



Lynoda

Lydia Reina Perea
Juan Antonio Rodríguez Rodríguez
David Ruiz Carmona

Contenido

1.	Descripción del problema a resolver.....	2
2.	Requerimientos del problema	3
3.	Esquema funcional	4
4.	Diseño de la base de datos.....	6
6.	Implementación de la aplicación	10
7.	Desarrollo de casos de prueba	17
8.	Extras.....	21
9.	Datos de acceso a administrador	21

1. Descripción del problema a resolver

Lynoda es, en pocas palabras, una plataforma social enfocada al ocio entre usuarios con un mismo interés.

Nuestra aplicación pretende ser un punto de encuentro entre personas con intereses comunes, de tal forma que los usuarios pueden debatir sobre temas específicos, así como enviar si lo prefieren, mensajes privados entre ellos.

Es un agregador de noticias donde los usuarios podrán realizar publicaciones, estas podrán ser votadas a favor o en contra por otros usuarios, a su vez podrán realizar comentarios de dichas publicaciones. Dichas publicaciones pertenecerán a una temática concreta.

Más concretamente, habrá tres tipos de usuarios que interaccionen con la aplicación.

Primero nos encontramos con los usuarios no registrados en el sistema, que podrán acceder a información que la aplicación web ofrece a modo de información sobre la actividad que esta lleva a cabo (información general, como funciona, objetivo de la aplicación, ...). También, podrán visualizar las publicaciones sin poder votar, ni ver los comentarios realizados a dicha publicación. Además, podrá acceder a el formulario de registro, donde procederá a registrarse en el sistema.

El usuario podrá loguearse en el sistema y pasará a ser un usuario logueado o usuario administrador.

Los usuarios logueados, podrán escribir publicaciones, donde otros usuarios podrán realizar comentarios y votar dicha publicación, pudiendo este modificarla en cualquier momento. Los comentarios escritos referentes a una publicación también serán votados por otros usuarios, definiéndose el orden de aparición en base a estos votos.

Un usuario logueado podrá suscribirse a distintas publicaciones, mostrándose en la página principal de usuario las novedades que hubiera en estas.

También tendrán un chat en el que podrán mantener conversaciones privadas con otros usuarios logueados en el sistema.

El usuario administrador tendrá un menú totalmente distinto al resto de usuarios donde podrá gestionar los diferentes elementos que componen en sistema, pudiendo así crear temáticas, eliminar cualquier componente del sistema, modificar usuarios, etc.

2. Requerimientos del problema

Objetivo 1 – Login de usuarios:

El sistema permitirá el acceso a los usuarios registrados en el sistema. Existirán 2 grupos de usuarios:

- Administrador. Tendrá permisos para realizar modificaciones entre las entidades del sistema.
- Usuario registrado. Tendrá acceso a la visualización y creación de publicaciones, así como a la realización de comentarios y votación de las publicaciones.

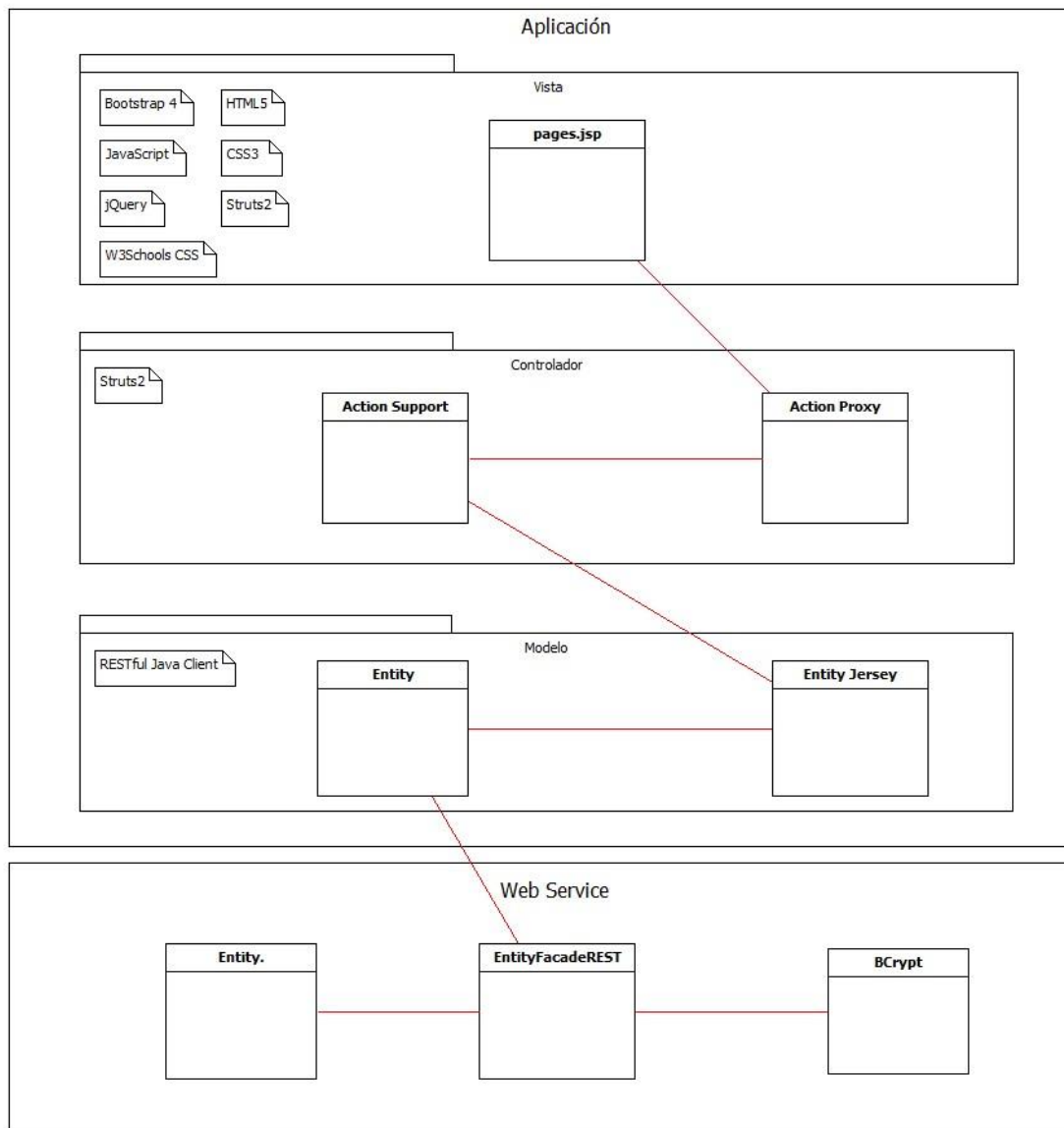
Objetivo 2 – Gestión de temática y publicaciones:

