Pokejas! Atrapalas todas

Memoria del proyecto de la asignatura de Tecnologías Web del curso 2018/2019.

El objetivo es crear una página web para anotar las quejas de un vecindario. En nuestro caso, hemos decidido un toque más y ambientarlo en el mundo pokemon de forma que nuestros vecinos sean estos adorables seres.

Autores

Laura Gómez Garrido
Paula Ruiz García

Fichero de Restauracion de la BD.

database.sql

Diseño de la aplicacion.

Requisitos Funcionales

Comunes a todos los usuarios

- Ver incidencia
- Visualizar incidencias
- Visualizar comentarios y valoraciones de cada incidencia
- Realizar nuevos comentarios
- Realizar valoraciones

Visitantes

- Registrarse y darse de alta **(OPCIONAL)** - Pag 6 guión.
- Iniciar sesión

Colaboradores y Administradores

- Crear nuevas incidencias
    - Aquí no es necesario poner la opción de añadir fotografías.

- Cerrar sesión
- Modificar datos personales, todo salvo Rango (y Estado, opcional)
- Ver sus propias incidencias
- Modificar sus propias incidencias
    - Aquí se podrán añadir tantas fotografías como el usuario desee. (Recomendado uso de AJAX y JavaScript)
- Borrar sus propias incidencias

Administradores

- Visualizar usuarios registrados
- Crear nuevo usuario
- Eliminar usuario
- Gestión de perfiles de usuarios.
- Modificar estado de una incidencia
- Consultar eventos del sistema (log)
- Modificar y borrar incidencias de cualquier usuario.
- Crear copia de seguridad de la BBDD.
- Subir copia de seguridad de la BBDD. **(OPCIONAL)**
- Borrar BBDD. **(OPCIONAL)**
- Otras tareas, como administrar BBDD. **¿Especificar cuáles implementamos?**

Log de la Aplicación

- Registro de los eventos principales del sistema, entre ellos:
- Identificación de un usuario.
- Registro de un usuario.
- Cada vez que un usuario modifique la BBDD.
- Cierres de sesión.
- Intentos de identificación erróneos.

Listado de incidencias

Se puede ordenar según los siguientes criterios:

- Antigüedad
- Valoraciones positivas
- Valoraciones positivas netas

Además, se puede filtrar según los siguientes criterios:

- Coincidencia de texto en título, descripción o palabras clave.**(Texto contenido, una única**

caja de texto)

- Lugar de la incidencia **(Desplegable)**
- Estado del mensaje **(Múltiple checkbox)**
- Seleccionar palabra clave **(Desplegable, OPCIONAL)**

