



Trabajo final de carrera

**INGENIERÍA TÉCNICA EN
INFORMÁTICA DE SISTEMAS**

**Facultat de Matemàtiques
Universitat de Barcelona**

**IVUS CHALLENGE 2011: DISEÑO E
IMPLEMENTACIÓN DE LA WEB**

Enric Cañada Muñoz

Director: Simone Balocco
Realizado en: Departament de Matemàtica
Aplicada i Anàlisi. UB
Barcelona, 17 de diciembre de 2012

*“Los informáticos hicimos un trato con Dios.
Él no tocaba los ordenadores y nosotros no hacíamos milagros.”*

Anónimo

1. INTRODUCCIÓN	1
1.1 Ámbito del proyecto	2
1.2 Motivación	2
1.3 Objetivos	3
1.4 Organización de la memoria	5
2. ANTECEDENTES	6
2.1 Situación actual	6
2.2 Conclusión	7
3. PLANIFICACIÓN	8
3.1 Análisis del tiempo de realización del proyecto	8
3.2 Especificaciones de las fases del proyecto	10
4. ANÁLISIS	12
4.1 Casos de uso	12
4.1.1 Caso de uso del usuario sin registrar en la web	12
4.1.2 Caso de uso del usuario registrado en la web.	13
4.1.3 Caso de uso del administrador.	15
4.2 Requerimientos generales: software y hardware	17
4.2.1 Usuarios	17
4.2.2 Servidor	18
5. DISEÑO	19
5.1 Diagrama de secuencia del sistema.	19
5.1.1 Registro de nuevo usuario	19
5.1.2 Login de usuario registrado y subida de fichero.	20
5.2 Base de datos, modelo relacional.	21
5.2.1 Modelo Entidad-Relación	21
5.2.2 Tablas	22
5.3 Diseño de estilo de la web.	23
5.3.1 Localización de los ficheros de imagen y CSS.	26
5.4 Algoritmos auxiliares	26
6. IMPLEMENTACIÓN Y RESULTADOS	29

6.1 Implementación de la web, verificación usuarios por parte del admin.	31
6.2 Resultados, capturas de la web.	32
6.3 Valoración económica del proyecto.	36
6.3.1 Análisis del tiempo de realización del proyecto.	36
6.3.2 Valoración del coste.	37
7. CONCLUSIÓN	37
8. REFERENCIAS BIBLIOGRÁFICAS	39
9. ANEXOS: MANUAL TÉCNICO	40
9.1 Configuración de Apache, la web	40
9.2 Migración de los datos en Mysql, mysqldump.	43
9.3 Configuración de Apache, la web	44
9.4 Configuración del CRON (MacOS y Linux)	45
9.5 Configuración de las tareas programadas (Sistemas Windows)	46
9.6 Estructura de los ficheros	48
9.6.1 Detalle de los ficheros	49
9.6.2 Otros ficheros importantes	53
9.6.3 Migración al servidor del CVC	55
10. ANEXOS: ADMINISTRACIÓN DE LA WEB	57
11. ANEXOS: MANUAL DEL USUARIO.	62
11.1 Registro	62
11.2 Verificación del registro	63
11.3 Subida de los datasets	63
11.4 Pérdida de la contraseña	64

1. Introducción

Esta memoria aborda el desarrollo detallado de la aplicación web para la gestión de los ficheros de *Lumen + External Elastic Laminae (Vessel inner and Outer Wall) Border Detection in IVUS 2011*.

En septiembre de 2011 se organizó en la conferencia MICCAI (Medical Image Computing and Computer Assisted Intervention) un challenge de segmentación (reconocimiento de los bordes de las arterias) sobre imágenes médicas intravasculares (IVUS). Posteriormente, se decidió permitir a nuevos grupos de investigación, que estuvieran interesados, en comparar sus algoritmos con el estado del arte. Como los datos que se utilizaron como referencia (contornos dibujados por los médicos), no se podían distribuir a la comunidad directamente, una propuesta fue la creación de un sitio web que permitiera el análisis y la cuantificación de los resultados de cada participante de forma automática.

La web permite el control y administración del challenge. Los usuarios pueden registrarse en la web, subir los “data-sets” que serán validados automáticamente y consultar el resultados final comparando con otros grupos de investigación.

El administrador, tendrá la posibilidad de validar usuarios, editar los mismos (incluidas las puntuaciones obtenidas) y otras opciones de gestión de la web.

1.1 **Ámbito del proyecto**

El proyecto nace para la administración y seguimiento del challenge IVUS 2011. Permite automatizar la subida de los ficheros de los usuarios y obtener resultados automáticos.

La web creada está diseñada para que los grupos de investigación reciban una respuesta rápida de sus datos que son evaluados en tiempo real y de forma automática por el sistema.

Además permite a los administradores tener los ficheros subidos por los grupos de forma ordenada directamente en su servidor, desde donde los pueden descargar mediante la utilización de un explorador de archivos incluido en la web, para su análisis más detallado.

El proyecto ha sido realizado utilizando PHP, HTML, Javascript implementado en Ajax con el soporte de bases de datos en MySQL.

1.2 **Motivación**

Uno de los principales motivos para la elección de este proyecto es el interés en el aprendizaje de nuevos lenguajes de programación orientados hacia la red. Para la elaboración de la web, hemos utilizado Ajax junto a PHP más mySQL, que durante la carrera no he tenido la oportunidad de aprenderlos, ya que solo se han tratado en la asignatura de “Programació per a la xarxa” a nivel muy básico.

El interés en el uso de PHP y Ajax viene en parte por el crecimiento observado en el mercado, ya que se puede observar el aumento del uso de estas tecnologías en internet, cada vez más creciente y donde poco a poco se van migrando todas las aplicaciones,

incluso las que desarrollaban históricamente como aplicaciones de escritorio, tales como Microsoft Office. A día de hoy, ampliar los conocimientos en este sector en auge, es una buena opción a la hora de ser un activo importante en cualquier empresa.

Y por último me interesaba ampliar los conocimientos sobre la gestión de la web, que quizás sea uno de los procesos más desconocidos dentro del ámbito académico. Entendiendo la gestión como procesos tales como la búsqueda de un servidor donde alojar la web apropiado para las necesidades de la aplicación, la utilización de alias, búsquedas de dominios, precios de estos servicios y las diferentes aspectos técnicos tales como las formas de acceso (SSH, FTP, WebDav).

1.3 Objetivos

El objetivo principal del proyecto es el análisis, diseño e implementación de la página web para la gestión del challenge IVUS 2011.

Para obtener este resultados distinguimos varias tareas:

- Diseño de una interfaz gráfica web.
- Diseño de la Base da datos que utilizaremos.
- Crear un interface para gestionar la administración de los usuarios, comprobar ficheros, cambiar noticias y seleccionar los mails de aviso.
- Crear un registro de usuarios.
- Crear una interface que permite a los usuarios, validarse y subir sus ficheros.
- Análisis de los ficheros subidos por los usuarios.
- Cálculo de los resultados obtenidos.

- Creación de una tabla dinámica donde se muestran las puntuaciones obtenidas por los usuarios.

Como objetivos más técnicos tenemos:

- Buscar el lenguaje que se adecue más a las necesidades para la realización de la web. En este caso, HTML, PHP, Ajax, y MySQL.
- Buscar un hosting que permitiera la tecnología necesaria para implantar la web. En este caso un Apache corriendo PHP en la última versión y una Base de datos, escogido un servidor de MySQL 5.
- Establecer un diseño de la web, mediante una plantilla y un css.
- Analizar las necesidades de las tablas para la base de datos y realizar la lógica de la web.

1.4 Organización de la memoria

En la memoria se recogerán al detalle los siguientes puntos del proyecto:

1. **Introducción:** Dar una visión del porqué del proyecto y cuales son los puntos que se han de tener en cuenta para llevarlo a cabo.
2. **Antecedentes:** Visión global de lo que conlleva el diseño de la aplicación.
3. **Planificación:** Diagrama de Gantt con las tareas y las fechas.
4. **Análisis:** casos de uso y requisitos del software.
5. **Diseño:** Diseño de la web y diagramas.
6. **Implementación y resultados:** Que tecnologías se han utilizado y cómo se han llevado a cabo en el proyecto.
7. **Conclusión:** Metas alcanzadas, conclusiones generales del proyecto.
8. **Referencias bibliográficas:** Recursos que se han compilado durante la realización del proyecto.
9. **Anexos:** Se incluyen dos manuales de uso, primero el manual técnico, que recoge toda la información de los ficheros, funcionamiento, localización y migración del CVC. Además de los puntos a tratar en el caso que se quiera continuar ampliando la aplicación. El segundo manual de usuario, indica el funcionamiento a los usuarios, como deben registrarse y el formato de los archivos

2. Antecedentes

Explicaremos brevemente la situación de la web y los problemas encontrados con el proceso actual y como podemos mejorar la situación utilizando las herramientas de las que disponemos.

2.1 Situación actual

La web actual del challenge IVUS 2011 <http://cbl.uh.edu/challenges> es una web de texto plano, estática, que no permite ningún tipo de interacción con los participantes. Esta primera página web, se creó sólo para anunciar la existencia del challenge en el 2011 y indicaba los criterios de evaluación a los participantes.

Esta primera página tenía una serie de limitaciones para su posterior uso tras el anuncio del challenge, teniendo en cuenta que hasta el 18/09/2011 no permitía a nuevos usuarios participar.

No ofrece una visión clara de los objetivos, ni la visualización de los usuarios, por lo que llevar un control de los participantes es mucho más complicado. Los usuarios tenían que enviar los resultados mediante un correo electrónico a los administradores del challenge.

La realización de la nueva web, permite tener automatizado partes del proceso anterior, como la subida de ficheros y el análisis de los mismos. La misma aplicación es la que se encarga de verificar los ficheros subidos y en el caso que no tengan el formato correcto, no permite que se suban los mismos al servidor.

Otro de los procesos a tratar es la validación de los usuarios. Permite tener un control más exhaustivo de los participantes en el concurso y un control de las fechas de acceso a la web de manera automática. Los usuarios deben haber subido el fichero de con sus resultados en un plazo máximo de 15 días. En el caso de no cumplir este requisito, se les enviará de manera automática un correo electrónico indicándoles el incumplimiento y se les inhabilitará de la web.

La idea de automatizar estos procesos es eliminar la necesidad de que un responsable se ocupe de esta tarea en los diferentes challenges que se realizarán a lo largo de los años.

2.2 Conclusión

El objetivo del proyecto es satisfacer y facilitar los procedimientos para que los administradores del challenge IVUS 2011 y los participantes tengan los resultados y el control de los mismos de una manera más rápida y efectiva.

También simplificaremos parte de la edición de la web, para que quien la ejecute, no sea necesario que tenga conocimientos de programación para poder editarla. Teniendo en cuenta que la parte que se suele necesitar un mayor número de actualizaciones es la sección de “Lastest news” y la sección de correos que se envían a los administradores de la web, es la que actualmente ha sido desarrollada primero como parte de esta edición tipo CMS.

La gestión de los usuarios activos en la web, permite hacer un seguimiento de los mismos de manera automática.

3. Planificación

Una vez definidos los objetivos del proyecto, procederemos a un análisis de los tiempos de implementación, reuniones, procesos de documentación, migraciones de hostings y pruebas finales.

3.1 Análisis del tiempo de realización del proyecto

Siguiendo el modelo de árbol, en el siguiente diagrama de Gantt se pueden visualizar las tareas y subtareas que se realizaron divididas en fases del proyecto. El objetivo es mostrar el tiempo dedicado previsto para las diferentes acciones. Como tareas globales tenemos la fase de pruebas, que durante todo el proceso se realizaba constantemente.

