

Desarrollo de aplicación web contra el acoso escolar.

Abel García Borrego

Resumen—Una de las mayores dificultades que nos encontramos hoy en día en el ámbito escolar es el acoso. Siendo una lacra social que debemos solucionar con la mayor prontitud posible. Este proyecto web nace con la idea de ayudar en el ámbito, siendo una aplicación web completa, donde el alumno puede hablar anónimamente con sus tutores escolares acerca del acoso, situaciones de peligro, obtener consejos y conocer distintos casos. El punto fuerte y enfoque de la aplicación es el asegurar el anonimato utilizando técnicas de ingeniería del software.

Palabras clave—Bullying, Acoso, Anonimato, Escuela, Instituto, Universidad, Php, Symfony, Responsive

Abstract—One of the biggest difficulties that we find today in the school environment is bullying. Being a social scourge we must solve as quickly as possible. This web project was born with the idea of helping in the field, being a complete web application, where the student can talk anonymously with his school tutors about bullying or unsafe actions, get advice and know different cases that may be similar. The strength and focus of the application is to ensure anonymity using software engineering techniques.

Index Terms—Bullying, Harrassment, School, High School, University, College, Php, Symfony, Responsive

◆

1 INTRODUCCIÓN

Este documento nace por la necesidad de documentar de forma completa el método de desarrollo del trabajo de fin de grado de Ingeniería Informática del proyecto *Desarrollo de aplicación web de apoyo a la lucha antiacoso escolar* en el segundo cuatrimestre del año académico 2016-2017.

La estructura de este documento se estructura de la siguiente manera:

1. Introducción
2. Objetivos
3. Contexto
4. Metodología y desarrollo
5. Resultados
6. Conclusiones i líneas futuras
7. Bibliografía

2 OBJETIVOS

La aplicación parte de la idea de ofrecer una ayuda social a través del trabajo de fin de grado. Una de las mayores lacras sociales es el acoso escolar siendo una oportunidad muy positiva utilizar conocimientos de ingeniería del software para ayudar a solucionar el problema.

La idea original era desarrollar una aplicación móvil, pero tuvimos en cuenta que en los colegios sería más útil una aplicación web, ya que los tutores les sería más fácil visualizarlo en una pantalla grande y organizada.

Para no perder las ventajas de un dispositivo móvil, decidimos que la web tendría la necesidad de ser responsive, para poderse visualizar en todos los navegadores comerciales posibles, tanto de sobremesa como de dispositivo móvil o Tablet.

Era interesante y fue considerado como objetivo clave era el garantizar el anonimato del infante, ya que consideramos que si no tiene miedo a represalias, le es más fácil comunicar en caso de sufrir o conocer algún acoso. También era muy necesario que el interfaz fuese sencilla y amigable, y que su gestión de los recursos fuese buena y calculada.

Finalmente, un objetivo primordial era hacer una buena planificación, lo que nos permitiera poder entregar el proyecto siguiendo los terminios, de una manera completa y óptima.

-
- E-mail de contacto: Abel.garciab@e-campus.uab.cat
 - Menció realizada: *Enginyeria del Software*
 - Trabajo tutorizado por: *Marc Tallo Sendra*
 - Curs 2016/17

3 CONTEXTO

Existen en el mercado varias aplicaciones web que comparten lucha con la nuestra. Durante el desarrollo del proyecto se han revisado varias, tanto en búsqueda de ideas funcionales, como para mejoras futuras.

En este apartado describiré unas cuantas.

3.1 StopIt

Se trata de una aplicación Estadounidense donde su lema es que algo proactivo es mejor que algo reactivo.

Su línea de actuación es:

- Bullying y Cyberbullying.
- Relaciones inapropiadas entre el estudiante y profesor.
- Intentos de violación.
- Violencia física.
- Posesión de armas.
- Narcotráfico.
- Tolerancia y discriminación.

Esta web tiene una manera de funcionar muy comercial, siguiendo mucho la legislación estadounidense.

Nuestra aplicación podría ser útil también para afrontar los problemas marcados, pero no consideramos un riesgo grande de posesión de armas o narcotráfico real en nuestro contexto de país.

Creemos más en una solución al Bullying, discriminación y violencia física.

3.2 Anti-Bullying Alliance

Se trata de una web del gobierno de Reino Unido. En esta aplicación web, se listan consejos, ayuda legal para la persona acosada y se aceptan donativos para apoyo por cursos legales.

