

Catàleg de productes: ArtNet TFC .NET + AJAX

Nom Estudiant: **Fernando Escalera Almendros**
ETIS

Nom Consultor: **David Gañán Jiménez**

11 de Juny del 2007

Resum del projecte

ArtNet és el treball final de carrera d'enginyeria tècnica de sistemes. La finalitat d'aquest projecte és desenvolupar una aplicació Web utilitzant la tecnologia .Net amb AJAX.

La tecnologia .NET intenta oferir una manera ràpida i econòmica però a la vegada segura i robusta de desenvolupar aplicacions, i juntament amb AJAX permet la construcció de Webs més ràpides, més interactives i més usables.

En concret en aquest projecte hem fet servir ASP.net, visual basic com llenguatge de programació, accés a base de dades amb ADO.net, i hem utilitzat els components del toolkit d'AJAX que millor s'adaptaven a les necessitats del projecte.

ArtNet és una galeria d'art digital. Accedint a la Web es mostren obres d'art que el públic pot comprar, però també conté un gestor perquè la persona responsable de la Web pugui fer el manteniment de tots els continguts de la Web.

En aquesta memòria apareixen detallades les fases d'anàlisi de requeriments, disseny i tot el procés d'implementació del projecte.

Índex de continguts

Portada	1
Resum del projecte	2
Taula de Contingut	3
Índex de figures	4
1. Introducció	5
1.1 Justificació del TFC	5
1.2 Objectius del TFC	5
1.3 Enfocament i mètode seguit.....	5
1.4 Planificació del projecte.....	6
1.5 Productes obtinguts	8
1.6 Descripció dels altres capítols.....	8
2. Anàlisi de requeriments	9
2.1 Funcionalitats de l'usuari de la Web	9
2.2 Funcionalitats de l'administrador de la Web.....	9
2.3 Diagrama casos d'ús.....	10
2.4 Casos d'ús per l'usuari	11
2.5 Casos d'ús per l'administrador.....	12
2.6 Model conceptual de classes	15
3. Disseny	16
3.1 Arquitectura del sistema.....	16
3.2 Diagrames de seqüència.....	17
3.4 Diagrames de classes de disseny	20
3.5 El model de dades	20
3.5 Diagrama lògic de la base de dades	22
3.5 Generació de la base de dades	23
3.6 Interfície gràfica de la Web	28
3.7 Interfície gràfica del gestor	33
4. Infraestructura del desenvolupament	39
4.1 ASP.NET – Descripció.....	39
4.2 ASP.NET – Funcionament	40
4.3 AJAX	42
4.4 AJAX - Components utilitzats	45
5. Conclusions	46
6. Glossari	47
7. Bibliografia	49

Índex de figures

Figura 1 – Planificació anàlisi i disseny	6
Figura 2 – Planificació implementació	7
Figura 3 – Planificació memòria i presentació.....	7
Figura 4 – Casos d'ús.....	10
Figura 5 – Model conceptual de classes.....	15
Figura 6 – Diagrama Seqüència - Registre.....	17
Figura 7 – Diagrama Seqüència - Login	17
Figura 8 – Diagrama Seqüència –Consulta projectes.....	18
Figura 9 – Diagrama Seqüència –Realitza comanda	18
Figura 10 – Diagrama Seqüència - Cerques	19
Figura 11 – Diagrama Seqüència -Contacta.....	19
Figura 12 – Model de classes de disseny	20
Figura 13 – Disseny lògic de la base de dades	22
Figura 14 – Interfície gràfica – Web - Home	28
Figura 15 – Interfície gràfica – Web - Productes	29
Figura 16 – Interfície gràfica – Web – Producte ampliat.....	30
Figura 17 – Interfície gràfica – Web – Consulta usuari identificat	31
Figura 18 – Interfície gràfica – Web – Detall carret.....	31
Figura 19 – Interfície gràfica – Web - Contacta	32
Figura 20 – Interfície gràfica – Gestor - Login.....	33
Figura 21 – Interfície gràfica – Gestor - Índex	34
Figura 22 – Interfície gràfica – Gestor - Productes.....	35
Figura 23 – Interfície gràfica – Gestor – Afegir, modificar productes.....	35
Figura 24 – Interfície gràfica – Gestor - Usuaris	36
Figura 25 – Interfície gràfica – Gestor - Afegir, modificar usuaris.....	36
Figura 26 – Interfície gràfica – Gestor - Categories.....	37
Figura 27 – Interfície gràfica – Gestor - SubCategories	37
Figura 28 – Interfície gràfica – Gestor - Comandes.....	38
Figura 29 – ASP.NET – Funcionament.....	40
Figura 30 – AJAX – Aplicació Web clàssica.....	43
Figura 31 – AJAX – Aplicació Web amb AJAX	44

1. Introducció

1.1 Justificació del TFC

La realització d'aquest projecte m'ha permès aprendre a utilitzar la tecnologia .NET juntament amb AJAX, ja que .NET és una tecnologia molt utilitzada per empreses tant per crear aplicacions Windows, per dispositius mòbils o aplicacions Web.

A part de formar-me en aquestes tecnologies també volia aprendre com treballar amb una arquitectura en tres capes, i per tot això vaig escollir fer un catàleg de productes.

La realització d'aquesta Web em permetria la utilització de ASP.NET, també d'AJAX, aprendre com funciona el Visual Studio 2005 i utilitzar connectivitat amb SQLServer 2005 mitjançant ADO.NET

1.2 Objectius del TFC

L'objectiu principal del treball final de carrera era l'aprenentatge de noves tecnologies com poden ser ASP.NET i AJAX desenvolupant per la Web un catàleg de productes.

La Web tindria dos parts molt diferenciades:

- La Web accessible a tot el públic: On es mostrarien els productes del catàleg, i el públic podria fer comandes a través de la Web.
- El gestor: On l'administrador podria realitzar el manteniment de tots els continguts de la Web.

1.3 Enfocament i mètode seguit

El projecte ArtNet és la possibilitat que té una empresa poder donar-se a conèixer a través de la Web així com ensenyar els seus productes i permetre al públic fer comandes a través de la Web.

El desenvolupament del projecte s'ha dividit en etapes per estructurar tot el projecte. Les tres etapes destacades són: anàlisi de requeriments, disseny i implementació.

La primera etapa és l'anàlisi de requeriments, que es basa en recollir tota la informació necessària per poder fer el disseny de la Web, identificant els objectes del domini del món real i les seves interaccions en l'àmbit de la venda a través d'Internet.

Una vegada finalitzat l'anàlisi de requeriments s'ha passat a l'etapa de disseny, on la pretensió era separar la lògica del programa, amb el disseny i amb la base de dades.

La següent etapa és la d'implementació. Aquestes dues primeres etapes són comunes per a qualsevol desenvolupament seriós, serà en la fase d'implementació on apareixen els objectes principals del projecte, la tecnologia .NET i AJAX.

Per tal d'implementar el projecte, el primer que calia era instal·lar tot el Framework 2.0 de Microsoft, instal·lar el visual Studio 2005 i el SQLServer 2005. A continuació caldria instal·lar ASP.NET 2.0 AJAX Extensions 1.0 al servidor, i també instal·lar ASP.NET AJAX Control Toolkit al Visual Studio 2005.

1.4 Planificació del projecte

Per tal de fer una planificació del projecte s'ha tingut en compte les entregues que s'havien de fer, marcades per la universitat.

Títol	Lliurament
Pla de treball	13/03/2007
Anàlisi i disseny	09/04/2007
Implementació	28/05/2007
Lliurament final	11/06/2007

El pla de treball:

El pla de treball descriu, de forma clara el problema que pretén resoldre el projecte, el treball concret que es portarà a terme i la seva descomposició en tasques i fites temporals.

Anàlisi i disseny:

El contingut de cada etapa, conté:

Anàlisi

- Anàlisi de requeriments
- Diagrama de casos d' us
- Model conceptual de classes

Disseny

- Arquitectura del sistema
- Diagrames de seqüència
- Diagrama de classes del disseny
- Disseny de la interfície gràfica
- Disseny de la base de dades

Planificació de l'entrega de l'anàlisi i el disseny:

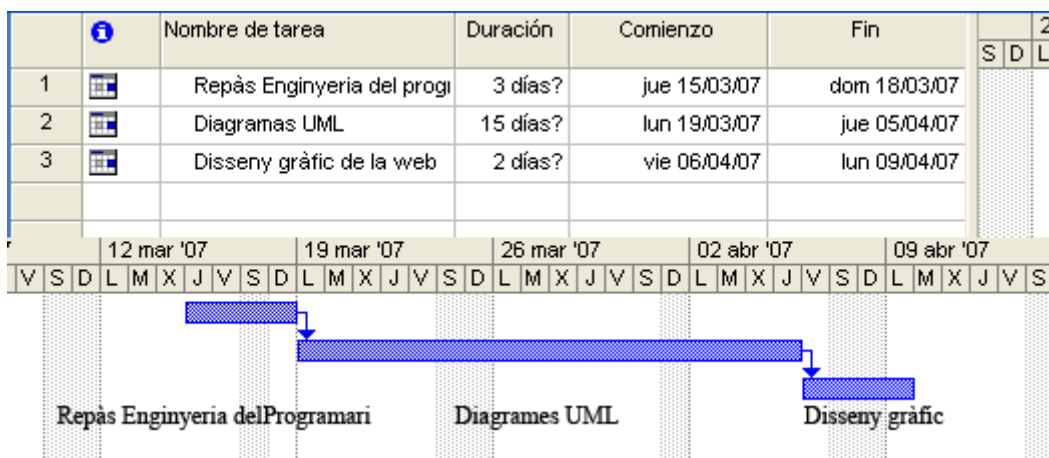


Figura 1

Implementació:

Conté el producte desenvolupat, tota la base de dades i un petit manual d'instal·lació.

Planificació de l'entrega de la implementació:

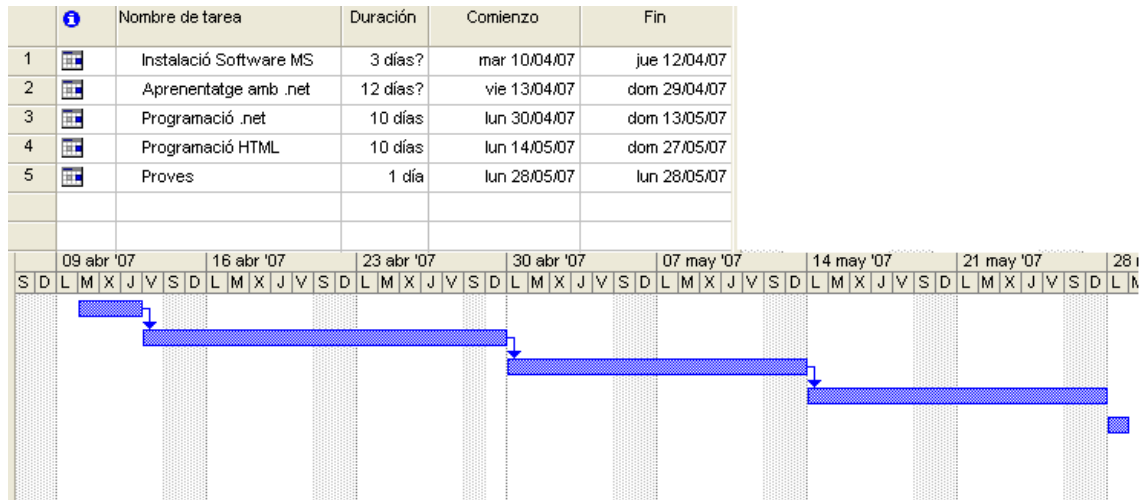


Figura 2

Lliurament final:

Conté tot el producte desenvolupat finalitzat, tota la base de dades i un petit manual d'instal·lació.

També conté la memòria que és aquest document, on es reflecteix tot el treball realitzat durant el curs, i la presentació virtual on es mostren totes les característiques més innovadores del projecte realitzat.

Planificació de l'entrega de la memòria i la presentació:

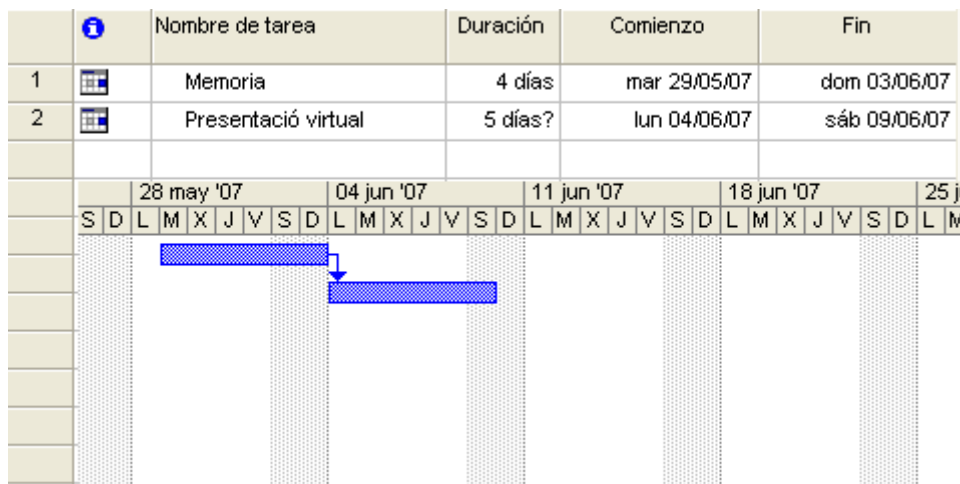


Figura 3

1.5 Productes obtinguts

Una vegada finalitzat el projecte hem obtingut els següents productes:

- Un document de planificació del projecte.
- Un document amb l'anàlisi i el disseny del projecte.
- El projecte, on es poden diferenciar dos parts:
 1. Un catàleg de productes que el públic pot consultar i fer comandes.
 2. Un gestor per l'administrador dels continguts de la Web.
- Una memòria on es mostra el treball realitzat durant el curs
- Una presentació virtual on s'expliquen les característiques més innovadores del projecte.