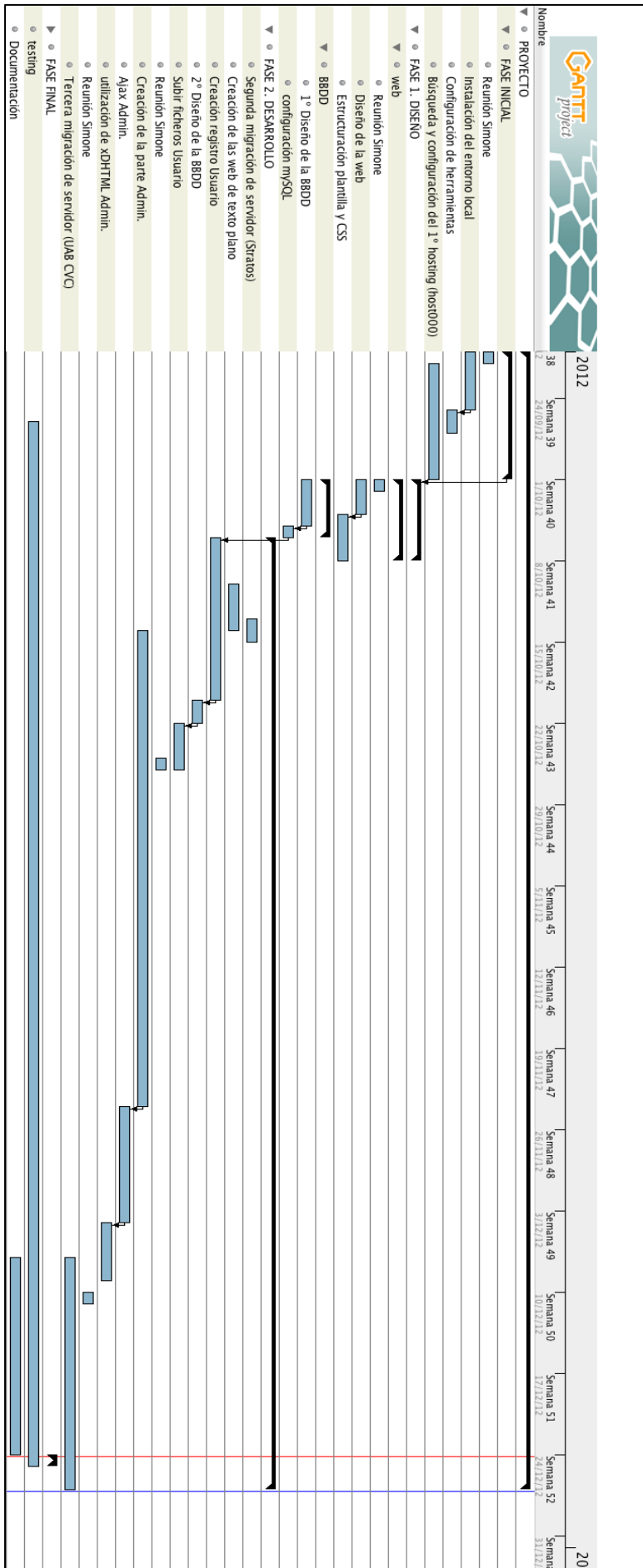


Diagrama de gantt 1

3.2 Especificaciones de las fases del proyecto

En el diagrama de Gantt podemos ver las diferentes fases en las que se ha dividido el proyecto. A continuación detallamos cada una de las fases.

FASE INICIAL

Fase que define los primeros objetivos del proyecto y las herramientas con las que trataremos los mismos.

- El desarrollo y preparación del entorno de trabajo, configuración de Komodo, Dreamweaver, instalación del servidor en local MAMP.
- Búsqueda del primer hosting que permitiera PHP + MySQL y fuera gratuito.
- Creación de la primera idea del proyecto.

FASE DISEÑO

En esta fase detallamos el diseño de la web y la base de datos.

- Estructura de la web, CSS, navegación, diseño base.
- Creación de la primera base de datos

FASE DE DESARROLLO

Fase de la implementación del proyecto.

- Migración a un segundo servidor, donde sí se permitía la implementación de funciones de PHP desactivadas en el primero.

- Implementación de las webs básicas en HTML.
- Desarrollo de la sección de administrador.
- Desarrollo de la sección de usuario.
- Mejoras con Ajax y la inclusión del xDHTML en la parte final del desarrollo.
- Migración al servidor definitivo de la aplicación, en el CVC.

FASE FINAL

- Testing: Desarrollo de las pruebas, que han sido implementadas durante todo el desarrollo del proyecto.
- Documentación: Durante la fase final del proyecto, la redacción de esta memoria.

4. Análisis

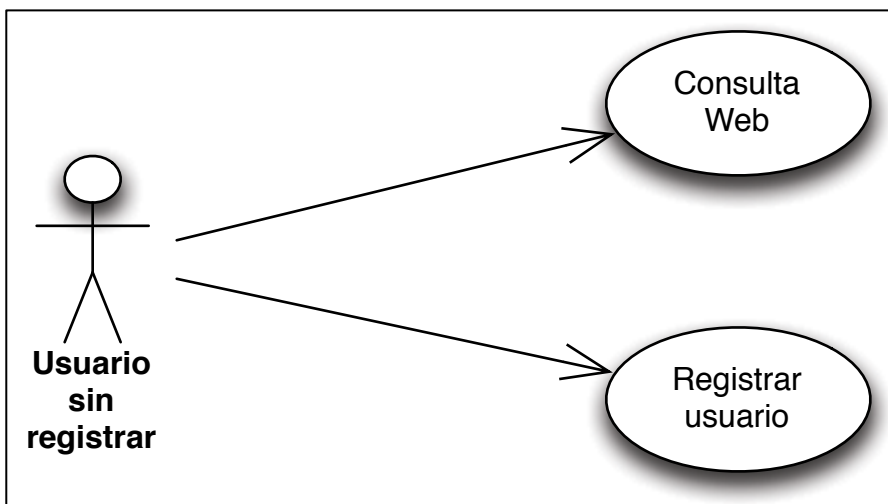
Con los objetivos definidos y las necesidades planteadas para su posterior resolución, procederemos al análisis.

En los siguientes puntos, analizaremos los casos de uso, el diseño visual de la web y una explicación de los requisitos necesarios para el funcionamiento de la misma.

4.1 Casos de uso

La aplicación se puede gestionar desde dos puntos de vista, como usuarios (registrados y sin registrar) y administrador. Desde estos puntos de vista hemos creado los siguientes casos de uso.

4.1.1 Caso de uso del usuario sin registrar en la web



Caso de uso, usuario sin registrar 2

El usuario sin registrar, puede consultar las secciones de la web y puede registrarse en la misma para poder subir los ficheros.

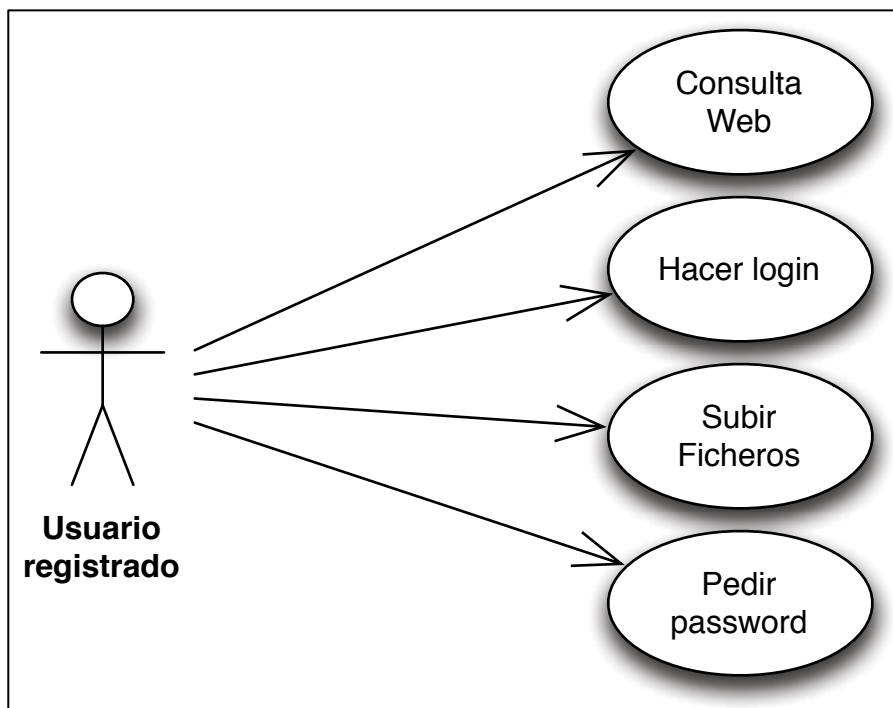
CU-001: Consulta web

CU-001	Consultar la web
Resumen	El usuario introduce la url en el navegador y empieza a explorar la web. La primera web en cargar es el inicio.
Actor Primario	Usuario sin registrar.
Precondición	Ninguna.
Postcondición	Ninguna.

CU-002: Registrar usuario

CU-002	Registrar usuario
Resumen	El usuario introduce sus datos correctamente. Una vez introduce todos y envía el formulario, se le informa que todo ha sido realizado correctamente.
Actor Primario	Usuario sin registrar.
Precondición	El usuario no está registrado
Postcondición	El usuario está registrado temporalmente en la web, a la espera de la verificación del admin. Se envía correo al administrador.

4.1.2 Caso de uso del usuario registrado en la web.



Caso de uso, usuario registrado 3

En el caso de un usuario que se haya registrado previamente en la web, puede realizar login en el apartado de “Upload files” de la web, donde le permitirá subir sus datasets.

También dentro de la misma sección tiene la opción de recuperar el password, si lo había olvidado.

CU-003: Hacer login en la web

CU-003	Hacer login
Resumen	El usuario, que ya esta registrado y verificado, hace login en la web, permitiéndole subir los ficheros del dataset.
Actor Primario	Usuario registrado.
Precondición	El usuario ya está registrado y verificado por un administrador
Postcondición	El usuario, una vez validado, se activa la función de poder subir los ficheros.

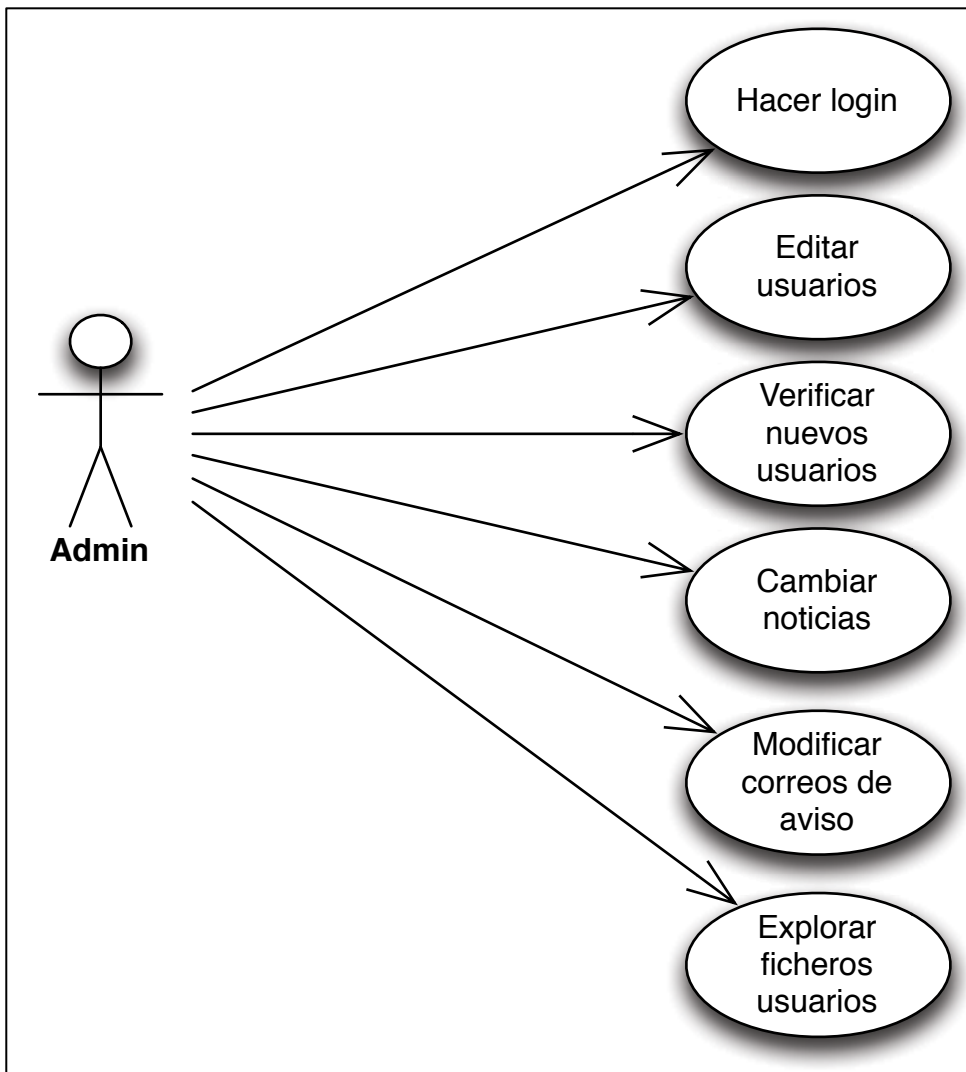
CU-004: Subir ficheros

CU-004	Subir ficheros
Resumen	El usuario puede subir su fichero zip con los datasets. Una vez subido, siempre que el fichero zip y el formato sean correctos, recibirá la puntuación adquirida.
Actor Primario	Usuario registrado.
Precondición	El usuario ya está registrado y verificado por un administrador, además ha realizado login
Postcondición	El usuario ya no puede subir más ficheros, siempre que haya sido correcta la subida y haya recibido ya la puntuación.

CU-005: Pedir password

CU-005	Pedir password
Resumen	El usuario, que ya esta registrado y verificado, ha olvidado la contraseña y la vuelve a pedir. Esta se enviará al email del registro.
Actor Primario	Usuario registrado.
Precondición	El usuario ya está registrado y verificado por un administrador
Postcondición	El usuario, recibirá la contraseña otra vez al email con el que se registró.

