

TESTS DE CONDUIR AMB .NET I AJAX AUTOTEST

Estudiant: Antonio Soria Hernández
Enginyeria Tècnica d'Informàtica de Sistemes
Consultor: David Gañán Jiménez

11 de Juny de 2007

RESUM DEL PROJECTE

Es tracta d'una aplicació Web que serveix perquè les persones puguin practicar i millorar els seus coneixements de cara a l'examen teòric de conduir. S'ha realitzat l'anàlisi, el disseny i la implementació utilitzant una arquitectura .NET amb 4 capes.

La finalitat d'aquest projecte és realitzar el desenvolupament d'una aplicació fent servir la **tecnologia .NET**, per tal d'ampliar els coneixement de programació Web mitjançant una tecnologia actualment en gran expansió, introduir-se i explorar les seves possibilitats.

S'ha fet servir **ASP.NET**, **Visual Basic.NET** i la seva connectivitat amb **SQL-Server 2005** mitjançant **ADO.NET**, concretament utilitzant **DataReaders** en front de **DataSets**, ja que els DataReaders donen a l'aplicació millors velocitats en execució. Per aconseguir una Web amb major interactivitat, velocitat i usabilitat s'ha utilitzat la tecnologia **AJAX**.

Al llarg d'aquesta memòria apareixeran detallades les fases d'anàlisi de requeriments, disseny i implementació d'aquesta aplicació. El mètode seguit, en el cicle de vida del TFC, és l'anomenat **Cascada**. Les fases que s'han proposat en aquesta evolució són: Inicialització, Anàlisi, Disseny i Implementació, que respectivament han donat lloc als documents Pla de treball, Anàlisi, Disseny i per últim el Producte final.

El "producte final" és una aplicació Web que permetrà als usuaris practicar els típics tests d'examen teòrics de conduir i també permetrà a l'administrador gestionar i fer el manteniment de la Web. També es fa una petita introducció a les tecnologies utilitzades com ASP.NET i AJAX.

Al capítol de conclusions es descriuen els resultats obtinguts, així com les dificultats trobades, i com durant tot el procés d'implementació s'ha aconseguit l'objectiu d'explorar i introduir-se en aquestes tecnologies.

Al capítol d'Annexos es detallen les possibles millores que hauria de tenir l'aplicació i possibles línies de continuació de l'aplicació.

PARAULES CLAU

Paraules Clau	Descripció
ADO.NET	<p>És la solució proposada per Microsoft per a permetre l'accés a bases de dades des de la seva plataforma .NET.</p>
Arquitectura.Net	<p>Plataforma independent del llenguatge pel desenvolupament de serveis Web.</p>
AJAX	<p>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.</p>
ASP.NET	<p>Conjunt de tecnologies de desenvolupament d'aplicacions web comercialitzat per Microsoft i usat per a construir llocs web domèstics, aplicacions web i serveis XML. Parteix de la plataforma .NET de Microsoft i és la tecnologia successora de la tecnologia Activi Server Pages (ASP).</p>
Mètode Cascada	<p>Mètode per gestionar el cicle de vida d'un Projecte.</p>
DataReaders	<p>És un objecte de ADO.NET per accedir a les dades, les característiques del qual es detallen a l'apartat 2.1.3 ADO.NET</p>

Paraules Clau	Descripció
DataSets	Objecte de ADO.NET per accedir a les dades, les característiques del qual detallaran a l'apartat 2.1.3 ADO.NET
Programació en capes	Estil de programació en el qual l'objectiu primordial és la separació de la lògica de negocis respecte la lògica de disseny.
SQL-Server 2005	Programari que dóna el servei i l'accés a la base de dades. Structured Query Language és un llenguatge estàndard de comunicació amb bases de dades. És un llenguatge normalitzat que permet treballar amb qualsevol tipus de llenguatge.
Tecnologia .NET	Marc de creació d'aplicacions que permet desenvolupar aplicacions Web, client/servidor, serveis Web.
Visual Basic.NET	Un dels llenguatges de programació orientat a objectes suportat per la plataforma .NET.

ÍNDEX DE CONTINGUTS

1. INTRODUCCIÓ	7
1.1 JUSTIFICACIÓ DEL TFC I CONTEXT EN EL QUAL ES DESENVOLUPA: PUNT DE PARTIDA I APORTACIÓ DEL TFC.....	7
1.2 OBJECTIUS DEL TFC.....	7
1.3 ENFOCAMENT I MÈTODE SEGUIT.....	8
1.4 PLANIFICACIÓ DEL PROJECTE.....	8
1.5 PRODUCTES OBTINGUTS.....	10
a) Pla de treball.....	10
b) Anàlisi.....	10
c) Disseny.....	10
d) Implementació.....	11
e) Memòria del projecte.....	12
f) Presentació del projecte.....	12
g) Manual d'instal·lació.....	12
1.6 BREU DESCRIPCIÓ DELS ALTRES CAPÍTOLS DE LA MEMÒRIA	12
2. RESTA DE CAPÍTOLS	14
2.1 RESUM DE LES TECNOLOGIES USADES	14
2.1.1 ASP.NET.....	14
2.1.2 Visual Basic.NET.....	15
2.1.3 ADO.NET.....	16
a) DataReaders:.....	16
b) Datareaders vs Datasets.....	16
2.1.4 CSS.....	17
a) Estils.....	17
b) Capes.....	17
2.1.5 AJAX.....	17
a) Controls ScriptManager i UpdatePanel.....	20
b) Imatges il·lustratives dels controls AJAX al TFC.....	20
2.2 FASE D'ANÀLISI.....	22
2.2.1 <i>Detall de les funcionalitats de l'aplicació</i>	22
a) Funcionalitats de l'administrador.....	22
b) Funcionalitats de l'usuari.....	23
2.2.2 <i>Diagrama de casos d'ús amb la seva especificació textual</i>	24
a) Especificació textual de l'administrador.....	25
b) Especificació textual de l'usuari.....	30
2.2.3 <i>Model conceptual de classes</i>	33
2.3 FASE DE DISSENY.....	33
2.3.1 <i>Concepte de la programació per capes</i>	33
2.3.2 <i>Arquitectura del TFC</i>	34
2.3.3 <i>Diagrames de seqüència</i>	35
2.3.4 <i>Diagrames de classes de disseny, disseny de les taules i diagrama de relacions entre taules</i>	39
2.3.5 <i>Disseny de la interfície gràfica</i>	42
a) Aplicació de gestió per l'administrador:.....	43
b) Aplicació Web.....	47
3. CONCLUSIONS	54
4. GLOSSARI.....	55
5. BIBLIOGRAFIA.....	58
6. ANNEXOS.....	59

ÍNDIX DE FIGURES

FIG 1: PLANIFICACIÓ I DIAGRAMA DE GANTT DEL 1R LLIURAMENT	9
FIG 2: CAPTURA DE LA PLANIFICACIÓ I DIAGRAMA DE GANTT DEL 2N LLIURAMENT.....	9
FIG 3: CAPTURA DE LA PLANIFICACIÓ I DIAGRAMA DE GANTT DEL 3R LLIURAMENT.....	10
FIG 4: MODEL DE FUNCIONAMENT AJAX 1	18
FIG 5: MODEL DE FUNCIONAMENT AJAX 2	19
FIG 6: SCRIPTMANAGER I UPDATEPANEL EN L'APLICACIÓ DE GESTIÓ	21
FIG 7: SCRIPTMANAGER I UPDATEPANEL EN L'APLICACIÓ WEB	21
FIG 8: CALENDAREXTENDER	22
FIG 9: DIAGRAMA DE CASOS D'ÚS.....	24
FIG 10: MODEL CONCEPTUAL DE CLASSES.....	33
FIG 11: DIAGRAMA DE SEQÜÈNCIA <i>REGISTRAR-SE</i>	35
FIG 12: DIAGRAMA DE SEQÜÈNCIA <i>LOGONEJAR-SE</i>	36
FIG 13: DIAGRAMA DE SEQÜÈNCIA <i>CONSULTAR I REALITZAR TEST</i>	37
FIG 14: DIAGRAMA DE SEQÜÈNCIA <i>CREAR FITXA</i>	38
FIG 15: DIAGRAMA DE SEQÜÈNCIA <i>VISUALITZAR FITXA</i>	38
FIG 16: DIAGRAMA DE CLASSES DE DISSENY	39
FIG 17: TAULA DE CATEGORIES.....	40
FIG 18: TAULA DE TESTS.....	40
FIG 19: TAULA DE PREGUNTES	40
FIG 20: TAULA DE RESPOSTES	40
FIG 21: TAULA D'USUARIS	41
FIG 22: TAULA DE FITXA	41
FIG 23: DIAGRAMA DE LES RELACIONS ENTRE TAULES	42
FIG 24: PÀGINA INICI DE L'APLICACIÓ DE GESTIÓ.....	43
FIG 25: PÀGINA DE MANTENIMENT DE CATEGORIES.....	44
FIG 26: PÀGINA DE MANTENIMENT DE TESTS.....	45
FIG 27: PÀGINA DE MANTENIMENT DE PREGUNTES.....	45
FIG 28: PÀGINA NOVA/MODIFICAR PREGUNTA	46
FIG 29: PÀGINA DE MANTENIMENT D'USUARIS.....	46
FIG 30: PÀGINA PER AFEGIR UN NOU USUARI.....	47
FIG 31: PÀGINA INICI DE L'APLICACIÓ WEB	48
FIG 32: IMATGE D'EXEMPLE D'IDENTIFICACIÓ INCORRECTA.....	48
FIG 33: PÀGINA PERQUÈ L'USUARI S'IDENTIFIQUI	49
FIG 34: PÀGINA DE TEST D'UN USUARI REGISTRAT	50
FIG 35: PÀGINA DE QÜESTIONARI	51
FIG 36: PART FINAL DE LA PÀGINA DEL QÜESTIONARI.....	52
FIG 37: PÀGINA DE REGISTRE DE L'USUARI	53

1. INTRODUCCIÓ

1.1 JUSTIFICACIÓ DEL TFC I CONTEXT EN EL QUAL ES DESENVOLUPA: PUNT DE PARTIDA I APORTACIÓ DEL TFC.

La finalitat és aprendre i millorar en la programació Web, i el TFC dóna la possibilitat d'assolir aquesta meta. S'ha escollit aquesta àrea pel gran interès en conèixer la tecnologia .NET i AJAX, perquè es tracta d'una tecnologia en una fase de gran expansió.

Per assolir aquest objectiu s'ha escollit una aplicació senzilla que permet realitzar i corregir tests d'exàmens teòrics de conduir i gestionar el manteniment de la Web, però el més destacat és que permet treballar amb ASP.NET, AJAX i amb DataReaders per la connexió amb bases de dades, per tant es treballa amb les últimes tecnologies.

S'ha optat per utilitzar la tecnologia AJAX perquè en aquest tipus d'aplicacions web resulta molt eficaç ja que permet un millor flux d'informació entre el client i el servidor, i redueix el temps d'espera i les transferències d'informació entre el client i el servidor.

Finalment s'ha decidit fer l'aplicació de Tests en comptes de la del catàleg de productes per la possibilitat que en un futur aquesta aplicació pugui ser totalment funcional en una autoescola real.

1.2 OBJECTIUS DEL TFC.

L'objectiu principal de la realització d'aquest projecte és l'aprenentatge de la tecnologia.NET, desenvolupant una aplicació que satisfaci els següents objectius:

- Que un usuari pugui entrenar-se i aprendre la tècnica dels tests per tal de poder presentar-se a l'examen de teoria del carnet de conduir.
- Que l'usuari pugui navegar de manera senzilla i intuïtiva per la web i pugui comprobar la seva evolució a través de la seva fitxa personal.
- Que l'administrador de la web pugui modificar els continguts i actualitzar els usuaris fàcilment.

D'altra banda, la consecució de l'objectiu principal comporta l'assoliment d'altres fites importants com són:

- Aprendre a realitzar una aplicació seguint totes les etapes del procés de desenvolupament: anàlisi, disseny, implementació i prova.
- Conèixer l'arquitectura .NET.
- Conèixer en profunditat el llenguatge Visual Basic .NET.
- Adquirir coneixements i experiència en un entorn de desenvolupament nou i cada vegada més present en el mercat laboral.
- Aprendre a utilitzar l'entorn de treball Visual Studio 2005.
- Desenvolupar components nous reutilitzables per altres aplicacions.
- Conèixer Microsoft SQL Server 2005 com a sistema de gestió de bases de dades.
- Aprendre el funcionament de la tecnologia AJAX, i algun dels seus controls.

1.3 ENFOCAMENT I MÈTODE SEGUIT

L'enfocament ha estat el cicle de vida del TFC, anomenat Cascada.

Les fases d'aquest cicle han estat: Inicialització, Anàlisi, Disseny i Implementació i aquestes fases han donat lloc als documents Pla de treball, Anàlisi, Disseny i el Producte final.

Evidentment no s'ha pogut complir a la perfecció ja que durant la fase d'implementació ha estat necessari fer un pas enrere i rectificar els documents Pla de treball, Anàlisi i Disseny, perquè s'han observat possibles millores respecte el plantejament inicial.

La clara diferenciació entre les funcionalitats dels dos possibles actors que interactuen amb l'aplicació ha permès utilitzar un mètode de treball en la fase d'implementació individualitzat en cadascun dels actors. Primer es van implementar les funcionalitats pròpies de l'administrador i finalment les dels usuaris. Es detallaran amb més profunditat a l'apartat *2.2.1 Detall de les funcionalitats de l'aplicació*.

1.4 PLANIFICACIÓ DEL PROJECTE

La planificació del projecte ha estat condicionada per les dates de lliurament de les diferents Pacs i el lliurament final.

La planificació no ha estat encertada durant la fase d'implementació, perquè ha estat necessari prolongar les tasques d'implementació de les polítiques de negoci i la implementació de la interfície gràfica. Per tant durant un temps la fase d'implementació ha conviscut amb la fase de la memòria i la presentació virtual.

