

TREBALL FI DE CARRERA

Gestió d'una Biblioteca

Àrea TFC .NET

Comentaris al desenvolupament

Lluís Oriol Canalda

Maig 2013

Consultor: Jairo Sarrias Guzmán

Implementació del projecte.

La implementació del projecte s'ha realitzat amb Visual Studio 2010 Professional, utilitzant Visual C# com a llenguatge de programació. Les tecnologies utilitzades per implementar el projecte han estat: Winforms, Windows Communication Foundation, LINQ, ASP.NET i pàgines d'estil CCS.

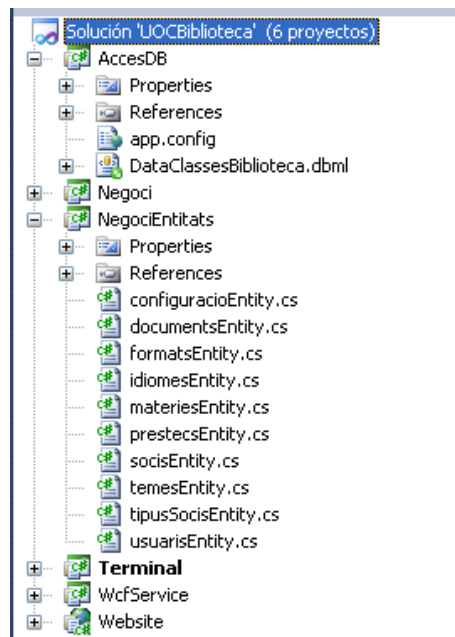
Capa de dades

La base de dades s'ha creat i emmagatzemat en SQL Server 2008. Per implementar la capa de dades s'ha utilitzat la tecnologia LINQ to SQL.

LINQ facilita l'accés a la base de dades, doncs ens permet encapsular dins del nostre projecte la definició de l'esquema de la base de dades, permetent realitzar les consultes cap a la nostra base de dades, utilitzant les classes mapejades de les diferents taules.

El projecte VS **AccesBD** conté l'esquema dbml de la base de dades

El projecte VS **NegociEntitats** conté les classes mapejades de les taules



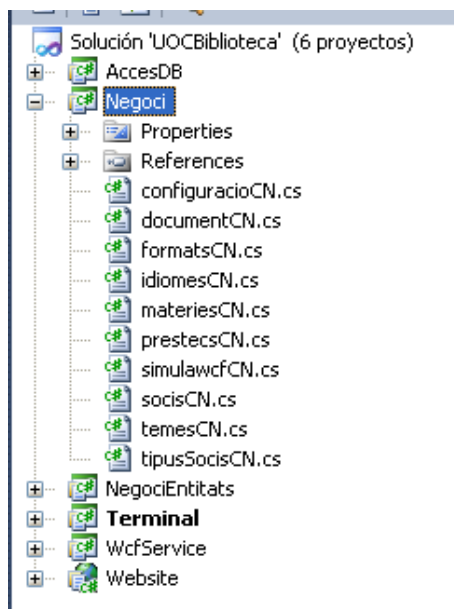
Capa de negoci

El projecte VS **Negoci** conté la lògica de negoci. Hi ha les classes que implementen les rutines per donar resposta a les accions realitzades per l'usuari a les capes de presentació.

S'ha creat una arxiu de classe independent per cada entitat de la base de dades. Dins de cada arxiu, podem trobar les accions de cerca, modificació, eliminació de les entitats de la base de dades, i també els càlculs o tractaments especials de certs atributs.

En les operacions d'accés a la base de dades, s'han incorporat el bloc d'intruccions try-catch per controlar els possibles errors en l'accés a la BD.

Aquestes classes interactuen amb la capa de servei WCF.



Per temes d'optimització del rendiment, també s'ha creat la classe **simulaWCF** que conté les mateixes estructures i operacions que s'implementen en el servei WCF.

Així doncs i opcionalment, canviant sols una instrucció de codi en les classes de la capa de presentació (en l'entorn de l'aplicació d'escriptori), i sense tenir que canviar res la lògica de funcionament, podem treballar utilitzant l'accés a una .DLL o bé, a un servei WCF publicat.

```
public partial class frm3Soci : Form
{
    //private ServiceReference1.Service1Client clientWCF = new ServiceReference1.Service1Client(); // Instancia del servei WCF
    private simulawcfCN clientWCF = new simulawcfCN(); // simula instancia WCF
    private List<tipusSocisEntity> ListTipusSocis;
    private List<prestecsEntity> ListPrestecs;
```

Capa de presentació

La implementació destinada cap a la capa de presentació, està formada per dues parts diferenciades: l'entorn d'escriptori i l'entorn web.

Escriptori

El projecte VS **Terminal** conté la part de l'aplicació d'escriptori.

Les classes implementen formularis Winforms, existint una classe per cada pantalla de l'aplicació.