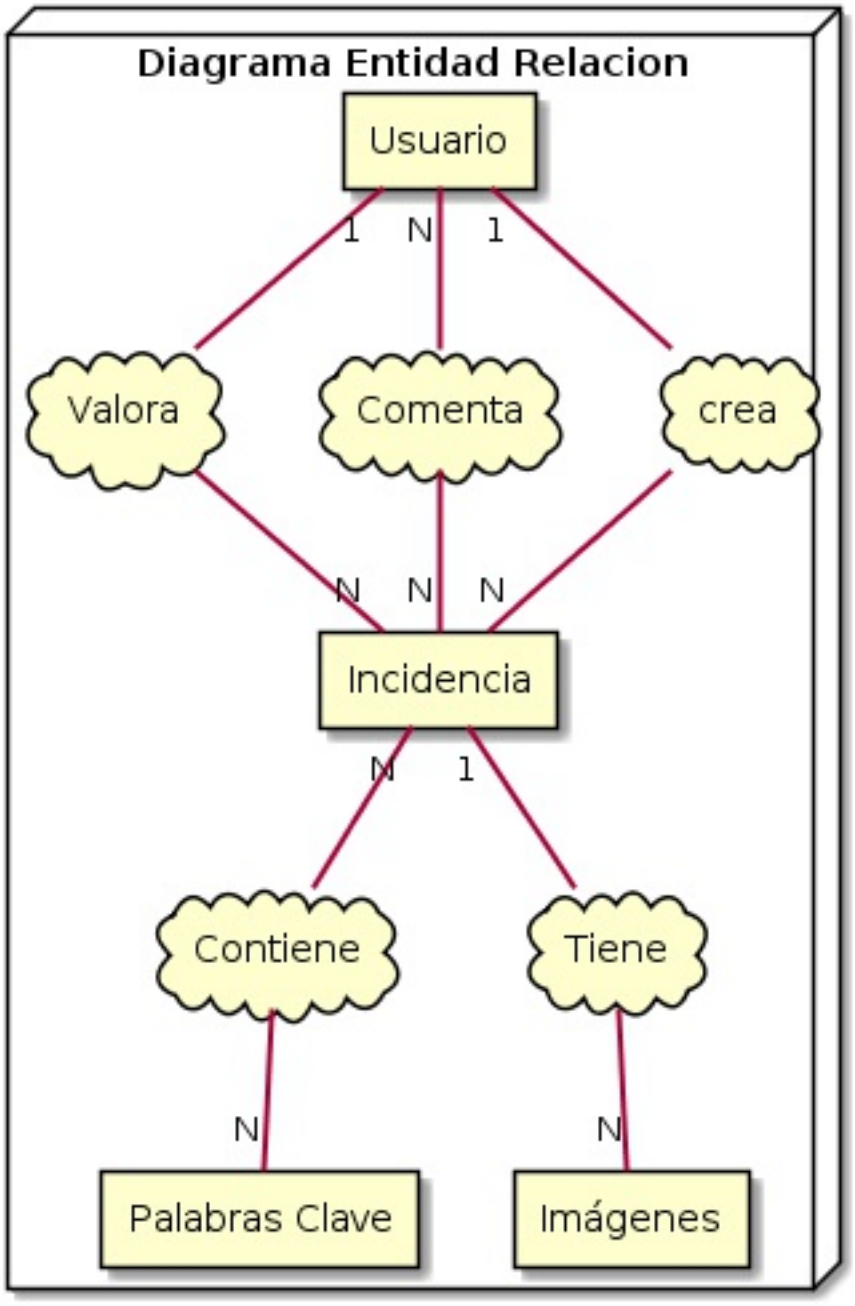
Barra Lateral:

- Ranking N usuarios que más incidencias añaden.
- Ranking N usuarios que más opinan
- Número de inciencias resueltas, pendientes, etc...

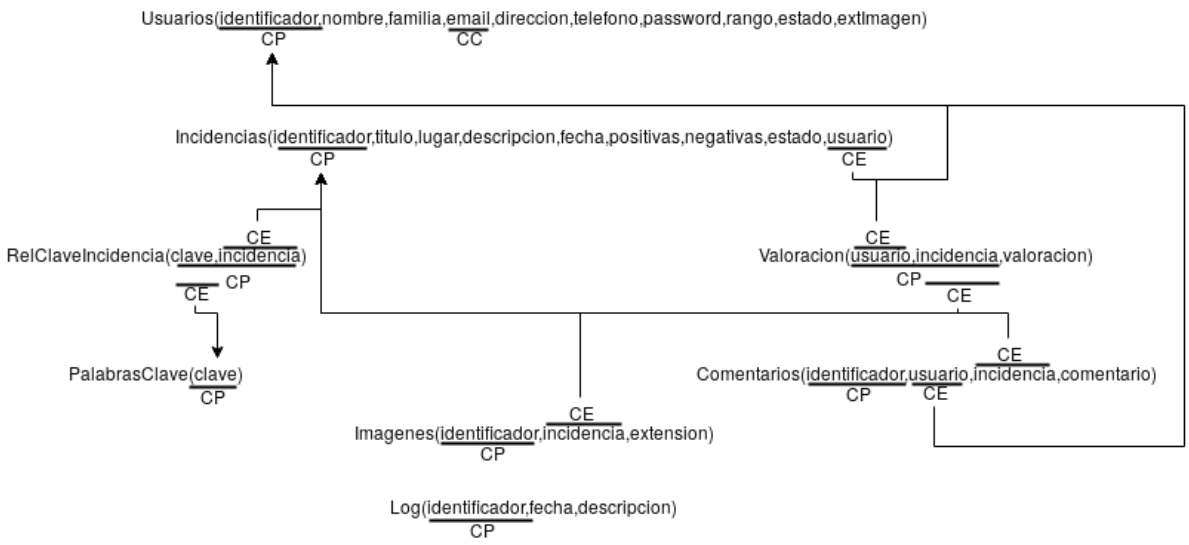
Datos a almacenar

A partir de los anteriores requisitos, esta claro que vamos a tener que almacenar información dentro de una base de datos. Por ello, primeros mostraremos el modelo entidad relación básico que habrá dentro de nuestra base de datos, seguido tendremos el diseño lógico de esta y, finalmente, mostraremos cómo son las tablas que hemos creado para esto junto con una breve descripción de la utilidad de cada una.