El enfoque que da esta aplicación es maximizar la prevención de ataques homofóbicos, bifóbicos y racistas.

3.3 Bully Tap

Se trata de una aplicación creada por una compañía neoyorkina, llamada **Sterko**. Los afectados pueden denunciar a los matones, contar sus historias en un foro y chatear en vivo con asesores certificados.

El software pretender no ser solo una vía de expresión o desahogo, sino que va bastante más allá. Todas las notificaciones de las víctimas son enviadas a sus respectivas escuelas. Si los centros no manejan los avisos de acoso en el momento oportuno o directamente los ignora por completo, los responsables de Bully Tap cursarán una demanda civil en contra de la escuela o acosado

Siendo una solución muy directa y dinámica. Nuestra aplicación no se acercará a estas funciones, ya que las consideramos muy radicales y preferimos la autogestión de los centros.

4 METODOLOGÍA Y DESARROLLO

Todo proyecto debe seguir una buena planificación y estudio del entorno, en este apartado detallamos la metodología aplicada, así como su planificación, elección del entorno, framework y un cálculo de costes asociados al proyecto.

4.1 Planificación temporal

| Nombre de la tarea | Duración | Inicio | Finalizar |
|---------------------------------|----------|----------|-----------|
| Fase 0: Reunión | 6d | 17/02/17 | 24/02/17 |
| Fase 1: Informe Inicial | 7d | 24/02/17 | 06/03/17 |
| Fase 2: Diseño e implementación | 56d | 05/03/17 | 19/05/17 |
| Fase 3: Test y corrección | 10d | 19/05/17 | 01/06/17 |
| Fase 4: Validación y entrega | 8d | 01/06/17 | 12/06/17 |

Figura 1. Cronograma de fases

Todas las fases iniciaban y finalizaban siguiendo los terminos asignados por la tutoría del proyecto.

La primera fase la utilizamos para familiarizarnos con el tutor asignado, empezar a hacernos una idea de lo que intentamos desarrollar y empezar a documentarnos.

En la segunda fase entregamos un informe inicial, donde documentábamos el contexto del proyecto, la metodología escogida y requisitos funcionales.

En la tercera Fase ya empezaba el desarrollo de la aplicación, esta etapa es la más larga ya que tiene casi todo el contenido del proyecto en ella.

Una cuarta fase fue destinada para el testeo y corrección de la aplicación.

La última fase es dedicada para la validación y entrega del proyecto, para la redacción del informe final.

4.2 Metodología de programación y framework

En los objetivos y para facilitar la compatibilidad web-móvil sugeríamos que la página web fuese responsive, inicialmente se intentó darle esta funcionalidad a través de librerías propias de HTML5 y CSS, pero estaba siendo un proceso muy costoso, por lo que optamos finalmente por desarrollar con el framework Bootstrap, siendo el más utilizado para el desarrollo de webs responsive y nos proporciona funcionalidades extras deseables como Drag&Drop en los siguientes apartados seguiremos describiendo este framework

También originalmente se planteó la idea de desarrollar la aplicación utilizando algún lenguaje web desde cero, pero se descartó la idea ya que el objetivo era desarrollar un proyecto usable, por lo que se hizo un estudio de Frameworks y decidimos usar el más popular: Symfony.

Symfony es un Framework OpenSource PHP bajo la licencia MIT. Las ventajas de este Framework es su grandísima comunidad de desarrolladores, se profundizará en él en el siguiente apartado.

El desarrollo de la aplicación ha sido incremental-espiral, trabajando en cada módulo independientemente y posteriormente testeando que se integraran bien en el conjunto.

4.2.1 Modelo Vista Controlador

Si se desea desarrollar proyectos web de una manera cómoda, limpia e inteligible es aconsejable utilizar el Modelo Vista Controlador (MVC).

Se trata de diferenciar y separar las distintas partes para delegar responsabilidad.

Cuando hablamos de Modelo nos referimos a todo el conjunto de información que el sistema desarrollado dispone, también llamado 'Lógica de negocio'. La petición de acceso es gestionada por la siguiente parte, conocida como Controlador.

El Controlador es el responsable de interactuar con el usuario gestionando las peticiones del usuario y pidiéndoselas al modelo, es el encargado de hacer las operaciones intermedias y enviárselas a la Vista.

