

TRABAJO FIN DE CARRERA

Aplicaciones web para trabajo colaborativo

Aplicación web para la gestión de incidencias en entornos virtuales

Autor

Daniel Castro Larefors

Consultor

Ferran Prados Carrasco

Agradecimientos

Quiero dar las gracias a mi familia y pareja que han estado aguantando durante varios años las tardes y fines de semana de estudio, entrega de PACs y preparación de exámenes. A mis compañeros de trabajo de los que uno nunca deja de aprender. A los consultores de la UOC que ha sabido resolver mis dudas. Y a las comunidades de programadores y paginas especializadas que amenizan mis mañanas leyendo sus noticias y de las que nunca dejo de absorber información.

1.- Resumen

El portal web de la UOC, debido a la cantidad de usuarios entre estudiantes y consultores que hacen uso del portal cada día, se enfrenta como otras empresas y portales de servicios, con la necesidad de ofrecer una atención al usuario para poder resolver dudas e incidencias de forma rápida, clara, y sencilla.

En este trabajo de fin de carrera, se nos da la oportunidad de realizar una herramienta que sirva para que los usuarios del campus virtual puedan comunicar sus incidencias y consultas al departamento adecuado que está capacitado en resolverla.

Aunque esta herramienta ya existe actualmente en la UOC, es un tipo de aplicación que siempre se puede mejorar en función de experiencias e ideas que hayan podido usarse en otras herramientas similares, por lo que trataremos de conseguir un resultado óptimo y cumpliendo los objetivos. Los cuales en pocas palabras serían que un usuario (consultor o estudiante) sea capaz de abrir una incidencia, que esta sea dirigida al departamento que resolverá la incidencia y que el usuario pueda conocer la respuesta a su incidencia.

2.- Índice

1.- Resumen	3
2.- Índice	4
3.- Introducción	6
3.1 Justificación del TFC y contexto en el que se desarrolla	6
3.2 Objetivos	7
3.3 Motivación	8
3.4 Enfoque y método seguido	9
3.5 Planificación de tareas	10
4.- Análisis	12
4.1 Requisitos no funcionales	12
4.2 Casos de uso	13
4.2.1 Actores	13
4.2.2 Guiones	13
4.2.3 Documentación formal, Diagrama de casos de uso	13
4.2.4 Documentación textual, descripción textual	14
4.3 Modelo ER	21
5.- Diseño	22
5.1 Requisitos funcionales	22
5.2 Diagrama de actividad	23
5.3 Diseño de interfaz	24
5.3.1 Vista de usuario	24
5.3.2 Vista de Gestor	27
5.3.3 Vista de Administrador	28
5.4 Diagrama de clases	28
5.4.1 Clase Users	28
5.4.2 Clase Departments	28
5.4.3 Clase Tickets	29
5.4.4 Clase Answers	29
5.4.5 Clase Questions	29
6.- Conclusiones	30

6.1 Síntesis.....	30
6.2 Nivel de consecución.....	31
6.3 Futuras mejoras	32
6.4 Instalación y pruebas.....	33
7.- Glosario	34
8.- Referencias.....	35

3.- Introducción

3.1 Justificación del TFC y contexto en el que se desarrolla

El trabajo de fin de carrera está vinculado con las aplicaciones web destinadas a ser herramientas colaborativas, y en concreto de las propuestas existentes se ha elegido la opción de aplicación web para gestión de incidencias en entornos virtuales.

Las aplicaciones de gestión de incidencia para servicios donde los usuarios son muy numerosos y en la mayoría de los casos no es necesaria una respuesta inmediata se utilizan muy a menudo en páginas web, existen diferentes opciones comerciales y de código abierto disponibles en internet para poder instalar. En este proyecto se tratara de hacer una herramienta a medida de las necesidades y requisitos del portal web de la UOC.

A la hora de realizar la programación del proyecto y el desarrollo se ha buscado la simplicidad al máximo ya que como en productos muy usados de grandes empresas como google, esto se traduce en una mejor experiencia para el usuario facilitándole el uso y consiguiendo mayor conversión de intentos de usar la herramienta satisfactoriamente.

Aunque estas herramientas pueden ser muy útiles en muchos entornos web, siempre dependen de un factor externo al desarrollo y diseño que es el factor humano que da el soporte final, el tiempo de respuesta, la precisión de las soluciones y el mensaje que transmite el operador o gestor del departamento a la hora de responder una incidencia es en la mayoría de los casos con lo que el usuario final se queda en este tipo de aplicaciones.

3.2 Objetivos

El objetivo del proyecto es la realización de una aplicación web que sirva para gestionar incidencias, detalladamente la aplicación deberá cumplir las siguientes funciones:

- § Permitir que un usuario (estudiantes o consultores) pueda realizar una consulta sobre un listado de temas relacionados con el campus virtual de la UOC.
- § Los temas de las consultas estarán relacionados internamente a un departamento u otro según la tipología, por lo que las consultas se asignaran automáticamente a un departamento.
- § El departamento podrá dar una respuesta a la consulta o si considera que no está bien dirigida la consulta, desviarla al departamento correspondiente.
- § En el caso de que se dé una respuesta la consulta quedara cerrada.
- § Si un usuario considera que no ha sido resuelta su consulta, podrá volver a abrir la incidencia para obtener la respuesta.

3.3 Motivación

El motivo por el cual he decido realizar este proyecto es que debido a mi trabajo en la empresa estamos constantemente dando soporte a usuarios, para este soporte utilizamos herramientas gratuitas de código abierto, pero en la mayoría de los casos, hacer cualquier tipo de modificación requiere de demasiado trabajo por cómo están realizadas.

Por este motivo quiero aprovechar este proyecto para tratar de aplicar los conocimientos que he adquirido usando estas herramientas y realizar una versión sencilla de usar tanto para el usuario como el administrador que hace uso de ella y que no requiera de complicadas configuraciones y conocimientos para poder hacer uso de ella. Pero que al mismo tiempo tenga las herramientas y funcionalidades necesarias para poder ser una aplicación completa y útil.