Diagrames de Gantt:

1r LLIURAMENT: ESPECIFICACIÓ DE DISSENY

	Nombre de tarea	Duración	Comienzo	Fin	Predeceso
1	Repas de Ingeniería del programari	3 días	mié 14/03/07	dom 18/03/07	
2	Aprentatge Microsoft Visio	5 días?	lun 19/03/07	dom 25/03/07	1
3	Anàlisi i descripció de les classes, Pre i postcondició	5 días?	lun 26/03/07	dom 01/04/07	2
4	Realització dels diagrames UML i la interfície gràfica	6 días?	lun 02/04/07	lun 09/04/07	3

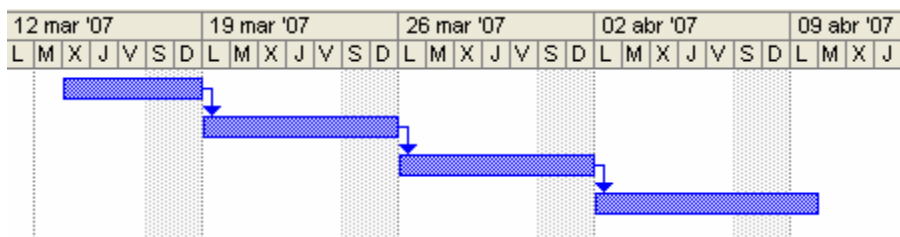


Fig 1: Planificació i Diagrama de Gantt del 1r lliurament

2n LLIURAMENT: IMPLEMENTACIÓ

	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	Repas de les tecnologies .NET	9 días?	mar 10/04/07	dom 22/04/07	
2	Creació de les classes i les taules de les BBDD	5 días?	lun 23/04/07	dom 29/04/07	1
3	Implementació de les polítiques de negoci	10 días?	lun 30/04/07	dom 13/05/07	2
4	Implementació de la interfície gràfica	9 días?	lun 14/05/07	jue 24/05/07	3
5	Comprovació final i depuració d'errors	2 días?	vie 25/05/07	lun 28/05/07	4

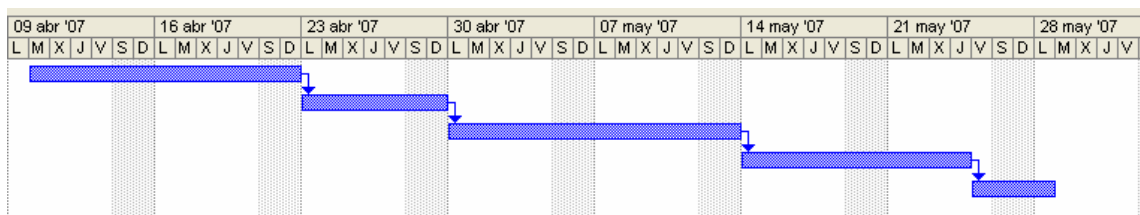






Fig 2: Captura de la Planificació i Diagrama de Gantt del 2n lliurament

3r LLIURAMENT: MEMORIA I PRESENTACIÓ VIRTUAL

		Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1		Redacció de la Memoria	4 días?	mar 29/05/07	dom 03/06/07	
2		Relització de la presentació virtual	5 días?	lun 04/06/07	dom 10/06/07	1
3		Comprovació final i depuració d'errors	1 día?	lun 11/06/07	lun 11/06/07	2

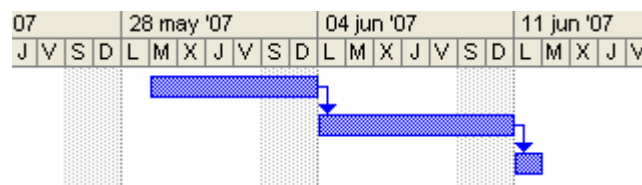


Fig 3: Captura de la Planificació i Diagrama de Gantt del 3r lliurament

1.5 PRODUCTES OBTINGUTS

A conseqüència del cicle de vida en cascada s'han obtingut els productes següents:

a) Pla de treball

Es la descripció inicial del projecte, on s'indiquen els objectius i es presenten els requeriments funcionals i tècnics necessaris.

Inclou la planificació temporal del projecte amb les dates de lliurament i la distribució del temps.

b) Anàlisi

El document d'anàlisi inclou una descripció detallada dels components del projecte i del seu funcionament.

Es mostren els requeriments funcionals de cadascuna de les aplicacions que componen el projecte mitjançant la descripció dels casos d'ús existents.

c) Disseny

El document de disseny conté una visió general de l'arquitectura de l'aplicació, el disseny de les classes principals del projecte, el disseny de la base de dades que s'utilitza i els diagrames de seqüència.

També es detallen les interfícies d'usuari i de l'administrador, amb una descripció de les pantalles corresponents a les captures de l'aplicació.

d) Implementació

El producte final és mortajanti_producte.zip, que contindrà els directoris corresponents de l'aplicació.

Els productes parcials obtinguts en la fase d'implementació són els següents:

- Els arxius .aspx trobem el codi que defineix els Web Forms representen la interface que veu l'usuari:
 - En el cas de l'administrador o aplicació de gestió:
 Default.aspx, Categories.aspx, Fitxa.aspx, Tests.aspx,
 Preguntes.aspx, NovaModificarPregunta.aspx,
 Usuaris.aspx, NouModificarUsuari.aspx,
 - En el cas de l'usuari o aplicació Web:
 Default.aspx, PreguntesUsuari.aspx, RegistreUsuari.aspx,
 TestAnonim.aspx, TestUsuari.aspx
- També s'ha fet us de les Master page, una per l'aplicació de gestió i l'altra per la web.
- Els arxius .aspx.vb on trobem el codi que implementa les funcionalitats relacionades amb el Web Form al qual està relacionat (code behind):
 - Aquest arxius van relacionats amb els Web Forms i per tant són els mateixos que els mencionats anteriorment, però amb l'extensió .aspx.vb. L'extensió "vb" ve determinada pel llenguatge de programació utilitzat, que en aquest cas és Visual Basic.
- Els arxius on es guarden totes les imatges que formen part de l'aplicació:
 - S'han guardat dins del directori "Imatges", que té dos subdirectoris: "Galeria", que és on es guarden les imatges de caràcter general i "Icons", on es guarden els Gifs que es necessiten pels botons i la resta de controls.
- Els arxius .vb, que corresponen a les capes de negoci i de dades. En el seu interior trobem totes les funcions i procediments que són cridats desde la capa de presentació, passen a la capa de negoci i van a parar a la capa de dades:
 - A la capa de negoci són:
 CategoriesCN.vb, FitxesCN.vb, PreguntesCN.vb, TestsCN.vb,
 UsuarisCN.vb
 - A la capa de dades són:
 CategoriesCD.vb, FitxesCD.vb, PreguntesCD.vb, TestsCD.vb,

UsuarisCD.vb, RespostesCD.vb i com a menció especial DatosComun.vb, que és on trobem la connexió amb la base de dades, la connectionString. Totes les classes de la capa de dades hereten de la classe DatosComun.vb.

- S'ha fet servir una altra capa, la capa comun, on es troben el directori control d'errades i el directori ModelObjects, que també fan servir arxius.vb:
 - Els ModelObjects: són les classes arrel de cada element del model, on trobem els atributs i les propietats de les classes, Categories.vb, Fitxes.vb, Preguntes.vb, Tests.vb, Usuaris.vb, Respostes.vb.
 - Control d'errades: trobem la classe DBExceptions.vb, que serveix per controlar totes les excepcions.

- L'arxiu Web.config, en el qual es guarda la configuració general de la aplicació web.

e) Memòria del projecte

És aquest document, mortajanti_memoria.pdf

f) Presentació del projecte

És la presentació multimèdia complementària a la memòria del projecte que fa èmfasi en els punts més importants i dóna una visió global del projecte.

El producte és un arxiu PowerPoint anomenat mortajanti_presentacio.ppt

g) Manual d'instal·lació

És la guia final on s'indica tot el necessari per poder instal·lar el projecte: quins requisits són necessaris, com instal·lar la base de dades del projecte.

El producte és un arxiu Word: Llegeix-me.doc

1.6 Breu descripció dels altres capítols de la memòria

S'enumeren els capítols que es detallen a l'apartat 2. *Resta de capítols*. Es realitza un resum dels elements més rellevats de la documentació que s'ha treballat. Es tracta de mostrar de forma genèrica allò que s'ha treballat a cada fase.

- A la *Fase d'anàlisi*:
 - L'anàlisi de requeriments detallat de les funcionalitats de l'aplicació.
 - El diagrama de casos amb la seva especificació, que serveixen per mostrar les funcions del sistema des del punt de vista de les seves interaccions amb l'exterior.
 - El model conceptual de classes.

- A la *Fase de disseny* es defineixen:
 - El concepte de programació per capes i les diferents capes que formen part de l'arquitectura de l'aplicació.
 - Els diagrames de seqüència.
 - El diagrama de classes de disseny, el disseny de les taules i el diagrama de relacions entre taules.
 - El disseny de la interfície gràfica.

Abans de detallar les fases d'anàlisi i disseny es farà un breu resum explicatiu de les tecnologies usades : ASP.NET, Visual Basic.NET, ADO.NET, CSS i AJAX.

2. RESTA DE CAPÍTOLS

2.1 Resum de les tecnologies usades

2.1.1 ASP.NET

És un conjunt de tecnologies de desenvolupament d'aplicacions web comercialitzat per Microsoft. És usat per programadors per a construir llocs web domèstics, aplicacions web i serveis XML. Forma part de la plataforma .NET de Microsoft i és la tecnologia successora de la tecnologia Activi Server Pages (ASP).

ASP.NET s'ha construït d'acord amb els següents principis:

- Facilitat de desenvolupament

ASP.NET introdueix un nou concepte, els "server controls", que permeten en forma d'etiquetes HTML tenir controls manejats pel servidor que identifiquen el navegador usat adaptant-se per a cada navegador. Tasques tedioses com la validació de dades es converteixen en fàcils i senzilles. Permet l'elecció del llenguatge de programació. Per defecte duu integrat C#, VB.NET i J#, però es pot usar un altre llenguatge. També permet independència de l'eina de desenvolupament. Es pot utilitzar des del Notepad, fins a la sofisticada i potent Visual Studio .NET, passant per la gratuïta Web Matriu. També és interessant la rica biblioteca de classes que duu incorporada, ja no és necessari obtenir components d'altres empreses per exemple per enviar un mail, fer "upload" d'un fitxer o generar gràfics en temps d'execució.

- Alt rendiment i escalabilitat

El codi és compilat per a ser executat en el CLR. Es pot optar per tenir-lo en el servidor precompilat o deixar que el servidor ho compili la primera vegada que ho executi. El resultat és de 3 a 5 vegades superior en velocitat que les antigues pàgines ASP. Presenta un ric sistema de *caché*. L'ús adequat del potent *caché* incorporat augmenta considerablement el rendiment i l'escalabilitat de l'aplicació. La *caché* permet *cachear* des de pàgines completes a parts completes, passant per conjunts de dades extretes de la base de dades. ASP.NET està preparat per a poder tenir granges de servidors web per a llocs amb alt volum de tràfic i repartir la càrrega entre diferents servidors.

- Millorada fiabilitat

ASP.NET és capaç de detectar pèrdues de memòria, problemes amb bloquejos i protecció davant de caigudes. Entre altres coses, és capaç de detectar aplicacions web que perden memòria, arrencant un altre procés .net amb una nova instància de l'aplicació per a tancar la qual perd memòria alliberant així la memòria perduda.

- Fàcil distribució i instal·lació

Una aplicació ASP.NET s'instal·la molt fàcilment, únicament copiant els fitxers que la componen a la web. No és necessari registrar cap component. Pots recompilar l'aplicació o enviar nous fitxers sense necessitat de reiniciar l'aplicació ni el servidor web.

2.1.2 Visual Basic.NET

És una versió de Visual Basic enfocada al desenvolupament d'aplicacions .NET. El llenguatge de programació és Visual Basic, que va aparèixer l'any 1991 com una evolució del QuickBasic que fabricava Microsoft. És un llenguatge de programació orientat a objectes (POO). Com a novetats més importants en la versió .NET es pot citar la possibilitat de definir àmbits de tipus, classes que poden derivar-se d'unes altres mitjançant herència, sobrecàrrega de mètodes, nou control estructurat d'excepcions o la creació d'aplicacions amb múltiples fils d'execució, a més de comptar amb l'extensa llibreria de .NET, amb la qual és possible desenvolupar tant Windows Applications i Web Forms, així com un extens nombre de clients per a bases de dades. Gràcies a aquestes millores en el que vindria sent Visual Basic 7.0 els programadors d'aquest llenguatge poden desenvolupar aplicacions més robustes que en el passat amb una base sòlida orientada a objectes. Actualment es troba vigent la versió 2005 (V.8), que treballa sota el .NET framework 2.0 i es troba en procés la versió 2007(V.9) que duu com nom clau "Orques". La última beta coneguda és la versió CTP de març de 2007, que treballa amb el .NET Framework 3.5.

Altres característiques importants són les següents:

- Disseny de controls d'usuari per a aplicacions Windows i Web.
- Programació de biblioteques de classe.
- Enviament de dades via documents XML.
- Generació de reports basats en Crystal Reports a partir d'informació obtinguda d'origens de dades (arxius de text, bases, etc.).

Es tracta de una àmplia gamma de característiques noves que permeten dissenyar aplicacions escalables en petites inversions de temps.

2.1.3 ADO.NET

La tecnologia ADO.NET és la solució proposada per Microsoft per a permetre l'accés a bases de dades des de la seva plataforma .NET. El que fan els objectes ADO (Objectes de Dades ActiveX) és permetre que una aplicació escrita en un llenguatge de programació qualsevol pugui comunicar-se amb una base de dades usant el SQL (Structured Query Language). Es podria dir que ADO.NET és un conjunt de classes que exposen serveis d'accés a dades al programador de .NET. Un aspecte important de ADO.NET és el fet que utilitza un model d'accés a dades pensat per a entorns desconnectats. Això vol dir que l'aplicació es connecta a l'origen de dades, realitza les operacions que necessiti, carrega en memòria els resultats obtinguts utilitzant variables i es desconnecta de l'origen de dades.

a) DataReaders:

Algunes característiques importants del DataReader són que no pot treballar de manera desconnectada, és un objecte només de lectura; la seva finalitat és la d'accedir el més ràpid possible a les dades i només permet moviment cap endavant a través dels registres, s'usen bucles per a recórrer el DataReader fins al final.

b) Datareaders vs Datasets

Si es necessita recuperar molts arxius ràpidament és convenient usar un DataReader. L'objecte DataReader és ràpid, va retornant dades d'única lectura del servidor, un registre alhora. A més, recuperant els resultats amb un DataReader es requereix significativament menys memòria que creant un DataSet. El DataReader no permet portar dades a l'atzar, ni permet actualitzar les dades. No obstant això, els proveïdors de les dades NET perfeccionen els DataReaders per treure quantitats grans de dades eficaçment.

A banda d'això, l'objecte DataSet serveix para emmagatzemar dades en forma desconnectada guardat en la memòria del client. En l'efecte, és una base de dades petita en sí mateix. Donat que el DataSet conté totes les dades que s'han recuperat, hi ha més opcions en la manera que es poden processar les dades. Es pot escollir a l'atzar amb les dades des del DataSet amb el update/insert/delete, es guarda a voluntat. també es poden manipular les dades correlatives com XML. Aquesta flexibilitat manté una funcionalitat impressionant per a qualsevol aplicació, però entra amb un cost alt en el consum de memòria.