La Vista se trata de la última parte del MVC, se encarga exclusivamente y obligatoriamente de mostrar los datos del Modelo procesadas por el Controlador, idealmente y por seguridad no debería hacer ningún procesamiento ni ejecución de código dinámico.

4.2.2 Symfony3

Symfony3 es la evolución natural de Symfony2, se trata de un framework PHP muy conocido y utilizado, sigue el concepto de MVC comentado anteriormente, la diferencia con la versión anterior es a la hora de ejecutar el servicio, tiene pequeños cambios en los comandos.

Otra diferencia importante es que se han reescrito muchas funciones predeterminadas y hecho cambio en su sistema de formularios.

4.2.2.1 Seguridad y rutas en Symfony3

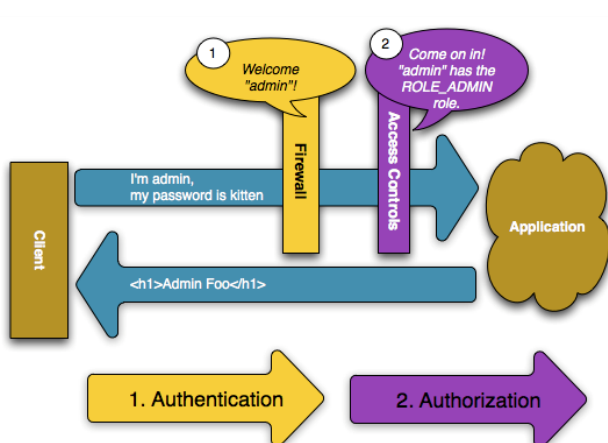


Figura 2. Roles, autenticación y autorización en Symfony3

Symfony3 dispone de un sistema completo de autenticación y autorización de roles. En nuestra aplicación esta

función es crítica, por tanto le hemos dado un uso muy intensivo.

Como vemos en la figura 2, al intentar acceder a una ruta concreta nos encontramos con un módulo llamado Firewall propio del entorno, este es el encargado de gestionar la autenticación de usuario creando una sesión a través del formulario de inicio de sesión, concediéndole los privilegios y parámetros recibidos en la base de datos. Es importante comprender que el Firewall simplemente otorga los datos creando una sesión, no interviene en el otorgar o denegar el acceso. El modulo Firewall es completamente configurable desde el archivo de control 'security.yml' pudiendo configurar por ejemplo si dejamos que un usuario anónimo tome datos de sesión.

El encargado de gestionar los accesos es el modulo "Access Controls", también configurable desde el archivo 'security.yml'. Este módulo es capaz de conocer el Rol del usuario y redirigirnos acordeamente.

El utilizar estos módulos de seguridad nos permite simplificar muchísimo el desarrollo y el nivel de detalle que el usuario debería ser capaz de ver.

La función propia de symfony más usada y cómoda para gestionar el control de roles es: `is_granted('ROLE')` devolviéndonos un booleano y mostrando al usuario la información acorde su perfil.

En el Apéndice A1 podemos ver un fragmento del fichero de rutas de nuestra aplicación, donde tenemos configurado que toda ruta hija de `/admin/` requiera tener un rol administrador. En el Apéndice A2 podemos ver la respuesta que nos devuelve el sistema en caso de intento de acceso como anónimo a la ruta `/admin/user` encargada de gestión de usuarios.

4.2.2.2 Entidades e interacción con la base de datos (Modelo).

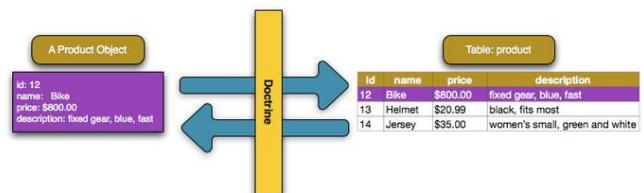


Figura 3. Mapeo de base de datos, conversión a Entidad

Symfony3 dispone de un módulo conocido como Doctrine, se trata de un ORM (del inglés Object-Relational mapping) siendo un módulo que permite realizar la conversión desde un objeto relacional que requiere persistencia (por ejemplo una tabla de una base de datos) hasta un tipo de datos usable en el desarrollo orientado a objetos, llamando a cada tabla convertida Entidad (Entity).

De esta manera y de una manera muy sencilla y rápida tenemos una aproximación a una base de datos orientada a objetos. Proporcionándonos las ventajas de la herencia y el polimorfismo.

El mismo Doctrine es el encargado de hacer la conversión hacia ambos sentidos, haciendo las interacciones con la base de datos muy sencillas.