3.4 Enfoque y método seguido.

Después de barajar distintas opciones de frameworks de programación en PHP, se ha optado por usar CakePHP¹, el principal motivo es que de todos los frameworks modelo – vista – controlador que hay ahora mismo, es de los que tiene mejor documentación y comunidad detrás que lo apoya, y suele ser bastante actualizado, lo cual se agradece cuando aparecen fallos que corregir o implementan mejoras.

Finalmente se ha usado la versión 2.0.0. para el proyecto, aunque cuando se comenzó con la programación y análisis de la aplicación aun era una versión beta.

Con el uso de este framework (así como con la mayoría), esperamos que los tiempos de implementación se vean reducidos en tareas cotidianas y nos permita acometer los objetivos en un tiempo correcto como para poder dedicar recursos a la usabilidad de la aplicación, punto muy importante a la hora de realizar una herramienta que van a usar otros usuarios y en algunos casos puede haber usuarios no formados.

¹ <http://cakephp.org/>

3.5 Planificación de tareas

Planificación de las tareas a realizar:

1. Plan de trabajo
2. Estudio y análisis
3. Diseño de la implementación
4. Implementación de la web
5. Prueba y corrección de errores
6. Documentación y finalización de la memoria
7. Preparación de la presentación
8. Resolución de preguntas

Fase 1

Plan de trabajo (PAC 1)

Duración: 26/09/2011 a 02/10/2011 (1 semana)

Presentación de la opción elegida para el desarrollo del TFC y planificación de las tareas del proyecto, además de otra serie de puntos a desarrollar en la entrega de la PAC.

Fase 2

Estudio y análisis (PAC 2)

Duración: 03/10/2011 a 16/10/2011 (2 semanas)

Estudio y análisis de las aplicaciones existentes en Internet para comparar y seleccionar las características destacadas de cada una. Además analizaremos la estructura que deberá tener nuestra aplicación con lo que hayamos aprendido de las demás.

Fase 3

Diseño de la implementación (PAC 3)

Duración: 17/10/2011 a 08/11/2011 (3 semanas)

Se realizarán diagramas de actividad, diagramas de secuencia, prototipo de la interfaz de usuario y administración.

En esta fase también seleccionaremos el lenguaje de programación que utilizaremos para el desarrollo.

Fase 4

Implementación de la web (PAC 4)

Duración: 09/11/2011 a 27/12/2011 (7 semanas)

Se programara la aplicación planteada en el proyecto utilizando un método de trabajo por metas que serán igual a los objetivos que debe cumplir el proyecto.

Fase 5

Prueba y corrección de errores

Duración: 28/12/2011 a 10/01/2012 (2 semanas)

Se buscara una serie de usuarios que hagan las funciones de alumnos y departamentos para que puedan probar el sistema desde ambos puntos de vista, cualquier fallo detectado será corregido en este periodo de tiempo.

Si ha quedado algo pendiente por falta de tiempo de la implementación se aprovechara este tiempo también para acabar funcionalidades.

Fase 6

Documentación y finalización de la memoria (PAC 4)

Duración: 20/12/2011 a 10/01/2012 (3 semanas)

Realización de la documentación del proyecto y finalización de la memoria la cual se habrá ido completando desde el inicio del proyecto. Esta documentación junto con la prueba de la aplicación y la presentación serán el último paso del proyecto.

Fase 7

Preparación de la presentación (PAC 4)

Duración: 03/01/2012 a 10/01/2012 (1 semana)

Tiempo para prepara la presentación del proyecto que acompaña a la memoria.

Fase 8

Resolución de preguntas

Duración: 10/01/2012 a 25/01/2012

Este el tiempo que está destinado a que los miembros del tribunal realicen las preguntas sobre el proyecto.

4.- Análisis

4.1 Requisitos no funcionales

- § **Rendimiento:** Los procesos de insertar incidencia desde el lado del usuario y responder incidencia desde el lado del departamento serán rápidas, no es necesario realizar consultas complejas en este proceso lo que no debería hacer que el rendimiento caiga. En general la aplicación no tiene grandes necesidades de hardware para funcionar, según el número de usuarios final, se podría escalar el hardware en una o varias máquinas, pero para este proyecto trabajaremos todo en la misma máquina.
- § **Distribución:** A la hora de distribuir el proyecto, será necesario instalar un servidor web y un servidor de base de datos MySQL, esto hace que se generen dos partes, por un lado la estructura de la base de datos con los datos mínimos para que funcione el sistema, y por otro lado los archivos del propio sistema.
- § **Fiabilidad:** Dado que precisamente esta aplicación servidora para informar de incidencias en el entorno virtual, un fallo en el proceso de envío de incidencias o lectura de la respuesta dada no es adecuado, se trabajara para asegurar que no se cometa ningún error en este sentido.
- § **Seguridad:** Sólo ciertos empleados de la universidad acreditados con su nombre de usuario y contraseña propios podrán realizar tareas en diversos apartados de la aplicación.
- § **Usabilidad:** La interfaz será sencilla y directa, se evitara en la medida de lo posible tener que realizar varios pasos para las tareas más sencillas, además se intentara evitar formularios complejos.
- § **Accesibilidad:** Se tratara de garantizar un nivel aceptable de accesibilidad para los usuarios con necesidades especiales, para ello se usaran las directrices de la WCAG²

² <http://www.w3.org/WAI/intro/wcag.php>

4.2 Casos de uso

Para poder aproximar la interacción entre el usuario y la aplicación se presentan los siguientes casos de uso, para así establecer pautas que servirán durante el diseño de la aplicación.

4.2.1 Actores

En este proyecto habrá dos actores principales, uno será el usuario final que será un estudiante o consultor que hará uso del entorno virtual de la UOC, por el momento no se contempla ninguna necesidad de diferenciar al estudiante del consultor. El otro actor principal será un miembro del alguno de los departamentos que podrán dar soporte a los estudiantes o consultores.