1.6 Descripció dels altres capítols

Ens els següents capítols veurem el cicle de vida del projecte ArtNet.

El primer que veurem serà l'anàlisi de requeriments, i els casos d'ús, tant per l'usuari com per l'administrador de la Web. A continuació veurem l'etapa de disseny, veurem el tipus d'arquitectura utilitzat i els diagrames de seqüència, de bases de dades i la interfície gràfica del projecte.

En capítols posteriors explicarem amb detall ASP.NET i AJAX, i acabarem la memòria amb un apartat de conclusions, un glossari i la bibliografia.

2. Anàlisi de requeriments

2.1 Funcionalitats de l'usuari de la Web

A continuació es mostren les funcionalitats de públic que lliurament accedeix a la Web:

Registrar-se omplint un formulari amb les seves dades personals.

Estar registrat a la Web li permetrà fer comandes via Web.

Si l'usuari no està registrat a la Web podrà registrar-se introduint les dades a un formulari.

Identificar-se a la Web (sempre i quan estigui registrat).

L'usuari que ha estat registrat correctament tindrà un nom i una contrasenya per tal d'accedir al sistema.

Consultar els productes del catàleg.

L'usuari quan accedeixi a la Web podrà consultar tot el catàleg de productes lliurament. Els productes del catàleg es distribueixen en categories i subcategories i poden estar d'oferta o ser una novetat.

Fer un cerca de productes del catàleg.

Facilitarà que l'usuari trobi el producte que busca.

Consultar els productes que són novetats o estan en oferta.

L'usuari podrà accedir als productes en oferta o novetat a través del menú.

Fer una comanda. (sempre i quan estigui registrat).

Hi haurà disponible el típic carret de la compra. Quan l'usuari finalitza d'introduir quantitats i vol enviar la comanda, el que es farà serà guardar la comanda en la base de dades per una posterior revisió de l'administrador.

Contactar amb l'administrador de la Web.

A través del menú l'usuari tindrà accés a un petit formulari on sol·licitar informació, o qualsevol dubte que es tingui de la Web.

2.2 Funcionalitats de l'administrador de la Web

A continuació es mostren les funcionalitats que l'administrador de la Web pot fer al gestor de continguts:

Fer login per tal d'accedir a l'aplicació de gestió

Pot introduir, modificar i eliminar categories.

Es mostrarà una llista amb totes les categories existents i l'usuari podrà introduir de noves, modificar-les o eliminar-les.

Pot introduir, modificar i eliminar subcategories.

Pot introduir, modificar i eliminar productes, i pujar la fotografia del producte via Web.

El sistema li mostrarà tot el llistat de productes. Podrà fer el manteniment dels productes. Al introduir un nou producte podrà pujar la fotografia del producte.

Pot posar els productes com a novetats, o oferta.

Al fer el manteniment de productes podrà destacar els productes com a novetat o oferta.

Pot consultar les comandes que s'han fet a través de la Web.

També podrà marcar la comanda com entregada.

Pot introduir, modificar i eliminar usuaris registrats.

Es mostrarà una llista amb tots els usuaris registrats, i podrà fer el manteniment de la taula.

2.3 Casos d'ús

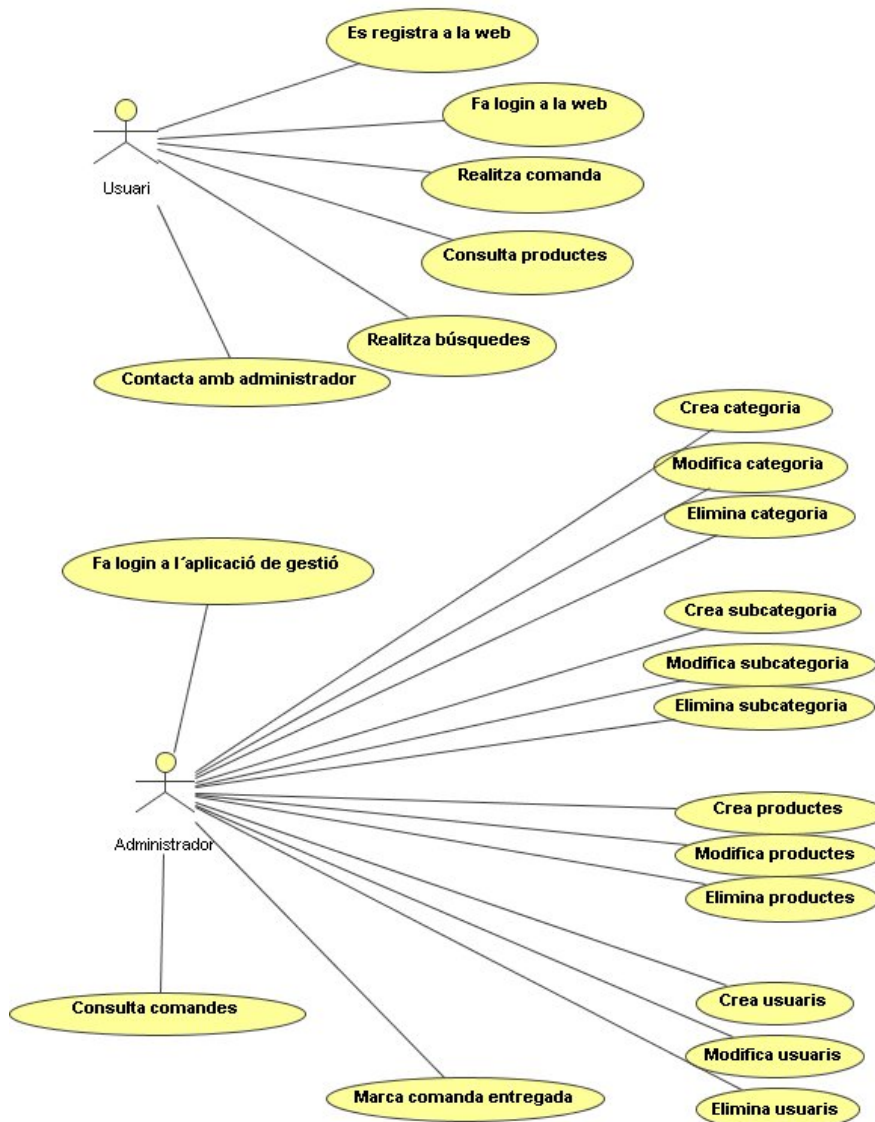


Figura 4

2.4 Casos d'ús per l'usuari de la Web

Cas d'ús : Es registra a la Web

Resum de la funcionalitat: L'usuari introdueix les seves dades personals per poder registrar-se a la Web.

Actors: Usuari.

Precondició: L'usuari no ha d'estar registrat.

Postcondició: Es registra un nou usuari.

Procés normal:

1. L'usuari omple els camps de formulari.
2. El sistema guarda les dades del registre.

Cas d'ús : Fa login a la Web

Resum de la funcionalitat: L'usuari introdueix usuari i contrasenya per identificar-se al sistema.

Actors: Usuari.

Precondició: L'usuari ha d'estar registrat.

Postcondició: L'usuari s'ha identificat al sistema.

Procés normal:

1. L'usuari omple els camps de usuari i contrasenya.
2. El sistema valida l'usuari i s'inicia una nova sessió.

Cas d'ús : Consulta Productes

Resum de la funcionalitat: L'usuari navega per la Web consultant els productes disponibles.

Actors: Usuari.

Precondició: -

Postcondició: El sistema mostra informació dels productes disponibles.

Procés normal:

1. L'usuari navega per la Web a través del menú
2. El sistema mostra informació dels productes disponibles.

Cas d'ús : Realitza comanda

Resum de la funcionalitat: L'usuari navega per la Web utilitzant el carret de la compra i una vegada ha inclòs tot el que vol, envia la comanda a l'administrador.

Actors: Usuari.

Precondició: L'usuari ha d'estar identificat.

Postcondició: -

Procés normal:

1. L'usuari omple de productes el carret de la compra.
2. Quan l'usuari vol, envia la comanda.
3. Quan s'envia la comanda el sistema guarda les dades de la comanda.

Cas d'ús : Realitza cerques

Resum de la funcionalitat: L'usuari pot cercar un producte a la Web.

Actors: Usuari.

Precondició: -

Postcondició: -

Procés normal:

1. L'usuari fa una cerca de productes.
2. El sistema mostra els productes que l'usuari busca

Cas d'ús : Contacta amb administrador

Resum de la funcionalitat: L'usuari pot contactar amb l'administrador de la Web.

Actors: Usuari.

Precondició: -

Postcondició: -

Procés normal:

1. L'usuari accedeix al formulari de consulta i ho envia.
2. El sistema envia una mail a l'usuari amb la consulta.

2.5 Casos d'ús per l'administrador de la Web

Cas d'ús : Fa login a la Web

Resum de la funcionalitat: L'administrador introdueix usuari i contrasenya per identificar-se al sistema.

Actors: Administrador.

Precondició: Només l'administrador té accés a l'aplicació.

Postcondició: L'administrador s'ha identificat al sistema.

Procés normal:

3. L'administrador omple els camps de usuari i contrasenya.
4. El sistema valida l'usuari.

Cas d'ús : Crear categories

Resum de la funcionalitat: L'administrador pot fer una nova categoria de la Web.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema.

Postcondició: -

Procés normal:

1. L'administrador accedeix al manteniment de categories.
2. L'administrador afegix una nova categoria

Cas d'ús : Modifica categories

Resum de la funcionalitat: L'administrador pot fer la modificació d'una categoria de la Web.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema.

Postcondició: -

Procés normal:

3. L'administrador accedeix al manteniment de categories.
4. L'administrador modifica una categoria.

Cas d'ús : Elimina categories

Resum de la funcionalitat: L'administrador pot fer la eliminació d'una categoria de la Web.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema.

Postcondició: -

Procés normal:

5. L'administrador accedeix al manteniment de categories.
6. L'administrador elimina una categoria.

Cas d'ús : Crear subcategories

Resum de la funcionalitat: L'administrador pot fer una nova subcategoria de la Web.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema.

Postcondició: -

Procés normal:

7. L'administrador accedeix al manteniment de subcategories.
8. L'administrador afegeix una nova subcategoria.

Cas d'ús : Modifica subcategories

Resum de la funcionalitat: L'administrador pot fer la modificació d'una subcategoria de la Web.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema

Postcondició: -

Procés normal:

9. L'administrador accedeix al manteniment de subcategories.
10. L'administrador modifica una subcategoria

Cas d'ús : Elimina subcategories

Resum de la funcionalitat: L'administrador pot fer la eliminació d'una subcategoria de la Web.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema

Postcondició: -

Procés normal:

11. L'administrador accedeix al manteniment de subcategories.
12. L'administrador elimina una subcategoria

Cas d'ús : Crea productes

Resum de la funcionalitat: L'administrador pot afegir un nou producte.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema

Postcondició: -

Procés normal:

1. L'administrador accedeix al manteniment de productes.
2. L'administrador pot afegir un nou producte.

Cas d'ús : Modifica productes

Resum de la funcionalitat: L'administrador pot modificar un producte.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema.

Postcondició: -

Procés normal:

3. L'administrador accedeix al manteniment de productes.
4. L'administrador pot modificar un producte.

Cas d'ús : Elimina productes

Resum de la funcionalitat: L'administrador pot eliminar un producte.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema.

Postcondició: -

Procés normal:

5. L'administrador accedeix al manteniment de productes.
6. L'administrador pot eliminar un producte.

Cas d'ús : Consulta comandes

Resum de la funcionalitat: L'administrador pot fer la consulta de les comandes que hi ha pendents i pot marcar-les com a entregades.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema.

Postcondició: -

Procés normal:

1. L'administrador accedeix a les comandes.
2. El sistema li mostra les comandes no entregades
3. L'administrador pot marcar les comandes com a entregades.

Cas d'ús : Crea usuaris

Resum de la funcionalitat: L'administrador pot afegir usuaris registrats a la Web.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema.

Postcondició: -

Procés normal:

1. L'administrador accedeix a usuaris registrats.
2. El sistema li mostra els usuari que estan registrats a la Web.
3. L'administrador pot afegir un nou usuari.

Cas d'ús : Modifica usuaris

Resum de la funcionalitat: L'administrador pot modificar les dades personals d'úsuaris registrats a la Web.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema.

Postcondició: -

Procés normal:

4. L'administrador accedeix a usuaris registrats.
5. El sistema li mostra els usuari que estan registrats a la Web.
6. L'administrador pot modificar usuaris registrats de la Web.

