Proyecto Esperanto

PAC 3

Comentarios relevantes sobre la implementación

Jesús Cerezuela Lorenzo

Índice

1. Introducción 3
2. Aspectos relevantes 3
2.1 Stores procedures Versus Linq 3
2.2 Stores procedures de búsqueda 3
2.3 Patrón Singleton 4
2.4 TinyMCE 4
2.5 Jquery dialogs 4
2.6 Session Helper 4
2.7 Helpers 5
2.8 Regions 5
2.9 Partials 5
2.10 Jquery 5
2.11 Ajax 5
2.12 Diseño de la web (CSS) 6
2.13 Validaciones 6
2.14 RazorJs 6
3. Conclusiones 6

1. Introducción.

El presente documento tiene la finalidad de mostrar los aspectos relevantes sobre la implementación

2. Aspectos relevantes

2.1 Stores procedures Versus Linq

A pesar de que resulta muy cómo trabajar con Linq sobre la capa de datos, he preferido hacer uso de stores procedures debido a que es posible modificar la lógica de los stores sin tener que hacer un deploy de la aplicación (siempre y cuando no se altere los parámetros de entrada ni el objeto de salida)

Por ejemplo si una vez que la aplicación está en producción y nos piden realizar la ordenación de los usuarios por orden de antigüedad, podemos actualizar el store sin necesidad de volver a publicar la aplicación.

Con Linq esto no sería posible ya que deberíamos tocar código y volver a publicar con el inconveniente que resulta en los entornos de producción.

Sin embargo he aprovechado la potencia de linqtoobjects para la capa de negocio.

2.2 Stores procedures de búsqueda

Para evitar tener que montar una cadena de texto en función de los parámetros de búsqueda que introduce el usuario y después lanzar un EXEC(SqlStr) he usado la siguiente sintaxis

```
SELECT
    UserId, NIF, Nombre, Apellido1, Apellido2, Telf, Direccion, Poblacion, Email,
    [Password], RolId, FechaNacimiento, FechaAlta, FechaBaja, Estado, IdIdioma
FROM dbo.Usuarios
WHERE (@Rol = -1) OR (RolId = @Rol))
AND ((@Estado = -1) OR (Estado = @Estado))
```

Por ejemplo si el usuario no informa el campo RolId (-1) retorna todos los registros

(@RolId = -1 retorna true), pero si informa este campo se ejecuta la parte del OR y filtra por RolId

2.3 Patrón Singleton

Las clases manager de la capa de negocio implementan el patrón singleton, con este patrón se asegura que sólo haya una instancia en ejecución.

Su uso es el siguiente:

Nombredelaclase.instance.metodo()

2.4 TinyMCE

Para proporcionar la posibilidad de usar texto enriquecido en la edición de libros y así que los usuarios que consultan el detalle del libro tengan una mejor user experience he usado el editor TiniMCE

Este plugin que se puede instalar directamente desde el management Nugget de visual studio y permite transformar un text area en un editor de texto con muchas opciones (configurable)

Su uso es relativamente sencillo aunque me ha dado problemas al intentar usarlo dentro de jquery dialogs.

2.5 Jquery dialogs

Con el jquery dialog he podido crear modal popup para mostrar los detalles de las reservas y los préstamos.

Además con los dialogs he substituido los alerts por dialogs dando así un aspecto más presentable a la aplicación.

2.6 Session Helper

La clase SessionHelper es una clase shared que permita centralizar todas las variables de sesión que se usa en la aplicación.

De este modo se evita duplicar variables de sesión que tienen el mismo cometido (cuando trabaja más de una persona en el mismo proyecto), por ejemplo:

Session (“IdLanguage”)

Session (“IdIdioma”)

Session (“CodiIdioma”)

2.7 Helpers

Para simplificar y reutilizar código he creado la clase HtmlExtensions que permite reutilizar combos en cualquier pantalla.

Esta clase me permite reutilizar también la función para las traducciones.

2.8 Regions

Para tener el código de una manera ordenada y poder colapsar de forma rápida regiones de código he usado #Regions.

2.9 Partials

Las partials es una de las mejores formas que permiten reutilizar código además de que posibilitan tener una estructura más clara y entendible.

Por ejemplo, tengo la partial Usuario que muestra los datos de un usuario. Esta partial la puedo usar en el detalle del usuario o en el alta/edición de usuarios.

En realidad es la misma partial para todas estas funcionalidades, pero se muestra de forma diferente (muestra / oculta controles y funcionalidades) en función del rol del usuario o en función del controlador que llame la partial.

2.10 Jquery

Usando Jquery ahorras horas de trabajo, al principio puede parecer que no (ya que tiene una curva de aprendizaje un tanto elevada) y puede tentarte la idea de usar javascript de forma convencional.

Pero si superas esta primera fase las ventajas son muchas. Por ejemplo, agrupar controles con la misma clase y aplicar un estilo en concreto.

2.11 Ajax

Es uno de los pilares de la aplicación. Con esta tecnología puedes realizar llamadas asíncronas para recuperar información de la base de datos y mostrarla en una parte concreta de la página.

De este modo evitas los refrescos de la página y se proporciona al usuario una agradable use experience.

2.12 Diseño de la web (CSS)

Para el diseño de la web he usado Css, esta metodología permite definir estilos en un documento aparte y por ejemplo si se quiere cambiar el aspecto de la aplicación con crear otra CSS es suficiente y no se tiene que tocar el código para nada.

Otra ventaja es la reutilización, si por ejemplo se quiere dar el mismo estilo a todos los input text con definir una clase para todos es suficiente y no es necesario ir control a control a definir su style.

2.13 Validaciones

De las validaciones de datos en la vista de usuario se encarga el fichero CF.js, este fichero implementa funciones que permiten validar, fechas y números entre otros.

Para validar DNI existe el fichero CIFValidator. Respecto a este fichero es muy importante destacar que la función validateDNI (DNI) tiene deshabilitado la validación de nifs para agilizar el proceso de pruebas.

Si se desea habilitar esta validación hay que quitar la línea return true que hay al principio de la función

2.14 RazorJs

RazorJs permite usar la sintaxis razor en fichero externos de javascript.

Debo comentar que no he sido capaz de hacerlo funcionar en mi proyecto y por eso no he podido usar ficheros externos js (ya que no reconoce el código razor)

Por este motivo el código javascript está en las páginas aunque sea una práctica que no me gusta en absoluto.

3. Conclusiones

En conclusión comentar que con este proyecto, he avanzado bastante con MVC, Jquery y Css que si al principio del proyecto me pareció interesante ahora me parece realmente espectacular las posibilidades que ofrece.