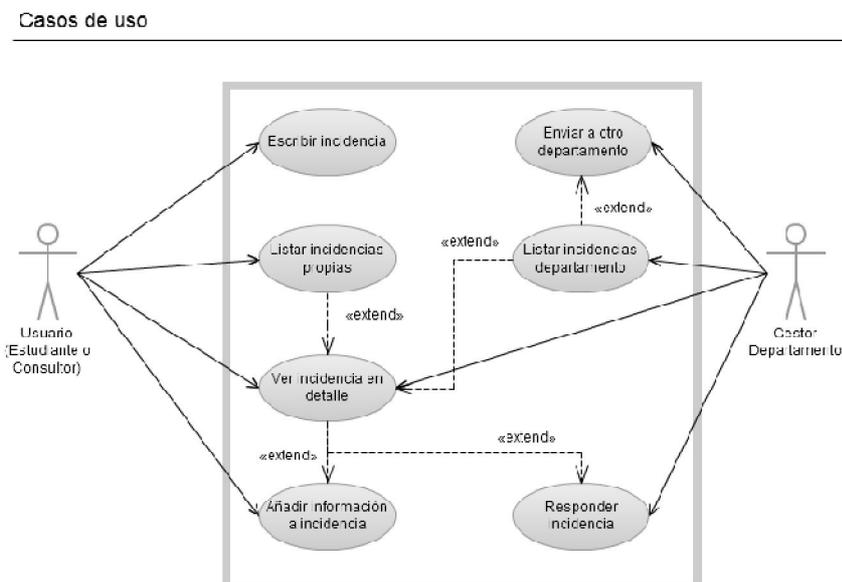
En ambos casos el perfil medio será un usuario iniciado en la informática, ya que previamente ya habrán hecho uso del entorno virtual y el proceso de incidencias debe estar integrado como una parte más.

4.2.2 Guiones

Al haber dos actores principales, tenemos varios guiones importantes. En primer lugar está el usuario que hace uso de la aplicación para realizar una consulta, además este usuario tiene la posibilidad de leer la respuesta a esta consulta cuando haya sido respondido.

En segundo lugar tenemos el guion del miembro del departamento que debe responder la consulta, el cual dispone de dos opciones, resolver la consulta si es algo que atañe a su departamento, o en el caso de que la consulta este mal dirigida, enviar la consulta a otro departamento.

4.2.3 Documentación formal, Diagrama de casos de uso



4.2.4 Documentación textual, descripción textual

Nombre:	Caso 1: Listar incidencias propias
Funcionalidad:	Consultar el historial de incidencias para poder seleccionar una en detalle
Actores:	Usuario (Estudiante o Consultor)
Precondición:	<ol style="list-style-type: none">1. El usuario tiene que estar identificado en el campus virtual.2. El usuario tiene que haber escrito incidencias que estén almacenadas en el sistema.
Flujo normal:	<ol style="list-style-type: none">1. El usuario entra en la aplicación de incidencias y se le muestra el listado de incidencias escritas, esta es la pantalla principal del usuario cuando está identificado.
Flujo alternativo	<ul style="list-style-type: none">- Si el usuario no tiene incidencias, se le mostrara un mensaje de "No hay incidencias para mostrar".- El usuario puede desde esta pantalla entrar en la opción de escribir una incidencia.
Postcondición:	Se mostrara el listado de incidencias ordenadas por fecha.

Nombre:	Caso 2: Escribir incidencia
Funcionalidad:	Recoger la incidencia del usuario en un formulario y enviarla al sistema
Actores:	Usuario (Estudiante o Consultor)
Precondición:	1. El usuario tiene que estar identificado en el campus virtual.
Flujo normal:	<ol style="list-style-type: none"> 1. El usuario rellena los campos y envía la incidencia. 2. Se comprueban las condiciones de los campos de texto, para ver que toda la información necesaria esta rellena. <ol style="list-style-type: none"> 2.1. En caso de error, se pide que se corrija. 2.2. Si todo esta correcto, se avisa al usuario y se vuelve a la pantalla de historial de incidencias.
Flujo alternativo	- El usuario quiere conocer el estado de una incidencia enviada con anterioridad.
Postcondición:	Se inserta una incidencia en el sistema con estado abierta.

Nombre:	Caso 3: Listar incidencias departamento
Funcionalidad:	Listar las incidencias que están pendientes de responder
Actores:	Gestor
Precondición:	<ol style="list-style-type: none"> 1. El usuario tiene que estar identificado en el gestor de incidencias. 2. El usuario tiene que estar en la pantalla de "Listado de consultas pendientes".
Flujo normal:	<ol style="list-style-type: none"> 1. El gestor entra en la aplicación de incidencias y se le muestra el listado de incidencias pendientes de responder por su departamento, esta es la pantalla principal cuando el gestor se identifica.
Flujo alternativo	<ul style="list-style-type: none"> - Si en el gestor de incidencias no hubiera consultas pendientes, se le mostrara un mensaje de "No hay consultas para mostrar".
Postcondición:	Se mostrara la consulta en detalle y un campo de texto para responder.

Nombre:	Caso 4: Ver incidencia en detalle
Funcionalidad:	Consultar el estado de una incidencia realizada con anterioridad
Actores:	(A) Usuario (Estudiante o Consultor) o (B) Gestor
Precondición:	<ol style="list-style-type: none"> 1. El usuario tiene que estar identificado en el campus virtual. <ol style="list-style-type: none"> A. El usuario tiene que estar en la pantalla de "Historial de incidencias". B. El gestor tiene que estar en la pantalla de listar incidencias de departamento.
Flujo normal:	<ol style="list-style-type: none"> 1. El usuario selecciona la incidencia que quiere ver detallada. <ol style="list-style-type: none"> A. El usuario tendrá la posibilidad de añadir información adicional a la incidencia ya abierta, o responder a preguntas del gestor que trata de darle una solución. B. El gestor tendrá la posibilidad de responder la incidencia o dirigirla a otro departamento.
Flujo alternativo	<ul style="list-style-type: none"> - Si el usuario no tiene consultas realizadas, se le mostrara un mensaje de "No hay incidencias para mostrar".
Postcondición:	Se mostrara la incidencia en detalle con la fecha de respuesta y estado de la misma y todo el historial de mensajes que se haya podido intercambiar entre gestor de departamento y usuario con problema.