4.1.3 Caso de uso del administrador.



Casos de uso de administrador 4

El administrador es el usuario que cuenta con más casos. Primero debe hacer login en la pestaña de “admin” en la propia web. Una vez ha realizado el login, ya puede realizar las diferentes acciones que se muestran en el gráfico.

CU-006: Editar usuarios

CU-006	Editar usuarios
Resumen	El administrador quiere modificar algún atributo de los usuarios.
Actor Primario	Usuario registrado.
Precondición	El admin está logeado y existen usuarios verificados.
Postcondición	Algún campo ha sido modificado y guardado en la base de datos.

CU-007: Verificar nuevos usuarios

CU-007	Verificar nuevos usuarios
Resumen	El administrador quiere verificar el usuario que acaba de registrarse. Comprueba los datos y si es correcto acepta como usuario
Actor Primario	administrador
Precondición	Hay usuarios temporales a la espera de ser verificados.
Postcondición	El usuario que ha sido verificado, recibe un mail de confirmación y ya puede subir ficheros a la web. En caso que sea rechazado se elimina de la base de datos.

CU-008: Cambiar noticias

CU-008	Cambiar noticias
Resumen	El administrador quiere cambiar las noticias de la web, sección Latest News
Actor Primario	Administrador
Precondición	
Postcondición	Las noticias ya están cambiadas en todas las páginas.

CU-009: Modificar correos de aviso

CU-009	Modificar correos de aviso
Resumen	Un administrador quiere cambiar los correos de los administradores
Actor Primario	Administrador
Precondición	
Postcondición	Se añaden o modifican correos a la bbdd para avisar

CU-010: Explorar ficheros de usuario

CU-010	Explorar ficheros de usuario
Resumen	Un administrador quiere comprobar los ficheros subidos, modificarlos o/y borrarlos
Actor Primario	Administrador
Precondición	Que algún usuario haya subido ficheros
Postcondición	Los ficheros han sido borrados, modificados o simplemente visualizados.

4.2 Requerimientos generales: software y hardware

Detallamos el tipo de hardware y software necesario para que la aplicación funcione correctamente. Dividiremos la sección en las necesidades de usuarios y del servidor donde se alojará la aplicación.

4.2.1 Usuarios

Al ser una aplicación web, es multiplataforma y consume los recursos que utilice el navegador.

Hardware: Conexión a internet y hardware capaz de ejecutar cualquiera de los navegadores.

Permite la navegación correctamente desde un dispositivo móvil, pero no la subida de ficheros y la verificación de los usuarios por parte de los administradores.

Se han realizado pruebas con diferentes ordenadores y tablets.

Software: Navegadores actuales con soporte de javascript.

- Mozilla Firefox.
- Google Chrome.
- Safari.
- Internet explorer.

4.2.2 Servidor

El servidor debe tener instalados:

- Un servidor de páginas web.
- Módulo de PHP.
- Una base de datos MySQL.
- Gestor de tareas programadas.

Una configuración válida para sistemas basados en Linux sería la siguiente:

- Apache + módulo PHP + MySQL + Cron

En un sistema Microsoft utilizaríamos la siguiente:

- Internet Information Service (IIS) + módulo PHP + SQL server + Programador de tareas.

En ambos casos, el servidor necesitará tener conexión a internet, y los puertos de conexión necesarios disponibles.

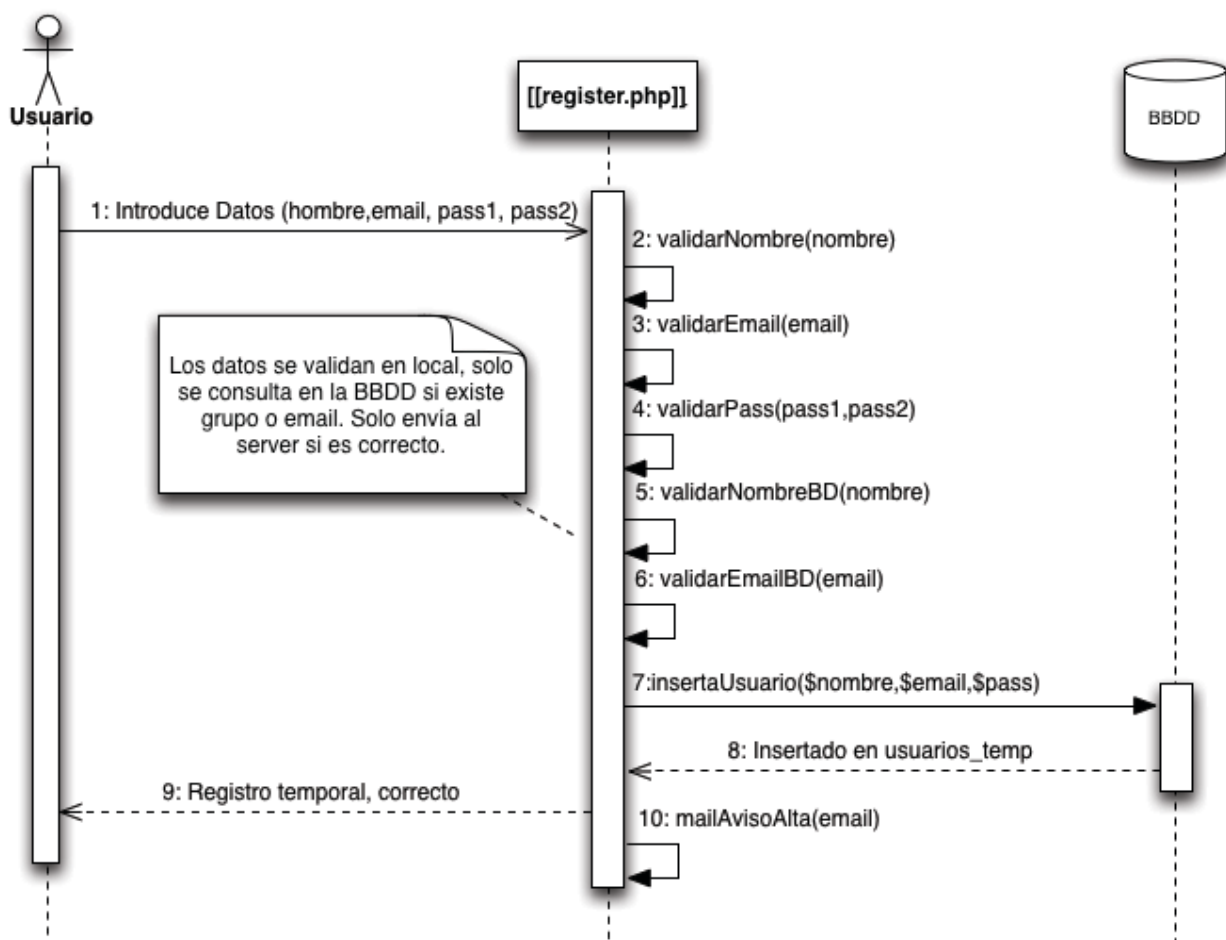
5. Diseño

En esta fase del proyecto se utilizan herramientas para modelar los aspectos dinámicos del sistema.

5.1 Diagrama de secuencia del sistema.

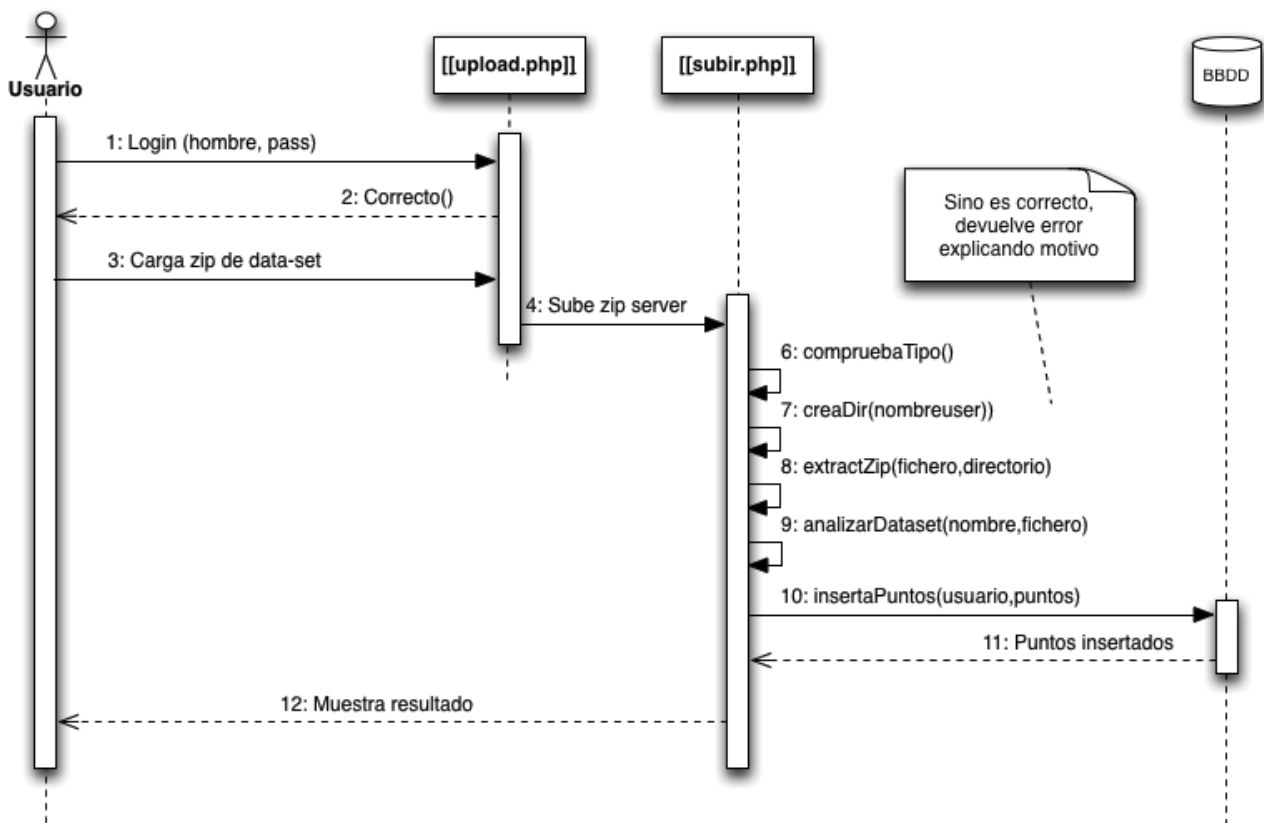
Mostraremos algunos diagramas de secuencia de las funciones más importantes que suceden en la web. Los diagramas de secuencia están adaptados a páginas PHP y no a las clases, ya que no se utilizan clases en el desarrollo de la aplicación, pero se da una visión mucho más clara del funcionamiento de la aplicación.

5.1.1 Registro de nuevo usuario



Registro de un usuario nuevo 5

5.1.2 Login de usuario registrado y subida de fichero.



Login de usuario y subida fichero 6

La carga de los ficheros por parte del usuario se hace mediante un formulario. Una vez en el server es donde se realizan las comprobaciones del fichero subido y si da error, indicamos al usuario donde se ha producido y eliminamos el fichero que ha subido del server.

5.2 Base de datos, modelo relacional.

Después de realizar el análisis de los datos, determinamos que es necesario la creación de una base de datos para el mantenimiento de la web. En esta base de datos guardaremos elementos como los usuarios, con puntuaciones y elementos de la configuración de la web como las noticias o los mails de aviso.

