

Chat de Soporte Técnico, ChatO

Miembros:

Guillermo Pastor Peña, Juan Blasco Cuenca, Raquel Sánchez Barahona, Miguel Ángel Monsalve Pardo, Irene Fender Soler, Sergio Rojo Herrero, Marta García Nevado, Mónica López Fernández, Fernando Castro Martín, Svetlana Stefanova Ivanova.

Resumen

Nuestro proyecto consiste en la creación de un Chat para Quental mediante el cual se podrán resolver dudas técnicas a sus clientes, en concreto Cruz Roja y Telefónica. El Chat se ha desarrollado siguiendo las especificaciones del cliente y mediante el uso de diversas tecnologías.

Para llevar a cabo el diseño del chat, hemos utilizado Liferay que es un portal de gestión de contenidos de código abierto escrito en Java. Un portal se define generalmente como una plataforma de software para construir aplicaciones y sitios web. Las plataformas de portal facilitan a los usuarios la construcción de páginas web y sitios web mediante el ensamblaje de Portlets en una página de portal.

Hemos creado dos Portlets para desarrollar todas las funcionalidades del proyecto, uno para el lado cliente y otro para el lado Quental.

Palabras Clave

Quental, Talentum Empleo, Fundación Telefónica, Java, Java EE, Liferay, Chat Soporte Técnico, JavaScript, Ajax, HTML5/CSS3

Índice

1	Descripción	1
2	Objetivos	1
3	Alcance del proyecto	1
4	Tecnologías aplicadas	1
5	Proceso	2
6	Conclusiones	3
7	Agradecimientos	3

1. Descripción

Dicho chat permitirá poner en contacto y en tiempo real a un equipo de soporte con un técnico del lado del cliente, que será el que expondrá sus dudas.

2. Objetivos

Crear una herramienta que facilite el trabajo del cliente en sus desarrollos internos utilizando la tecnología Liferay, es decir que será un SPOC (Single Point Of Contact).

3. Alcance del proyecto

Guiándonos por las especificaciones que nos ha pedido Quental, el proyecto "ChatO" tiene las siguientes características:

- Los usuarios del chat no podrán interactuar entre ellos, sólo lo harán con los técnicos de soporte
- El técnico de soporte podrá incluir a otro técnico en el chat establecido para un usuario. En ningún momento el usuario sabrá con cuántos técnicos está chateando.
- El cliente se identificará con un alias.
- Los técnicos de soporte serán anónimos de cara al cliente.
- Se permitirá enviar ficheros para facilitar al programador proporcionar indicaciones estándar a cliente con la solución o pasos a seguir.
- Para facilitar las búsquedas en la base de datos, las conversaciones serán etiquetadas y tendrán una descripción, ambas se definirán antes de ser cerradas por el técnico.
- El cliente podrá seleccionar una categoría preestablecida para orientar a los técnicos sobre la duda que tiene.
- Las conversaciones han de ser guardadas en BBDD.
- El servicio técnico tendrá acceso a todas las conversaciones pendientes de ser atendidas.
- El servicio técnico podrá salir de una conversación temporalmente pudiendo retomarla más tarde.
- El servicio técnico podrá descargar una conversación en curso en formato pdf la cual podrá ser enviada al cliente.

4. Tecnologías aplicadas

- Portal de gestión de contenidos Liferay 6.2 basado en Java EE.

- Apache Tomcat 7.
- MVC Portlets.
- CSS3/HTML5.
- Maven.
- Javascript (AJAX/ JQuery/ AlloyUI).
- Eclipse IDE con plugin de LifeRay.
- Git y GitHub.

5. Proceso

Nos basamos en el modelo vista controlador (MVC) que es un patrón que define la organización independiente del Modelo (Objetos de Negocio), la Vista (interfaz con el usuario u otro sistema) y el Controlador (controlador del workflow de la aplicación). De esta forma, dividimos el sistema en tres capas donde tenemos la encapsulación de los datos, la interfaz o vista y la lógica interna o controlador. El patrón de arquitectura "modelo vista controlador", es una filosofía de diseño de aplicaciones, compuesta por:

- **Modelo**
 - Contiene el núcleo de la funcionalidad (dominio) de la aplicación.
 - Encapsula el estado de la aplicación.
 - No sabe nada / independiente del Controlador y la Vista.
- **Vista**
 - Es la presentación del Modelo.
 - Puede acceder al Modelo pero nunca cambiar su estado.
 - Puede ser notificada cuando hay un cambio de estado en el Modelo.
- **Controlador**
 - Reacciona a la petición del Cliente, ejecutando la acción adecuada y creando el modelo pertinente

