

**Escenario elegido:** Una ToDo List para la organización y gestión de de proyectos con sus respectivas tareas.

**Requerimientos TPI Web:**

**1. Proyecto basado en el patrón MVC con Razor Pages**

La aplicación está estructurada utilizando el patrón MVC: los modelos se encuentran en la carpeta Models, los controladores en Controllers, y las vistas están desarrolladas con Razor Pages en Views. Esto garantiza una separación clara entre la lógica de negocio, la presentación y el acceso a datos.

**2. Registro de nuevos usuarios**

El controlador CuentaController incluye acciones para mostrar el formulario de registro y almacenar nuevos usuarios. Se validan la unicidad del correo y el nombre de usuario antes de crear la cuenta.

**3. Inicio de sesión mediante usuario y contraseña**

El sistema permite iniciar sesión utilizando nombre de usuario o correo electrónico junto con la contraseña, verificando los datos en la base de datos.

**4. Contraseñas almacenadas utilizando hash y salt**

Las contraseñas se almacenan utilizando el algoritmo PBKDF2 con SHA256 y salt aleatorio. Esto se realiza mediante la clase PasswordHelper.

**5. Recuperación de contraseña mediante un correo electrónico con un enlace único**

El sistema permite a los usuarios solicitar la recuperación de su contraseña. Se genera un token único con vencimiento, se almacena temporalmente y se envía por correo electrónico un enlace que permite el restablecimiento de la contraseña.

**6. Listado paginado de elementos**

No se completó el requerimiento.

**7. Vista de detalle de un elemento específico**

Se visualizan los proyectos y sus tareas mediante un componente tipo acordeón, con los detalles como la descripción, estado de la tarea y prioridades.

#### **8. Crear un nuevo elemento (con campos y al menos un combo/select)**

Se pueden crear nuevos proyectos con tareas asociadas. Las tareas incluyen campos tipo combo/select para seleccionar el estado y la prioridad.

#### **9. Editar un elemento existente**

Los proyectos y tareas pueden ser editados desde la interfaz de usuario mediante formularios con botones de guardar, que actualizan los datos en la base de datos.

#### **10. Visualización de un código QR generado desde el backend**

Se utiliza la biblioteca QRCode para generar códigos QR desde el backend .NET. Estos se visualizan correctamente en su vista correspondiente.

#### **11. Acceso a funcionalidad exclusiva mediante código QR**

Actualmente se genera un código QR con un enlace, el usuario puede ingresar un enlace al campo solicitado y generar un QR al cual se puede acceder mediante el celular o un scanner para QRs.