5.2.1 Modelo Entidad-Relación

El esquema de la base de datos es el siguiente:

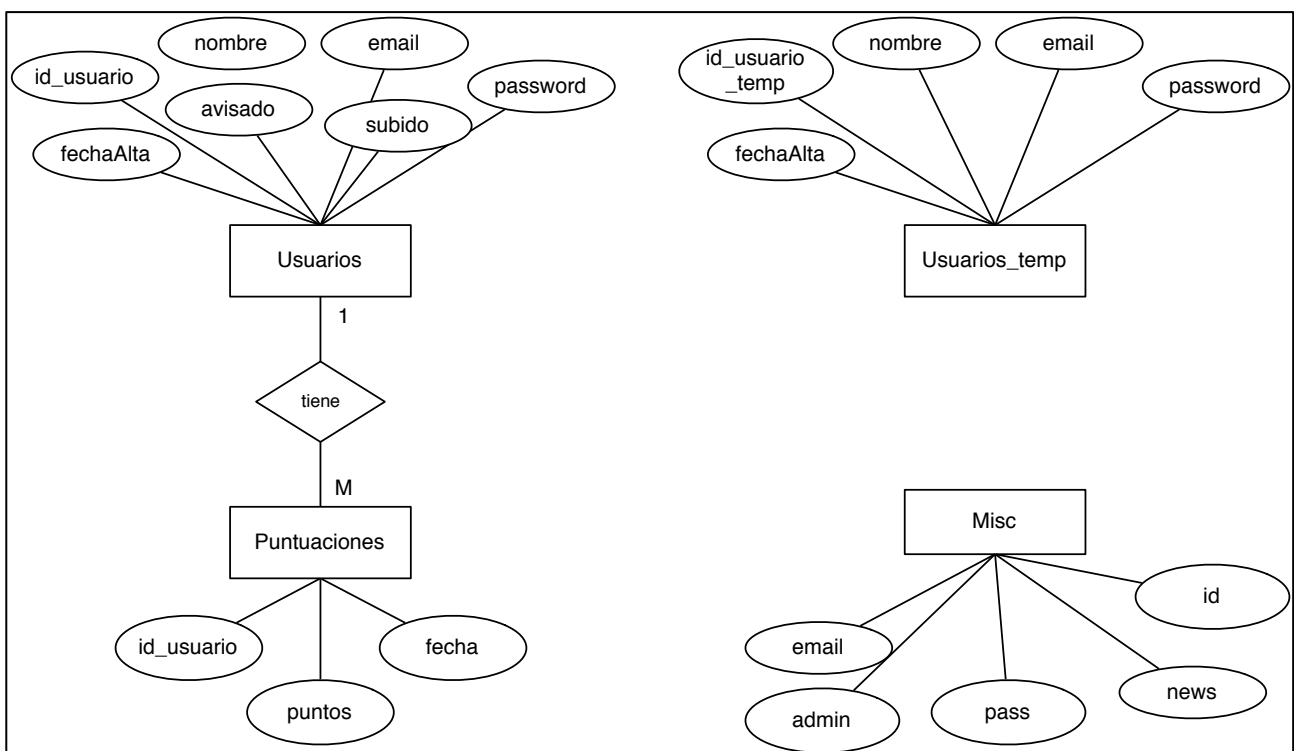


Diagrama entidad relación de la base de datos 7

En la base de datos podemos observar como los usuarios temporales no tienen relación con los usuarios finales ya que muchos de los temporales nunca llegarán a ser usuarios de la web y no es necesaria la relación.

La tabla “Misc”, que es la que se encarga de los datos de configuración de la web, la idea de implementarla es la posible ampliación de la aplicación, que detallaremos en otro apartado.

5.2.2 Tablas

Detallaremos el contenido de las tablas.

USUARIOS

Esta tabla mantiene los registros de los usuarios registrados en la aplicación, indicando además si ya han subido los ficheros. Uno de los campos a resaltar es *avisado* que se utiliza para la gestión de los usuarios que ya están avisados que han sido validados en la web por un administrador. Es un campo de check (boolean)

PUNTUACIONES

Tabla de las puntuaciones. Permite un histórico de los puntos obtenidos por los grupos.

Actualmente no permitimos que un usuario suba más de una puntuación por motivos del diseño. Seguramente esto sea una de las cosas que se quieran cambiar en una revisión del proyecto, por lo que de esta forma, ya dejamos esta parte implementada.

En principio, una vez un usuario sube un fichero y recibe una puntuación, no puede volver a subir otra. Pero si quisiéramos que sí pudiera, eliminado el fichero anterior y modificando el valor de *enviado* de la tabla usuarios, podría volver a subir sus resultados.

USUARIOS_TEMP

En esta tabla se recogen todos los usuarios que han enviado su registro mediante la sección de “register” de la web. No todos los usuarios que se intenten registrar serán

válidos, por lo tanto, no tiene relación con la tabla final de USUARIOS. Los registros no válidos, serán eliminados por el administrador desde la aplicación.

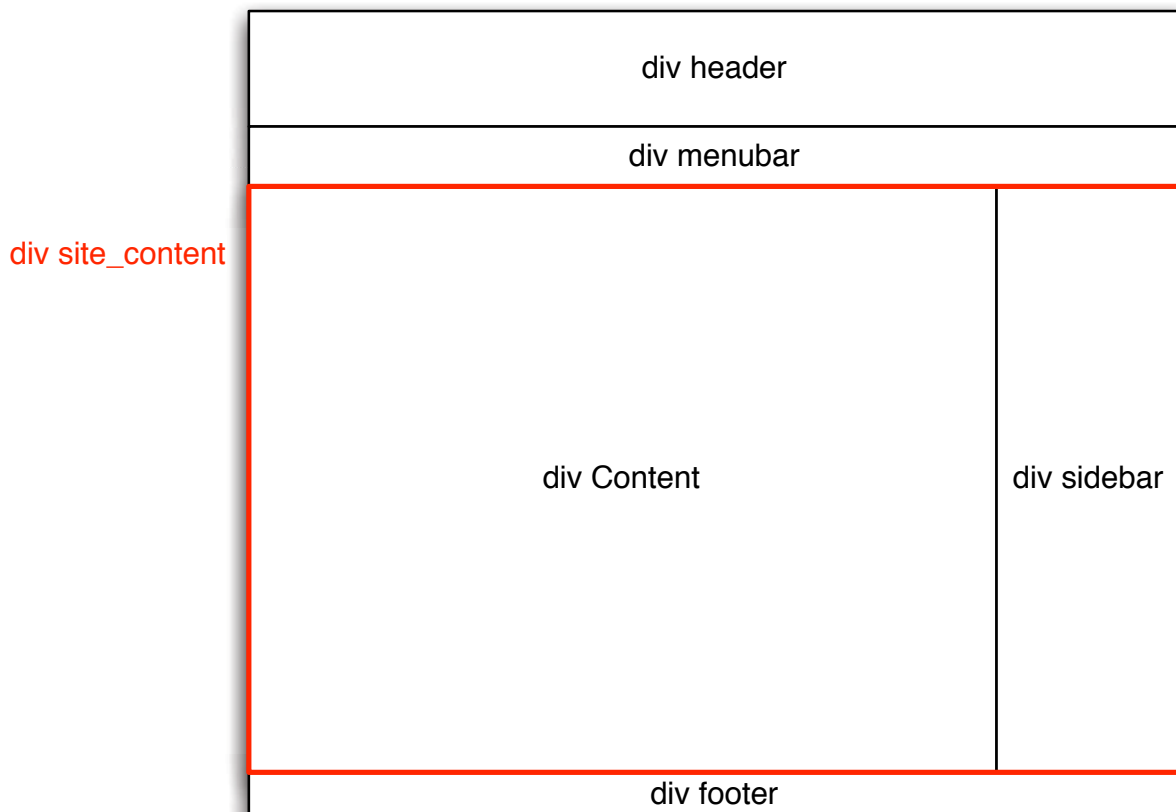
MISC

Se recogen las configuraciones de la aplicación. Se ha incluido esta tabla para permitir un desarrollo escalable de la web.

5.3 Diseño de estilo de la web.

La web, basada en un plantilla gratuita que podemos encontrar aquí <http://www.html5webtemplates.co.uk/> ha sido modificada para se ajuste a las especificaciones del proyecto.

El esquema de la web es el siguiente:



diseño de los divs en el css 8

Este es el esquema de diseño general de la web, pasamos a detallarlos:

- **div id main:** Es el contenedor de toda la web.

- **div id header:** Incluye la cabecera de la web. En **div id logo** encontramos la imagen, que está definida en el css y **div logo_text**, que define el texto superpuesto al logotipo de la web.

- **div menubar:** Carga las pestañas de la web. Se definen como unas listas de html (`pestaña`) y la que está seleccionada lleva la clase (`class="selected"`) de este modo la resalta indicando que es la seleccionada.

- **div id site_content:** este div incluye div content y div sidebar.

- **div id content:** cuerpo principal de la web. En este div se muestra el contenido de la web.

- **div id sidebar:** La sidebar, es la barra lateral, tiene dos partes: "Latest news", que se carga mediante la función de `cargaNews()` donde se rellena con la información que hayamos editado en la parte de administrador y los "Useful Links" donde podemos ver los links de las universidades ub y uab.

- **div id footer:** El pie de la web. Donde podemos ver la links a los mismos enlaces que tenemos en el menú superior, además tenemos el link con el logo de la Universitat de Barcelona.

Lumen + External Elastic Laminae

Border Detection In IVUS Challenge



Home

Dataset

Evaluation

Register user

Upload files

Results

Faqs

About

Admin

Overview

We are pleased to announce a Lumen + External Elastic Laminae (Vessel Inner and Outer Wall) Border Detection in IVUS Challenge, associated with MICCAI 2011 "Computing and Visualization for (Intra) Vascular Imaging workshop (to be held on September 18, 2011 in Toronto, Canada). Participants will be provided with several IVUS pullback datasets from different frequencies and with various degrees of difficulty. Manual annotations will be available for evaluating the results (comparison and scoring). The aim of this challenge is to provide the participants with an opportunity to present/evaluate their segmentation methods with challenging datasets. A journal paper summarizing the challenge results will be co-authored by the challenge participants and the challenge organizers.

The Challenge

The segmentation challenge is divided into two categories:

- Segmentation of the inner wall. (Lumen segmentation)
- Segmentation of the outer wall. (media/adventitia segmentation)

A team can choose to participate in one or both categories. Clinical recordings from 20 MHz and 40 MHz probes will be provided. The data types are:

- Single-frame datasets with adjacent frames in DICOM B RF (40 MHz) format chosen at random instants of the cardiac cycle (not gated).
- Multi-frame datasets in DICOM (20 MHz) format (sequences, ECG or image-based gated).

Even though the 3D sequences consist of 20-50 gated frames to provide sufficient spatial context, in this challenge, only a specified subset of up to 5 frames will be used for evaluation. The results of all the submitted methods will be evaluated and discussed during a designated session of the workshop.

Latest News

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient .

Useful Links

- [dtroni](#)
- [prova dtroniGrid](#)

HOME | DATASET | EVALUATION | REGISTER | UPLOAD FILES | RESULTS | FAQS | ABOUT | ADMIN USERS



resultado final de la web 9

5.3.1 Localización de los ficheros de imagen y CSS.

Todas las imágenes, logo de la web, separadores, marcador de links y estilos de la web las encontramos dentro del servidor en la raíz, directorio **/style**.

Dentro de este directorio también se encuentra el fichero de estilo, **style.css** donde están definidas todas las opciones anteriores de divs y estilos.

5.4 Algoritmos auxiliares

Explicaremos el desarrollo de algoritmos que se han implementado durante el desarrollo de la web dedicados a tareas específicas del proyecto. Encontraremos estos algoritmos implementados en el fichero **subir.php** de la web.

Primero detallaremos los data-set. Los data-set son los ficheros zips que suben los participantes con sus resultados, para ser analizados por la aplicación.

Los datasets son ficheros zips organizados con la siguiente estructura en árbol

```
Nombre.zip
  - Data_set_A
    - LABELS
      lum_frame_14_0014_003.txt
      .
      .
  - Data_set_B
    - LABELS
      lum_frame_01_0001_003.txt
```

Contenido de los data-set 10

Data_set_a, **Data_set_B** y **LABELS** son directorios.

lum_frame_X_XXX_XX.txt son ficheros de texto con el siguiente formato.

```
120,183.5
120.5,184
120.5,185
120.5,186
120.5,187
120.5,188
121,188.5
121.5,189
121.5,190
121.5,191
121.5,192
121.5,193
122,193.5
122.5,194
```

Formato de los ficheros incluidos en el data-set 11

Para su análisis se ha diseñado e implementado el siguiente algoritmo.

Función *analizarDataset(\$nombreUsuario, \$nombrezip)*

1. Descomprimir el zip, ya se ha verificado que es un zip y se ha descomprimido con éxito.
2. Recorremos el primer directorio del data-set (**Data_set_A/LABELS**) y comprobamos si existe, si no existe salimos indicando el error.
3. Si existe iniciamos la iteración el directorio **LABELS**.
4. Mientras que no es final del fichero "lum_fram_X_XX.txt"
 - a. Leer línea, dividida por el carácter ",", evitando líneas en blanco.
 - b. Restar el primer valor del segundo valor.
 - c. Guardar el resultado en la variable total.

5. Una vez recorrido el primer data-set, hacemos lo mismo con el segundo data-set comprobando si existe los directorios otra vez
6. Cerrar fichero.
7. Imprimir resultado.
8. Insertar el resultado en la base de datos.

El proceso de la resta, es simplemente orientativo. En ese lugar se añadirá la función que realmente realiza la comparación de los puntos obtenidos con los puntos reales y de ahí nacerá la verdadera puntuación. Esto de momento no está implementado, ya que no forma parte del proyecto.

Función *eliminarDir(\$completedir, \$deleteme)*

Es una función recursiva que se encarga de eliminar los directorios con su contenido. El problema radica que en PHP no se pueden borrar los directorios que tienen datos dentro. Para eso, esta función, se encarga de eliminar el contenido de los directorios y luego borrarse el mismo.

1. Leer el nombre del directorio a eliminar y la opción un bool, \$deleteme
2. Abrir el directorio, mientras que no encontremos un directorio eliminar ficheros, evitando "." y ".."
3. Si es un directorio, volvemos a llamar a la función con el bool activado
4. Si directorio está vacío, eliminar el directorio y continuar iterando los ficheros.