Nombre:	Caso 5: Añadir información a incidencia
Funcionalidad:	Añadir consultas o comentarios a una respuesta recibida
Actores:	Usuario (Estudiante o Consultor)
Precondición:	<ol style="list-style-type: none"> 1. El usuario tiene que estar identificado en el gestor de incidencias. 2. El usuario tiene que estar en la pantalla de vista en detalle de incidencia.
Flujo normal:	<ol style="list-style-type: none"> 1. El usuario quiere información adicional o considera que no está completa la respuesta recibida, así que decide añadir una consulta sobre la incidencia. <ol style="list-style-type: none"> 2.1. El usuario puede estar respondiendo a una solicitud de más información que han hecho desde el departamento al que hizo la pregunta. 2.2. El usuario puede estar respondiendo a una posible solución que le han ofrecido y que puede no ser la respuesta correcta o que le falten detalles para resolver la duda.
Flujo alternativo	<ul style="list-style-type: none"> - El usuario está conforme con la respuesta y cierra la incidencia.
Postcondición:	La incidencia volver a estar en estado pendiente.

Nombre:	Caso 6: Enviar a otro departamento
Funcionalidad:	Desviar la incidencia a otro departamento que pueda resolver la incidencia
Actores:	Gestor
Precondición:	<ol style="list-style-type: none"> 1. El usuario tiene que estar identificado en el gestor de incidencias. 2. El usuario tiene que estar en la pantalla de vista en detalle de incidencia.
Flujo normal:	<ol style="list-style-type: none"> 1. El gestor decide que su departamento no es el correcto para responder la pregunta y desvía la incidencia a otro departamento más adecuado. 2. La incidencia deja de aparecer en el listado del departamento del gestor que la movió.
Flujo alternativo	<ul style="list-style-type: none"> - Si el departamento del gestor es el adecuado responderá la incidencia.
Postcondición:	La incidencia cambiara de departamento y permanecerá abierta.

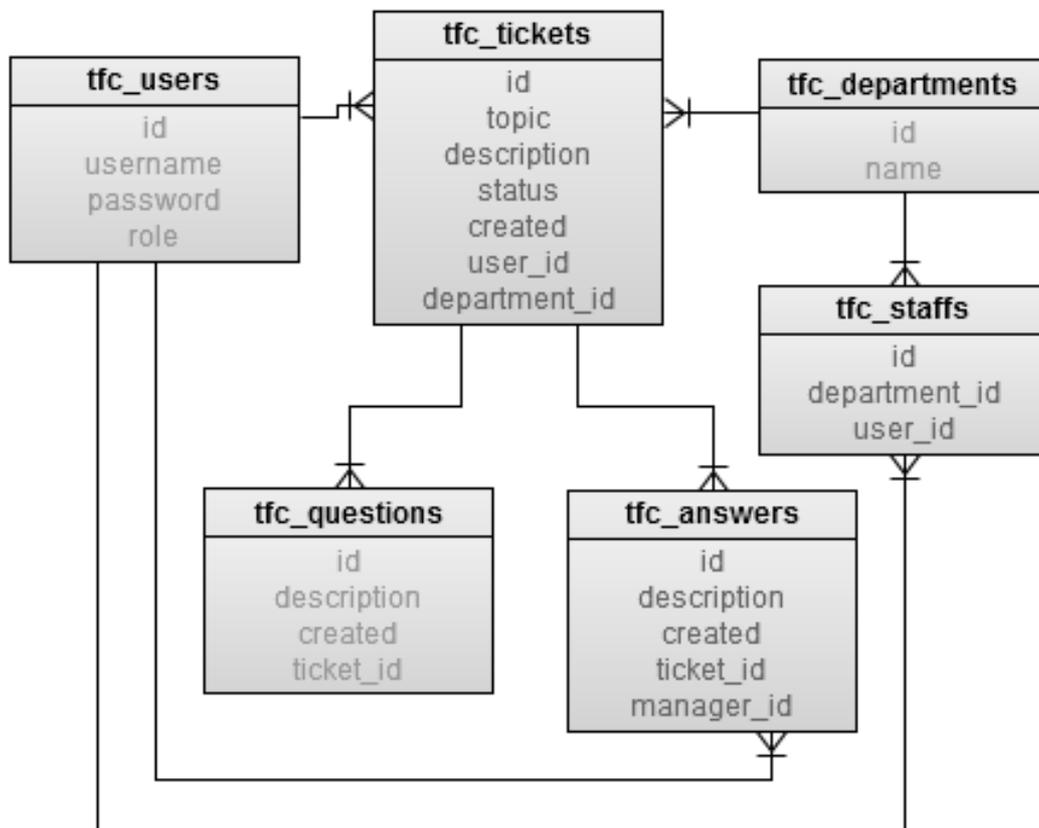
Nombre:	Caso 7: Responder incidencia
Funcionalidad:	Responder una incidencia de un usuario
Actores:	Gestor
Precondición:	<ol style="list-style-type: none"> 1. El usuario tiene que estar identificado en el gestor de incidencias. 2. El usuario tiene que estar en la pantalla de vista en detalle de incidencia.
Flujo normal:	<ol style="list-style-type: none"> 1. El gestor decide responder la pregunta de la incidencia abierta. 2. Antes de responder elige si es una solución al problema planteado o no dispone de información suficiente para resolver la incidencia y esta solicitando más información.
Flujo alternativo	<ul style="list-style-type: none"> - Si la incidencia no corresponde con el departamento desviara la pregunta al departamento adecuado.
Postcondición:	La incidencia pasara a estado solucionado o solicitud de información a la espera de que el usuario la lea y la cierre.

4.3 Modelo ER

Sobre el modelo ER se han hecho pocos cambios del diseño original, la nomenclatura de las tablas y los campos se ha hecho en ingles por costumbre a la hora de programar, además se ha añadido el prefijo "tfc_" para los casos en el que la tablas comparten base de datos con otras aplicaciones sea más fácil identificarlas.