2.1.4 CSS

Els fulls d'estil vénen a intentar tornar a separar en un document l'estil lògic de l'estil físic, deixant aquest últim en blocs de definició d'estils separats de l'estructura del document. CSS són les sigles de "Cascade StyleSheet". Es tracta d'una especificació sobre els estils físics aplicables a un document HTML, i tracta de donar la separació definitiva de la lògica (estructura) i el físic (presentació) del document.

a) Estils

L'estil lògic es refereix a la lògica del document: capçaleres, paràgrafs, etc. No es preocupa de l'aparença final, sinó de l'estructura del document. Per contra, l'estil físic no es preocupa de l'estructura del document, sinó per l'aparença final: paràgrafs amb un cert tipus de lletra, taules amb un determinat color de fons, entre d'altres.

La finalitat dels fulles d'estil és crear uns estils físics, separats de les etiquetes HTML (en lloc de com paràmetres de les etiquetes), i aplicar-los en els blocs de text en els quals es vulguin aplicar. Aquests estils podran ser modificats en algunes ocasions des de Javascript, i això comença a donar-nos una mica més d'interactivitat.

b) Capes

Per altra banda, tenim les capes, que vénen a donar-nos la solució al problema de posar elements just en la posició que volguem, evitant-nos haver de fer artificis per a obtenir el resultat buscat. Una capa serà una part més del document que pot ser situada eaqualsevol posició del mateix, aconseguint que es solapi sobre alguns elements si és el que necessitem, adequant els seus marges i altres propietats al que volguem realitzar.

2.1.5 AJAX

AJAX no és una tecnologia, són realment moltes tecnologies, cadascuna florint pel seu propi mèrit, unint-se en poderoses noves formes. AJAX incorpora:

- Presentació basada en estàndards usant XHTML i CSS;
- Exhibició i interacció dinàmiques usant el Document Object Model;
- Intercanvi i manipulació de dades usant XML and XSLT;
- Recuperació de dades asincrònica usant XMLHttpRequest;
- Javascript posant tot junt.

El model clàssic d'aplicacions Web funciona d'aquesta forma: la majoria de les accions de l'usuari en la interfície disparen un requeriment HTTP al servidor web. El servidor efectua un

procés (recopila informació, processa nombres, parlant amb diversos sistemes propietaris), i li retorna una pàgina HTML al client. Aquest és un model adaptat de l'ús original de la Web com un mitjà hipertextual, però ja se sap, el que fa a la Web bona per al hipertext, no la fa necessàriament bona per a les aplicacions de programari.

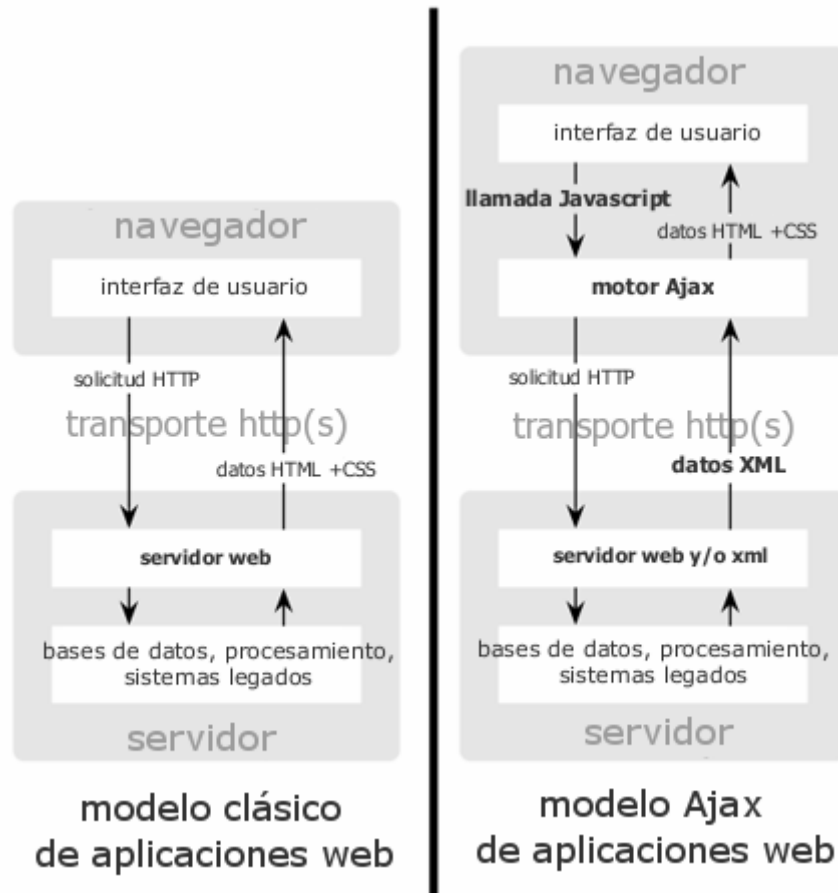
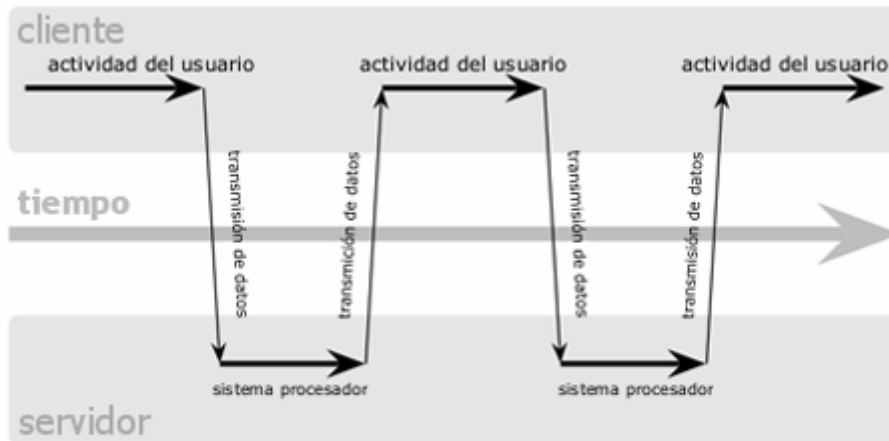


Fig 4: Model de funcionament AJAX 1

El model tradicional per a les aplicacions Web (part esquerra de la Fig 4) comparat amb el model de AJAX (part dreta de la Fig 4).

Una aplicació AJAX elimina la naturalesa d'arrencar –frenar – arrencar -frenar de la interacció en la Web introduint un intermediari -un motor AJAX- entre l'usuari i el servidor. Semblaria que sumar una capa a l'aplicació la faria menys reactiva, però és el contrari. En comptes de carregar un pàgina Web, a l'inici de la sessió el navegador carrega el motor AJAX (escrit en Javascript i usualment en un frame ocult). Aquest motor és el responsable per renderitzar la interfície que l'usuari veu i per comunicar-se amb el servidor en nom de l'usuari. El motor AJAX permet que la interacció de l'usuari amb l'aplicació succeeixi sincrònicament (independentment de la comunicació amb el servidor). Així l'usuari mai estarà mirant una finestra en blanc del navegador i una icona de rellotge de sorra esperant que el servidor faci una mica.

modelo clásico de aplicaciones web (síncrono)



modelo Ajax de aplicaciones web (asíncrono)

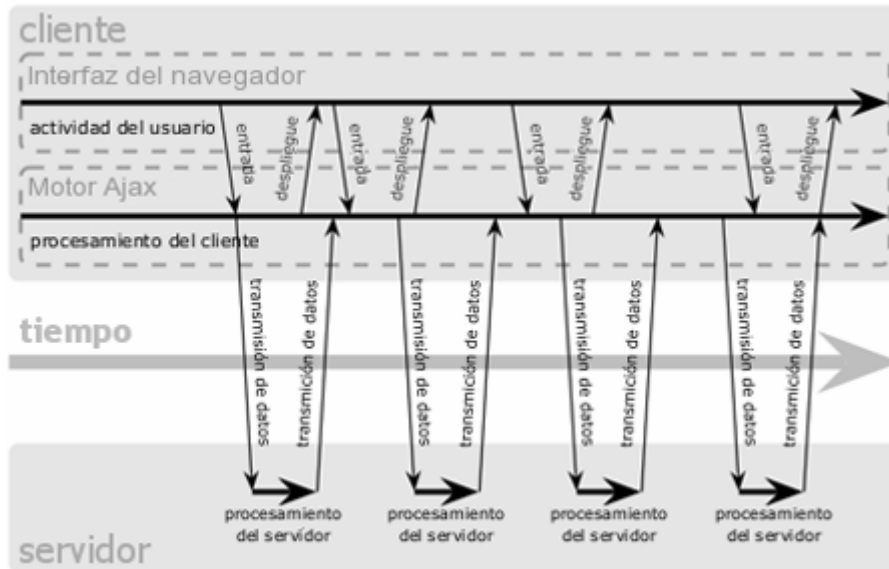


Fig 5: Model de funcionament AJAX 2

El patró d'interacció sincrònica d'una aplicació Web tradicional (part superior Fig 5) comparada amb el patró asincrònic d'una aplicació AJAX (part inferior Fig 5).

Cada acció d'un usuari que normalment generaria un requeriment HTTP pren la forma d'una crida Javascript al motor AJAX en comptes d'aquest requeriment. Qualsevol resposta a una acció de l'usuari que no requereixi una viatge de tornada al servidor (com una simple validació de dades, edició de dades en memòria, fins i tot una mica de navegació) és manejat pel seu compte. Si el motor necessita quelcom del servidor per a respondre (sigui enviant dades per a processar, carregant un codi addicional, o recuperant noves dades) fa aquestes comandes sincrònicament, usualment usant XML, sense frenar la interacció de l'usuari amb l'aplicació.

a) Controls ScriptManager i UpdatePanel

Els controls més importants d'AJAX són ScriptManager i UpdatePanel, els quals permeten la renderització parcial d'una pàgina, concretament tot el que es trobi dintre del UpdatePanel serà renderitzat, quan es produeixi un esdeveniment determinat.

L'ScriptManager és el control més important en un Web Site amb AJAX, ja que és el responsable de carregar les llibreries del client. És l'encarregat de controlar tots els components d'AJAX que afegim a la nostra pàgina.

Les seves propietats més importants són:

- EnablePartialRendering: permet la renderització parcial d'una pàgina. Per a habilitar les funcionalitats d'AJAX cal posar-lo a true.
- OnPageError: controla els errors que succeeixen a la pàgina.

L'UpdatePanel és el control que més directament ens proporciona la possibilitat de tenir una pàgina 100% AJAX. La filosofia consisteix en que tot el que hi hagi dintre del UpdatePanel "la ContentTemplate" produirà comportament AJAX.

Les seves propietats més importants són:

- Mode = Always, l'UpdatePanel s'actualitzarà sempre que es produeixi un PageLoad, per exemple a conseqüència d'una actualització d'altre UpdatePanel.
Mode = Conditional, l'UpdatePanel només s'actualitzarà quan els elements de dintre de l'UpdatePanel ho requereixin.
- Els Triggers ens serveixen per a definir quins esdeveniments o modificacions ha de caçar l'UpdatePanel per a modificar-se. Si el control que produeix l'esdeveniment que volem caçar no està dintre del ContentTemplate del UpdatePanel, pel que en teoria fer click sobre aquest control suposaria que la pàgina es recarregués totalment. No obstant això, amb la propietat Triggers del UpdatePanel, es pot indicar que desitgem caçar l'esdeveniment "Clic", per exemple del control "Button1" .

b) Imatges il.lustratives dels controls AJAX al TFC

Es pot veure com han estat molt utilitzats aquests controls. Per exemple a la part de l'aplicació de gestió:

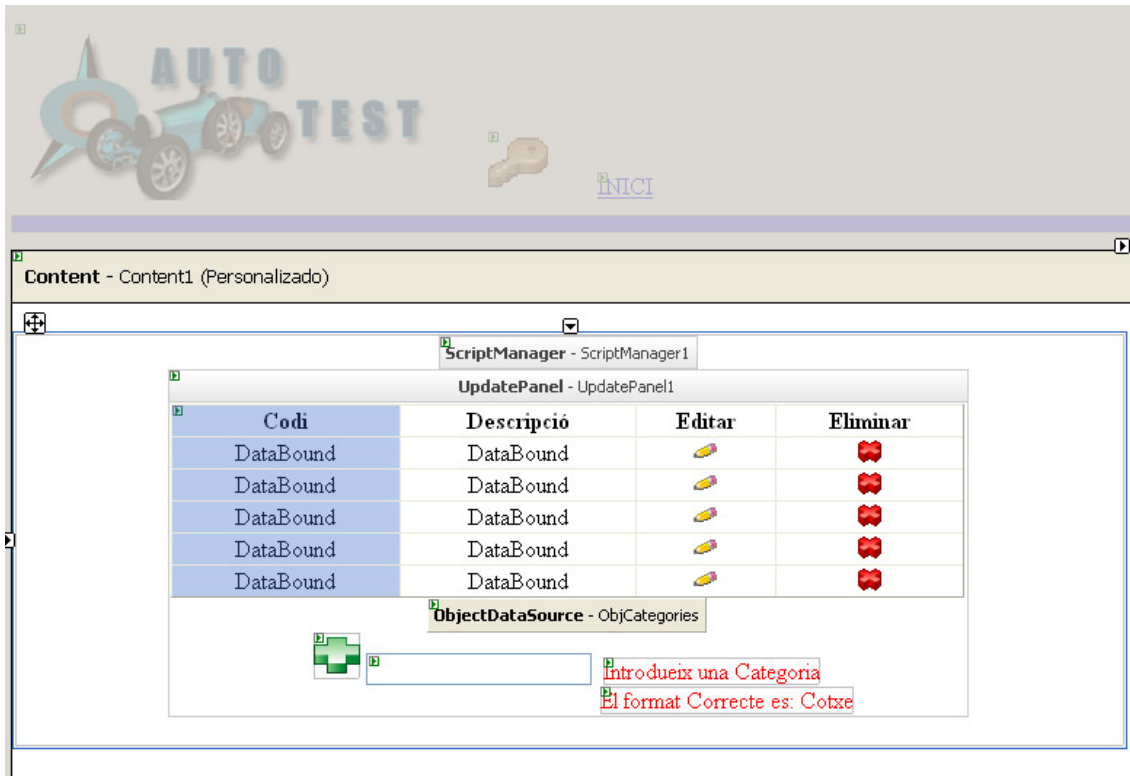


Fig 6: ScriptManager i UpdatePanel en l'aplicació de gestió

I, per exemple, en la part de l'aplicació Web:

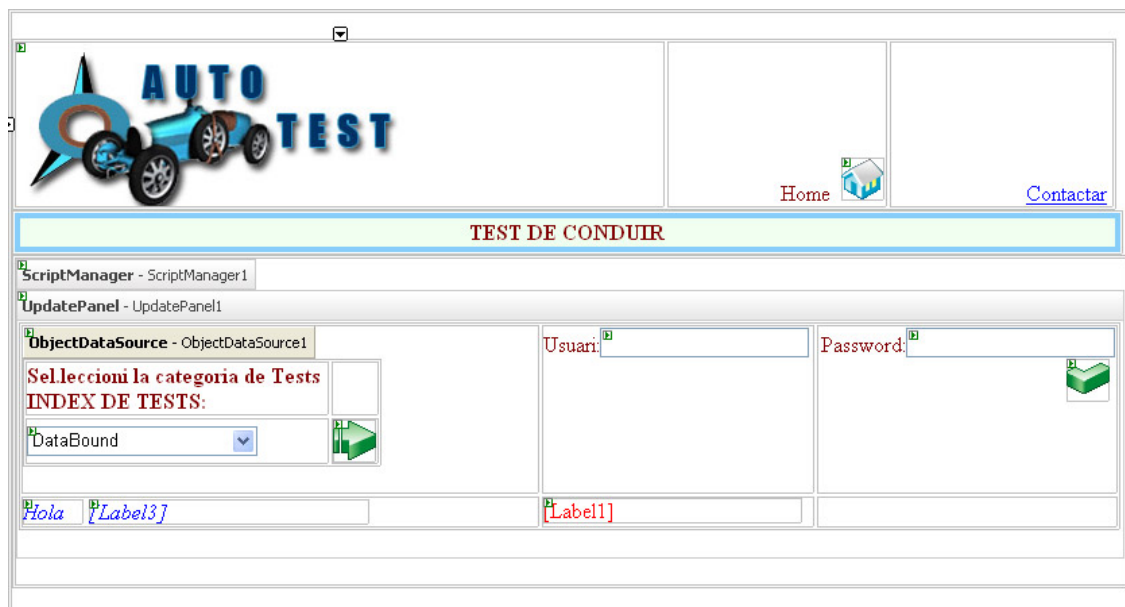


Fig 7: ScriptManager i UpdatePanel en l'aplicació Web

També s'ha utilitzat el control CalendarExtender de AJAX toolkit, que és un paquet de controls addicionals amb tecnologia AJAX). Permet associar un TargetControlID="Date1" calendari a un CssClass="ClassName" Format="MMMM d, yyyy" quadre de text PopupButtonID="Image1" />

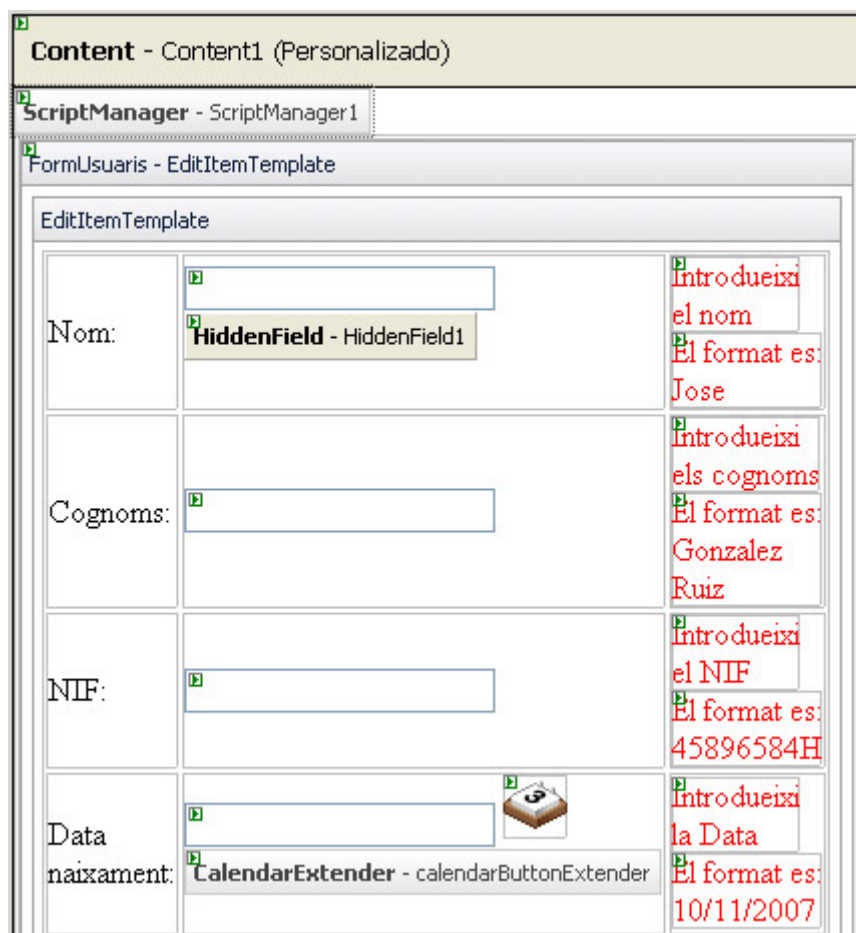


Fig 8: CalendarExtender

2.2 Fase d'anàlisi

2.2.1 Detall de les funcionalitats de l'aplicació

Les funcionalitats que compleix l'aplicació són les següents:

a) Funcionalitats de l'administrador

- Logonejar-se per gestionar i fer el manteniment de l'aplicació: l'aplicació estarà dins un fitxer protegit per tal que tan sols pugui accedir l'administrador amb el seu nom d'usuari i password.
- Introduir, modificar i eliminar usuaris registrats: l'aplicació mostrarà un Gridview amb tots els usuaris registrats, i podrà fer el manteniment del Gridview.

- Introduir, modificar i eliminar categories: l'aplicació mostrarà un Gridview amb totes les categories existents i podrà introduir-ne de noves, modificar-les o eliminar-les.
- Introduir, modificar i eliminar tests.
- Introduir, modificar i eliminar preguntes i respostes, i les imatges associades: l'aplicació mostrarà tot el llistat de preguntes. Podrà fer el manteniment de les preguntes i introduir-hi una imatge relacionada.
- Consultar la fitxa personal d'un usuari.

b) Funcionalitats de l'usuari

- Registrar-se a la web: permetrà a l'usuari tenir una fitxa personal que el sistema li generarà automàticament per tal de seguir la seva evolució. Si l'usuari no està registrat a la web podrà registrar-s'hi introduint les seves dades en un formulari.
- Logonejar-se: utilitzant el nom d'usuari i el password. D'aquesta manera el sistema sap que ha de guardar els resultats de l'usuari per tal d'actualitzar la seva fitxa.
- Consultar i realitzar els tests de la web: l'usuari podrà consultar tots els tests de la web lliurement, no és necessari que estigui registrat. Podrà realitzar els tests que més li interessin per categoria (cotxe, motocicleta, camió,...).
- Visualitzar fitxa: l'usuari, després de logonejar-se i seleccionar una categoria per fer un test, visualitzarà la seva fitxa i l'aplicació li presentarà una relació de tots els tests que ha realitzat amb les seves errades per test, marcarà els aprovats i els suspesos, la data de realització del test, i l'aplicació calcularà la mitjana d' errors i el % dels tests aprovats totals.

2.2.2 Diagrama de casos d'ús amb la seva especificació textual

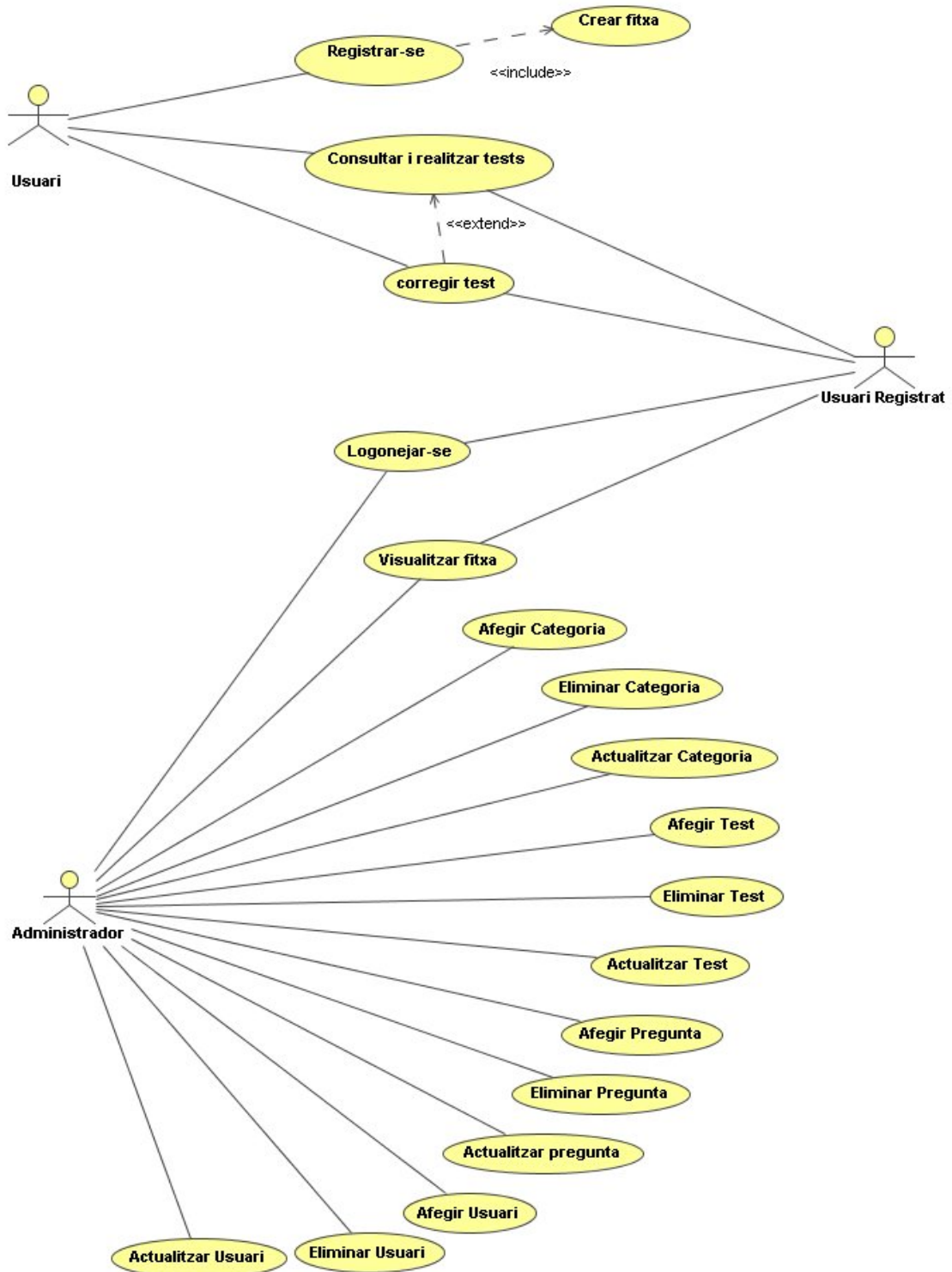


Fig 9: Diagrama de casos d'ús

a) Especificació textual de l'administrador

- Cas d'ús Logonejar-se a l'aplicació

- Resum de la funcionalitat: l'administrador introdueix usuari i contrasenya per identificar-se al sistema.
- Actors: Administrador.
- Casos d'ús relacionats: tots, ja que no pot accedir a l'aplicació de manteniment si no s'ha logonejat.
- Precondició: L'administrador és l'únic que té accés a l'aplicació.
- Postcondició: L'administrador s'ha identificat al sistema.
- Procés normal:
 1. L'administrador omple els camps d'usuari i password.
 2. El sistema valida l'administrador.

- Cas d'ús Afegir categoria

- Resum de la funcionalitat: l'administrador podrà afegir categories al sistema.
- Actors: Administrador.
- Casos d'ús relacionats: Logonejar-se a l'aplicació.
- Precondició: l'administrador ha d'estar identificat pel sistema.
- Postcondició: el sistema disposa d'una nova categoria.
- Procés normal:
 1. L'administrador accedeix a aplicació de gestió, concretament a la secció de categories.
 2. L'administrador afegeix una nova categoria.
 3. El sistema enregistra el canvi.

- Cas d'ús Eliminar categoria

- Resum de la funcionalitat: l'administrador podrà eliminar categories del sistema.
- Actors: Administrador.
- Casos d'ús relacionats: Logonejar-se a l'aplicació, Afegir categoria.
- Precondició: l'administrador ha d'estar identificat pel sistema i s'ha hagut d'afegir pel cap baix una categoria.
- Postcondició: El sistema disposa d'una categoria menys.
- Procés normal:
 1. L'administrador accedeix a aplicació de gestió, concretament a la secció de categories.

2. L'administrador elimina una categoria existent.
3. El sistema enregistra el canvi.

- Cas d'ús Actualitzar categoria

- Resum de la funcionalitat: l'administrador podrà Actualitzar categories del sistema.
- Actors: Administrador.
- Casos d'ús relacionats: Logonejar-se a l'aplicació, Afegir categoria.
- Precondició: l'administrador ha d'estar identificat pel sistema i s'ha hagut d'afegir pel cap baix una categoria.
- Postcondició: El sistema ha modificat una categoria.
- Procés normal:
 1. L'administrador accedeix a aplicació de gestió, concretament a la secció de categories.
 2. L'administrador selecciona una categoria existent.
 3. L'administrador modifica una categoria existent
 4. El sistema enregistra el canvi.

- Cas d'ús Afegir Test

- Resum de la funcionalitat: l'administrador podrà afegir tests al sistema.
- Actors: Administrador.
- Casos d'ús relacionats: Logonejar-se a l'aplicació, Afegir categoria
- Precondició: l'administrador ha d'estar identificat pel sistema i s'ha hagut d'afegir pel cap baix una categoria.
- Postcondició: el sistema disposa d'un nou test
- Procés normal:
 1. L'administrador accedeix a aplicació de gestió, concretament a la secció de test.
 2. L'administrador filtra per categoria.
 3. L'administrador afegeix un nou test.
 4. El sistema enregistra el canvi.

- Cas d'ús Eliminar Test

- Resum de la funcionalitat: l'administrador podrà eliminar tests del sistema.
- Actors: Administrador.
- Casos d'ús relacionats: Logonejar-se a l'aplicació, Afegir test.
- Precondició: l'administrador ha d'estar identificat pel sistema i s'ha hagut d'afegir pel cap baix un test.
- Postcondició: el sistema disposa d'un test menys.
- Procés normal:

1. L'administrador accedeix a aplicació de gestió, concretament a la secció de test.
2. L'administrador filtra per categoria.
3. L'administrador elimina un test existent.
4. El sistema enregistra el canvi.

- Cas d'ús Actualitzar Test

- Resum de la funcionalitat: l'administrador podrà Actualitzar tests del sistema.
- Actors: Administrador.
- Casos d'ús relacionats: Logonejar-se a l'aplicació, Afegir test.
- Precondició: l'administrador ha d'estar identificat pel sistema i s'ha hagut d'afegir pel cap baix un test.
- Postcondició: El sistema ha modificat un test.
- Procés normal:
 1. L'administrador accedeix a aplicació de gestió, concretament a la secció de tests.
 2. L'administrador filtra per categoria.
 3. L'administrador selecciona una test existent.
 4. L'administrador modifica una test existent.
 5. El sistema enregistra el canvi.

