

GESTOR DE DOCUMENTOS

ÍNDICE:

PREÁMBULO.....	2
INTRODUCCIÓN.....	2
INGENIERÍA DEL SOFTWARE II.....	3
DISEÑO DE BASE DE DATOS.....	5
SISTEMAS WEB.....	7
CONCLUSIONES.....	9

PREÁMBULO

El proyecto, un gestor de documentos, une tres asignaturas impartidas en la [UPV/EHU](#). Las cuales son:

- Diseño de Base de Datos (BD), impartida por Arantza Illarramendi.
- Ingeniería del Software II (IS II), dada por Jon Iturrioz y Alfredo Goñi.
- Sistemas Web (SW), impartida por José Ángel Vadillo y José Miguel Blanco.

José Miguel Blanco, profesor de SW, ha sido el principal promotor del proyecto.

INTRODUCCIÓN

El proyecto consiste en un gestor de documentos, esto es, un sitio web donde puedes subir tus apuntes, compartirlos con los demás, y descargarlos de tu interés.

Para ello se han definido diferentes roles:

- Usuario no registrado
- Usuario registrado
- Profesor
- Administrador

Para llevar a cabo el proyecto he utilizado [Symfony](#). Symfony es un framework, basado en el Modelo Vista Controlador, diseñado para optimizar el desarrollo de las aplicaciones web. Este framework está basado íntegramente en PHP y nació para la web (no fue una transformación desde otra plataforma).

A continuación, hablaré de las uniones de cada asignatura con el proyecto. Igualmente seguiré comentando sobre Symfony y sus posibilidades.

INGENIERÍA DEL SOFTWARE II

Aunque al principio pensaba que los conceptos de esta asignatura iban a ser los más difíciles de unir al proyecto, después de entender Symfony y su manera de trabajar me di cuenta que iba a ser al contrario.

En clase, estudiamos [Hibernate](#) y [Spring](#) (aparte de [JSF](#)). Estos son unos frameworks orientados a JAVA, que en teoría, te facilita la tarea de programar. Digo en teoría, porque al principio cuesta mucho entenderlos. Es más, antes de trabajar con Symfony, tenía muy mala imagen de estos. Seguramente porque no había descubierto todas las posibilidades que ofrecen.

Evidentemente no podemos comparar, entre ellos, estos frameworks, ya que cada uno cumple una función y se usan en diferentes plataformas (JAVA, PHP etc.). Aun así, hay varias cosas que los unen:

- Cuesta entender su funcionamiento, pero una vez entendido te facilita la programación.
- Te permite utilizar soluciones robustas ya implementadas, y no tener que implementarlas con los riesgos que acarrea.
- Tienen mucho soporte por detrás, tanto oficial (actualizaciones, soluciones de bugs, manuales etc.), como de usuarios “anónimos” (foros, blogs etc.).

Centrándonos en Symfony, me atrevería a decir que es “un todo”. Aparte de tener todo lo anteriormente dicho, hay que recalcar el muy buen soporte que tiene. Ya de por si el [manual oficial](#) en inglés es muy bueno (incluso con traducción gratuita al [castellano](#)) y además tienes muchos otros tutoriales, o consejos en Internet.

Por otro lado, en la asignatura hemos visto patrones de diseño y pruebas unitarias. En este sentido, Symfony tiene integrado otro framework destinado a las pruebas: [PHPUnit](#). Este, sí podríamos compararlo con [JUnit](#).

Aunque cada uno tiene sus particularidades (uno es para PHP, el otro para JAVA; formas de hacer los tests etc.), el fin es el mismo: puedes programar pruebas unitarias para comprobar la integridad del software.

En el caso de Symfony y PHPUnit, su uso es muy sencillo: Dentro de la carpeta “Tests” de cada Bundle¹, tendríamos que programar las pruebas que queramos hacer. Después solo ejecutar el siguiente comando en la consola:

```
$ phpunit -c app/
```

Realmente estas pruebas son muy sencillas, pero a la vez muy útiles. Esto solo ha sido un pequeño resumen de PHPUnit, pero Symfony, junto con PHPUnit, da muchas más posibilidades. Todas estas posibilidades las podemos encontrar en el [manual de Symfony](#), o en el propio [manual de PHPUnit](#).

¹“[Un bundle](#) no es más que un directorio que almacena todo lo relacionado con una función específica, incluyendo clases PHP, configuración, e incluso hojas de estilo y archivos de Javascript.”

DISEÑO DE BASE DE DATOS

Esta asignatura parece la más fácil de integrar en el proyecto. Y realmente es así. Hoy en día, no nos podemos encontrar ningún proyecto que no tenga una base de datos para persistir la información del proyecto y de los usuarios.

Tenía claro cómo iba a integrar la base de datos en el proyecto, hasta que apareció [Doctrine](#). Según iba leyendo el manual oficial de Symfony, descubrí el [apartado destinado a Doctrine](#).

En ese momento cambie totalmente de idea y me di cuenta que no tenía por qué ser la más fácil. Me encontraba ante otro “framework”. En este caso un mapeador de objetos-relacional (ORM). Ahí me di cuenta que la idea de hacer una BD totalmente ajena al proyecto no era buena.

Doctrine es una librería ajena a Symfony, pero al integrarla en el proyecto, simplifica el acceso y manejo de los datos de la base de datos. Además, no solo se centra en una BD ORM, sino que también tiene soporte para una arquitectura ODM (Object Document Mapping), mucho menos conocida.

Me hubiese gustado haber trabajado con ODM, y con [MongoDB](#)². Pero realmente, se escapaba del alcance del propio proyecto y del temario de la asignatura (solo damos base de datos relacionales).