En nuestro caso, tanto los usuarios como los gestores vendrán del campus virtual, pero para poder realizar en su momento demostraciones y pruebas, hemos creado una tabla de usuarios, la única peculiaridad que tiene esta tabla, pero que se podría poner en una tabla intermedia si el sistema donde se instala no se pudieran hacer modificaciones es el campo de "rol", que nos indica que tipo de usuario es. Los usuarios que pertenecen a un departamento porque son gestores están relacionados con la tabla departamentos por la tabla staff, donde se indica que usuario pertenece a que departamento.

Las incidencias pertenecerán tanto a un usuario como a un departamento (a no ser que el gestor haga un cambio de departamento), las incidencias podrán tener respuestas de los gestores, o más preguntas complementarias del usuario que la realiza. Para llevar un control interno se ha relacionado al respuesta del gestor con el usuario al que pertenece ese gestor, así se podrá seguir la respuesta en caso necesario (motivos de evaluación, etc).



5.- Diseño

5.1 Requisitos funcionales

Los usuarios podrán escribir incidencias, estas se guardaran en la base de datos a la espera de que un gestor del departamento elegido por el usuario a la hora de escribir la incidencia responda, puede darse el caso de que el departamento elegido no sea el adecuado, en este caso, el gestor tiene la opción de transferir la incidencia al departamento adecuado.

El gestor responderá a la incidencia e indicar si está ofreciendo una solución o necesita más información para poder resolver la incidencia del usuario y el usuario podrá leer desde su historial de incidencias la respuesta recibida a su pregunta. Pueden ocurrir tres cosas, que el usuario considere solucionado su problema y decida cerrar la incidencia en cuyo caso marcaremos la incidencia como cerrada, que el usuario decida que no está solucionada y responda en la misma incidencia enviando una pregunta más, en cuyo caso el proceso volverá a empezar o que el usuario deba añadir información solicitada por el gestor para poder resolver la incidencia.

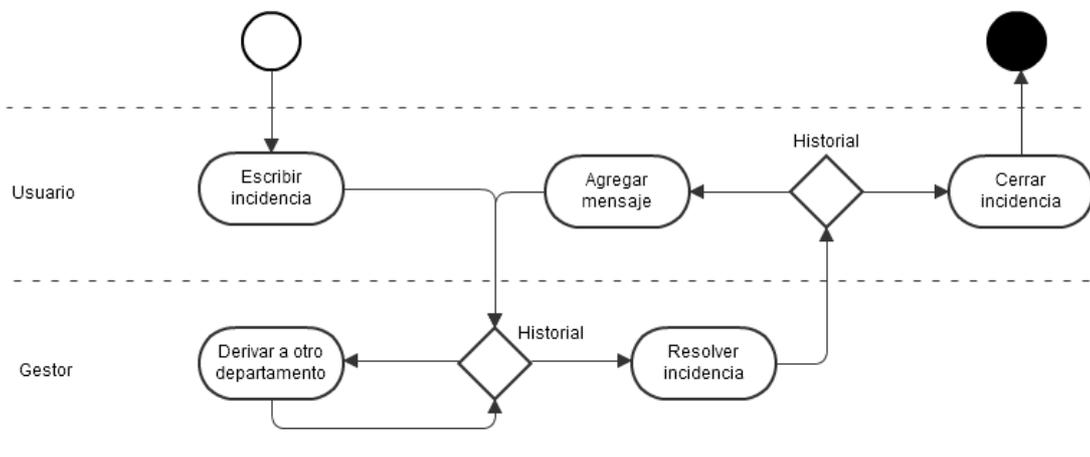
Los gestores tienen la posibilidad de corregir mensajes enviados si no han pasado 24 horas desde que enviaron el mensaje, esto se ha hecho para poder corregir entre cosas posibles faltas ortográficas o por si con el paso del tiempo el gestor ha tenido opción de completar la respuesta con mas información de la que estaba pendiente. Al usuario no se le ha brindado esta posibilidad aunque si tras un estudio se considerara una opción útil seria fácilmente programable.

5.2 Diagrama de actividad

El diagrama de actividad, muestra el flujo normal de trabajo que llevara la aplicación. El mecanismo es bastante sencillo, cuando un usuario escribe una incidencia, esta tiene dos posibles opciones, que se derive a otro departamento por que el usuario no haya seleccionado correctamente el departamento que puede resolver la incidencia, o que se dé una respuesta al usuario para resolver la incidencia o se solicite mas información al usuario.

Cuando el usuario vuelva a ver la respuesta a la incidencia, tendrá que decidir si está satisfecho y la da por cerrara, o por el contrario sigue teniendo problemas / dudas y decide continuar preguntando en la misma incidencia, en este último caso, todo el proceso vuelve a empezar.

Diagrama de actividad



5.3 Diseño de interfaz

5.3.1 Vista de usuario

El diseño de la interfaz que usaremos será muy similar en estética al nuevo diseño que la empresa Google ha adoptado para sus aplicaciones web como gmail y google docs. Este diseño minimalista y limpio permite a los usuarios menos formados entender rápidamente el funcionamiento de sus páginas, por lo que se considera adecuado para una página como la que pretendemos crear.

Además del estilo limpio y fácil de entender, los diseños de google carecen apenas de imágenes lo que hace que la carga de las paginas sea más rápida, ese aspecto nos interesa en nuestro caso ya que la pagina que queremos crear es un trámite que en ocasiones puede resultar hasta molesto para los usuarios ya que se encuentran ante un problema, y la rapidez de carga y sencillez del sistema para gestionar sus incidencias harán más llevadero el proceso.