El sistema deberá permitir la gestión de temáticas, publicaciones y comentarios. Las publicaciones pertenecerán a una temática concreta. Las publicaciones podrán tener comentarios, teniendo ambos una puntuación obtenida por los votos realizados por los usuarios registrados en el sistema. Se permitirá la creación, eliminación y modificación. También se podrán suscribir a distintas publicaciones.

Objetivo 3 – Gestión de chat:

El sistema deberá permitir la gestión de chats y mensajes privados. Los chats podrán ser creados por los usuarios registrados para poder establecer conversaciones privadas entre dos usuarios registrados. Se permitirá la creación, eliminación y modificación de los chats, así como la visualización, creación y modificación de los mensajes privados que lo componen.

3. Esquema funcional



El proyecto está dividido en dos sistemas, el primero que será la aplicación en sí, con la lógica de negocio, y el segundo, que es un web service que proporciona el acceso a los datos que usa el primero.

La aplicación sigue el patrón arquitectónico MVC, dividiendo el modelo, la vista y el controlador en tres capas diferenciadas. Cada una de las capas se explican a continuación:

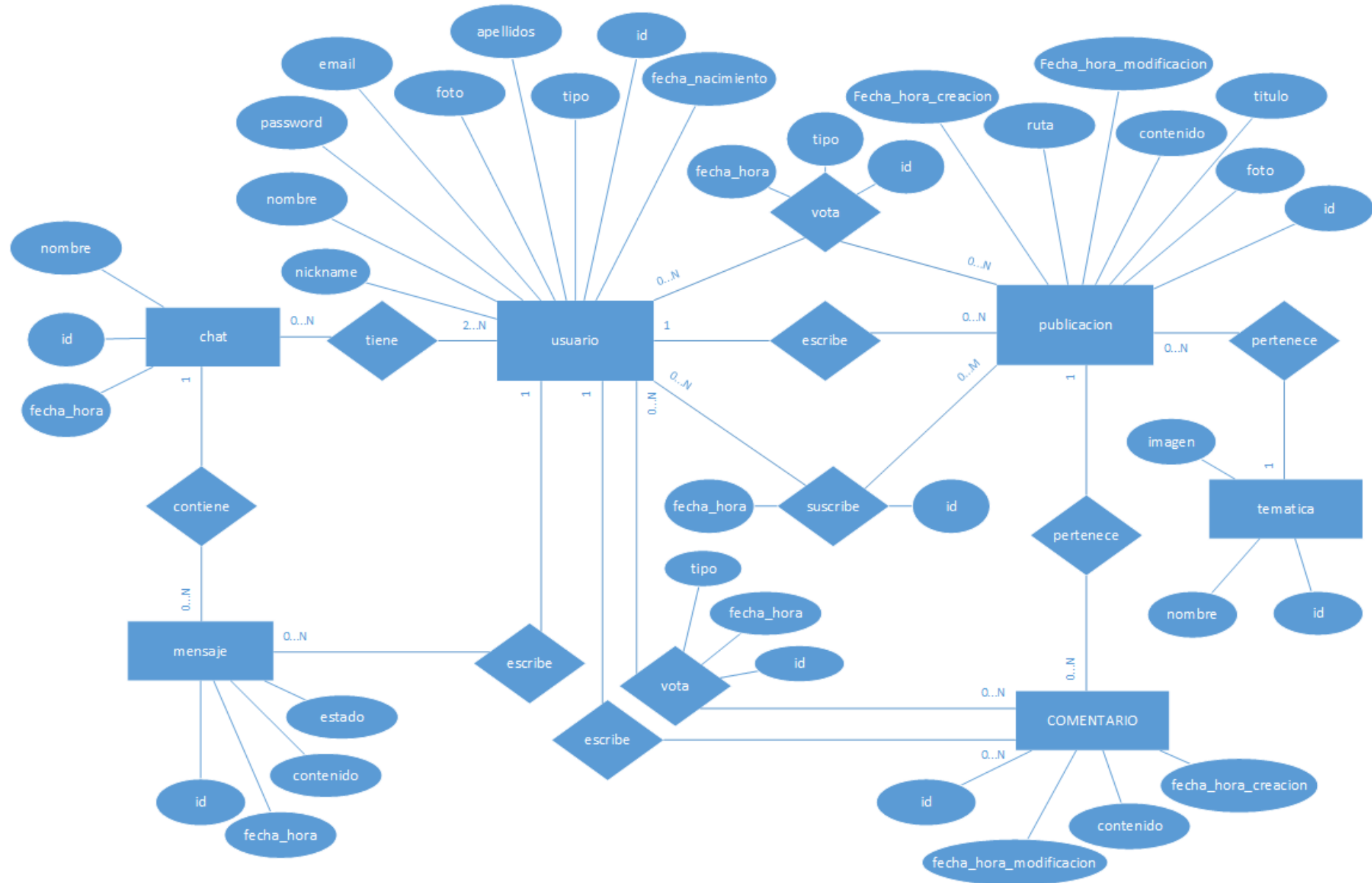
- **Vista:** En esta capa se muestran los datos y la interfaz gráfica con la que interactúan los usuarios del sistema. En esta capa se han utilizado diferentes tecnologías para la maquetación del contenido, e interactúa directamente con la siguiente capa, la capa del controlador.
- **Controlador:** Ésta capa hace de intermediario entre la vista y el modelo, gestionando las peticiones que se hacen desde la vista y proporcionando los datos recogidos de la capa de Modelo a la vista. En esta capa se hace uso de Struts2, y encontramos los siguientes elementos:

- Action Proxy: Se comunica directamente con la vista, recoge los envíos en formularios de la vista y llama a el Action correspondiente en función de la llamada recibida.
 - Action Support: Clase java que se comunica con directamente con la capa de modelo, haciendo uso de esta para persistir o recoger datos.
- Modelo: En esta capa se encuentran las entidades del sistema, con sus métodos y atributos. Nos encontramos dos tipos de entidades en esta capa:
 - EntityJersey: Entidad creada a partir de la información proporcionada por el WADL del web service, que contiene los métodos implementados en el web service del que hace uso.
 - Entity: Entidad que contiene los métodos y atributos de las entidades reales de la base de datos. Son usadas para la encapsulación de los datos recibidos o enviados al web service a través de una entidad EntityJersey.

El Web Service es de tipo REST. Se ha decidido usar este tipo de web service por la facilidad de uso en comparación a SOAP, ya que REST se encarga de acceder a la base de datos por sí sólo, sin necesidad de otro framework como hibernate. Se encarga del acceso a la base de datos que nutre la aplicación, y dentro de éste encontramos distintos componentes:

- Entity: Entidad que contiene los métodos y atributos de las entidades reales de la base de datos. Son usadas para la encapsulación de los datos recibidos desde la aplicación.
- EntityFacadeREST: Clase java que accede directamente a la base de datos mediante lenguaje JQPL. Usa las clases Entity para la encapsulación de los datos que entran o salen de la base de datos.
- BCrypt: Clase java que proporciona funcionalidades para encriptar y comprobar claves del que hace uso las entidades EntityFacadeREST.

4. Diseño de la base de datos



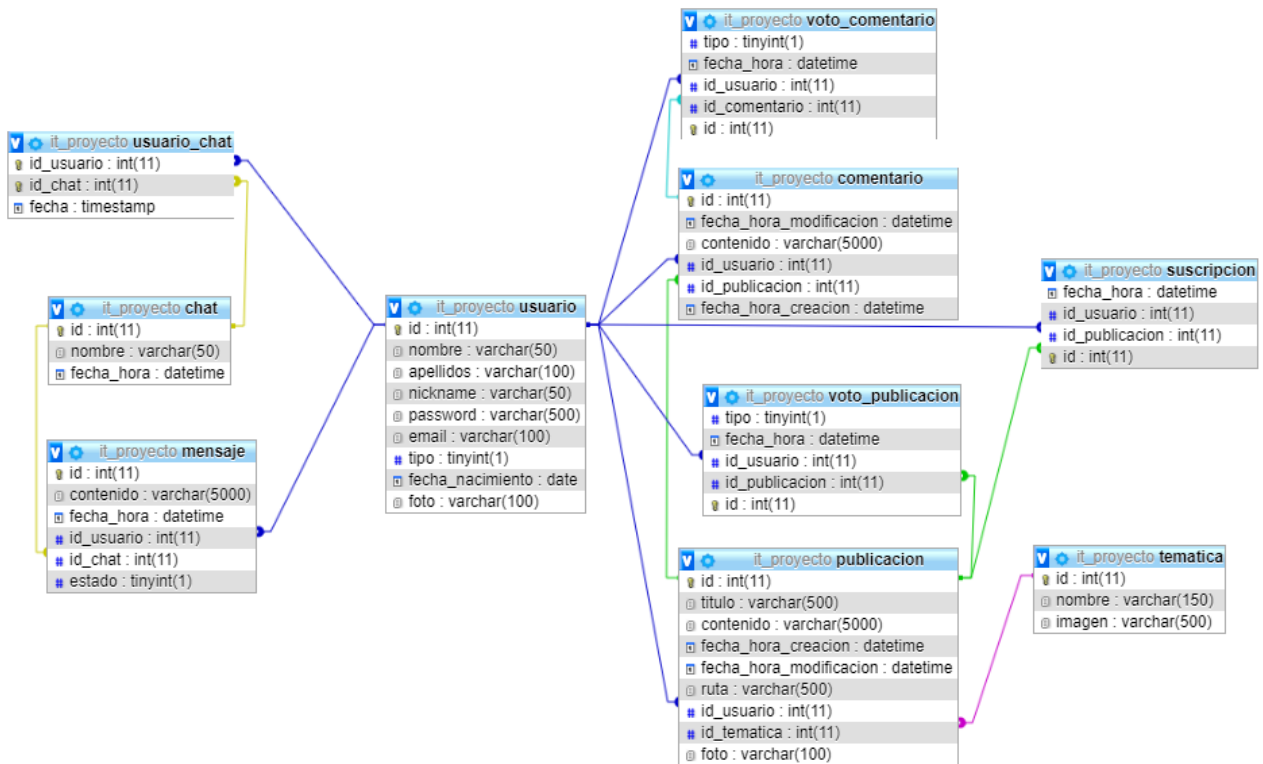
Entidades:

- chat: Entidad que corresponde a una conversación entre varios usuarios del sistema. Contiene los mensajes emitidos por los usuarios que integran el chat.
- mensaje: Entidad correspondiente a un mensaje privado enviado por un usuario y que pertenece a un chat.
- usuario: Entidad correspondiente a los usuarios de sistema, tanto usuarios normales como administradores.
- publicacion: Entidad que corresponde a un post creado por un usuario y al cual pueden acceder los demás usuarios del sistema. Esta entidad contiene una serie de comentarios registrados por los usuarios.
- tematica: Entidad que representa un tipo de contenido al que serán asociadas las distintas publicaciones del sistema.
- comentario: Entidad que representa la respuesta de los distintos usuarios referentes a una publicación.

Relaciones:

- mensaje-chat: Un chat tendrá asociados distintos mensajes.
- usuario-chat: Un usuario tendrá asociados distintos chats, y a su vez, el chat tendrá asociados distintos usuarios registrados en el sistema.
- usuario-mensaje: Un mensaje estará asociado con un único usuario registrado en el sistema.
- usuario-comentario (escribe): Un usuario estará relacionado con un comentario que éste ha escrito.
- usuario-comentario (vota): Un usuario votará un comentario, quedándose así registrado el voto, y a su vez, el comentario está relacionado con el voto emitido por el usuario.
- usuario-publicacion (vota): Un usuario votará una publicación, quedándose así registrado el voto, y a su vez, la publicación está relacionada con el voto emitido por el usuario.
- usuario-publicacion (escribe): Una publicación estará relacionada con un único usuario, señalando que éste último es el autor de dicha publicación.
- usuario-suscribe (suscripción): Un usuario está relacionada con una publicación, quedando así registrada la suscripción en el sistema, y a su vez, la publicación también estará relacionada con el usuario, dejando así la relación bilateral de las entidades registrada.
- publicacion-comentario: Un comentario está relacionado con una publicación determinada, dejando constancia de que comentario pertenece a que publicación.
- publicacion-temática: una publicación está relacionada con una temática, dejando constancia de que temática pertenece a que publicación.

5. DISEÑO FÍSICO



USUARIO		
Atributo	Tipo	Descripción
id	INT(11) PK	Identificador único del usuario
nombre	VARCHAR(50)	Nombre del usuario
apellidos	VARCHAR(100)	Apellidos del usuario
nickname	VARCHAR(50)	Nick name del usuario
password	VARCHAR(500)	Contraseña del usuario
email	VARCHAR(100)	Email del usuario
tipo	TINYINT(1)	Tipo de usuario (1 administrados, 0 usuario normal)
fecha_nacimiento	DATE	Fecha de nacimiento de usuario

CHAT		
Atributo	Tipo	Descripción
id	INT(11) PK	Identificador único del chat
nombre	VARCHAR(50)	Nombre del chat
fecha_hora	DATETIME	Fecha y hora de creación del chat

USUARIO_CHAT		
Atributo	Tipo	Descripción
id_usuario	INT(11) PK FK	Identificador único del usuario
id_chat	INT(11) PK FK	Identificador único del chat
fecha	TIMESTAMP	Fecha y hora de creación del registro

MENSAJE		
Atributo	Tipo	Descripción
id	INT(11) PK	Identificador único del mensaje
contenido	VARCHAR(5000)	Texto del contenido del mensaje
fecha_hora	DATETIME	Fecha y hora de creación del mensaje
Id_usuario	VARCHAR(50)	Identificador único del usuario
Id_chat	VARCHAR(500)	Identificador único del chat
estado	TINYINT(1)	Estado en el que se encuentra el mensaje

COMENTARIO		
Atributo	Tipo	Descripción
id	INT(11) PK	Identificador único del comentario
fecha_hora_modificacion	DATETIME	Fecha y hora de modificación
contenido	VARCHAR(5000)	Contenido del comentario
id_usuario	INT(11) FK	Identificador único del usuario
id_publicacion	INT(11) FK	Identificador único de la publicación
fecha_hora_creacion	DATETIME	Fecha y hora de creación del comentario

VOTO_COMENTARIO		
Atributo	Tipo	Descripción
tipo	TINYINT(1)	Tipo de voto (1 positivo, 0 negativo)
fecha_hora	DATETIME	Fecha y hora creación del voto
id_usuario	INT(11) FK	Identificador único del usuario
id_comentario	INT(11) FK	Identificador único del comentario
id	INT(11) PK	Identificador único del voto

PUBLICACIÓN		
Atributo	Tipo	Descripción
id	INT(11) PK	Identificador único de la publicación
titulo	VARCHAR(500)	Título de la publicación
comentario	VARCHAR(5000)	Texto del contenido de la publicación
fecha_hora_creacion	DATETIME	Fecha y hora de creación de la publicación
fecha_hora_modificacion	DATETIME	Fecha y hora de modificación de la publicación
ruta	VARCHAR(500)	Ruta en el sistema del fichero adjunto a la publicación
id_usuario	INT(11) FK	Identificador único del usuario
id_tematica	INT(11) FK	Identificador único de la temática
foto	VARCHAR(100)	Ruta de la foto de la publicación