6. Implementación y resultados

A continuación detallamos qué tecnologías y herramientas hemos utilizado a la hora de implementar el proyecto.

Lenguaje HTML y PHP

La elección de PHP se fundamenta en los puntos siguientes:

- Es de código abierto y funciona en la mayoría de servidores, tanto basados en Linux/unix y como en los que corren bajo plataformas Windows como por ejemplo el IIS. Por este motivo se descartó ASP, ya que al principio se desarrollo en un entorno de servidores de Linux y no era viable la implementación en ASP.
- Es un sistema Flexible y robusto. El problema que tenemos en frente de ASP es que no tiene un buen sistema para realizar debug y no tiene un IDE que nos permita desarrollar rápidamente como sí tiene ASP.
- La comunidad que desarrolla para PHP es muy importante, por lo que es fácil encontrar código y soluciones a problemas que se producen durante el desarrollo de la aplicación.

MySQL

MySQL es uno de los sistemas gestores de bases de datos más utilizados, con más de 6 millones de instalaciones y en aplicaciones como Amazon, Drupal, Flickr. Desarrollado como software libre, con dos tipos de licencias. Por un lado bajo GNU GPL, pero si queremos que este incorporado en un producto privativo se debe comprar una licencia. Su base de datos es muy rápida en lectura, pero con problemas de integridad si se producen

muchas modificaciones a la vez. Si la aplicación tiene poca concurrencia y más lectura de datos, como es nuestro caso, se recomienda el uso de MySQL.

Javascript (Ajax)

Acrónimo de Asynchronous JavaScript And XML. Esta técnica nos permite que se ejecute en el navegador de los usuarios mientras mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma, se pueden realizar cambios sin necesidad de recargar la web lo que permite una mayor interactividad con el usuario y una gran mejora de la usabilidad. En la parte del administrador por ejemplo, nos permite la edición simultánea de diferentes secciones de la web.

CSS (Cascading Style Sheets)

CSS se ha utilizado para la maquetación de la web base, de los estilos y los diferentes apartados visual de la aplicación. La idea de desarrollar con CSS, es separar la estructura del diseño, de la parte lógica por lo que es más fácil de editar cualquiera de las dos. Las ventajas de utilizarlo son:

- Centralizar la presentación del sitio web con lo que las actualizaciones de estilo se agilizan.
- Permite cambiarlo a los usuarios localmente, lo que permite las mejoras de accesibilidad.
- Mostrar diferentes hojas de estilo para diferentes dispositivos (por ejemplo, crear una vista para impresora)

Detallamos el software utilizado para la implementación del proyecto.

Komodo edit

El editor Komodo (versión Edit) permite el desarrollo de muchos lenguajes de programación. Es gratuito, en la versión Edit, y multiplataforma. Tiene muchas ayudas a la programación como resaltar la sintaxis, autocompletar código y link con las funciones.

Dreamweaver CS6

Editor de páginas web. En el proyecto ha sido utilizado para la maquetación de la misma, pruebas con el servidor y la utilidad live, que permite la visualización de código PHP en tiempo real. Se ha utilizado para la edición de la página web cuando no estaba online.

Photoshop CS6

Editor de imágenes, utilizado para la creación del logo y modificación de algunas de las imágenes que podemos ver en la web.

6.1 Implementación de la web, verificación usuarios por parte del admin.

Para implementar la sección de los usuarios en la parte de administrador hemos utilizado **dhtmlxGrid**. Estas librerías, que tienen licencia gratuitas siempre que el proyecto sea también gratuito, están realizadas en javascript e implementadas con Ajax. Es personalizable y editable, mediante la utilización de temas.

Además, permite la utilización de Drag-and-drop en el navegador, por lo que la aplicación web mejora en los términos de usabilidad e interacción con el usuario.

6.2 Resultados, capturas de la web.

Mostramos algunas capturas del resultado final de la web.

Primera página de la web.

The screenshot shows the homepage of the 'Lumen + External Elastic Laminae' challenge website. The header features the title 'Lumen + External Elastic Laminae' and the subtitle 'Border Detection In IVUS Challenge'. The navigation menu includes 'Home', 'Dataset', 'Evaluation', 'Register user', 'Upload files', 'Results', 'Faqs', 'About', and 'Admin'. The main content area is divided into two columns. The left column contains an 'Overview' section with a paragraph of text and a 'The Challenge' section with a list of categories and data types. The right column contains 'Latest News' and 'Useful Links' sections. The footer includes a navigation bar with links to 'HOME | DATASET | EVALUATION | REGISTER | UPLOAD FILES | RESULTS | FAQS | ABOUT | ADMIN USERS' and the logos of the University of Barcelona and CVC.

Lumen + External Elastic Laminae
Border Detection In IVUS Challenge

UNIVERSITAT DE BARCELONA
CVC

Home Dataset Evaluation Register user Upload files Results Faqs About Admin

Overview

We are pleased to announce a Lumen + External Elastic Laminae (Vessel, Inner and Outer Wall) Border Detection in IVUS Challenge, associated with MICCAI 2011 "Computing and Visualization for (Inte) Vascular Imaging workshop (to be held on September 18, 2011 in Toronto, Canada). Participants will be provided with several IVUS pullback datasets from different frequencies and with various degrees of difficulty. Manual annotations will be available for evaluating the results (comparison and scoring). The aim of this challenge is to provide the participants with an opportunity to present/evaluate their segmentation methods with challenging datasets. A journal paper summarizing the challenge results will be co-authored by the challenge participants and the challenge organizers.

The Challenge

The segmentation challenge is divided into two categories:

- Segmentation of the inner wall. (lumen segmentation)
- Segmentation of the outer wall. (media/adventitia segmentation)

A team can choose to participate in one or both categories. Clinical recordings from 20 MHz and 40 MHz probes will be provided. The data types are:

- Single-frame datasets with adjacent frames in DICOM B RF (40 MHz) format chosen at random instants of the cardiac cycle (not gated).
- Multi-frame datasets in DICOM (20 MHz) format (sequences, ECG or image-based gated).

Even though the 3D sequences consist of 20-50 gated frames to provide sufficient spatial context, in this challenge, only a specified subset of up to 5 frames will be used for evaluation. The results of all the submitted methods will be evaluated and discussed during a designated session of the workshop.

Latest News

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient. The only cure for insomnia is to get more sleep.

Useful Links

- www.lb.edu
- www.cvc.uab.es

HOME | DATASET | EVALUATION | REGISTER | UPLOAD FILES | RESULTS | FAQS | ABOUT | ADMIN USERS

UNIVERSITAT DE BARCELONA
CVC

Web inicio 12

Detalle de la sección registro.

Register User

Username:

Email:

Password:

Repeat password:

REGISTER

Registro de la web 13

Aquí vemos un ejemplo de los resultados.

Results

TOTAL SCORES

GROUP NAME	EMAIL	DATE	TOTAL POINTS
Michy_Mouse	hola@gmail.com	23-11-2012	70000
Prueba1	p@p.com	21-11-2012	2000
Pepito	balocco.simone@gmail.com	21-11-2012	1000
Luisete	l@l.com	25-11-2012	20

Muestra de resultados 14

Sección de administrador. Se puede ver las tablas que permiten drag-and-drop.

The screenshot shows an administrator interface with a top navigation bar containing the following tabs: Home, DataSet, Evaluation, Register user, Upload files, Results, and Faqs.

The main content area is divided into several sections:

- Users - double click to edit**: A table with columns NAME, PASS, EMAIL, ALTA, and POINTS. The data is as follows:

NAME	PASS	EMAIL	ALTA	POINTS
Pepito	ub	balocco.simone@gmail.com	2012-11-21	1000
Prueba1	prueba1	p@p.com	2012-11-21	2000
Michy_Mouse	Michy_Mouse@a.com	hola@gmail.com	2012-11-23	70000
Luisete	luis	l@l.com	2012-11-25	20

A "Remove Row" button is located below the table.
- Temporal Users - Drag and Drop to move users**: A table with columns NAME, PASS, EMAIL, ALTA, POINTS, and ACT. The data is as follows:

NAME	PASS	EMAIL	ALTA	POINTS	ACT
milab	petia	radevap@gmail.com	2012-12-03	0	<input type="checkbox"/>
Usuario_prueba	prueba	prueba@test.com	2012-12-28	0	<input type="checkbox"/>

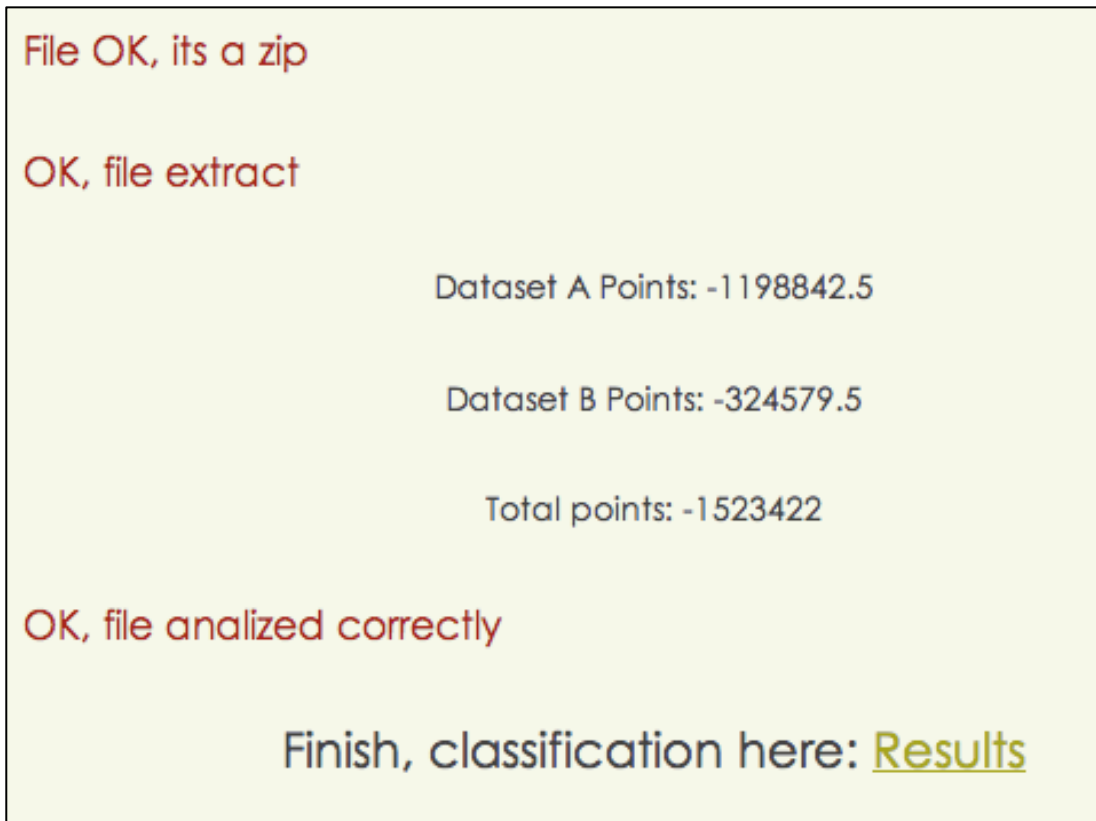
A "Remove Row" button is located below the table.
- Edit latest news**: A text area containing the placeholder text: "Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient .". A "Change news" button is located below the text area.
- Send mails - Mails for the admin. use ',' to separate mails.**: A text input field containing "darkwell@gmail.com". A "Change mails" button is located below the input field.
- File Explorer**: A section with a "Subidas" link and a "[logout]" button in the bottom right corner.

On the right side of the interface, there is a sidebar with the following content:

- Latest**: A section with placeholder text: "Lorem ipsum consectetur Aenean commodo dolor. Aenean sociis natoque magnis dis parturient only cure for get more sleep".
- Useful**: A section with two links: "www.u" and "www.c".

Sección administrador. 15

Este es un ejemplo de un usuario validado que ha subido un fichero zip data-set correcto.



detalle de la web, subido fichero con éxito. 16

En la siguiente captura ilustramos la subida de un fichero zip, que no tiene la estructura de directorios interna bien implementada.

```
File OK, its a zip
OK, file extract
ERROR, directory DatasetA NOT found
ZIP with correct format?
Some error in the procedure. You can try to upload file again Upload File
deleting temporal files...
deleting all files in: subidas/enric
all done
```

detalle de la web, subido fichero con directorios mal formateados 17

6.3 Valoración económica del proyecto.

Analizamos la valoración de costes tanto de tiempo como del coste económico. Los valores de horas son aproximados.

6.3.1 Análisis del tiempo de realización del proyecto.

Concepto	horas
Formación	140
Análisis	180
Implementación	280
Documentación	50
Total	650

Cuadro de tiempo realización 18

6.3.2 Valoración del coste.

Concepto	horas	Precio hora	Precio Total (€)
Formación	140	0€	0
Análisis	180	50€	9.000
Implementación	280	40€	11.200
Documentación técnica	50	40€	2.000
Total	650		22.200

Tabla de precio y tiempo 19

Los costes de implantación de la aplicación de la aplicación depende de la contratación del hosting. En este caso, al ser un proyecto universitario, son llevamos a cabo por el CVC (centre de visió per computació) de la UAB.