Cas d'ús : Elimina usuaris

Resum de la funcionalitat: L administrador pot eliminar usuaris registrats a la Web.

Actors: Administrador.

Precondició: L'administrador ha d'estar identificat pel sistema.

Postcondició: -

Procés normal:

7. L'administrador accedeix a usuaris registrats.
8. El sistema li mostra els usuaris que estan registrats a la Web.
9. L'administrador pot eliminar usuaris registrats a la Web.

2.5 Model conceptual de classes

El model conceptual de classes mostra gràficament els conceptes (classes d'objectes), els atributs i les associacions més importants del domini del problema.

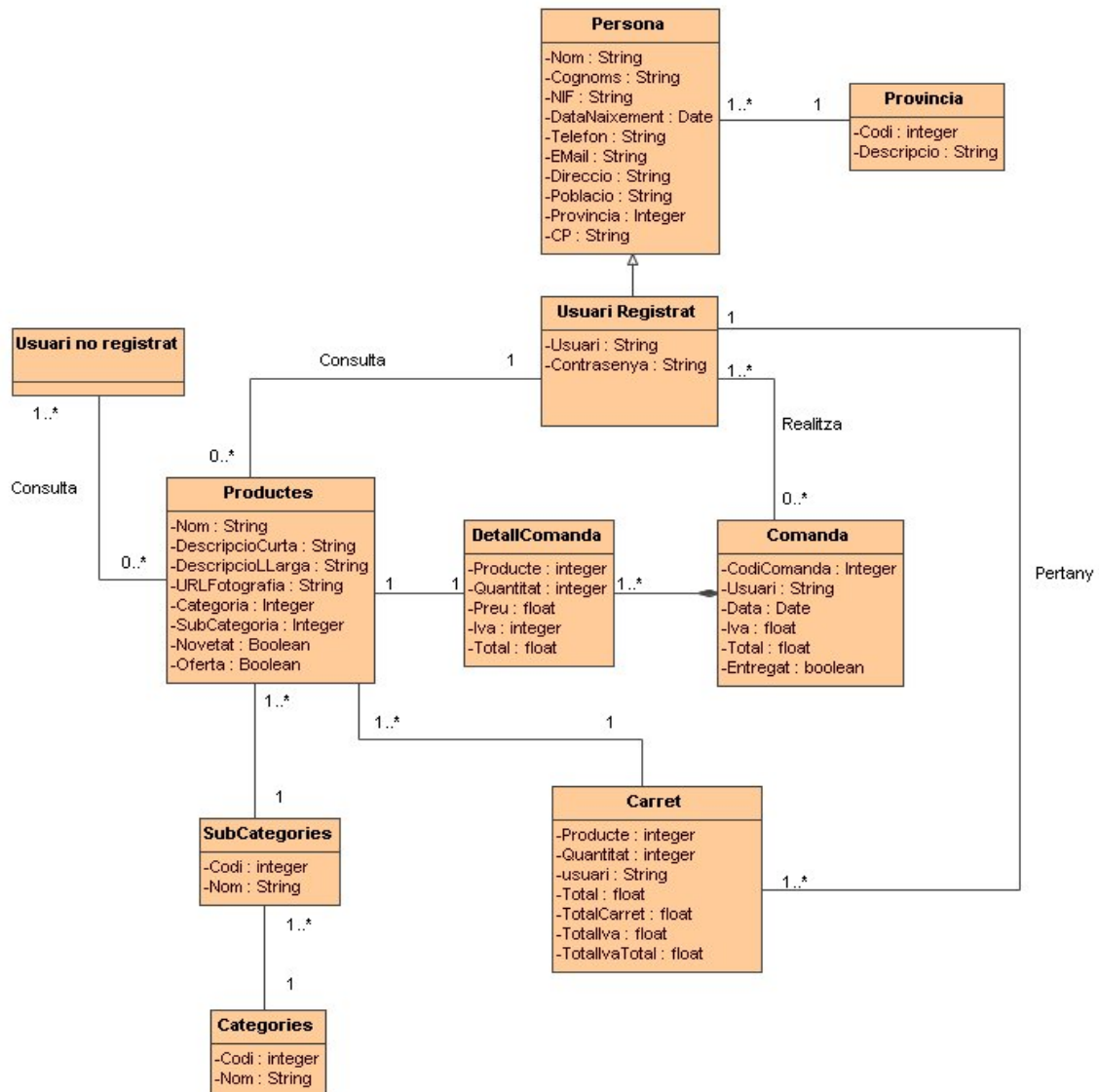


Figura 5

3. Disseny

3.1 Arquitectura del sistema

La programació per capes és un estil de programació en la que l'objectiu primordial és la separació de la lògica de negoci de la lògica de disseny, un exemple bàsic és separar la capa de dades de la capa de presentació a l'usuari.

El principal avantatge d'aquest estil, és que el desenvolupament es pot realitzar en diversos nivells i en el cas d'un canvi només s'ataca al nivell afectat sense haver de revisar entre codi barrejat.

El projecte tenim tres capes bàsiques, que són presentació, negoci i base de dades i dues capes més, una capa comuna a totes les altres capes on tindrem el model de classes personalitzades i una altra capa pel membership on es controlaran tots els processos de validació d'usuari.

Resumim breument les 5 capes que tenim al projecte:

1 Capa de presentació: És la que veu l'usuari, presenta el sistema a l'usuari, li comunica la informació i captura la informació. Aquesta capa es comunica únicament amb la capa de negoci.

2.- Capa de negoci: En aquesta capa és on es marquen totes les regles del projecte. Aquesta capa es comunica amb la capa de presentació, per rebre les sol·licituds i sol·licitar al gestor de base de dades para guardar o recuperar dades.

3.- Capa de dades: És la capa on es troben les dades. Reben sol·licituds d'emmagatzematge o recuperació d'informació des de la capa de negoci.

4.- Capa comuna: És la capa on es troba el model de classes personalitzades, i el control d'errors. Aquesta capa és visible des de totes les altres capes.

5.- Capa membership: Controla la validació dels usuaris que fan login a la Web.

3.2 Diagrames de seqüència

Es registra a la Web:

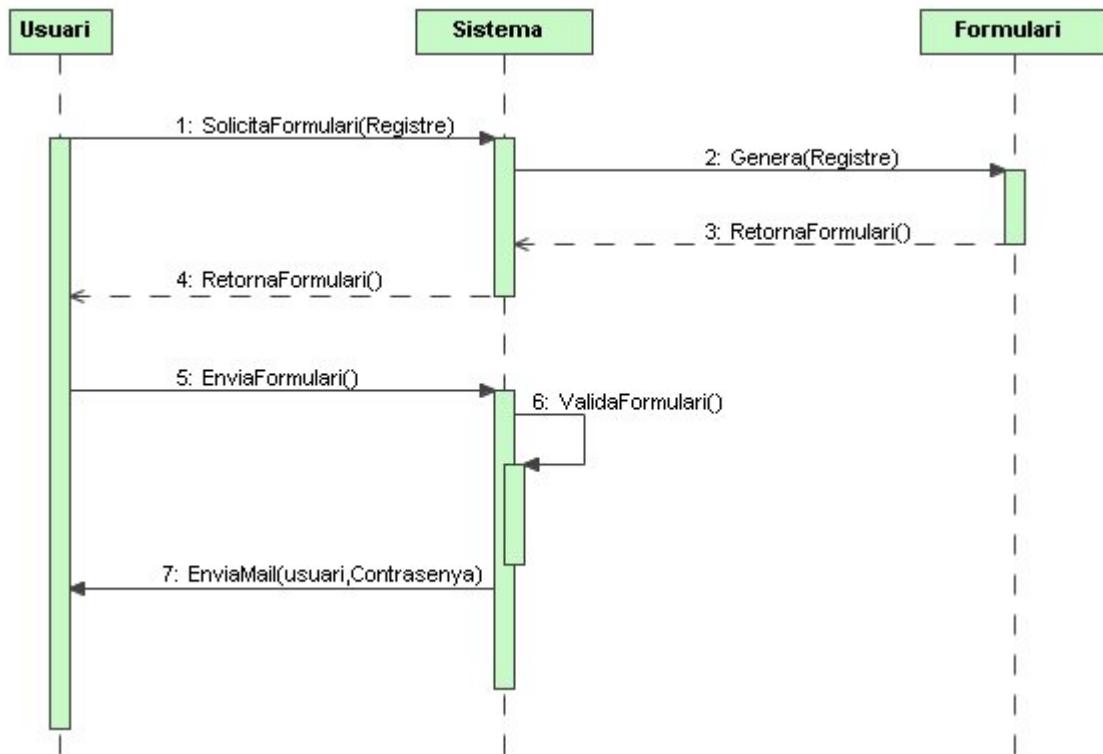


Figura 6

Fa login a la Web

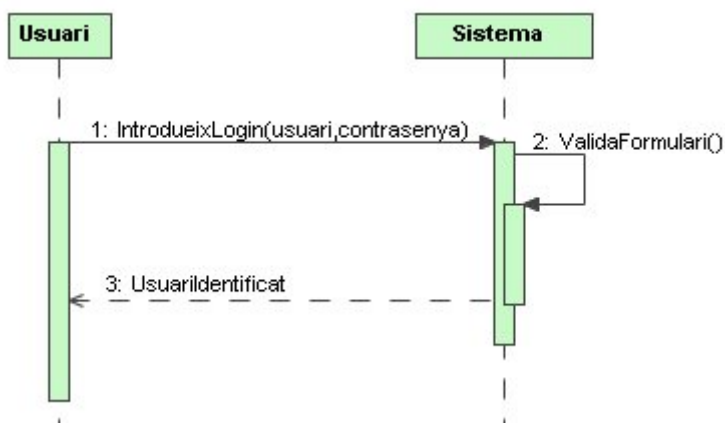


Figura 7

Consulta Productes

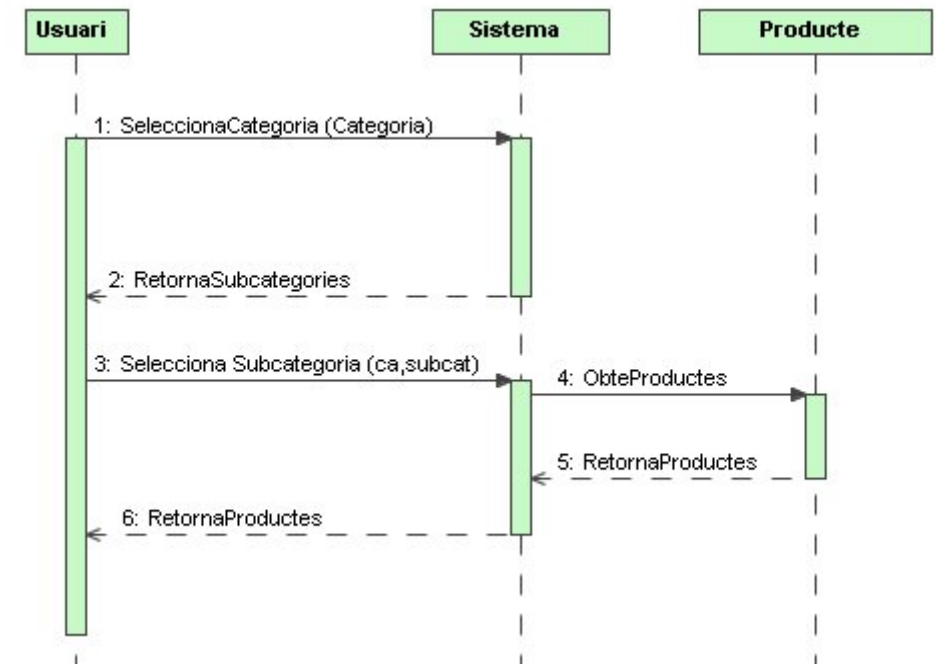


Figura 8

Realitza comanda

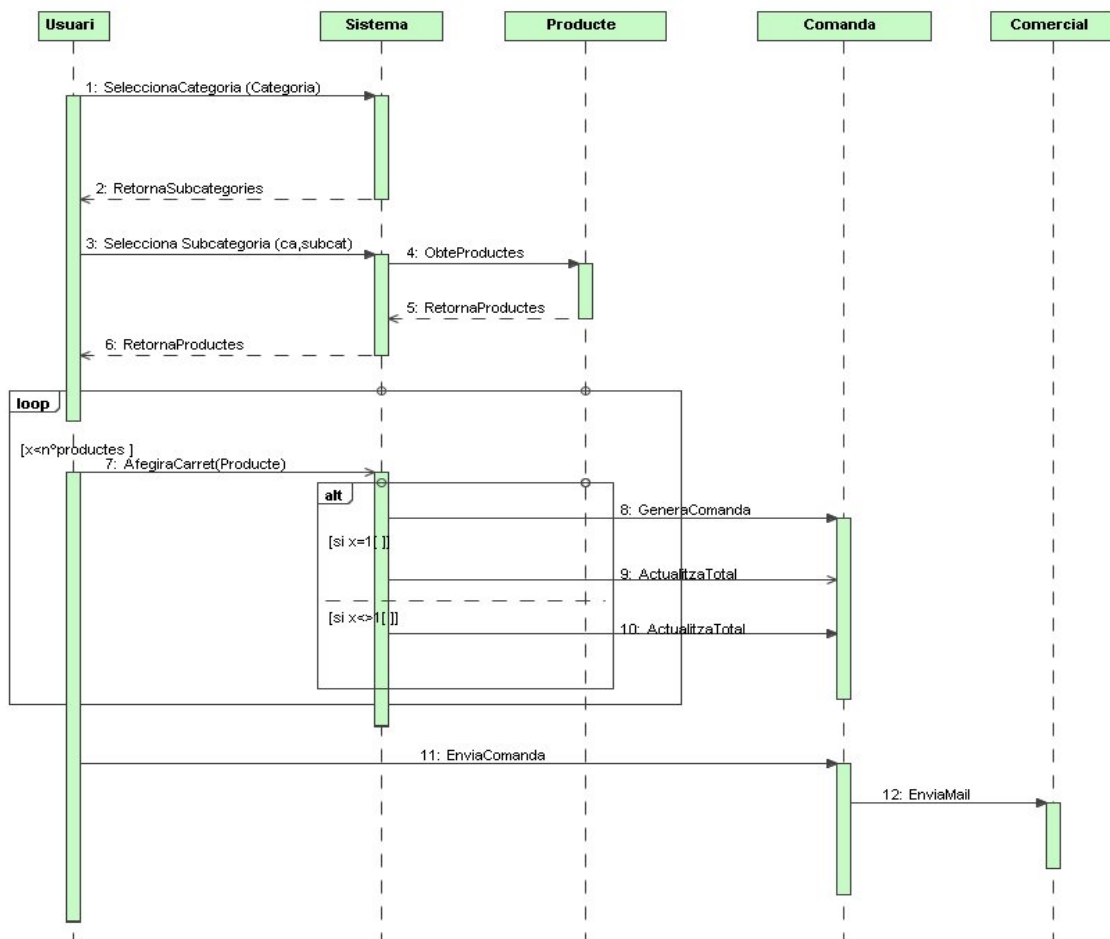


Figura 9

Realitza cerques

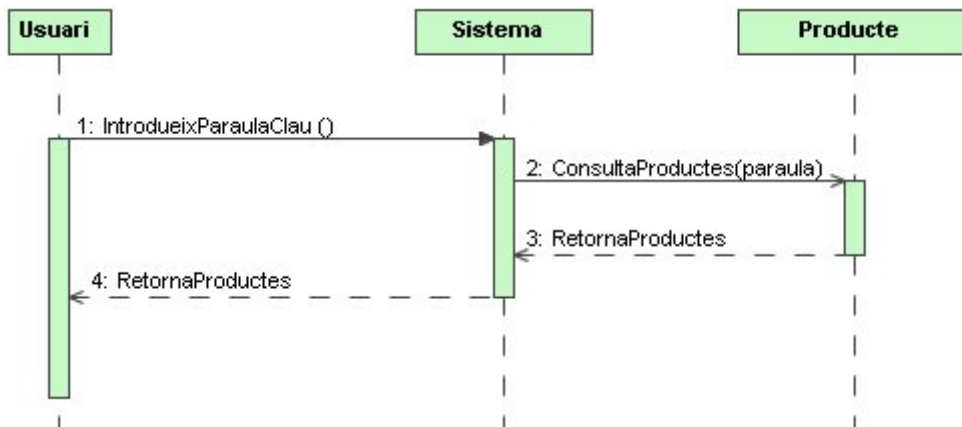


Figura 10

Contacta amb administrador

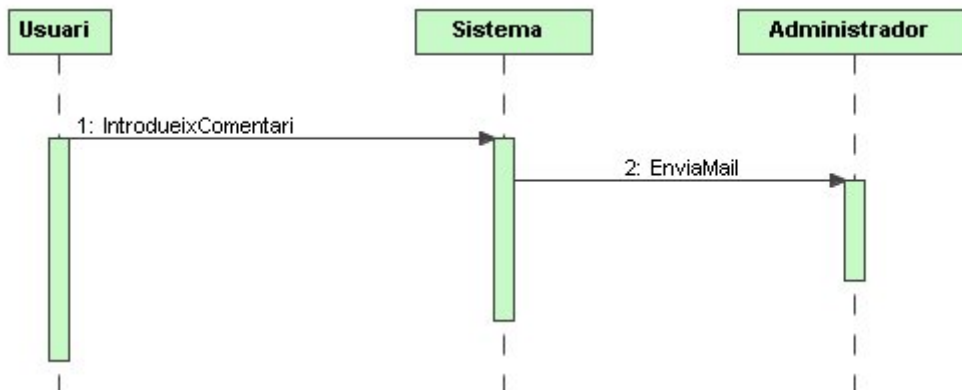


Figura 11

3.3 Model de classes de disseny

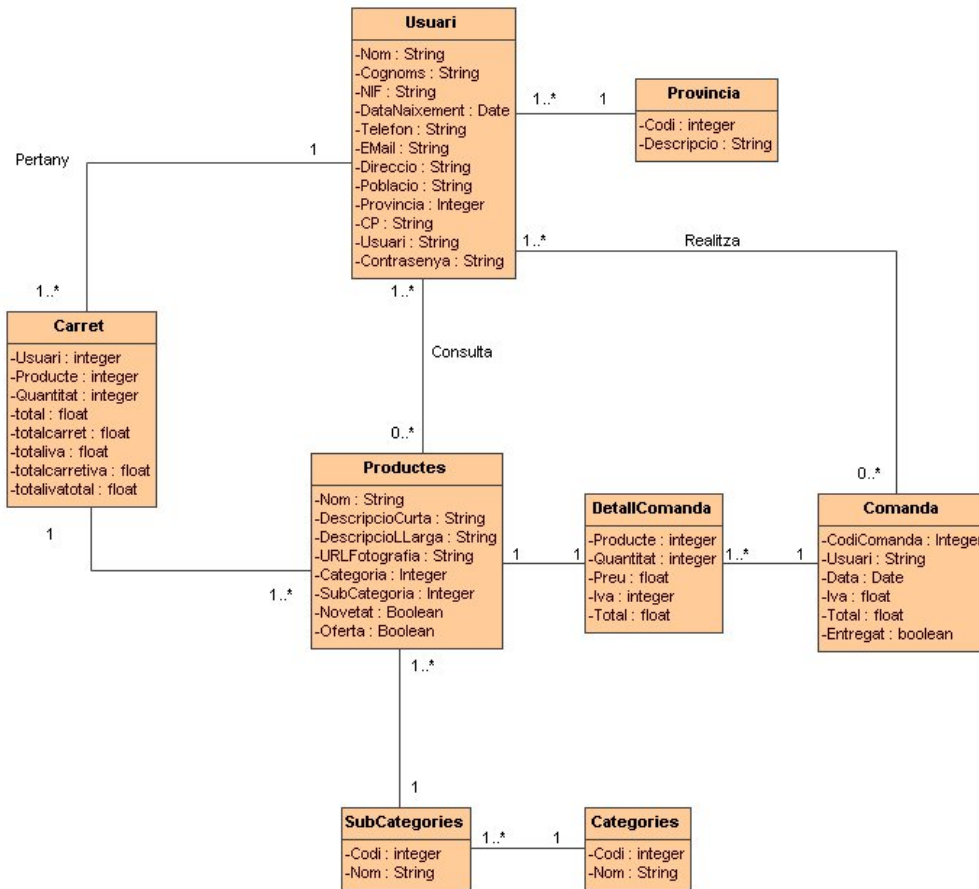


Figura 12

3.4 Model de dades

A l'hora desenvolupar i entregar el projecte s'ha utilitzat el servidor de base de dades SQLServer 2005, encara que en tota l'aplicació s'ha utilitzat una classe base anomenada DBProviderfactory. Aquesta classe ens permet connectar a una base de dades simplement passant-li com a paràmetre la base de dades i el proveïdor de base de dades. Per tant utilitzant aquesta classe i respectant ANSI SQL s'obté una Web crossdatabase.

Per utilitzar aquesta classe importem primer l'espai de noms: `System.Data.Common` I la utilitzem d'aquesta forma:

```

factory = DbProviderFactories.GetFactory(providerName)
con = factory.CreateConnection()
con.ConnectionString = connectionstring
con.Open()

com = factory.CreateCommand
com.CommandText = sql
com.Connection = con

Dim paramId As DbParameter = factory.CreateParameter()
paramId.ParameterName = PARM_CATEGORIA_ID
paramId.DbType = DbType.Int32
paramId.Value = id
com.Parameters.Add(paramId)
  
```

```
Using reader As DbDataReader = com.ExecuteReader()  
End Using
```

Si ens fixem:

- En comptes d'obrir un SqlDataReader obrim un DbDataReader
- A l'hora de definir els paràmetres utilitzem DbParameter en comptes de SqlParameter

Per tal de treballar amb la base de dades s'ha utilitzat ADO.NET.

ADO.NET és el més recent en una extensa línia de tecnologies d'accés a bases de dades que va començar fa diversos anys amb la interfície de programació d'aplicacions (API) de la connectivitat oberta de base de dades (Open Database Connectivity, ODBC). Transcorregut aquest temps, han succeït diverses coses interessants. Per exemple, COM va començar a treballar en el terreny de les bases de dades, el que va culminar amb OLE DB.

Posteriorment, ActiveX® Data Objects (ADO) (més o menys, una versió automatitzada de OLE DB) s'ha triat per a regir la comunitat Visual Basic® i ASP dels desenvolupadors de base de dades basats en Windows®. Amb .NET, Microsoft ofereix un marc de treball general (biblioteca de classe de marc de treball) que s'ampliarà per a cobrir totes les API de Windows existents, entre moltes altres coses. En concret, inclourà diverses biblioteques d'ús freqüent que, en l'actualitat, es troben disponibles a través d'objectes COM independents. Entre elles, es troben els models d'objectes XML i ADO que s'han integrat en un subarbre de classes denominat ADO.NET.

ADO.NET representa el substrat que serà la base de les aplicacions .NET compatibles amb dades. A diferència de ADO, ADO.NET s'ha dissenyat seguint específicament unes directrius més generals i menys orientades a la base de dades. ADO.NET reuneix totes les classes que permeten l'accés a dades. Aquestes classes representen els objectes que contenen dades i que mostren les capacitats normals de les bases de dades: índexs, ordenació, vistes. Encara que ADO.NET és la solució definitiva per a les aplicacions de base de dades de .NET, destaca per un disseny global que no es fixa tant en les bases de dades com el model ADO.

ADO.NET és un nou model de programació per a l'accés a dades que necessita un compromís i un enteniment complets. Qualsevol tasca de ADO resulta d'extrema ajuda per a generar aplicacions consistents i per a resoldre antics problemes d'una forma diferent, encara que de manera més elegant i consistent.

L'accés als orígens de dades en ADO.NET es regeix a través de proveïdors administrats. Respecte a les funcions, un proveïdor administrat és molt semblat a un proveïdor OLE DB, encara que amb dues diferències importants. Primer, funcionen en l'entorn .NET i recuperen i exposen dades a través de classes .NET, com SqlDataReader i DataTable. Segon, l'arquitectura resulta més senzilla, ja que s'ha optimitzat per a .NET. ADO.NET incorpora dos tipus de proveïdors administrats: un per a SQL Server™ 7.0 i posterior i altre per a tots els altres proveïdors OLE DB que pugui tenir instal·lats. Les classes que s'utilitzen en ambdues situacions són distintes, encara que segueixen una convenció de nomenclatura semblant. Els noms són iguals, excepte pels prefixos. El prefix és SQL en el primer cas i ADO en el segon.

ADO.NET ofereix dos objectes per manipular les dades d'un origen de dades. Es tracta dels objectes DataSet i SqlDataReader. El primer és una memòria caché dels registres que es poden visitar en qualsevol adreça i modificar com es desitgi. El segon és un objecte optimitzat per a desplaçar-se per registres de només lectura i només cap a endavant. L'aspecte de DataSet és similar a un cursor estàtic, mentre que l'objecte SqlDataReader és l'equivalent en .NET al cursor de només lectura de ADO.

A tota l'aplicació s'ha utilitzat Datareaders, ja que són molt més ràpids en execució. Segons "Optimización para asp.net 2.0. DotNetMania", un Datareader amb mil registres pot ser fins 30 vegades més ràpid que una Dataset amb els mateixos mil registres. Com la Web pot acumular molts productes vaig decidir realitzar tota la Web amb Datareaders, així quan hi hagi un gran volum de dades la Web carregarà més ràpidament.