VOTO_PUBLICACIÓN		
Atributo	Tipo	Descripción
tipo	TINYINT(1)	Tipo de voto (1 positivo, 0 negativo)
fecha_hora	DATETIME	Fecha y hora creación del voto
id_usuario	INT(11) FK	Identificador único del usuario
id_publicación	INT(11) FK	Identificador único de la publicación
id	INT(11) PK	Identificador único del voto

SUSCRIPCIÓN		
Atributo	Tipo	Descripción
id	INT(11) PK	Identificador único de la suscripción
fecha_hora	DATETIME	Fecha y hora creación de la suscripción
id_usuario	INT(11) FK	Identificador único del usuario
id_publicación	INT(11) FK	Identificador único de la publicación

TEMÁTICA		
Atributo	Tipo	Descripción
id	INT(11) PK	Identificador único de la temática
nombre	VARCHAR(150)	Nombre de la temática
imagen	VARCHAR(500)	Ruta en el servidor de la imagen asociada a la temática

6. Implementación de la aplicación

El proyecto consta de dos partes: la aplicación y el web service que proporciona los datos.

Proyecto_it

Es la aplicación, que contiene la lógica de negocio y que lleva a cabo las funcionalidades haciendo uso del web service, que proporciona el acceso a los datos. Dicha aplicación será con la que interactúa el usuario.

Vistas

Vistas de usuarios no logueados:

- cabeceraNoLogado: muestra la barra de navegación para los usuarios no logueados en la aplicación, dando las opciones de inicio, temas, publicaciones, registrarse y login.
- index: página principal de la aplicación, en la que se muestra la portada de la página y las temáticas más activas.
- publicacionesNoLogado: página en la que se muestran todas las publicaciones de la aplicación, mostrando el número de votos positivos y negativos, el número de comentarios y la fecha de creación. Pidiendo al usuario que se logue si desea visualizar el contenido y comentarios de la publicación.
- temas: página que muestra todas las temáticas de la aplicación.
- registro: página en la que el usuario no logueado podrá registrarse.
- login: venta emergente en la que el usuario se loguea para acceder a las funcionalidades de la aplicación.

Vista de usuarios logueados:

- cabeceraLogado: muestra la barra de navegación para los usuarios logueados en la aplicación, dando las opciones de home, temas, mis publicaciones, perfin, chat y logout.
- home: página que muestra las publicaciones en las que el usuario logueado está suscrito, mostrándolas según el número de comentario.
- temas: página que muestra todas las temáticas de la aplicación.

- publicaciones: página a la que se accede desde temas seleccionando una temática. Muestra todas las publicaciones que pertenecen a dicha temática.
- publicacion: se accede al seleccionar una publicación. Muestra el contenido de la publicación seleccionada y permite visualizar y realizar comentarios.
- misPublicaciones: página que muestra las publicaciones que el usuario logueado ha creado y la eliminación de estas, permitiendo también la creación de nuevas publicaciones.
- addPublicacion: página a la que se accede desde misPublicaciones. Permite la creación de nuevas publicaciones en la aplicación.
- perfil: página que muestra los datos del usuario logueado, permitiendo la modificación de alguno de sus datos si lo desea y la eliminación de su cuenta en la aplicación.
- chats: página que muestra los chats activos que el usuario tiene, permite crear o modificar dichos chats y mostrar el chat para enviar mensajes privados con otro usuario.

Vista de administrador:

- cabeceraAdmin: muestra la barra de navegación para el administrador de la aplicación, dando las opciones de usuarios, temáticas, publicaciones y logout.
- usuariosAdmin: página que muestra todos los usuarios dados de alta en la aplicación, permitiendo la eliminación y edición de estos.
- editarUsuarioAdmin: página a la que se accede desde usuarioAdmin. Permite la edición del usuario.
- tematicasAdmin: muestra todas las temáticas de la aplicación, dando la opción de eliminar, editar y añadir nuevas temáticas.
- addTematica: página a la que se accede desde tematicasAdmin, permitiendo la creación de nuevas temáticas.
- editarTemática: página a la que se accede desde tematicasAdmin, permitiendo la edición de la temática seleccionada.
- publicacionesAdmin: muestra todas las publicaciones de la aplicación, permitiendo la eliminación de estas.

Acciones

- menu_top: contiene las acciones que se ejecutan en la barra de navegación de los usuarios no logueados. Tiene los siguientes métodos:
 - inicio: Método para obtener las temáticas más populares.
 - temas: Método para obtener todas las temáticas del sistema.
 - publicaciones: Método para obtener todas las publicaciones del sistema.
 - filtroPubli: Método para filtrar las publicaciones por título.
 - registro: Método que te lleva a la página registro.
- menu_top_logueado: contiene las acciones que se ejecutan en la barra de navegación de los usuarios logueados. Tiene los siguientes métodos:
 - chats: Método para mostrar todos los chats del usuario.
 - misPublicaciones: Método para listar todas las publicaciones que ha creado el usuario logueado.
 - home: Método para listar las publicaciones en las que un usuario está suscrito.