Por ejemplo, los formularios tendrán los campos necesarios, con espacio para escribir y sin distracciones que puedan provocar dudas al usuario.

demo Desconectar

Añadir ticket:

Listado de tickets

Asunto:

Descripción:

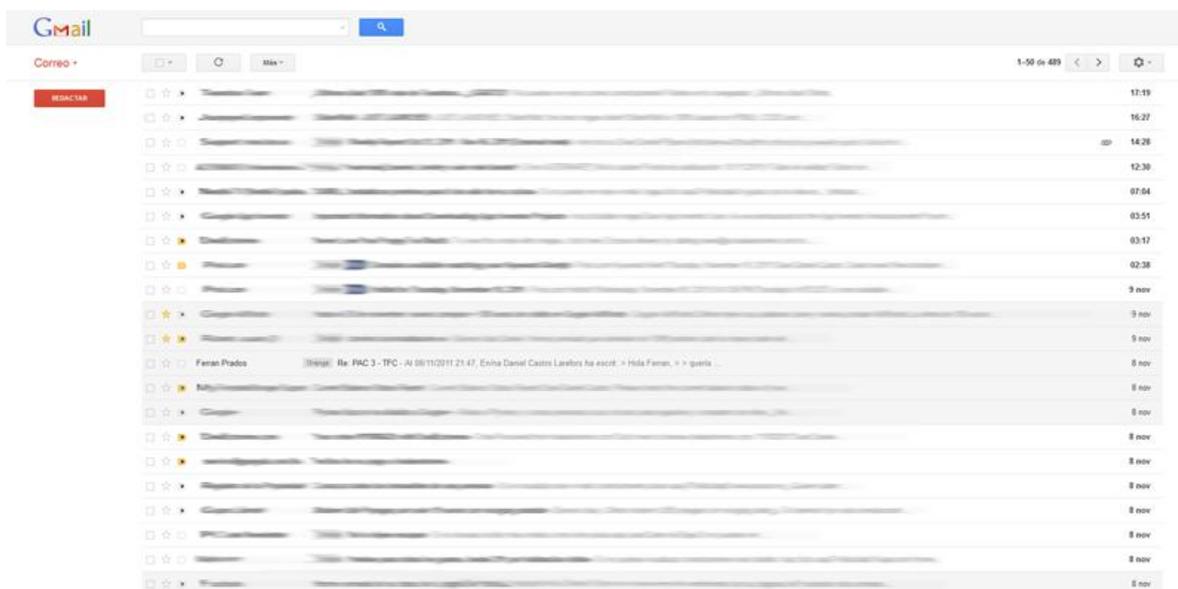
Departamento:

Enviar

Sistema de gestión incidencias en entornos virtuales - TFC UOC 2011/2012 - Daniel Castro Larefors

El historial de incidencias será similar a la bandeja de entrada de gmail, aparecerán las incidencias ordenadas por fecha de modificación (así saldrán arriba las ultimas respondidas y no las ultimas creadas, que podrían estar aún pendientes de responder). En el caso de los usuarios, no será necesario mostrar el nombre ya que serán incidencias propias, solo el titulo de la incidencia.

Ejemplo Gmail



Resultado obtenido en el TFC

#	Estado	Asunto	Departamento	Creado	Últ. actualización	Opciones
11	Cerrado	Incidencia 1	Informática	07/01/12	08/01/12 12:01	
13	Abierto	Incidencia 3	Biblioteca	07/01/12	07/01/12 21:09	<input type="button" value="Cerrar"/>
12	Abierto	Incidencia 2	Secretaria	07/01/12	07/01/12 21:09	<input type="button" value="Cerrar"/>

Sistema de gestión de incidencias en entornos virtuales - TFC UOC 2011/2012 - Daniel Castro Larefors

Cuando el usuario entre a ver en detalle una incidencia usaremos un sistema similar a como Gmail enseña el intercambio de correos entre usuarios. El usuario tendrá un campo de texto para añadir un mensaje a la incidencia de forma rápida. De cada respuesta podrá ver quién es el personal de la UOC que le responde. Si el desarrollo se hiciera sobre el portal de la UOC aprovecharíamos las imágenes de perfil de los usuarios para añadirlo a las respuestas, estas y otras mejoras se propondrán en la versión final del proyecto. Al igual que en Gmail, se ve fácilmente el tiempo transcurrido del mensaje. Además los mensajes del usuario están sombreados para distinguirlos más rápidamente.

Ejemplo Gmail



Resultado obtenido en el TFC

demo Desconectar

Incidencias usuario: demo

Listado de tickets

(#11) Incidencia 1 Cerrado

demo 07/01/12 21:09
Este es el texto de la incidencia 1

gestor_informatica 07/01/12 21:25 (15 hours, 18 minutes ago)
primera respuesta

demo 08/01/12 11:58 (44 minutes ago)
el usuario añade más información

gestor_informatica 08/01/12 12:01 (42 minutes ago)
propuesta de solución

Responder:

Enviar

Sistema de gestión incidencias en entornos virtuales - TFC UOC 2011/2012 - Daniel Castro Larefors

5.3.2 Vista de Gestor

El diseño del administrador de los departamentos será muy similar al de los usuarios, la única diferenciación el historial será que se identificara al usuario en cada incidencia, y se priorizaran visualmente las incidencias que aun no han recibido respuesta ya sea marcándolas con algún ícono identificativo o sombreando de algún color en el que se refleje la necesidad de atenderlo lo antes posible.

gestor_informatica Desconectar

Incidencias usuario: gestor_informatica

Listado de tickets

(#11) Incidencia 1 Cerrado Informática Enviar

demo 07/01/12 21:09
Este es el texto de la incidencia 1

gestor_informatica Editar respuesta Borrar respuesta 07/01/12 21:25 (15 hours, 20 minutes ago)
primera respuesta

demo 08/01/12 11:58 (46 minutes ago)
el usuario añade más información

gestor_informatica Editar respuesta Borrar respuesta 08/01/12 12:01 (44 minutes ago)
propuesta de solución

Responder:

Estado Cerrado

Enviar

Sistema de gestión incidencias en entornos virtuales - TFC UOC 2011/2012 - Daniel Castro Larefors

Las principales diferencias entre el usuario y el gestor son las opciones que necesita el gestor para poder gestionar las incidencias, tiene opción de transferir la incidencia a otro departamento si lo considera necesario, además cuando responda podrá cambiar el estado de la incidencia si está solicitando información extra o está proponiendo una solución.