- Cas d'ús Afegir Pregunta

- Resum de la funcionalitat: l'administrador podrà afegir preguntes i les seves respostes relacionades a un test existent al sistema.
- Actors: Administrador.
- Casos d'ús relacionats: Logonejar-se a l'aplicació, Afegir Test.
- Precondició: l'administrador ha d'estar identificat pel sistema i s'ha hagut d'afegir pel cap baix un test.
- Postcondició: el sistema disposa d'una nova pregunta del test seleccionat.
- Procés normal:
 1. L'administrador accedeix a aplicació de gestió, concretament a la secció de preguntes.
 2. L'administrador filtra per categoria i per test i clica el botó d'afegir pregunta.
 3. El sistema mostra una nova pantalla amb tots els camps necessaris relacionats amb pregunta
 4. L'administrador afegeix una nova pregunta al test seleccionat.
 5. El sistema enregistra el canvi.

- Cas d'ús Eliminar Pregunta

- Resum de la funcionalitat: l'administrador podrà eliminar preguntes d'un test existent al sistema.
- Actors: Administrador.
- Casos d'ús relacionats: Logonejar-se a l'aplicació, Afegir pregunta.
- Precondició: l'administrador ha d'estar identificat pel sistema i s'ha hagut d'afegir pel cap baix una pregunta.
- Postcondició: El sistema disposa d'una pregunta menys del test seleccionat.
- Procés normal:
 1. L'administrador accedeix a aplicació de gestió, concretament a la secció de preguntes.
 2. L'administrador filtra per categoria i per test.
 3. L'administrador elimina una pregunta i les respostes relacionades, existents del test seleccionat.
 4. El sistema enregistra el canvi.

- Cas d'ús Actualitzar Pregunta

- Resum de la funcionalitat: l'administrador podrà actualitzar preguntes d'un test del sistema.
- Actors: Administrador.
- Casos d'ús relacionats: Logonejar-se a l'aplicació, Afegir test.
- Precondició: l'administrador ha d'estar identificat pel sistema i s'ha hagut d'afegir pel cap baix una pregunta.
- Postcondició: el sistema ha modificat una pregunta del test seleccionat.
- Procés normal:
 1. L'administrador accedeix a aplicació de gestió, concretament a la secció de preguntes.
 2. L'administrador filtra per categoria i per test i clica el botó d'editar pregunta.
 3. El sistema mostra una nova pantalla amb tots els camps necessaris omplerts, relacionats amb la pregunta escollida.
 4. L'administrador modifica algun camp relacionat amb la pregunta del test seleccionat.
 5. El sistema enregistra el canvi.

- Cas d'ús Afegir Usuari

- Resum de la funcionalitat: l'administrador podrà afegir usuaris del sistema.
- Actors: Administrador.

- Casos d'ús relacionats: Logonejar-se a l'aplicació,
- Precondició: l'administrador ha d'estar identificat pel sistema.
- Postcondició: el sistema disposa d'un nou usuari registrat.
- Procés normal:
 1. L'administrador accedeix a aplicació de gestió, concretament a la secció d'usuaris.
 2. L'administrador clica el botó d'afegir usuari.
 3. El sistema mostra una nova pantalla amb tots els camps necessaris relacionats amb usuari.
 4. L'administrador afegeix un nou usuari.
 5. El sistema enregistra el canvi.

- Cas d'ús Eliminar Usuari
 - Resum de la funcionalitat: l'administrador podrà eliminar usuaris del sistema.
 - Actores: Administrador.
 - Casos d'ús relacionats: Logonejar-se a l'aplicació.
 - Precondició: l'administrador ha d'estar identificat pel sistema i s'ha hagut d'afegir pel cap baix un usuari per part de l'administrador o s'ha enregistrat un nou usuari mitjançant la web.
 - Postcondició: el sistema disposa d'un usuari registrat menys.
 - Procés normal:
 1. L'administrador accedeix a aplicació de gestió, concretament a la secció d'usuaris.
 2. L'administrador clica el botó d'eliminar usuari.
 3. L'administrador elimina un usuari.
 4. El sistema enregistra el canvi.

- Cas d'ús Actualitzar Usuari
 - Resum de la funcionalitat: l'administrador podrà Actualitzar Usuaris del sistema.
 - Actores: Administrador.
 - Casos d'ús relacionats: Logonejar-se a l'aplicació.
 - Precondició: l'administrador ha d'estar identificat pel sistema i s'ha hagut d'afegir pel cap baix un usuari per part de l'administrador o s'ha enregistrat un nou usuari mitjançant la web.
 - Postcondició: el sistema ha modificat un usuari registrat.
 - Procés normal:
 1. L'administrador accedeix a aplicació de gestió, concretament a la secció d'usuaris.
 2. L'administrador clica el botó editar usuari.

3. El sistema mostra una nova pantalla amb tots els camps relacionats amb l'usuari seleccionat omplerts.
4. L'administrador modifica algun d'aquests camps relacionats amb l'usuari.
5. El sistema enregistra el canvi.

b) Especificació textual de l'usuari

- Cas d'ús Registrar-se

- Resum de la funcionalitat: l'usuari introdueix les seves dades personals per a poder registrar-se a la web i el sistema automàticament crea la seva fitxa personal.
- Actors: Usuari.
- Casos d'ús relacionats: Crear fitxa
- 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.
 3. El sistema executa el cas d'ús Crear fitxa.

- Cas d'ús Crear fitxa

- Resum de la funcionalitat: l'usuari es registra i el sistema crea una fitxa personal on anirà guardant tots els resultats dels tests que l'usuari vagi fent.
- Actors: Usuari.
- Casos d'ús relacionats: Registrar-se.
- Precondició: Registrar-se.
- Postcondició: -
- Procés normal:
 1. El sistema indica a l'usuari l'obligació de registrar-se i li mostra el formulari per a registrar-se.
 2. L'usuari es registra.
 3. El sistema mostra la fitxa a l'usuari.

- Cas d'ús Consultar i realitzar tests

- Resum de la funcionalitat: l'usuari navega i prova de realitzar test per la web.
- Actors: Usuari o Usuari Registrat.
- Casos d'ús relacionats: Corregir Test
- Precondició: -

- Postcondició: -
- Procés normal:
 1. L'usuari navega per la web.
 2. El sistema mostra els tests disponibles.
 3. L'usuari realitza un test.
 4. El sistema permet fer la correcció, mostrant on són els errors i finalment diu el nº d'errors total i si el test està aprovat.

- Cas d'ús Corregir tests
 - Resum de la funcionalitat: L'usuari demana al sistema que corregeixi el test que acaba de realitzar i el sistema li mostra les errades i el encerts, finalment el sistema mostra si el test està aprovat o suspès.
 - Actors: Usuari o Usuari Registrat.
 - Casos d'ús relacionats: Consultar i realitzar tests.
 - Precondició: Haver realitzat un test
 - Postcondició: -
 - Procés normal:
 1. L'usuari realitza un test i li demana al sistema que el corregeixi.
 2. El sistema el corregeix i mostra les errades i encerts en cada pregunta.
 3. Si l'usuari està registrat, el sistema guarda el nº d'errades de l'usuari en aquest test.
 4. El sistema mostra el nº d'errors total i si el test està aprovat.

- Cas d'ús Logonejar-se
 - Resum de la funcionalitat: L'Usuari Registrat introdueix usuari i contrasenya per identificar-se al sistema.
 - Actors: Usuari Registrat.
 - Casos d'ús relacionats: Visualitzar fitxa
 - Precondició: L'usuari ha d'estar registrat.
 - Postcondició: L'usuari s'ha identificat al sistema.
 - Procés normal:
 1. L'usuari registrat omple els camps d'usuari i contrasenya.
 2. El sistema valida l'usuari i s'inicia una nova sessió.
 3. El sistema crea la seva fitxa personal.

- Cas d'ús Visualitzar fitxa

- Resum de la funcionalitat: Opció 1) El sistema mostra la fitxa personal a l'usuari registrat. Opció 2) El sistema mostra la fitxa personal de l'usuari registrat seleccionat a l'administrador.
- Actors: Usuari Registrat o Administrador.
- Casos d'ús relacionats: Logonejar-se.
- Precondició: Opció 1) Logonejar-se. Opció 2) L'administrador ha d'estar identificat pel sistema i s'ha hagut d'afegir pel cap baix un usuari per part de l'administrador.
- Postcondició: -
- Procés normal:
 1. Opció 1) L'usuari registrat s'identifica (Logonejar-se).
Opció 2) L'administrador s'identifica (Logonejar-se).
 2. Opció 1) L'usuari registrat demana anar a la pagina on hi ha la seva fitxa.
Opció 2) L'administrador clicka al botó "veure fitxa" de l'usuari registrat seleccionat.
 3. El sistema li mostra la seva fitxa.

2.2.3 Model conceptual de classes

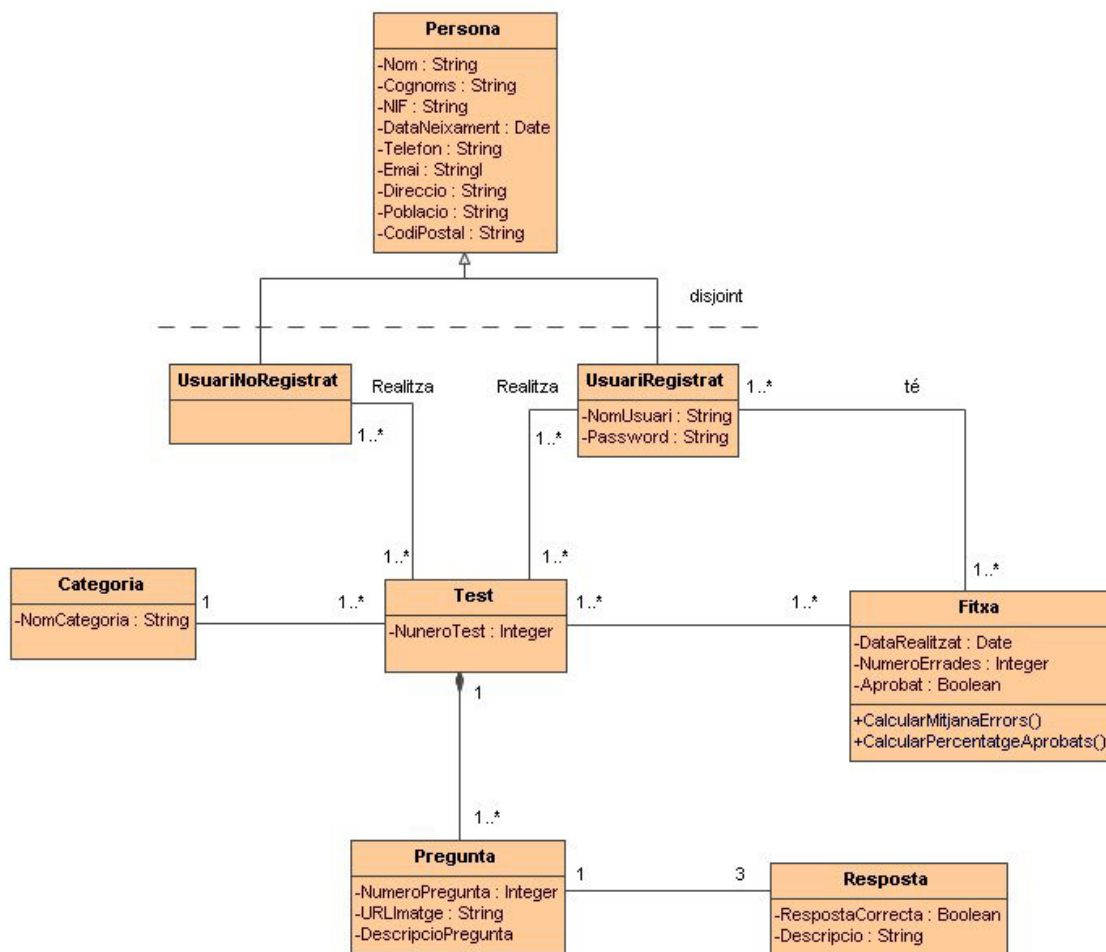


Fig 10: Model conceptual de classes

2.3 Fase de disseny

2.3.1 Concepte de la programació per capes

La programació per capes és un estil de programació en la qual l'objectiu fonamental és la separació de la lògica de negocis de la lògica de disseny. Un exemple bàsic d'això és separar la capa de dades de la capa de presentació a l'usuari.

L'avantatge principal d'aquest estil és que el desenvolupament es pot dur a terme en diversos nivells i en cas d'algun canvi només s'ataca el nivell requerit sense haver de revisar entre codi barrejat. Un bon exemple d'aquest mètode de programació seriosa és el Model d'interconnexió de sistemes oberts. A més a més, permet distribuir el treball de creació

d'una aplicació per nivells. D'aquesta manera, cada grup de treball està totalment abstret de la resta de nivells, simplement és necessari conèixer la API que existeix entre nivells.

En el disseny de sistemes informàtics actual se sol usar les arquitectures multinivell o Programació per capes. En aquestes arquitectures a cada nivell se li confia una missió simple, la qual cosa permet el disseny d'arquitectures escalables (que poden ampliar-se amb facilitat en cas que les necessitats augmentin). El disseny més en voga actualment és el disseny en tres nivells (o en tres capes).

1. Capa de presentació: és aquella que veu l'usuari (hi ha qui la denomina "capa d'usuari"). Presenta el sistema a l'usuari, li comunica la informació i captura la informació de l'usuari donant un mínim de procés (realitza un filtrat previ per a comprovar que no hi ha errors de format). Aquesta capa es comunica únicament amb la capa de negoci.
2. Capa de negoci: és on resideixen els programes que s'executen, rebent les peticions de l'usuari i enviant les respostes després del procés. Es denomina capa de negoci (i fins i tot de lògica del negoci) perquè és aquí on s'estableixen totes les regles que han de complir-se. Aquesta capa es comunica amb la capa de presentació, per a rebre les sol·licituds i presentar els resultats, i amb la capa de dades, per a sol·licitar al gestor de base de dades per a emmagatzemar o recuperar dades d'ell.
3. Capa de dades: és on resideixen les dades. Està formada per un o més gestor de bases de dades que realitza tot l'emmagatzematge de dades, rebent sol·licituds d'emmagatzematge o recuperació d'informació des de la capa de negoci. Totes aquestes capes poden residir en un únic ordinador (no seria el normal), si bé el més usual és que hi hagi una multitud d'ordinadors on resideix la capa de presentació (són els clients de l'arquitectura client/servidor). Les capes de negoci i de dades poden residir en el mateix ordinador, i si el creixement de les necessitats ho aconsella es poden separar en dos o més ordinadors. Així, si la grandària o complexitat de la base de dades augmenta, es pot separar en diversos ordinadors, els quals rebran les peticions de l'ordinador on resideixi la capa de negoci.

