de la UOC cuyas acciones se han registrado en los logs.

   Dada la pareja dirección IP e identificador de sesión, denominaremos **secuencia de navegación** al conjunto de recursos visitados por una determinada dirección IP durante una sesión de trabajo; es decir, que una secuencia de navegación estará formada por las URL a las que ha accedido un usuario desde que inicia sesión en el Campus Virtual hasta que finaliza su actividad. De esa forma, si un mismo usuario (identificado por la dirección IP del dispositivo desde el que ha accedido) iniciara una nueva sesión, el conjunto de recursos accedidos se consideraría parte de otra secuencia.

2. **Periodos de tiempo de navegación**.

   Este patrón de navegación permitirá identificar los tiempos en los que un usuario accede a los recursos, así como los periodos temporales donde se concentra la mayor cantidad de tráfico.

   Dada una determinada dirección IP e identificador de sesión, describiremos las horas a las que se conecta el usuario y con qué frecuencia solicita los diferentes recursos en una o varias sesiones de trabajo.

   Así pues se mostrará las horas a las que determinado usuario, en determinada sesión realiza peticiones de recursos del Campus Virtual de la UOC.

3. **Respuestas del servidor**.

   Este patrón de navegación permitirá identificar las respuestas del servidor apache del Sitio Web del Campus de la UOC que obtiene el usuario en una o varias sesiones. Esto tendrá especial interés en caso de los accesos erróneos y poder determinar comportamientos particulares.

   Dada una dirección IP que indique el dispositivo de conexión, y una sesión en la que se conecte, vinculará las respuestas que se ha obtenido del Servidor Web.

   De esta forma se mostrarán los códigos de respuesta (indicados en puntos anteriores) que determinada IP a obtenido en determinada sesión.

4. **Tipos de recursos accedidos**.
   Este patrón de navegación permitirá determinar los tipos de recursos a los que se accede mayoritariamente en cada dispositivo y en cada sesión.
   Dada una determinada IP en una cierta sesión se detallará los tipos de recursos accedidos (jpg, htm, cgi, php,….) y establecer patrones de uso de los estudiantes.

5. **Exploradores web usados**.
   Este patrón de navegación establecerá los exploradores web más usados y vinculará las diferentes sesiones con diferentes dispositivos con el explorador usado. Esto tiene especial interés para definir patrones de uso de navegadores web en las diferentes sesiones del usuario.
   Si bien, evidentemente cada sesión (secuencia de navegación) corresponderá únicamente a un tipo de explorador, será posible sacar conclusiones con respecto a tipos de exploradores más utilizados o con estudios más complejos, con exploradores por zonas geográficas o exploradores relacionados con códigos de error.

6. **Ubicación desde donde se navega**.
   Este patrón de navegación establecerá las ubicaciones geográficas correspondientes a una determinada IP para una sesión en concreto. Al igual que en el punto anterior, evidentemente, cada sesión de navegación corresponderá a una única ubicación, pero será posible sacar conclusiones de las ubicaciones desde donde navegan los diferentes usuarios en cada sesión, mayor concentración de conexiones, o estudios más complejos como son que combinen por ejemplo las horas en las que se conecta cada zona geográfica.
   Es importante tener en cuenta que el dato de la ubicación geográfica no viene definido en los LOGs del Servidor Apache, por ello tendremos que diseñar mecanismos que permitan obtener de manera externa este dato a través de la dirección IP.

5.4. Métodos aplicados

Después de haber aplicado el proceso de minería de datos, estructurándolos y realizando su procesado, será necesario utilizar métodos de análisis de dichos datos.

5.4.1. K-Means

K-Means (o método de los centroides) es un **método de agregación** (del inglés **clustering**) que, como tal, propone a partir de un conjunto de datos, la obtención de una **enumeración de grupos** (clusters) **de objetos con características similares**. Concretamente, el método K-Means, se basa en la obtención de un número *k* de grupos que es **fijado al principio del proceso**.

La teoría dice que el proceso comienza fijando un punto inicial del espacio (denominado semilla o, en inglés, seed) como centro del grupo potencial que se va a formar. Esta semilla puede ser bien uno de los objetos que forman parte del conjunto de datos inicial, bien una combinación de valores creada de forma artificial representando un resumen de las

características de varios objetos. A partir de esto, se pueden extraer los pasos que sigue el método K-Means:

1. Seleccionar las semillas iniciales

2. Calcular los centros

3. Asignar objetos al grupo con centro más próximo

4. Recalcular los centros

5. Continuar hasta que no haya variación en los grupos

Una vez formados los grupos, se determina, para cada objeto a estudiar, que centroide tiene más cerca asignando dicho objeto al grupo representado por el centroide más próximo. A este proceso se le denomina agregación.

Como resultado de la agregación de objetos, es necesario volver a calcular el centro de cada grupo. Esto se hace obteniendo para cada dimensión, el valor medio de todos los objetos que forman parte del grupo tratado. Una vez calculados los nuevos centros, el proceso se inicia otra vez y se repite hasta que, en dos iteraciones consecutivas no se produzcan cambios en los centros (o se produzcan pocos).

5.5. Aplicación para el procesamiento de los logs.

A la salida del aplicativo, devolverá como producto un fichero intermedio y uno final que se podrán estudiar con una herramienta de análisis de datos.

5.5.1. Aplicación seleccionada

En nuestro caso la aplicación seleccionada para el estudio será Weka, que contiene una colección de herramientas de visualización y algoritmos para análisis de datos y modelado predictivo, que además cuenta con una interfaz gráfica de usuario muy atractiva que permite acceder fácilmente a todas sus funcionalidades.

Es una herramienta de código libre que se puede descargar fácilmente en:
http://www.cs.waikato.ac.nz/ml/weka/index_downloading.html

Las principales características de esta herramienta son las siguientes:

- Está disponible libremente bajo la licencia pública general de GNU.
- Es muy portable porque está completamente implementado en Java y puede correr en casi cualquier plataforma.
- Contiene una extensa colección de técnicas para preprocesamiento de datos y modelado.
- Es fácil de utilizar por un principiante gracias a su interfaz gráfica de usuario.

Weka soporta varias tareas estándar de minería de datos, especialmente, preprocesamiento de datos, clustering, clasificación, regresión, visualización, y selección. Todas las técnicas de Weka se fundamentan en la asunción de que los datos están disponibles en un fichero plano (*flat file*) o una relación, en la que cada registro de datos está descrito por un número fijo de atributos (normalmente numéricos o nominales, aunque también se soportan otros tipos). Weka también proporciona acceso a bases de datos vía SQL gracias a la conexión JDBC (*Java Database Connectivity*) y puede procesar el resultado devuelto por una consulta hecha a la base de datos. No puede realizar minería de datos multi-relacional, pero existen aplicaciones que pueden convertir una colección de tablas relacionadas de una base de datos en una única tabla que ya puede ser procesada con Weka.

Circunscribiéndonos al caso que nos ocupa, al PFC, el método seleccionado será K-Means, que detallaremos a continuación, pero antes de eso, sirva como guía de uso rápida, daremos unos pasos para indicar el funcionamiento de Weka.

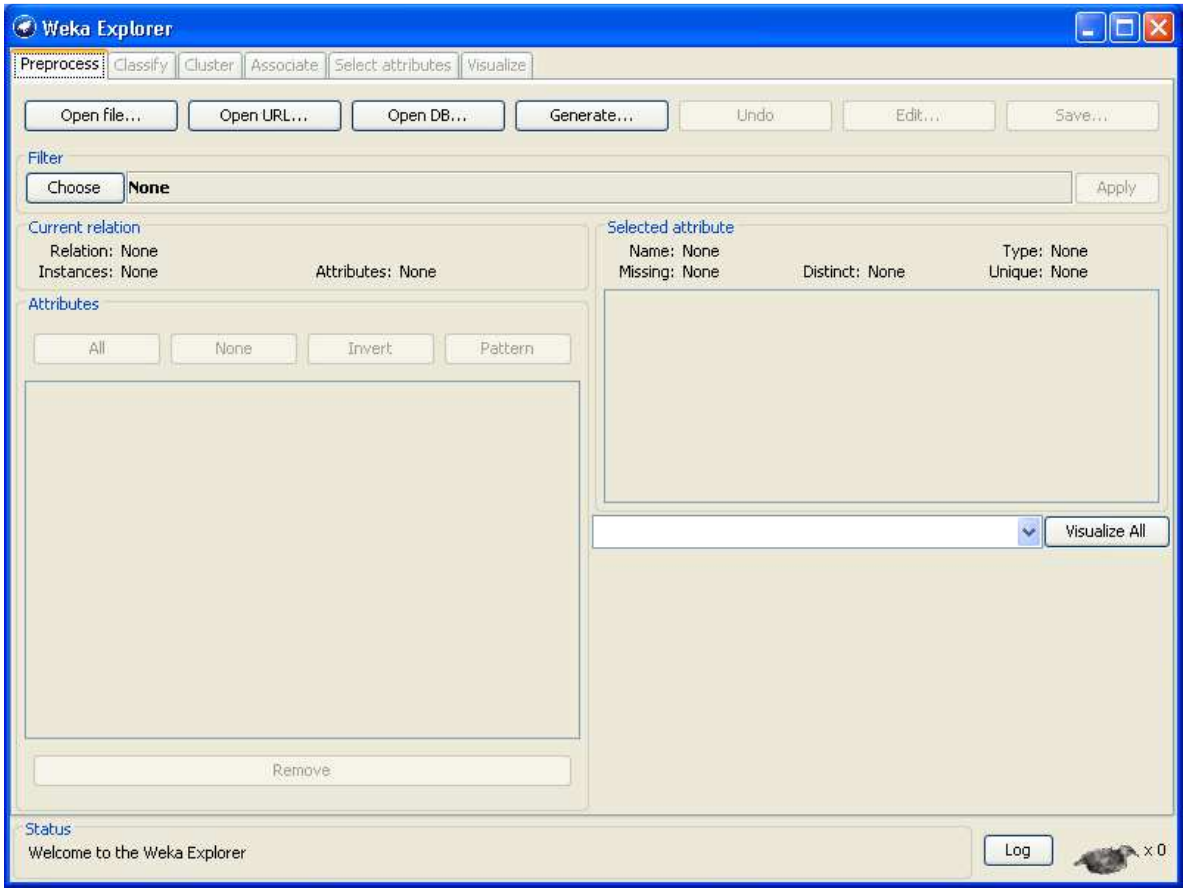
1. **Interfaz principal de Weka.**

   Según arranca Weka, se mostrará su pantalla principal que permite acceder a todas las funcionalidades. Exploraremos las imprescindibles para el ámbito del actual PFC. Aun así, existen multitud de manuales y guías en internet que explican en profundidad todos los aspectos del programa. A modo de ejemplo a continuación se nombran dos ejemplos.

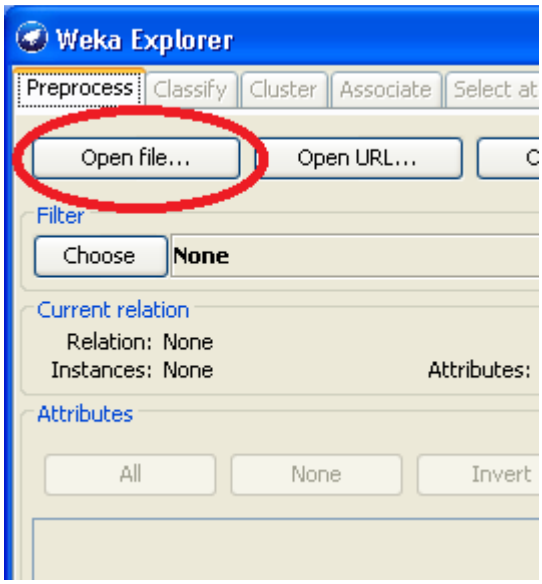
   http://ocw.uc3m.es/ingenieria-informatica/herramientas-de-la-inteligencia-artificial/contenidos/transparencias/TutorialWeka.pdf
   http://isa.umh.es/asignaturas/crss/turorialWEKA.pdf

2. **Carga de datos**

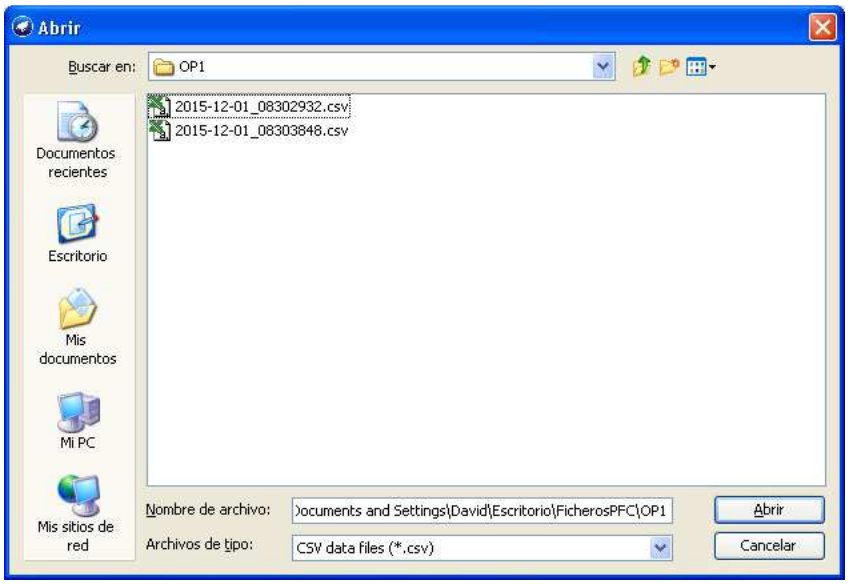
   Para poder comenzar a usar Weka necesitaremos cargar una muestra de datos que queramos estudiar, para se deberá clicar en el botón "Explorer" y aparecerá la siguiente pantalla, que será donde cargaremos y aplicaremos el algoritmo.



Abrimos el fichero clicando en "Open file…"

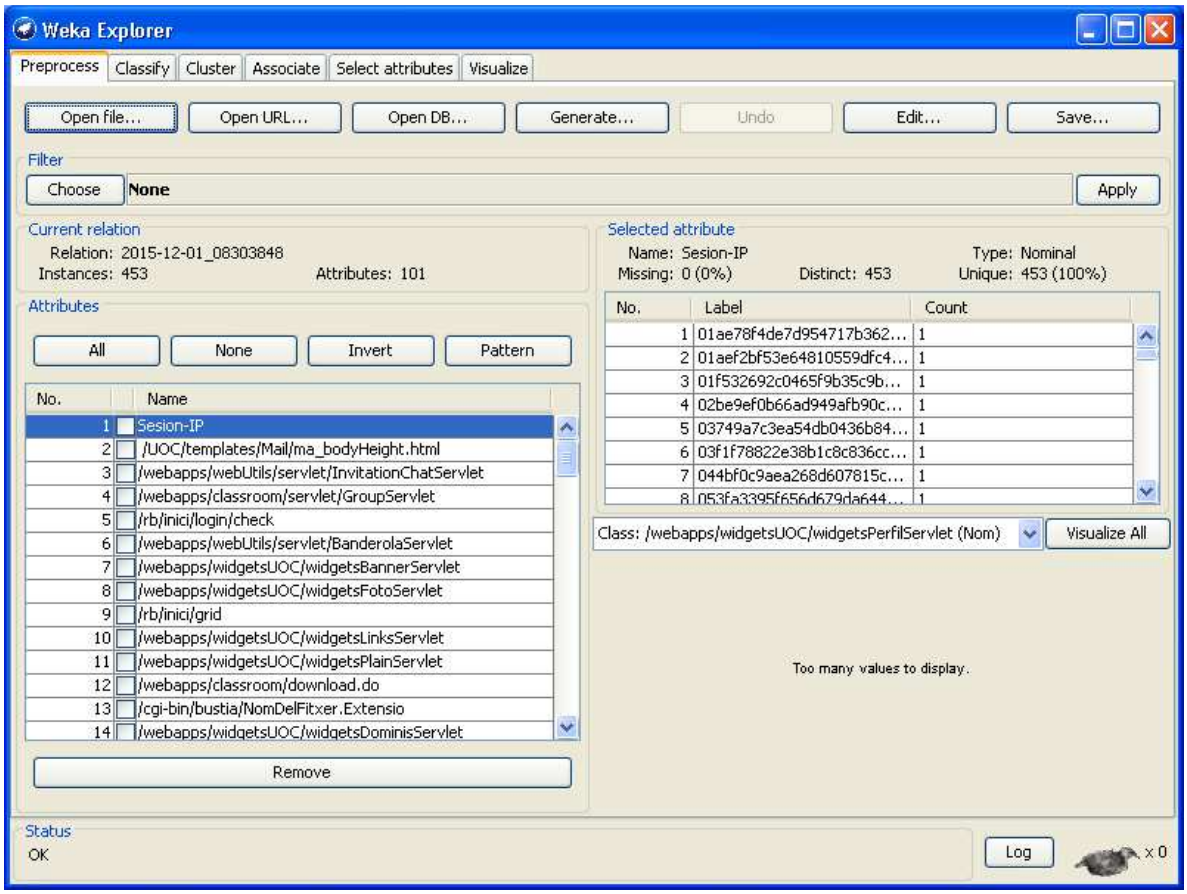


Y seleccionaremos el archivo que deseemos estudiar



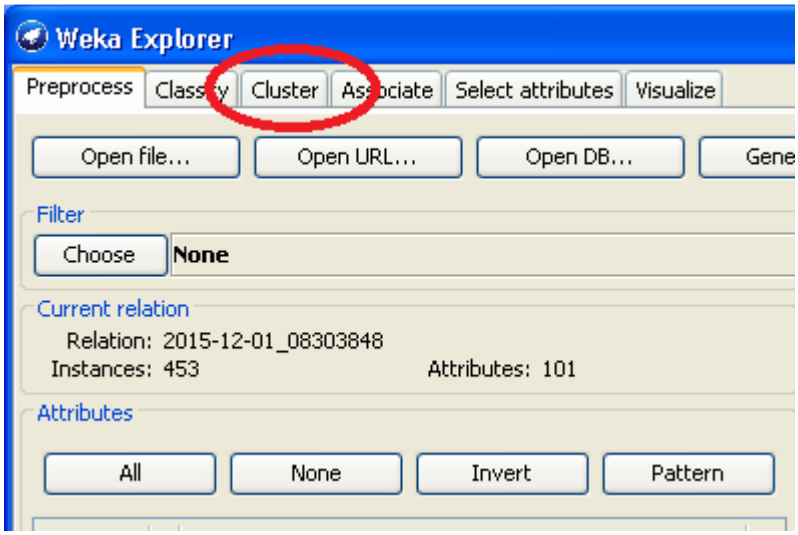
En nuestro caso el aplicativo devolverá archivos de exportación CSV, en concreto dos, un "*fichero intermedio*" y un "*fichero final*".
Seleccionaremos el que vayamos a estudiar, y al hacerlo se mostrará el resumen de la carga de datos.

- Atributos (columnas del CSV)
- Instancias (Filas del CSV)
- Clases

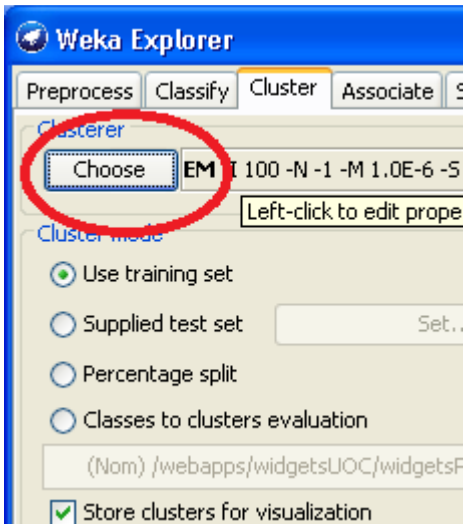


3. Aplicación de algoritmo K-Means.

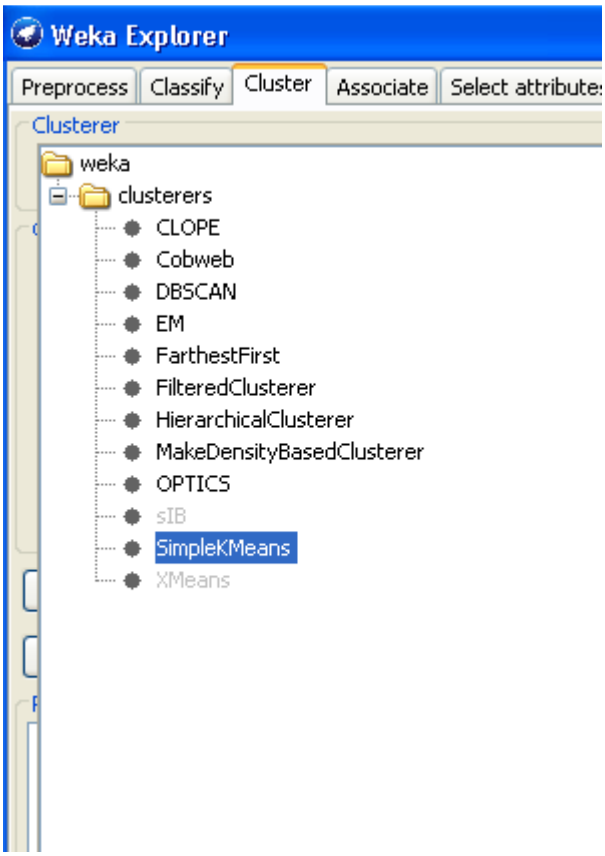
Ahora ya estaremos en disposición de aplicar el algoritmo k-Means, para ello iremos a la pestalla "Cluster"



Y seleccionaremos el botón "Choose" para elegir el método cluster que queremos aplicar.



Se nos desplegará un árbol donde aparecen todos los algoritmos disponibles. Seleccionamos "SimpleKMeans"



En este punto ya tenemos cargados los datos y seleccionado el algoritmo. Solo quedaría parametrizarlo que abordaremos en el siguiente punto.

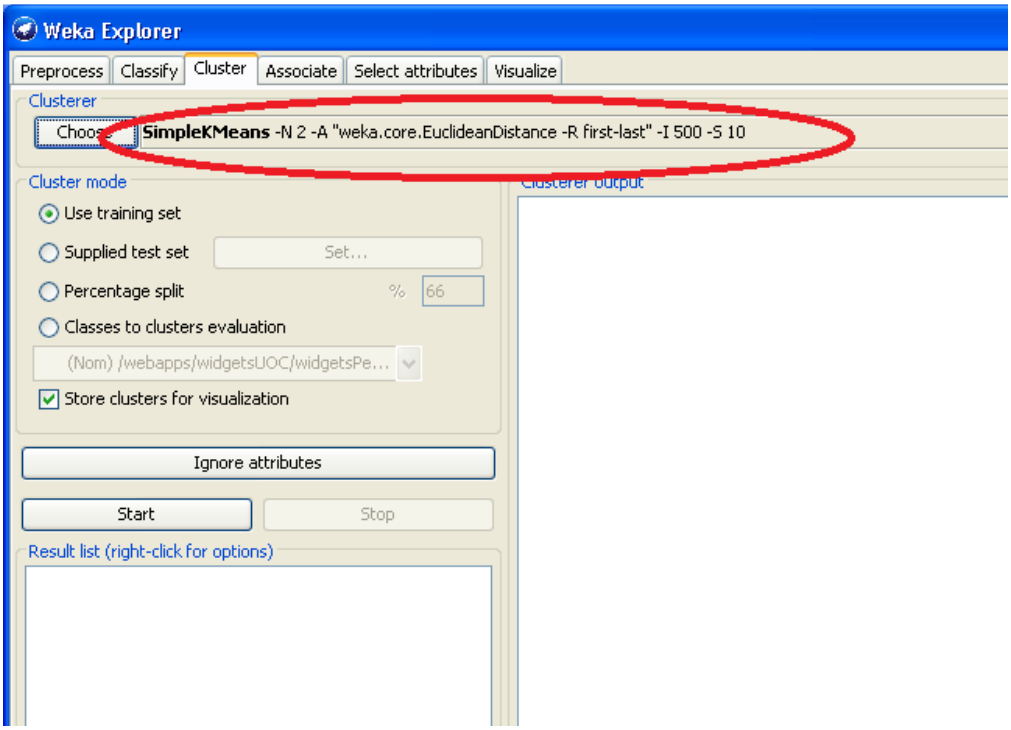
5.5.2. K-means

Weka implementa el método K-Means a través de un algoritmo denominado SimpleKMeans implementado en la clase *weka.clusterers.SimpleKMeans*. La ejecución de

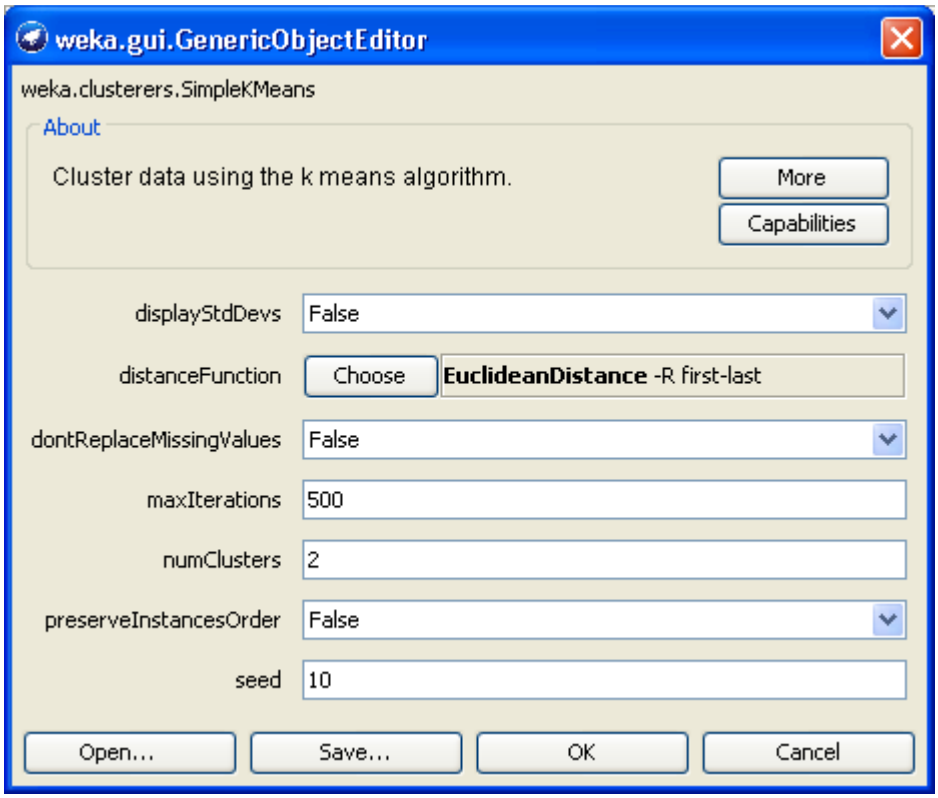
este algoritmo es parametrizable a través de una serie de opciones donde las más relevantes son las siguientes:

Parámetro	Descripción
numClusters	Total de agrupaciones (clusters)
seed	Valor de la semilla a partir de la cual se generará el numero aleatorio que inicializará los centros de los clusters

Para acceder a la pantalla de parametrización se hace clicando en el nombre del algoritmo



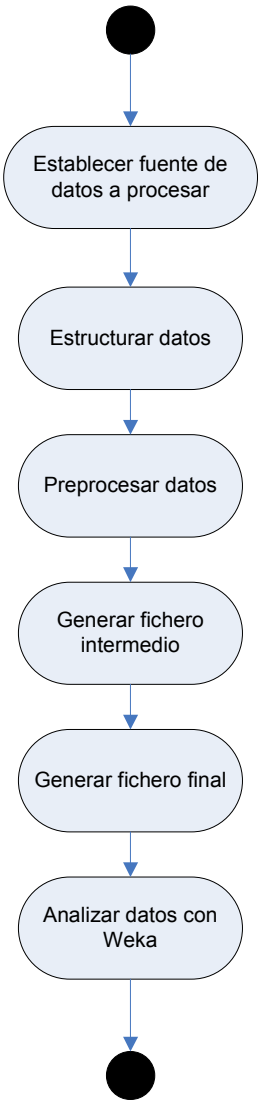
Se accede a la pantalla de parametrización donde se podrá cambiar los dos parámetros que acabamos de describir y otros más específicos del algoritmo.



6. Especificación del proceso de análisis de patrones de navegación

El objetivo del actual PFC es el Análisis de patrones de navegación de los estudiantes del Campus Virtual de la UOC a partir de los ficheros LOGs generados por el servidor Web del Sitio Web www.uoc.edu. En este punto se especificarán las actividades para cumplir el citado objetivo.

Sirva como visión general del proceso a realizar el siguiente flujograma:



6.1. Establecer fuente de datos a procesar

Esta primera actividad va a consistir únicamente en seleccionar la fuente de datos que se va a importar y que va a ser fuente de estudio.

Es importante hacer notar como requerimiento que estos datos de entrada deben tener un formato muy claro y concreto, y es el que genera el servidor web Apache en sus Logs de sistema y que se describe en el punto 5.2.1.

6.2. Estructurar datos

La segunda actividad del proceso consiste en importar y estructurar los datos procedentes del fichero de logs en texto plano generado por el servidor web apache y descrito en el punto anterior. Haciendo este proceso de importación se deberá resolver los problemas 1 (cantidad ingente de información), 2 (Soporte de la información) y 4 (información desestructurada) descritos en el punto 5.1. Aplicando diferentes mecanismos.

Consistirá en ir línea por línea del fichero de texto leyendo datos. Sirva como nota adjunta que en este punto se configurarán dos mecanismos para paliar en cierta medida el problema 1 descrito de la cantidad ingente de información. El caso es que, como describíamos en el

punto 5.1, estamos hablando de ficheros de texto de varios Gigabytes de información y 13 millones de filas. Evidentemente, esta cantidad de información es prácticamente inabordable y debemos evitar caer en bucles pseudo-infinitos, por ello se ha habilitado dos mecanismos:

1. **Leer fichero entre filas**. Se habilitarán dos opciones de entrada de datos. Leer Fichero completo que leerá el fichero log desde la primera a la última fila (bajo responsabilidad del usuario) y leer fichero entre una fila de inicio X y una fila de fin Y, que leerá el fichero log entre estos límites. Este último caso permite hacer el estudio de datos entre ciertos límites abordables por el programa y que el usuario deberá determinar de acuerdo a la capacidad del host donde se ejecute.
2. **Hacer subconjuntos de datos**. La otra opción consistirá en que el usuario edite a mano el fichero LOG total en pequeños subconjuntos de datos manejables y que los vaya importando secuencialmente en el aplicativo de análisis de datos.

