Documentación Blusqueda:

* **def load\_user(user\_id):** Carga el usuario actual, toma como parámetro el id del usuario.
* **recientes():** Retorna un array de 5 los blogs recientes que están activos y públicos.
* **queryBuscar(pablara):** Hace una query a la base de datos y retorna un array de diccionarios que en el titulo tienen la palabra ingresada.
* **def allowed\_file(filename):** Limita las extensiones que se pueden subir
* **def isEmailValid(email):** Valida si es correcto un email usando el paquete validate\_email
* **def isUsernameValid1(user):** verifica basados en la expresión regular definida anteriormente que el usuario tenga caracteres permitidos.
* **def isPasswordValid(password**): verifica basados en la expresión regular definida anteriormente que la contraseña tenga caracteres permitidos
* **def db\_crear():** Crea la base de datos usando los modelos.
* **def db\_drop():** Elimina todas las tablas de las bases de datos.
* **def db\_seed():** Llena las tablas de la base de datos con datos de prueba.
* **def home():** Renderiza la pagina de index.
* **def login():** Toma los datos del usuario los compara con la base de datos. Si el usuario existe le inicia la sesión y lo redirige al panel de blog, de lo contrario muestra el mensaje de error.
* **def logout():** Cierra la sesión del usuario y lo redirige la pagina de login.
* **crearBlog():** Toma los datos dados por el usuario y crea una nueva fila de blog en la base de datos. Verifica que los datos proporcionados sean validos y que el nombre de la imagen no pueda causar daños a la base de datos. Una vez el blog es creado redirecciona al usuario al la pagina del blog.
* **def panelBlog():** Muestra los blog creados por el usuario que estén activos y da la opción de filtrarlos por categoría además de crear uno nuevo.
* **def verificarCorreo():** renderiza el HTML de verificar correo
* **def validacion(validacion=''):** Toma el valor pasado por la url y hace una query en la base de datos, si existe inicia la sesión del usuario, de lo contrario muestra un mensaje de error.
* **def descagar(blogId:int):** Toma el id de blog pasado por url y retonar un pdf con el contenido del blog para su descarga.
* **def paginaBlog(blogId= None):** Toma el id de blog, si el blog es publico renderiza el blog, si no es publico pero pertenece al usuario tambien lo renderiza de lo contrario muestra el mensaje de error.
* **def resultadoBusqueda(palabra ="" ):** toma la palabra pasada por POST o url y muestra los blogs que la contienen. Además da la opción de filtrarlos por categorías.
* **def panelUsuario():** Si el usuario es administrador le da la opción de eliminar usuarios, de lo contrario solo muestra la opciones de usuario.
* **def crearCuenta():** Toma los datos del usuario, verifica que sean validos, crea una nueva fila en usuarios de la base de datos y envía un correo con un link para validar el usuario.
* **def editar(blogId=0):** Toma el id de blog para llenar los espacios por el usuario y le permite cambiar la información.
* **def eliminar(blogId=0):** Toma el id del blog y lo deshabilita cambian el estad de 1 a 0.
* **def recuperarPassword**(): Toma un correo y si existe en la base de datos envía un link que le permite al usuario cambiar su contraseña.
* **def recuperar(validacion=''):** Toma la nueva contraseña y la cambia.