2.3.2 Arquitectura del TFC

S'ha utilitzat un model de quatre capes: capa de presentació, de dades, de negoci i una capa comuna per a totes elles.

1. Capa de dades : amb un model de base de dades de SQLServer hi ha totes les dades en taules com Categories, Tests, Preguntes, etc. En aquesta capa es realitzen

tots els accessos i totes les lectures i escriptures a la base de dades. S'han fet servir els datareader per la seva velocitat.

2. Capa negoci: en aquesta se situen tots els components necessaris que assegurin les funcions i procediments a realitzar pel sistema, com verificacions de dades i lògica de programa.
3. Capa Comun: en aquesta es troben totes les classes comunes que formen part de diverses capes, com són el control d'errades i els ModelObjects (són les classes arrel de cada element del model on trobem els atributs i les propietats de les classes, Categories.vb, Fitxes.vb, Preguntes.vb, Tests.vb, Usuaris.vb, Respostes.vb).
4. Capa de presentació: és la interfície gràfica per l'usuari i per l'administrador. En aquesta capa es defineix la presentació de la pàgina principal, la pàgina on es carreguen les preguntes dels tests, la fitxa, la zona de registre, les pàgines de l'aplicació de gestió per l'administrador, etc. S'utilitzen fulls d'estil CSS, masterpages i controls d'AJAX. D'aquesta manera el disseny i la presentació dels serveis i la informació resten separats de les dades i les funcionalitats de l'aplicació.

2.3.3 Diagrames de seqüència

A continuació s'adjunten alguns diagrames de seqüència de casos d'ús per part dels usuaris.

REGISTRAR-SE

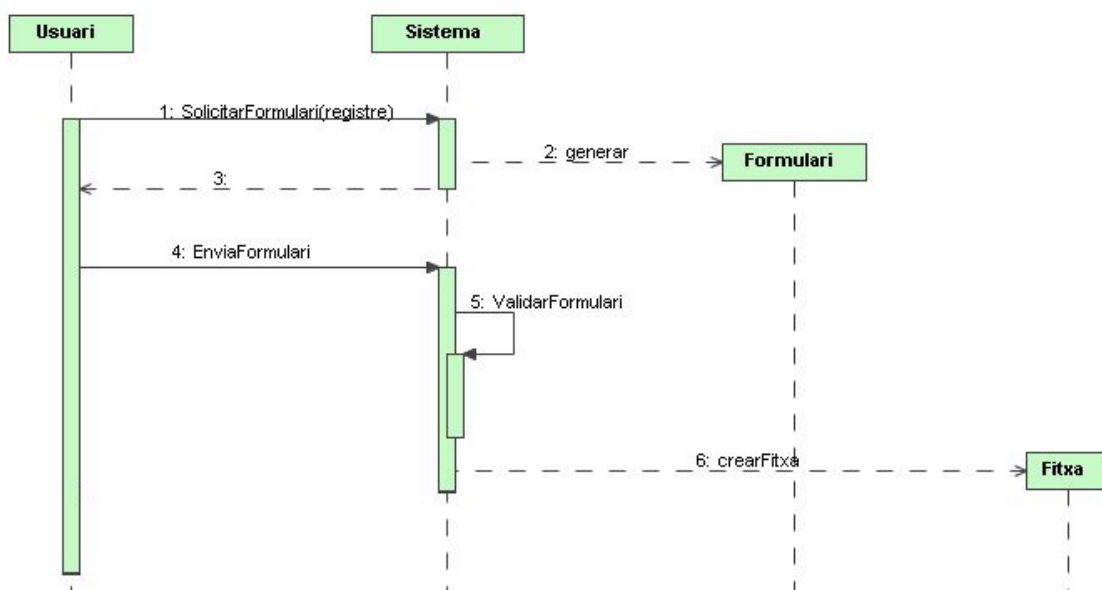


Fig 11: Diagrama de seqüència *Registrar-se*

LOGONEJAR-SE

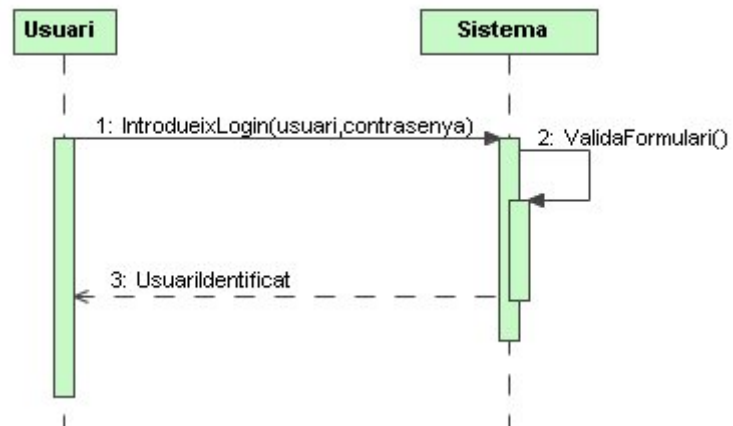


Fig 12: Diagrama de seqüència *Logonejar-se*

CONSULTAR I REALITZAR TEST

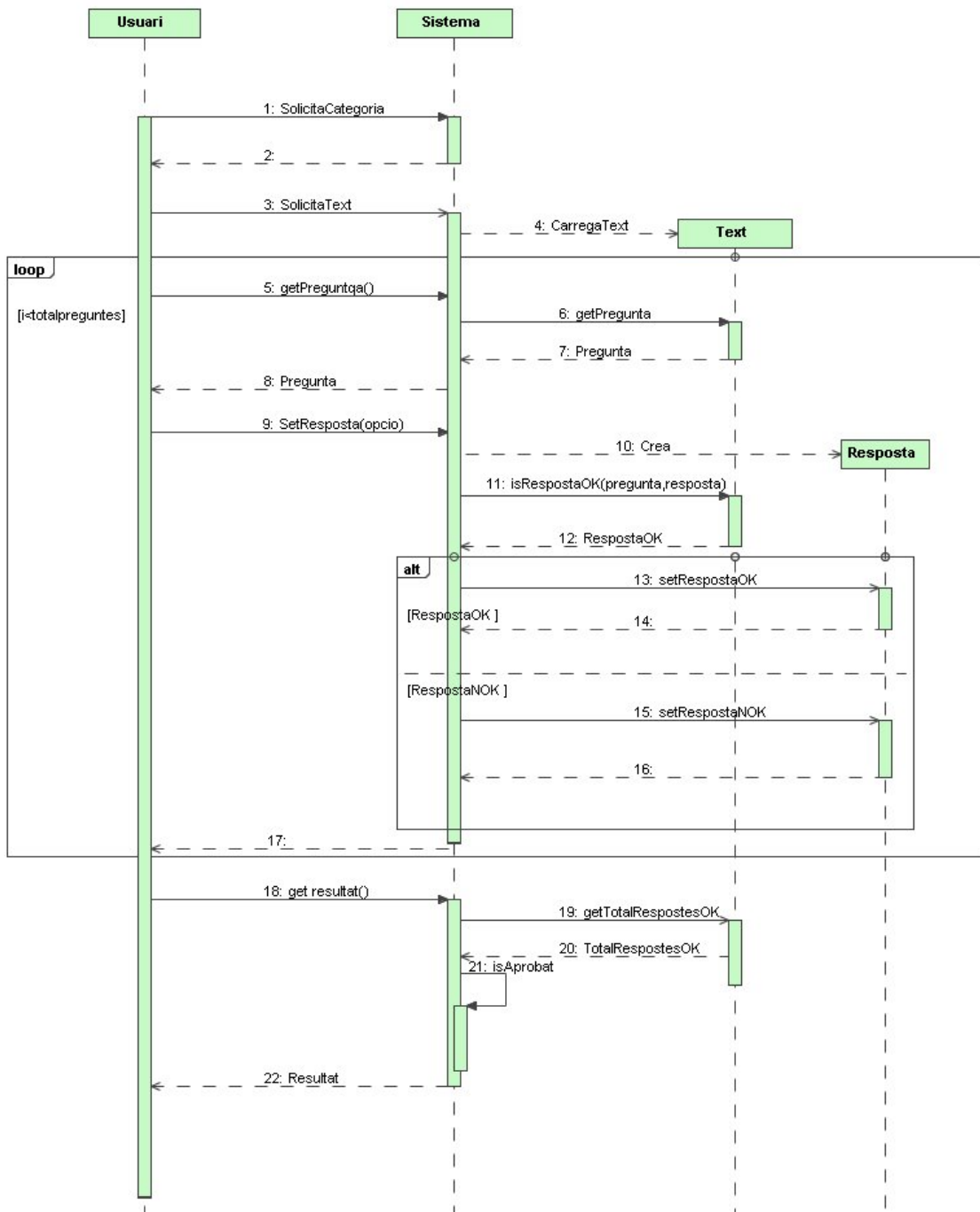


Fig 13: Diagrama de seqüència Consultar i realitzar test

CREAR FITXA

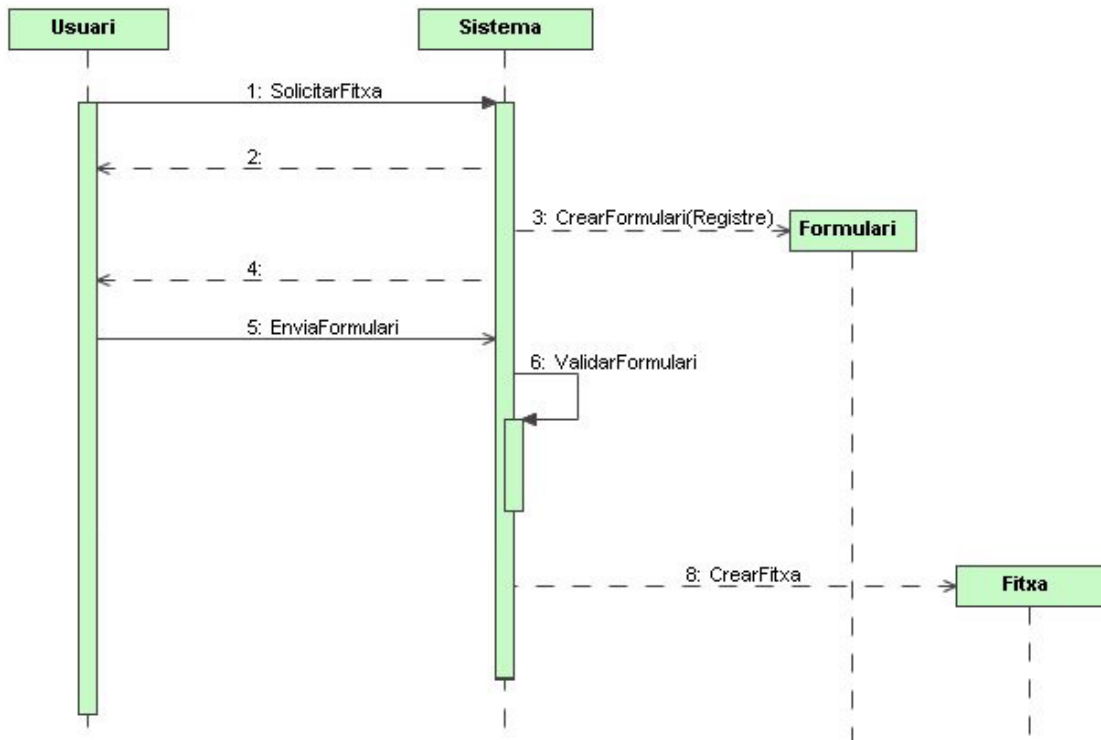


Fig 14: Diagrama de seqüència *Crear fitxa*

VISUALITZAR FITXA

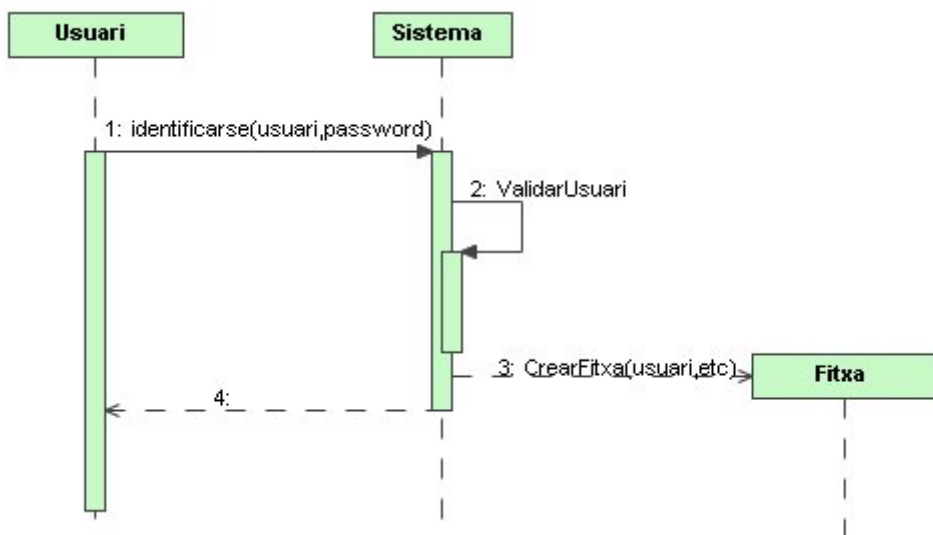


Fig 15: Diagrama de seqüència *Visualitzar fitxa*

2.3.4 Diagrames de classes de disseny, disseny de les taules i diagrama de relacions entre taules

S'ha optat per eliminar l'herència entre Persona, UsuariRegistrat i UsuariNoRegistrat, s'ha agrupat tots els atributs en la classe Usuari, i no s'ha canviat res més ja que no hi ha herència múltiple ni classes associatives i relacions ternàries.