Una de las decisiones en tecnología que tuvimos que tomar al desarrollar el proyecto, era si utilizar bases de datos relacionales (MySQL) o no relacionales (MongoDB), el hecho de que Symfony nos proporcionara las herramientas para trabajar fácilmente con bases de datos relacionales sumado a que priorizamos la seguridad e integridad de los datos sobre su velocidad de transacciones nos hicieron decidimos por MySQL.

Hay que prestar especial atención en no utilizar palabras reservadas en el procesador de la base de datos, ya que Doctrine puede darlos por válido e intentar procesarlas, dando un error dentro del procesador que puede causar errores de difícil solución.

Symfony3 tiene integrado dentro de Doctrine diferentes funciones de búsqueda, inserción, filtro, eliminación y actualización, no obstante si se quiere hacer una consulta avanzada y con alto grado de complejidad es capaz de gestionar la petición en SQL puro.

Doctrine también es capaz de gestionar modelos no relacionales, como por ejemplo MongoDB, con el modulo llamado ODM, pero no profundizaremos ya que en nuestro proyecto no ha sido utilizado ningún modelo no relacional.

4.2.2.3 Controlador

En Symfony3 el controlador toma mucha relevancia, ya que es el encargado de ejecutar, procesar y redirigir las peticiones del usuario.

Por definición en Symfony3 cada controlador es una clase independiente asignado a una ruta, con sus diferentes funciones.

Cada función hace el procesamiento de datos y se envía a la vista como parámetros retornados.

```
public function indexAction(Request $request)
{
    $em = $this->getDoctrine()->getEntityManager();
    $entry_repo = $em->getRepository("proyectoBundle:User");
    $usuarios = $entry_repo->findAll();
    //Devuelve todos los usuarios en la base de datos.
    $totalItems = count($entry_repo);
    //Cuenta el numero de usuarios, para evitar hacer conteos en la vista
    return $this->render("proyectoBundle:User:index.html.twig", array(
        // Devuelve a la vista la lista de usuarios y su totalidad, lista para mostrar.
        "usuarios" => $usuarios,
        "totalItems" => $totalItems,
    ));
}
```

Figura 4. Ejemplo de listado de usuarios en el controlador.

En la figura 4 podemos ver un pequeño ejemplo de nuestro proyecto de la función de listado de usuarios.

Doctrine nos hace la conexión y conversión a la base de datos persistente, posteriormente aplicamos un filtro de búsqueda, guardándolo en una variable. Adicionalmente hace un conteo de usuarios totales, para evitar que la vista haga ninguna ejecución. Finalmente lo envía a la vista,

completamente procesado y preparado para mostrar.

El controlador en Symfony3 es el responsable de gestionar los errores que puedan ocasionarse, sobre todo los errores de página no encontrada y posibles errores con el servidor.

4.2.2.4 Vistas en Symfony3 (Vista)

Las vistas en Symfony3 son desarrolladas con el método de Plantillas.

Una plantilla es simplemente un texto generado desde cualquier formato basado en texto, por ejemplo HTML, XML, CSV, LaTeX.

Al estar el proyecto enfocado en el PHP lo natural sería utilizar lo conocido como Plantilla PHP, siendo un texto procesador como PHP pero conteniendo pequeños textos planos, pero esto resulta confuso, y es por ello que Symfony3 dispone de un lenguaje propio de plantillas llamado Twig.

La filosofía de Twig es crear plantillas altamente leíbles con características muy amigables para los diseñadores web, usualmente menos habituados a la programación lógica-funcional.

Con una alta probabilidad, un proyecto web comparte muchísimos elementos en común entre sí, como cabeceras, pie de páginas, menús y estilos.

Al igual que una clase PHP tiene herencia entre plantillas, que te permite crear un bloque base, conocido como 'layout' conteniendo todos los elementos comunes, y modificando sobrescribiendo, similar en las subclases PHP.

El concepto equivalente a clase PHP en Twig se le conoce como *block*.

Por características de Twig está pensado para no procesar etiquetas PHP, ya que está diseñado para generar única y exclusivamente presentación.

Sus características más destacables y que le diferencian del Plantillado PHP, es la capacidad de generar filtros que solo afectarán a la vista, asegurándonos que ni el Modelo ni el Controlador pueden ser modificados, dándonos una doble seguridad a los datos.