3.5 Diagrama lògic de base de dades

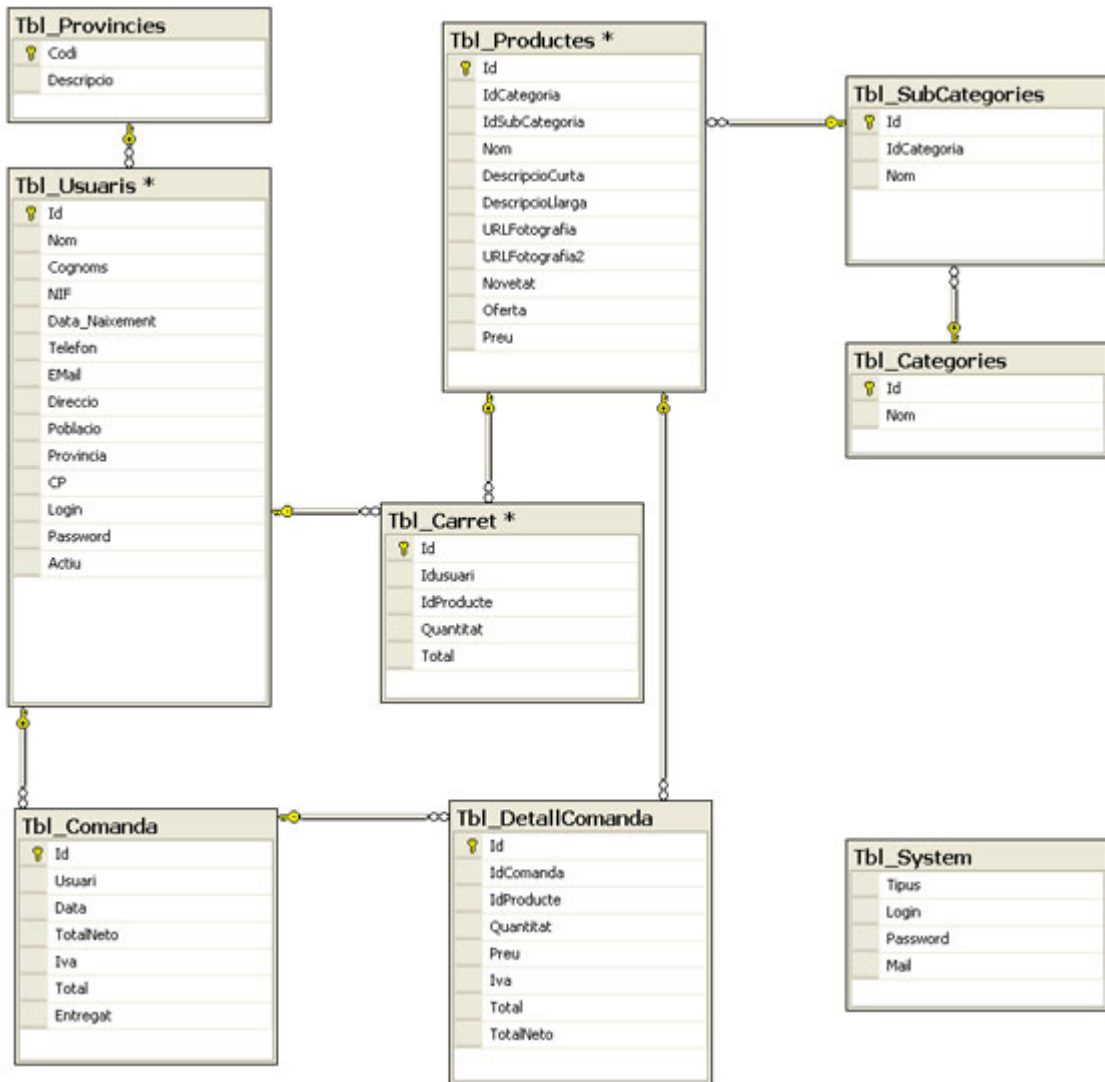


Figura 13

3.5 Generació de la base de dades

```


Tbl_Carret

USE [ArtNetDatabase]
GO
/***** Object: Table [dbo].[Tbl_Carret]   Script Date: 06/09/2007 14:11:08 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tbl_Carret](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Idusuari] [int] NOT NULL,
    [IdProducte] [int] NOT NULL,
    [Quantitat] [int] NOT NULL,
    [Total] [real] NOT NULL,
    CONSTRAINT [PK_Tbl_Carret] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[Tbl_Carret] WITH CHECK ADD CONSTRAINT [FK_Tbl_Carret_Tbl_Productes] FOREIGN
KEY([IdProducte])
REFERENCES [dbo].[Tbl_Productes] ([Id])
GO
ALTER TABLE [dbo].[Tbl_Carret] CHECK CONSTRAINT [FK_Tbl_Carret_Tbl_Productes]
GO
ALTER TABLE [dbo].[Tbl_Carret] WITH CHECK ADD CONSTRAINT [FK_Tbl_Carret_Tbl_Usuaris] FOREIGN
KEY([Idusuari])
REFERENCES [dbo].[Tbl_Usuaris] ([Id])
GO
ALTER TABLE [dbo].[Tbl_Carret] CHECK CONSTRAINT [FK_Tbl_Carret_Tbl_Usuaris]
```

```


Tbl_Categories

USE [ArtNetDatabase]
GO
/***** Object: Table [dbo].[Tbl_Categories]   Script Date: 06/09/2007 14:12:42 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tbl_Categories](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Nom] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Tbl_Categories] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Tbl_Comanda

```
USE [ArtNetDatabase]
GO
/***** Object: Table [dbo].[Tbl_Comanda]   Script Date: 06/09/2007 14:14:00 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tbl_Comanda](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Usuari] [int] NOT NULL,
    [Data] [datetime] NOT NULL,
    [TotalNeto] [money] NOT NULL,
    [Iva] [money] NOT NULL,
    [Total] [money] NOT NULL,
    [Entregat] [bit] NOT NULL CONSTRAINT [DF_Tbl_Comanda_Entregat] DEFAULT ((0)),
    CONSTRAINT [PK_Tbl_Comanda] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[Tbl_Comanda] WITH CHECK ADD CONSTRAINT [FK_Tbl_Comanda_Tbl_Usuaris]
FOREIGN KEY([Usuari])
REFERENCES [dbo].[Tbl_Usuaris] ([Id])
GO
ALTER TABLE [dbo].[Tbl_Comanda] CHECK CONSTRAINT [FK_Tbl_Comanda_Tbl_Usuaris]
```

Tbl_DetallComanda

```
USE [ArtNetDatabase]
GO
/***** Object: Table [dbo].[Tbl_DetallComanda]   Script Date: 06/09/2007 14:14:17 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tbl_DetallComanda](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdComanda] [int] NOT NULL,
    [IdProducte] [int] NOT NULL,
    [Quantitat] [int] NOT NULL,
    [Preu] [money] NOT NULL,
    [Iva] [money] NOT NULL,
    [Total] [money] NOT NULL,
    [TotalNeto] [money] NOT NULL,
    CONSTRAINT [PK_Tbl_DetallComanda] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[Tbl_DetallComanda] WITH CHECK ADD CONSTRAINT
[FK_Tbl_DetallComanda_Tbl_Comanda] FOREIGN KEY([IdComanda])
REFERENCES [dbo].[Tbl_Comanda] ([Id])
GO
ALTER TABLE [dbo].[Tbl_DetallComanda] CHECK CONSTRAINT [FK_Tbl_DetallComanda_Tbl_Comanda]
GO
ALTER TABLE [dbo].[Tbl_DetallComanda] WITH CHECK ADD CONSTRAINT
[FK_Tbl_DetallComanda_Tbl_Productes] FOREIGN KEY([IdProducte])
REFERENCES [dbo].[Tbl_Productes] ([Id])
GO
ALTER TABLE [dbo].[Tbl_DetallComanda] CHECK CONSTRAINT [FK_Tbl_DetallComanda_Tbl_Productes]
```


Tbl_Productes

```
USE [ArtNetDatabase]
GO
/***** Object: Table [dbo].[Tbl_Productes]   Script Date: 06/09/2007 14:14:48 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tbl_Productes](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdCategoria] [int] NOT NULL,
    [IdSubCategoria] [int] NOT NULL,
    [Nom] [nvarchar](50) NOT NULL,
    [DescripcionCurta] [ntext] NOT NULL,
    [DescripcionLarga] [ntext] NOT NULL,
    [URLFotografia] [nvarchar](150) NULL,
    [URLFotografia2] [nvarchar](150) NULL,
    [Novetat] [bit] NOT NULL,
    [Oferta] [bit] NOT NULL,
    [Preu] [real] NOT NULL,
    CONSTRAINT [PK_Tbl_Productes] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]

GO
ALTER TABLE [dbo].[Tbl_Productes] WITH CHECK ADD CONSTRAINT
[FK_Tbl_Productes_Tbl_SubCategories] FOREIGN KEY([IdSubCategoria])
REFERENCES [dbo].[Tbl_SubCategories] ([Id])
GO
ALTER TABLE [dbo].[Tbl_Productes] CHECK CONSTRAINT [FK_Tbl_Productes_Tbl_SubCategories]
```

Tbl_Provincies

```
USE [ArtNetDatabase]
GO
/***** Object: Table [dbo].[Tbl_Provincies]   Script Date: 06/09/2007 14:15:10 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tbl_Provincies](
    [Codi] [int] IDENTITY(1,1) NOT NULL,
    [Descripcion] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Tbl_Provincies] PRIMARY KEY CLUSTERED
(
    [Codi] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
```

Tbl_Subcategories

```
USE [ArtNetDatabase]
GO
/***** Object: Table [dbo].[Tbl_SubCategories]   Script Date: 06/09/2007 14:15:40 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tbl_SubCategories](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [IdCategoria] [int] NOT NULL,
    [Nom] [nvarchar](50) NOT NULL,
    CONSTRAINT [PK_Tbl_SubCategories] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[Tbl_SubCategories] WITH CHECK ADD CONSTRAINT
[FK_Tbl_SubCategories_Tbl_Categories] FOREIGN KEY([IdCategoria])
REFERENCES [dbo].[Tbl_Categories] ([Id])
GO
ALTER TABLE [dbo].[Tbl_SubCategories] CHECK CONSTRAINT [FK_Tbl_SubCategories_Tbl_Categories]
```

Tbl_System

```
USE [ArtNetDatabase]
GO
/***** Object: Table [dbo].[Tbl_System]   Script Date: 06/09/2007 14:16:10 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tbl_System](
    [Tipus] [nchar](20) NOT NULL,
    [Login] [nchar](20) NULL,
    [Password] [nchar](20) NULL,
    [Mail] [nchar](30) NULL
) ON [PRIMARY]
```

Tbl_Usuaris

```
USE [ArtNetDatabase]
GO
/***** Object: Table [dbo].[Tbl_Usuaris]  Script Date: 06/09/2007 14:16:47 *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE TABLE [dbo].[Tbl_Usuaris](
    [Id] [int] IDENTITY(1,1) NOT NULL,
    [Nom] [nvarchar](50) NOT NULL,
    [Cognoms] [nvarchar](150) NOT NULL,
    [NIF] [nvarchar](10) NOT NULL,
    [Data_Naixement] [datetime] NOT NULL,
    [Telefon] [nvarchar](15) NOT NULL,
    [EMail] [nvarchar](50) NOT NULL,
    [Direccio] [nvarchar](150) NOT NULL,
    [Poblacio] [nvarchar](50) NOT NULL,
    [Provincia] [int] NOT NULL,
    [CP] [nvarchar](5) NOT NULL,
    [Login] [nvarchar](50) NOT NULL,
    [Password] [nvarchar](50) NOT NULL,
    [Actiu] [bit] NOT NULL CONSTRAINT [DF_Tbl_Usuaris_Activo] DEFAULT ((0)),
    CONSTRAINT [PK_Tbl_Usuaris] PRIMARY KEY CLUSTERED
(
    [Id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF, IGNORE_DUP_KEY = OFF,
    ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]