- menu_top_admin: contiene las acciones que se ejecutan en la barra de navegación del usuario administrador. Tiene los siguientes métodos:
 - usuarios: Metodo para obtener una lista de todos los usuarios del sistema.
 - publicacionesAdmin: Metodo para obtener todas las publicaciones del sistema.
 - tematicasAdmin: Metodo para obtener todas las tematicas del sistema.
- loginLogout: acciones para loguearse y desloguearse en la aplicación. Contiene los siguientes métodos:
 - login: Método loguearse en la aplicación, guarda al usuario en la session y guardar el id del usuario en una cookie.
 - logout: Método para desloguearse de la aplicación, eliminar la session y las cookies del usuario.
- PublicacionAccion:
 - verPublicacion: Método para obtener todos los comentarios de una publicación.
- MostrarPublicacion: conjunto de acciones para visualizar publicaciones. Contiene los siguientes métodos:
 - editUser: Método que hace uso del web service para obtener un usuario con el id indicado.
 - perfil: Método para obtener el usuario que se ha logeado y postrarlo en la vista de perfil.
 - filtrarUser: Método que hace uso del web service para obtener una lista de los usuarios que cumpla con la condición del filtro.
 - removeUser: Método para eliminar un usuario de la base de datos.
 - removeMiCuenta: Método para eliminar la cuenta del usuario logueado.
- MostrarTematicas: conjunto de acciones para visualizar temáticas. Contiene los siguientes métodos:
 - editTematica: Método que hace uso del web service para obtener una temática con el id especificado y mostrarlo en la página editarTemática.
 - filtrarTema: Método que hace uso del web service para obtener una lista de las temáticas que cumplen con la condición del filtrado.
 - removeTematica: Método para eliminar una temática de la base de datos.
- MostrarPublicaciones: conjunto de acciones para visualizar publicaciones. Contiene los siguientes métodos:
 - removePublicacion: Método para eliminar una publicación.
 - filtroPubli: Método para filtrar las publicaciones por el título.
 - filtrarPublicacionesTemas: Método para filtrar las publicaciones por el título. Este método se hace uso desde la parte del administrador.
 - filtroMisPublicaciones: Método para filtrar las publicaciones por el título que el usuario ha creado.
 - filtroPublicacionesSuscrito: Método para filtrar las publicaciones por el título en las que está suscrito el usuario.

- publicacionesByTema: Método para obtener las publicaciones asociadas a un tema.
 - publicacionesByTemaAdmin: Método para obtener todas las publicaciones de una temática.
- GestionVotoSuscripcion: conjunto de acciones para votar, suscribirte y dejar de seguir publicaciones. Contiene los siguientes métodos:
 - votoPositivoPublicacion: Método que hace uso del Web Service para permite que un usuario vote positivamente una publicación.
 - votoNegativoPublicacion: Método que hace uso del Web Service para permite que un usuario vote negativamente una publicación.
 - unFollowPublicacion: Método que hace uso del Web Service para permite que un usuario deje de seguir una publicación de la que fue suscrito.
 - followPublicacion: Método que hace uso del Web Service para permite que un usuario sea seguidor de una publicación.
- GestionUsuario: conjunto de acciones para la gestión de los usuarios. Contiene los siguientes métodos:
 - editUserPersistencia: Método que hace uso del web service para guardar las modificaciones de un usuario en la base de datos.
 - addUsuario: Método que hace uso del web service para guardar a un nuevo usuario en la base de datos.
 - Validate: Método para validar los campos del formulario de registrar, perfil y editar usuario del administrador.
- GestionTematica: conjunto de acciones para la gestión de las temáticas. Contiene los siguientes métodos:
 - editTemaPersistencia: Método que hace uso del web service para guardar las modificaciones de la temática en la base de datos.
 - addTemaPersistencia: Método que hace uso del web service para guardar la nueva temática en la base de datos.
 - Validate: Método para validar los campos del formulario de nueva temática y editar temática del administrador.
- GestionPublicacion: conjunto de acciones para la gestión de las publicaciones. Contiene los siguientes métodos:
 - addPublicacion: Método que hace uso del web service para guardar la nueva publicación en la base de datos.
 - Validate: Método para validar los campos del formulario de nueva publicación.
- ComentarioAccion: conjunto de acciones para la gestión de los comentarios. Contiene los siguientes métodos:
 - votoPositivo: Método que hace uso del Web Service para permite que un usuario vote positivamente un comentario.
 - votoNegativo: Método que hace uso del Web Service para permite que un usuario vote negativamente un comentario.

- comentarPublicacion: Método que permite a un usuario poner un comentario a una publicación.
 - Validate: Método para validar los campos a la hora de crear un nuevo comentario.
- ChatAccion: conjunto de acciones para la gestión de los chats. Contiene los siguientes métodos:
- crearChat: Método que hace uso del Web Service para crear un nuevo chat entre dos personas.
 - borrarChat: Método que hace uso del Web Service para borrar un chat.
 - editarChat: Método que hace uso del Web Service para editar un chat.
 - Validate: Método para validar los campos a la hora de la modificación y creación de un nuevo chat.

WS

Se encuentran todos los métodos del web service y los pojos de la base de datos.

webService

Web service que proporciona el acceso a los datos.