Una característica clave y beneficiosa del sistema Twig es que utiliza clases nativas del PHP y utiliza el caché, dando un sistema muy rápido y estable. En principio es un proceso automático y listo para ser usado sin gestionarlo directamente, pero durante el desarrollo de nuestro proyecto hemos tenido que gestionar el caché manualmente, ya que al realizar tantas modificaciones en tan poco tiempo en modo de reparación de fallos, el caché le costaba bastante tiempo actualizarse y daba grandes usos de memoria.

Una vez que el sistema ya no sufría modificaciones funcionales importantes y constantes, los recursos se volvieron estables y reducidos, se puede apreciar en el apéndice A3 que la carga total del sistema es de tan solo unos 12MB.

4.2.2.5 Bootstrap

Bootstrap es un conjunto de herramientas lanzada por Twitter en mitades del 2011, es de código abierto.

Es muy usado porque proporciona un gran abanico de plantillas de tipografías, formularios, menús de navegación todos utilizando HTML y CSS.

Entre sus funciones más destacables están el ser totalmente *responsive* por lo que es el sistema indicado para complementar el apartado de Vistas en Twig proporcionadas por Symfony. Para incluir Bootstrap solo hay que incluir un par de líneas de inclusión de código en el apartado de Plantillas (layout) de Symfony, estas líneas se pueden encontrar en la web oficial de Bootstrap y su uso posterior es muy sencillo, ya que se integran perfectamente con cualquier programa IDE de desarrollo.

4.2.2.5.1 Ampliación de vistas Twig aplicando Bootstrap

En apartados anteriores habíamos visto la gran versatilidad y utilidad que nos proporciona Twig, excepto porque su sistema de plantillado es dificultoso de tratar como *responsive*. Para solucionar este inconveniente utilizamos Bootstrap, es totalmente compatible con Symfony y Twig.

Bootstrap es ideal para generar formularios visuales, ya que es la parte que tiene más problemas visuales generalmente de un sistema web.

También dispone de una muy buena gestión de las alertas, siendo unas alertas muy dinámicas y visualmente agradables.

En el Apéndice A6 podemos ver un ejemplo de nuestro formulario de registro, dándonos una alerta verde dinámica indicándonos que el registro ha sido satisfactorio.

4.3 Requisitos del proyecto

La prioridad del proyecto es garantizar el anonimato del alumno, si así lo desea, por eso la toma de requisitos se toma con esa idea principal.

La siguiente tabla nos muestra a grandes rasgos los requisitos principales junto a su grado de criticidad.

| Requisito | Criticidad/Prioridad |
|---|----------------------|
| R1 - Interfaz amigable | Media |
| R2 - Responsive | Alta |
| R3 - Asegurar el anonimato del alumno. | Máxima |
| R4 - Sistema seguro, prevención de ataques y accesos restringidos | Máxima |
| R5 - Rendimiento adecuado | Media |

Siendo la interfaz amigable una prioridad media, ya que aunque es un resultado muy deseable, no es la prioridad en este proyecto concreto.

La *responsiveness* del sistema tiene una prioridad alta, ya que es una de las condiciones por las que nos decantamos por hacer un proyecto web en vez de una aplicación Android nativa.

La seguridad del sistema y la garantía del anonimato es nuestra prioridad más alta.

El rendimiento será gestionando utilizando buenas prácticas de Modelo Vista Controlador

4.4 Entorno de desarrollo

Al desarrollar un proyecto es muy importante conocer los recursos de que dispones con el fin de estudiar si es viable el desarrollo del proyecto computacionalmente.

A su vez también es importante evaluar los recursos software que utilizamos. En esta sección se detallará sobre los recursos que contábamos para el desarrollo del proyecto.

4.4.1 Entorno Hardware

Nuestro proyecto ha sido desarrollado utilizando un equipo:

- Windows 10 PRO
- Intel Core i3 510 @ 2,93 GHZ
- RAM 8GB 1600 MHz
- Disco duro Solido de 256 GB
- Conexión a internet 30mbps/3,5mbps

4.4.2 Entorno Software

Nuestro proyecto ha sido desarrollado utilizando los siguientes recursos Software.

PhpStorm

Idle de PHP que se integra increíblemente bien con Symfony. Nos permite configurar visualmente la base de datos, para hacer pruebas de conexiones. Es un idle muy amigable que utiliza el motor IntelliJ IDEA muy usado por ejemplo en Android Studio. Un inconveniente es que es una aplicación de pago, pero la facilidad con la que ayuda a desarrollar puede justificar bastante su precio.