Como íbamos explicando, el proceso leerá línea por línea obteniendo y extrayendo la información de cada una y metiéndola en campos de estructuras de memoria diseñados a tal fin (problema 4), que permanecerá ahí cargada pendiente de Preprocesar.

Además de esto a medida que se vayan leyendo los datos del fichero de texto line por línea se deberá permitir opción al usuario de indicar su desea consolidar o no la información en base de datos para mejorar la explotación de datos en futuros usos (problema 2). Para ello habrá que definir un modelo de datos que lo soporte.

Se han identificado los siguientes campos contenedores de información en una línea de registro de log.

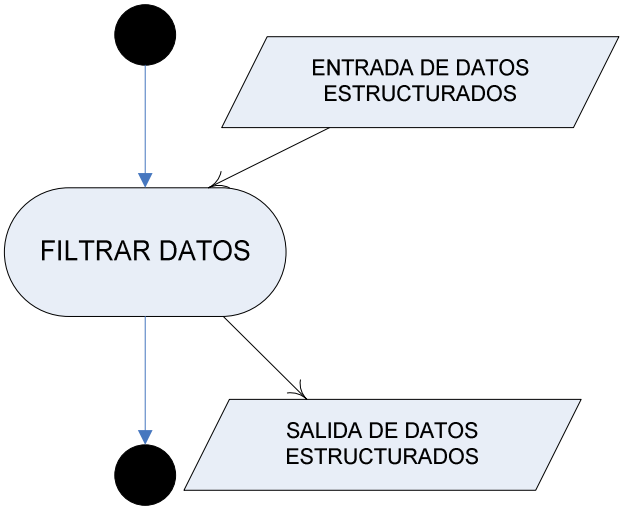
Estructura de datos LineaLOG		
Campo	**Tipo**	**Descripción**
Fecha	Cadena.	Fecha en que se produjo el registro delo LOG. fecha en que se realiza la petición al Servidor web de la página
Hora	Cadena	Hora en que se produjo el registro del LOG. Hora en que se pide la página. (El usuario accede a la web)
Zonahoraria	Cadena.	Diferencia horaria respecto GMT
IP	Cadena	Direccion IP en formato de cuatro octetos desde la cual el usuario se conecta
Method	Cadena.	Método de acceso al recurso. GET/POST
URL	Cadena	Direccion completa de acceso a la página, parámetros incluidos.
Recurso	Cadena.	Recuso al que accede la web. (quitando los parámetros del querystring) . Sin parámetros. En este paso del preprocesamiento, se eliminarán de las URL registradas en el log todos los parámetros que acompañan al

		recurso accedido guardándose en el array parámetros; es decir, la secuencia de caracteres que sigue al carácter '?' (incluyendo a este último). En este paso del flujo de trabajo, también hay que tener presentes otros casos menos numerosos, como cuando en vez de el carácter '?' es el carácter ';' el que actúa de separador
TipoRecurso	Cadena	Tipo de recurso al que se accede. Extensión. Por ejemplo jpg, gif, html,…
Sesion	Cadena.	Identificador de la sesión de navegación que asigna el servidor web a la sesión de navegación. En el momento de realizar el procesamiento de los logs, será muy importante distinguir las sesiones de trabajo de los usuarios cuyas operaciones se han registrado en los log. Por ello, en este paso del preprocesamiento se extraerá a un nuev campo con el identificador de sesión que figura en el elemento 'ruta recurso solicitado'. Recordar que el identificador de sesión puede distinguirse por ir precedido por los caracteres ?s= o &s= en función de si el parámetro s es el primero o no en una determinada URL
Parametros()	Cadena	Array o tabla de parámetros incluidos en el querystring de la llamada web. De 0 a N elementos
Protocolo	Cadena	Habitualmente HTTP
VersionProtocolo	Cadena.	Habitualmente 1.1
CodRespuesta	Cadena	Codigo de la respuesta devuelta por el servidor WEB. Enumerados en el pto 5.2.2
Referer	Cadena	Direccion WEB completa del origen de la petición WEB
Explorador	Cadena.	Explorador web en donde se está navegando. IE, Firefox, Chrome,…
Tamaño	Cadena	Tamaño en bytes del recuso solicitado.
Otros		Cajón desastre de otros datos no importantes.

6.3. Preprocesar datos

En este punto ya tendríamos TODA la información en estructuras de memoria estructurada, habría que pasar al siguiente paso que es la generación del fichero intermedio, sin embargo hay que tener en cuenta que no toda la información contenida en los LOGs es interesante de cara al análisis de patrones, hay que hacer previamente un proceso de limpieza.

Seguiremos el siguiente diagrama de flujo:

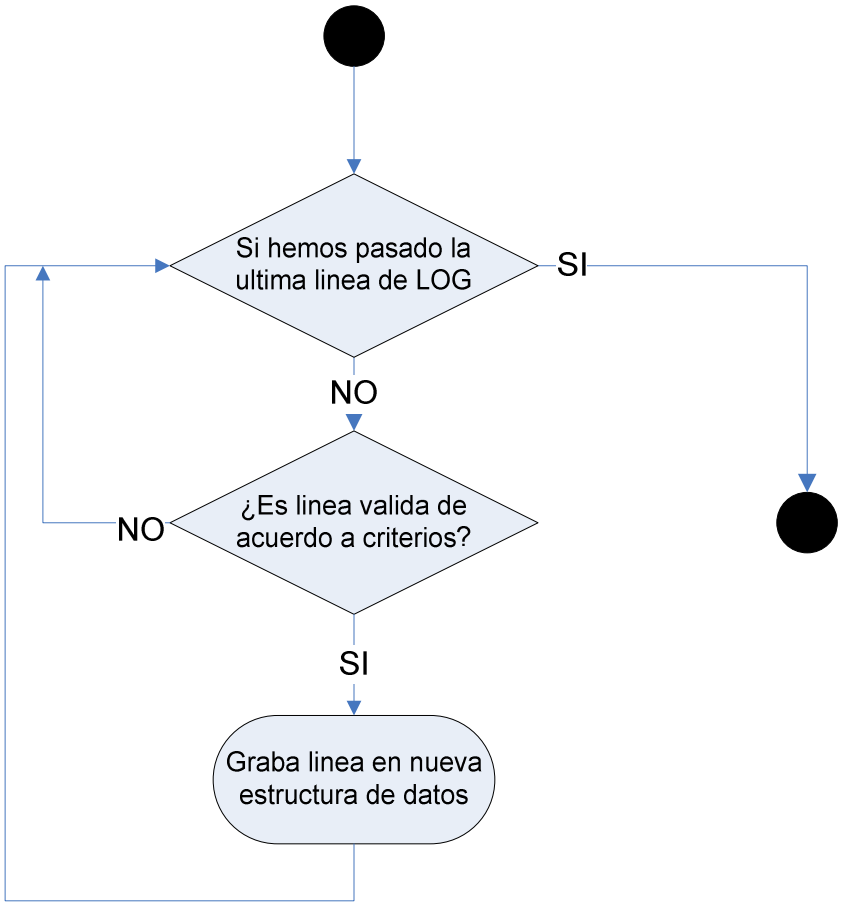


Como entrada de datos tendremos los datos estructurados en estructuras de memoria.

Como proceso, se iteraría por cada dato, determinando su conveniencia a incluir en el estudio de acuerdo a unos criterios que vamos a describir a continuación.

Como salida de datos tendríamos una estructura de memoria gemela a la de entrada, pero con los datos validos filtrados, es decir, un subconjunto del conjunto de datos de entrada.

Entrando en detalle, el flujograma para el proceso "FILTRAR DATOS" seria el siguiente:



Los criterios de “filtrado” de líneas que NO se incluirán en el análisis son las siguientes:

1. Operaciones no realizadas con el protocolo HTTP.
2. Refieran a accesos a:
   - Ficheros de imagen (JPG, GIF, PNG y BMP)
   - Iconos (ICO)
   - Ficheros JavaScript (JS)
   - Hojas de estilo CSS
   - Películas Flash (SWF)
   - Ficheros con extensión 'text'.
3. No refieran a operaciones realizadas por estudiantes del Campus Virtual, por tratarse generalmente de operaciones realizadas por **sistemas automatizados**. Estas líneas pueden distinguirse porque el elemento 'navegador' toma como valor un guion.
4. Para realizar el procesamiento de los logs, interesará determinar los accesos realizados durante una sesión de trabajo. Los identificadores de sesión pueden distinguirse por ir precedidos por los caracteres ?s= o &s= en función de si el parámetro s es el primero o no respectivamente en una determinada URL.
   Aunque lo deseable sería descartar todas aquellas **líneas** en las que el elemento 'ruta recurso solicitado' o 'ruta origen' **no incluya una referencia al identificador de sesión**, el análisis inicial de los ficheros de log ha desaconsejado hacer esto ya que implicaría la pérdida de excesivas líneas. En cualquier caso, no se ha eliminado la referencia a esta actividad para tenerla presente en futuros estudios.
5. Incluyan un **código de respuesta que señale un error** en el acceso al recurso, esto es, todas aquellas líneas cuyo código de respuesta no empiece por 2 (rango 200 a 299).
6. Refieran a una **dirección IP para la cual se han detectado valores anómalos**.

6.4. Generar fichero intermedio

Ya tendríamos los datos del fichero LOG (o un subconjunto de él) estructurados, filtradas las líneas inútiles y cargados en memoria. Ahora se deberá, en función de las 6 opciones enumeradas en el punto 5.3, generar el fichero intermedio adecuado dependiendo de sus datos.

6.4.1. Fichero intermedio Opción 1- Recursos accedidos.

Este patrón permitirá identificar a aquellos **recursos solicitados de manera más usual** por los usuarios del Campus Virtual de la UOC cuyas acciones se han registrado en los logs. Así pues relacionará IP-Sesión-Recurso descartando en este caso el resto de elementos. Generando línea por línea y ordenado por identificador de sesión un fichero CSV (y XML) con los siguiente campos separados por comas.

IP, Recurso, IdSesion

Un posible ejemplo seria:

```
IP,RECURSO,SESION
111.111.111.111,uoc/recurso1,1111111111ª
111.111.111.111,uoc/recurso2,1111111111ª
222.222.222.222,uoc/recurso1,2222222222ª
333.333.333.333,uoc/recurso1,3333333333ª
333.333.333.333,uoc/recurso3,3333333333ª
```

444.444.444.444,uoc/recurso3,4444444444ª
555.555.555.555,uoc/recurso1,5555555555ª
555.555.555.555,uoc/recurso2,5555555555ª
555.555.555.555,uoc/recurso3,5555555555ª
111.111.111.111,uoc/recurso1,1111111111b

6.4.2. Fichero intermedio Opción 2 - Periodos de tiempo de navegación.

Este patrón de navegación permitirá identificar los tiempos en los que un usuario accede a los recursos, así como los periodos temporales donde se concentra la mayor cantidad de tráfico.

Producirá un fichero de datos de exportación, CSV, de acuerdo a una estructura determinada que se utilizará para introducirlo en el aplicativo Weka (que describiremos más adelante) o como entrada para generar el fichero final. La única diferencia son los datos analizados. Esta estructura será, ordenada por id de sesión, la siguiente:

IP, FechaHora, Sesión

Un posible ejemplo seria:

IP, Fecha, SESION
111.111.111.111,10/Mar/2015-00:00:15,1111111111ª
111.111.111.111, 10/Mar/2015-00:00:16,1111111111ª
222.222.222.222, 10/Mar/2015-00:00:16,2222222222ª
333.333.333.333, 10/Mar/2015-00:00:17,3333333333ª

6.4.3. Fichero intermedio Opción 3 - Respuestas del servidor.

Este patrón de navegación permitirá identificar las respuestas del servidor apache del Sitio Web del Campus de la UOC que obtiene el usuario en una o varias sesiones. Esto tendrá especial interés en caso de los accesos erróneos y poder determinar comportamientos particulares.

La estructura del fichero CSV que generará será la siguiente (ordenado por id Sesion):

IP, Código de Respuesta, Sesión

Un posible ejemplo seria:

IP, Fecha, SESION
111.111.111.111,200,1111111111ª
111.111.111.111, 200,1111111111ª
222.222.222.222, 300,2222222222ª
333.333.333.333, 400,3333333333ª

NOTA: Los códigos de respuesta del Servidor WEB Apache, se han definido en el pto 5.2.2.

6.4.4. Fichero intermedio Opción 4 - Tipos de recursos accedidos.

Este patrón de navegación permitirá determinar los tipos de recursos a los que se accede mayoritariamente en cada dispositivo y en cada sesión.

La estructura del fichero CSV que generará será la siguiente (ordenado por id Sesión):

IP, Tipo de Recuso, Sesión

Un posible ejemplo seria:

IP, Fecha, SESION
111.111.111.111,jpg,1111111111ª
111.111.111.111, html,1111111111ª
222.222.222.222, jsp,2222222222ª
333.333.333.333, gif,3333333333ª

```
181.132.53.242,,1b908280de00fd0c69c6848fafcaac083dbb27ad8d16cad6f74868905736b4564b6ffb42cd9d1a09acaa7a31944aa0c4c574d602dd1dd95a71f3762e89474f45
181.132.53.242,,1b908280de00fd0c69c6848fafcaac083dbb27ad8d16cad6f74868905736b4564b6ffb42cd9d1a09acaa7a31944aa0c4c574d602dd1dd95a71f3762e89474f45
77.225.174.90,html,1bb12093bab016ea2e972776f9c8a6a43adaa2e21a561797b014c4ec7a720d5f70fbb4889f99ec22e7d5418150d394868c98b19f1f764fe02f1a571a20998e7b
77.225.174.90,html,1bb12093bab016ea2e972776f9c8a6a43adaa2e21a561797b014c4ec7a720d5f70fbb4889f99ec22e7d5418150d394868c98b19f1f764fe02f1a571a20998e7b
77.225.174.90,,1bb12093bab016ea2e972776f9c8a6a43adaa2e21a561797b014c4ec7a720d5f70fbb4889f99ec22e7d5418150d394868c98b19f1f764fe02f1a571a20998e7b
77.225.174.90,dat,1bb12093bab016ea2e972776f9c8a6a43adaa2e21a561797b014c4ec7a720d5f70fbb4889f99ec22e7d5418150d394868c98b19f1f764fe02f1a571a20998e7b
77.225.174.90,htm,1bb12093bab016ea2e972776f9c8a6a43adaa2e21a561797b014c4ec7a720d5f70fbb4889f99ec22e7d5418150d394868c98b19f1f764fe02f1a571a20998e7b
77.225.174.90,,1bb12093bab016ea2e972776f9c8a6a43adaa2e21a561797b014c4ec7a720d5f70fbb4889f99ec22e7d5418150d394868c98b19f1f764fe02f1a571a20998e7b
83.55.100.61,,1bef76c443a530d80d9a89b1e79a1414464fe00c2c6944f20db555c668d1208c274e018d9976974a8d6c122748211f70fec112e87db1f3ae4b58d8a5c4baa905
83.55.100.61,,1bef76c443a530d80d9a89b1e79a1414464fe00c2c6944f20db555c668d1208c274e018d9976974a8d6c122748211f70fec112e87db1f3ae4b58d8a5c4baa905
83.55.100.61,,1bef76c443a530d80d9a89b1e79a1414464fe00c2c6944f20db555c668d1208c274e018d9976974a8d6c122748211f70fec112e87db1f3ae4b58d8a5c4baa905
192.168.234.65,,1c82e821edbcda7ce6d785efb5020df70b374b58153037542dc406994886eeef5cbae79417c306f5bf52020b61c024e08ac56747d9708046984c4bff16a17d16
192.168.234.65,,1c82e821edbcda7ce6d785efb5020df70b374b58153037542dc406994886eeef5cbae79417c306f5bf52020b61c024e08ac56747d9708046984c4bff16a17d16
81.36.138.245,,1cf51ba88f113341bed9ace2afe35a807a590b53540820fd64fbb97a8ea4be2acf86510ccf861dffffd7f43aff0ee23b3a8454d765aee5c17c19a13d1a1ca0b5
83.55.12.134,,204527dca3a58fbfe5a737e80527f235c08a55cbb981121af4fdfe47659de697528893519de006a3e6a36964acb6d03b56604256b98ff646e3289adc5b003ebe
88.9.77.238,,20d87875c706c52b6a21cedbc1f3c009b692e02dca8c220cf86bf734b59d04f61cd91a855c5f0ad34a8cdd12a0a7266583692627119df7ae86dec1e081694b47
88.9.77.238,,20d87875c706c52b6a21cedbc1f3c009b692e02dca8c220cf86bf734b59d04f61cd91a855c5f0ad34a8cdd12a0a7266583692627119df7ae86dec1e081694b47
89.131.97.84,,22682648397 8dfd9aadb46df106c927c98a42d7f707027eab3bbb73f4e1589636161765cd1654a7dd74b3155011c891ea4af524b26a032e735e38177ca5f62ed
89.131.97.84,,226826483978dfd9aadb46df106c927c98a42d7f707027eab3bbb73f4e1589636161765cd1654a7dd74b3155011c891ea4af524b26a032e735e38177ca5f62ed
37.14.97.87,,233320a823254946ab011f08579c75902fc863cc4490eb2a24df7e5a98e81de42dd89b375aff51211a84932c137b03b3f4ad510c01f87dac00e3ac6c48d8653d
37.14.97.87,,233320a823254946ab011f08579c75902fc863cc4490eb2a24df7e5a98e81de42dd89b375aff51211a84932c137b03b3f4ad510c01f87dac00e3ac6c48d8653d
90.171.59.2,jsp,246126be67b5dbf533ff71792e145b8514b76fa02f78dbb27d04023cfcacecadc948171fa3ea34c714d0e85b3ff1b552f467996e10405b14f4829490610f2437
90.171.59.2,jsp,246126be67b5dbf533ff71792e145b8514b76fa02f78dbb27d04023cfcacecadc948171fa3ea34c714d0e85b3ff1b552f467996e10405b14f4829490610f2437
90.171.59.2,,246126be67b5dbf533ff71792e145b8514b76fa02f78dbb27d04023cfcacecadc948171fa3ea34c714d0e85b3ff1b552f467996e10405b14f4829490610f2437
90.171.59.2,jsp,246126be67b5dbf533ff71792e145b8514b76fa02f78dbb27d04023cfcacecadc948171fa3ea34c714d0e85b3ff1b552f467996e10405b14f4829490610f2437
90.171.59.2,jsp,246126be67b5dbf533ff71792e145b8514b76fa02f78dbb27d04023cfcacecadc948171fa3ea34c714d0e85b3ff1b552f467996e10405b14f4829490610f2437
90.171.59.2,do,246126be67b5dbf533ff71792e145b8514b76fa02f78dbb27d04023cfcacecadc948171fa3ea34c714d0e85b3ff1b552f467996e10405b14f4829490610f2437
```

6.4.5. Fichero intermedio Opción 5 - Exploradores web usados.

Este patrón de navegación establecerá los exploradores web más usados y vinculará las diferentes sesiones con diferentes dispositivos con el explorador usado. Esto tiene especial interés para definir patrones de uso de navegadores web en las diferentes sesiones del usuario.

La estructura del fichero CSV que generará será la siguiente (ordenado por id Sesión):

IP, Explorador web, Sesión

Un posible ejemplo seria:

IP, Fecha, SESION
111.111.111.111,Mozilla Firefox,1111111111ª
111.111.111.111, IE9,1111111111ª
222.222.222.222, Google Chrome,2222222222ª
333.333.333.333, Mozilla Firefox,3333333333ª

```
88.13.89.216,Mozilla/5.0 (Windows; U; Windows NT 6.0; es-ES; rv:1.9.2.26) Gecko/20120128 Firefox/3.6.26 ( .NET CLR 3.5.30729; .NET4.0C),003a28a6f28c933a37a076da3f00c63e6965e23b5b1ef601b3a208511bc1ffc9e83908e
88.13.89.216,Mozilla/5.0 (Windows; U; Windows NT 6.0; es-ES; rv:1.9.2.26) Gecko/20120128 Firefox/3.6.26 ( .NET CLR 3.5.30729; .NET4.0C),003a28a6f28c933a37a076da3f00c63e6965e23b5b1ef601b3a208511bc1ffc9e83908e
79.148.251.140,Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0),01ae78f4de7d954717b362d0aedfe3f44071ab31d7c2c6d4ff87a894d1c6f6c85b61c5ab806c4d486ffa0946bd9c5d70f179398252c4a32333df02963
79.148.251.140,Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; WOW64; Trident/5.0),01ae78f4de7d954717b362d0aedfe3f44071ab31d7c2c6d4ff87a894d1c6f6c85b61c5ab806c4d486ffa0946bd9c5d70f179398252c4a32333df02963
88.19.144.28,Mozilla/5.0 (Windows; U; Windows NT 6.1; ca; rv:1.9.2.27) Gecko/20120216 Firefox/3.6.27,01aef2bf53e64810559dfc4bb75fda05fbeb2c16c75e6983a9022bd99695cc7d1ed0cf98a1bc445766cbaa693e7a1f791539b0b1da
88.19.144.28,Mozilla/5.0 (Windows; U; Windows NT 6.1; ca; rv:1.9.2.27) Gecko/20120216 Firefox/3.6.27,01aef2bf53e64810559dfc4bb75fda05fbeb2c16c75e6983a9022bd99695cc7d1ed0cf98a1bc445766cbaa693e7a1f791539b0b1da
84.126.53.217,Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/534.53.11 (KHTML, like Gecko) Version/5.1.3 Safari/534.53.10,01f532692c0465f9b35c9b59f33a951714eba48fd9c837b01879b66bb35118c3659b0ce41
84.126.53.217,Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/534.53.11 (KHTML, like Gecko) Version/5.1.3 Safari/534.53.10,01f532692c0465f9b35c9b59f33a951714eba48fd9c837b01879b66bb35118c3659b0ce41
88.5.84.98,Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.0; Trident/5.0),02be9ef0b66ad949afb90ccb8902e442a4fd216c96fcdae82d67db26743d77c0cd090ed1b306529b933c7954b028ddd10c575e552181ad3787565a54af8afe24
88.5.84.98,Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.0; Trident/5.0),02be9ef0b66ad949afb90ccb8902e442a4fd216c96fcdae82d67db26743d77c0cd090ed1b306529b933c7954b028ddd10c575e552181ad3787565a54af8afe24
88.23.198.240,Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0; chromeframe/17.0.963.78),03749a7c3ea54db0436b846bf051dc84f88f8194c918cc7c7c8185ab6140cf867b27bfbec16ad9ca23360a171fa5285ae45016b9
186.144.81.132,Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.78 Safari/535.11,03f1f78822e38b1c8c836cc5f8c87930698bf197019dae222581370ab3da7abb411d13312534130ae2b1
186.144.81.132,Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/535.11 (KHTML, like Gecko) Chrome/17.0.963.78 Safari/535.11,03f1f78822e38b1c8c836cc5f8c87930698bf197019dae222581370ab3da7abb411d13312534130ae2b1
188.79.69.78,Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; GTB7.3; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; .NET CLR 1.1.4322; .NET CLR 3.5.30729; .NET CLR 3.0.30618),044bf0c9aea268d607815cff926c
188.79.69.78,Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; GTB7.3; SLCC1; .NET CLR 2.0.50727; Media Center PC 5.0; .NET CLR 1.1.4322; .NET CLR 3.5.30729; .NET CLR 3.0.30618),044bf0c9aea268d607815cff926c
80.33.57.108,Mozilla/5.0 (Windows NT 6.1; WOW64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2,053fa3395f656d679da6440f3a0e8753180084a25ff4ed3b908ee7b41a39d295e8a425539ab1269dc705ad86260dbceaffea5922646850074b230
192.168.1.130,Java/1.5.0_15,053fa3395f656d679da6440f3a0e8753180084a25ff4ed3b908ee7b41a39d295e8a425539ab1269dc705ad86260dbceaffea5922646850074b230d017d33006d
80.33.57.108,Mozilla/5.0 (Windows NT 6.1; WOW64; rv:10.0.2) Gecko/20100101 Firefox/10.0.2,053fa3395f656d679da6440f3a0e8753180084a25ff4ed3b908ee7b41a39d295e8a425539ab1269dc705ad86260dbceaffea5922646850074b230
80.102.169.126,Mozilla/5.0 (X11; Linux x86_64; rv:9.0.1) Gecko/20100101 Firefox/9.0.1 Iceweasel/9.0.1,05ba66625e7c6e673b051bf4ca3e1a5b3405865c47c0ff0d2a9dc7e01861d82e3aaf275b164f1623f2ca97677740702511d0b8bf
194.179.34.16,Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; GTB7.2; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648; .NET4.0C; .NET4.0E; .NET CLR 3.0.4506.2152; .NET
192.168.1.130,Java/1.5.0_15,05be89a1e2484ccb33b46a7c19b96cfe7addf70a5d6a355baaa9b87978eb14ed5d019c7f0cd094a7eebb53a35df6221713865f3fd391fd2767c198bdbcdf43de
194.179.34.16,Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; GTB7.2; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648; .NET4.0C; .NET4.0E; .NET CLR 3.0.4506.2152; .NET
194.179.34.16,Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; GTB7.2; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648; .NET4.0C; .NET4.0E; .NET CLR 3.0.4506.2152; .NET
192.168.1.130,Java/1.5.0_15,05be89a1e2484ccb33b46a7c19b96cfe7addf70a5d6a355baaa9b87978eb14ed5d019c7f0cd094a7eebb53a35df6221713865f3fd391fd2767c198bdbcdf43de
192.168.1.130,Java/1.5.0_15,05be89a1e2484ccb33b46a7c19b96cfe7addf70a5d6a355baaa9b87978eb14ed5d019c7f0cd094a7eebb53a35df6221713865f3fd391fd2767c198bdbcdf43de
192.168.1.130,Java/1.5.0_15,05be89a1e2484ccb33b46a7c19b96cfe7addf70a5d6a355baaa9b87978eb14ed5d019c7f0cd094a7eebb53a35df6221713865f3fd391fd2767c198bdbcdf43de
194.179.34.16,Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; GTB7.2; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648; .NET4.0C; .NET4.0E; .NET CLR 3.0.4506.2152; .NET
194.179.34.16,Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; GTB7.2; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648; .NET4.0C; .NET4.0E; .NET CLR 3.0.4506.2152; .NET
194.179.34.16,Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; GTB7.2; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648; .NET4.0C; .NET4.0E; .NET CLR 3.0.4506.2152; .NET
194.179.34.16,Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1; GTB7.2; .NET CLR 1.1.4322; .NET CLR 2.0.50727; .NET CLR 3.0.04506.30; .NET CLR 3.0.04506.648; .NET4.0C; .NET4.0E; .NET CLR 3.0.4506.2152; .NET
83.35.255.158,Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET CLR 1.1.4322),06
83.35.255.158,Mozilla/4.0 (compatible; MSIE 8.0; Windows NT 5.1; Trident/4.0; .NET CLR 2.0.50727; .NET CLR 3.0.04506.648; .NET CLR 3.5.21022; .NET CLR 3.0.4506.2152; .NET CLR 3.5.30729; .NET CLR 1.1.4322),06
77.210.111.217,Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; BTRS7360; GTB7.3; Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) ; SLCC1; .NET CLR 2.0.50727; .NET CLR 1.1.4322; .NET CLR 3.5.30729;
77.210.111.217,Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; BTRS7360; GTB7.3; Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) ; SLCC1; .NET CLR 2.0.50727; .NET CLR 1.1.4322; .NET CLR 3.5.30729;
77.210.111.217,Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.0; BTRS7360; GTB7.3; Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1; SV1) ; SLCC1; .NET CLR 2.0.50727; .NET CLR 1.1.4322; .NET CLR 3.5.30729;
82.213.176.137,Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/534.53.11 (KHTML, like Gecko) Version/5.1.3 Safari/534.53.10,068a54ce4ec61dd8c6dbd9fa7bc5294d437d126e8369cb756378e04c26b05684c24b694e
82.213.176.137,Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/534.53.11 (KHTML, like Gecko) Version/5.1.3 Safari/534.53.10,068a54ce4ec61dd8c6dbd9fa7bc5294d437d126e8369cb756378e04c26b05684c24b694e
82.213.176.137,Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_3) AppleWebKit/534.53.11 (KHTML, like Gecko) Version/5.1.3 Safari/534.53.10,068a54ce4ec61dd8c6dbd9fa7bc5294d437d126e8369cb756378e04c26b05684c24b694e
213.60.221.66,Mozilla/5.0 (Windows; U; Windows NT 5.1; es-ES; rv:1.9.2.27) Gecko/20120216 Firefox/3.6.27 ( .NET CLR 3.5.30729),06c764dfc5136e0d20d379c48403dcf0aec9d3e2e6ac6aa1f03cf515acf6418c67c424554c9d7e0c
213.60.221.66,Mozilla/5.0 (Windows; U; Windows NT 5.1; es-ES; rv:1.9.2.27) Gecko/20120216 Firefox/3.6.27 ( .NET CLR 3.5.30729),06c764dfc5136e0d20d379c48403dcf0aec9d3e2e6ac6aa1f03cf515acf6418c67c424554c9d7e0c
88.6.55.182,Mozilla/5.0 (Windows NT 5.1; rv:9.0.1) Gecko/20100101 Firefox/9.0.1,0784e50bfeca8149d476d1ee6148b4ee90f34b37d0eda795d7137ca5cd0667b9dec09de6396fd81ab80b1256b4fbb02bc2ab511a5a21343ca3785d588ec209e
88.6.55.182,Mozilla/5.0 (Windows NT 5.1; rv:9.0.1) Gecko/20100101 Firefox/9.0.1,0784e50bfeca8149d476d1ee6148b4ee90f34b37d0eda795d7137ca5cd0667b9dec09de6396fd81ab80b1256b4fbb02bc2ab511a5a21343ca3785d588ec209e
```

6.4.6. Fichero intermedio Opción 6 - Ubicación desde donde se navega.

Este patrón de navegación establecerá las ubicaciones geográficas correspondientes a una determinada IP para una sesión en concreto. En este caso es importante tener en cuenta que el dato de la ubicación geográfica no viene definido en los LOGs del Servidor Apache, por ello tendremos que diseñar mecanismos que permitan obtener de manera externa este dato a través de la dirección IP.