GO
ALTER TABLE [dbo].[Tbl_Usuaris] WITH CHECK ADD CONSTRAINT [FK_Tbl_Usuaris_Tbl_Provincies]
FOREIGN KEY([Provincia])
REFERENCES [dbo].[Tbl_Provincies] ([Codi])
GO
ALTER TABLE [dbo].[Tbl_Usuaris] CHECK CONSTRAINT [FK_Tbl_Usuaris_Tbl_Provincies]
```

3.6 Interfície gràfica de la Web

Pel disseny de la Web s'ha utilitzat Masterpages.

La característica de les MasterPages ens proporciona la habilitat de definir una estructura i uns elements d'interfície comuns per la Web, així com la capçalera de la pàgina, amb el menú horitzontal, el menú vertical, la part de la identificació a la Web i el carret de la compra.

Així evitem la duplicació innecessària de codi per estructures o comportaments de la Web que són compartits.

Per tant des de tota la Web podrem utilitzar els dos menús disponibles (horitzontal i vertical), podrem identificar-nos a la Web, i si estem identificats a la Web podrem tenir sempre visible el carret de la compra amb els productes escollits.

A part de totes aquestes funcionalitats, que són comunes per a totes les pantalles descriurem les funcionalitats pròpies de cada una de les pantalles de la Web.

Home:

The screenshot shows the home page of ArtNet, a multidisciplinary art gallery. The layout includes a top navigation bar with icons for Home, Products, News, Offers, and Contact. A left sidebar titled 'DISCIPLINAS' lists various art forms such as Painting, Sculpture, and Photography. The main content area is divided into sections: a welcome message, a login section with fields for username and password, and a contact section. The footer indicates the website was developed by Fernando Escalera Almendros.

Figura 14

Quan l'usuari accedeix a la Web va a para a la pàgina de home. Des de aquesta pantalla l'usuari pot:

- Accedir a contacte per enviar qualsevol comentari a l'administrador.
- Pot veure la descripció de l'empresa i un petit manual que explica com navegar per la Web.

Productes:



HOME PRODUCTOS NOVEDADES OFERTAS CONTACTO


DISCIPLINAS

- Pintura
 - ◊ Técnica mixta
 - ◊ Plástica
 - ◊ Pictórica
- Escultura
 - ◊ Barro
 - ◊ Madera
 - ◊ Mármol
- Fotografía
 - ◊ Blanco y negro
 - ◊ Paisajes
 - ◊ Sellos
- Grabados
 - ◊ Geométricos
 - ◊ Históricos
 - ◊ Variados

Productos

BUSCAR:


Puerto tercermundista



Esta es una obra que muestra el amor que Adán tenía por Eva en el paraíso. Se puede apreciar los contornos difuminados del paisaje.

Disciplina: Fotografía - Blanco y negro
Precio: 165,50 €

Despedida y cierre



Esta es una obra que muestra el amor que Adán tenía por Eva en el paraíso. Se puede apreciar los contornos difuminados del paisaje.

Disciplina: Fotografía - Blanco y negro
Precio: 99,00 €

Identificación

Para poder realizar pedidos necesitará estar identificado en la web.
Introduzca usuario y contraseña:

Usuario:

Contraseña:


Si no está registrado en la web puede registrarse en ArtNet gratuitamente.
Registrarme ahora >>

1 2

web realizada por Fernando Escalera Almendros

Figura 15

Quan l'usuari accedeix a la pantalla de productes pot:

- Visualitzar els productes i navegar través de la paginació de productes.
- Pot ampliar la imatge.
- Si l'usuari està registrat i apareixerà una icona, , per poder afegir els productes al carret de la compra.

Producte ampliat:



- DISCIPLINAS**
- Pintura
 - ◊ Técnica mixta
 - ◊ Plástica
 - ◊ Pictórica
 - Escultura
 - ◊ Barro
 - ◊ Madera
 - ◊ Mármol
 - Fotografía
 - ◊ Blanco y negro
 - ◊ Paisajes
 - ◊ Sellos
 - Grabados
 - ◊ Geométricos
 - ◊ Históricos
 - ◊ Variados

Productos

BUSCAR:



Puerto tercermundista

Esta es una obra que muestra el amor que Adán tenía por Eva en el paraíso. Se puede apreciar los contornos difuminados del paisaje.

Disciplina: Fotografía - Blanco y negro

Precio: 165,50 €

Identificación

Para poder realizar pedidos necesitará estar identificado en la web. Introduzca usuario y contraseña:

Usuario:

Contraseña:


Si no está registrado en la web puede registrarse en ArtNet gratuitamente. [Registrarme ahora >>](#)



Volver

Figura 16

Quan l'usuari accedeix a la pantalla de producte ampliat pot:

- Visualitzar el producte seleccionat.
- Si l'usuari està registrat apareixerà una icona, , per poder afegir el producte al carret de la compra.
- Tornar a la pàgina de productes.

Usuari registrat consultant el producte:



HOME PRODUCTOS NOVEDADES OFERTAS CONTACTO

DISCIPLINAS

- Pintura
 - ◊ Técnica mixta
 - ◊ Plástica
 - ◊ Pictórica
- Escultura
 - ◊ Barro
 - ◊ Madera
 - ◊ Mármol
- Fotografía
 - ◊ Blanco y negro
 - ◊ Paisajes
 - ◊ Sellos
- Grabados
 - ◊ Geométricos
 - ◊ Históricos
 - ◊ Variados

Productos

BUSCAR:

Bienvenido fer
Su cesta de la compra:

Producto	Cant.	Precio	Total
Puerto tercermundista	1	165,50€	165,50€
Despedida y cierre	1	99,00€	99,00€
		Total Neto:	264,50€
		Total Iva:	42,32€
		Total pagar:	306,82€

Comprar

1 2

Figura 17

Detall del carret de la compra:

Bienvenido fer
Su cesta de la compra:

Producto	Cant.	Precio	Total
Puerto tercermundista	1	165,50€	165,50€
Despedida y cierre	1	99,00€	99,00€
		Total Neto:	264,50€
		Total Iva:	42,32€
		Total pagar:	306,82€

Comprar

Figura 18

L'usuari pot en tot moment veure el carret de la compra amb les imatges dels productes que ha afegit al carret

Contacte:

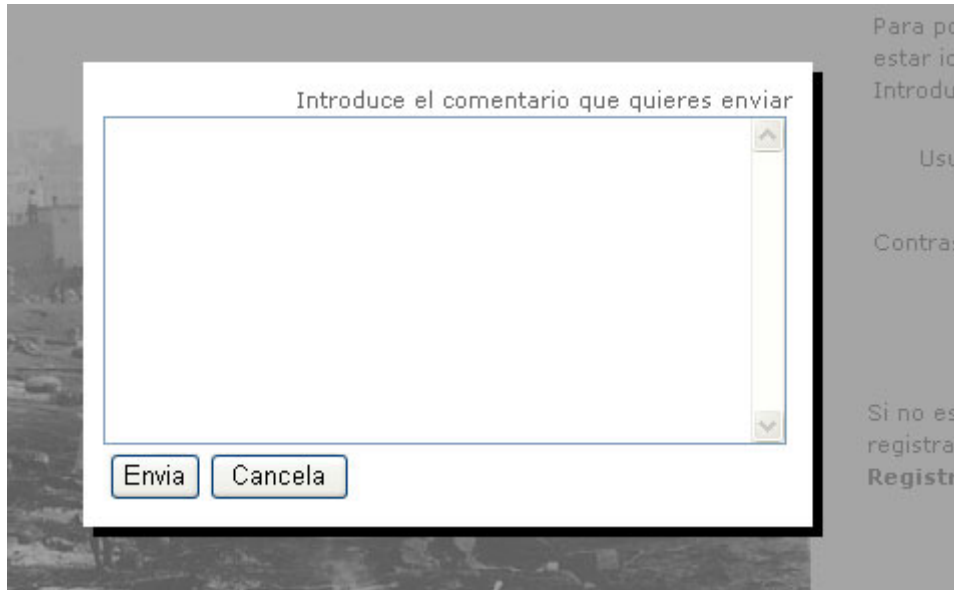


Figura 19

Quan l'usuari fa clic sobre el menú contacto li apareix una pantalla on l'usuari pot:

- Escriure un comentari a l'administrador de la Web.

3.6 Interfície gràfica del gestor de continguts

Al igual que la Web, també s'ha utilitzat una masterpage pel gestor amb un link l'índex del gestor.

Pantalla de Login

Login:



Artnet
Galeria de arte multidisciplinar

INDICE

Acceso al gestor ArtNet

Desde esta pantalla podrá logonearse para poder entrar en la aplicación de mantenimiento de ArtNet.

Introduzca usuario y contraseña

Usuario:

Password:

Enviar

Figura 20

Quan l'administrador accedeix al gestor el primer que li demana és que s'identifiqui al gestor.

Pantalla d'índice

Index:



 INDICE

Índice gestor de contenidos

Desde esta pantalla tiene acceso a los contenidos de la base de datos de la web ArtNet. Cada uno de los links que se muestran a continuación contiene un programa de gestión para facilitar la entrada, modificación y eliminación de los datos.



Mantenimiento de productos

Desde esta pantalla tendrá acceso a todos los productos de ArtNet. Para acceder a los productos deberá seleccionar la categoría y la subcategoría asociada al producto. Podrá dar de alta, modificar y eliminar productos de la base de datos de ArtNet.



Mantenimiento de usuarios

Desde esta pantalla podrá realizar el mantenimiento de los usuarios registrados en la web.



Mantenimiento de Categorías

Desde esta pantalla podrá realizar el mantenimiento de categorías.



Mantenimiento de SubCategorías

Desde esta pantalla podrá realizar el mantenimiento de subcategorías.



Consulta de pedidos

Desde esta pantalla podrá realizar consultas de los pedidos realizados a través de la web así como eliminar pedidos, o marcar los pedidos como entregados al cliente.

Figura 21

Aquesta és la pantalla principal des de la qual l'administrador accedirà a totes les aplicacions de manteniment del gestor.

Pantallas de mantenimiento de productos

Mantenimiento de productos:



Desde esta pantalla tendrá acceso a todos los productos de Artnet.
Para acceder a los productos seleccione la categoría y la subcategoría asociada al producto. Podrá dar de alta, modificar y eliminar productos de la base de datos de Artnet.

Categoría:

SubCategoría:

		Nombre Producto	Precio	Categoría	SubCategoría
	<input type="checkbox"/>	El pájaro sobre el nido	120,50	Pintura	Técnica mixta
	<input checked="" type="checkbox"/>	La libertad de los pájaros	125,50	Pintura	Técnica mixta
	<input checked="" type="checkbox"/>	Pájaro de viento	150,90	Pintura	Técnica mixta

Nuevo Producto:

Figura22

En aquesta pantalla l'usuari pot:

- Filtrar els productes per categoria i subcategoria.
- Eliminar productes
- Accedir a la pantalla per poder modificar o afegir productes.

Afegir i modificar productes:



Desde esta pantalla podrá dar de alta nuevos producto y modificar productos existentes.

Nombre:

Categoría:

SubCategoría:

Descripción corta:

Descripción larga:

Precio:

Novedad:

Oferta:

Figura 23

Pantalles de manteniment d'usuaris

Manteniment d'usuaris:



Mantenimiento de usuarios

Desde esta pantalla podrá relizar el mantenimiento de usuarios.

Tipos de usuario: **Activos**

	Código	Login	Nombre	NIF	Teléfono	E-Mail	Activo	Ha realizado pedido
	23	fer	Marcos Alonso Sevilla	45521414B	545454565	marcos@hotmail.com	Sí	Sí
	30	carlota	Carlota García Moreno	43516358L	46546513	yo@hotmail.com	Sí	Sí

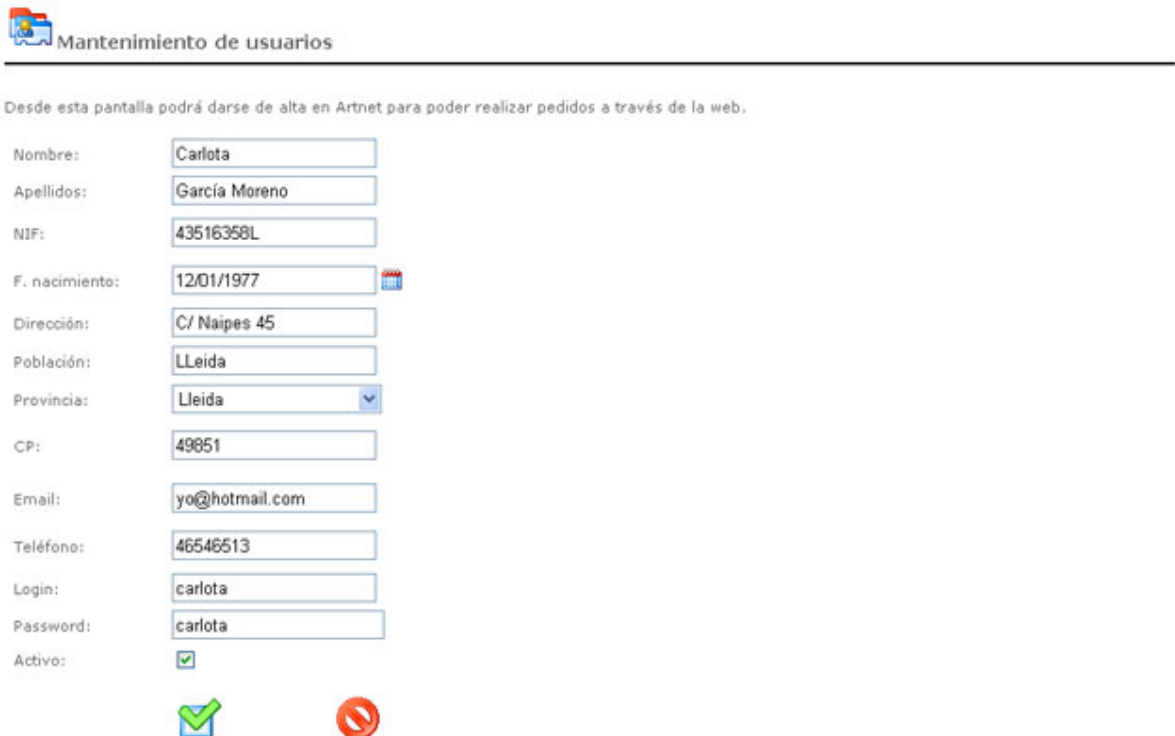
Nuevo usuario:

Figura 24

- Aquesta és la pantalla des de on l'administrador pot:
- Filtrar usuarios segons el seu estat: Actiu i no actiu.
 - Eliminar usuarios
 - Accedir a la pantalla per afegir o modificar usuarios.

Tant per el formulari d'usuari com pel carret de la compra s'ha utilitzat controls d'usuari. Els controls d'usuari ens permeten utilitzar aquests controls en diferents pàgines sense haver de duplicar codi o fer diverses pantalles iguals.

Afegir i modificar usuarios:



Mantenimiento de usuarios

Desde esta pantalla podrá darse de alta en Artnet para poder realizar pedidos a través de la web.

Nombre:

Apellidos:

NIF:

F. nacimiento:

Dirección:

Población:

Provincia:

CP:

Email:

Teléfono:

Login:

Password:

Activo:

Figura 25

Pantalla de mantenimiento de categorías



Figura 26

Des de aquesta pantalla l'administrador pot afegir, modificar i eliminar categories.

Pantalla de mantenimiento de subcategorias



Figura 27

Des de aquesta pantalla l'administrador pot:

- Filtrar subcategorias per categories.
- Afegir, modificar i eliminar subcategorias.

Pantalla de consulta i revisió de comandes



Mantenimiento de pedidos

Desde esta pantalla podrá realizar consultas de los pedidos realizados a través de la web así como eliminar pedidos, o marcar los pedidos como entregados al cliente.

Pedidos realizados a través de la web:

Mostrar:

		Código	Nombre	Fecha	Total	Iva	Total a pagar	Entregado
		67	Carlota García Moreno	27/05/2007	454,00 €	16,00 %	527,16 €	Si
		69	Marcos Alonso Sevilla	27/05/2007	226,00 €	16,00 %	261,00 €	Si
		71	Marcos Alonso Sevilla	27/05/2007	166,00 €	16,00 %	191,98 €	Si

Detalle del Pedido:

Producto	Cantidad	Precio	Total Neto	Iva	Total
El camino del hombre	1	2,00	100,00 €	16,00 %	115,42 €
La libertad de los pájaros	1	2,00	126,00 €	16,00 %	145,58 €

Figura 28

Des de aquesta pantalla l'administrador pot:

- Filtrar les comandes per entregades o no entregades.
- Marcar les comandes com a entregades o no entregades
- Eliminar comandes
- Consultar el detall de les comandes.

4. Infraestructura del desenvolupament

4.1 ASP.NET – Descripció