- En el paquete ws, se encuentran todo los pojos de la base de datos.
- BCrypt: clase para encriptar y desencriptar las contraseñas de los usuarios.
- ChatFacadeREST: conjunto de métodos para la gestión de los chats.
 - getChatsUsuario: Devuelve los chats de los que es participante el usuario con id pasado como parámetro.
 - getUserById: Devuelve un usuario que coincide con el id pasado como parámetro.
 - getChatByNameAndTime: Devuelve un chat que coincide con el nombre y tiempo pasado como parámetro.
 - getUsuariosDeUnChat: Devuelve una lista de usuarios asociado al chat a un chat.
 - getChatById: Devuelve un chat que coincide con el id pasado como parámetro.
 - getChatPorNombre: Devuelve todos los chats que coincide con el nombre pasado como parámetro.
- ComentarioFacadeREST: conjunto de métodos para la gestión de los comentarios.
 - getComentariosPublicacion: Devuelve una lista de comentarios pertenecientes a una publicación que coincide con el id pasado como parámetro.
 - getPublicacionById: Devuelve una publicación que coincide con el id pasado como parámetro.
 - getUltimoComentarioPublicacion: Método que devuelve el último comentario de una publicación.
 - getNumComentarios: Método para obtener el número de comentarios de una publicación.

- MensajeFacadeREST: conjunto de métodos para la gestión de los mensajes.
 - getMensajesChat: Devuelve los mensajes que pertenecen al chat con el id pasado como parámetro.
 - getChatById: Devuelve un chat cuyo id corresponde con el id pasado como parámetro.

- PublicacionFacadeREST: conjunto de métodos para la gestión de las publicaciones.
 - getPublicacionPorTitulo: Método para obtener las publicaciones cuyo título corresponda con el pasado como parámetro.
 - getUsuario: Método auxiliar para obtener un usuario a partir del identificador.
 - getMisPublicaciones: Método para obtener todas las publicaciones de un usuario, ordenandos de forma descendente.
 - getPublicacionesByTema: Devuelve un listado de publicaciones perteneciente a la temática que coincide con el id pasado como parámetro.
 - getPublicacionesSuscritoOrdenadoNumComentarios: Obtiene las publicaciones a las que está suscrito un usuario y ordenadas descendientemente por número de comentarios.
 - getTematicaById: Devuelve una temática que coincide con el id pasado como parámetro.
 - getNumPublicacionesByTematica: Devuelve el número de publicaciones perteneciente a la temática que coincide con el id pasado como parámetro.
 - getMisPublicacionesPorTitulo: Método para obtener las publicaciones cuyo título corresponda con el título pasado como parámetro correspondiente a las publicaciones que el usuario ha creado.
 - getPublicacionesSuscritoPorTitulo: Devuelve las publicaciones en las que está suscrito el usuario con id pasado como parámetro y filtradas por el titulo proporcionado por parámetro.

- SuscripcionFacadeREST: conjunto de métodos para la gestión de las suscripciones.
 - getPublicacionSuscrito: Método para obtener todas las publicaciones en las que el usuario esta suscrito.
 - getSuscripcion: Método para obtener una suscripción a partir del identificador del usuario y la publicación.

- TematicaFacadeREST: conjunto de métodos para la gestión de las temáticas.
 - getTematicasPorNombre: Método para obtener las temáticas cuyo nombre corresponda con el nombre pasado como parámetro.
 - getTematicasMasPopulares: Devuelve las temáticas más populares del sistema.
 - encontrarTematica: Método para obtener las temáticas cuyo nombre corresponda con el nombre como parámetro.

- UsuarioFacadeREST: conjunto de métodos para la gestión de los usuarios.
 - getUsuariosPorNombre: Método para obtener los usuarios cuyo nombre corresponda con el nombre pasado como parámetro.
 - Login: Método para comprobar si el usuario y password proporcionada por el usuario esta registrado en el sistema.

- getUsuarioByUsername: Método que devuelve un usuario cuyo nickname corresponda con el proporcionado como parámetro.
 - getUsuarioTipo: Metodo que devuelve una lista de usuarios de tipo 0 (todos menos los administradores)
 - getUsuariosChat: Devuelve una lista de los usuarios que estan asociado a un chat cuyo id corresponda con uno de los usuarios del chat.
 - existeNickname: Método para comprobar si ya existe un usuario con el nickname pasado por parámetro.
- VotoComentarioFacadeREST: conjunto de métodos para la gestión de los votos de los comentarios.
 - getVotosPositivos: Devuelve el número de votos positivos que tiene el comentario con el id pasado como parámetro.
 - getVotosNegativos: Devuelve el número de votos negativos que tiene el comentario con el id pasado como parámetro.
 - getComentarioById: Devuelve un Comentario que coincide con el id pasado como parámetro.
 - haVotado: Método para probar si un usuario a votado un comentario o no.
 - VotoPublicacionFacadeREST: conjunto de métodos para la gestión de los votos de las publicaciones.
 - getVotosNegativos: Método para obtener el número de votos negativos de una publicación. Tipo 0 representa un voto negativo.
 - getVotosPositivos: Método para obtener el número de votos positivos de una publicación. Tipo 1 representa un voto positivo.
 - getVotoPublicacion: Método para obtener un voto de la publicación a partir del identificador del usuario y la publicación.
 - existeVotoPublicacion: Método para comprobar si ya se ha votado la publicación.

7. Desarrollo de casos de prueba

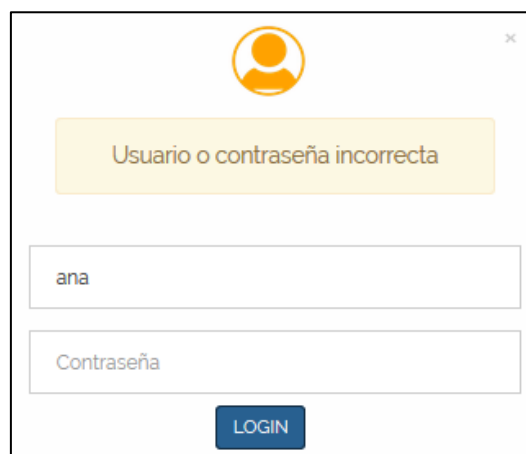
Casos de prueba en login:

Loguearse con campos vacíos.