Como opción valida puede estudiarse diferentes opciones:

- Servicios externos WEB.
- Servicios internos WEB.

Pero el caso es que devuelva una ubicación geográfica a partir de una dirección IP.

Un posible ejemplo seria:

IP, Fecha, SESION
111.111.111.111,Madrid, Firefox,1111111111ª
111.111.111.111, Barcelona,1111111111ª
222.222.222.222, Sevilla,2222222222ª
333.333.333.333, Albacete,3333333333ª


```
77.210.111.217,Spain-Albacete-Castilla-La Mancha,06678caaea0270155e678afd2e9febf16204db7603218bec8fdaa3ddfaca2482f7d82cb023293e745c02fb29b17560df382bddffca579e340b6daa5ba4dce961
77.230.4.91,Spain-Tarragona-Catalonia,09cadda77b7ed202750644e22bcff2d909640e9ba0466d9a6ec3ba7c35a77807e9175d1f1fe00f566575bd98108e7d4a622b6151d660ecec94f4c2c0b44e20c6
88.15.228.149,Spain-Cabrils-Catalonia,0afda5caabb0e01caebd0015adeb960b654d05e1433a8d079f96cc39f67e7fb8b3f8845bc5f8adfe564f421b2bab7e91319e57bd9fc64564bcdd2b8eeee4c2f5
88.15.228.149,Spain-Cabrils-Catalonia,0afda5caabb0e01caebd0015adeb960b654d05e1433a8d079f96cc39f67e7fb8b3f8845bc5f8adfe564f421b2bab7e91319e57bd9fc64564bcdd2b8eeee4c2f5
88.15.228.149,Spain-Cabrils-Catalonia,0afda5caabb0e01caebd0015adeb960b654d05e1433a8d079f96cc39f67e7fb8b3f8845bc5f8adfe564f421b2bab7e91319e57bd9fc64564bcdd2b8eeee4c2f5
192.168.1.137,-----,0afda5caabb0e01caebd0015adeb960b654d05e1433a8d079f96cc39f67e7fb8b3f8845bc5f8adfe564f421b2bab7e91319e57bd9fc64564bcdd2b8eeee4c2f5&ini=1331334000000&tipo=month&l=a&appId=UOC
201.240.140.205,Peru-Lima-Lima,1271bb509f2e09a364dbf4c2402bf037707631f10e1c8b1a93c81821b61241573a0b658fc111871f03fe58d05725f83173851b3d88d670411d8e192c8e131bdf
81.184.46.28,Spain-Barcelona-Catalonia,24e0deb964b39910c9dca7d72c45feb9ec1c6a9237342019549fb38a178ba63d0d980b1a1e26626202a373c84dce444cbb5ff6ad4c82458d6928b8b8f595be47
192.168.1.130,-----,24e0deb964b39910c9dca7d72c45feb9ec1c6a9237342019549fb38a178ba63d0d980b1a1e26626202a373c84dce444cbb5ff6ad4c82458d6928b8b8f595be47&ini=1331334000000&tipo=month&l=a&appId=UOC
83.60.13.167,Spain-Olot-Catalonia,5ae84ab8f4e40c6d3c34bb3d2d46ae235d30e57e0c01e1cf1339d858da71f58181d24e34b968fb277829bd02162b8510408deb0bab113cdac35ad025fecb4960
85.50.152.118,Spain-Badalona-Catalonia,71ab96e67e040d76a47bdc5e32499d492e7347f68a6f4c4d282004b71314082f064a4ac74f341c892ed924ef29719af29a526c3bdf063e0fa4559098b5904f46
85.60.108.1,Spain-Madrid-Madrid,73c4cdf5148a8355598ccc7860448ffc38009b90be11fea4a00e1067492f4fbac531b7fce83a55cab6945fa75cfbbd1ea9e65174c92e5458f8804e0d13300236
213.98.187.100,Spain-Madrid-Madrid,92b43eeba3fba110635e8e622052ff365387adf46c393336cd688c496c1389fdb04705b3c29e17b18935d0267f4f3b59223485a7d30ceed990afc6ba09f99735
77.210.26.134,Spain-Rosal-Galicia,aba044bd62159c9635cf71f04f794dbe7dd0e82545ea2590c3b493e922a86302c2370c888b7b7ae34a63029c25c550c19eda5db64d404880dc7cbdd04f37301d
79.148.245.220,Spain-Madrid-Madrid,aba9768c096444380ce97fe4f82fa904ba9914e9bf620a84909396c16042c257015f2a9a36d5f4acbf4ae58d61006605e6212e071d2757189f59013621080990&appid=%27%20+%20_user_app%20+%20
213.231.109.75,Spain-Las Palmas de Gran Canaria-Canarias,c99d31548991f68b9b212195278e5627e9f8f06abb9efdcd3c825ca8144bbd760e690f9674e7efaed4141192563054 41b5c7e230424112fd6a943fe12d3e34a7
82.158.75.115,Spain-Madrid-Madrid,ca4622686cd64673abff5d66dc4ba450b6bca0c375eee8af31e308eee9788d05a8e4115a74862084ff6150abece69d4794e29c89e8c29f573e0166ac7b61c800
62.37.70.212,Spain-Vic-Catalonia,d11a917ae0e6d860bf824ca419d1cc023b55a45c1e0aff82043cbde00e4da8cf4f35f81f3e9ac680114d955dc7e54bad34158b1728634b246e28a8e0c60c8820
77.227.12.203,Spain-Madrid-Madrid,df091ad45139000e30f2afa582e2bbc5c344769ca4dcc68832b139e58517b6ab02b999291d8b45ef50f58ec63c1ab68c6d956238863b628e558e9e95087b403a
79.150.91.165,Spain-Madrid-Madrid,e4429af02266fa55772070575ac39b50fd79814861d76445761f41293f877f65c4641d4272b8424beb8ce1b8fa6000b862b9a78e55c5fd1634da8a5705a63371
188.77.169.8,Spain-Amposta-Catalonia,ef79c55f1da0963b2e7b309d2164db416e0581eaae2c2e5bc052a854641782070 4a2e2832eabeb9d61eb14fce699b820e5998c624233bc51da8e4ac681d8d563
79.155.224.81,Spain-Lleida-Catalonia,f1f01dedfda360ea8163fcb18fe3e2a898ff71f59f181d6f3ff29dc5f88309a3081914a1340e4bf45ef3c3c6beadc5384fb96822ddea1ba70a1c441fd6be319f
81.202.216.29,Spain-Valencia-Comunidad Valenciana,fc0f3963a2ec6e333f48312fc8766cfeb1feefc3927a1fcd9c6fd57ace4d3453dda37f732e49d5754a4c18c76441d9a4413a0b930032363a185f2597b9ce625f
88.26.196.48,Spain-Viladecans-Catalonia,fcb225feff8b07fc0e4d04fadbcc6b716d8fca3b900739c1ba6b115ab1b84f8e1f5c38a297073e5a126ce98a263853c1760c15670cb48fb045ec12b788a9f13e
```


Como opción de mejora al proyecto, además de la exportación en CSV, se ha añadido por defecto que el fichero intermedio genere un fichero XML con la información estructurada. Tendrá el siguiente formato (para el caso de la opción 1):

```
<?xml version="1.0" encoding="UTF-8"?>
<ArrayOfRegOpcion1 xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <RegOpcion1>
      <IP>Direccion IP</IP>
      <RecursoRecurso</Recurso>
      <Sesion>Id de Sesion</Sesion>
      <Indice>0</Indice>
  </RegOpcion1>
</ArrayOfRegOpcion1>
```

```xml
<?xml version="1.0" encoding="UTF-8"?>
- <ArrayOfRegOpcion1 xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  - <RegOpcion1>
      <IP>88.13.89.216</IP>
      <Recurso>/UOC/templates/Mail/ma_bodyHeight.html</Recurso>
      <Sesion>003a28a6f28c933a37a076da3f00c63e6965e23b5b1ef601b3a208511bc1ffc9e83908efc330ee3b14bde8b34f7aa8da4292280a59f6c26241786a81ffd5f2d6</Sesion>
      <Indice>0</Indice>
    </RegOpcion1>
  - <RegOpcion1>
      <IP>88.13.89.216</IP>
      <Recurso>/UOC/templates/Mail/ma_bodyHeight.html</Recurso>
      <Sesion>003a28a6f28c933a37a076da3f00c63e6965e23b5b1ef601b3a208511bc1ffc9e83908efc330ee3b14bde8b34f7aa8da4292280a59f6c26241786a81ffd5f2d6</Sesion>
      <Indice>0</Indice>
    </RegOpcion1>
  - <RegOpcion1>
      <IP>79.148.251.140</IP>
      <Recurso>/webapps/webUtils/servlet/InvitationChatServlet</Recurso>
      <Sesion>01ae78f4de7d954717b362d0aedfe3f44071ab31d7c2c6d4ff87a894d1c6f6c85b61c5ab806c4d486ffa0946bd9c5d70f179398252c4a32333df02963c8005c2</Sesion>
      <Indice>0</Indice>
    </RegOpcion1>
  - <RegOpcion1>
      <IP>79.148.251.140</IP>
      <Recurso>/webapps/webUtils/servlet/InvitationChatServlet</Recurso>
      <Sesion>01ae78f4de7d954717b362d0aedfe3f44071ab31d7c2c6d4ff87a894d1c6f6c85b61c5ab806c4d486ffa0946bd9c5d70f179398252c4a32333df02963c8005c2</Sesion>
      <Indice>0</Indice>
    </RegOpcion1>
  - <RegOpcion1>
      <IP>88.19.144.28</IP>
      <Recurso>/webapps/webUtils/servlet/InvitationChatServlet</Recurso>
      <Sesion>01aef2bf53e64810559dfc4bb75fda05fbeb2c16c75e6983a9022bd99695cc7d1ed0cf98a1bc445766cbaa693e7a1f791539b0b1da29356ede46be3fb3c72667</Sesion>
      <Indice>0</Indice>
    </RegOpcion1>
  - <RegOpcion1>
      <IP>88.19.144.28</IP>
      <Recurso>/webapps/webUtils/servlet/InvitationChatServlet</Recurso>
      <Sesion>01aef2bf53e64810559dfc4bb75fda05fbeb2c16c75e6983a9022bd99695cc7d1ed0cf98a1bc445766cbaa693e7a1f791539b0b1da29356ede46be3fb3c72667</Sesion>
      <Indice>0</Indice>
    </RegOpcion1>
  - <RegOpcion1>
      <IP>84.126.53.217</IP>
      <Recurso>/webapps/webUtils/servlet/InvitationChatServlet</Recurso>
      <Sesion>01f532692c0465f9b35c9b59f33a951714eba48fd9c837b01879b66bb35118c3659b0ce418f1894cdc509c2252923f11f30a6e01c715f9f46d3968cf7d091729</Sesion>
```


La estructura seria análoga para el resto de las 5 opciones restantes.

Con el fichero intermedio generado, cabrían dos opciones.

1. Introducirlo en el analizador de datos, Weka, y estudiar los resultados obtenidos.
2. A partir de él, generar el fichero final.

6.5. Generar fichero final

Llegados a este punto ya hemos generado el fichero intermedio y hemos decidido generar el fichero final para introducirlo posteriormente en la herramienta de análisis de datos Weka.

6.5.1. Fichero final Opción 1- Recursos accedidos

A partir del fichero intermedio de Opción 1 Generado previamente, se habrá de generar otro fichero de exportación de datos, CSV, en otro formato para su posterior introducción en WEKA.

El formato será una matriz de datos con una estructura muy concreta, tantas filas como sesiones diferentes y con tantas columnas como Recursos, se navegue. Y en las intersecciones, si una sesión X ha navegado por el recurso Y se pondrá un “T” en la intersección X, Y, en caso contrario habrá que poner “F”.

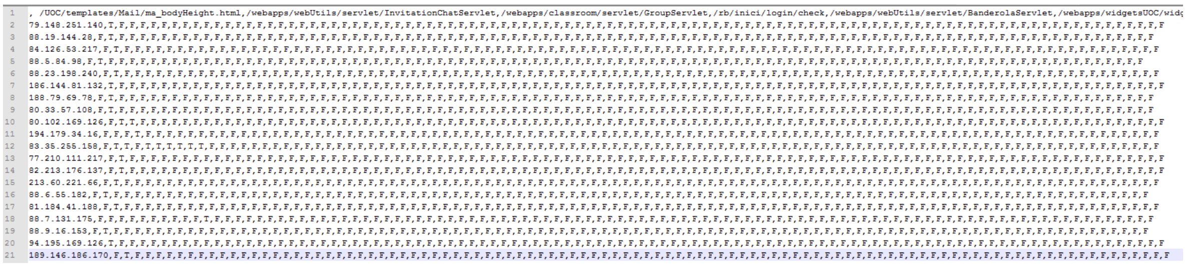
De esta forma, dadas las N direcciones únicas identificadas en el paso anterior, la apariencia de la cabecera será similar a la siguiente:

- IP
- Recurso_1
- Recurso_2
- …
- Recurso_N

Un posible ejemplo seria:

IP,uoc/recurso1,uoc/recurso2,uoc/recurso3
111.111.111.111,T,T,F
222.222.222.222,T,F,F
333.333.333.333,T,F,T
444.444.444.444,F,F,T
555.555.555.555,T,T,T
111.111.111.111,T,F,F

Puede observarse que se distingue entre la secuencia de acciones realizada por la dirección IP 111.111.111.111 durante una sesión inicial de trabajo, de la secuencia de acciones realizada por la misma dirección IP en una sesión de trabajo diferente



NOTA: El resto de opciones se deducen claramente de la anterior. No se especifican más.

6.5.2. Fichero final Opción 2- Periodos de tiempo de navegación.

6.5.3. Fichero final Opción 3- Respuestas del servidor.

6.5.4. Fichero final Opción 4 -Tipos de recursos accedidos.

6.5.5. Fichero final Opción 5 - Exploradores web usados.

6.5.6. Fichero final Opción 6- Ubicación desde donde se navega.

6.6. Analizar datos con Weka

Como ultima tarea a realizar ya solo queda analizar los datos obtenidos después del preprocesado de la información. Esto se hará, como hemos indicado anteriormente con una herramienta de análisis de datos y modelado predictivo que, Weka, que hemos definido anteriormente en el punto 5.5.1

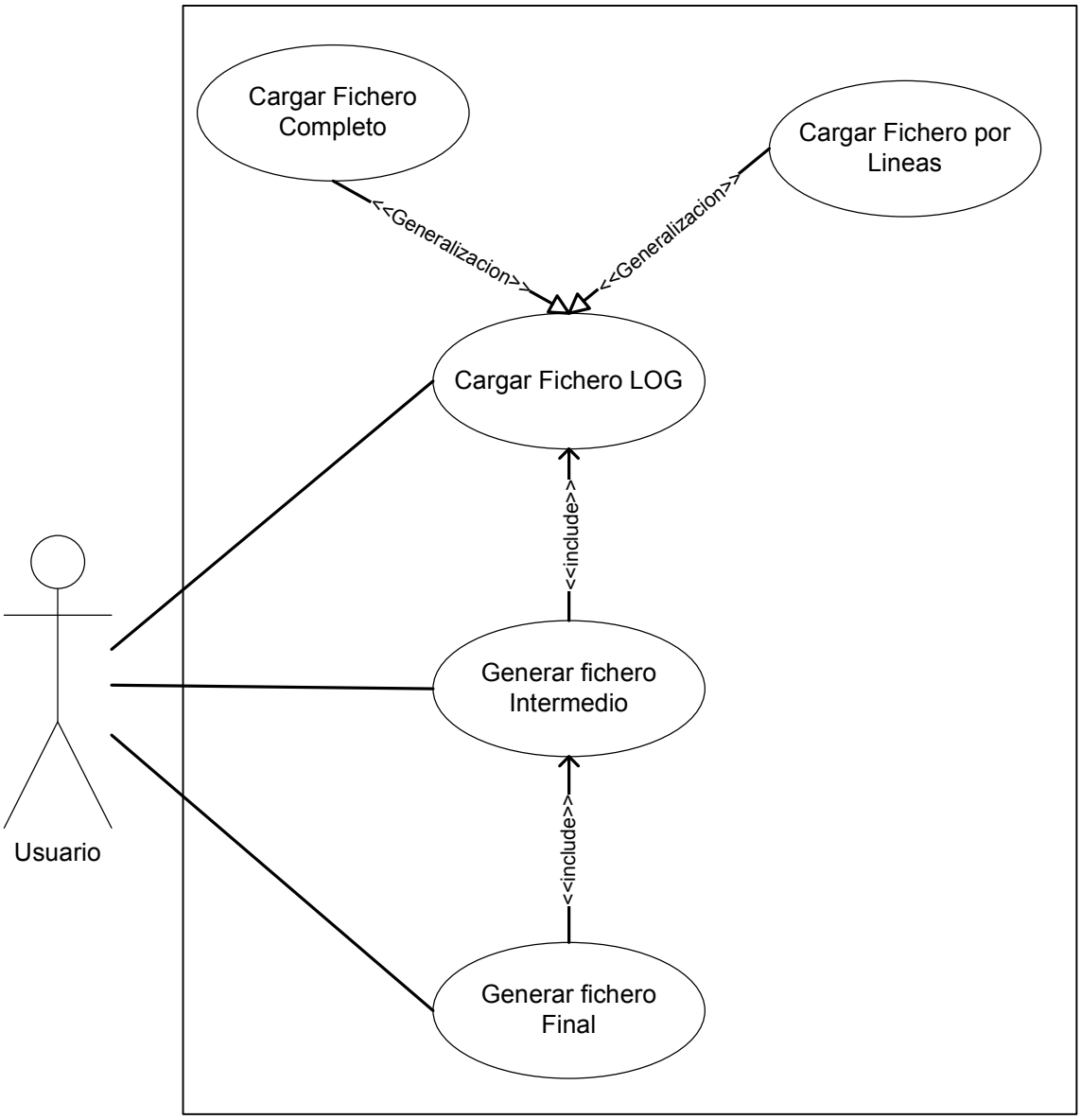
7. Estudio funcional de la aplicación

Como se ha indicado anteriormente (punto 4.1.1.),se partirá de un fichero de texto de LOGs generados por un servidor Web apache, con Informacion desestructurada y se pretende preprocesarla para que esta sea “importable” por herramientas de análisis de datos (weka).

En este apartado diseñaremos las especificaciones oportunas para la construcción del software.

7.1. Casos de uso

Definiremos los diversos casos de uso de la aplicación.



Actores

El único actor que actuará en todos los casos de uso es aquel usuario que necesite obtener datos estadísticos de las navegaciones de los usuarios a través del portal de la UOC.

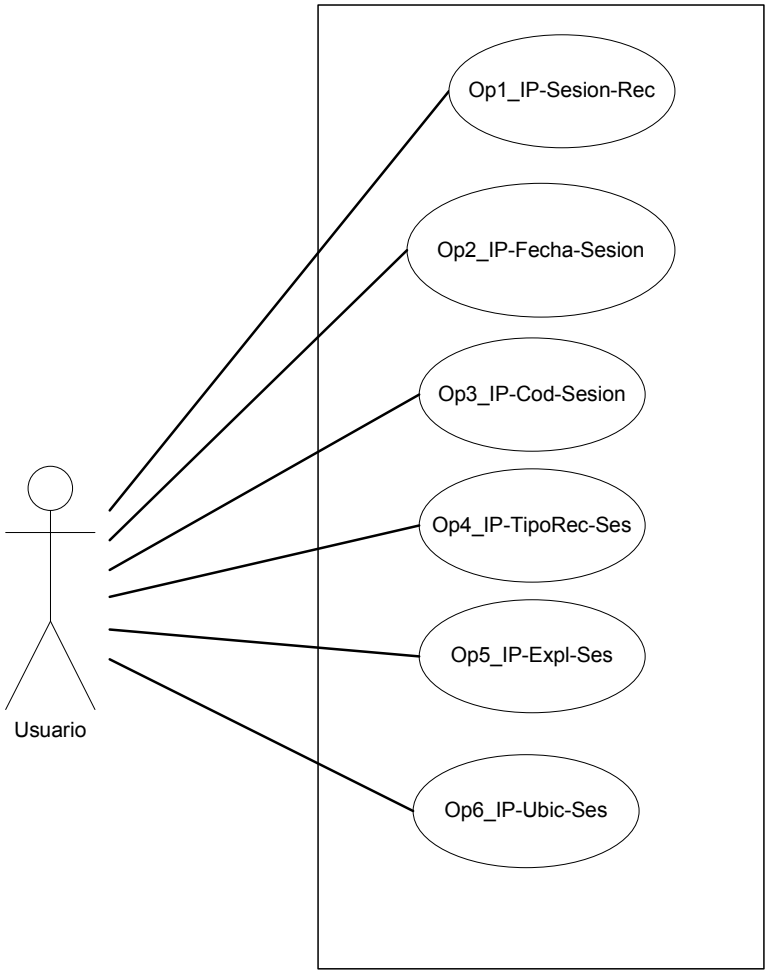
Para ello, a partir del fichero LOG que haya generado el servidor Web Apache, se cargará en la aplicación, se procesarán las líneas y se generará un archivo de exportación CSV, que posteriormente se estudiará mediante Weka.

CASO DE USO	**Cargar Fichero LOG**
Identificador	CU-001
Descripción	Apertura y carga en memoria un fichero LOG del Campus Virtual de la UOC.
Precondiciones	El fichero debe tener un formato valido

Secuenc. normal	Se trata de una generalización de los dos casos de uso posteriores
Secuencia alt.	Se define en los dos puntos posteriores
PostCondiciones	Se define en los dos puntos posteriores

CASO DE USO	**Cargar Fichero LOG Completo**
Identificador	CU-002
Descripción	Apertura y carga en memoria un fichero LOG del Campus Virtual de la UOC en su totalidad.
Precondiciones	El fichero debe tener un formato valido
Secuenc. normal	1. El actor demanda cargar un fichero LOG alojado en la ruta correcta 2. El actor indica si se consolida la Informacion del LOG en BD o no. 3. El sistema lee el fichero en su totalidad, procesa la Informacion estructurándola y guardándola en memoria 4. Adicionalmente si se ha indicado también lo guarda en BD.
Secuencia alt.	En caso de error presenta mensaje
PostCondiciones	A la salida el fichero estará estructurado en estructuras de memoria y adicionalmente puede estar consolidado en BD.

CASO DE USO	**Cargar Fichero LOG por filas**
Identificador	CU-003
Descripción	Apertura y carga en memoria un fichero LOG del Campus Virtual de la UOC desde una linea inicio hasta una linea final.
Precondiciones	El fichero debe tener un formato valido
Secuenc. normal	1. El actor demanda cargar un fichero LOG alojado en la ruta correcta 2. El actor indica la línea desde la que se produce la lectura y la línea hasta la que se lee. 3. Igualmente indica si se consolida la Informacion del LOG en BD o no. 4. El sistema lee el fichero entre los limites indicados, procesa la Informacion estructurándola y guardándola en memoria 5. Adicionalmente si se ha indicado también lo guarda en BD.
Secuencia alt.	En caso de error presenta mensaje
PostCondiciones	A la salida el fichero estará estructurado en estructuras de memoria y adicionalmente puede estar consolidado en BD.

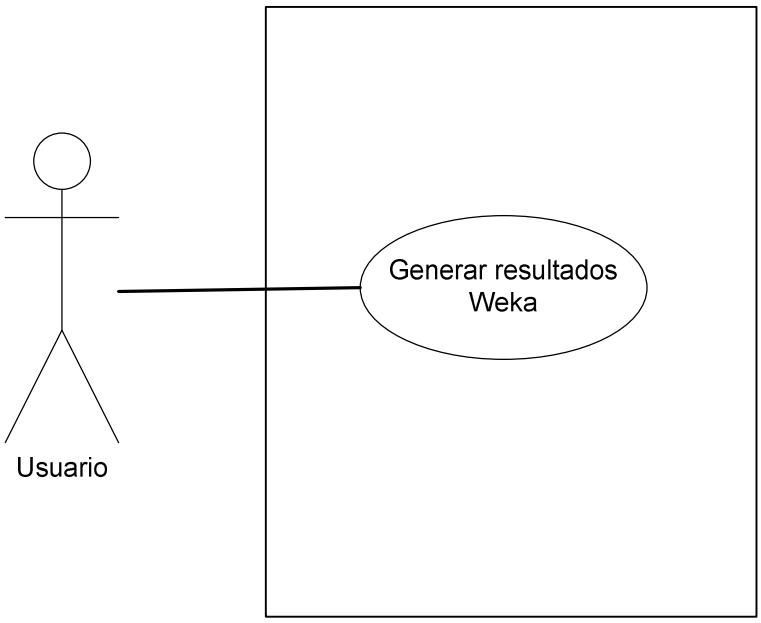


CASO DE USO	**Op1_IP-Sesion-Recursos accedidos**
Identificador	CU-004
Descripción	Generación del fichero intermedio (opción 1). **Recursos accedidos**
Precondiciones	El fichero de LOG debe estar cargado correctamente en memoria.
Secuenc. normal	1. El actor habrá escogido la opción de Fichero intermedio que desea generar 2. A partir de ella, se generará el fichero CSV y XML con los datos adecuados a la elección. Como se ha definido en el pto 6.4
Secuencia alt.	En caso de error se muestra mensaje
PostCondiciones	Se generará dos ficheros de exportación de datos, uno CSV y otro XML.

NOTA: El resto de opciones, 2, 3, 4, y 5 son análogas y no se describen

CASO DE USO	**Op6_IP-Sesion-Ubicación desde donde se navega**
Identificador	CU-005

Descripción	Generación del fichero intermedio (opción 6). **Ubicación desde donde se navega**
Precondiciones	El fichero de LOG debe estar cargado correctamente en memoria.
Secuenc. normal	1. El actor habrá escogido la opción de Fichero intermedio que desea generar 2. A partir de ella, se generará el fichero CSV y XML con los datos adecuados a la elección. Como se ha definido en el pto 6.4 3. El sistema deberá preguntar a un sistema interno o externo la ubicación que tiene una determinada IP
Secuencia alt.	En caso de error se muestra mensaje
PostCondiciones	Se generará dos ficheros de exportación de datos, uno CSV y otro XML.



NOTA: El resto de opciones, 2, 3, 4, y 5 son análogas y no se describen

CASO DE USO	**Generar fichero final Op1-6**
Identificador	CU-006
Descripción	Generación del fichero final (opciones 1 a 6).
Precondiciones	1. El fichero de LOG debe estar cargado correctamente en memoria. 2. El fichero intermedio debe estar generado
Secuenc. normal	1. El actor habrá escogido la opción de Fichero Final que desea generar 2. A partir de ella, se generará el fichero CSV con los datos adecuados a la elección. Como se ha definido en el pto 6.5
Secuencia alt.	En caso de error se muestra mensaje

PostCondiciones	Se generará dos ficheros de exportación de datos, uno CSV y otro XML.

7.2. DISEÑO TÉCNICO

7.2.1. INTRODUCCION

El estudio técnico estará organizado de la siguiente forma:

- **Necesidades de Arquitectura**: Introducción global de las necesidades de arquitectura.
- **Tecnologías**: Explicación de las decisiones tecnológicas tomadas para llevar a cabo este proyecto.
- **Diseño de arquitectura**: Arquitectura a utilizar.
- **Modelo de datos**.
- **Diagrama de clases**

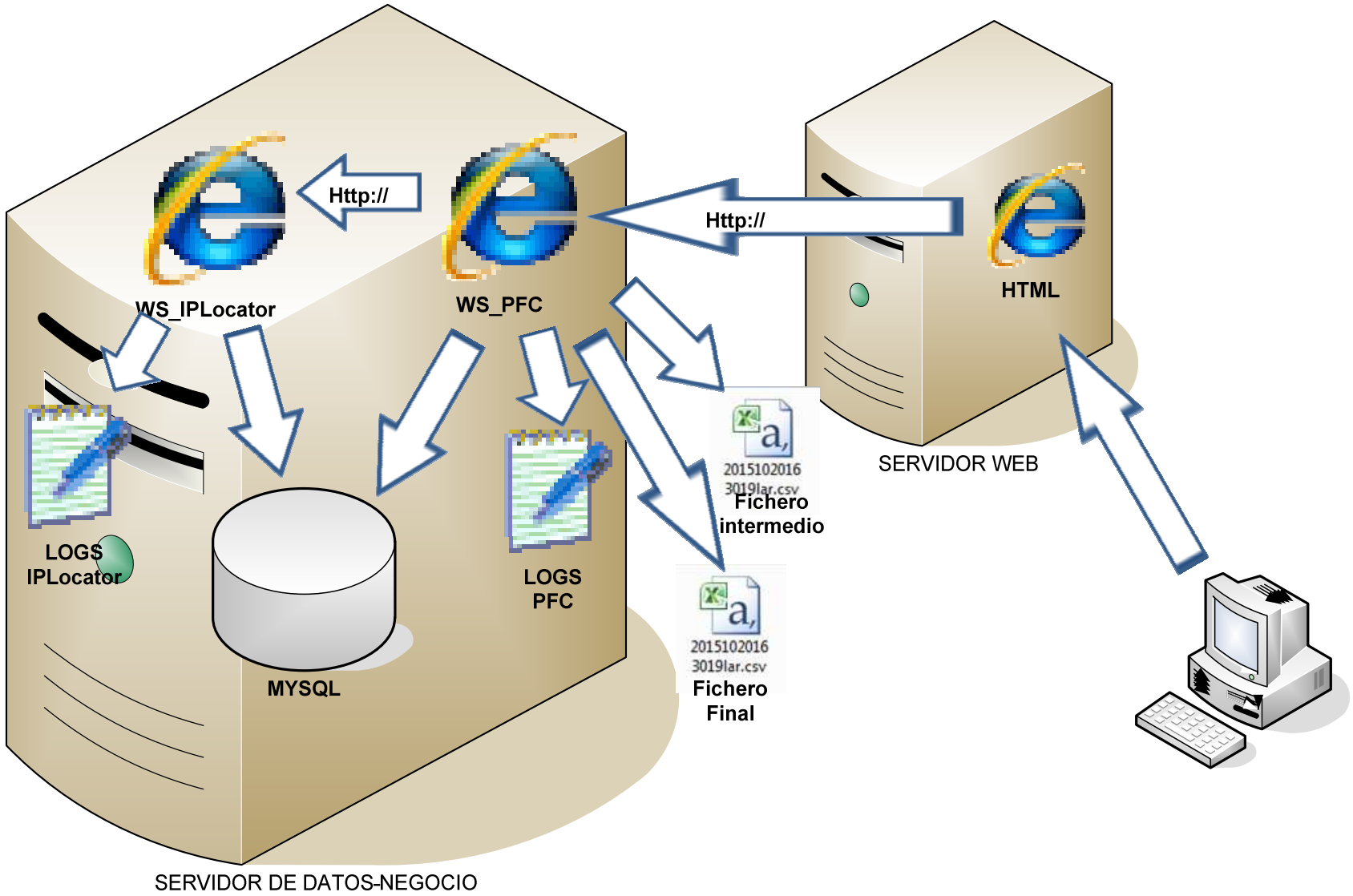
7.2.2. NECESIDADES DE ARQUITECTURA

El proyecto debe cumplir los siguientes objetivos:

1. Dotar de información al usuario que permita estudiar los patrones de navegación dentro del Campus de la UOC
2. Además de los patrones de navegación, se deberá permitir el estudio de todo lo que la fuente de información permita. Ficheros LOG del Web server.
3. El sistema deberá estar diseñado siguiendo estándares abiertos. HTTP, XML, SOA y Web Services
4. Incorporar ciertas mejoras adicionales a lo expresamente requerido.
    a. Fácil parametrización de aplicación
    b. Consolidación de datos en BD.
    c. Atacar a varios motores de BD siendo transparente al usuario
    d. Exportación de datos a lenguajes estructurados. XML.
    e. Incorporar LOGs de sistema que permitan auditoria de acciones y mejorar solución de errores correctivos.
5. Diseño siguiente buenas practicas, estructurado, flexible, eficiente,…
6. Paradigmas de Sistemas Distribuidos.
    - Estándares abiertos
    - Arquitectura en tres capas.
    - Arquitectura en diferentes servidores

Para conseguir lo anterior se montará una arquitectura en tres capas sobre una arquitectura de sistemas de dos servidores físicos, que se comunicará entre ellos mediante llamas HTTP.