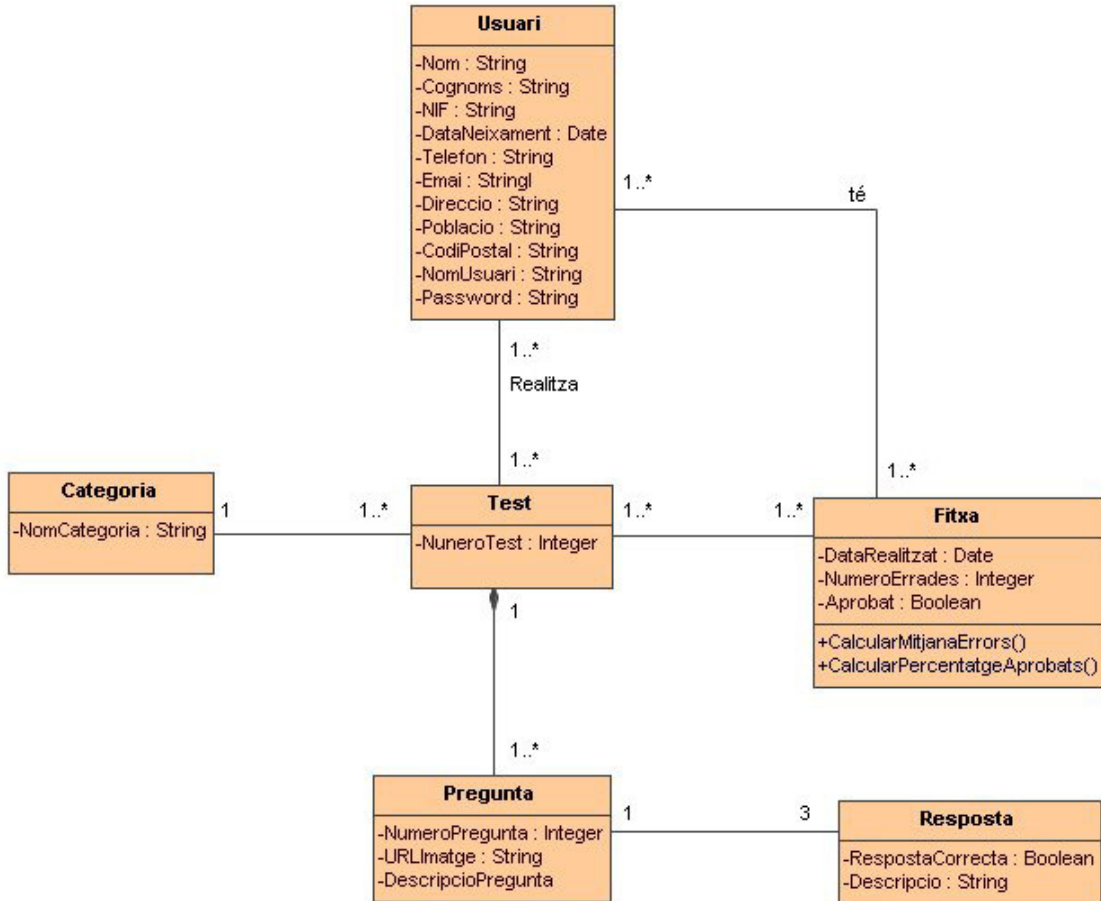


Fig 16: Diagrama de classes de disseny

Disseny de les taules

Es mostra com han estat dissenyades les diferents taules:

S'ha optat per afegir el camp "id" a totes les taules per facilitar les relacions entre les taules.

Taula de Categories:


ANNA_TONI-EEE...bo.Categories		ANNA_TONI-EEE... dbo.Respostes	
Column Name	Data Type	Allow Nulls	
 Id	int	<input type="checkbox"/>	
Nom_Categoria	nvarchar(50)	<input type="checkbox"/>	
		<input type="checkbox"/>	

Fig 17: Taula de categories

Taula de Tests:


ANNA_TONI-EEE...BD - dbo.Tests		ANNA_TONI-EEE... dbo.Respostes	
Column Name	Data Type	Allow Nulls	
 Id	int	<input type="checkbox"/>	
IdCategoria	int	<input checked="" type="checkbox"/>	
Numero_Test	nvarchar(50)	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Fig 18: Taula de tests

Taula de Preguntes:


ANNA_TONI-EEE... dbo.Preguntes		ANNA_TONI-EEE3...D - dbo.Fitxes	
Column Name	Data Type	Allow Nulls	
 Id	int	<input type="checkbox"/>	
Id_Test	int	<input checked="" type="checkbox"/>	
Numero_Pregunta	int	<input checked="" type="checkbox"/>	
Descripcio_Pregunta	nvarchar(500)	<input checked="" type="checkbox"/>	
Resposta1	int	<input checked="" type="checkbox"/>	
Resposta2	int	<input checked="" type="checkbox"/>	
Resposta3	int	<input checked="" type="checkbox"/>	
URL_Fotografia	nvarchar(50)	<input checked="" type="checkbox"/>	
		<input type="checkbox"/>	

Fig 19: Taula de preguntes

Taula de Respostes:


ANNA_TONI-EEE... dbo.Respostes		ANNA_TONI-EEE... dbo.Preguntes	
Column Name	Data Type	Allow Nulls	
 Id	int	<input type="checkbox"/>	
Descripcio_Resposta	nvarchar(500)	<input checked="" type="checkbox"/>	
Resposta_Correcta	bit	<input type="checkbox"/>	
		<input type="checkbox"/>	

Fig 20: Taula de respostes

Taula d'Usuaris:

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Nom	nvarchar(50)	<input checked="" type="checkbox"/>
Cognoms	nvarchar(50)	<input checked="" type="checkbox"/>
Nif	nvarchar(50)	<input checked="" type="checkbox"/>
Data_Naixament	datetime	<input checked="" type="checkbox"/>
Direccio	nvarchar(50)	<input checked="" type="checkbox"/>
Poblacio	nvarchar(50)	<input checked="" type="checkbox"/>
Codi_Postal	nvarchar(50)	<input checked="" type="checkbox"/>
Email	nvarchar(50)	<input checked="" type="checkbox"/>
Telefon	nvarchar(50)	<input checked="" type="checkbox"/>
Login	nvarchar(50)	<input checked="" type="checkbox"/>
Password	nvarchar(50)	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Fig 21: Taula d'usuaris

Taula de Fitxa:

Column Name	Data Type	Allow Nulls
Id	int	<input type="checkbox"/>
Usuari	int	<input checked="" type="checkbox"/>
Test	int	<input checked="" type="checkbox"/>
Data_Realitzat	datetime	<input checked="" type="checkbox"/>
Numero_Errades	int	<input checked="" type="checkbox"/>
Aprovat	bit	<input checked="" type="checkbox"/>
		<input type="checkbox"/>

Fig 22: Taula de fitxa

Diagrama de les relacions entre les taules

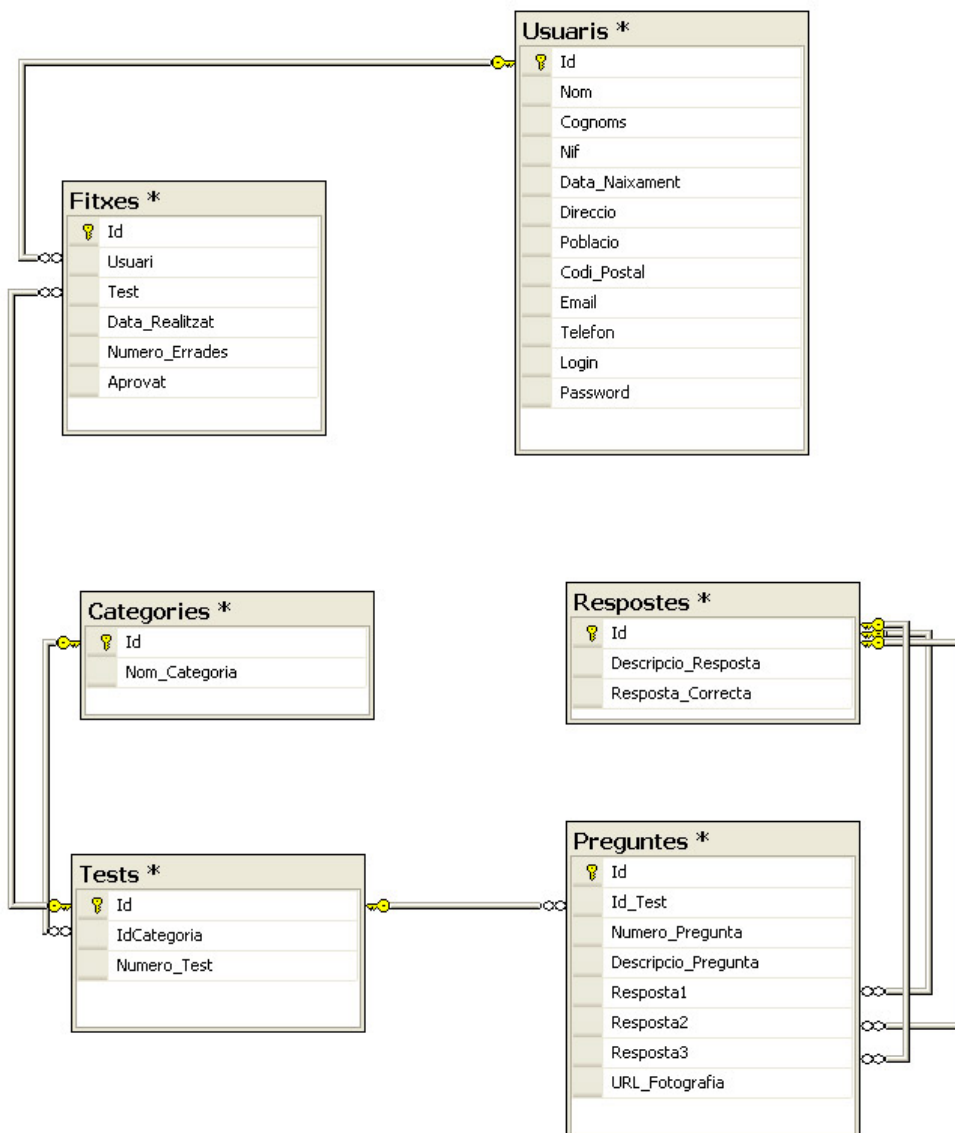


Fig 23: Diagrama de les relacions entre taules

2.3.5 Disseny de la interfície gràfica

Aquesta aplicació es pot dividir en dues semi-aplicacions, l'aplicació de gestió per part de l'administrador i l'aplicació Web que és la que l'usuari farà servir.

Es mostren per separat les interfícies gràfiques de les dues semi-aplicacions amb els comentaris pertinents.

S'ha procurat una interfície el més intuïtiva i senzilla possible, utilitzant els ImageButton amb imatges intuïtives que donen agilitat a la web i ajuden en l'aspecte estètic.

S'han fet servir els controls d'AJAX com UpdatePanel, ScriptManager i CalendarExtender. Aquests controls han donat a l'aplicació un aire de velocitat que ha fet que l'aplicació sigui molt dinàmica, ja que qualsevol modificació en la base de dades que estigui sota la influència de l'UpdatePanel és pràcticament immediata, perquè no es torna a carregar la pàgina, evitant les petites estones de pàgines en blanc (en procés de càrrega de dades).

També s'han utilitzat els ValidateControls com el RequiredFieldValidator i RegularExpressionValidator, que ajuden l'usuari a introduir correctament les dades i eviten problemes d'introducció errònia de dades, obligant a l'usuari a introduir les dades en un format definit.

Ha estat necessari descarregar una dll (Validators.dll, Validators.pdb), per tal que aquests controls funcionessin amb la influència dels controls d'AJAX. Això dóna una idea sobre el novedós que és la tecnologia AJAX.

a) Aplicació de gestió per l'administrador:

1. Pàgina inici de l'aplicació de gestió. L'administrador pot escollir d'entrada entre fer el manteniment de Categories, Test, Preguntes o Usuaris. S'ha utilitzat una masterpage per poder veure en tot moment el logotip de la web i el Link "INICI" per tornar a aquesta pàgina inicial.



Fig 24: Pàgina inici de l'aplicació de gestió

2. Aquesta és la pàgina que veurà l'administrador si selecciona fer el manteniment de categories. S'ha utilitzat un Gridview que està lligat a un ObjectDataSource, i tots aquests controls estan sota la influència del corresponent UpdatePanel i l'Scriptmanager.



Fig 25: Pàgina de manteniment de categories

3. Aquesta és la pàgina que veurà l'administrador si selecciona fer el manteniment de Tests. Cal remarcar la importància de l'utilització en els controls que mostren i permeten modificar les dades a l'usuari (Gridviews, Formviews i DataList..) de les TemplateFields que permeten separar els modes d'edició i inserció, encara que hi ha més Templates possibles.

El control DropDownList permet filtrar el Gridview per categoria.



Fig 26: Pàgina de manteniment de Tests

4. Aquesta és la pagina que veurà l'administrador si selecciona fer el manteniment de Preguntes. El Grid mostra les preguntes ordenades per el número de pregunta, prèviament l'administrador ha filtrat per categoria i per test.


Filtrar per Categoria: Motocicleta

Filtrar per Tests: 001

Codi	Descripció	Test	Pregunta	Editar	Eliminar
82	Motocicleta	001	1		
83	Motocicleta	001	2		
84	Motocicleta	001	3		
85	Motocicleta	001	4		
86	Motocicleta	001	5		
87	Motocicleta	001	6		
88	Motocicleta	001	7		
89	Motocicleta	001	8		
90	Motocicleta	001	9		
91	Motocicleta	001	10		
92	Motocicleta	001	11		
93	Motocicleta	001	12		
94	Motocicleta	001	13		
95	Motocicleta	001	14		
96	Motocicleta	001	15		
97	Motocicleta	001	16		
98	Motocicleta	001	17		
99	Motocicleta	001	18		
100	Motocicleta	001	19		
101	Motocicleta	001	20		

Fig 27: Pàgina de manteniment de preguntes

5. Aquesta és la pàgina que veurà l'administrador si vol afegir una nova pregunta. El control utilitzat en aquesta ocasió per interactuar amb les dades és un FormView que es troba a la InsertTemplate. S'han utilitzat els RadioButtons amb el mateix nom a la propietat GroupName, per tal que treballin conjuntament. També s'ha fet servir el control FileUpload per pujar imatges relacionades amb les preguntes.



[INICI](#)

Numero de Pregunta:

Categoria:

Test:

Descripció de Pregunta:

Descripció Resposta A:

Descripció Resposta B:

Descripció Resposta C:

URL

Fotografia:

Fig 28: Pàgina nova/modificar pregunta

6. Aquesta és la pàgina que veurà l'administrador si selecciona fer el manteniment d'Usuaris. Es mostra un Grid amb la novetat d'una nova columna (Fitxa), que permet veure la fitxa personal de l'usuari seleccionat.



[INICI](#)

Codi	Nom	Cognoms	Usuari	Fitxa	Editar	Eliminar
1	Juan	morales gonzalez	juangon			
3	Antonio	Gonzalez Aro	Angon			
5	perico	de los palotes	admin			

Fes click al botó afegir si vols introduir un usuari.

Fig 29: Pàgina de manteniment d'usuaris

7. Aquesta és la pàgina que veurà l'administrador si vol afegir un nou usuari. Es poden veure en acció els ValidateControls i el control CalendarExtender d'AJAX.



Fig 30: Pàgina per afegir un nou usuari

b) Aplicació Web

1. Pàgina inici de l'aplicació Web. L'usuari pot escollir entre fer login i accedir amb la seva sessió o accedir directament al test, seleccionant prèviament la categoria al DropDownList. S'ha utilitzat una masterpage per poder veure en tot moment el logotip de la web i l'ImageButton "Home" per tornar a aquesta pàgina inicial. També hi ha la possibilitat de contactar amb l'administrador, el Link "contactar" obre el correu predeterminat.

Aquesta pàgina inicial s'ha intentat no sobrecarregar-la, i per tant s'han escollit dues imatges aclaridores i uns breus textos de benvinguda i ànims.