A screenshot of a login form. At the top is a circular profile icon placeholder. Below it is a red error message box that says "El campo usuario es obligatorio". Underneath is an empty text input field labeled "Nombre de Usuario". Below that is another red error message box that says "La contraseña es obligatoria". Underneath is an empty text input field labeled "Contraseña". At the bottom is an orange button labeled "LOGIN".

Usuario o contraseña incorrectos.



A screenshot of the same login form. The "Nombre de Usuario" field now contains the text "ana". The "Contraseña" field is empty. A yellow error message box at the top says "Usuario o contraseña incorrecta". The "LOGIN" button is now blue.

Casos de prueba en las publicaciones:

Añadir publicación con los campos vacío.

Nueva Publicación

El título es obligatorio

Título:

Foto:

Examinar... No se ha seleccionado ningún archivo.

El contenido es obligatorio

Contenido:

Formato no correcto

Enlace:

Temática:

Animales

Crear

Añadir publicación con formato de campo enlace incorrecto.

Nueva Publicación

Título:

Foto:

Examinar... No se ha seleccionado ningún archivo.

Contenido:

Que bonitos son los gatos

Formato no correcto

Enlace:

Temática:

Animales

Crear

Casos de prueba en usuarios:

Creación y edición de usuario con campos vacíos.

El nombre es obligatorio	El apellido es obligatorio
Nombre: <input type="text" value="Introduzca el nombre"/>	Apellidos: <input type="text" value="Introduzca los apellidos"/>
El nombre de usuario es obligatorio	El email es obligatorio
Nombre de usuario: <input type="text" value="Introduzca el nickname"/>	Email: <input type="text" value="Introduzca el email"/>
La contraseña es obligatoria	La fecha de nacimiento es obligatorio
Contraseña: <input type="text" value="Introduzca la contraseña"/>	Fecha de nacimiento: <input type="text" value="dd/mm/aaaa"/>
<input type="button" value="Enviar"/>	

Creación y edición usuario con nombre ya existente

Nombre: <input type="text" value="Ana"/>	Apellidos: <input type="text" value="Castillo"/>
El nombre de usuario ya existe	Email: <input type="text" value="ana@ana.com"/>
Nombre de usuario: <input type="text" value="ana"/>	Contraseña: <input type="text" value="Introduzca la contraseña"/>
Fecha de nacimiento: <input type="text" value="2/12/96"/>	
<input type="button" value="Enviar"/>	

Creación y edición de usuario con contraseña inferior a cinco caracteres

Nombre: <input type="text" value="Ana"/>	Apellidos: <input type="text" value="Castillo"/>
Nombre de usuario: <input type="text" value="anaCastillo"/>	Email: <input type="text" value="ana@ana.com"/>
Tamaño mínimo 5 caracteres	Fecha de nacimiento: <input type="text" value="2/12/96"/>
Contraseña: <input type="password" value="..."/>	
<input type="button" value="Enviar"/>	

Creación y edición de usuario con formato de campo email incorrecto.

Nombre:	<input type="text" value="Ana"/>	Apellidos:	<input type="text" value="Castillo"/>
Nombre de usuario:	<input type="text" value="anaCastillo"/>	<div>Formato incorrecto</div>	
Contraseña:	<input type="password" value="Introduzca la contraseña"/>	Email:	<input type="text" value="ana@ana"/>
		Fecha de nacimiento:	<input type="text" value="2/12/96"/>
<div>Enviar</div>			

Casos de prueba en chat:

Creación y edición de chat con campos vacíos.

Nombre del chat

El nombre del chat no puede estar vacío

Usuario

nono

CREAR

Casos de prueba en temática:

Creación y edición de temática con campos vacíos.

Nueva Tematica

El nombre es obligatorio

Título:

Imagen:

La imagen es obligatoria

Examinar...

No se ha seleccionado ningún archivo.

Crear

8. Extras

-Patrón decorador: Usado para ahorrar llamadas al Web Service, ya que algunas veces necesitamos los datos de una entidad, pero además necesitamos algún dato de otra entidad que está relacionado con la primera. Para ello hemos usado una clase java que contiene los datos de ambas entidades, y ésta en sí misma es tratada como una entidad para recoger sus datos.

-Encriptación de claves: Dado el agujero de seguridad que hemos detectado en las llamadas al web service (Ej: en las llamadas al web service desde javascript se puede ver la cadena de conexión con el mismo, y éste puede devolver los datos de un usuario, incluyendo la contraseña), hemos decidido usar una librería que proporciona el cifrado de claves, así como la comprobación de una clave con su hash, de tal forma que nunca sabremos realmente la contraseña del usuario, porque lo que realmente será almacenado en la base de datos será el hash de la encriptación de la clave original.

-Bootstrap: Hemos optado por usar esta tecnología por la facilidad de empleo para darle estilos a una web. Además, ya ofrece la posibilidad de hacer responsive los elementos que se maquetan.

-Tratamiento de ficheros: Veíamos que era una funcionalidad básica que se pudiesen subir ficheros a la web, más concretamente, imágenes. Sin esta funcionalidad la aplicación no sería igual de funcional.

-Subida a un servidor: Era necesario que la aplicación estuviese accesible desde cualquier parte para que los usuarios pudiesen usarla. Se ha subido el web service a un servidor y la aplicación a otra, haciendo uso esta última del web service para acceder a los datos.

9. Datos de acceso a administrador

Usuario	Contraseña
admin	1234