Modelo Entidad-Relación



Diseo Lógico



Tablas

Cada una de las categorías se refiere a una tabla en sql por lo que también indicaremos los tipos de cada dato.

Incidencias

Cuando un vecino del municipio detecta el mal funcionamiento de algún servicio, puede proceder a poner una incidencia en el sitio web para que esta sea conocida por otros vecinos y por las autoridades con competencia para su resolución. Dicha incidencia debe contener los siguientes datos:

- Identificador (int) - Primary Key, Autoincrement
- Titulo (varchar(50)) NOT NUll
- Lugar (varchar(50)) NOT NULL
- Descripcion (varchar(500))
- Positivas (int) default 0
- Negativas (int) default 0
- Fecha (timestamp) default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP
- Usuario (int) - Foreign Key
- Estado (Enum - Pendiente, Comprobada, Tramitada, Irresoluble, Resuelta)

Tanto las fotografias, como las palabras clave, las valoracones no anónimas y los comentarios serán almacenados en unas tablas independientesy se conseguirá la información a través de operaciones de tablas.

Usuarios

Estos usuarios son los que van a nutrir de contenido al sitio web. Serán usuarios registrados en el

sistema y tienen la posibilidad de añadir nuevas incidencias y de valorar o comentar las ya existentes. La información que se almacena de ellos es la siguiente:

- Identificador (int) - Primary Key, Autoincrement
- Nombre (varchar (50))
- Apellidos, o en nuestro caso Familia (varchar(20))
- Email (varchar(50))
- Dirección (varchar(100))
- Teléfono (varchar(15))
- Password (varchar(15))
- Rango (Enum - Administrador, Colaborador)
- Estado (Enum - Inactivo, Activo)
- extImagen (VARCHAR(10))

El identificador será también el nombre de las imágenes a la hora de almacenarlo. Funcionamiento similar al de la tabla imágenes. Por eso, almacenamos también la extensión que tiene nuestra imagen a la hora de guardarla, para no tener problemas con el formato de la imagen a la hora de abrirla.

En nuestro caso particular, los nombres de las imágenes serán de la forma `Usuario-{identificador usuario}.{extension}`

Usuarios ya registrados

- email: pikapi@gmail.com contraseña: ashesmiamo
- email: eeveelucion@correo.com contraseña:flareonpresident
- email: growlithe@admin.com contraseña:admin
- email: lindoskitty@gmail.com contraseña:delgatoalhecho

Log

Registro de los eventos principales del sistema, contendrá:

- Identificador (int) - Primary Key, Autoincrement
- Fecha (timestamp) default CURRENT_TIMESTAMP on update CURRENT_TIMESTAMP
- descripcion VARCHAR(300)

Palabras clave

Necesario para facilitar la búsqueda de incidencias dependiendo de las palabras clave que esta contenga. Tendrá una relación con incidencias de muchos a muchos. Esta tabla, es básicamente, para asegurarnos de que las palabras clave que escribimos son únicas.

- Clave (varchar(30)) - Primary Key

RelClaveIncidencia

Necesaria para establecer la relación de muchos a muchos entre incidencias y palabras claves.

- Clave (varchar(30)) - Foreing Key
- Incidencia (int) - Foreign Key
  Juntos, forman la Primary Key.

Imágenes

Una incidencia puede contener muchas imágenes asociadas.

- Identificador (int) - Primary Key, Autoincrement
- Incidencia (int) - Foreign Key
- Extension (VARCHAR(10)) - FOREIGN KEY

La idea es almacenar las imágenes en una carpeta dentro de nuestra web para que así sea más rápida la carga y además así la velocidad de acceso será mayor y el almacenamiento en la base de datos será menor. El identificador de cada imagen en la base de datos servirá para controlar el nombre que tendrá la imagen en la carpeta. No se especifica el nombre de la carpeta donde se almacenará para que así sea el propio progamador web quien pueda decidir donde almacenarlo.

En nuestro caso particular, los nombres de las imágenes serán de la forma `Incidencia-{identificador incidencia}-{identificador imagen}.{extension}` y estarán almacenadas en la carpeta imágenes.

Valoración

Relación de una valoración realizada por un usuario respecto a una incidencia concreta.