Abajo se incorpora un esquema.


Http://
WS_IPLocator
WS_PFC
Http://
HTML
SERVIDOR WEB
LOGS
IPLocator
MYSQL
LOGS
PFC
2015102016
3019lar.csv
Fichero
intermedio
2015102016
3019lar.csv
Fichero
Final
SERVIDOR DE DATOS-NEGOCIO

7.2.3. TECNOLOGIAS

Acabamos de ver cuál será el esquema necesario. Para conseguir esto usaremos la siguiente tecnología:

1. **VirtualBOX**. Es un software que permite la virtualización de máquinas físicas. Con el que podremos tener varios servidores virtuales en una maquina física.
   https://es.wikipedia.org/wiki/VirtualBox
   https://www.virtualbox.org/

2. **Microsoft Windows 2003 Server Standard Edition R2**. En nuestro caso usaremos licencias de evaluación. Se trata de un sistema operativo de Microsoft para servidores
   https://es.wikipedia.org/wiki/Windows_Server_2003

3. **MySQL Database.** Motor de bases de datos que permite el almacenamiento de datos**.**
   http://www.mysql.com/
   https://es.wikipedia.org/wiki/MySQL

4. **MySQL Workbech.** Herramienta visual de gestión de base de datos MySQL**.**

5. **Drivers de conexión para .Net a base de datos MySQL**. Permitirá la conexión desde .Net a bases de datos MySQL

6. **IIS. Internet Information Server**. Servidor web que deberá estar instalado y configurado en el Servidor de negocio y en el Servidor web, permite gestionar peticiones http.

7.2.4. DISEÑO DE LA ARQUITECTURA

7.2.4.1. Entorno de producción

La arquitectura anterior se ha desplegado, por medio de Virtual BOX, dos máquinas virtuales Windows 2003 Server R2 (versión de evaluación). Servidor de datos y negocio y otra de servidor WEB.

- Servidor Datos y Negocio
  - Windows 2003 Server Standard Edition R2, actualizado
  - Se crea una LAN 172.29.100./24 y se le asigna una IP del rango.
  - Aunque el Programa está preparado para motor de BD SQL Server y MySQL. En esta ocasión nos hemos decantado por MySQL. Y se desplegará un motor de BD MySQL con dos instancias:
    1. Instancia para la consolidación de datos de los LOG.
    2. Instancia “IPlocation_db11” donde se encuentran los datos de las ubicaciones de las IPs.
  - Se despliega el WS de IPLocator en una ruta del Servidor

- Se configura un Directorio Virtual. Para permitir el acceso de llamadas http.
- Y de la misma forma se desplegaría el WS_PFC y el directorio virtual en el IIS para que la capa WEB pueda acceder. A partir de aquí ya tendríamos montado el Servidor de Datos y Negocio y permitiría obtener datos en bruto.
- Se puede probar el funcionamiento en este punto del programa por medio de programas que ataquen a WebService como por ejemplo el SOAPIU, el cual se puede bajar una versión gratuita en:
  http://www.soapui.org/
  https://en.wikipedia.org/wiki/SoapUI

- Servidor WEB

  Dada la flexibilidad del sistema se podría programar la capa de presentación con cualquier interfaz. Sea un móvil, un cliente pesado,… en nuestro caso lo programaremos como un cliente ligero WEB que ataque al WS residente en la máquina de Negocio.

  Así pues se trata de una maquinal virtualizada con Virtual Box o VMWare y esta seria de las siguientes características:
  - Windows 2003 Server Standard Edition R2, actualizado
  - LAN 172.29.100./24 y se le asigna una IP del rango.
  - Habría que configurarse el IIS y un Directorio Virtual o Sitio web que aloje la parte WEB.
  - Hecho esto solo quedaría desplegar el proyecto en el servidor WEB.

7.2.4.2. Localizador IPs

En puntos anteriores hemos ido explicando brevemente que en la generación de ficheros intermedios y en la generación de ficheros finales, en la opción 6- Ubicación desde donde se navega, se precisaba un servicio externo que devolviese la ubicación exacta a la que pertenece una determinada IP. Este servicio podría ser externo, ubicado en internet y que la aplicación se conectase en cada IP que se estudie, o interno, el cual será necesario importar a nuestros sistemas.

Al igual que el resto del aplicativo se intentará cumplir todas las condiciones y decisiones tomadas respecto a calidad del software, protocolos abiertos, flexibles,…

Después de valorar diferentes opciones que ofrece el mercado me he decantado por importar un sistema en forma de WebService que suministre las ubicaciones para cada IP.

Para ello, necesitaremos una fuente externa de datos que permita obtener este dato para cada una de las IPs del patrón de navegación. Actualmente en el mercado existen diferentes opciones pero nos hemos decantado por la siguiente opción:

www.ip2location.com. Se trata de una página en donde permite descargarse una base de datos MySQL o SQL Server con los diferentes rangos de IPs asignados a cada ubicación geográfica mundial con el siguiente modelo de datos.

ip2location_db11	
PK PK	**ip_from** **IP_to**
	country_code country_name region_name city_name latitude longitude zip_code time_zone

De esta forma se necesitará montar un motor de base de datos en alguna tecnología de las anteriormente citada e implementar un método para consultar dinámicamente por cada IP de la navegación, que ubicación tiene. Por qué se requiere dinamismo, utilizar protocolos abiertos y estándares actuales, se ha optado por implantar un WebService que permita esta consulta con el siguiente interfaz (WSDL)

```
<?xml version="1.0" encoding="UTF-8"?>

<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
targetNamespace="http://tempuri.org/"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
xmlns:s="http://www.w3.org/2001/XMLSchema" xmlns:tns="http://tempuri.org/"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:tm="http://microsoft.com/wsdl/mime/textMatching/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/">
<wsdl:types>
<s:schema targetNamespace="http://tempuri.org/" elementFormDefault="qualified">
<s:element name="BuscaIP"><s:complexType>
<s:sequence>
<s:element name="IP" type="s:string" maxOccurs="1" minOccurs="0"/>
</s:sequence>
</s:complexType>
</s:element>
<s:element name="BuscaIPResponse">
<s:complexType>
<s:sequence>
<s:element name="BuscaIPResult" type="tns:RegistroResultado" maxOccurs="1"
minOccurs="1"/>
</s:sequence>
</s:complexType>
</s:element>
<s:complexType name="RegistroResultado">
<s:sequence>
<s:element name="IP_Inicio" type="s:string" maxOccurs="1" minOccurs="0"/>
<s:element name="IP_Fin" type="s:string" maxOccurs="1" minOccurs="0"/>
<s:element name="CodigoPais" type="s:string" maxOccurs="1" minOccurs="0"/>
```

```
<s:element name="Pais" type="s:string" maxOccurs="1" minOccurs="0"/>
<s:element name="Region" type="s:string" maxOccurs="1" minOccurs="0"/
><s:element name="Ciudad" type="s:string" maxOccurs="1" minOccurs="0"/>
<s:element name="Latitud" type="s:string" maxOccurs="1" minOccurs="0"/>
<s:element name="Longitud" type="s:string" maxOccurs="1" minOccurs="0"/
><s:element name="CodigoZip" type="s:string" maxOccurs="1" minOccurs="0"/>
<s:element name="ZonaHoraria" type="s:string" maxOccurs="1" minOccurs="0"/>
<s:element name="Resultado" type="tns:Resultado" maxOccurs="1" minOccurs="1"/>
</s:sequence>
</s:complexType>
<s:complexType name="Resultado">
<s:sequence>
<s:element name="HayError" type="s:boolean" maxOccurs="1" minOccurs="1"/>
<s:element name="Codigo" type="tns:CodigosError" maxOccurs="1" minOccurs="1"/>
<s:element name="Resultado" type="s:string" maxOccurs="1" minOccurs="0"/>
</s:sequence>
</s:complexType>
<s:simpleType name="CodigosError">
<s:restriction base="s:string">
<s:enumeration value="Error_Inesperado"/>
<s:enumeration value="Exito"/>
<s:enumeration value="Warning"/>
</s:restriction>
</s:simpleType>
</s:schema>
</wsdl:types>
<wsdl:message name="BuscaIPSoapIn">
<wsdl:part name="parameters" element="tns:BuscaIP"/>
</wsdl:message>
<wsdl:message name="BuscaIPSoapOut">
<wsdl:part name="parameters" element="tns:BuscaIPResponse"/>
</wsdl:message>
<wsdl:portType name="Service1Soap">
<wsdl:operation name="BuscaIP">
<wsdl:input message="tns:BuscaIPSoapIn"/>
<wsdl:output message="tns:BuscaIPSoapOut"/>
</wsdl:operation>
</wsdl:portType>
<wsdl:binding name="Service1Soap" type="tns:Service1Soap">
<soap:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="BuscaIP">
<soap:operation style="document" soapAction="http://tempuri.org/BuscaIP"/>
<wsdl:input>
<soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="Service1Soap12" type="tns:Service1Soap">
<soap12:binding transport="http://schemas.xmlsoap.org/soap/http"/>
<wsdl:operation name="BuscaIP">
<soap12:operation style="document" soapAction="http://tempuri.org/BuscaIP"/>
<wsdl:input>
<soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
<soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:service name="Service1">
<wsdl:port name="Service1Soap" binding="tns:Service1Soap">
<soap:address location="http://localhost:1437/WS_IPLocator.asmx"/>
</wsdl:port>
```

```
<wsdl:port name="Service1Soap12" binding="tns:Service1Soap12">
<soap12:address location="http://localhost:1437/WS_IPLocator.asmx"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>
```

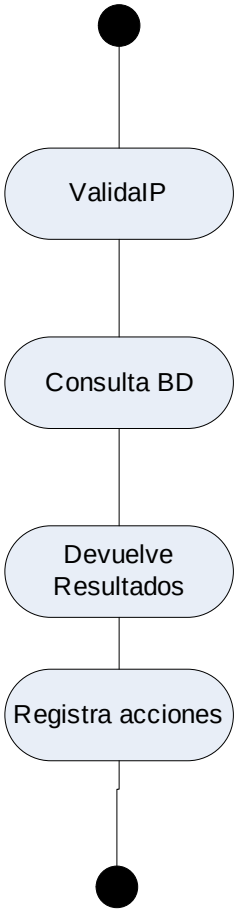
Se publicará en el Servidor de datos al que se consultará vía http al Webservice que, como se ve en el WSDL, tendrá una única operación:

- BuscaIP (Ip) , con un parámetro de entrada tipo cadena en el que se mete una Direccion IP valida y tras consulta a la BD devolverá una estructura con los resultados obtenidos en caso que sean correctos.

```
Structure RegistroResultado
    IP_Inicio As String
    IP_Fin As String
    CodigoPais As String
    Pais As String
    Region As String
    Ciudad As String
    Latitud As String
    Longitud As String
    CodigoZip As String
    ZonaHoraria As String
    Resultado As Resultado
End Structure
```

Donde:

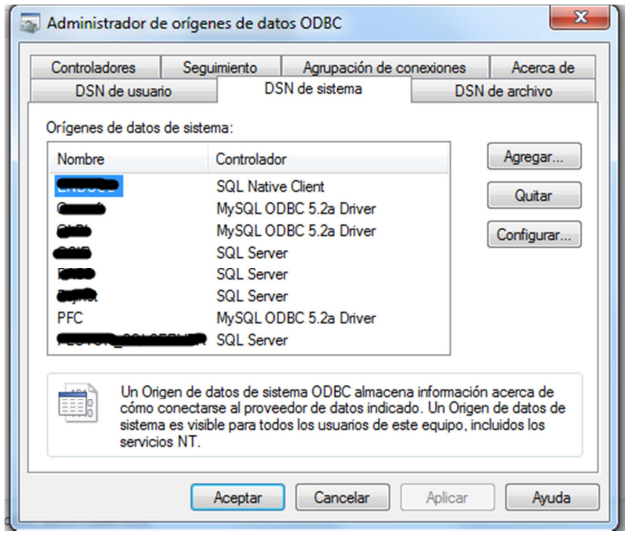
- Ip_*Inicio*: Inicio del rango asignado a la zona geográfica a la cual pertenece la IP pasada como parámetro y que hay que ubicar
- Ip_*Fin*: Fin del rango asignado a la zona geográfica a la cual pertenece la IP pasada como parámetro y que hay que ubicar.
- *CodigoPais:* Código del Pais resultado de la búsqueda de la IP
- *Región:* Región dentro del país donde se ubica la IP
- *Ciudad:* Ciudad a la que pertenece la IP
- *Latitud:* Latitud a la que pertenece la IP
- *Longitud:* Longitud a la que pertenece la IP
- *CodigoZip:* CodigoZIP p código postal al que pertenece la IP
- *ZonaHoraria:* Zona horaria a la que pertenece la ubicación de la IP
- *Resultado:* Cadena de texto con el resultado de la operación de búsqueda.
    - Si todo ha ido correcto devolverá OK
    - Si ha habido error, devolverá descripción del error



7.2.4.3. Entorno de programación.

El desarrollo del Proyecto se está realizado en el siguiente entorno:

- Tener instalado un ODBC llamado “PFC”



Para poder instalar el ODBC habrá que haberse descargado el driver que permite el acceso a Bases de datos MySQL

https://dev.mysql.com/downloads/connector/net/6.9.html

- Windows 7 Professional versión 6.1. Service Pack 1
- Visual Studio 2005 Version 8.0.50727

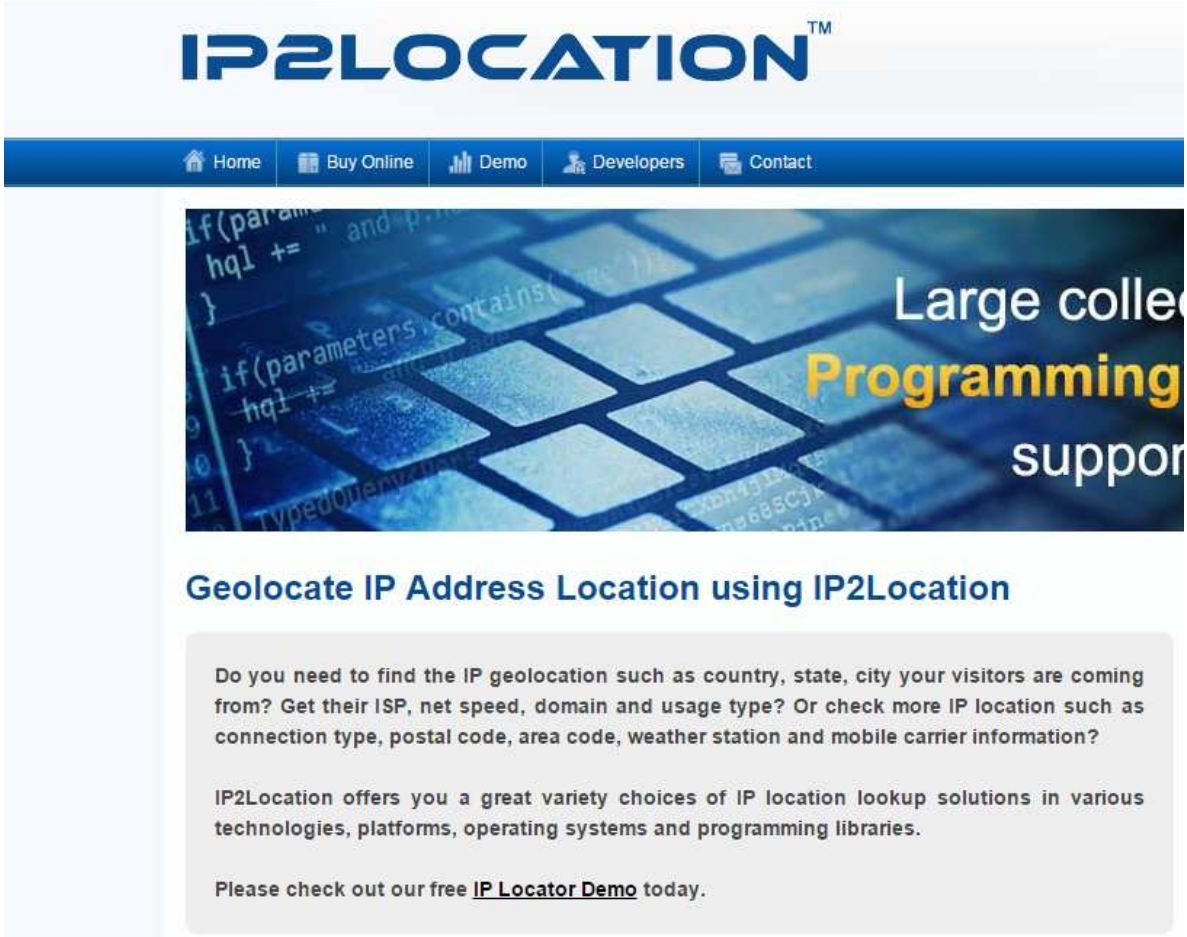
https://es.wikipedia.org/wiki/Microsoft_Visual_Studio

7.2.5. Modelo de datos.

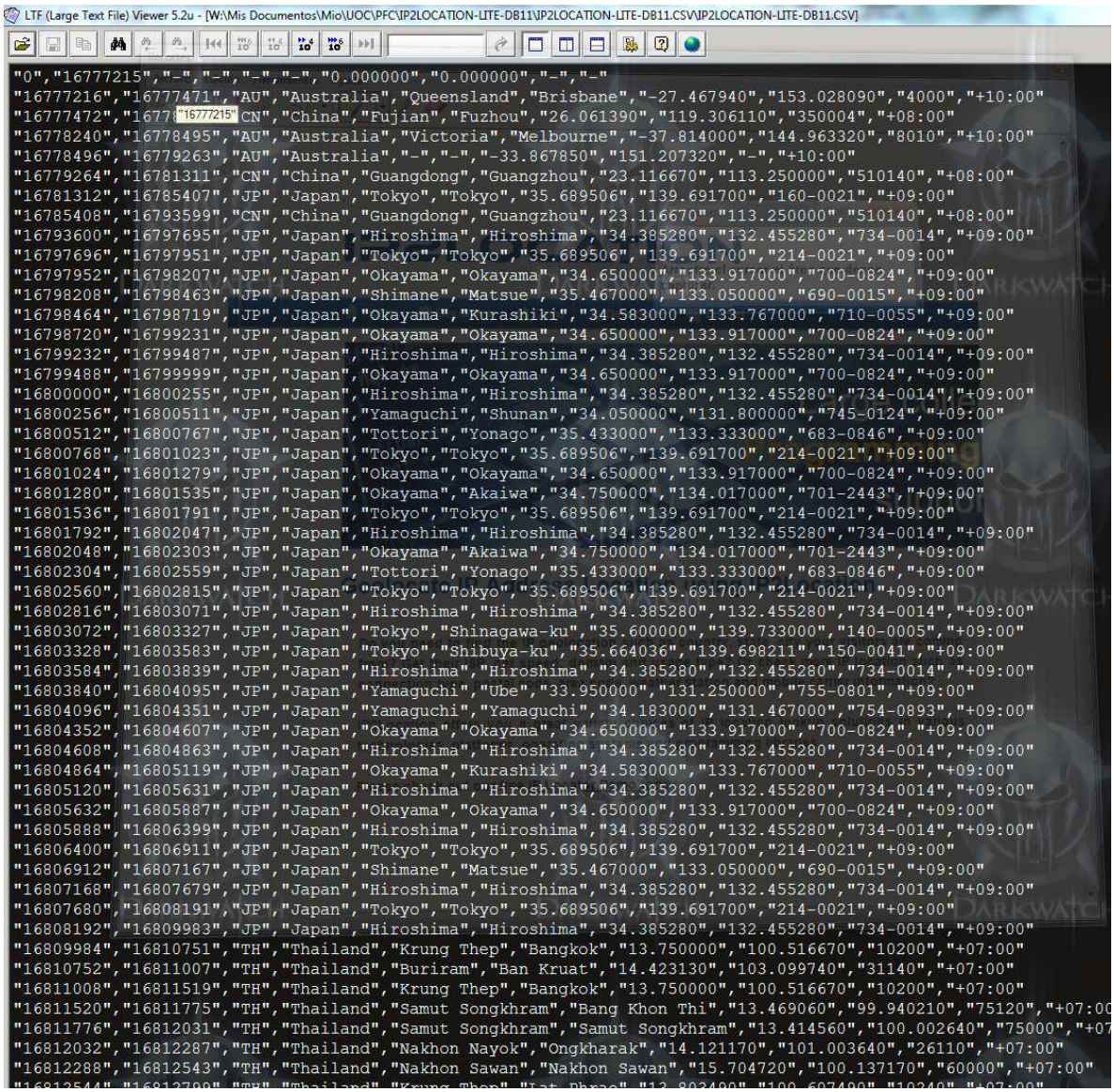
Se ha diseñado el Proyecto actual para que consolide un cierto nivel de información, si bien es cierto, este tipo de información no va a ser muy complejo, sera de dos tipos

7.2.5.1. IP2Location_db11

Se explicó en parte en el punto anterior, su utilidad será almacenar las diferentes ubicaciones que hay para determinados rangos de IPs. Se trata de una base de datos que está disponible en la web **www.ip2location.com** .



En ella nos descargaremos gratuitamente hasta la versión DB11 un archivo CSV con los datos de todos los rangos de IPs y sus ubicaciones.



Además de los datos, disponemos de scripts que permiten importar el CSV a un determinado Motor de BD

Database Setup

Below are the steps to set up the database for both IPv4 and IPv6 data

IPv4 | IPv6

MSSQL | MySQL | PostgreSQL

Create table

```
CREATE DATABASE ip2location;
USE ip2location;
CREATE TABLE `ip2location_db11`(
        `ip_from` INT(10) UNSIGNED,
        `ip_to` INT(10) UNSIGNED,
        `country_code` CHAR(2),
        `country_name` VARCHAR(64),
        `region_name` VARCHAR(128),
        `city_name` VARCHAR(128),
        `latitude` DOUBLE,
        `longitude` DOUBLE,
        `zip_code` VARCHAR(30),
        `time_zone` VARCHAR(8),
        INDEX `idx_ip_from` (`ip_from`),
        INDEX `idx_ip_to` (`ip_to`),
        INDEX `idx_ip_from_to` (`ip_from`, `ip_to`)
) ENGINE=MyISAM DEFAULT CHARSET=utf8 COLLATE=utf8_bin;
```

Import the database

```
LOAD DATA LOCAL
        INFILE 'IP2LOCATION-LITE-DB11.CSV'
INTO TABLE
        `ip2location_db11`
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\r\n'
IGNORE 0 LINES;
```

Como se observa en el anterior script, la estructura de BD es la siguiente:

ip2location_db11	
PK **PK**	**ip_from** **IP_to**
	country_code country_name region_name city_name latitude longitude zip_code time_zone

Con los diferentes campos, tipos de datos y función de cada campo:

Nombre Columna	Tipo	Oblig.	Función
ip_from	Int(10)	SI	Ip de inicio del rango de IPs
IP_to	Int(10)	SI	IP de fin de rango de IPs

Country_code	Char(2)	SI	Codigo del país. ES, GB, FR, US,..
Country_name	Varchar(64)	SI	Nombre del país al que pertenece el rango de IPs
Region_name	Varchar(128)	SI	Region dentro del país
City_name	Varchar(128)	SI	Ciudad
Latitude	Doublé	SI	Latitud donde se ubica la IP
Longitude	Doublé	SI	Longitud donde se ubica la IP
Zip_Code	Varchar(30)	SI	ZIP/Codigo postal
Time_zone	Varchar(8)	SI	Diferencia horararia según GMT

Como se ve en la descripción de los campos, las IPs mundiales se asignan por zonas, es decir, para una zona tendremos un rango de IPs asignadas, es de esta forma como se determinará la ubicación de la misma. Así por ejemplo, Madrid, tendrá asignadas 1 o varias zonas desde la IP 111.111.111.111 hasta la IP 111.111.255.255. Y de idéntica forma para otro rango. No se mezclan ips de un rango-zona con otro.

Importante. Hay que hacer notar que los valores clave en las búsquedas "Ip_from" e "IP_to" son valores numéricos, es por ello que habrá que convertir las IPs, cadena de caracteres de 15 elementos, a valores numéricos.

7.2.5.2. LOG_UOC

Dado que la fuente primaria de datos son los ficheros de texto que produce el Servidor Web Apache de la UOC, y que se trata de un soporte muy lento y precario. Para permitir acceso rápido y una explotación de datos más eficiente Se ha diseñado el Proyecto de tal forma que el usuario podrá escoger si desea que los datos de los LOGs que vaya a leer, sean consolidados en una BD propietaria.

Como ya digo esto redunda en:

- Datos más seguros.
- Datos más estructurados
- Acceso a datos más rápido.
- Explotación de datos más fácil y eficiente.

Para ello, se ha diseñado la solución para que sea compatible con motores de BD, SQL server y MySQL.

Solo bastaría indicar en el fichero de configuración "web.config" los datos de conexión y a que base de datos nos enfrentamos.

El Modelo de datos es muy sencillo, únicamente tiene una tabla.

LOG	
PK	**Identificador**
	Fecha **hora** Zona_horaria **IP** Method URL Recurso TipoRecurso Sesion Parametros Protocolo VersionProtocolo CodRespuesta Referer Explorador Tamano Otros

Eso sí, dada la cantidad ingente de información que almacenará hay que cuidar mucho donde, como y de qué forma se definen los índices. La descripción de los campos será la siguiente:

Nombre Columna	Tipo	Oblig.	Función
Identificador	Bigint	SI	Clave primaria, será un numero entero correlativo para identificar unívocamente un registro
Fecha	Date	SI	Fecha en que se hizo el registro en el LOG
Hora	String	SI	Hora en que se hizo el registro
Zona_Horaria	String	SI	Diferencia horaria según GMT
IP	String(15)	SI	Direccion IP del cliente que navegó por la UOC
Method	Char(10)	SI	Metodo de acceso GET/POST
URL	Char(500)	SI	Direccion completa del recuso al que accede
Recurso	Char(250)	SI	Recurso al que se accede. URL sin parámetros.
TipoRecurso	Char(10)	SI	Tipo de recurso al que se accede. JPG, GIF, HTM,..
Sesion	Char(500)	SI	ID de sesión que se le asigna a la navegación del usuario

Parametros	Char(500)	SI	Serializacion de los parámetros del Querystring de la URL
Protocolo	Char(10)	SI	Protocolo de la navegación. Normalmente HTTP
VersionProtocolo	Char(10)	SI	1.1 normalmente
CodRespuesta	Char(10)	SI	Codigo que devolverá el servidor WEB. Explicado en 2.4
Referer	Char(500)	SI	URL desde la que se accede al recurso solicitado.
Explorador	Char(250)	SI	Explorador web con el que se navega.
Tamano	Int	SI	Tamano en bytes del recurso solicitado
Otros	Char(500)	NO	Campo donde caben los campos no identificados

En definitiva, se trata de la consolidación en BD de la información de una forma estructurada y fácilmente explotable del fichero LOG. Habrá una fila por cada línea del fichero de texto LOG con toda la información que ofrece

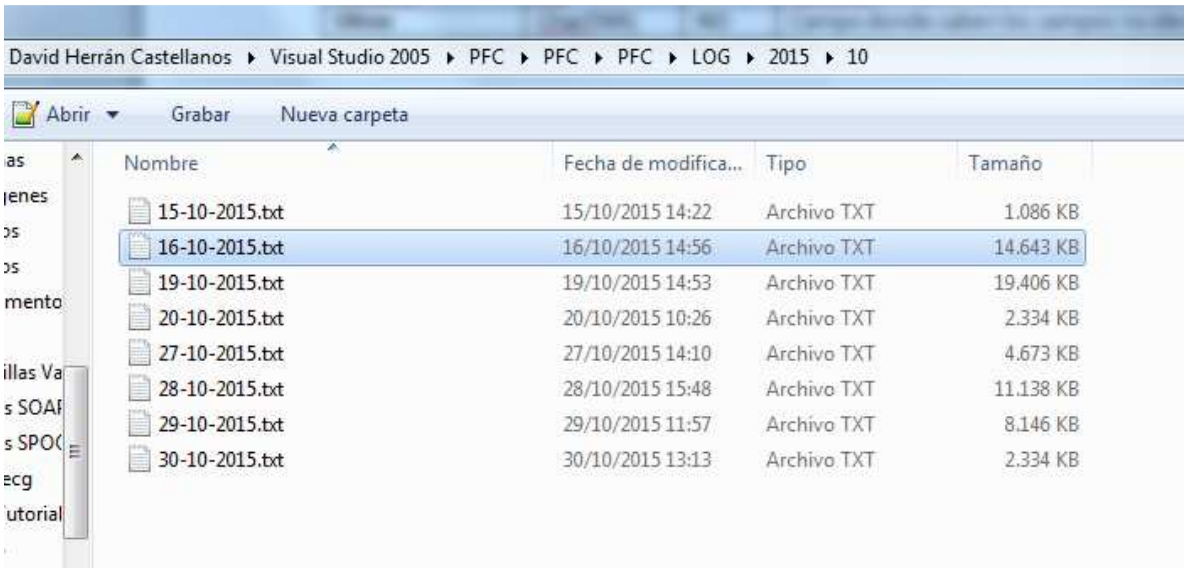
7.2.5.3. Nivel de sistema de ficheros-Logs de sistema

Como ya explicábamos en puntos anteriores (pto 2.4.7.) una de las mejoras que se añaden al Proyecto es que se ha desarrollado un módulo completo de LOG que permita registrar fácilmente y tener una auditoria clara de las acciones que realiza, tanto el usuario, como el propio programa en su ejecución. Esto tiene dos fines:

1. Auditar al usuario y tener una trazabilidad de las acciones que realiza.
2. Depurar el programa en tiempo de ejecución.