La nomenclatura de les classes segueixen el patró

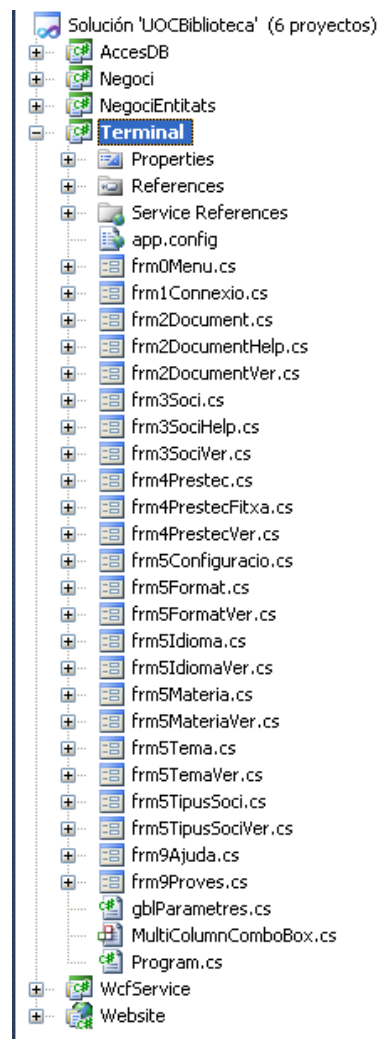
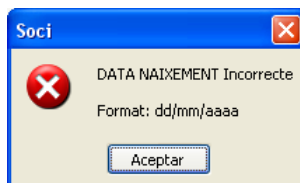
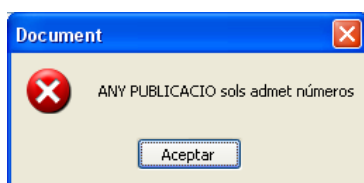
frmNxxxxx

N = indica l'ordre d'opció en relació al menú principal

xxxxx = indica l'apartat

- S'han dissenyat totes les interfícies amb el mateix estil, per tal de facilitar la comprensió i utilització a l'usuari final.

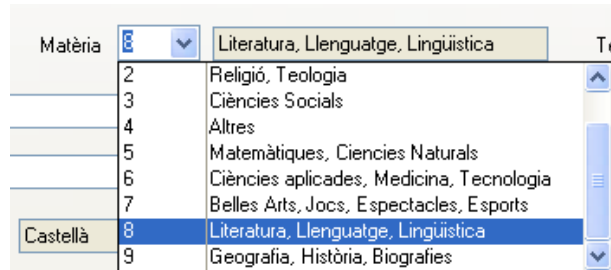
- S'han incorporat control per l'entrada en els camps que requereixen un tractament especial, com són els camps numèrics i els camps data. L'aplicació requereix que el format dels camps data sigui dd/mm/aaaa complert.



- Com que el menú principal sempre és accessible, s'ha incorporat un control sobre les pantalles actives, no permeten a l'usuari que executi dues vegades la mateixa opció. Per exemple, entrar dues vegades a l'opció de gestió de documents.



- S'ha implementat un nou format de camp Combobox, que incorpora dos atribut: el primer considerat com índex i el segon com a descripció, permetent així a l'usuari veure clarament quina és el valor escollit. Una vegada seleccionat el valor, es transporta cap a la pantalla cap a un camp TextBox-ReadOnly



- Malgrat que les operacions que implementen els mètodes dins les classes incorporen noms "lògics" que descriuen la seva funció, s'han incorporat nombrosos comentaris d'ajuda dins del codi, per facilitar la seva comprensió i el posterior manteniment.

```

moureCamps_a_Pantalla();

//Desactivar opcions segons usuari actiu
// si rol=2 (usuari normal), desactivar opcions
gblParametres parametres = gblParametres.Instance();
if (parametres.usuariRol == 2)
{
    acceptarButton.Enabled = false;
    eliminarButton.Enabled = false;
}

```

Plataforma Web

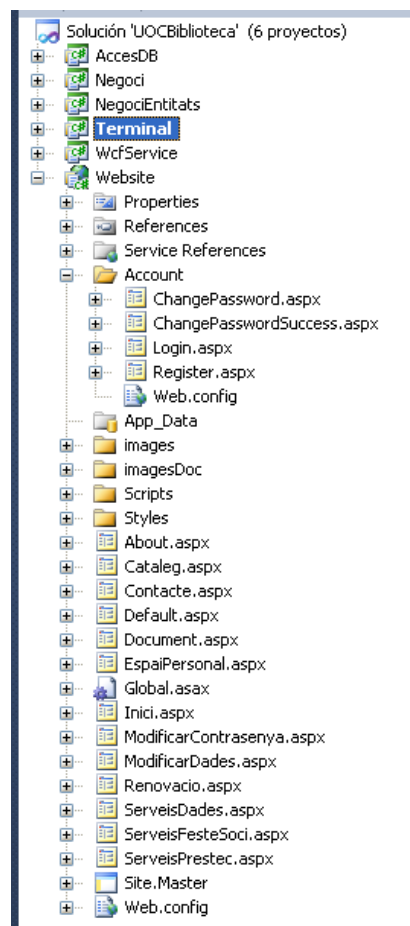
El projecte VS **Website** conté les classes ASP.NET de l'entorn Web.

Per la pàgina màster *Site.Master*, s'ha conservat l'estil original de Visual Studio, doncs m'has semblat força atractiu, malgrat que segurament hi hauran moltes solucions amb el mateix estil.

S'han realitzat alguns canvis en la fulla d'estil generada per defecte *Site.css* per tal de poder configurar el menú lateral que incorporen les pàgines de l'apartat servei.

Igual que en l'entorn d'escriptori, les pàgines .aspx s'han dissenyat seguint el mateix estil, intuïtiu i simple per l'usuari.

En el desenvolupament de les pàgines .aspx, s'ha contemplat i controlat el comportament ocasionat per l'efecte "postback".



Per les pàgines que presenten una llista d'elements, com pot ser la visualització del catàleg, o bé, l'historial de préstecs d'un soci, s'ha habilitat la paginació per diferents pàgines d'informació.

En cas de que l'usuari desitgi ampliar una opció (per exemple, veure el detall d'un document), quan es retorna a la llista, es controla que l'aplicació es situa en la mateixa pàgina d'informació en que es trobava.