En nuestro entorno solo hemos hecho una modificación al IDLE estándar y es la inclusión del Pluggin de terceros para Symfony3, esta pequeña modificación nos permite una integración prácticamente completa con el Framework, llegando a incluso darnos consejo de estilo y de buenas prácticas.

XAMPP

Es una aplicación muy potente y mayormente conocida. Sirve para correr una instancia de LAMP (Linux Apache MySql Php) sobre Windows. Consume muy pocos recursos y es muy estable.

Su configuración base es automática, pero nos permite hacer todo tipo de configuraciones avanzadas con gran facilidad, ya que cuenta con el gestor de base de datos PhpMyAdmin. En nuestra configuración hemos dejado los puertos predeterminadamente, la única modificación que hemos hecho es darle más recursos a los parámetros de MySql, ya que por norma está limitado para evitar bases de dato demasiado grandes y no tener problemas de inconsistencia.

4.5 Coste de desarrollo del proyecto

Es muy útil realizar un cálculo de presupuesto en el desarrollo de un proyecto, ya que nos permite evaluar la viabilidad económica y de los recursos que tenemos.

Las horas de trabajo del desarrollador humano se han estimado en veinte euros por hora. El coste de equipo se ha prorrateado a vida útil del equipo estimado, siendo estos doscientos cincuenta días laborales de media multiplicados por tres años naturales y ocho horas de trabajo por día laboral.

Finalmente se ha estimado dos donaciones económicas para estimular el software libre, una al FrameWork PHP Symfony y otra a la suite de aplicaciones Xampp.

Para simplificar tanto el cálculo, como la visualización de los datos, todos los precios están redondeados al euro.

4.5.1 Coste en personal

En este subapartado calculamos el precio del personal humano, es una estimación de las horas dedicadas a cada parte durante el transcurso del proyecto.

| Concepto | Horas | Precio | Total |
|---------------------------------|-------|--------|---------|
| Análisis de requisitos y diseño | 150 | 20 €/h | 3000€ |
| Desarrollo | 400 | 20 €/h | 8000€ |
| Test | 65 | 20 €/h | 1300€ |
| Total | 605 | | 12.300€ |

4.5.2 Coste en equipamiento

En este subapartado hacemos una estimación de coste del equipamiento al que proveemos para tal de desarrollar el producto, como hemos indicado en la introducción del apartado se ha prorrateado a las horas de vida útil del equipo.

4.5.2.1 Hardware y periféricos

| Concepto | Horas de uso | Precio del equipo (€) | Vida útil en horas | Precio por hora (€) | Total (€) |
|-------------------------|--------------|-----------------------|--------------------|---------------------|-----------|
| Pc I3 + ratón y teclado | 605 | 900 | 6000 | 0.15 | 91 |
| Pantalla LG | 605 | 250 | 6000 | 0.04 | 25 |
| Total | | | | | 116 |

4.5.2.2 Software

| Concepto | Horas de uso | Precio de la licencia (€) | Vida útil en horas | Precio por hora (€) | Total (€) |
|------------|--------------|---------------------------|--------------------|---------------------|-----------|
| PhPS-torm | 605 | 199 | 2000 | 0.10 | 61 |
| Windows 10 | 605 | 299 | 6000 | 0.05 | 30 |
| Symfony | 605 | 50 (*) | | | 50 |
| XAMPP | 605 | 50 (*) | | | 50 |
| Total | | | | | 191 |

Nótese que la vida útil del software PHPStorm es de 2000 horas, ya que se trata de una licencia anual.

Los dos precios de licencia anteriormente marcados como (*) se refieren a una donación libre a ambas fundaciones de Software libre por el uso de su producto, no teniendo vida útil limitada.

4.5.3 Otros

Otros gastos derivados del proyecto son el coste eléctrico, calculado como 605h de proyecto * 0,94kwh que usa nuestro equipo * 0,14€/kwh de la red eléctrica, siendo el total de 81€.

Otro gasto derivado imputable al proyecto son 150€ de un curso online de Symfony3 que recibí para entender mejor el sistema, y se le aplica concepto de formación al desarrollador.