7. Conclusión

Los objetivos iniciales que se han llevado a cabo son los siguientes:

- Creación de una nueva web, con un interfaz gráfico más agradable, actualizado y con una mejora de la usabilidad.
- Simplificación de las validaciones de los participantes en el sistema.
- Automatización del control de tiempos de los usuarios, lo que permite a los administradores no estar pendientes de esta tarea.
- Automatización del análisis de los ficheros subidos por los usuarios, y resultados obtenidos.
- Definir una base de datos para el acceso de toda la información de la web de manera ordenada y cómoda.

Las mejoras y ampliaciones de la aplicación serían:

- Añadir Ajax en la sección del usuario, cuando se realiza la gestión de los ficheros y que esto diera una sensación de mayor interactividad por parte del usuario.
- Verificación de una mayor seguridad en la página web, a la hora de cómo navegan los password.
- Evitar en una mayor medida los ataques tanto a la base de datos, como la inyecciones de código por PHP (cosa que se ha realizado en cierta medida).
- La búsqueda e implementación de otra tecnología a la hora de guardar los datos.
- Tener la página web, por ejemplo, en varios idiomas.
- Una mayor edición de la web. Por ejemplo, que se pudieran editar los links contenidos de páginas, etc. La creación de un pequeño CMS propio.
- Mejorar la accesibilidad de la web para personas con problemas de visión.

8. Referencias bibliogrficas

Ttulo: **Enginyeria del software, Disseny UML**, Edicions UPC
Autor: Cristina Gomez – Enric Mayol

Ttulo: **PHP Manual**, PHP Documentation Group
Autor: Mehdi Achour, Friedhelm Betz, ...

Ttulo: **MySQL Reference Manual**.
Autor: Oracle

Ttulo: **Learning PHP, MySQL, JavaScript, and CSS, 2nd Edition**, O'Reilly Media
Autor: Robin Nixon

9. ANEXOS: Manual técnico

Manual que incluye la instalación del servidor de forma local, la migración al servidor del CVC, configuración de los ficheros y la edición de los más importantes ficheros.

9.1 Configuración de Apache, la web

Para la instalación del servidor, para trabajar en local, hemos realizado la instalación de **MAMP**, acrónimo de Macintosh, Apache, MySQL, PHP. Esta aplicación se encarga de instalar en un Mac, un servidor apache, con el módulo de PHP 5 instalado y un servidor de MySQL funcionando. También instala las herramientas para poder administrarlos tanto gráficamente, como por consola.

Este tipo de software esta tanto disponible para Linux, llamado **LAMP**, como para sistemas Windows, en este caso que llama **WAMP**.

Detallaremos la versión para MAC que es la que se ha seleccionado para desarrollar la guía.

Primero, descargamos el software desde la web de <http://www.mamp.info/en/index.html>.



Imagen de la web de MAMP 20

Seleccionaremos la opción gratuita (la gris).

Una vez descargada e instalada, por defecto nos aparecerá esta pantalla



Pantalla de la aplicación 21

Como podemos ver en la imagen están iniciados los servidores tanto de Apache como de MySQL.

En las preferencias, podemos cambiar como inicia la aplicación, los puertos, etc.

Configuraremos lo siguiente:

- Puerto Apache: 8888, puerto por defecto.
- Puerto MySQL: 8889, puerto por defecto.
- PHP, versión 5.4.4
- Apache, seleccionaremos donde trabajaremos con la web.

Nos pedirá la validación y una vez funcione. Para comprobar el funcionamiento correcto podemos hacer click sobre Test MAMP PRO o directamente en el navegador, accediendo a localhost:8888

9.2 Migración de los datos en Mysql, mysqldump.

Para acceder al MySQL desde consola, debemos utilizar este comando en la consola.

```
/Applications/MAMP/Library/bin/mysql --host=localhost -u -p
```

Tenemos un script, que genera la base de datos desde cero con toda la información referente a campos, triggers. Para realizar la carga se utiliza el siguiente comando:

```
mysql database < backup.sql
```

En caso que tengamos que realizar una migración debemos exportar los datos de la base de datos actual, para poder insertarlos en el nuevo servidor. Para eso utilizaremos el comando mysqldump.

```
mysqldump ---user [user name] ---password=[password][database name] > [dump file]
```

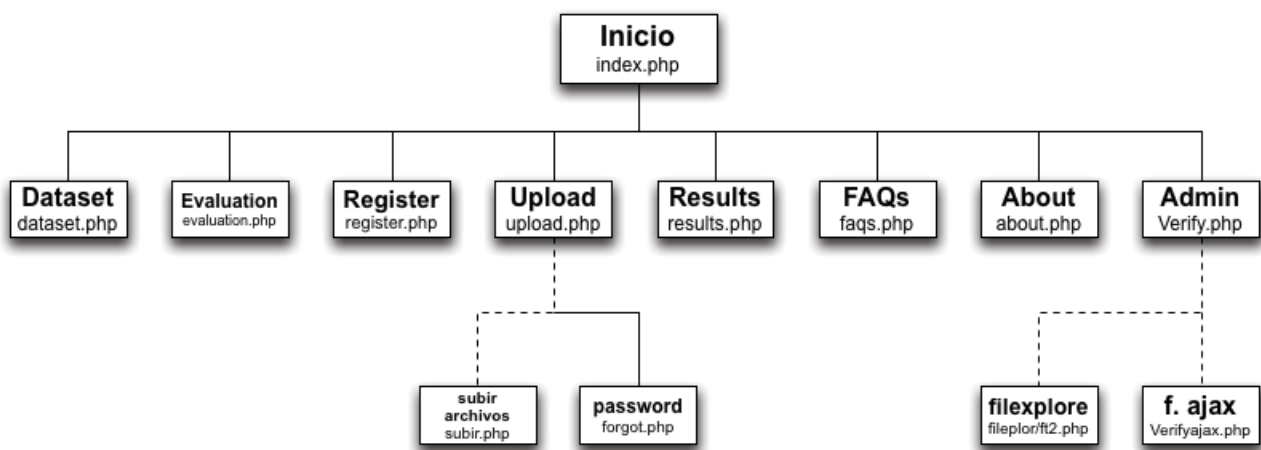
Con esto tenemos una copia de la base de datos para volver a cargar los datos utilizaremos

mysql database < backup.sql

Para realizar la carga desde phpMyAdmin, se utilizará la opción de importar fichero SQL (que previamente hemos realizado)

9.3 Configuración de Apache, la web

Para la configuración de apache, se debe copiar todo el contenido, respetando la estructura de directorios definida anteriormente, en el **documentroot**. Este parámetro ya está definido en la configuración anterior. En los demás sistemas operativos, el funcionamiento es el mismo. Es muy importante que hayamos respetado la estructura de los directorios, ya que sino, no encontrará los archivos y la web no funcionará. La estructura debería ser la siguiente:



Estructura de la web simplificado, ficheros 22

9.4 Configuración del CRON (MacOS y Linux)

Con MySQL y el Apache iniciados, necesitamos que ejecuten los siguientes scripts en PHP de forma automática.

El primero se ejecutará cada 15 minutos y el segundo cada día. Para configurar estos scripts en Linux, haremos lo siguiente:

Teniendo el **cron** activado, ejecutaremos:

```
crontab -E
```

Esto nos abrirá un editor donde podemos escribir la llamada al fichero php que tiene que ejecutarse

```
*/3* * * * /var/www/html/tfc/cron3dias.php
```

```
*/15 * * * * /var/www/html/tfc/cron15dias.php
```

una vez los tenemos insertados, automáticamente se ejecutarán cada 15 minutos y 1 día.

Con el comando CRONTAB -L podemos ver todos los scripts que se lanzaran y cuando.

Para eliminar el que tengamos asociado, usaremos CRONTAB -R

9.5 Configuración de las tareas programadas (Sistemas Windows)

En Windows, podemos realizar una tarea programada para que se ejecuten los archivos cada cierto tiempo, al igual que lo hacemos en los demás sistemas operativos. Para eso necesitamos que la aplicación de tareas programadas este instalada.

Para poder ejecutar los scripts de tareas programadas, necesitamos crear unos ficheros .bat que tendrán este formato.

24h.bat

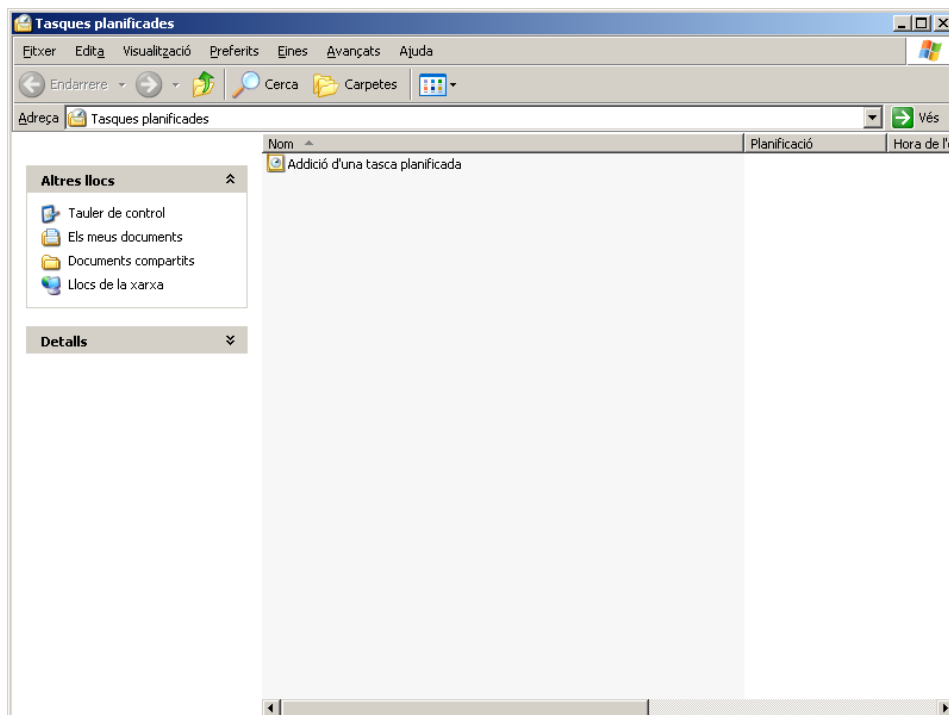
```
C:\servidor\php\php -f C:\servidor\scripts\cron3dias.php
```

15min.bat

```
C:\servidor\php\php -f C:\servidor\scripts\cronuseract.php
```

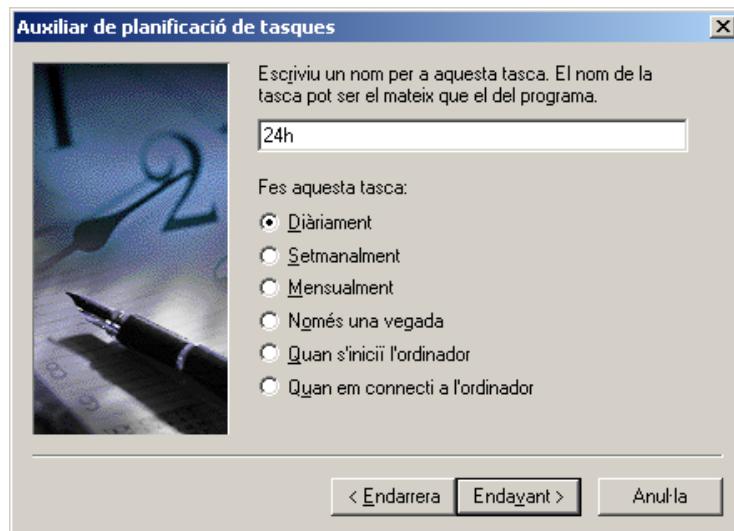
Una vez tenemos los dos ficheros creados, vamos al administrador de tareas. Lo encontraremos en:

Inicio > Accesorios > Herramientas del sistema > Tareas programadas



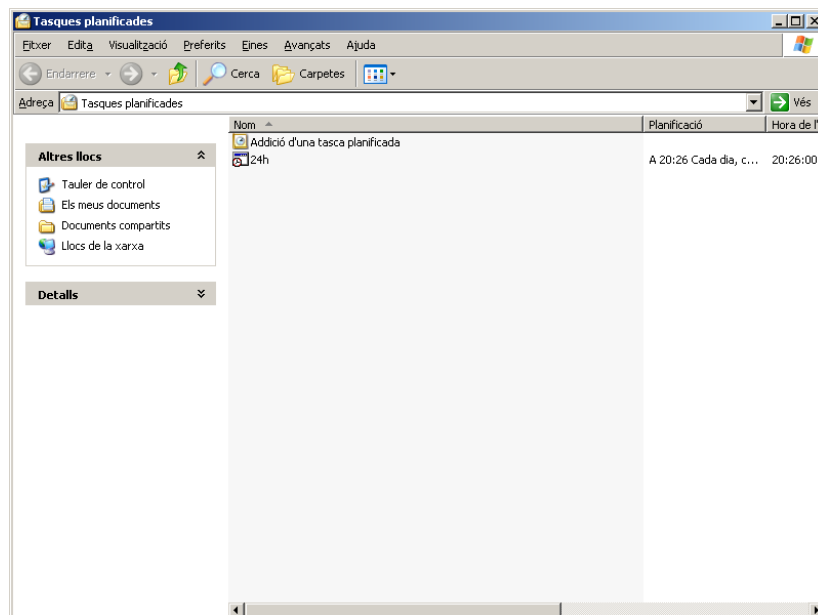
Programador de tareas 23

Seleccionamos añadir “una nueva tareas programa”, y seguimos el “wizard”. Seleccionamos la ubicación del fichero bat creado, nombre y duración. En este caso diariamente.