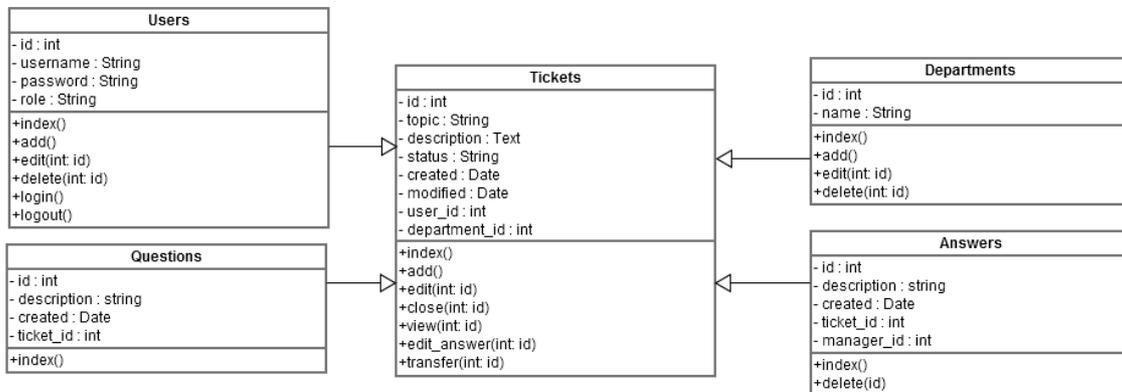
Además el gestor, tiene la opción de editar o borrar una respuesta si no han pasado más de 24 horas, esto da margen de error en el caso de que haya faltas ortográficas, confusión en la pregunta y la respuesta sea inadecuada, etc. En tiempos más prolongados ya no se da la opción para evitar que se pueda perder el sentido de los mensajes iniciales.

5.3.3 Vista de Administrador

El usuario de tipo administrador tiene acceso a una serie de pantallas que no tienen los usuarios y gestores. Estas pantallas sirven para añadir, editar o eliminar usuarios, y añadir, editar o eliminar departamentos. No hemos incluido capturas de pantalla ya que no se encontraban entre los objetivos del proyecto aunque eran un paso necesario para un sistema completo de gestión de incidencias, en las instrucciones de prueba se da acceso también al usuario administrador en caso de querer consultar dichas pantallas.

5.4 Diagrama de clases

Este es el diagrama final del proyecto, ha habido cambios sobre el mencionado en fases anteriores.



5.4.1 Clase Users

En el controlador **Users**, la función "*index*" sirve para visualizar los usuarios introducidos en el sistema, la función "*add*" introduce nuevos usuarios en la base de datos, cuando damos de alta un usuario, especificamos el rol que va a tener, y en caso de que sea gestor, le asignamos el departamento al que pertenece. A través de las funciones de guardado múltiple de CakePHP y las relaciones establecidas en los modelos, podemos guardar toda la información desde la función "*add*", creando los registros necesarios en la tabla de usuarios y en la tabla de Staff que sale detallada en el modelo ER. Las funciones "*edit*" y "*delete*" tal como sus nombres indican editan datos de usuario o eliminan un usuario. Estas funciones están pensadas para usarse desde un usuario con rol de Administrador.

Las funciones "*login*" y "*logout*" son las encargadas de comprobar que el usuario existe en la base de datos de usuarios y de crear o destruir la sesión en cada caso.

5.4.2 Clase Departments

Esta clase sirve para dar de alta, editar o eliminar Departamentos del sistema, únicamente puede hacer uso de estas funciones un usuario con rol de Administrador.

5.4.3 Clase Tickets

En la función "*index*" mostramos los tickets, en el caso de que el usuario identificado sea un usuario (alumno o consultor) vera únicamente sus tickets, que habremos filtrado usando el ID de usuario extraído de la sesión. Si el usuario identificado es un gestor, obtenemos el departamento al que pertenece y mostramos únicamente los tickets de ese departamento.

La función "*add*" solo la puede usar un usuario normal, sirve para insertar una incidencia. La función "*edit*" no es visible para los usuarios, ya que no se permite editar una incidencia, el usuario siempre tiene la opción de añadir un mensaje en caso de querer añadir más información.

Desde la vista "*view*" de Tickets hacemos la mayoría del trabajo, en un primer plano es la función para visualizar en detalle un ticket, pero además se usa para guardar las respuestas o mensajes que puedan añadir los usuarios o gestores a ese mismo ticket, en la vista de esta función, hacemos la edición y/o eliminación de una respuesta de un gestor, llamando después a las funciones correspondientes.

Siendo gestor, y en la vista de esta función y cuando las normas lo permite, podemos eliminar una respuesta de un gestor, que hará la llamada al controlador correspondiente, además desde la vista de este mismo gestor podemos hacer la llamada a editar una respuesta de un controlador, en este caso, llamaremos a otra función "*edit_answer*" y vista que nos cambiara el mensaje que estamos editando por una caja de texto para poder hacer los cambios y guardarlos, una vez guardados, volveremos a la vista en detalle.

También siendo gestor desde esta vista podremos transferir de departamento el ticket visualizado, en este caso llamaremos a una función "*transfer*" que se encargara del cambio de departamento.

5.4.4 Clase Answers

Desde esta clase únicamente accedemos a la hora de borrar respuestas asociadas a un ticket, estas respuestas solo se pueden borrar durante las 24 horas posteriores a ser escritas.

5.4.5 Clase Questions

Esta clase no tienen ninguna función que utilicemos, únicamente hacemos uso del modelo y las opciones de validación para comprobar que los mensajes de los usuarios se insertar correctamente, la función de introducir mensajes está programada desde la misma vista en detalle de los tickets.

6.- Conclusiones

6.1 Síntesis

El desarrollo de esta aplicación web ha sido bastante instructiva, el primero de los puntos a favor ha sido la toma de contacto con la versión 2.0.0 de CakePHP, ya que hasta el momento solo se había hecho uso de la versión 1.3.0, aunque los cambios a nivel de programación no son muy drásticos, hay mejoras y cambios sutiles. Con el uso de CakePHP se ha podido acelerar y aprovechar procesos automatizados que ya tiene en su núcleo, cosas que han ayudado a completar objetivos.

