Base de datos y servicios PHP

Especialidad PHP



Felipe Izquierdo Romero

Índice

[1 Descripción del problema](#DescripciónDelProblema)

[2 Desarrollo](#Desarrollo)

1. Descripción del problema

Crear una aplicación web para la gestión de tareas. Esta aplicación podrá registrar los usuarios que trabajan durante una jornada laboral. La aplicación también podrá crear tareas y poder asignarlas a un usuario durante la creación de esta o posteriormente.

Los usuarios y las tareas asignadas y sin asignar se podrán visualizar en forma de tablas y para asignar tareas deberá de tener un desplegable con las tareas sin asignar y los usuarios a los que se les puede asignar tareas que serán quienes no tengan tareas asignadas.

1. Desarrollo

Para comenzar con el siguiente laboratorio he reutilizado parte del código del paso a paso para la gestión de usuarios. He añadido otro archivo “.php” con contenido “html” para visualizar las tareas asignadas, sin asignar y los formularios para crear una tarea o asignar tareas existentes. Para cambiar entre estos “.php” con una etiqueta “<a>” podrás navegar entre las dos vistas.



Ilustración 1- Navegación hacia administrador\_ tareas.php



Ilustración 2- Navegación hacia index.php

El administrador de tareas contiene dos formularios para agregar una tarea o para asignar una tarea existente a un usuario libre.

Texto

Descripción generada automáticamente

Ilustración 3- Formulario agregar tarea

Captura de pantalla de computadora

Descripción generada automáticamente

Ilustración 4- Asignar tarea

Contiene dos objetos que se encargarán de la conexión con la base de datos “ConexionDB” y el objeto que recogerá la conexión y tiene definidas las sentencias y retornará los valores “MySQLRepository”.



Ilustración 5- Objetos relaccionados con la BBDD

Para visualizar las tablas y el combo box para la asignación de tareas a usuarios que están disponibles existen las siguientes variables.

Texto

Descripción generada automáticamente

Ilustración 6- Variables para visualizar datos

Estos dos formularios en el mismo documento para diferenciar entre que formulario fue quien desencadeno la llamada tienen el mismo atributo name “form\_type” pero el atributo value es diferente de esa manera se distinguen con un bloque “if”.



Ilustración 7- Condicional agregar tarea

Posteriormente recogerá los valores del formulario, creará un objeto Tarea y agregará la tarea en la base de datos e imprimirá un mensaje de éxito o de error según el resultado.

Texto

Descripción generada automáticamente

Ilustración 8- Bloque de agregar tarea

Si es asignar tarea además comprueba que no haya un campo Sin Asignar que por defecto devolverá una cadena de caracteres con la palabra null.

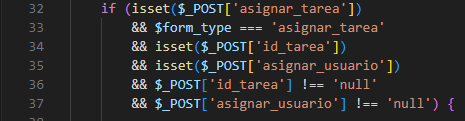


Ilustración 9- Condicional asignar tarea

Si los datos están correctamente introducidos recogerá el id de la tarea y del usuario para insertar en el registro de tarea el id del usuario al que corresponde la tarea.

Texto

Descripción generada automáticamente

Ilustración 10- Bloque asignar tarea

Como parte final de este modulo de php se cierra la conexión con la base de datos para asegurar su integridad y consumir menos recursos.



Ilustración 11- Cierre de la conexión

El modulo de usuarios por nombrar también su funcionamiento contiene un formulario para agregar un usuario y una tabla para visualizar los usuarios existentes.

Texto

Descripción generada automáticamente

Ilustración 12- Cuerpo html de usuarios

Texto

Descripción generada automáticamente

Ilustración 13- Cuerpo de php de usuarios

Profundizando en las clases y servicios que conforman la aplicación web: Tareas, Usuarios, mySqlRepository y ConexionDB.

mySqlRepository Recoge como propiedad la conexión a la base de datos en su constructor.

Texto

Descripción generada automáticamente

Ilustración 14- Constructor MySqlRepository

Los métodos que contiene son las sentencias sql para interactuar con la base de datos.

Texto

Descripción generada automáticamente

Ilustración 15- Sentencias sql (1)

Texto

Descripción generada automáticamente

Ilustración 16- Sentencias sql (2)

ConexiónDB recoge en el constructor las variables de entorno necesarias para conectar a la base de datos y contiene el método para cerrar conexión con la base de datos.

Texto

Descripción generada automáticamente

Ilustración 17- Servicio ConexionDB

La clase usuario implementa la interfaz usuario y esta clase recoge el nombre y el email del usuario y tiene métodos para retornar esos valores.

Texto

Descripción generada automáticamente

Ilustración 18- Clase usuario

Texto

Descripción generada automáticamente

Ilustración 19- Interfaz usuario

La clase tarea se comporta de manera similar, pero con otros valores, nombre y descripción.

Texto

Descripción generada automáticamente

Ilustración 20- Clase Tarea

Texto

Descripción generada automáticamente

Ilustración 21- Interfaz Tarea

Para finalizar un pequeño apunte es que el contenedor de Docker iniciaría el servicio de base de datos sin tablas, para que se creen las tablas y se cree un usuario de prueba y no me de error el programa he añadido en el “docker-compose.yml“ en la etiqueta “volume” un “init.sql “ que contiene el siguiente código sql.

Pantalla de computadora con letras

Descripción generada automáticamente con confianza media

Ilustración 22- init.sql