Muestra del planificador 24

Seleccionamos la hora de la ejecución y el usuario que lanzará la tarea. Una vez creada, nos aparecerá la siguiente pantalla.

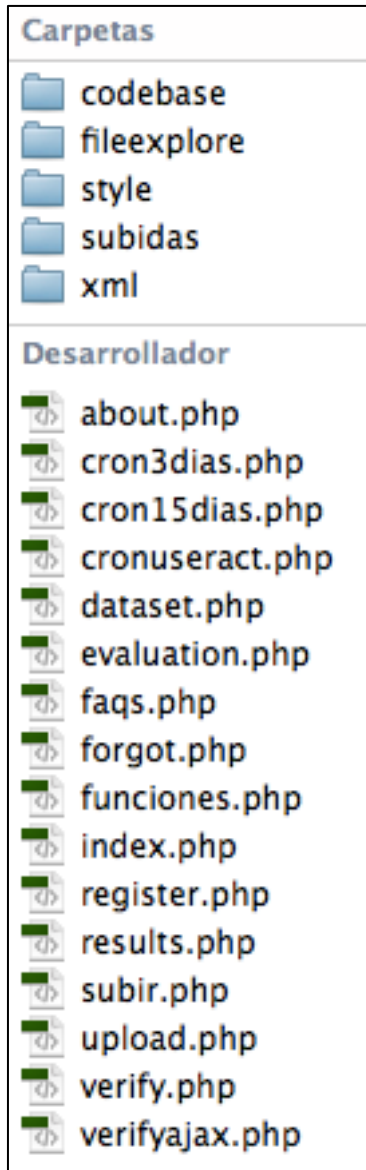


Muestra del script preparado para la ejecución 25

Realizamos los mismos pasos para la creación del otro script, que se lanzará cada 15 minutos.

9.6 Estructura de los ficheros

Los directorios de la web tienen la siguiente estructura que pasaremos a detallar.



Los directorios que se muestran contienen lo siguiente:

codebase: Contiene todos los ficheros necesarios para el funcionamiento correcto del xDHTMLgrid con el connector y el dataprocessor. Estos ficheros son los que se encargan de cargar los datos en las tablas que aparecen dentro de la sección de administradores de la web. No se tienen que modificar ningún fichero de esta carpeta.

Fileexplore: Contiene el explorador de ficheros. El fichero principal es **ft2.php**, y **config.php** que se encarga de la configuración.

Style: Donde se encuentra el CSS de la web, **style.css** y las imágenes como el logotipo de la web, por ejemplo.

Subidas: Directorio donde se subirán los ficheros de los usuarios. La estructura interna se genera automáticamente, y es accesible mediante fileexplorer

Xml: Contiene los ficheros de configuración del xDHTMLgrid. Aquí dentro detallan la base de datos, los accesos que tendrá y la consulta a realizar a la hora de llenar las tablas de la sección de usuarios. **connectorPHP.php** que es el de la tabla usuarios y **connectortempPHP.php** que es el de la tabla de usuarios temporales.

9.6.1 Detalle de los ficheros

Los ficheros incluidos son los siguientes:

About.php

Sobre la gente que ha participado en el proyecto. El contenido es simple HTML, que podemos encontrar en el div de <content>. Como el resto de webs, carga en el div <sidebar> las “lastest news” desde la base de datos con la función *cargaNews()* que se encuentra dentro del fichero **funciones.php**, del que hacemos un include en todas las webs del sistema.

Cron3dias.php y Cron15dias.php

Estos son los archivos que se ejecutan cada día, vía cron o tarea programada, para avisar a los usuarios de que van a caducar sus cuentas. Los ficheros ejecutan una consulta en la BBDD que comprueba la fecha.

Cronuseract.php

Este fichero se lanza cada 15 minutos. Se encarga de comprobar los usuarios que hay en la base de datos que ya han sido verificados por los administradores pero no han sido

avisados de que este paso ha sido realizado. Funciona con una consulta del campo *avisado* en la tabla users.

Dataset.php

Página sobre la estructura de los datasets. HTML al igual que la web de **about.php**

Evaluation.php

Sobre como se evalúan la información. HTML igual que la webs de información. Recordar que todas tienen la llamada a **funciones.php** para la carga de las noticias.

Faqs.php

Faqs de la web. HTML igual que la webs de información. Recordar que todas tienen la llamada a **funciones.php** para la carga de las noticias.

Forgot.php

Se encarga de enviar el password a los usuarios que se han olvidado de él. Se llama desde **upload.php**. El usuario pone su nombre y se le envía un correo con el password, mediante la función que está incluida *mailEnvioPass()*

Funciones.php

Fichero de funciones de apoyo. Este fichero, que se carga en todas las páginas, trae una serie de funciones que se utilizan en la mayoría de ficheros.

- *Conectarbd()*: Es la función que se encarga de conectar con la base de datos. Tiene los parámetros de conexión como son dirección de la base de datos, usuario, password, y base de datos

- *cargaNews()*: Recoge el campo de noticias de la base de datos.

Estas son las funciones que mandan correos. Si se quiere modificar los correos enviados se tienen que editar.

- *mailAvisoAlta(\$email)*: envía un correo al administrador avisando que alguien se ha intentado dar de alta.
- *mailExpiradoUsuario(\$email)*: Se avisa al usuario que se ha expirado su tiempo 15 días.
- *mailAvisaUsuario(\$email)*: Avisa al usuario que le faltan 3 días para expirar.

Index.php

Índice de la web. La primera página que se llama al abrir la dirección del navegador. HTML.

Register.php

Se encarga del registro por parte de los usuarios en la web. Realiza una verificación local antes de enviar nada a la web para no cargar el servidor.

- *Validarnombre(\$string)*: Se encarga de que el nombre no tenga caracteres extraños y que sea mayor de 4 cifras. Permite el uso de underscore “_”
- *ValidarEmail(\$string)*: Válida que se haya introducido una dirección correcta de correo. Sino tiene formato, no permite la entrada.
- *ValidarPass(\$string1, \$string2)*: Comprueba que los dos passwords sean igual, y que no contenga caracteres extraños.

- *ValidarEmailBD(\$string)*: Comprueba en la base de datos que no existe ese correo.
- *ValidarNombreBD(\$string)*: Comprueba que no existe ese nombre en la base de datos.
- *insertarUsuario(\$nombre, \$email, \$pass1)*: Inserta el usuario en la BBDD.

Una vez todo esta correcto lanza *insertarUsuario()* y lo añade a la base de datos. Si encuentra cualquier error lo avisa al usuario. Por último, las variables \$nombreant y \$emailAnt se utilizan para que el usuario no tenga que volver a escribir el nombre y el email en caso que cometiera un error, por ejemplo, en el password.

Results.php

Muestra una tabla con los resultados obtenidos por todos los grupos, ordenados por mayor número de puntos. Es una consulta SQL, la web se ejecuta dinámicamente cada vez que se llama.

Subir.php

Esta es la web que se encarga de la creación de la estructura de directorios, descompresión y del análisis de los ficheros subidos. Solo se puede acceder si has ido por la web de **upload.php**. Muestra el análisis del fichero así como los errores que se producen durante el proceso.

Upload.php

La que se encarga del login del usuario y de la subida del fichero. Tiene, como hemos dicho anteriormente, el enlace por si el usuario ha perdido la contraseña.

Una vez el usuario hace login, se abre la de subida de fichero, que es enviado mediante formulario a **subir.php** que es quien se encarga de toda la gestión del fichero.

Verify.php

La web del administrador. Una vez se hace login aparecen las opciones anteriormente descritas.

La carga del xdhtmlgrid, el mail de los admins, las noticias, etc.

VerifyAjax.php

Se encarga de las llamadas a las funciones desde la web de **verfy.php**. Desde aquí realizan los cambios pero al ser llamados por Ajax, no perdemos el contacto con la web del administrador, de forma que siempre estamos en esta.

9.6.2 Otros ficheros importantes

Fileexplore/config.php

Este fichero incluye la configuración del fileexplorer que podemos ver en la web de administradores. Es necesario editar los siguientes parámetros:

```
define("USERNAME", "ub"); // Your default username.  
define("PASSWORD", "ub"); // Your default password.  
$ft["settings"]["DIR"] = "../subidas"; // Your default directory.
```

Se debe tener especial cuidado en la ubicación del directorio.

Style/style.css

Este es el fichero de estilo de la web, el css.

Xml/connectorPHP.php

En este fichero es donde se encuentran los parámetros de conexión con la base de datos de xDHTMLgrid. La conexión se realiza contra la tabla de usuarios.

```
<?php
    date_default_timezone_set('Europe/Madrid');
    require("../codebase/dhtmlxConnector/codebase/grid_connector.php");
    $res=mysql_connect("158.109.8.29","ecanada","tfc4challen");
    mysql_select_db("grup_petia_ivuschallenge2011");
    $gridConn = new GridConnector($res,"MySQL");
    $gridConn->enable_log("/tmp/grid.log",true);
    $gridConn->
        render_table("usuarios","id_usuario","nombre,password,email,fechaAlta,puntos");
?>
```

En el código encontramos la cadena de conexión a la base de datos, con el usuario y la contraseña, La base de datos, el tipo de conexión y los campos que mostrará la tabla.

Xml/connectorPHPtemp.php

Este fichero es la forma que tiene el xDHTMLgrid de conectar con la base de datos. Desde aquí se configuran los parámetros y ataca a la tabla usuarios_temp.

```
<?php
    date_default_timezone_set('Europe/Madrid');
    require("../codebase/dhtmlxConnector/codebase/grid_connector.php");
    $res=mysql_connect("158.109.8.29","ecanada","tfc4challen");
    mysql_select_db("grup_petia_ivuschallenge2011");
    $gridConn = new GridConnector($res,"MySQL");
    $gridConn->enable_log("/tmp/grid.log",true);
    $gridConn-
>render_table("usuarios_temp","id_usuario_temp","nombre,password,email,fechaAlta,p
untos,activado");
?>
```

Igual que en el caso anterior, en el código encontramos la cadena de conexión a la base de datos, con el usuario y la contraseña, La base de datos, el tipo de conexión y los campos que mostrará la tabla.

9.6.3 Migración al servidor del CVC

9.6.3.1 Ficheros de la web.

Para realizar la última migración del servidor, ha sido necesario subir los ficheros de la web al servidor del CVC (Centre de Visió per Computador).

El servidor corre un IIS, con el módulo de PHP y la función de envío de mails activado. Tenemos acceso a un directorio en red, al que se accede por WebDav. Para acceder a recursos WebDav, tenemos diferentes opciones de software dependiendo del sistema operativo que utilizemos. En el sistema operativo Windows, desde el CVC nos recomiendan el uso del software **bitkinex** 3.2.3 que es gratuito. Para configurarlo debemos asignar los siguientes parámetros de conexión.

- directorio: **https://www.cvc.uab.es/IVUSchallenge2011.**
- usuario: **ecanada**
- password: **tfc4challen**

con este programa configurado, mostrará un directorio llamado **IVUSchallenge2011** que contendrá todos los ficheros de la página web.

9.6.3.2 Base de datos

Para la Base de datos tenemos los siguientes parámetros:

- IP: **158.109.8.29**
- Nombre de la BBDD: **grup_petia_IVUSchallenge2011**
- Usuario: **ecanada**
- Password: **tfc4challen**

Para poder acceder a la gestión de la base de datos, mediante phpadmin, se debe abrir una sesión de VPN contra el servidor del CVC.