Aun así, Doctrine abstrae el uso de sentencias SQL y permite migrar en cualquier momento de base de datos sin cambiar el código, ya que las consultas genera automáticamente la librería. Solo tendríamos que [configurar el tipo de BD](#) que es.

Doctrine es muy fácil de utilizar. Después de configurarla, tenemos que utilizar dos líneas para guardar un objeto (por ejemplo un usuario):

```
$em->persist($objetoAGuardar);
```

```
$em->flush();
```

² [Wikipedia](#): sistema de base de datos NoSQL orientado a documentos, desarrollado bajo el concepto de código abierto.

Con esas dos líneas se habría guardado el objeto. Como se puede apreciar, facilita mucho la persistencia de la información, al no tener que crear sentencias SQL.

Aun así, sí que [tendríamos que mapear](#) las clases PHP para que Doctrine pueda guardar automáticamente la información. La librería [admite relaciones](#) “muchos a uno”, “uno a muchos”, “uno a uno”, “muchos a muchos”...

Realmente, para comprender todo esto, tienes que saber la base, es decir, como diseñar una base de datos, ya que de otra forma costará mucho entender el porqué de hacer las cosas así. Por ejemplo, un campo multivalor se representaría en PHP como un vector y este se mapearía con una relación tipo “uno a muchos”.

SISTEMAS WEB

En esta asignatura, entre otras cosas, hemos visto la base de las tecnologías más unidas al proyecto: PHP, Javascript etc. Es decir, hemos visto lo básico de algunas tecnologías web.

Cuando empecé el proyecto tenía ganas de utilizar un framework. Pero me daba para atrás solo de pensar las experiencias con otros. En este caso empecé de manera diferente.

Después de tener la recomendación de algún compañero, empecé a leer la [documentación oficial](#) de Symfony (también en [castellano](#)). Fui poco a poco comprendiendo cada concepto, y me fui sintiendo a gusto con la herramienta.

Me di cuenta que utilizar un framework para construir una web no tenía nada que ver con lo que habíamos dado en clase. Es verdad, al igual que en otros casos, que para poder trabajar con Symfony tienes que tener un nivel básico de, por lo menos, PHP.

Antes de comenzar con el proyecto en sí, empecé con las típicas pruebas de “Hola Mundo” etc. Según iba entendiendo el framework, las pruebas cada vez eran más avanzadas.

En ese momento, tuve claro por qué utilizar Symfony y no hacer una página web desde 0.

Haber empezado el proyecto sin tener nada, hubiese significado tener que programar todas las funciones, incluidas las que conllevan más seguridad. Todo esto lo arregla Symfony. Este framework, nacido para la web, ofrece mucho en el tema de seguridad. Muchas de las cosas típicas (como el logueo), ya lo tienes implementado y además de una forma muy segura y clara. Eso no quiere decir que no tengas que hacer nada, siempre vas a tener que hacer parte (la lógica en sí): consultas a la BD, controlar los intentos de registro/logueo etc.

Si todo esto lo tendríamos que hacer a mano, seguramente tendríamos muchos fallos (¿Para qué hacer lo que ya está bien hecho?). Aun así, la comunidad de Symfony también se puede confundir y tener fallos, pero al ser tan grande esa comunidad, sacaran una actualización con bastante rapidez para solucionar ese bug. Sin olvidar que hay empresas que se dedican (pagando) a dar soporte personalizado a Symfony, en caso de necesidad.

Por otra parte, creo que es muy importante señalar que hay muchos desarrollos libres ([bundles de terceros](#)) que se pueden integrar de una manera muy sencilla en Symfony. Además, hay algunos que son muy útiles y a su vez, al tener tanta trayectoria, son muy robustos. Por ejemplo, FOSUserBundle³.

Para terminar, se puede decir, que aunque lo que hemos dado en clase no se utilice al 100% en el proyecto, que sí es imprescindible para poder entender y trabajar con Symfony. Y desde luego, recomendaría utilizar algún tipo de framework cuando se quiera hacer cualquier página web “decente” y con seguridad.

³ [Cita:](#) “Facilita la gestión de los usuarios en las aplicaciones Symfony2. Se encarga de realizar tareas muy comunes en las aplicaciones que manejan usuarios y que Symfony2 por el momento no soporta. Entre otras, simplifica el registro de usuarios (incluyendo el envío opcional de un email de confirmación) y la opción “olvido mi contraseña”. Compatible con Doctrine, Propel, y MongoDB/CouchDB.”

CONCLUSIONES

Aunque el principio fuese muy duro, en el sentido de que no sabía qué me iba a encontrar, y teniendo en cuenta que alguna que otra vez casi lo doy por imposible, tengo que reconocer que ha sido un proyecto que me ha gustado mucho.

Creo que este tipo de proyectos se tendrían que fomentar más a lo largo de la carrera, ya que es una experiencia que se acerca más a la vida real.

Durante la realización del gestor de documentos, he cambiado totalmente la idea de los frameworks, su utilidad y sus ventajas; dejando patente el inconveniente más grande: su aprendizaje.

También he podido descubrir que hay un mundo que no se estudia en la universidad sobre las bases de datos y su relación con el software.

Por otra parte, me ha ayudado a saber cómo afrontar un problema nuevo: como aprender a solucionarlo en un tiempo limitado y con la exigencia de un proyecto final. Además, me ha obligado a organizarme el tiempo de una manera diferente.

En definitiva, ha sido un proyecto que me ha hecho juntar la materia de las tres asignaturas implicadas, lo que conlleva terminar de entender la propia materia. Ha sido muy positivo.