- Identificador de Usuario (int) - Foreign Key
- Identificador de incidencia (int) - Foreign Key
- Valoración (ENUM - Positiva, Negativa)
  Usuario e incidencias forman, juntos, la clave primaria. En este caso, sólo controlamos que los usuarios registrados no hagan más de una valoración para cada incidencia. Nótese que cada vez que hacemos una nueva valoración para una determinada incidencia el contador correspondiente que posee la incidencia crece. De esta forma, esta tabla servirá para controlar que las valoraciones de nuestros usuarios sean únicas para cada incidencia y no para utilizarla como contador.

Comentarios

Relación entre los comentarios realizados por los usuarios y las incidencias en las que sucede. La clave primaria será un identificador porque también pueden realizar comentarios un usuario anónimo y, además, no hay límite de comentarios por usuario en cada incidencia.

- Identificador (int) - Primary Key, Autoincrement
- Identificador de Usuario (int) - Foreign Key
- Identificador de incidencia (int) - Foreign Key
- Comentario (varchar(300))

Restricciones y anotaciones

- Tanto los usuarios registrados como los visitantes podrán hacer una única valoración de cada incidencia. **(Uso de cookies en caso de los invitados, tablas en caso de usuarios registrados.)**
- Se deberá mostrar un mensaje informando sobre el éxito o fracaso de cualquier modificación en la BBDD.
- Los administradores no pueden registrarse de forma autónoma. Será otro administrador quien le aporte dicho rango.
- Siempre debe de haber al menos un administrador.
- Implementar avatar por defecto.
- Implementar administrador por defecto en caso de que no haya ninguno. (Usuario: admin Contraseña: admin). **Cuidado, esto debe de funcionar tanto al inicio del sistema como si al restaurar la BBDD no se pone ningún administrador.**
- Mostrar incidencias por lotes **(OPCIONAL)**

Desarrollo de la práctica

Consultas a la Base de Datos

Para este apartado hemos desarrollado una clase llamada BaseDeDatos que se encuentra guardada en la ruta `proyecto/core`. En esta clase no sólo nos encargamos de realizar la conexión a la base de datos de nuestra página sino que también es la encargada de realizar todas las consultas. Es decir, cuando vayamos a realizar una consulta no nos tendremos que preocupar de ver cómo hacemos la consulta ni tampoco de cómo evitar algunos problemas como pueden ser la inyección sql, porque ya lo hace esta clase.

Todas y cada una de las consultas que vayamos a utilizar están implementadas aquí de forma segura y nos ahorramos el tener que ir pensando cómo debemos de redactar una consulta cada vez que sea necesario en nuestro código php. Es decir, estamos añadiendo una pequeña capa de abstracción.

Modelos de datos

Sin embargo, no siempre es del todo útil trabajar sólo con los resultados de la base de datos y por ello hemos implementado algunas clases que nos ayudan a almacenar los datos y agruparlos de forma útil dentro de nuestro código. Estas clases también no serán de utilidad a la hora de utilizar Twig para nuestro código html. Estas, se encuentran en la ruta `proyecto/core/modelo` y son las siguientes:

- Aside: tiene como constructor el recibir un número, este número será el que utilizará para delimitar el tamaño de los rankings que esta clase almacena. Después de todo, en definitiva se encarga de consultar la base de datos para obtener la información interesante que hemos utilizado para crear el aside de nuestra página como puede ser el ranking de usarios con más incidencias, el ranking de usuarios con más comentarios, el número de incidencias pendientes, entre otros.
- Comentario: clase con un constructor que se encarga de asignar los valores de las variables que almacena. Sirve como representación de un comentario con toda su información asociada.
- Usuario: clase con un constructor que se encarga de asignar los valores de las varibles que almacena. Sirve como representación de usuario con toda su información asociada.
- Log: clase con un constructor que se encarga de asignar los valroes de las variables que almacena. Sirve como representación de un elemento de nuestro log de la aplicación con toda su información asociada.
- Incidencia: clase con un constructor con el id de la incidencia que representa. Esta clase se encarga de realizar la búsqueda por sí misma a base de datos para obtener todos los datos que la representan. No sólo eso, sino que también realiza la consultas necesarias para crear el usuario asociado a la insidencia y los comentarios asociados junto con sus usuarios utilizando los constructores de asignación de las otras clases.

Nótese, que a veces será la propia base quien nos dará estas clases ya creadas y otras seremos nosotros quienes los creemos a partir de determinados parámaetros.