4.5.4 Total y conclusiones del presupuesto

El total acumulado del proyecto asciende a 12.838€. Siendo la parte más destacable el coste en el personal con un 95% del valor total. Esta cifra era un valor esperado, ya que no hemos utilizado equipos con grandes rendimientos ni hemos tenido que pagar licencias de software grande. En un ámbito empresarial es muy probable que nos encontráramos que el coste proporcional en software sea más grande.

Una parte de este coste de software tan reducido es que hemos utilizado alternativas libres y gratuitas.

En el apartado de equipos, ha sido también reducido porque hemos reaprovechado equipamiento hardware ya existente.

Considero que el coste estimado es realista y realizable, y entra dentro de los valores de precio de desarrollo de aplicaciones técnicas.

5 RESULTADOS

En los requisitos nos marcábamos los siguientes objetivos:

| Requisito | Criticidad/Prioridad |
|---|----------------------|
| R1 - Interfaz amigable | Media |
| R2 - Responsive | Alta |
| R3 - Asegurar el anonimato del alumno. | Máxima |
| R4 - Sistema seguro, prevención de ataques y accesos restringidos | Máxima |
| R5 - Rendimiento adecuado | Media |

R1 y R2 se han resuelto sencillamente aplicando el mismo método, el uso de Bootstrap que nos proporciona a la vez *responsiveness* como una interfaz amigable y adecuada.

Para el requisito R3 se ha optado por dejar como opcional el introducir correo electrónico y nombre personal, todo el proceso de registro se procesa a través de un nombre de usuario elegido por el alumno.

A lo largo del sistema no se muestra en ningún punto nada que no sea el nombre de usuario. El usuario debe ser consciente de que a través del nombre de usuario y los comentarios que haga se pueden llegar a reconocer, por ello se ha habilitado la opción de comentario anónimo donde ni siquiera se muestra el nombre de usuario.

El requisito R4 ha sido el que más dificultades ha traído hasta que se ha aprendido a manejar correctamente el módulo de seguridad de Symfony. Para la seguridad de los datos del usuario la contraseña se ha encriptado con el algoritmo ByCript con un tamaño de clave de 184bits con un Salt en el caso de test de 4, pero es conveniente mantener este dato secreto para garantizar la seguridad del sistema.

La elección del algoritmo de cifrado de contraseña ha sido motivado a que tanto PHP como Symfony tienen funciones integradas específicas, lo que hacen un cálculo muy seguro y rápido.

En el anexo A4 podemos observar en el campo password como la contraseña está correctamente cifrada.

Debido a la gran versatilidad de las rutas de Symfony y que todo el sistema está integrado con autenticación por roles, podemos considerar que el sistema es seguro y robusto.

Una de las ventajas de usar el sistema de seguridad de Symfony es que en caso de actualización o detección de error en los mecanismos de seguridad la comunidad es muy probable que haga un parche rápido, por lo que tendríamos un sistema actualizado y seguro de nuevo.

Para verificar si cumplimos el requisito R4, revisamos el anexo A3 donde la consola Symfony nos informa:

- La página ha tardado en procesarse totalmente 624ms (Controlador).
- Ha usado 12.2MB de memoria RAM total.
- La carga de la vista ha sido de solo 17ms (Vista)
- Para cargar los datos de la base de datos (Modelo)
 - o Se han necesitado 2 consultas/peticiones
 - o El tiempo total para resolverlas ha sido de 2.62ms

Son unos recursos y tiempo muy buenos, en parte gracias al sistema de caché que dispone Symfony detallados en el apartado 4.2.2.4

6 CONCLUSIONES I LÍNEAS FUTURAS

Tras la finalización del proyecto considero que su realización ha sido una experiencia muy positiva y productiva a nivel personal. Se ha conseguido un nivel de desarrollo usable y amigable para el usuario.

Estoy satisfecho con el rigor en la privacidad y seguridad que se ha llevado en el proyecto, me ha permitido aprender mucho sobre el desarrollo sobre framework.

El acoso escolar es algo que se debe combatir, cada día leemos noticias acerca de ello, es un proceso que provoca mucho miedo e impotencia al afectado.

Es por ello que a través de este trabajo de fin de grado, he priorizado darle la tranquilidad al alumno y asegurarle que la aplicación no revelará su identidad. Esto se ha conseguido trabajando con contraseñas cifradas en la base de datos y con no obligar a registrarse con correo electrónico.