Lo primero que hará el programa al iniciarse es cargar del fichero de parámetros "web.config" el parámetro que indica el nivel de LOG en que nos encontramos y al pasar por un CheckPoint si se determina la necesidad de grabar acción, lo hará en una ruta donde se encuentra el programa, en la carpeta LOG dentro de un fichero con el día actual y extensión txt.

Así por ejemplo, si el programa se encuentra en la ruta : "C:\UOC\PFC" y queremos grabar un LOG el 6 de noviembre del 2015. El registro con la acción del usuario se grabará en "C:\UOC\PFC\06-11-2015.txt"



Dentro del fichero de LOG, cada línea del mismo corresponde a un apunte o acción con los siguientes campos separados por pipes.

<<Fechahora>> |<<Modulo>>| <<Metodo>>| <<Accion>> | <<Resultado>>

Dónde:

- *Fechahora*: <<Fecha Hora>> en que se realiza el registro del log. Tomará la fecha y hora de la fecha del sistema del Servidor de datos.
- *Modulo*: Objeto que realiza el registro del LOG.
- *Método*. Operación del objeto que hace el registro de LOG.
- *Acción*: Descripción a modo de texto libre de la operación que se está realizando. En la medida de lo posible si incorporarán valores de variables clave.
- *Resultado*. De la operación, 0-> Correcto, Valor Negativo-> Error de algún tipo.

Se muestra un extracto abajo.

```
16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 2 OK | 0 |  |
16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 3 OK | 0 |  |
16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 4 OK | 0 |  |
16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 5 OK | 0 |  |
16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 6 OK | 0 |  |
16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 7 OK | 0 |  |
16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 8 OK | 0 |  |
16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 9 OK | 0 |  |
16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 8 lineas | 0 |  |
16/10/2015 10:24:30 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 5 lineas | 0 |  |
16/10/2015 10:26:27 | [IMPORT] |  | [PreProcesaFichero] |  Se ha procesado 1 lineas y se han conservado 1
lineas | 0 |  |
16/10/2015 10:26:27 | [PRINCIPAL] |  | [FicheroInt1_RecursosNavegacion] | -ERROR INESPERADO-Índice fuera de
los límites de la matriz.-RECIBIDO:  | -1 |  |
16/10/2015 10:28:26 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 6 lineas | 0 |  |
16/10/2015 10:28:58 | [IMPORT] |  | [PreProcesaFichero] |  Se ha procesado 1 lineas y se han conservado 1
lineas | 0 |  |
16/10/2015 10:28:58 | [PRINCIPAL] |  | [FicheroInt1_RecursosNavegacion] | -ERROR INESPERADO-Índice fuera de
los límites de la matriz.-RECIBIDO:  | -1 |  |
16/10/2015 10:31:19 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 7 lineas | 0 |  |
```

16/10/2015 10:31:38 | [IMPORT] | | [PreProcesaFichero] | Se ha procesado 1 lineas y se han conservado 1 lineas | 0 | |
16/10/2015 10:31:38 | [PRINCIPAL] | | [FicheroInt1_RecursosNavegacion] | -ERROR INESPERADO-Índice fuera de los límites de la matriz.-RECIBIDO: | -1 | |
16/10/2015 10:46:51 | [IMPORT] | | [LeerFichero] | Se ha procesado 7 lineas | 0 | |
16/10/2015 10:48:01 | [PRINCIPAL] | | [FicheroInt1_RecursosNavegacion] | -ERROR INESPERADO-Referencia a objeto no establecida como instancia de un objeto.-RECIBIDO: | -1 | |
16/10/2015 10:48:57 | [IMPORT] | | [LeerFichero] | Se ha procesado 7 lineas | 0 | |
16/10/2015 10:49:40 | [PRINCIPAL] | | [GeneraFicheroIntermedio] | -ERROR INESPERADO-Referencia a objeto no establecida como instancia de un objeto.-RECIBIDO: Opcion=1 | -1 | |
16/10/2015 10:51:46 | [IMPORT] | | [LeerFichero] | Se ha procesado 6 lineas | 0 | |
16/10/2015 10:52:07 | [PRINCIPAL] | | [GeneraFicheroIntermedio] | -ERROR INESPERADO-Referencia a objeto no establecida como instancia de un objeto.-RECIBIDO: Opcion=1 | -1 | |
16/10/2015 10:52:30 | [IMPORT] | | [LeerFichero] | Se ha procesado 5 lineas | 0 | |
16/10/2015 10:53:00 | [PRINCIPAL] | | [PreProcesaFichero] | -ERROR INESPERADO-El valor no puede ser nulo. Nombre del parámetro: El argumento 'Array' tiene un valor Nothing.-RECIBIDO: | -1 | |
16/10/2015 10:53:00 | [PRINCIPAL] | | [GeneraFicheroIntermedio] | - ERROR_en el preprocesado del fichero -RECIBIDO Opcion=1 | -3 | |
16/10/2015 10:53:43 | [PRINCIPAL] | | [PreProcesaFichero] | -ERROR INESPERADO-El valor no puede ser nulo. Nombre del parámetro: El argumento 'Array' tiene un valor Nothing.-RECIBIDO: | -1 | |
16/10/2015 10:53:43 | [PRINCIPAL] | | [GeneraFicheroIntermedio] | - ERROR_en el preprocesado del fichero -RECIBIDO Opcion=1 | -3 | |
16/10/2015 10:55:22 | [IMPORT] | | [LeerFichero] | Se ha procesado 6 lineas | 0 | |
16/10/2015 10:55:46 | [PRINCIPAL] | | [PreProcesaFichero] | -ERROR INESPERADO-El valor no puede ser nulo. Nombre del parámetro: El argumento 'Array' tiene un valor Nothing.-RECIBIDO: | -1 | |
16/10/2015 10:55:46 | [PRINCIPAL] | | [GeneraFicheroIntermedio] | - ERROR_en el preprocesado del fichero -RECIBIDO Opcion=1 | -3 | |
16/10/2015 10:56:39 | [IMPORT] | | [AbreArchivoLOG] | -ERROR INESPERADO-El valor no está dentro del intervalo esperado.-RECIBIDO: Ruta= | 0 | |
16/10/2015 10:56:39 | [IMPORT] | | [LeerFicheroCompleto] | - ERROR_Fichero NO cargado previamente-RECIBIDO Ruta=,GuardaEnBD=False | -2 | |
16/10/2015 10:56:50 | [IMPORT] | | [AbreArchivoLOG] | -ERROR INESPERADO-El valor no está dentro del intervalo esperado.-RECIBIDO: Ruta= | 0 | |
16/10/2015 10:56:58 | [IMPORT] | | [LeerFicheroCompleto] | - ERROR_Fichero NO cargado previamente-RECIBIDO Ruta=,GuardaEnBD=False | -2 | |
16/10/2015 11:22:29 | [IMPORT] | | [LeerFichero] | - Se ha PROCESADO la linea 1 OK | 0 | |
16/10/2015 11:22:32 | [IMPORT] | | [LeerFichero] | - Se ha PROCESADO la linea 2 OK | 0 | |

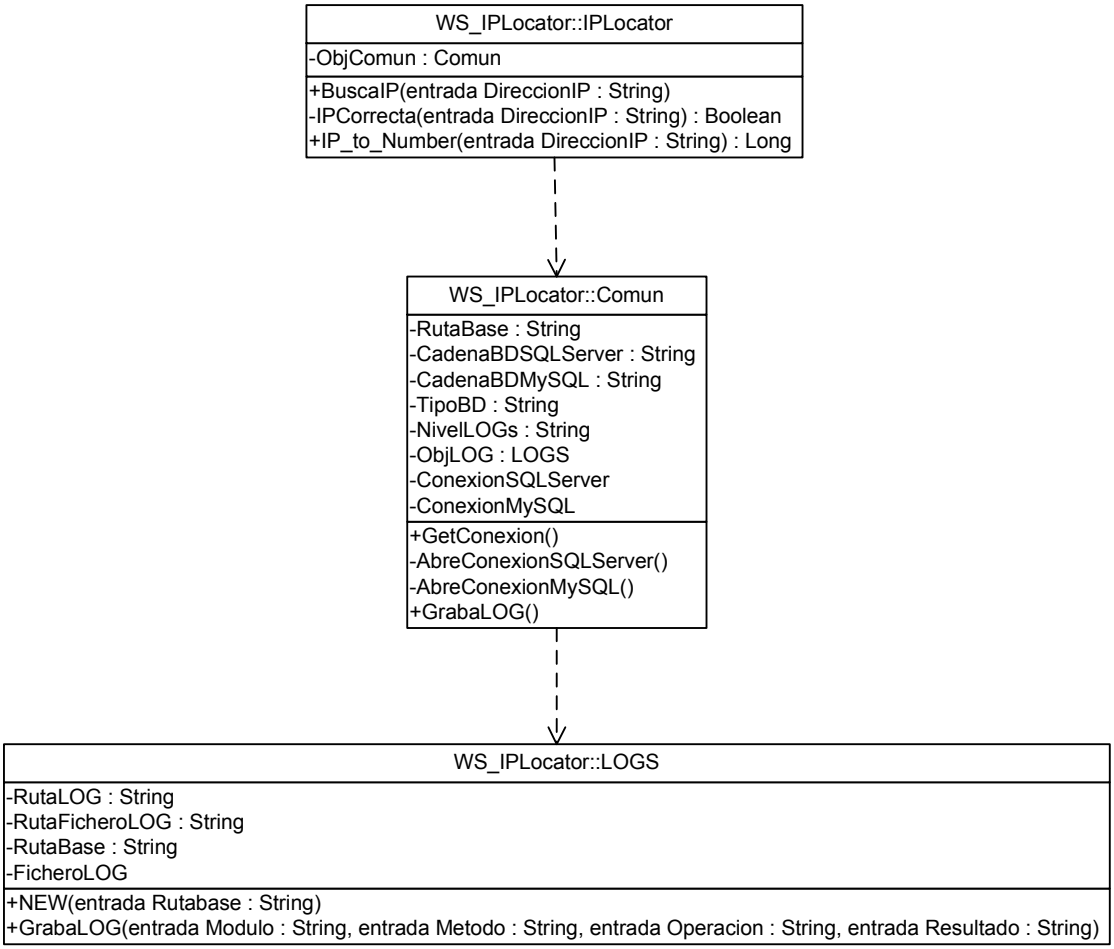
```
1  16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 2 OK | 0 |  |
2  16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 3 OK | 0 |  |
3  16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 4 OK | 0 |  |
4  16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 5 OK | 0 |  |
5  16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 6 OK | 0 |  |
6  16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 7 OK | 0 |  |
7  16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 8 OK | 0 |  |
8  16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha PROCESADO la linea 9 OK | 0 |  |
9  16/10/2015 8:20:22 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 8 lineas | 0 |  |
10 16/10/2015 10:24:30 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 5 lineas | 0 |  |
11 16/10/2015 10:26:27 | [IMPORT] |  | [PreProcesaFichero] |  Se ha procesado 1 lineas y se han conservado 1 lineas | 0 |  |
12 16/10/2015 10:26:27 | [PRINCIPAL] |  | [FicheroInt1_RecursosNavegacion] | -ERROR INESPERADO-Índice fuera de los límites de la matriz.-RECIBIDO:  | -1 |  |
13 16/10/2015 10:28:26 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 6 lineas | 0 |  |
14 16/10/2015 10:28:58 | [IMPORT] |  | [PreProcesaFichero] |  Se ha procesado 1 lineas y se han conservado 1 lineas | 0 |  |
15 16/10/2015 10:28:58 | [PRINCIPAL] |  | [FicheroInt1_RecursosNavegacion] | -ERROR INESPERADO-Índice fuera de los límites de la matriz.-RECIBIDO:  | -1 |  |
16 16/10/2015 10:31:19 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 7 lineas | 0 |  |
17 16/10/2015 10:31:38 | [IMPORT] |  | [PreProcesaFichero] |  Se ha procesado 1 lineas y se han conservado 1 lineas | 0 |  |
18 16/10/2015 10:31:38 | [PRINCIPAL] |  | [FicheroInt1_RecursosNavegacion] | -ERROR INESPERADO-Índice fuera de los límites de la matriz.-RECIBIDO:  | -1 |  |
19 16/10/2015 10:46:51 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 7 lineas | 0 |  |
20 16/10/2015 10:48:01 | [PRINCIPAL] |  | [FicheroInt1_RecursosNavegacion] | -ERROR INESPERADO-Referencia a objeto no establecida como instancia de un objeto.-RECIB
21 16/10/2015 10:48:57 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 7 lineas | 0 |  |
22 16/10/2015 10:49:40 | [PRINCIPAL] |  | [GeneraFicheroIntermedio] | -ERROR INESPERADO-Referencia a objeto no establecida como instancia de un objeto.-RECIBIDO:  Op
23 16/10/2015 10:51:46 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 6 lineas | 0 |  |
24 16/10/2015 10:52:07 | [PRINCIPAL] |  | [GeneraFicheroIntermedio] | -ERROR INESPERADO-Referencia a objeto no establecida como instancia de un objeto.-RECIBIDO:  Op
25 16/10/2015 10:52:30 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 5 lineas | 0 |  |
26 16/10/2015 10:53:00 | [PRINCIPAL] |  | [PreProcesaFichero] | -ERROR INESPERADO-El valor no puede ser nulo.
27 Nombre del parámetro: El argumento 'Array' tiene un valor Nothing.-RECIBIDO:  | -1 |  |
28 16/10/2015 10:53:00 | [PRINCIPAL] |  | [GeneraFicheroIntermedio] | - ERROR_en el preprocesado del fichero -RECIBIDO Opcion=1 | -3 |  |
29 16/10/2015 10:53:43 | [PRINCIPAL] |  | [PreProcesaFichero] | -ERROR INESPERADO-El valor no puede ser nulo.
30 Nombre del parámetro: El argumento 'Array' tiene un valor Nothing.-RECIBIDO:  | -1 |  |
31 16/10/2015 10:53:43 | [PRINCIPAL] |  | [GeneraFicheroIntermedio] | - ERROR_en el preprocesado del fichero -RECIBIDO Opcion=1 | -3 |  |
32 16/10/2015 10:55:22 | [IMPORT] |  | [LeerFichero] |  Se ha procesado 6 lineas | 0 |  |
33 16/10/2015 10:55:46 | [PRINCIPAL] |  | [PreProcesaFichero] | -ERROR INESPERADO-El valor no puede ser nulo.
34 Nombre del parámetro: El argumento 'Array' tiene un valor Nothing.-RECIBIDO:  | -1 |  |
35 16/10/2015 10:55:46 | [PRINCIPAL] |  | [GeneraFicheroIntermedio] | - ERROR_en el preprocesado del fichero -RECIBIDO Opcion=1 | -3 |  |
36 16/10/2015 10:56:39 | [IMPORT] |  | [AbreArchivoLOG] | -ERROR INESPERADO-El valor no está dentro del intervalo esperado.-RECIBIDO: Ruta= | 0 |  |
37 16/10/2015 10:56:39 | [IMPORT] |  | [LeerFicheroCompleto] | - ERROR_Fichero NO cargado previamente-RECIBIDO Ruta=,GuardaEnBD=False | -2 |  |
38 16/10/2015 10:56:50 | [IMPORT] |  | [AbreArchivoLOG] | -ERROR INESPERADO-El valor no está dentro del intervalo esperado.-RECIBIDO: Ruta= | 0 |  |
39 16/10/2015 10:56:58 | [IMPORT] |  | [LeerFicheroCompleto] | - ERROR_Fichero NO cargado previamente-RECIBIDO Ruta=,GuardaEnBD=False | -2 |  |
40 16/10/2015 11:22:29 | [IMPORT] |  | [LeerFichero] | - Se ha PROCESADO la linea 1 OK | 0 |  |
```

```
<configuration>

  <configSections>
    <sectionGroup name="applicationSettings" type="System.Configurat
      <section name="PFC.My.MySettings" type="System.Configuration.C
    </sectionGroup>
  </configSections>
  <appSettings>
    <add key="TipoBD" value="MYSQL"/>
    <add key="NivelLOGS" value="1"/>
    <!--
      0 ->  SIN LOG
      1 ->  LOG POR DEFECTO
      2 ->  LOG DEPURACION
    -->
    <add key="RutaTemporal" value="C:\Temp\"/>
  </appSettings>
  <connectionStrings>
    <add name="MySQLConnectionString" connectionString="Server=172.2
    <!--    <add name="SQLServerConnectionString" connectionString="
    <add name="SQLServerConnectionString" connectionString="Data Sou
  </connectionStrings>

    <system.web>
```

7.2.6. Diagrama de clases

7.2.6.1. WS_IPLocator

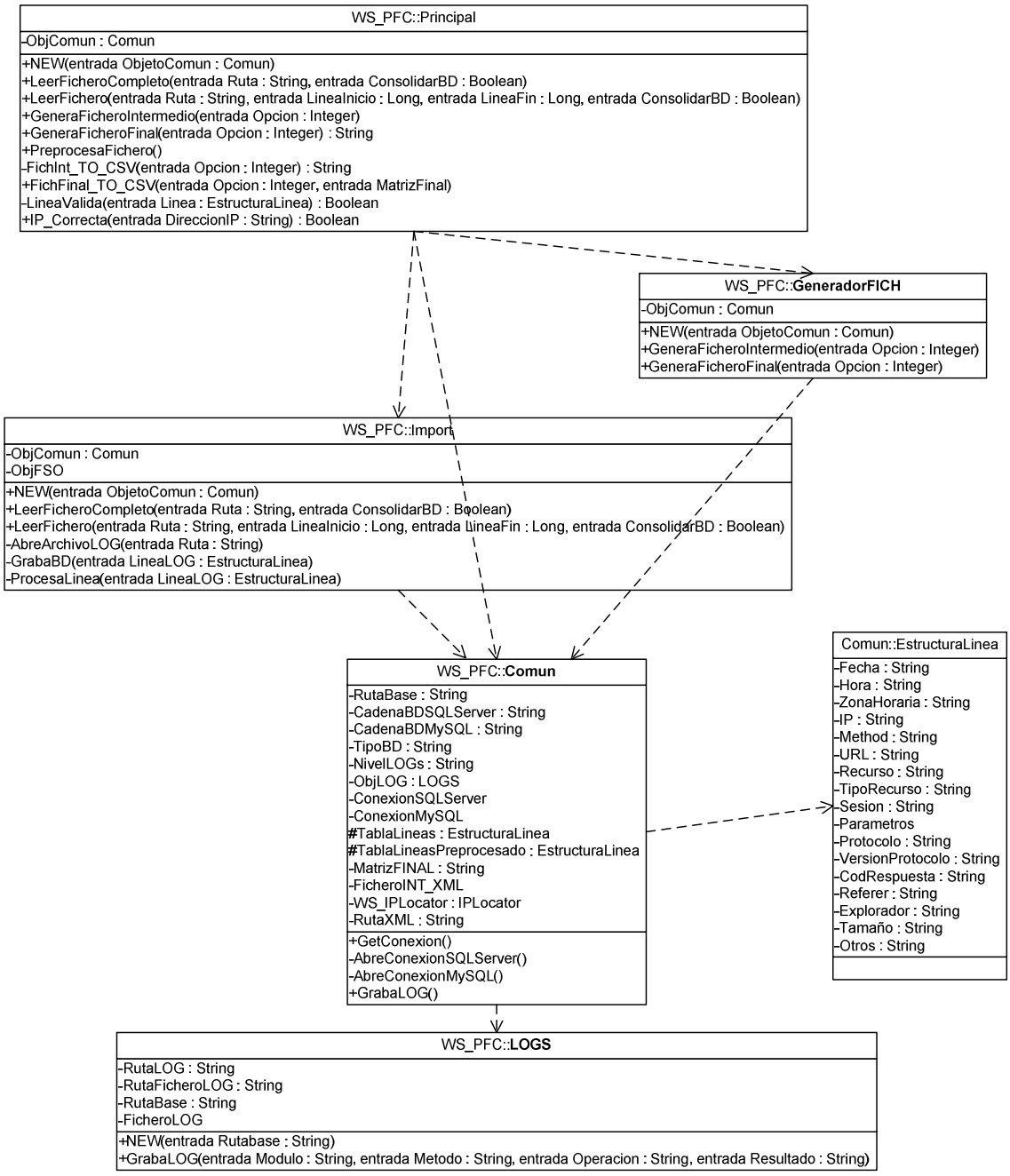
Se trata del Web Service ubicado en la máquina de negocio que suministrará la ubicación de una determinada IP



La descripción de cada clase será:

Paquete	WS_IPLocator
Clase	**Descripción**
IPLocator	Será la clase principal a la que se acceda desde el exterior. Tendrá un único método a modo de interfaz BuscaIP que a partir de una IP pasada como parámetro devolverá la ubicación
Comun	Será un objeto que se instanciará al inicio de la aplicación y cuya referencia será pasada de objeto a objeto y contiene todos los métodos, parámetros y atributos comunes a la aplicación y compartidos por todos
LOGS	Será instanciado y llamado desde el objeto Comun y su objetivo es grabar los Logs de sistema de la aplicación.

7.2.6.2. WS_PFC



La descripción de cada clase será:

Paquete	WS_PFC
Clase	**Descripción**
WS_PFC	Es la clase principal del WS que proporciona el interfaz de entrada y a través de la cual se irá llamado a unos u otros métodos de objetos. Instanciará el objeto común y lo distribuirá por cada objeto auxiliar Ademas, una vez generados los ficheros intermedios y finales, provee un método

	para exportarlos a CSV y XML por un lado y CSV en el otro.
GeneradorFICH	Deberemos tener cargado el fichero LOG de navegación del campus las estructuras de memoria diseñadas a tal fin guardadas en el Objeto Común. Este objeto genera los ficheros intermedios y ficheros finales a partir de la opcion seleccionada
Import	Este objeto permite importar los datos de los LOGS de navegación del Campus soportados en txt y los estrctura en memoria.
Común	Será un objeto que se instanciará al inicio de la aplicación y cuya referencia será pasada de objeto a objeto y contiene todos los métodos, parámetros y atributos comunes a la aplicación y compartidos por todos
LOGS	Será instanciado y llamado desde el objeto Comun y su objetivo es grabar los Logs de sistema de la aplicación.

7.2.6.3. PFC_WEB

PFC_WEB::**HOME.aspx**
-WS_PFC : Principal -FicheroCargado : Boolean -FicheroIntermedioGenerado : Boolean -OpcionSeleccionada : Integer
+CargarFicheroCompleto(entrada Ruta : String, entrada ConsolidarBD : Boolean) +CargarFicheroPorLineas(entrada Ruta : String, entrada LineaInicio : Long, entrada LineaFin : Long, entrada ConsolidarBD : Boolean) +GenerarFicheroIntermedio(entrada Opcion : Integer) +GenerarFicheroFinal(entrada Opcion : Integer) : String

Paquete	**PFC_WEB**
Clase	**Descripción**
HOME.aspx	Es la capa de presentación del PFC, que se alojará en el servidor WEB y a la que acceda el usuario. Provee una página web con el interfaz gráfico y que llamará al WS_PFC a través de su interfaz (Leer fichero completo o por líneas y Generar fichero intermedio o final)

7.3. DISEÑO DEL CODIGO FUENTE

El código fuente del proyecto desde el punto de vista de su programación software se estructura por medio de 3 proyectos o soluciones:

1. WS_IPLocator. Web Service ubicado en la máquina de negocio
2. WS_PFC. Web Service ubicado en la máquina de negocio
3. PFC. Capa de presentación programada en ASP.NET

Cada una de estas soluciones la describiremos detalladamente en los puntos siguientes, pero antes de en esto, hay que decir que se ha programado usando:

- Visual Studio 2005 version 2005. Con .NET framework 3.5

- Dentro del paquete anterior se ha utilizado el lenguaje Visual Basic .Net

Las razones para hacerlo así son las siguientes:

1. Particularmente estoy más familiarizado con esta herramienta.
2. Considero que llegando al mismo resultado que otras del mercado (j2ee, java, php,…) el paquete .Net es el que más rendimiento coste/beneficio da, con otras herramientas el coste de instalación, configuración y aprendizaje de la misma es mucho mayor.
   Por ejemplo, en otras experiencias que he tenido, el esfuerzo de conseguir hacer funcionar el debugger en eclipse fue titánico. Y eso no es trabajo productivo.
3. La herramienta de "debugger" o depuración es muy buena en .NET
4. Aun cuando tengo conocimientos avanzados en vb.net, siempre aparecen nuevos retos y dudas. La curva de aprendizaje y resolución de problemas en .NET creo que es muchísimo menos dura que con J2ee

CAPA DE NEGOCIO

7.3.1. *WS_IPLocator*

Se trata de un Web Service desarrollado en Visual Basic.NET 2005 bajo .NET Framework 3.5. (Gracias a las facilidades que ofrece Visual Studio, solo seleccionando "Aplicación del Servicio Web", el entorno de programación ya cargará todas las librerías, el debugger y en general todo lo necesario para el desarrollo. Incluso un pequeño "Hello World" para se comience la programación a partir de ahí)

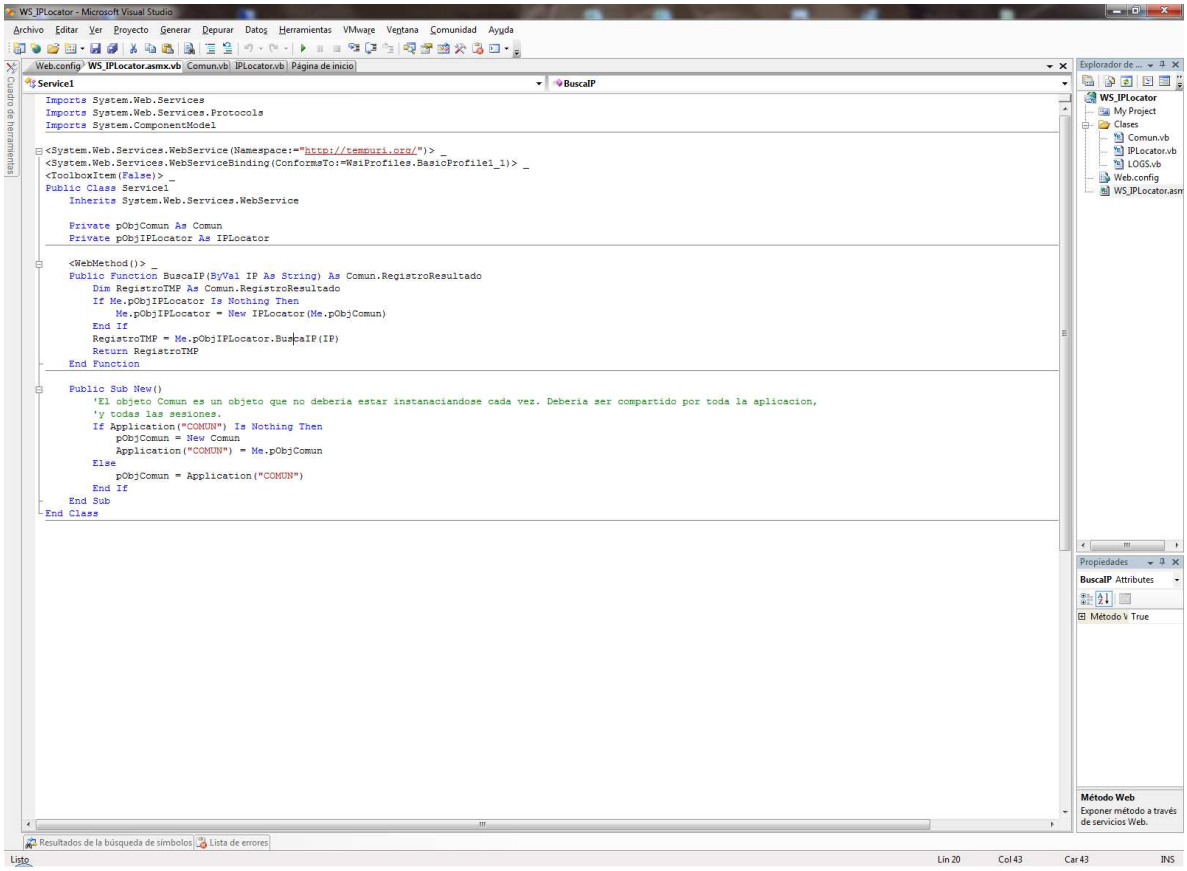
Su función será obtener los datos de la ubicación de una determinada IP como son:

- Ip_*Inicio*: Inicio del rango asignado a la zona geográfica a la cual pertenece la IP pasada como parámetro y que hay que ubicar
- Ip_*Fin*: Fin del rango asignado a la zona geográfica a la cual pertenece la IP pasada como parámetro y que hay que ubicar.
- *CodigoPais:* Código del Pais resultado de la búsqueda de la IP
- *Región:* Región dentro del país donde se ubica la IP
- *Ciudad:* Ciudad a la que pertenece la IP
- *Latitud:* Latitud a la que pertenece la IP
- *Longitud:* Longitud a la que pertenece la IP
- *CodigoZip:* CodigoZIP p código postal al que pertenece la IP
- *ZonaHoraria:* Zona horaria a la que pertenece la ubicación de la IP
- *Resultado:* Cadena de texto con el resultado de la operación de búsqueda.
  - Si todo ha ido correcto devolverá OK
  - Si ha habido error, devolverá descripción del error

A partir de una IP, consultando a una Base de datos (Esto es gracias a que cada provincia, zona, país tiene un rango de IPs asignado y solo se le asignan Ips de ese rango).

