**MEMORIA BLOG**

### Necesidad:

Tal y como se explica en el enunciado “se desea crear una aplicación básica para gestionar un **blog.** El tema del blog será el preferido por el alumn@.”

### Selección del tema:

He escogido la elaboración de una aplicación web referida al mundo del café. Las características de la aplicación hasta la fecha serán las desarrolladas a continuación.

### Características:

-En la página de inicio se muestran las entradas en orden de más actuales a más antiguas, y de forma paginada usando librería ZebraPagination (que no ZebraPaginacion).

-Se presenta un input para poder realizar búsquedas de las entradas correspondientes al criterio elegido. Se realiza la búsqueda tanto en el título como en descripción.

-Un usuario sin logarse puede ver las entradas. Puede registrarse y puede ingresar en su zona personal. Cuando se registra y algo falla, se rellenan los campos con la información previa si ésta era correcta.

-Un usuario logado que no es administrador, puede ver todas las entradas. Aparece un botón de edición solo en las entradas que él ha escrito. Puede editarla, pero no dejamos borrarla. Esta acción he considerado que sea sólo responsabilidad del administrador. Además las validaciones hacen que no pueda modificar una entrada con sólo espacios vacíos. También puede crear entradas con un botón que le aparece en su zona. También puede modificar sus datos y actualizarlos.

-Un usuario logado como administrador tiene en su zona opciones para gestionar las categorías, entradas y usuarios. Puede modificar y borrar en todas las áreas.

### Mejoras

-Creación de nuevo modelo “EntradaModificada” en la que participen también los datos tanto de **usuario** como de **categoría** enviando el *nombre* del usuario y el *nombre* de la categoría. Ahora mismo sólo se muestra el propio id de usuario y el id de categoría como referencia entre tablas, no dando una información deseada al usuario sobre esos campos.

-El usuario tiene que introducir su fecha de nacimiento cuando modifica sus datos. No se rellena automáticamente. Ese dato no se guarda en la sesión. O bien se guarda en la sesión o se recuperan todos los datos con una consulta a su registro.

-Propagación de errores con **throw $e** en los “catch (\PDOException $e) {}”.

-CSS por supuesto.

### Consideraciones de funcionamiento

-Se necesita crear la base de datos con el script dentro de la carpeta SQL del proyecto.

-Para establecer un usuario administrador lo haremos desde el menú Registrar de la propia aplicación, usando así la funcionalidad de password\_hash de PHP. Luego en la propia base de datos mariaDb, desde phpmyadmin**, cambiaremos el rol que por defecto se ingresa como un 1, por un 7.**

No puedo proporcinar un insert directo porque la función de PHP password\_hash no la podría usar, y la verificación con password\_verify no sería posible, y son parte de la lógica de la aplicación. Se puede usar en un insert directo en phpmyadmin usando el hash SHA2, resolviendo el problema de la password pero no resolviendo la funcionalidad de la aplicación.