De cara a una mejora futura y ampliación del proyecto, sería positivo desarrollar:

- Aplicación nativa para sistemas móviles
 - o Apoyo con el GPS
 - o Botón de socorro
 - o Fotografía al momento

- Mejora del sistema de foro
 - o Añadir más funciones desde el navegador
- Mejora general de rendimiento, debido a que soy novato en Symfony3 se podría exprimir bastante más los rendimientos.
- Mejoras en el interfaz visual, ya que le he dado poca prioridad es mejorable fácilmente.

Agradecimientos

Quiero agradecer a mi tutor de proyecto, Marc Tallo, por su excelente guía durante el transcurso del proyecto y por sus consejos y ayuda en el enfoque de éste.

A Andrea Ros Rivas, por darme la idea inicial de crear una aplicación contra el acoso en las instituciones educativas.

A Pol Bonastre, por sus incansables horas de consultas sobre buenas prácticas de desarrollo durante todo el proceso del proyecto.

Y, sobre todo, agradecer a mi familia y amigos por todo el apoyo anímico que me han proporcionado a lo largo de este proyecto y durante el transcurso de la carrera universitaria.

BIBLIOGRAFÍA

- [1] <http://symfony.com/doc/current/security.html>
- [2] <http://getbootstrap.com/components/#top>
- [3] <http://symfony.com/doc/current/components/security/authorization.html>
- [4] <http://symfony.com/doc/current/components/form.html>
- [5] <http://symfony.com/doc/current/components/config.html>
- [6] <http://www.mecd.gob.es/educacion-mecd/mc/convivencia-escolar/recursos/enlacesinteres/acoso-escolar.html>
- [7] <https://symfony.com/doc/current/doctrine.html>
- [8] <http://www.apachefriends.org/en/xampp.html>
- [9] https://en.wikipedia.org/wiki/Software_development_process#Spiral_development
- [10] <http://www.mytechlogy.com/IT-blogs/11156/top-5-responsive-frameworks-for-2016-compared/#.WQqaVKIIGUk>
- [11] <http://blog.meanjs.org/post/76726660228/forking-out-of-an-opensource-conflict>
- [12] <http://stopitsolutions.com/>
- [13] <https://www.anti-bullyingalliance.org.uk/>

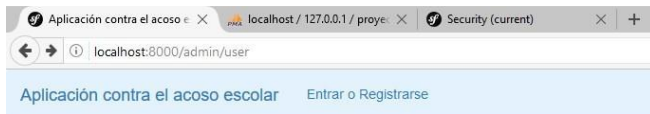
APENDICE

A1. TABLA DE RUTAS

```

routing.yml x  UserController.php x  User.php x  Conversacion.php x
Document 1/1
1 login:
2   path: /login
3   defaults: { _controller: proyectoBundle:User:login }
4
5 login_check:
6   path: /login_check
7
8 logout:
9   path: /logout
10
11 proyecto_homepage:
12   path: /proyecto
13   defaults: { _controller: proyectoBundle:Default:index }
14
15 admin_user:
16   path: /admin/user
17   defaults: { _controller: proyectoBundle:User:index }
18
19
20 admin_user_delete:
21   path: /admin/user/delete/{id}
22   defaults: { _controller: proyectoBundle:User:delete }
    
```

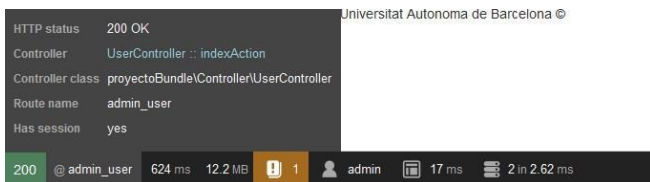
A2. INTENTO DE ACCESO DENEGADO



Acceso denegado, debes iniciar sesión como administrador.

Aplicación contra el acoso escolar - Abel García 2017 - Universitat Autònoma de Barcelona ©

A3. CONSOLA DE ESTADO DE SYMFONY



A4. ENTRADA EN LA BASE DE DATOS CON CONTRASEÑA ENCRIPTADA

| user_id | nickname | name | email | password | role |
|---------|-------------|-------------|------------------------|--|-----------|
| 6 | test_alumno | test_alumno | test_alumno@alumno.com | \$2y\$04\$EY3NEACTDg4Fgnc7HRqG2nk5C3pWWEwOYwYXZD | ROLE_USER |

A5. FORMULARIO Y ALERTA BOOTSTRAP

Regístrate

El usuario se ha creado correctamente

Nombre de usuario

Nombre

Contraseña

Email