Enmarcado dentro del PFC, indicar que este WebService solo se utilizará en la Opcion 6 de todas las disponibles.

El código fuente se estructura de la siguiente forma:



- **WS_IPLocator.asmx.**
  Se trata del interfaz de entrada al aplicativo que tendrá dos propiedades privadas.

```
Private pObjComun As Comun
Private pObjIPLocator As IPLocator
```

- Objeto Comun. Se trata de un objeto que definiremos posteriormente. En él se incluye:
  1. Conexiones a bases de datos.
  2. Parámetros de aplicación que será usados por todos los módulos de la aplicación.
  3. Instancia del Objeto LOG. Que escribirá los LOGs de control y auditoria de la aplicación.

  Se instanciará al principio de toda la ejecución, en el constructor de WS_IPLocator y cuya referencia se irá pasando por todos los módulos de la solución. Igualmente, el objeto se serializará y se incluirá dentro de las variables de Aplicación del servidor.

Esto se hace para, dado que todo su contenido será usado por toda la aplicación, evitar tener que instanciar el objeto común, abrir conexiones,… en cada módulo al que acceda la ejecución.

- Objeto IPLocator. Será el objeto con la lógica de negocio.

Con respecto a los métodos, únicamente contará con un método público con el siguiente Interfaz:

- `Public Function BuscaIP(ByVal IP As String) As Comun.RegistroResultado`

En el que se pasará una IP y devolverá el resultado (en forma de registro o strcture que definiremos en el punto de abajo) después de atacar a la base de datos MySQL "ip2location".

- **Web.config**

  Es un fichero de configuración que guarda los parámetros de la aplicación en modo texto. El objeto común en su constructor los leerá y los cargará en memoria.

  Estos son:

Parámetro	Valores posibles	Valor Actual
TipoBD	<!-- MYSQL -> Motor de BD MySQL SQLSERVER -> Motor de BD SQL Server -->	MYSQL
NivelLOGS	<!-- 0 -> SIN LOG 1 -> LOG POR DEFECTO 2 -> LOG DEPURACION -->	1
RutaTemporal		C:\Temp\
ConnnectionStrings		
MySQLConnection String	Server=172.29.100.93;Port=3306;Database=ip2location; Uid=Conexion;Pwd=variable;	
SQLServerConnectionString	Data Source=172.29.100.93;Initial Catalog=IPLocator;Persist Security Info=False;MultipleActiveResultSets=False;User ID=BBDD\Administrador; Password = laracocina" providerName="System.Data.SqlClient	

- **Comun.vb**

  Objeto que almacenará todas las propiedades y métodos comunes a toda la solución.

  Podrá ser llamado desde cualquier punto del proyecto. Para hacer esto en todos los objetos/clases del proyecto habrá una propiedad `Private pObjComun As Comun` que se referenciará en el constructor del propio objeto

```
Public Sub New(ByVal ObjetoComun As Comun)
```

```
        pObjComun = ObjetoComun
    End Sub
```

Es decir, dicho de otra forma, se instanciará un objeto común que obtendrá los valores de los parámetros de web.config y que será común y compartido por todos los objetos de la solución

Tendrá las siguientes Propiedades Públicas:

- **Enumerado TipoDB**. Se trata un tipo de dato enumerado que guarda el tipo de base de datos en el que nos encontramos. El objeto común en su constructor leerá el parámetro de web.config y dará valor a una variable con el tipo de dato "TipoBD".

```
Public Enum TipoBD
    SqlServer = 0
    MySQL = 1
End Enum
```

- **NivelLOG**. Ídem del anterior, se trata de un tipo de dato que enumera los posibles valores de los niveles del LOG. El objeto común en su constructor leerá el parámetro de web.config y dará valor a una variable con el tipo de dato "NivelLOG".

```
Public Enum NivelLOG
    SinLogs = 0
    LogPorDefecto = 1
    LogDepuracion = 2
End Enum
```

- **CodigosError**. Ídem de los anteriores

```
Public Enum CodigosError
    Error_Inesperado = -1
    Exito = 0
    Warning = 1
End Enum
```

- **Resultado.** Estructura de datos definida en el RegistroResultado que devolverá en la salida las operaciones el resultado de las mismas.

```
Public Structure Resultado
    Dim HayError As Boolean
    Dim Codigo As CodigosError
    Dim Resultado As String
End Structure
```

- **RegistroResultado.** Será el resultado que devolverá el WS_IPLocator a través de su único método

```
Public Function BuscaIP(ByVal IP As String) As
Comun.RegistroResultado

    Public Structure RegistroResultado
        'ip_from, IP_to, country_code, country_name, region_name,
city_name, latitude, longitude, zip_code, time_zone
        Dim IP_Inicio As String
        Dim IP_Fin As String
        Dim CodigoPais As String
        Dim Pais As String
        Dim Region As String
        Dim Ciudad As String
        Dim Latitud As String
        Dim Longitud As String
```

```
        Dim CodigoZip As String
        Dim ZonaHoraria As String

        Dim Resultado As Resultado
    End Structure
```

Y las siguientes privadas:

```
    Private pRutaBase As String

    Private pCadenaBDMySQL As String
    Private pCadenaBDSQLServer As String
    Private pTipoBD As TipoBD
    Private pNivelLOGS As Integer
    Private pRutaTemporal As String

    Private pObjLOG As LOGS

    Private pConexionMySQL As
MySql.Data.MySqlClient.MySqlConnection
    Private pConexionSQLServer As
System.Data.SqlClient.SqlConnection
```

- **pRutaBase**. Guarda la Ruta completa dentro del Sistema de ficheros donde se encuentra implantada la aplicación. Es obtenida mediante la llamada `System.AppDomain.CurrentDomain.BaseDirectory` en el constructor.
- **pCadenaBDMySQL** y **pCadenaBDSQLSERVER**. Dos variables que guardan las cadenas de copnexion para las bases de datos MySQL y SqlServer respectivamente y que son obtenidas en el constructor del objeto de los parámetros de WebConfig `MySQLConnectionString` y `SQLServerConnectionString` respectivamente.
- **pNivelLOGS**. Variable que guarda el nivel de LOG en el que se encuentra la ejecución en este momento y será obteneido de un parámetro de WEB.config "`NivelLOGS`"
- **pRutaTemporal**. Variable que guarda la ruta temporal donde se descargrán los CSVs y XMLs que se vayan generando. Será un valor obtenido por el constructor del objeto del parámetro de Web.config `RutaTemporal`
- **pObjLOG**. Instancia del objeto LOG se instanciará en el constructor y que permite, a través de una variable de propiedad de solo lectura el acceso al objeto LOG y a poder escribir los logs de control y auditoria de la aplicación.

- **pConexionMySQL** y **pConexionSQLServer**. Variables de tipo conexión (de cada uno de sus motores) que guardan las conexiones a la BD de localización de ubicaciones de las IPs pasadas como parámetros.

```
Imports Microsoft.VisualBasic
Imports System.Data
Imports System
Imports System.Configuration

Public Class Comun
    'Parametros de la Aplicacion
    Private pRutaBase As String

    Private pCadenaBDMySQL As String
    Private pCadenaBDSQLServer As String
    Private pTipoBD As TipoBD
    Private pNivelLOGS As Integer
    Private pRutaTemporal As String

    Private pObjLOG As LOGS

    Private pConexionMySQL As MySql.Data.MySqlClient.MySqlConnection
    Private pConexionSQLServer As System.Data.SqlClient.SqlConnection

    Private pComando_SqlServer As SqlClient.SqlCommand
    Private pDataReader_SQLServer As SqlClient.SqlDataReader
    Private pComando_MySQL As MySql.Data.MySqlClient.MySqlCommand
    Private pDataReader_MySQL As MySql.Data.MySqlClient.MySqlDataReader

    Public Structure RegistroResultado
        'ip_from, IP_to, country_code, country_name, region_name, city_name, latitude, longi
        Dim IP_Inicio As String
        Dim IP_Fin As String
        Dim CodigoPais As String
        Dim Pais As String
        Dim Region As String
        Dim Ciudad As String
        Dim Latitud As String
        Dim Longitud As String
        Dim CodigoZip As String
        Dim ZonaHoraria As String

        Dim Resultado As Resultado
    End Structure

    Public Enum TipoBD ...
    Public Enum NivelLOG ...
    Public Structure Resultado ...
    Public Enum CodigosError ...

    Public ReadOnly Property GetConexion ...
    Public Function AbreConexionSQLServer ...
    Private Function AbreConexionMySQL ...
    Public Sub New ...
    Public Function GrabaLOG ...
    Public ReadOnly Property RutaTemporal ...
End Class
```

Además de las propiedades, el método tendrá una variable de propiedad de solo lectura por cada variable privada, para permitir el acceso del exterior del objeto al valor de las variables.

Con respecto a las operaciones del objeto, las principales serán tres:

- **Constructor**. Será el constructor del objeto y su misión, ademas de instanciar el propio objeto COMUN, será dar valores a las propiedades del objeto. Lo hará fundamentalmente de dos formas, leyendo los parámetros de web.config o de alguna otra forma alternativa como por

ejemplo para obtener la ruta base de la aplicación con `System.AppDomain.CurrentDomain.BaseDirectory`.

- **GetConexion**. Dependiendo del tipo de BD en la que nos encontremos, que indicará el parámetro de Web.Config "TIPOBD" devolverá a la salida de la función, la conexión de la base datos abierta, si no lo está el se ocupará de abrirla.
- **GrabaLOG**. Será un método que únicamente funcione como interfaz para acceder al método análogo del objeto privado LOG.

```
    Public Function GrabaLOG(ByVal NivelLog As Comun.NivelLOG, ByVal Modulo As
String, ByVal Metodo As String, ByVal Operacion As String, ByVal Resultado As
CodigosError, Optional ByVal Usuario As String = "", Optional ByVal IP As String
= "") As Integer
        If pNivelLOGS >= NivelLog Then
            pObjLOG.GrabaLOG(Modulo, Metodo, Operacion, Resultado, Usuario, IP)
        End If
    End Function
```

- **LOGS.vb**

  Objeto que implementa la mejora explicada en el punto 2.6.2.1 y que permite la grabación de los logs de sistema dentro de la ruta base de la aplicación.

  `[RutaBase]/LOG/[fechahoy].txt`

  Como datos reseñables de esta clase/objeto se puede destacar las siguientes propiedades/operaciones:

  - **FicheroLOG**. Propiedad privada que guarda la referencia al Objeto que se encargará de escribir en el fichero de texto perteneciente al LOG en curso según la estructura definida en el punto 2.6.2.1. Se instanciará en el constructor del objeto

    ```
    Private pFicheroLOG As System.IO.StreamWriter
    ```

  - **GrabaLOG**. Operación publica que escribe realmente en el fichero log una entrada de registro (previa apertura del mismo en formato "Append").

    Se pasa como parámetro Modulo, Método, Operación, Resultado, Usuario, cada uno con el objeto definido en el punto anterior.

    En el caso de Usuario, IP en este momento no se usa, pero se mantiene por si en un futuro se desea implementar un módulo de seguridad que permita autenticarse a un usuario y mantener un control de sesiones de navegación.

```
Public Function GrabaLOG(ByVal Modulo As String, ByVal Metodo As String, ByVal Operacion As
String, ByVal Resultado As Comun.CodigosError, Optional ByVal Usuario As String = "", Optional
ByVal IP As String = "") As Integer
```

- **IPLocator.vb**

  Se trata del objeto clave, el que realizará el trabajo de negocio a la hora de buscar la ubicación de una IP atacando a la base de datos, siendo el receptor de

todas las peticiones del interfaz público del WS (WS_IpLocator.asmx) a través de un único método público.

```
Public Function BuscaIP(ByVal IP As String) As Comun.RegistroResultado
```

El objeto únicamente tendrá una propiedad, privada, que será el propio objeto común.

```
Private pObjComun As Comun
```

Que se usará para acceder a todas las funcionalidades necesarias.

Con respecto a las operaciones, tendrá las siguientes:

- **Ip_to_number**. Método privado que convertirá una cadena con una IP con cuatro octetos del tipo "255.255.255.255" en un valor numérico. Esto es porque en la base de datos, las Ips vienen definidas como valores numéricos.

  ```
  Private Function IP_to_Number(ByVal ip As String) As Long
  ```

- **IpCorrecta**. Método Privado que devolverá true/false en funcion que si la IP pasada por parámetro es válida o no.

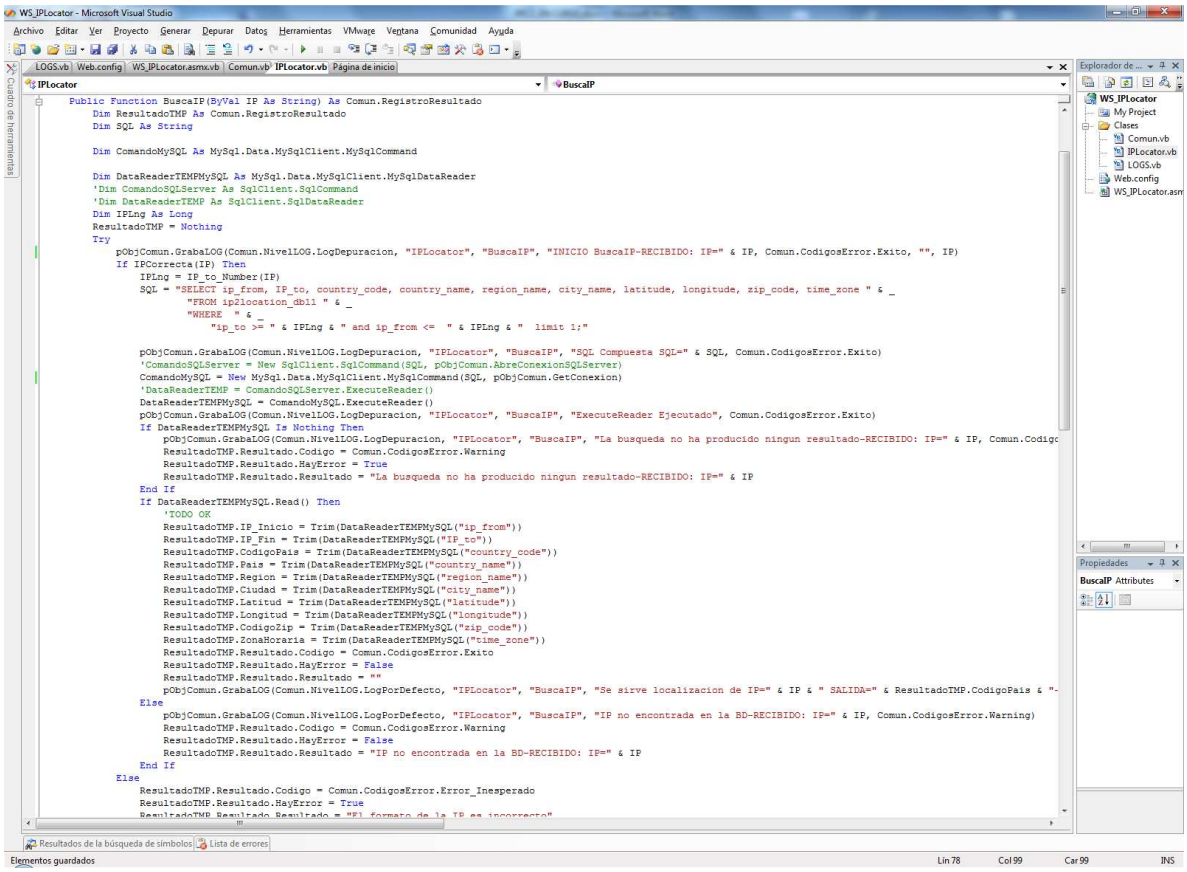
  ```
  Private Function IPCorrecta(ByVal IP As String) As Boolean
  ```

- **BuscaIP**. Será el único método público del objeto y el receptor de todas las peticiones realizadas al WebService.

  Tendrá la siguiente interfaz:

  ```
  Public Function BuscaIP(ByVal IP As String) As
  Comun.RegistroResultado
  ```

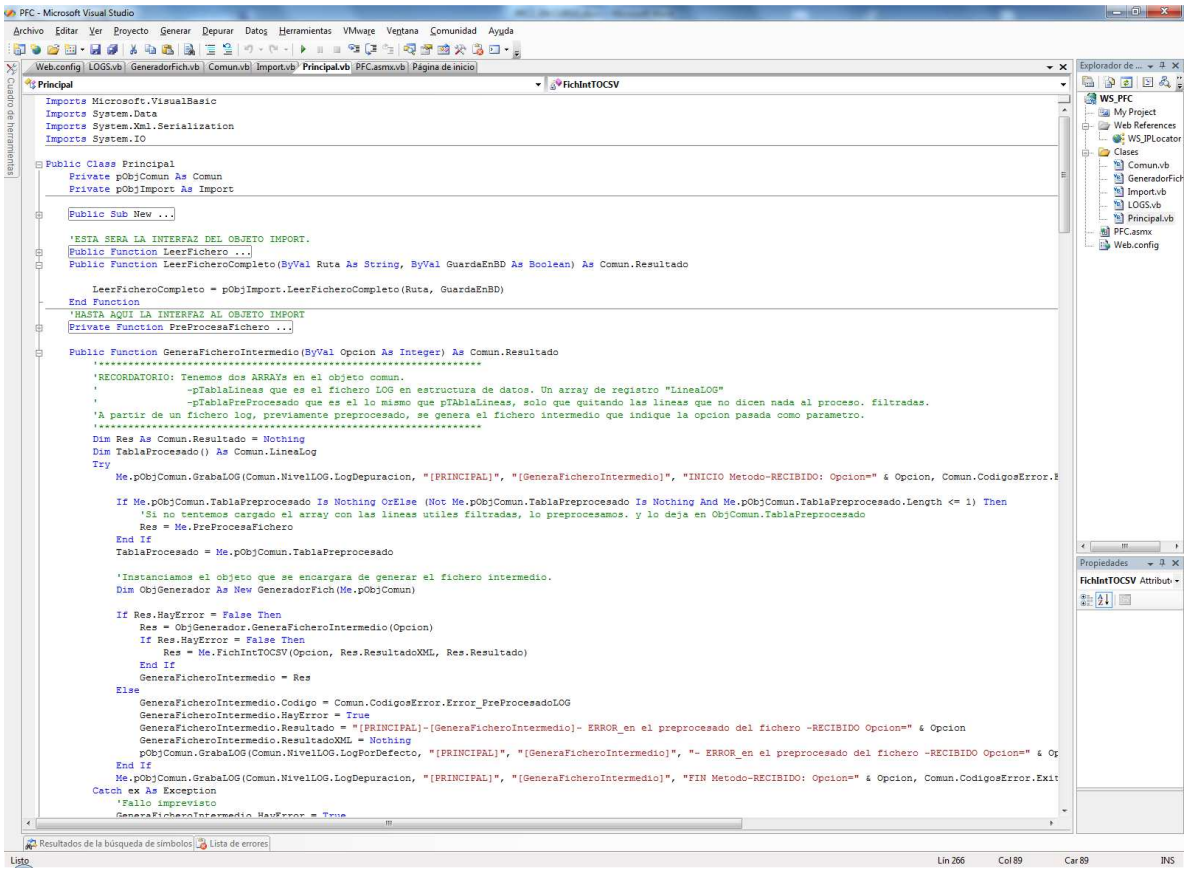
  A partir de la IP pasada como parámetro, validará si es correcta y consultará a la base de datos los datos relativos a la ubicación de la IP.



7.3.2. ***WS_PFC***

El objeto de este Web Service es recibir todas las peticiones de la capa de presentación (cualquiera que sea esta), darlas curso. A nivel genérico, aunque ahora las definiremos detalladamente, son:

1. La Lectura y procesado de los LOGS de navegación de la UOC.edu
2. Generación de ficheros intermedios en formato CSV y XML previa selección de una de las 6 opciones disponibles.
3. Generación de ficheros finales en formato CSV, previo elección de alguna de las 6 opciones programadas.



Esto lo realizará invocando a las operaciones que ahora describimos.

El aplicativo constará de los siguientes Módulos/Objetos.

- **Web.config.**
  Al igual que en el punto anterior, el Web.Config sirve para indicar todos los parámetros de la aplicación. En este caso serán:

Parámetro	Valores posibles	Valor Actual
TipoBD	<!-- MYSQL -> Motor de BD MySQL SQLSERVER -> Motor de BD SQL Server -->	MYSQL
NivelLOGS	<!-- 0 -> SIN LOG 1 -> LOG POR DEFECTO 2 -> LOG DEPURACION -->	1
RutaTemporal		C:\Temp\
ConnnectionStrings		
MySQLConnection String	Server=172.29.100.93;Port=3306;Database=ip2location;Uid=Conexion;Pwd=variable;	

SQLServerConnectionString	Data Source=172.29.100.93;Initial Catalog=IPLocator;Persist Security Info=False;MultipleActiveResultSets=False;User ID=BBDD\Administrador; Password = laracocina" providerName="System.Data.SqlClient
WebService IPLocator	
PFC_WS_IPLocator_Service1	<applicationSettings> <PFC.My.MySettings> <setting name="PFC_WS_IPLocator_Service1" serializeAs="String">\<value>http://172.29.100.93/WS_IPLocator/WS_IPLocator.asmx\</value> </setting> </PFC.My.MySettings> </applicationSettings>

En el caso del parámetro PFC_WS_IPLocator_Service1 (el ultimo identificado), se trata de la referencia al WebService IPLocator, antes definido, y necesario para poder invocarlo desde WS_PFC

- **LOGs.vb.**
  Tanto el uso, como la programación del objeto LOG (LOG.vb), será idéntica a la ya definida en el WS_IPLocator (pto. 3.5.1)

- **Comun.vb.**
  Este objeto Comun.vb tiene el mismo objetivo que el anterior del pto 3.5.1, únicamente que aquí esta circunscrito al ámbito de WS_PFC, en vez del WS_IPLocator.
  Objeto que almacenará todas las propiedades y métodos comunes a toda la solución.
  Podrá ser llamado desde cualquier punto del proyecto. Para hacer esto en todos los objetos/clases del proyecto habrá una propiedad `Private pObjComun As Comun` que se referenciará en el constructor del propio objeto

```
Public Sub New(ByVal ObjetoComun As Comun)
        pObjComun = ObjetoComun
End Sub
```

  Es decir, dicho de otra forma, se instanciará un objeto común que obtendrá los valores de los parámetros de web.config y que será común y compartido por todos los objetos de la solución

  Tendrá las siguientes Propiedades (por no repetir, se enumerarán las que difieran del Objeto COmun.vb del WS_IPLocator):
  - **Las ya enumeradas en Comun.vb del pto 3.5.1**.

- **Tipo de dato Comun.LineaLog.** Se trata de un registro de datos que agrupa toda la información relativa a una línea de log. Es la forma en que se estructurará la información del registro. Tendrá los siguientes campos:

Estructura (registro) LineaLOG		
Campo	**Tipo**	**Descripción**
Fecha	Cadena.	Fecha en que se produjo el registro delo LOG
Hora	Cadena	Hora en que se produjo el registro del LOG
Zonahoraria	Cadena.	Diferencia horaria respecto GMT
IP	Cadena	Direccion IP en formato de cuatro octetos
Method	Cadena.	Método de acceso al recurso. GET/POST
URL	Cadena	Direccion completa de acceso a la página, parámetros incluidos.
Recurso	Cadena.	Recuso al que accede la web. (quitando los parámetros del querystring)
TipoRecurso	Cadena	Tipo de recurso al que se accede. Extensión. Por ejemplo jpg, gif, html,…
Sesion	Cadena.	Identificador de la sesión de navegación
Parametros()	Cadena	Array o tabla de parámetros incluidos en el querystring de la llamada web
Protocolo	Cadena	Habitualmente HTTP
VersionProtocolo	Cadena.	Habitualmente 1.1
CodRespuesta	Cadena	Codigo de la respuesta devuelta por el servidor WEB. Enumerados en el pto 2.4.1.
Referer	Cadena	Direccion WEB completa del origen de la petición WEB
Explorador	Cadena.	Explorador web en donde se está navegando. IE, Firefox, Chrome,…
Tamaño	Cadena	Tamaño en bytes del recuso solicitado.
Otros		Cajon desastre de otros datos no importants.

```
Public Structure LineaLog
    Dim Fecha As String
    Dim Hora As String
    Dim ZonaHoraria As String    'Zona horaria
    Dim IP As String
    Dim Method As String         'GET/POST
    Dim URL As String            'URL pedida al servidor
    Dim Recurso As String
    Dim TipoRecurso As String        '.ico, .js, .txt,...
```

```
        Dim Sesion As String
        Dim Parametros() As String
        Dim Protocolo As String     'por ejemplo Http
        Dim VersionProtocolo As String   '1.1
        Dim CodRespuesta As Long'Cod devuelto por el servidor al
cliente.
        Dim Referer As String
        Dim Explorador As String    'Mozilla, Chrome, IE,...
        Dim Tamaño As Long          'Tamaño de la pagina
        Dim Otros As String
    End Structure
```

- **pTablaLineas**. Se trata de una tabla (Array) de LineaLog que almacenará las TODAS las líneas del fichero LOG, tendrá tantos elementos como líneas de Log haya.
- **pTablaPreprocesado**. Se trata de una tabla (Array) de elementos LineaLog que almacenará las líneas del fichero LOG después de haber limpiado las líneas no relevantes.
  El criterio para el filtrado de estas líneas está descrito en el punto 2.4.3
- **pMatrizFinal** (,). Será un array de dos dimensiones de elementos carácter con las estructuras descritas en los puntos 2.3.9 a 2.3.14.
- **pFicheroIntXML.** Se trata de la serialización del XML que contiene el fichero intermedio generado**.**

```
    'TABLAS QUE TENDRAN EN MEMORIA EL FICHERO LOG.
    '   pTablaLineas tendra el fichero completo
    '   pTablaPreProcesado tendra el fichero log con las filas basura
filtradas.
    Private pTablaLineas() As Comun.LineaLog 'Tabla de registros que
guarda en memoria el archivo log.
    Private pTablaPreProcesado() As Comun.LineaLog
    Friend MatrizFinal(,) As String

    Private pFicheroInt_XML As XmlSerializer
    Private pRutaXML As String

    Private pWS_IPLocator As WS_IPLocator.Service1
```

- **Tipo de dato Comun.Resultado.** Estructura de datos usada por la mayoría de métodos del WS_PFC que devolverá en la salida las operaciones el resultado de las mismas.

Estructura (registro) Resultado		
Campo	**Tipo**	**Descripción**
HayError	Boolean	Boolean True/false que indica si la operación ha resultado correcta o incorrecta

Codigo	CodigosError	Código devuelto por la operación
		`Public Enum CodigosError`
		`Error_FormatoCampo = -5`
		`Error_OpcionSeleccionadaNoValida = -4`

		`Error_PreProcesadoLOG = -3`
		`Error_Apertura_Fichero = -2`
		`Error_Inesperado = -1`
		`Exito = 0`
		`Warning = 1`
		`End Enum`

Resultado	Cadena.	Descripcion del resultado
ResultadoXML	XMLSerialzer	Donde se permita (FicheroIntermedio) se trata de la serialización del XML resultado. Para la comunicación entre operaciones(métodos) hay que pasar como parámetro los datos del fichero intermedio, en vez de usar rutas del filesystem y que lea de fichero, el .NET me permite pasarlo a XML y serialziar automáticamente la estructura de memoria. Para el caso del fichero Final, dado que es una matriz de dos dimensiones el .Net no soporta este tipo de serialziacion.

```
Public Structure Resultado
    Dim HayError As Boolean
    Dim Codigo As CodigosError
    Dim Resultado As String
    Dim ResultadoXML As XmlSerializer
End Structure
```

- **PFC.asmx.**
  Se trata del interfaz de entrada al Web Service según el WSDL definido en el punto 2.5.1.2, el cual consta de 4 métodos públicos (a los que se podrá acceder desde la capa de presentación)

Service1

Las operaciones siguientes son compatibles. Para una definición formal, revise la descripción de servicios.

- GeneraFicheroFinal
- GeneraFicheroIntermedio
- LeerFichero
- LeerFicheroCompleto

Este servicio Web utiliza http://tempuri.org/ como espacio de nombres predeterminado.

Recomendación: cambiar el espacio de nombres predeterminado antes de hacer público el servicio Web XML.

Cada servicio Web XML necesita un espacio de nombres único para que las aplicaciones de cliente puedan distinguir este servicio de otros servicio
en desarrollo, pero los servicios Web XML publicados deberían utilizar un espacio de nombres más permanente.