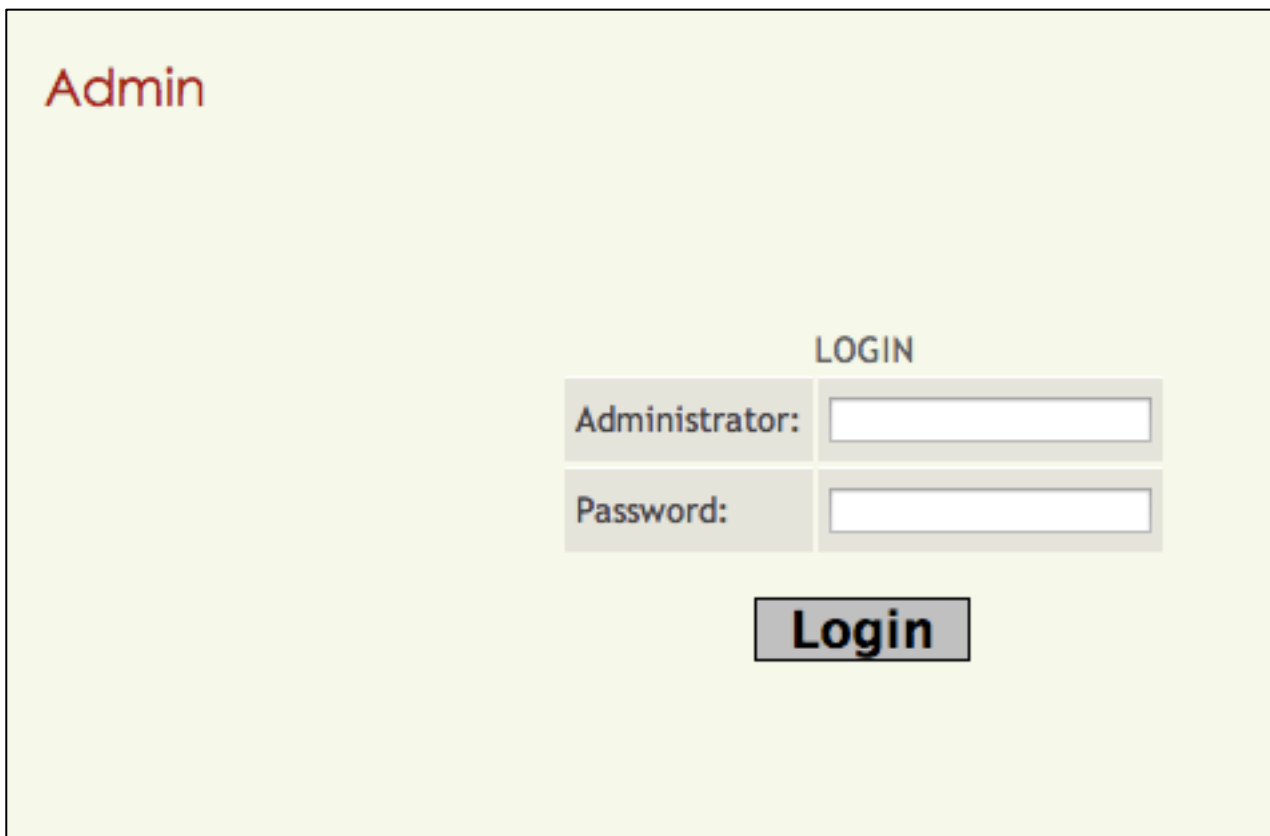
Desde el CVC nos recomiendan la utilización del **netextender**. Una vez instalado nos pedirá la validación con usuario y contraseña.

- Usuario: **ecanada**
- Password: **tfc4challen**

Con esto ya podremos acceder al phpadmin y comprobar el estado de la base de datos, su modificación, y la administración de las copias de seguridad.

10. ANEXOS: Administración de la web

Desde la url <http://www.cvc.uab.es/IVUSchallenge2011/verify.php> nos aparecerá la siguiente pantalla.



Admin

LOGIN

Administrator:

Password:

Login

Login de administrador, detalle de la web 26

Para hacer login, usaremos el usuario ub y password ub. Cuando accedemos, nos aparecerá la pantalla de administrador.

Users - double click to edit

NAME	PASS	EMAIL	ALTA	POINTS
Pepito	ub	balocco.simone@gmail.com	2012-11-21	1000
Prueba1	prueba1	p@p.com	2012-11-21	2000
Michy_Mouse	Michy_Mouse@a.com	hola@gmail.com	2012-11-23	70000
Luisete	luis	l@l.com	2012-11-25	20

Remove Row

Temporal Users - Drag and Drop to move users

NAME	PASS	EMAIL	ALTA	POINTS	ACT
milab	petia	radevap@gmail.com	2012-12-03	0	<input type="checkbox"/>

Remove Row

Edit latest news

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Aenean commodo ligula eget dolor. Aenean massa. Cum sociis natoque penatibus et magnis dis parturient .

Change news

Send mails - Mails for the admin. use ',' to separate mails.

darkwell@gmail.com

Change mails

Detalle de la imagen de login 27

Pasamos a detallar por partes las diferentes opciones:

- **Tablas de usuarios:** Desde aquí podemos editar todos los campos de los usuarios, y hacer la verificación de los usuarios temporal a los usuarios activados en la web. Para mover los usuarios simplemente los arrastraremos desde una tabla a otra. Con el botón de “remove row” Podemos eliminar los registros de los usuarios que no queramos, tanto desde los temporales hasta los que ya están verificados.
- **News:** Desde aquí se puede cambiar la sección de latest news que aparece al lado de las web. Este texto aparece a la derecha y es totalmente editable.
- **Mails:** Se introducirán los emails de los administradores. Todos los emails que se introduzcan aquí, separados por comas, serán avisados cada vez que algún usuario quiera validarse.
- **File Explorer:** La sección del file explore nos permite acceder a los ficheros subidos por los usuarios. Tanto a los ficheros directamente, como los ficheros comprimidos. Además nos permite la gestión de éstos, borrar, renombrar, eliminar, etc. Esta parte pedirá validación del usuario, que volverá a ser **ub, ub**, ya que también puede ser accedida desde fuera de la misma.

File Explorer

Subidas

[logout]

Upload files

Seleccionar archivo No se ha seleccionado ningún archivo

Upload

Max: 1.91 MB / 5000 MB

Create folder

Folder File From URL

Ok

Search files & folders

Search

Search only this folder and below

Files Sort by name ▾

◇ Gatetes	2 files	1
◇ PruebaSubida	1 files	1
◇ kiki	1 files	1
◇ pepito	1 files	1
◇ popo	2 files	1
◇ sube	1 files	1
◇ test	2 files	1

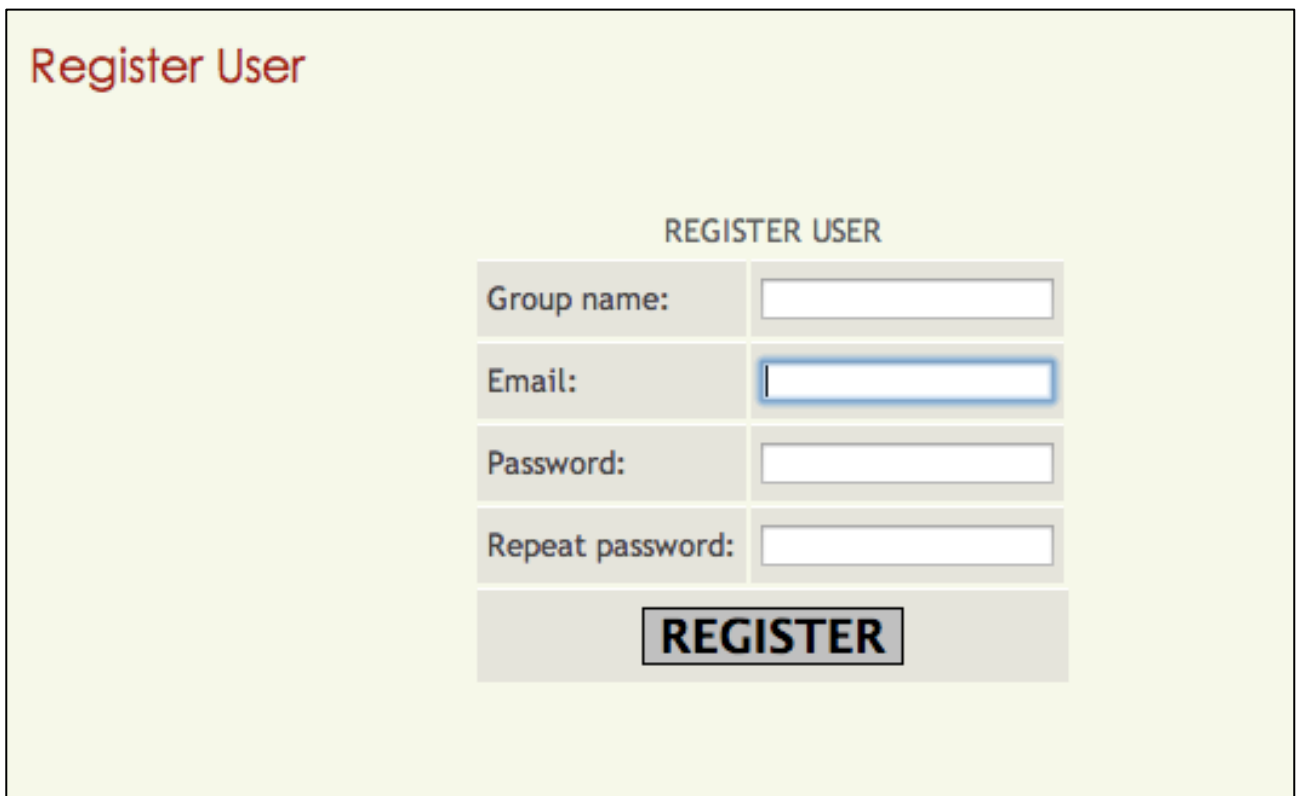
7 folders - 0 files

Detalle del file explorer 28

11. ANEXOS: Manual del usuario.

11.1 Registro

Para poder subir los ficheros a la aplicación, primero el usuario debe validarse, como hemos visto en los casos de uso. Para esto, el usuario se registrará en la web mediante la url una vez accedido, se mostrará lo siguiente:



The image shows a web form titled "Register User" with a sub-header "REGISTER USER". The form contains four input fields: "Group name:", "Email:", "Password:", and "Repeat password:". Below the fields is a "REGISTER" button. The "Email" field is highlighted with a blue border.

Imagen del registro 29

El usuario debe rellenar los siguientes campos con las especificaciones que siguen.

- Mínimo 4 caracteres solo caracteres alfanúmericos y underscore aceptado para el nombre. No debe estar usado en la base de datos.
- El e-mail debe tener formato correcto, sino no nos dejará registrar. Entendemos por formato correcto, palabra@palabra.dominio.
- Las contraseñas tienen que coincidir.

Una vez se ha completado el registro, se envía un correo a los administradores (configurado como hemos visto anteriormente, desde la parte del administrador)

11.2 Verificación del registro

La verificación del usuario, se realiza manualmente por parte del administrador, ya que es un requisito importante y que no puede ser automatizado. Una vez verificado por parte del administrador, se le envía un mail al usuario(en un intervalo de un máximo de 15 minutos desde que el administrador lo ha aceptado) informando que ya puede realizar la subida de sus data-sets.

11.3 Subida de los datasets

Los data-sets deben tener el formato que se indica en la web, sección “Evaluation”. Sino cumplen este requisito se informará al usuario, y se borrarán del servidor.

El formato es el siguiente:

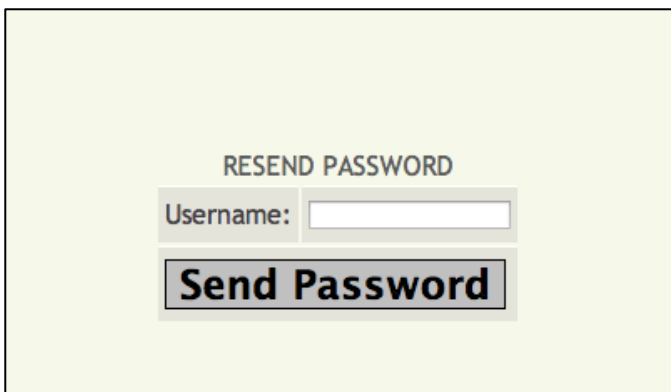
- Carpeta nombre del participante.
 - Carpeta Data_set_A
 - lum_frame_XX_YYYY_003.txt
 - ...
 - Carpeta Data_set_B
 - lum_frame_XX_YYYY_003.txt

▪ ...

Una vez subidos, se indicará al usuario que todo ha sido correcto y que puede ver sus resultados en la sección “Results” de la misma web. Una vez allí, puede comprobar la posición en la que está respecto a los demás grupos de investigación.

11.4 Pérdida de la contraseña

En el caso que el usuario pierda la contraseña con la que realizó el registro, en la misma pestaña de “Upload Files” puede clicar sobre el enlace que “Forgot password?”. Una vez pulsado el link, el password volverá a ser enviado a la misma dirección que se registro.

A screenshot of a web form titled "RESEND PASSWORD". The form is centered on a light green background. It contains a label "Username:" followed by a text input field. Below the input field is a button with the text "Send Password" in a bold, black font.

Detalle de la web forgot.php 30

Página intencionalmente dejada en blanco