También se ha aprovechado para hacer una aplicación completamente en HTML 5 y CSS3, permitiendo comprobar las mejoras que traen las evoluciones de estas tecnologías, al ser una tecnología aun en fase de pruebas, se ha hecho uso de una librería llamada "*moderniz*" que se encarga de suplir la carencia de los navegadores antiguos a la hora de renderizar funciones de HTML5 y CSS3, así cumplimos con la premisa de ser compatible con la mayoría de navegadores y no perdemos las funcionalidades que nos ofrecen estas nuevas versiones.

A nivel de aplicación, el grado de satisfacción es alto el funcionamiento es sencillo y ágil como se pretendía, las pantallas no son complicadas y permiten una carga rápida. El diseño como ya se especifico es simple y trata de no crear dudas en los usuarios. Aunque finalmente las opciones son menos de las que debería tener un sistema de estas características para ser completo, es un primer paso que debido a como está programado, es fácilmente escalable y fácil de añadir las funciones se consideren necesarias en un entorno real.

6.2 Nivel de consecución

De los objetivos iniciales planteados ha quedado pendiente crear un listado de temas de consulta relacionados con los departamentos, desde el cual, el usuario pueda elegir y se asigne automáticamente a ese departamento, en su lugar, el usuario debe elegir al departamento que quiere dirigir la pregunta y escribir el título de la consulta, que aunque no estaba contemplado en los objetivos, es un paso necesario ya que no siempre en los listados de temas están todas las opciones así que el poder escribir el tema y asignarlo a un departamento era un requisito necesario que no se contempló en los objetivos.

Este objetivo queda reflejado en las futuras mejoras.

6.3 Futuras mejoras

Entre las mejoras a plantear en una segunda versión sería finalizar el sistema de temas por departamento a la hora de plantear una incidencia. Otra de las mejoras sería añadir un repertorio de respuestas predefinidas para los gestores, a compartir entre los miembros del mismo departamento, esto permitiría que a las preguntas más comunes se pudiera seleccionar la respuesta predefinida.

A nivel de funcionalidades que se han quedado fuera por falta de tiempo y priorizar en otras tareas, sería añadir paginación a la sección de visualización de incidencias y visualización en detalle de una incidencia en caso de que se intercambiaran muchos mensajes. La posibilidad de ordenar el listado de incidencias por las diferentes columnas (estado, fecha de creación,...).

Teniendo acceso a un sistema más completo de usuarios, también se podrían añadir las fotos de los usuarios al estilo de Gmail y acceso directo a los perfiles de dichos usuarios haciendo clic sobre su nombre para facilitar la tarea de los gestores.

A nivel de programación quedarían por hacer varias comprobaciones de permisos y validaciones para asegurarnos que todos los campos aceptan el contenido que tienen que aceptar, por ejemplo el campo de texto del ticket y los mensajes en todos los sistemas de tickets suelen estar limitados por longitud.

6.4 Instalación y pruebas

Todos los archivos necesarios para la instalación en local del proyecto van incluidos, así como un archivo .sql con la estructura de la base de datos y una serie de datos introducidos de demostración. Únicamente hay que poner la carpeta de tfc en la raíz del servidor web y entrar por localhost/tfc/ o dominio.com/tfc

Una vez subido los archivos y cargado el SQL, como es probable que el nombre de la base de datos, usuario y contraseña no coincida hay que editar el archivo:

.../tfc/app/Config/database.php

En las líneas 62 a 71 se encuentran los parámetros de usuario (login), contraseña (password) y nombre de la base de datos (database).

A continuación se detallan los usuarios existentes en el sistema:

Administrador desde el que se pueden dar de alta usuarios nuevos o departamentos:

Usuario: admin Contraseña: demo

Gestores de departamentos:

Usuario: gestor_informatica Contraseña: demo

Usuario: gestor_secretaria Contraseña: demo

Usuario: gestor_biblioteca Contraseña: demo

Usuario normal (Alumno o Consultor):

Usuario: demo Contraseña: demo

Se ha dado de alta una versión online del proyecto en www.danielcastro.es/tfc con los mismos usuarios para poder probarlo.

7.- Glosario

Framework

La palabra inglesa "**framework**" define, en términos generales, un conjunto estandarizado de conceptos, prácticas y criterios para enfocar un tipo de problemática particular, que sirve como referencia para enfrentar y resolver nuevos problemas de índole similar.

Modelo Vista Controlador (MVC)

Se trata de un patrón de arquitectura del software que separa los datos, la interfaz y la lógica en tres componentes. Existen varios "frameworks" en distintos lenguajes de programación que hacen uso de este patrón.

Web Content Accessibility Guidelines (WCAG)

Son parte de una serie de recomendaciones sobre accesibilidad web publicadas por la W3C, estas recomendaciones van dirigidas a realizar el contenido de las páginas web disponible para personas con alguna incapacidad. La última versión de esta guía es la 2.0.

PHP

Lenguaje de programación interpretado, diseñado para la creación de webs dinámicas.

HTML 5

Quinta revisión importante del lenguaje de la "*World Wide Web*", HTML. Aunque se encuentra en modo experimental es compatible con las últimas versiones de los navegadores, además en este proyecto se ha hecho uso de la librería "*modernizr*" para suplir las carencias de los navegadores viejos.

CSS

Proviene del inglés "*Cascading Style Sheets*", hojas de estilo en cascada. Es un lenguaje usado para definir la presentación de un documento HTML.

8.- Referencias

Sistemas de tickets analizados:

<http://www.osticket.com/>

<http://www.zendesk.com> (Pago)

<http://www.eticketsupport.com> (Gratis)

Recursos usados, documentación y sitios de consulta de programación:

http://es.wikipedia.org/wiki/Modelo_Vista_Controlador

<http://www.wampserver.com/en/>

<http://www.modernizr.com/>

<http://jquery.com/>

<http://www.cakephp.org>

<http://groups.google.com/group/cake-php>

<http://www.php.net/>

<http://stackoverflow.com/>

<http://www.w3schools.com/html5/default.asp>