ASP.NET és un marc de treball de programació generat en Common Language Runtime que pogués utilitzar-se en un servidor per a generar aplicacions Web. ASP.NET ofereix diversos avantatges importants sobre els models de programació Web anteriors:

Millor rendiment:

ASP.NET és un codi de Common Language Runtime compilat que s'executa en el servidor. A diferència dels seus predecessors, ASP.NET pot aprofitar els avantatges de l'enllaç anticipat, la compilació just-in-time, l'optimització nativa i els serveis de caché des del primer moment. Això suposa un increment espectacular del rendiment abans de si més no escriure una línia de codi.

Compatibilitat amb eines de primer nivell:

El marc de treball d'ASP.NET es complementa amb un dissenyador i una caixa d'eines molt complets en l'entorn integrat de programació (Integrated Development Environment, IDE) de Visual Studio. L'edició WYSIWYG, els controls de servidor d'arrossegat i col·locar i la implementació automàtica són només algunes de les característiques que proporciona aquesta eficaç eina.

Eficàcia i flexibilitat:

Degut al fet que ASP.NET es basa en Common Language Runtime, l'eficàcia i la flexibilitat de tota aquesta plataforma es troba disponible per als programadors d'aplicacions Web. La biblioteca de classes de .NET Framework, la Missatgeria i les solucions d'accés a dades es troben accessibles des del Web de manera uniforme. ASP.NET és també independent del llenguatge, pel que pot triar el llenguatge que millor s'adapti a l'aplicació o dividir l'aplicació en diversos llenguatges. A més, Common Language Runtime garanteix que la inversió existent en programació basada en COM es conserva al migrar a ASP.NET.

Simplicitat:

ASP.NET facilita la realització de tasques comunes, des del senzill enviament de formularis i l'autenticació del client fins a la implementació i la configuració de llocs. Per exemple, el marc de treball de pàgina d'ASP.NET permet generar interfícies d'usuari, que separen clarament la lògica d'aplicació del codi de presentació, i controlar esdeveniments en un senzill model de processament de formularis de tipus Visual Basic. A més, Common Language Runtime simplifica la programació, amb serveis de codi administrat com el recompte de referència automàtic i el recol·lector d'elements no utilitzats.

Facilitat d'ús:

ASP.NET utilitza un sistema de configuració jeràrquic, basat en text, que simplifica l'aplicació de la configuració a l'entorn de servidor i les aplicacions Web. Degut al fet que la informació de configuració s'emmagatzema com a text sense format, es pot aplicar la nova configuració sense l'ajuda d'eines d'administració local. Aquesta filosofia de "administració local zero" s'estén així mateix a la implementació de les aplicacions ASP.NET Framework. Una aplicació ASP.NET Framework s'implementa en un servidor senzillament mitjançant la còpia dels arxius necessaris al servidor. No es requereix reiniciar el servidor, ni tan sols per a implementar o reemplaçar el codi compilat en execució.

Escalabilitat i disponibilitat:

ASP.NET s'ha dissenyat tenint en compte la escalabilitat, amb característiques dissenyades específicament a mesura, amb la finalitat de millorar el rendiment en entorns agrupats i de múltiples processadors. A més, el motor de temps d'execució d'ASP.NET controla i administra els processos de prop, pel que si un no es comporta adequadament (filtracions, bloquejos), es pot crear un procés nou en el seu lloc, el que ajuda a mantenir l'aplicació disponible constantment per a controlar sol·licituds.

Possibilitat de personalització i extensibilitat

ASP.NET presenta una arquitectura ben dissenyada que permet als programadors inserir el seu codi en el nivell adequat. De fet, és possible estendre o reemplaçar qualsevol component del motor de temps d'execució d'ASP.NET amb el seu propi component escrit personalitzat.

La implementació de l'autenticació personalitzada mai ha estat més fàcil.

Seguretat:

Amb l'autenticació de Windows integrada i la configuració per aplicació, es pot tenir la completa seguretat que les aplicacions estan fora de perill.

4.2 ASP.NET – Funcionament

Processament de ASP.NET i IIS

Les aplicacions Web ASP.NET i els serveis Web es processen mitjançant codi que s'executa en una única instància del procés de treball d'ASP.NET (aspnetwp.exe), encara que en equips de diversos processadors es poden configurar diverses instàncies, una per processador.

Si s'habilita l'accés anònim en IIS, IIS crea un testimoni d'accés de Windows per al compte d'usuari d'Internet anònim (normalment, IUSR_MACHINE). Les sol·licituds de tipus d'arxius ASP.NET es controlen mitjançant una extensió ISAPI d'ASP.NET (aspnetisapi.dll), que s'executa en l'espai d'adreces del procés de IIS (inetinfo.exe). Utilitza una canalització amb nom per a comunicar-se amb el procés de treball d'ASP.NET.

Comunicació entre IIS i ASP.net

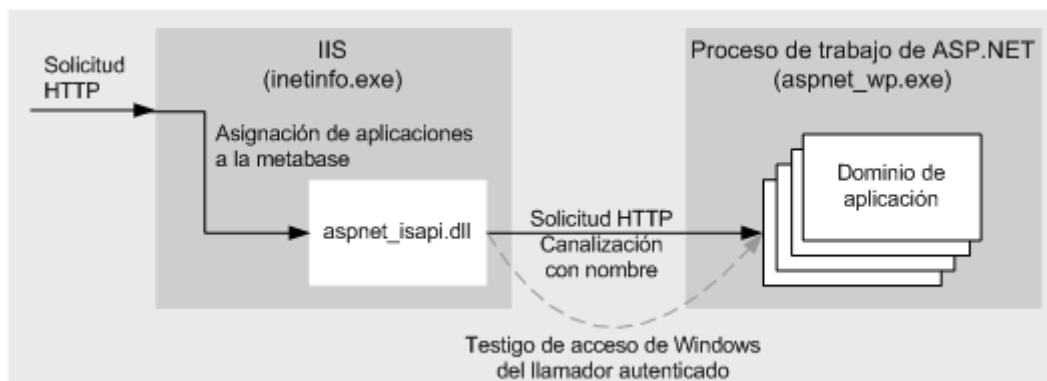


Figura 29

Aïllament d'aplicacions

Per a proporcionar aïllament s'utilitzen dominis d'aplicació independents en el procés de treball (un per cada directori virtual IIS o, en altres paraules, un per cada aplicació Web ASP.NET o servei Web). Es contraposa a la tecnologia ASP clàssica, on el nivell de protecció de l'aplicació configurat en la metabase IIS determinava si l'aplicació ASP havia d'executar-se en el procés amb IIS (inetinfo.exe), fora del procés en una instància dedicada de Dllhost.exe o en una instància compartida (agrupada) de Dllhost.exe

L'extensió ISAPI d'ASP.NET

L'extensió ISAPI d'ASP.NET (aspnetisapi.dll) s'executa en l'espai d'adreces del procés IIS (inetinfo.exe) i reenvia les sol·licituds de tipus d'arxiu ASP.NET al procés de treball d'ASP.NET a través d'una canalització amb nom. Els tipus d'arxiu d'ASP.NET s'assignen a l'extensió ISAPI d'ASP.NET mitjançant assignacions definides en la metabase IIS. Les assignacions de tipus d'arxiu d'ASP.NET estàndard (.aspx, .asmx, .rem, .soap) s'estableixen al instal·lar .NET Framework.

Anatomia d'una sol·licitud Web

La biblioteca ISAPI d'ASP.NET (Aspnetisapi.dll) s'executa en l'espai d'adreces del procés IIS (inetinfo.exe). Distribueix les sol·licituds a l'objecte HttpRuntime en el procés de treball d'ASP.NET (Aspnetwp.exe).

El següent conjunt d'accions es produeix com resposta a cada sol·licitud Web que rep ASP.NET: L'objecte HttpRuntime examina la sol·licitud i la reenvia a una instància d'un objecte HttpApplication. Hi ha almenys una instància de l'objecte HttpApplication per cada domini d'aplicació (els objectes s'agrupen) i un domini d'aplicació per cada directori virtual IIS.

La sol·licitud inicial d'un arxiu en un determinat directori virtual comporta la creació d'un domini d'aplicació i un nou objecte HttpApplication. Es llegeix una llista de mòduls HTTP de l'arxiu Machine.config (continguts en l'element). Es poden agregar a l'arxiu Web.config mòduls HTTP addicionals per a una aplicació concreta.

Control d'esdeveniments

L'objecte HttpApplication activa el conjunt d'esdeveniments que apareixen a la següent taula:

Event	Notes
BeginRequest	S'activa abans d'iniciar el processament de la sol·licitud
AuthenticateRequest	Per autenticar la crida
AuthorizeRequest	Per realitzar comprovacions d'accés
ResolveRequestCache	Per obtenir una resposta de la caché
AcquireRequestState	Per carregar el estat de la sessió
PreRequestHandlerExecute	S'activa immediatament abans d'enviar la sol·licitud al objecte de controlador
PostRequestHandlerExecute	S'activa immediatament després d'enviar la sol·licitud a l'objecte de controlador
ReleaseRequestState	Per emmagatzemar l'estat de la sessió
UpdateRequestCache	Per actualitzar la caché de resposta
EndRequest	S'activa al finalitzar el processament
PreSendRequestHeaders	S'activa abans d'enviar les capçaleres de resposta emmagatzemats en búfer
PreSendRequestContent	S'activa abans d'enviar el cos de resposta emmagatzemats en búfer

Cadascun dels mòduls HTTP pot enllaçar aquests esdeveniments (proporcionant els seus propis controls d' esdeveniments).

4.3 AJAX

Historia

Tot i que el terme "Ajax" fos creat en 2005, la història de les tecnologies que permeten Ajax es remunta a una dècada abans amb la iniciativa de Microsoft en el desenvolupament de Scripting Remot. No obstant això, les tècniques per a la càrrega asíncrona de continguts en una pàgina existent sense requerir recarrega completa remunten al temps de l'element iframe (introduït en Internet Explorer 3 en 1996) i el tipus d'element layer (introduït en Netscape 4 en 1997, abandonat durant les primeres etapes de desenvolupament de Mozilla).

Ambdós tipus d'element tenien l'atribut src que podia prendre qualsevol adreça URL externa, i carregant una pàgina que contingui javascript que manipuli la pàgina paterna, poden assolir-se efectes semblants al Ajax.

El Microsoft's Remote Scripting (o MSRS, introduït en 1998) va resultar un substitut més elegant per a aquestes tècniques, amb enviament de dades a través d'un applet Java el qual es pot comunicar amb el client usant Javascript.

Aquesta tècnica va funcionar en ambdós navegadors, Internet Explorer versió 4 i Netscape Navigator versió 4. Microsoft la va utilitzar en el Outlook Web Access proveït amb la versió 2000 de Microsoft Exchange Server.

La comunitat de desenvolupadors web, primer col·laborant per mitjà del grup de notícies microsoft.public.scripting.remote i després usant blogs, van desenvolupar una gamma de tècniques de scripting remot per a aconseguir els mateixos resultats en diferents navegadors. Els primers exemples inclouen la llibreria JSRS en l'any 2000, la introducció a la tècnica imatge/cookie en el mateix any i la tècnica Javascript sota demanda (Javascript on Demand) en 2002.

En aquest any, es va realitzar una modificació per part de la comunitat d'usuaris al Microsoft's Remote Scripting per reemplaçar el applet Java per XMLHttpRequest.

Des que XMLHttpRequest està implementat en la majoria dels navegadors, rarament s'utilitzen tècniques alternatives. No obstant això, encara s'utilitzen on es requereix una major compatibilitat, una reduïda implementació, o accés creuat entre llocs web. Una alternativa, el Terminal SVG (basat en SVG), empra una connexió persistent per a l'intercanvi continu entre el navegador i el servidor.

Descripció

AJAX, acrònim de Asynchronous Javascript And XML (Javascript i XML asíncrons), és una tècnica de desenvolupament web per a crear aplicacions interactives.

Aquestes s'executen en el client, és a dir, en el navegador de l'usuari, i manté comunicació asíncrona amb el servidor en segon pla. D'aquesta forma és possible realitzar canvis sobre la mateixa pàgina sense necessitat de recarregar-la. Això significa augmentar la interactivitat, velocitat i usabilitat en la mateixa. AJAX és una combinació de tres tecnologies ja existents: XHTML (o HTML) i fulles d'estils en cascada (CSS) per al disseny que acompanya a la informació.