Para conseguir un chat funcional lo primero hemos creado tres entidades:

- **Conversación:** nos permitirá identificar mediante un id, que será clave primaria (PK) y autoincremental, cada una de las conversaciones que un cliente inicie, facilitando su almacenamiento en la base de datos.
- **Mensaje:** cada uno de los mensajes que se intercambien entre cliente y soporte estará identificado mediante un id, que igualmente es autoincremental y PK, de esta manera podrán agruparse por conversación, ya que existe una relación n:1 entre mensajes y conversación.
- **Asignación:** esta entidad nos sirve para discriminar las conversaciones que están asignadas a un técnico de aquellas que no lo están. Esto nos sirve, de manera interna, para identificar que técnico o técnicos han atendido una conversación. Se relaciona con la entidad conversación 1:n.

Una vez definidas las entidades construimos el servicio mediante la opción Service Builder de Eclipse, el cual nos permite crear la lógica del negocio y nos genera una base de datos que dejará constancia de las conversaciones abiertas (pendientes y en curso) y cerradas.

Nuestro proyecto consta de dos Portlets, uno por el lado del cliente y otro por lado de soporte técnico:

- El Portlet cliente: el cliente se registrará con un alias y seleccionará una categoría de conversación, entrando al chat.
- El Portlet de soporte: el técnico accederá a una primera vista donde podrá ver las conversaciones pendientes de contestar y las que ya tiene asignadas junto con las que ha sido invitado a participar (por otro técnico). Una vez seleccione una conversación, se abre el chat del lado soporte donde tiene las siguientes funcionalidades extras:
 - Incluir a otro técnico en la conversación.
 - Crear un archivo pdf con la conversación.
 - Adjuntar archivos.
 - Antes de cerrar la conversación deberá etiquetar y añadir una pequeña descripción de la misma para facilitar búsquedas posteriores.

La comunicación entre Portlets se ha realizado mediante consulta por Ajax contra la base de datos. La frecuencia de la consulta se hace con un temporizador en JavaScript, con una frecuencia de comprobación de un segundo.

Para definir la apariencia de los Portlets hemos desarrollado 3 Themes uno para Quental, otro para Telefónica y el último para Cruz Roja. Los Themes nos ayudan con la maquetación básica del portal y su funcionalidad. En cada uno de los Theme se ha intentado respetar la identidad corporativa de cada una de las empresas. Aparte se ha diseñado un pequeño logotipo para el Chat. Se ha utilizado velocity (CSS3 y HTML5). Para la arquitectura de Liferay se ha creado la siguiente estructura:

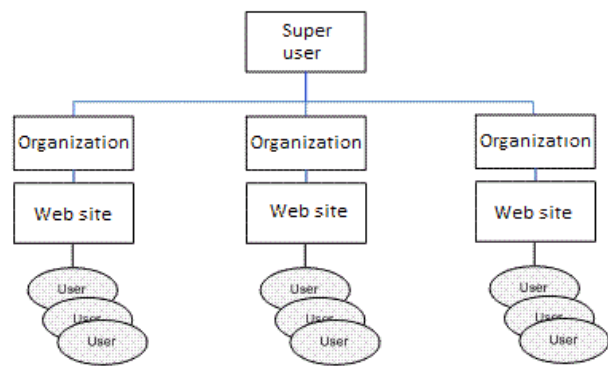


Figura 1. Esquema de las organizaciones

- Tres organizaciones: Quental, Cruz Roja y Telefónica.
- Cada organización cuenta con un sitio web.
- Dos grupos de usuarios: uno de técnicos para Quental y otro de usuarios para Cruz Roja y Telefónica.
- Roles: los usuarios tienen rol de Guest y existe un rol de superadministrador que tiene acceso para editar las tres organizaciones.

Para el desarrollo del trabajo, los diez integrantes del equipo nos hemos organizado con metodologías ágiles, se ha trabajado por pares y se ha coordinado todo el trabajo mediante la herramienta de gestión de proyectos Trello.

6. Conclusiones

Con el trabajo realizado por el equipo se ha conseguido obtener un prototipo de chat funcional que podrá ser implementado en cualquier portal que quiera incluir un soporte técnico, además de poder desarrollar tantas funcionalidades extras como se necesiten.

7. Agradecimientos

Agradecemos a Telefónica la oportunidad de entrar en el programa Talentum Empleo. A Quental por permitirnos desarrollar un proyecto real. A The Bold Choice por facilitarnos las herramientas necesarias para el trabajo en equipo. A Thinking with you por las clases de agilismo y Scrum. Y un último pero no menos importante gracias a todos los componentes del equipo ChatO, por lo aprendido, lo discutido y lo leído.