Debe identificar su servicio Web XML con un espacio de nombres que controle. Por ejemplo, puede utilizar el nombre de dominio de Internet de su
nombres de servicios Web XML parecen direcciones URL, éstos no pueden señalar a recursos reales en el Web. (Los espacios de nombres de los se

En los servicios Web XML que se crean con ASP.NET, se puede cambiar el espacio de nombres predeterminado utilizando la propiedad Namespace
contiene los métodos del servicio Web XML. A continuación se muestra un ejemplo de código que establece el espacio de nombres en "http://micr

C#

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public class MyWebService {
    // implementación
}
```

Visual Basic

```
<WebService(Namespace:="http://microsoft.com/webservices/")> Public Class MyWebService
    ' implementación
End Class
```

C++

```
[WebService(Namespace="http://microsoft.com/webservices/")]
public ref class MyWebService {
    // implementación
};
```

Para obtener más detalles acerca de espacios de nombres XML, vea la sugerencia W3C en Espacio de nombres en XML.

Para obtener más detalles acerca de WSDL, vea Especificación WSDL.

Para obtener más detalles sobre los URI, vea RFC 2396.

El objeto constará de dos propiedades privadas:

- **pObjComun**. Instancia privada del Objeto Común que tendrá el mismo objetivo que en el punto anterior. Se pasará de objeto a objeto para poder acceder a todas las características comunes del Aplicativo, conexiones a bases de datos, parámetros, etc..
- **pObjPrincipal**. Instancia del Objeto Principal, donde se invocará a cada una de las operaciones del WebService PFC.

```
Private pObjComun As Comun
Private pObjPrincipal As Principal
```

Y tantas operaciones públicas como tenga el interfaz público, que hemos definido arriba.

- **LeerFichero**. Se trata de un método que recibirá la petición del usuario, realizará las validaciones de los parámetros de entrada y la reenviará al Objeto Principal que es el que realizará las operaciones Leerá la información del fichero LOG de navegación de la UOC línea por línea y estructurará y grabará la información en estructuras de datos de

memoria del objeto común (presente y accesible durante toda la ejecución). Además realizará comprobaciones de los datos que lee y de la estructura del fichero LOG sea la convenida. (explicada pto 2.3.15)
Pasará como parámetro:

- Ruta: Ruta completa del fichero LOG.
- LineaInicio: Línea a partir de la cual se leerá el LOG
- LineaFinal: Línea hasta la que se leerá el LOG.
- GuardaEnBD: Booleano que indique si se guarda o no la información leída en los LOGS en BD.

Si "GuardaEnBD" es verdadero, ademas grabará la información en base de datos.
En la salida devolverá un registro de tipo Comun.Resultado

```
Public Function LeerFichero(ByVal Ruta As String, ByVal LineaInicio As Long,
ByVal LineaFin As Long, ByVal GuardaEnBD As Boolean) As Comun.Resultado
```

- **LeerFicheroCompleto**. Se trata de un método que recibirá la petición del usuario, realizará las validaciones de los parámetros de entrada y la reenviará al Objeto Principal que es el que realizará las operaciones. leerá el fichero completo. Es decir, leerá el fichero log, y grabará en estructuras de memoria del objeto común la información estructurada leída. Además podrá o no grabar la información en BD.

  IMPORTANTE: Según los ejemplos que manejamos y dado el volumen de información de que se trata (13 mill de filas), hay que tener mucho cuidado con esta opción, ya que su ejecución puede tardar mucho tiempo o incluso desbordar la maquina donde se ejecuta.

  En la salida devolverá un registro de tipo Comun.Resultado

```
Public Function LeerFicheroCompleto(ByVal Ruta As String, ByVal GuardaEnBD As
Boolean) As Comun.Resultado
```

- **GenerarFicheroIntermedio**. Recibirá las peticiones del usuario, validará los parámetros de entrada y reenviará la petición al objeto Principal que es el que realizará el trabajo definido en puntos 2.4.3 a 2.4.5.

```
Public Function GeneraFicheroIntermedio(ByVal Opcion As Integer) As
Comun.Resultado
```

- **GenerarFicheroFinal**. Definido en puntos 2.4.6. Idem de los puntos anteriores.

```
Public Function GeneraFicheroFinal(ByVal Opcion As Integer) As Comun.Resultado
```

- **Import.vb.**

  Se trata del módulo donde se programará toda la lógica que permite la importación de datos del fichero de texto LOG en texto plano.

  Como interfaz publica, solo tendrá dos operaciones LeerFichero y LeerFicheroCompleto (el resto de módulos que incorporan estos métodos son únicamente interfaces, aquí es donde se programará de echo). Ambos tienen el mismo resultado final.

  - Cargar en la estructura de memoria del objeto Común (tabla de registros LineasLog)
  - Grabar el fichero intermedio en la ruta convenida en formato CSV y XML.
  - Devolver a la salida la ruta donde se grabaron los dos ficheros anteriores
  - Devolver a la salida la serialización del fichero XML generado.

  Tambien se han programado una serie de operaciones privadas que enumeraremos.

  Las Propiedades del método serán las siguientes:

  - **pObjComun**. Presente en todos los objetos y tendrá la misma misión. Recibirá la referencia al objeto Comun que pasa de objeto en objeto.
  - **pObjFSO**. Objeto de sistema que hará la lectura del fichero de texto
  - **pArchivo**. Representa el archivo de texto LOG

```
Private pObjComun As Comun
Private pObjFSO As FileSystemObject
Private pArchivo As TextStream'Objeto del archivo de LOGs que se leera.
```

  Referente a los métodos u operaciones, tememos las ya citadas públicas LeerFichero y LeerFicheroCompleto.

  LeerFichero. Cuya misión ya hemos descrito, se le pasará como parámetro:

  - Ruta: Ruta completa del fichero LOG.
  - LineaInicio: Línea a partir de la cual se leerá el LOG
  - LineaFinal: Línea hasta la que se leerá el LOG.
  - GuardaEnBD: Booleano que indique si se guarda o no la información leída en los LOGS en BD.

  Si “GuardaEnBD” es verdadero, ademas grabará la información en base de datos.

  Abrirá el fichero y desde la línea de inicio hasta la línea de fin, irá línea por línea leyéndolas y pasando cada una, cada vez, a un método privado que se encargará de procesar la información de cada línea. En este caso “ProcesaLinea”, que devolverá un registro de la línea en formato estructurado, `Comun.LineaLog`, al recibirlo se incorporará a un array de LineaLog.

```
Public Function LeerFichero(ByVal Ruta As String, ByVal LineaInicio As Long, ByVal LineaFin As Long, ByVal GuardaEnBD As Boolean) As Comun.Resultado
```

- **LeerFicheroCompleto**. Ídem del anterior, con la única salvedad, que en este caso leerá el fichero completo, no desde un inicio hasta un fin, e igualmente lo pasará cada línea a "ProcesaLinea" y se incorporará a un array de LineaLog.

Hasta aquí sería la interfaz pública, cuyos objetivos los hemos descrito en la página anterior. Para conseguirlos un método clave es "ProcesaLinea"

- **ProcesaLinea**. Metodo privado del objeto IMPORT, que recibirá una cadena de texto que contiene la línea del Fichero LOG de navegacion del campus de la UOC y devolverá los datos estrcuturados en una estructura Comun.LineaLOG.
  Para ello se parte de un formato acordado y descrito en el punto 2.3.15 e irá extrayendo la información y añadiéndola a la estructura citada, que será la que devuelva a la salida.
  Este método quizá necesite algunos métodos privados menores

```
Private Function ProcesaLinea(ByVal Linea As String) As Comun.LineaLog
```

- **GeneradorFICH.vb.**
  Será el modulo que a partir de tener cargado en memoria los datos necesarios, generará el fichero intermedio o fichero final en cualquiera de las 6 opciones disponibles.
  Como interfaz público el objeto tendrá únicamente dos métodos "`GeneraFicheroIntermedio`" y "`GeneraFicheroFinal`"
  Se tendrá una única propiedad, una privada, que será el objeto común
  ```
  Private pObjComun As Comun
  ```
  Con misión análoga a la de los otros objetos.
  El objeto tendrá las siguientes operaciones públicas.
  - **GeneraFicheroIntermedio**. Tendrá como parámetro de entrada un número entero del 1 al 6 que representa las 6 opciones disponibles para exportar el fichero intermedio CSV descritas en los casos de uso 2.1 a 2.6.
    Además, como requerimiento "Req_2", deberá haberse cargado en memoria (Array de registros LineaLog) en formato estructurado el fichero LOG o un subconjunto de él. Además deberá haberse "Preprocesado" el fichero LOG (en su estructura de memoria), esta operación se realiza en el modulo Principal.vb
    Con estos dos requerimientos cumplidos, el método recibirá la opción de generación del fichero intermedio (1-6) e invocará al método privado correspondiente especializado en realizar la exportación y generación de la opción en concreto. Describiremos alguno de ellos.
    1. FicheroInt1. Se le pasará la tabla de registros con sus líneas inútiles ya filtradas, la recorrerá por completo

cumplimentando, en este caso, IP, Sesión y Recurso (que son los afectados de la opción 1) en un Registro habilitado para ese uso (RegOpcion1)

```
    Public Structure RegOpcion1
        Implements IComparable
        'Registro que contendrá la opcion 1 de las posibles para
estudiar. Los recursos por los que navega en una sesion.
        Dim IP As String
        Dim Recurso As String
        Dim Sesion As String
        Dim Indice As Long

        Public Function CompareTo(ByVal obj As Object) As Integer
Implements System.IComparable.CompareTo
            Return Me.Sesion.CompareTo(CType(obj,
RegOpcion1).Sesion)
        End Function
    End Structure
```

Después de haberlo hecho, ordenará la tabla por sesión para tener todos las navegaciones del usuario juntas entre si.

```
Private Function FicheroInt1(ByVal TablaPreprocesada() As Comun.LineaLog) As
Comun.Resultado
```

2. <u>FinchInt2</u>. Se trata del mismo proceso que el anterior (FichInt1), solo que en este caso, ya que estamos en la Opcion2, los datos que se va a estudiar son diferentes. En este caso serán IP, FechaHora, Sesion (Como se describe en el caso de uso 2.2).
   Cumplimentará los datos del Registro (RegOpcion2) y lo ordenará por sesión.

```
    Public Structure RegOpcion2
        Implements IComparable
        'Registro que contendrá la opcion 1 de las posibles para
estudiar. Los recursos por los que navega en una sesion.
        Dim IP As String
        Dim Fecha As String
        Dim Sesion As String
        Dim Indice As Long
        Public Function CompareTo(ByVal obj As Object) As
Integer Implements System.IComparable.CompareTo
            Return Me.Sesion.CompareTo(CType(obj,
RegOpcion2).Sesion)
        End Function
    End Structure
```

```
Public Function GeneraFicheroIntermedio(ByVal Opcion As Integer)
As Comun.Resultado
```

- **GeneraFicheroFinal**. Se meterá como parámetro una de las 6 opciones disponibles y cumplido el Req_3 y Req_4, generará un fichero CSV con el formato indicado en los Ptos 2.3.9 a 2.3.14 (dependiendo de la opción seleccionada).

A partir de la tabla de memoria con el fichero intermedio (Array de registros RegOpcion1 para la opcion1) y dependiendo de la opción seleccionada, invocará al método privado especializado en ello.

1. FicheroFINAL_1. Es el método privado que se encarga, a partir de la tabla de RegOpcion1, generar una Matriz de dos dimensiones de caracteres que mantiene en memoria el formato del fichero FINAL.

   Esta matriz tendrá tantas filas como Sesiones diferentes haya y tantas columnas como Recursos diferentes (por supuesto para el caso de opción 1).

   El proceso será el siguiente:

   - Se iterará la tabla intermedia comprobando el número de filas y columnas que habrá que tener. Como hemos dicho antes, serán sesiones y recursos diferentes.

     Hecho esto, ya tenemos las dimensiones de la matriz de caracteres.
   - Ahora habrá que recorrer la matriz de caracteres comprobando si la IP de la sesión i ha navegado por el recurso j. En caso afirmativo "T", si no "F".

Despues de esto se devolvería a Principal.vb que solo tendría que exportarlo a CSV.

```
Private Function FicheroFINAL_1() As Comun.Resultado
```

2. FicheroFINAL_2. El proceso es idéntico al anterior solo que los campos afectados en este caso son IP, Sesion y FechaHora. El resto es análogo.

```
Private Function FicheroFINAL_2() As Comun.Resultado
```

```
Public Function GeneraFicheroFinal(ByVal Opcion As Integer) As
Comun.Resultado
```

- **Principal.vb.**

  Será el modulo que reciba todas las peticiones de trabajo del interfaz de usuario del WS y se encargará de gestionarlas invocando a otros modulos u objetos.

  Como propiedades privadas tendrá las instancias a los objetos Común e Import, que evidentemente son necesarias para poder tener acceso a su interfaz pública.

```
Private pObjComun As Comun
Private pObjImport As Import
```

Sus operaciones públicas, en general, únicamente serán interfaces que reciben la petición y la reenvían al objeto especializado en hacer el trabajo:

- **LeerFichero**. Invocará a la operación del mismo nombre del objeto IMPORT.

- **LeerFicheroCompleto**. Invocará a la operación del mismo nombre del objeto IMPORT

- **GeneraFicheroIntermedio**. Principalmente hará 3 operaciones:

  1. "Preprocesar" el array de memoria con el fichero LOG. Esto significa que deberá filtrar las líneas inútiles que aparecen en el LOG según los criterios descritos en el pto 2.4.3. El resultado será un array del tipo
     ```
     Private pTablaPreProcesado() As Comun.LineaLog
     ```
     Que está en el objeto Comun (para que esté disponible a todo el proyecto).
  2. Invocar al método del objeto GeneradorFich que es el que realizará el trabajo.
     ```
     Res = ObjGenerador.GeneraFicheroIntermedio(Opcion)
     ```
  3. La operación anterior devolverá el resultado en XML serializado que será el que pasemos al método privado "`FichIntTOCSV`" que exportará el resultado a CSV y XML.

- **GeneraFicheroFinal**. Invocará a la operación del mismo nombre del objeto especializado en la generación de los ficheros intermedio y final, "GeneradorFich".

  ```
  'Instanciamos el objeto que se encargará de generar el fichero
  FINAL e invocamos el metodo.
  ObjGeneradorFich = New GeneradorFich(Me.pObjComun)
  Res = ObjGeneradorFich.GeneraFicheroFinal(Opcion)
  ```

  Después de haberlo realizado se invocará al método privado que exportará la matriz de dos dimensiones de caracteres a CSV. "`FichFINAL_TOCSV`".

Las operaciones privadas de este objeto serán las ya comentadas brevemente.

- **PreprocesaFichero**. No todas las líneas del fichero LOG serán útiles para nuestro estudio. Este método recorrerá todas ellas y si es válida de acuerdo a los criterios del pto 2.4.3 la llevará a otro Array "`pTablaPreProcesado`" con la siguiente definición.

  ```
  Private pTablaPreProcesado() As Comun.LineaLog
  ```

  Los criterios se definen en el método Privado "LineaValida"

```
Private Function LineaValida(ByVal Linea As Comun.LineaLog) As
Boolean
```

- **IPCorrecta**. Es una utilidad privada del objeto que devuelve un True si la IP pasada como parámetro es correcta en cuanto a su formato. Devolverá false en caso contrario.

- **FichIntToCSV**. Método privado del objeto principal que exportará a CSV y XML, en la ruta indicada en Web.config, el contenido de la tabla de memoria que tiene el fichero LOG intermedio.

  Lo primero que hará es deserializar el XML pasado como parámetro en una tabla e iterar por ella escribiendo en el fichero CSV los datos correspondientes.

  En cuanto al XML la exportación se hace de forma inmediata con la instrucción.

```
Dim x As XmlSerializer = New
XmlSerializer(GetType(Comun.RegOpcion1()))
Dim writer As TextWriter = New StreamWriter(RutaSalida)
x.Serialize(writer, TablaFich_Int1)
```