1. **Document Object Model (DOM)** accedit amb un llenguatge de scripting per part de l'usuari, especialment implementacions ECMAScript com Javascript i JScript, per a mostrar i interactuar dinàmicament amb la informació presentada.

2. **L'objecte XMLHttpRequest** per a intercanviar dades asíncronament amb el servidor web. En alguns frameworks i en algunes situacions concretes, s'utilitza un objecte iframe en lloc del XMLHttpRequest per a realitzar aquests intercanvis.
3. **XML** és el format usat comunament per a la transferència de tornada al servidor, encara que qualsevol format pot funcionar, incloent HTML, text pla, JSON i fins a EBML. Com el DHTML, LAMP o SPA,

AJAX no constitueix una tecnologia en si, sinó que és un terme que engloba a un grup d'aquestes que treballen conjuntament.

Funcionament

El model operatiu de Ajax transforma la forma habitual de processar les dades en la web. Veiem les diferències com treballa una web tradicional i una web amb ajax.

Aplicació Web tradicional:

En una aplicació normal el navegador interacciona, en l'ordinador de l'usuari, amb aquest.

Quan s'inicia la sessió, es carrega una pàgina HTML enviada pel servidor a petició del navegador. Quan es prem un enllaç, s'omple un formulari, o es realitza una consulta, el navegador construeix una petició http que s'envia al servidor de la web en qüestió, localitzat arreu del món.

Amb aquesta informació el servidor realitza l'operació que s'ha demanat, potser consulta una base de dades, recupera un gràfic o realitza un càlcul, i construeix una nova pàgina usant possiblement CSS i HTML o simplement envia el contingut preexistent al navegador que, al seu torn la presenta, esborrant la pàgina anterior. En molts casos la nova pàgina és molt similar a la precedent, potser amb un gràfic més ampliat o una informació addicional. El problema d'aquest esquema, bé conegut per tots, és que mentre el servidor està atafegat processant la petició (entre moltes altres) nosaltres estem esperant pacientment que ens arribi la nova pàgina, el que no sempre ocorre ràpidament. Tot succeeix en el costat del servidor mentre que en el nostre costat el navegador només espera.

Diagrama aplicació Web tradicional

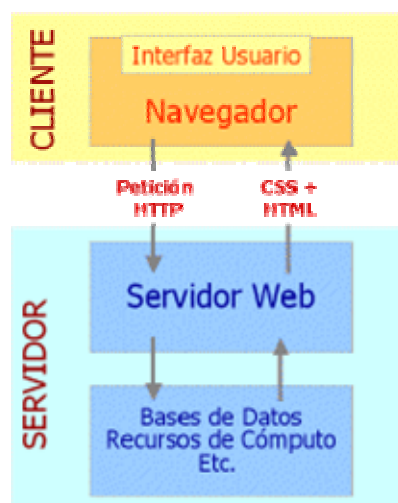


Figura 30

Aplicació Web amb AJAX:

Ajax introdueix un intermediari entre la interacció amb l'usuari i el servidor que es denomina "Motor Ajax". La pàgina que rep el navegador a l'inici de la sessió, en comptes de contenir una pàgina HTML habitual, el que conté és un motor Ajax enviat pel servidor, escrit en Javascript i habitualment camuflat en un marc (frame) invisible.

El motor Ajax el que fa és proporcionar un marc flexible i independent del servidor que realitza tota la interacció amb l'usuari i que pot contenir eines de càlcul, sistemes sofisticats de producció i interacció amb gràfics o qualsevol cosa que es pugui programar en Javascript. D'aquesta forma la interacció es deslliga del servidor (es fa asíncrona, d'aquí l'A de Ajax) i mentre interaccionem amb l'aplicació, el navegador es pot estar comunicant amb el servidor sense fer-nos esperar que acabi la transmissió de dades. Això és possible perquè qualsevol interacció es fa contra el motor Ajax i no contra el servidor.

Una vegada s'han carregat les dades es pot interaccionar amb ells mitjançant els programes Javascript descarregats. Tot allò que no estigui carregat i es necessiti del servidor es pot demanar usant XML, en segon pla, sense esborrar la pàgina ni parar la interacció de l'usuari.

Aplicació Web amb AJAX

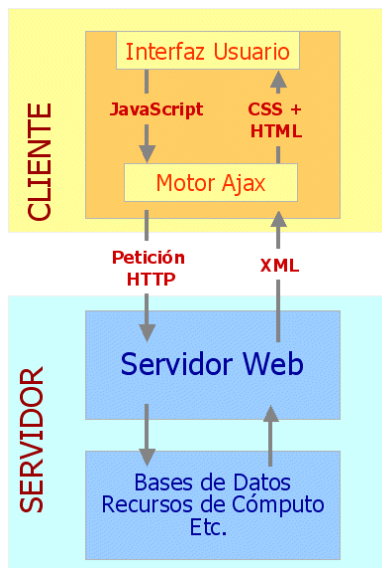


Figura 31

4.4 AJAX – Components utilitzats

Al projecte s'ha fet servir el toolkit d' AJAX. En concret els components AJAX que s'han utilitzat són:

Updatepanel:

Serveix a l' hora d' actualitzar, en comptes de realitzar un post de la pàgina complerta al servidor, actualitza només la part afectada. Els paquets que s'envien són més petits i millora la interactivitat amb l' usuari.

ModalPopUpExtender:

Mostra una pantalla en un pop up i no permet a l'usuari interactuar amb la pàgina fins que es tanqui.

CalendarExtender:

Serveix per introduir una data més fàcilment a una caixa de text.

HoverMenuExtender:

Serveix per associar una finestra emergent a un control de forma que quan l'usuari es situa amb el mouse per sobre es mostra la finestra.

AnimationExtender:

És un control especialment indicat per realitzar efectes d' animació.

DragPannelExtender:

Permet col·locar finestres d'una web en la posició que l'usuari desitgi.

5. Conclusions

El meu interès per la programació .NET és evident, ja que conec perfectament la tecnologia asp, però aprendre una nova tecnologia amb tants canvis era una tasca difícil.

Mai havia treballat en programació en tres capes ni en .net ni amb AJAX, és per això que s'ha fet un esforç personal molt gran i una dedicació quasi completa al projecte. Personalment estic molt content del resultat, ja no només de com ha quedat el producte final sinó pels coneixements adquirits i la possibilitat de treballar en un futur en aquestes tecnologies.

Amb el temps del que hem disposat s'ha realitzat un producte finalitzat, ja que considero que té les funcionalitats més importants d'un catàleg de productes, ara bé amb més temps es podrien haver desenvolupat més funcionalitats i es podrien haver utilitzat més components d'AJAX.

Durant tot el procés d'implementació m'he adonat del gran abast que té .NET, i de les grans possibilitats. Em sembla que podria fer tres projectes com el que s'ha realitzat i encara em quedarien coses per aprendre. Espero amb el temps anar perfeccionant tècniques de programació, estar a la última de les noves tecnologies que vagin sortint i així col·laborar perquè entre tots puguem fer webs més vistoses, més interactives i més usables.

5. Glossari

ActiveX: És una tecnologia de Microsoft per al desenvolupament de pàgines dinàmiques. Té presència en la programació del costat del servidor i del costat del client

ADO.NET: Realitza les operacions bàsiques de base de dades.

ASP.NET. Arquitectura integrada en el Framework .NET per a desenvolupar aplicacions web.

Cas d'ús. Acció que duu a terme un actor sobre un objecte, un actor no ha de ser necessàriament una persona.

CLR: Common Language Runtime (CLR) és el nucli de la plataforma .NET. És el motor encarregat de gestionar l'execució de les aplicacions a les que ofereix nombrosos serveis que simplifiquen el seu desenvolupament i afavoreixen la seva fiabilitat i seguretat.

Compilació just-in-time: És una tècnica per a millorar el rendiment de sistemes de programació que compilen a bytecode, consistent a traduir el bytecode a codi màquina natiu en temps d'execució.

CrossDatabase: Terme utilitzat quan un programa és executable per a diferents bases de dades.

Datareader: Objecte optimitzat per a desplaçar-se per registres de només lectura i només cap a endavant.

Dataset: És una representació resident en memòria de dades que proporciona un model de programació relacional coherent independentment de l'origen de dades.

Diagrama de Seqüència: És un dels diagrames més efectius per modelar interacció entre objectes a un sistema.

DOM: És una forma de representar els elements d'un document estructurat (tal com una pàgina web HTML o un document XML) com objectes que tenen els seus propis mètodes i propietats

Framework .NET. De l'anglès, Bastiment. Marc on corren les aplicacions independentment del sistema operatiu. Llançat per Microsoft com a resposta a la plataforma Java.

HTML. Llenguatge d'etiquetes per a crear pàgines web estàtiques.

HttpApplication: Defineix els mètodes, les propietats i els esdeveniments comuns a tots els objectes d'aplicació inclosos en una aplicació ASP.NET

HttpRunTime: Estableix els valors de configuració HTTP d'ASP.NET en temps d'execució que determinen com es processa una sol·licitud d'una aplicació ASP.NET

IDE: Integrated development environment, és a dir, un entorn integrat de desenvolupament.

IIS: Internet Information Services (o Server), IIS, és una sèrie de serveis per als ordinadors que funcionen amb Windows. Els serveis que ofereix són: FTP, SMTP, NNTP i HTTP/HTTPS.

MasterPages: Proporciona la habilitat de definir una estructura i uns elements d'interfície comuns per la Web, evita la duplicació innecessària de codi per estructures o comportaments de la Web que són compartits.

Membership: Configura els paràmetres per a l'administració i autenticació de comptes d'usuari utilitzant la subscripció ASP.NET.

SQLServer 2005: És una plataforma global de base de dades que ofereix administració de dades empresarials amb eines integrades d'intel·ligència empresarial.

Toolkit AJAX: Conté els components AJAX per tal de facilitar la programació d'aplicacions amb AJAX.

Visual Studio 2005: Programa desenvolupat per Microsoft que permet la construcció d'aplicacions Windows, web o dispositius mòbils.

WYSIWYG: És l'acrònim de What You See Is What You Get (en anglès, "el que veus és el que obtens"). S'aplica als processadors de text i altres editors de text amb format (com els editors d'HTML) que permeten escriure un document veient directament el resultat final.

XML. Format estàndard d'intercanvi de dades que permet la comunicació entre diferents sistemes.

Control d'usuari. Porció de codi que integra la vista i el controlador i que es pot reutilitzar fàcilment.

XMLHttpRequest: És una interfície emprada per a realitzar peticions HTTP i HTTPS a servidors WEB.

5. Bibliografía

Llibres:

- Optimización con ASP.NET 2.0. DotNetMania . Braulio Díez y Reyes García
- Programación con ASP.NET 2.0. ANAYA. Jesse Liberty, Dan Hurwitz.

Referències d'Internet:

- <http://www.asp.net/learn/dataaccess/tutorial01cs.aspx?tabid=63>
- <http://www.asp.net/learn/dataaccess/tutorial22cs.aspx?tabid=63>
- <http://msdn2.microsoft.com/en-us/library/ms972948.aspx>
- <http://www.123aspx.com/redir.aspx?res=34447>
- <http://www.dotnetbips.com/articles/c1e0ca90-5f5d-47aa-a739-492b562e810a.aspx>
- <http://www.gridviewguy.com/ArticleDetails.aspx?articleID=139>
- <http://msdn2.microsoft.com/en-us/library/ms998347.aspx>
- <http://forums.asp.net/p/1060418/1520778.aspx#1520778>
- <http://blogs.msdn.com/mattgi/archive/2007/01/23/asp-net-ajax-validators.aspx>
- <http://ajaxwidgets.com/AllControlsSamples/RequiredFieldValidator.aspx>
- <http://ajax.asp.net/Default.aspx>
- <http://www.codeplex.com/AtlasControlToolkit/Release/ProjectReleases.aspx?ReleaseId=1425>
- <http://www.forosdelweb.com/showthread.php?p=1974023>
- <http://www.herensoft.net/desktopdefault.aspx?tabid=643>
- <http://www.123aspx.com/redir.aspx?res=34447>
- <http://msdn.microsoft.com/msdnmag/issues/04/08/gridview/default.aspx>
- <http://www.codeproject.com/useritems/GridViewDetailsView.asp>
- <http://www.asp.net/learn/dataaccess/tutorial20cs.aspx>
- http://www.elguille.info/colabora/NET2006/gcuadra_GridViewPersonalizando_Partel.htm
- <http://ajax.asp.net/default.aspx?tabid=47>