```
Private Function FichIntTOCSV(ByVal Opcion As Integer, ByVal Fichero As XmlSerializer,
ByVal RutaXML As String) As Comun.Resultado.
```

- **FichFinalToCSV**. A partir de la matriz de memoria de dos dimensiones de caracteres, exportará a la ruta indicada en “web.config” el fichero final en formato CSV. Lo hará de acuerdo a la opcion seleccionada y al formato descrito en los Ptos 2.3.9 a 2.3.14.

CAPA DE PRESENTACION

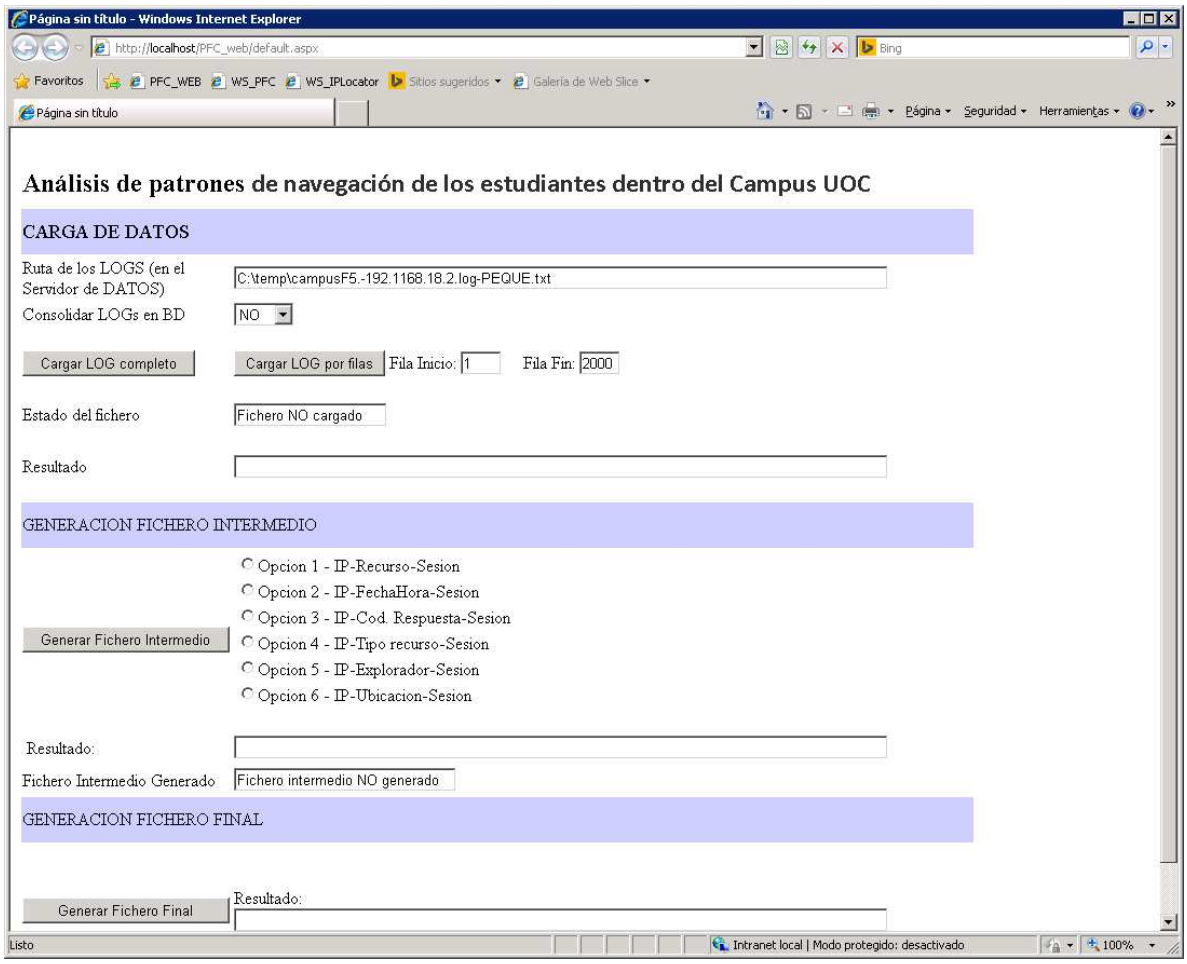
7.3.3. *PFC (Capa de presentación)*

Se va a tratar de una página web sencilla que permita acceder a la interfaz pública del WS_PFC.

1. **Leer Fichero Completo**, indicándole:
    a. la ruta completa donde está fichero LOG.
    b. Si/NO para consolidar fichero LOG en Base de datos.

2. **Leer Fichero** indicándole:
    a. Línea de inicio de lectura
    b. Línea de fin de lectura
    c. Ruta completa donde está el fichero LOG.
    d. Si/NO para consolidar fichero LOG en Base de datos.
3. **Generar Fichero Intermedio**
    a. Indicando la opción (1-6) seleccionada.
4. **Generar Fichero Final** (en este caso la opción ya está seleccionada del fichero intermedio).

Se desarrollará en ASP.Net y se desplegará en el Sitio WEB del Servidor WEB con la utilidad de "publicación de Sitios web" que ofrece el entorno de programación Visual Studio 2005.



8. Plan de pruebas

8.1. DEFINICION DE PLAN DE PRUEBAS

Se van a definir dos tipos de pruebas

- Pruebas de explotación de la plataforma con datos reales.
- Estudio Experimental

8.2. PRUEBAS DE EXPLOTACIÓN DE LA PLATAFORMA CON DATOS REALES

El input más importante del Proyecto, será evidentemente, el fichero LOG con las navegaciones de los usuarios del campus de la UOC. Para todo el conjunto de pruebas se usará un fichero "campusF5.-192.168.18.2.log-20120311" de 6,06 GB y 13,032916 millones de filas o algún subconjunto de él suministrado por el Tutor que además corresponde con los LOGs de datos reales de navegación de un día en el Campus Virtual de la UOC.

Será la siguiente:

Caso de uso:	1.1 Cargar Fichero completo
Grupo de pruebas:	Importación de datos

#	Prueba	Resultado esperado	Resultado obtenido

F1	Cargar "campusF5.-192.1168.18.2.log-PEQUE.txt con **5.500 filas**	Importación correcta en menos de 5sg.	Importacion OK en 2,53 sg
			
			`<?xml version="1.0" encoding="UTF-8"?>`
			
			`<Resultado xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><HayError>false</HayError><Codigo>Exito</Codigo><Resultado> Se ha procesado 5671 lineas</Resultado></Resultado>`

F2	Cargar "campusF5.-192.1168.18.2.log-MED.txt con **25.531 filas**	Importación correcta en menos de 60sg.	Importación Ok, en 37,31 sg.
			
			`<?xml version="1.0" encoding="UTF-8"?>`
			
			`<Resultado xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><HayError>false</HayError><Codigo>Exito</Codigo><Resultado> Se ha procesado 25168 lineas</Resultado></Resultado>`

F3	Cargar "campusF5.-192.1168.18.2.log-GRANDE.txt con **116.832 filas**	Importación correcta en menos de 5 min.	Importación Ok, en 6min 23,34 sg.
			
			`<?xml version="1.0" encoding="UTF-8"?>`
			
			`<Resultado xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><HayError>false</HayError><Codigo>Exito</Codigo><Resultado> Se ha procesado 113832 lineas</Resultado></Resultado>`

F4	Cargar fichero con línea mal formada. (Comillas)	Devolución del error y registro en LOG	Se devuelve mensaje siguiente: `<?xml version="1.0" encoding="UTF-8"?>` `<Resultado xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><HayError>true</HayError><Codigo>Error_Inesperado</Codigo><Resultado>[IMPORT]-[LeerFicheroCompleto]-ERROR INESPERADO-La conversión de la cadena "" en el tipo 'Long' no es válida.-RECIBIDO:Ruta=W:\Mis Documentos\Mio\UOC\PFC\LOGS\campusF5.-192.1168.18.2.log-GRANDE.txt,GuardaEnBD=False</Resultado></Resultado>`  Y se registra line de log:  `06/11/2015 10:42:15 | [IMPORT] | | [ProcesaLinea] | -ERROR INESPERADO-La conversión de la cadena "ozilla/5.0"" en el tipo 'Long' no es válida.-RECIBIDO: | 0 | |`  `06/11/2015 10:42:15 | [IMPORT] | | [LeerFicheroCompleto] | -ERROR INESPERADO-La conversión de la cadena "" en el tipo 'Long' no es válida.-RECIBIDO: Ruta=W:\Mis Documentos\Mio\UOC\PFC\LOGS\campusF5.-192.1168.18.2.log-`

			GRANDE.txt,GuardaEnBD=False | 0 | |
	[10/Mar/2012:00:08:17 +0100] 1.202.218.8 "GET /robots.txt HTTP/1.0" 404 "-" ""Mozilla/5.0" 1 1		
F5	Cargar Fichero con IP invalida 10.256.20.12	Devolución de error y registro en LOG	Se devuelve el siguiente mensaje: <?xml version="1.0" encoding="UTF-8"?> <Resultado xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><HayError>true</HayError><Codigo>Error_Inesperado</Codigo><Resultado>[IMPORT]-[LeerFicheroCompleto]-ERROR IP INCORRECTA-La dirección IP indicada no es valida IP: 10.256.20.12-RECIBIDO:Ruta=W:\Mis Documentos\Mio\UOC\PFC\LOGS\campusF5.-192.1168.18.2.log-PEQUE.txt,GuardaEnBD=False</Resultado></Resultado> Y se registra en línea log: 06/11/2015 11:40:34 | [IMPORT] | | [ProcesaLinea] | -ERROR INESPERADO-IP invalida IP: 10.256.20.12 -RECIBIDO: | 0 | |

	Caso de uso:	1.2 Cargar Fichero por líneas	
	Grupo de pruebas:	Importación de datos	
#	Prueba	Resultado esperado	Resultado obtenido
F6	Cargar "campusF5.-192.1168.18.2.log-PEQUE.txt con **5.500 filas** **Inicio: 1** **Fin:10**	Importación correcta en menos de 5sg de 11 lineas.	Importacion OK en 1,16 sg <?xml version="1.0" encoding="UTF-8"?> <Resultado xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><HayError>false</HayError><Codigo>Exito</Codigo><Resultado> Se ha procesado 11 lineas</Resultado></Resultado>
F7	Cargar "campusF5.-192.1168.18.2.log-MED.txt con **25.531 filas** **Inicio: 10** **Fin:12**	Importación correcta en menos de 1sg de 3 lineas	Importación Ok.. <?xml version="1.0" encoding="UTF-8"?> <Resultado xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><HayError>false</HayError><Codigo>Exito</Codigo><Resul

			tado> Se ha procesado 3 lineas</Resultado></Resultado>
F8	Cargar "campusF5.-192.1168.18.2.log-PEQUE.txt con **5.500 filas** **Inicio: 2** **Fin:2**	Importación correcta en menos de 1sg de 1 linea	Importacion OK. <?xml version="1.0" encoding="UTF-8"?> <Resultado xmlns="**http://tempuri.org/**" xmlns:xsd="**http://www.w3.org/2001/XMLSchema**" xmlns:xsi="**http://www.w3.org/2001/XMLSchema-instance**"><HayError>false</HayError><Codigo>Exito</Codigo><Resultado> Se ha procesado 1 lineas</Resultado></Resultado>
F9	Cargar "campusF5.-192.1168.18.2.log-MED.txt con **25.531 filas** **Inicio: -1** **Fin:12**	Error	Se devuelve el siguiente mensaje: <?xml version="1.0" encoding="UTF-8"?> <Resultado xmlns="**http://tempuri.org/**" xmlns:xsd="**http://www.w3.org/2001/XMLSchema**" xmlns:xsi="**http://www.w3.org/2001/XMLSchema-instance**"><HayError>true</HayError><Codigo>Error_Inesperado</Codigo><Resultado>[IMPORT]-[LeerFichero]-La línea de inicio debe ser un valor positivo y menor que la línea de fin-RECIBIDO:Ruta=W:\Mis Documentos\Mio\UOC\PFC\LOGS\campusF5.-192.1168.18.2.log-PEQUE.txt,LineaInicio =-1, LineaFin= 12,GuardaEnBD=False</Resultado></Resultado> Y se registra en línea log: 06/11/2015 11:40:34 | [IMPORT] | | [ProcesaLinea] | -ERROR DE ENTRADA DE DATOS-La línea de inicio debe ser un valor positivo y menor que la línea de fin-RECIBIDO:Ruta=W:\Mis Documentos\Mio\UOC\PFC\LOGS\campusF5.-192.1168.18.2.log-PEQUE.txt,LineaInicio =-1, LineaFin= 12,GuardaEnBD=False | -1 | |

Caso de uso:	**2 Generar fichero intermedio**		
	Grupo de pruebas:	**Generar ficherointermedio**	
#	**Prueba**	**Resultado esperado**	**Resultado obtenido**
F6	Generar fichero sin cargarlo previamente.	Detectar error	Se devuelve el siguiente mensaje: <?xml version="1.0" encoding="UTF-8"?> <Resultado xmlns="**http://tempuri.org/**" xmlns:xsd="**http://www.w3.org/2001/XMLSchema**" xmlns:xsi="**http://www.w3.org/2001/XMLSchema-instance**"><HayError>true</HayError><Codigo>Error_Inesperado</Codigo><Resultado>[IMPORT]-[LeerFichero]-La línea de inicio debe ser un valor positivo y menor que la línea de fin-

			RECIBIDO:Ruta=W:\Mis Documentos\Mio\UOC\PFC\LOGS\campusF5.-192.1168.18.2.log-PEQUE.txt,LineaInicio =-1, LineaFin= 12,GuardaEnBD=False</Resultado></Resultado> Y se registra en línea log: 06/11/2015 11:40:34 | [IMPORT] | | [ProcesaLinea] | -ERROR DE ENTRADA DE DATOS-La línea de inicio debe ser un valor positivo y menor que la línea de fin-RECIBIDO:Ruta=W:\Mis Documentos\Mio\UOC\PFC\LOGS\campusF5.-192.1168.18.2.log-PEQUE.txt,LineaInicio =-1, LineaFin= 12,GuardaEnBD=False | -1 |
F7	Introducir opción invalida. Opcion: -1	Importación correcta en menos de 1sg de 3 lineas	Se devuelve el siguiente mensaje: <?xml version="1.0" encoding="UTF-8"?> <Resultado xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><HayError>true</HayError><Codigo>Error_OpcionSeleccionadaNoValida</Codigo><Resultado>[GeneradorFICH-[GeneraFicheroIntermedio]-ERROR - Opcion no implementada-RECIBIDO: Opcion=-1</Resultado></Resultado> Y se registra en línea log: 06/11/2015 12:02:40 | [GeneradorFICH] | | [GeneraFicheroIntermedio] | -ERROR - Opcion no implementada-RECIBIDO: Opcion=-1 | -1 | |
F8	GenerarFichero con opcion valida. Opcion =1	Generacion OK	Generar el fichero intermedio CSV y XML OK Devuelve el mensaje. <?xml version="1.0" encoding="UTF-8"?> <Resultado xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><HayError>false</HayError><Codigo>Exito</Codigo><Resultado>[PRINCIPAL-FichIntTOCSV]-Fichero Intermedio Generado RUTAS:C:\Temp\2015-11-06_12054503.csv | C:\Temp\2015-11-06_12054503.xml-RECIBIDO Opcion=1</Resultado><ResultadoXML/></Resultado> Se registra en LOG: 06/11/2015 12:05:45 | [IMPORT] | | [PreProcesaFichero] | Se ha procesado 5671 lineas y se han conservado 1295 lineas | 0 | |
F9			

Dado que la `programación de las opciones 1 a 5 son análogas, se va a generalizar con la opcion 1 y alguna otra aleatoria que se pueda hacer.

Es diferente para el caso la opción 6 que debe relacionarse con un WebService que devuelve la ubicación de una determinada IP.

Caso de uso:		2.6 Generar fichero intermedio (Opcion 6)	
	Grupo de pruebas:	Generar fichero intermedio	
#	Prueba	Resultado esperado	Resultado obtenido
F10	Generar fichero correctamente.	Generacion OK	Generar el fichero intermedio CSV y XML OK Devuelve el mensaje. <?xml version="1.0" encoding="UTF-8"?> <Resultado xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><HayError>false</HayError><Codigo>Exito</Codigo><Resultado>[PRINCIPAL-FichIntTOCSV]-Fichero Intermedio Generado RUTAS:C:\Temp\2015-11-06_12272660.csv | C:\Temp\2015-11-06_12272660.xml-RECIBIDO Opcion=1</Resultado><ResultadoXML/></Resultado> Se registra en LOG: 06/11/2015 12:15:23 | [PRINCIPAL] | | [FichIntToCSV] | Fichero Intermedio Generado RUTAS:C:\Temp\2015-11-06_12272660.csv | C:\Temp\2015-11-06_12272660.xml-RECIBIDO Opcion=1| 0 | |
F11	Invocar operación sin estar habilitado el WS	Presentar Error	Presentar el mensaje <?xml version="1.0" encoding="UTF-8"?> <Resultado xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><HayError>true</HayError><Codigo>Error_Inesperado</Codigo><Resultado>[PRINCIPAL-[FicheroInt6]-ERROR INESPERADO-No es posible conectar con el servidor remoto-RECIBIDO: </Resultado></Resultado> Se registra LOG: 06/11/2015 12:18:16 | [PRINCIPAL] | | [FicheroInt6] | -ERROR INESPERADO-No es posible conectar con el servidor remoto-RECIBIDO: | -1 | |

Caso de uso:		3 Generar fichero final	
	Grupo de pruebas:	Generar fichero Final	
#	Prueba	Resultado esperado	Resultado obtenido

F12	Generar el Fichero FINAL OK	Generacion OK	Presentar el mensaje: <?xml version="1.0" encoding="UTF-8"?> <Resultado xmlns="http://tempuri.org/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"><HayError>false</HayError><Codigo>Exito</Codigo><Resultado>[PRINCIPAL-FichFINAL_TOCSV]-Fichero FINAL Generado RUTAS:C:\Temp\2015-11-06_12444505.csv-RECIBIDO:</Resultado></Resultado> Se registra LOG: 06/11/2015 12:44:45 | [PRINCIPAL] | | [FichFINAL_TOCSV] | Fichero FINAL Generado RUTAS:C:\Temp\2015-11-06_12444505.csv-RECIBIDO: | 0 | |
F13			

8.3. Estudio experimental

El estudio experimental consiste en realizar una batería de pruebas que permitan determinar un estudio de rendimiento en el entorno de producción. En el caso que nos ocupa, desde la UOC se ha habilitado un entorno de producción que se accede por el protocolo RDP, yaren.uoc.edu y habiendo desplegado el PFC siguiendo el manual de Administrador que se adjunta al actual documento, será en este entorno en donde realizaremos las pruebas.

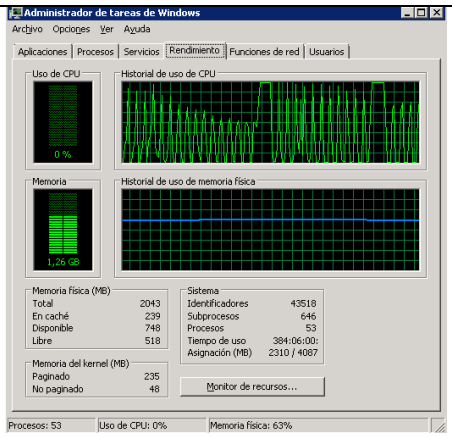
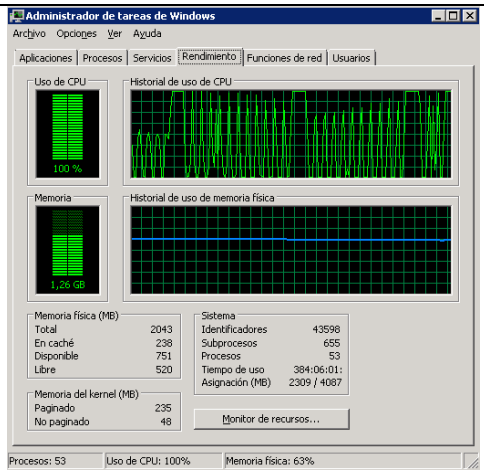
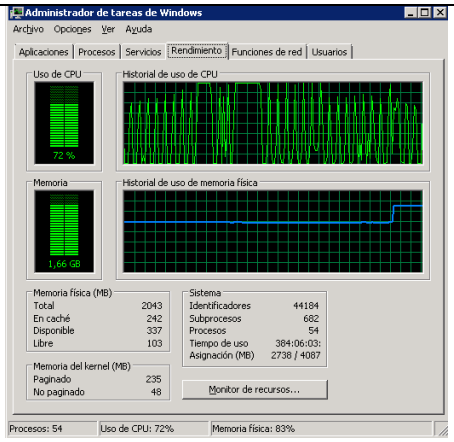
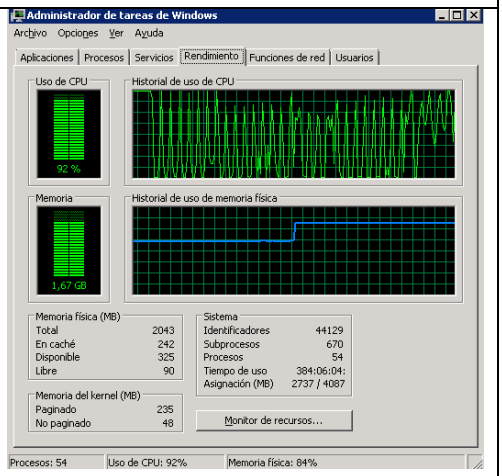
La máquina suministrada por la UOC tiene las siguientes características:

Yaren.uoc.edu	
S.O.	Windows Server 2008 R2 Standard
Procesador	Intel XEON 2.27 Ghz
Memoria	2 GB
HDD	23.8 GB

	Estudio EXP	1
	Grupo de pruebas:	**Eficiencia de procesamiento/ Escalabilidad en cap. cómputo**
#	**Prueba**	**Resultado obtenido**
EP1	**LineasLOG= 5.671**	Inicio: 02/12/2015 12:46:50

	ConsolidarBD=NO **Operac= LeerFicheroCompleto**	FIN: 02/12/2015 12:46:58 **Total: 8 seg** Resultado: Fichero Cargado
EP2	**LineasLOG= 5.671** **ConsolidarBD=SI** **Operac= LeerFicheroCompleto**	Inicio: 02/12/2015 12:52:53 FIN: 02/12/2015 12:53:29 **Total: 36 seg** Resultado: Fichero Cargado
EP3	**LineasLOG= 2.000** **ConsolidarBD=NO** **Operac= LeerFicheroPorFilas**	Inicio: 02/12/2015 12:57:45 FIN: 02/12/2015 12:57:47 **Total: 2seg** Resultado: Fichero Cargado
EP4	**LineasLOG= 10.000** **ConsolidarBD=SI** **Operac= LeerFicheroPorFilas**	Inicio: 02/12/2015 12:59:28 FIN: 02/12/2015 13:00:25 **Total: 56 seg** Resultado: Fichero Cargado
EP5	**LineasLOG= 25.000** **ConsolidarBD=NO** **Operac= LeerFicheroPorFilas**	Inicio: 02/12/2015 13:13:12 FIN: 02/12/2015 13:15:19 **Total: 2min 7 seg** Resultado: Fichero Cargado
EP6	**Operac=GenerarFichIntermedio** **Opcion = 1** **Filas Cargadas = 10.000**	Inicio: 02/12/2015 13:04:07 FIN: 02/12/2015 13:04:07 **Total: 1 seg** Resultado: Fichero intermedio generado
EP7	**Operac=GenerarFichIntermedio** **Opcion = 6** **Filas Cargadas = 10.000**	Inicio: 02/12/2015 13:07:36 FIN: 02/12/2015 13:07:41 **Total: 5 seg** Resultado: Fichero intermedio generado
EP8	**Operac=GenerarFichIntermedio** **Opcion = 1** **Filas Cargadas = 25.000**	Inicio: 02/12/2015 14:12:29 FIN: 02/12/2015 14:12:47 **Total: 18 seg** Resultado: Fichero intermedio generado
EP9	**Operac=GenerarFichIntermedio** **Opcion = 6** **Filas Cargadas = 2.000**	Inicio: 02/12/2015 13:15:07 FIN: 02/12/2015 13:27:07 **Total: 54 seg** Resultado: Fichero intermedio generado
EP10	**Operac=GenerarFichFinal** **Opcion = 1** **Filas Cargadas = 25.000**	Inicio: 02/12/2015 14:23:00 FIN: 02/12/2015 14:23:05 **Total: 5 seg** Resultado: Fichero Final generado

	Estudio EXP	**2**	
	Grupo de pruebas:	**Eficiencia de memoria**	
#	**Prueba**	**Memoria antes**	**Memoria ocupada en el proceso**

EM1	**LineasLOG= 5.671** **ConsolidarBD=NO**		
EM2	**LineasLOG= 5.671** **ConsolidarBD=SI**		

	Estudio EXP	3
	Grupo de pruebas:	**Ocupación en disco**
#	**Prueba**	**Ocupación HDD**
EHD1	**Filas BD=0** **ConsolidarBD=NO** **ModoDepuracion=NO**	22.015.741.952 Bytes (20.5 GB)
EHD2	**Filas BD=14856** **ConsolidarBD=SI** **ModoDepuracion=NO**	22.022.369.280 Bytes (20.5 GB)
EHD3	**Filas BD=52.504** **ConsolidarBD=SI** **ModoDepuracion=NO**	22.50.119.680 tes (20.5 GB)

8.4. PRUEBA PILOTO

Como se ha indicado anteriormente, la finalidad de esta prueba piloto es verificar la viabilidad del proceso descrito antes de iniciar el trabajo con datos reales. En este apartado se detalla el desarrollo de la prueba que se realizó durante la fase de especificación y análisis del proyecto y los resultados obtenidos

8.4.1. *Preparación inicial*

Para realizar la prueba se ha utilizado un fichero producido por el aplicativo PFC con el formato siguiente:

- **Fichero Intermedio**

88.13.89.216	Spain-Santa Barbara-Catalonia	003a28a6f28c933a37a076da3f00c63e6965e23b5b1ef601b3a208511bc1ffc9e83908efc330ee3b14bde8b34f7aa8da4292280a59f6c26241786a81ffd5f2d6
88.13.89.216	Spain-Santa Barbara-Catalonia	003a28a6f28c933a37a076da3f00c63e6965e23b5b1ef601b3a208511bc1ffc9e83908efc330ee3b14bde8b34f7aa8da4292280a59f6c26241786a81ffd5f2d6
79.148.251.140	Spain-Las Palmas de Gran Canaria-Canarias	01ae78f4de7d954717b362d0aedfe3f44071ab31d7c2c6d4ff87a894d1c6f6c85b61c5ab806c4d486ffa0946bd9c5d70f179398252c4a32333df02963c8005c2
79.148.251.140	Spain-Las Palmas de Gran Canaria-Canarias	01ae78f4de7d954717b362d0aedfe3f44071ab31d7c2c6d4ff87a894d1c6f6c85b61c5ab806c4d486ffa0946bd9c5d70f179398252c4a32333df02963c8005c2
88.19.144.28	Spain-Ciutat Vella-Catalonia	01aef2bf53e64810559dfc4bb75fda05fbeb2c16c75e6983a9022bd99695cc7d1ed0cf98a1bc445766cbaa693e7a1f791539b0b1da29356ede46be3fb3c72667
88.19.144.28	Spain-Ciutat Vella-Catalonia	01aef2bf53e64810559dfc4bb75fda05fbeb2c16c75e6983a9022bd99695cc7d1ed0cf98a1bc445766cbaa693e7a1f791539b0b1da29356ede46be3fb3c72667
84.126.53.217	Spain-Valencia-Comunidad Valenciana	01f532692c0465f9b35c9b59f33a951714eba48fd9c837b01879b66bb35118c3659b0ce418f1894cdc509c2252923f11f30a6e01c715f9f46d3968cf7d091729
84.126.53.217	Spain-Valencia-Comunidad Valenciana	01f532692c0465f9b35c9b59f33a951714eba48fd9c837b01879b66bb35118c3659b0ce418f1894cdc509c2252923f11f30a6e01c715f9f46d3968cf7d091729
88.5.84.98	Spain-Madrid-Madrid	02be9ef0b66ad949afb90ccb8902e442a4fd216c96fcdae82d67db26743d77c0cd090ed1b306529b933c7954b028ddd10c575e552181ad3787565a54af8afe24
88.5.84.98	Spain-Madrid-Madrid	02be9ef0b66ad949afb90ccb8902e442a4fd216c96fcdae82d67db26743d77c0cd090ed1b306529b933c7954b028ddd10c575e552181ad3787565a54af8afe24
88.23.198.240	Spain-Terrassa-Catalonia	03749a7c3ea54db0436b846bf051dc84f88f8194c918cc7c7c8185ab6140cf867b27bfbec16ad9ca23360a171fa5285ae45016b9a701a405655a10c84c148e23
186.144.81.132	Colombia-Bucaramanga-Santander	03f1f78822e38b1c8c836cc5f8c87930698bf197019dae222581370ab3da7abb411d13312534130ae2b16e8bd1a016ea1841f4f490d0c52a8886056d73ecc453

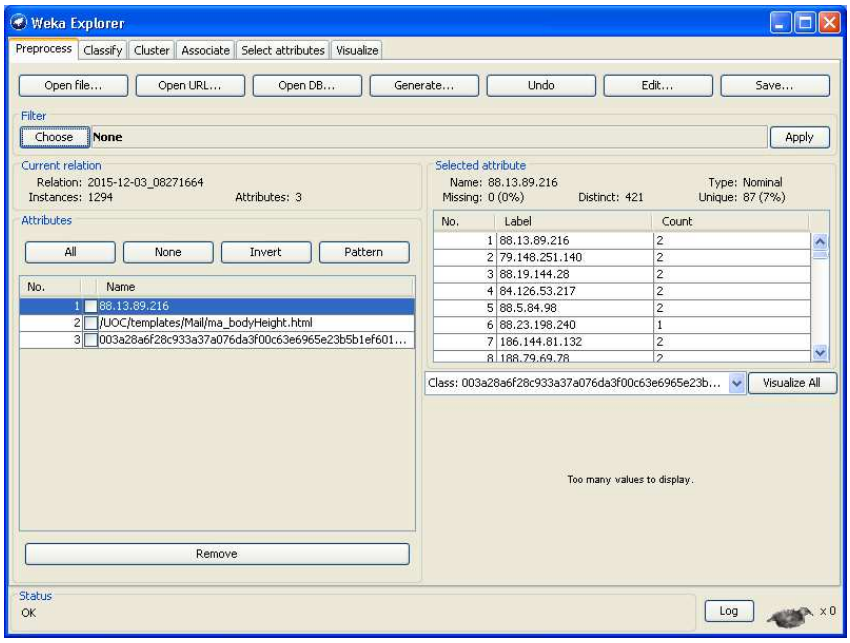
Este fichero se ha cargado en Weka mediante el módulo 'Weka Explorer' y sobre él, se ha realizado la ejecución del algoritmo.

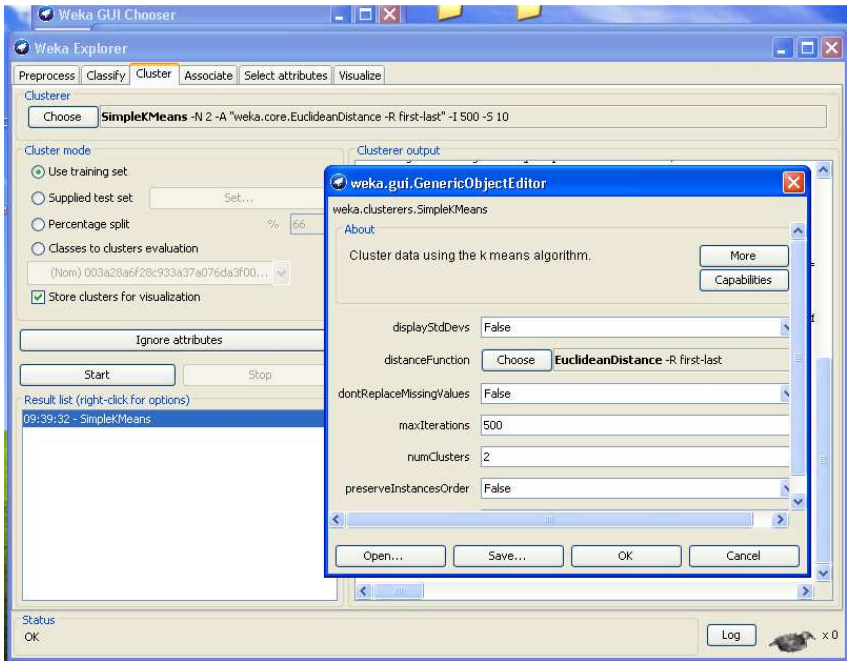
- **Fichero Final**

8.4.2. ***SimpleKMeans***

Se ha ejecutado el algoritmo de agregación (clustering) SimpleKMenans, conservando los parámetros por defecto fijados por Weka.

- **Fichero Intermedio**



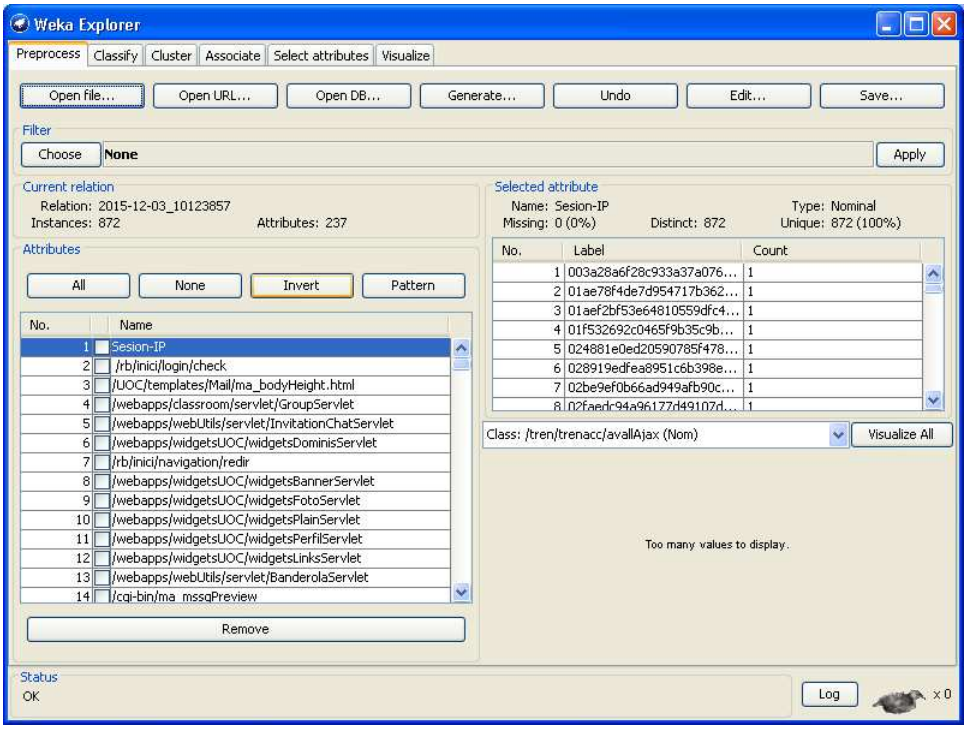


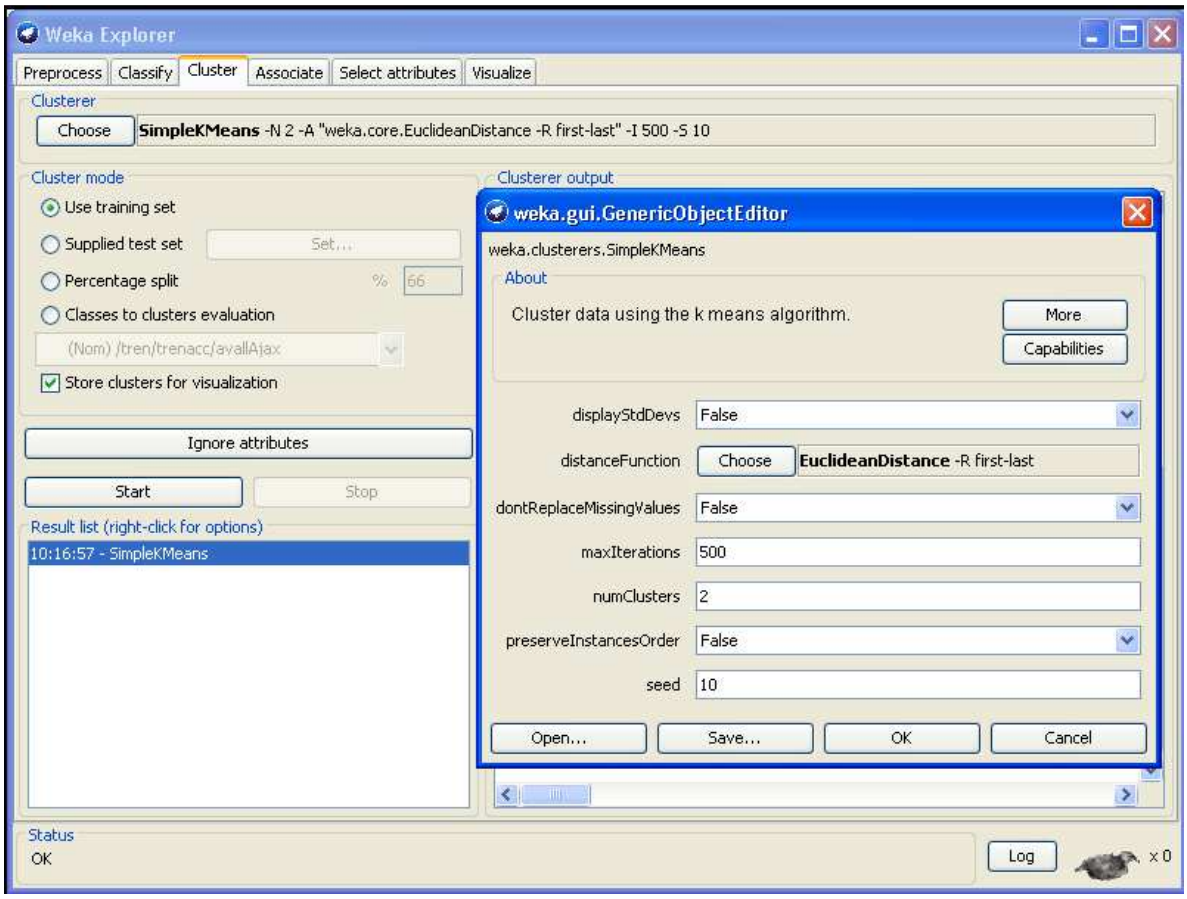
Con los siguientes resultados

```
kMeans
======

Number of iterations: 2
Within cluster sum of squared errors: 14639.0
Missing values globally replaced with mean/mode

Cluster centroids:

Cluster#
Attribute
Full                                                                    Data
0
1

(5939)
(5928)
(11)
==========================================================================
==========================================================================
==========================================================================
==========================================================================
==========================================================================
==========================================================================
==========================================================================
==============
188.86.81.116
192.168.1.137
192.168.1.137
62.83.107.228
/rb/inici/login/check
/webapps/webUtils/servlet/InvitationChatServlet
```

```
/webapps/webUtils/servlet/InvitationChatServlet
/webapps/webUtils/servlet/InvitationChatServlet
001d22b481b85166176515d67b87830ee8955763c70fa3f90433d4635d778333db9d68aa
5fe8be682935726fcf24f1e56e3ba27ce291599f0b6f45dc17354c60
75c439101aec2153852c0d6339275021a005c888cf6bf8b6a3a6d662310adc7f180cf8ac
1ad3464705f6948910ee7518acfcd84643dd3017acfd48a29c3d78c3
75c439101aec2153852c0d6339275021a005c888cf6bf8b6a3a6d662310adc7f180cf8ac
1ad3464705f6948910ee7518acfcd84643dd3017acfd48a29c3d78c3
17459d00c977fd262482d99122f68db12d13763e8df1a290b8cfe05b2265ab901cf419d8
9857f7c07011c684582b2e234ee53bb3587756c700ccc13fa2fc01b4



Time taken to build model (full training data) : 0.03 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       5928 (100%)
1         11 (  0%)
```

- **Fichero Final**






```
kMeans
======

Number of iterations: 4
Within cluster sum of squared errors: 3190.0
Missing values globally replaced with mean/mode

Cluster centroids:

Cluster#
Attribute
Full                                                                    Data
0
1

(872)
(623)
(249)
==============================================================================
==============================================================================
==============================================================================
==============================================================================
==============================================================================
==============================================================================
==============================================================================
=====
……
```

```
Time taken to build model (full training data) : 0.19 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       623 ( 71%)
1       249 ( 29%)
```

Como interpretación de los resultados, decir que el 71% de las secuencias de navegación has sido asociadas al grupo 0 y el 29% al grupo 1. Esto quiere decir, que los usuarios del Campus Virtual, en cada una de sus sesiones de trabajo, suelen acceder habitualmente a los recursos asociados al grupo 0 mayoritariamente mas que a los recursos asignados al grupo 1.

9. Conclusiones y mejoras

9.1. Conclusiones

El objetido fijado al comienzo del PFC era la de análisis de patrones de navegación de los estudiantes del Campus Virtual de la UOC, se ha pretendido principalmente

1. Poder explotar los datos de las mayores formas posibles, en funcion de las posibilidades que ofrecen los LOGs del Servidor Web Apache.
2. Utilizar para ello herramientas y tecnologías flexibles, abiertas,…

Si bien, a nivel particular, no dominaba las herramientas de aprendizaje automatico como Weka, me ha permitido iniciarme en ellas.

9.2. Mejoras

Se proponen las siguientes mejoras:

1. Los logs generados por el Servidor Web Apache, generan mucho “ruido”, líneas de registro de logs que no son útiles para realizar el análisis de navegación del Campus de la UOC. Es por eso que se podría realizar un proceso que consolidase únicamente las líneas de log útiles. Asi evitaríamos que cada vez que se importasen los datos del fichero LOG se leyesen las líneas superfluas. Hay que tener en cuenta que solo el 20% de las líneas de los LOGs se usan para analizar la navegación.

2. Se han implementado 6 opciones de minería de datos de los LOGs de registro, pero todas ellas relacionan IP-Sesion y otro campo de datos.
   Una posible mejora seria flexibilizar estos campos y que fuese el usuario que bajo su criterio se auto-generase su propia descarga.

3. El aplicativo se ha pensado para que toda la información entrante y saliente del mismo se haga desde el Servidor de datos. Es decir, por ejemplo para cargar los datos, se

establecerá una ruta concreta C:\temp o \\direccionIP\RecursoCompartido del filesystem a donde deberá tener acceso.
Una mejora podría ser que el sistema permitiese hacer "Upload" de la información, y que esas rutas del FileSystem sean del equipo cliente, suministrando ademas una interfaz de selección de archivo gráfica.

4. Análogamente al punto anterior, los resultados se descargan en la misma ruta del filesystem del Servidor de Datos, al cual el usuario deberá tener permiso de acceso. Una posible mejora seria que el usuario desde la capa de cliente ASP.NET pueda descargarse el archivo resultado.

5. La opción 6 del aplicativo, aquella que devuelve la ubicación de una determinada IP, consulta por cada IP al WS_IPlocator, que a su vez consulta con una base de datos MySQL, que aunque esta indizada y se busca por ese índice, la ejecución del algoritmo y la consulta reiterada a ese WS en valores mayores de 1000 elementos, se hace inabordable. Se podría establecer mecanismos para agilizar esta búsqueda.
    a. Más recursos hardware.
    b. Prioridades en hilos de ejecución del MySQL.
    c. Optimizar índices
    d. …

10. Bibliografía

1. *Wikipedia K-Means*
   https://es.wikipedia.org/wiki/K-means
2. *Apache Access Log Format*
   http://support.moonpoint.com/network/web/server/apache/log_format.php
3. *WebTaller - Aprender CGI. ¿Cuál es la diferencia entre GET y POST?.*
   http://www.webtaller.com/construccion/lenguajes/cgi/lessons/diferencia.php
4. *HTTP ("Hyper Text Transfer Protocol").*
   http://www.cicei.com/ocon/gsi/tut_tcpip/3376c426.html#SECTION00733300000000000000
5. *Weka. Wikipedia. Disponible en:*
   http://es.wikipedia.org/wiki/Weka_%28aprendizaje_autom%C3%A1tico%29
6. *Weka*
   http://www.cs.waikato.ac.nz/ml/weka/
7. *VirtualBox. Wikipedia*
   https://es.wikipedia.org/wiki/VirtualBox
8. *VirtualBox.*
   https://www.virtualbox.org/

9. *MySQL. Wikipedia*
   https://es.wikipedia.org/wiki/MySQL
10. *MySQL.*
   http://www.mysql.com/

11. *Windows Server 2003*
   https://es.wikipedia.org/wiki/Windows_Server_2003
12. *SoapUI*
   http://www.soapui.org/

13. *SoapUI.Wikipedia*
   https://en.wikipedia.org/wiki/SoapUI

14. *Ip2Location*
   www.ip2location.com