Hi ha dos ImageButtons més, un que redirecciona a l'usuari a la pàgina de Tests i l'altre cap a la pàgina de registre.



Telèfon: 932548551 E-mail autotest@test.com

Fig 31: Pàgina inici de l'aplicació web

- Aquí es pot veure com l'aplicació detecta una identificació incorrecta d'un usuari. Tots aquests controls de la MasterPage estan sota la influència d'un UpdatePanel.

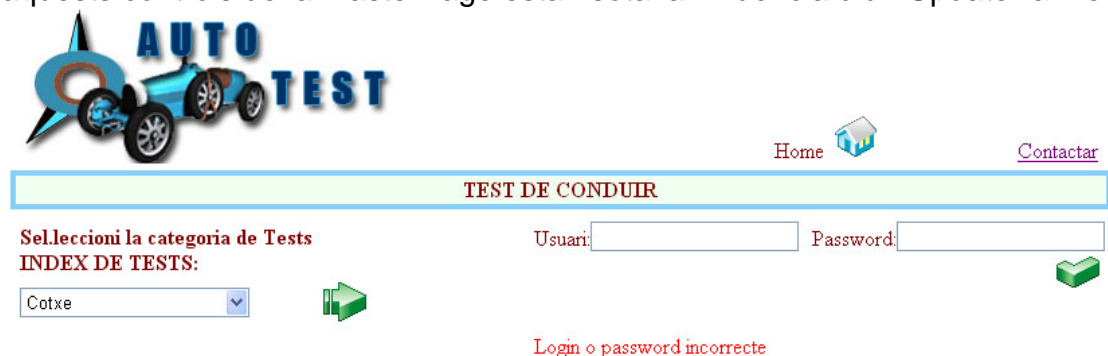
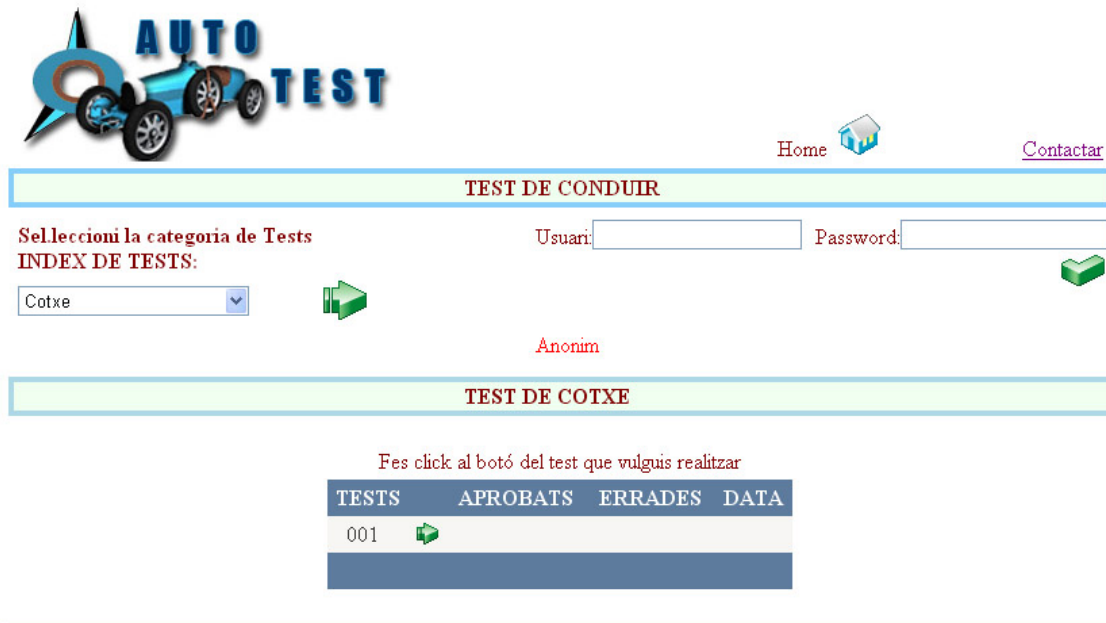



Fig 32: Imatge d'exemple d'identificació incorrecta


3. Aquesta és la pàgina que veu un usuari no identificat quan ha seleccionat una categoria per realitzar un test. Com es pot veure es carrega un Grid que dóna la possibilitat d'escollir un dels tests disponibles, en aquest cas tan sols n'hi ha un.





TEST DE CONDUIR

[Home](#) 
[Contactar](#)

Seleccioni la categoria de Tests
INDEX DE TESTS:


Usuari: Password: 

Cotxe  

Anonim

TEST DE COTXE

Fes click al botó del test que vulguis realitzar

TESTS	APROBATS	ERRADES	DATA
001			

Telèfon: 932548551 E-mail autotest@test.com

Fig 33: Pàgina perquè l'usuari s'identifiqui

4. Aquesta és la pàgina que veu un usuari Registrat quan ha seleccionat una categoria per realitzar un test. També es carrega un Grid que dóna la possibilitat de triar un dels tests disponibles i a la vegada un altre Grid permet veure la seva fitxa personal per la categoria que ha escollit. També es mostren dades estadístiques com el percentatge de test aprovats i la mitjana d'errors. L'aplicació li dóna la benvinguda i mostra el nom i cognoms de l'usuari que s'ha logonejat correctament.



[Home](#)

[Contactar](#)

TEST DE CONDUIR

Seleccioni la categoria de Tests
INDEX DE TESTS:

Usuari: Password:

Cotxe

Hola *Juan morales gonzalez*

TEST DE COTXE

Percentatge aprovats: **20%**

Mitjana d'errors **10,4**

Col·leció de tests

Aquesta es la teva fitxa personal

TESTS
001

TESTS	APROBAT	ERRADES	DATA
001		3	07/05/2007 5:37:40
001		2	26/05/2007 12:39:31
001		2	26/05/2007 20:05:14
001		21	28/05/2007 18:39:47
001		24	01/06/2007 18:19:48

Telèfon: 932548551 E-mail autotest@test.com

Fig 34: Pàgina de test d'un usuari registrat

- Aquesta pàgina correspon al moment posterior a corregir un test. L'aplicació mostra on s'han produït els errors i també marca els encerts, tot això utilitzant imatges per aconseguir una web més senzilla i visual. El sistema comptabilitza el número d'errades que l'usuari ha fet en aquest test i li mostra si el test està aprovat o suspès.

AUTO TEST

Home  [Contactar](#)

TEST DE CONDUIR

Seleccioni la categoria de Tests
INDEX DE TESTS:
 Cotxe 

Usuari: Password: 

Hola *Juan morales gonzalez*

Numero de Test: **001** N° d'errades: **2**
 Categoria: **Cotxe** El Test estaria: **APROVAT**

1- ¿Circula bien un camión a 100 km./h en una autopista?
 a) Sí, si efectúa un adelantamiento.
 b) No.
 c) Sí

2- ¿Cual será la velocidad máxima a la que podrá circular una furgoneta con remolque ligero por una autovía con arcén de menos de 1,5 m. de ancho?
 a) 90 Km./h. más 20 Km./h. para adelantar.
 b) 90 Km./h.
 c) 80 Km./h.

Fig 35: Pàgina de qüestionari

6. En aquesta imatge es mostren els dos botons existents al final del qüestionari. El botó Corregir dóna a l'usuari la possibilitat de corregir el seu test i si l'usuari està identificat i registrat, el sistema guarda els resultats a la seva fitxa personal. El botó Anar a Fitxa redirecciona l'usuari registrat i identificat a la pàgina on pot veure la seva fitxa, però si l'usuari és anònim aleshores l'aplicació el redirecciona cap a la pàgina de registre.

29- ¿Se deben bajar las pendientes en punto muerto?

a) Sí, con el motor al ralentí.

b) No.


c) Sí, y frenando con el freno de mano

30- En la calzada de doble sentido de circulación de la fotografía no están marcados los carriles, ¿está obligado a circular por la derecha?

a) No, porque como no hay carriles pintados puedo circular por el centro de la calzada.

b) Sí.

c) No, porque no se aproxima ningún vehículo por el sentido contrario.



Corregir Anar a Fitxa

Telèfon: 932548551 E-mail autotest@test.com

Fig 36: Part final de la pàgina del qüestionari

7. L'usuari pot registrar-se en aquesta pàgina i aleshores podrà gaudir de la seva fitxa personal per seguir la seva evolució.

[Home](#) [Contactar](#)**TEST DE CONDUIR**

Seleccioni la categoria de Tests

INDEX DE TESTS:Cotxe Usuari: Password: [Anonim](#)

Vols Tenir una fitxa personal!

Enregistrat gratuïtament i podràs veure fàcilment en la teva fitxa la teva evolució

Nom: Cognoms: NIF: Data naixament: Direcció: Població: Codi Postal: Telèfon: Email: Login: Password: 

Despres de confirmar el teu registre, has de fer el login per accedir a Autotes com a usuari registrat!

Fig 37: Pàgina de registre de l'usuari

3. CONCLUSIONS

El resultat final del treball ha estat molt satisfactori perquè ha permès realitzar una primera aproximació a la tecnologia .NET de Microsoft i la tecnologia AJAX, adquirint suficients coneixements com per a desenvolupar una aplicació Web de complexitat mitjana. També ha servit com a repàs i posada en pràctica de nombrosos conceptes treballats en diverses assignatures al llarg de la carrera, especialment de les assignatures de programació, bases de dades i enginyeria del programari. Es pot dir que s'han arribat a tots els objectius proposats inicialment, tant pel desenvolupament de l'aplicació com pels coneixements adquirits. D'altra banda, la pròpia redacció de la memòria és un exercici molt recomanable per a assolir una base sòlida en la redacció de documentació tècnica.

Cal puntualitzar que ha estat la primera vegada que s'ha programat amb capes amb un resultat prou satisfactori i també s'han pogut recordar moltes intruccions d'SQL que han facilitat l'èxit dels objectius del producte.

També cal destacar que si bé s'han trobat moltes dificultats durant la fase d'implementació, aquestes han pogut ser subsanades. Per exemple, la connexió amb la base de dades donava un error que s'ha solucionat donant una valor al pool de connexions més gran de l'habitual; o en el correcte funcionament del ValidateControls quan interactuava amb la tecnologia AJAX, que ha estat necessari la instal.lació d'una llibreria nova.

Finalment, en relació al producte obtingut es pot afirmar que, en resum, l'aplicació permet que un usuari pugui entrenar-se i aprendre la tècnica dels tests per tal de poder presentar-se a l'examen de teoria pel carnet de conduir, a través d'una navegació per la web senzilla i intuïtiva, i paral.lelament l'administrador de la web pot modificar-ne els continguts i actualitzar els usuaris fàcilment.

4. GLOSSARI

Terme	Descripció
ADO.NET	Solució proposada per Microsoft per a permetre l'accés a bases de dades desde la seva plataforma .NET
AJAX	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.
Arquitectura.Net	Plataforma independent del llenguatge pel desenvolupament de serveis Web.
ASP.NET	Conjunt de tecnologies de desenvolupament d'aplicacions web comercialitzat per Microsoft. És usat per a construir llocs web domèstics, aplicacions web i serveis XML. Parteix de la plataforma .NET de Microsoft i és la tecnologia successora de la tecnologia Activi Server Pages (ASP).
Cas d'ús	Diagrama que mostra les funcions d'un sistema des del punt de vista de les seves interaccions amb l'exterior.
DataReaders	Objecte de ADO.NET per accedir a les dades, les característiques del qual es detallen a l'apartat 2.3.1 ADO.NET

Terme	Descripció
DataSets	Objecte de ADO.NET per accedir a les dades, les característiques del qual es detallen a l'apartat 2.3.1 ADO:NET
Diagrama de Classes	Descripció visual que representa l'estructura i el comportament d'objectes del sistema.
Diagrama de Gantt	Representació gràfica d'una planificació temporal, sobre dos eixos: el vertical és on es disposen les tasques del projecte i l'horitzontal on es representa el temps de les tasques.
MasterPage (pàgina principal)	En ASP.NET, pàgina que defineix el disseny d'un conjunt de pàgines. Una pàgina principal pot contenir text estàtic i controls que han d'aparèixer en totes les pàgines. Les pàgines principals es combinen en temps d'execució amb les pàgines de contingut que defineixen el contingut específic de la pàgina.
Mètode Cascada	Mètode per gestionar el cicle de vida d'un Projecte.
Programació en capes	Estil de programació en el qual l'objectiu primordial és la separació de la lògica de negocis de la lògica de disseny.
SQL	Structured Query és un llenguatge estàndard de comunicació amb bases de dades. És un llenguatge normalitzat que permet treballar amb qualsevol tipus de llenguatge.

Terme	Descripció
SQL-Server 2005	Sistema de gestió de base de dades relacionals (SGBD) basat en el llenguatge SQL, capaç de posar en disposició de molts usuaris grans quantitats de dades de manera simultània. Versió 2005.
Tecnologia .NET	Marc de creació d'aplicacions que permet desenvolupar aplicacions web, client/servidor, Serveis Web.
TFC	Treball final de carrera.
Visual Basic.NET	Un dels llenguatge de programació orientat a objectes, suportat per la plataforma .NET

5. BIBLIOGRAFIA

- Manuals de consulta
 - Diseño de aplicaciones con Microsoft ASP.NET.
Douglas J.Reilly, editorial. Mc Graw Hill
 - Microsoft Visual Basic 2005.
. Charre Ojeda, Francisco. editorial Anaya Multimedia
- Referències procedents d'Internet:
 - **Video:** Advanced UI Controls for **ASP.NET** AJAX **Learn** how **Microsoft's ASP.NET** AJAX framework brings a whole new era of richness and interactivity to web [...] <http://www.asp.net/learn/videos/default.aspx>
 - GridViewGuy is a knowledge based website that hosts ASP.NET articles.
<http://www.gridviewguy.com/ArticleDetails.aspx?articleID=139>
 - Test de autoescuela gratis para el carnet de conducir, tests de examen ciclomotor, test motocicleta, tests autoescuelas.
<http://www.todotest.com/>
 - Iniciación a la programación en ADO .NET, uso de comandos SQL en una aplicación de consola de Visual Basic .NET.
http://www.elquille.info/colabora/puntoNET/miliuco_adonet.htm
- Altres enllaços d'interès:
 - Enciclopèdia Lliure Wikipedia <http://en.wikipedia.org/>
 - Fòrum ASP.NET <http://www.forosdelweb.com>
 - Fòrum Microsoft MSDN <http://forums.microsoft.com/msdn/default.aspx?siteid=1>
 - MSDN Espanyol <http://www.microsoft.com/spanish/msdn/spain/default.mspix>

6. ANNEXOS

Les funcionalitats que es podrien desenvolupar per tal d'ampliar i perfeccionar l'aplicació són els següents:

- Cada vegada que un usuari s'enregistri al sistema, l'administrador hauria de rebre un mail informant-li de la nova alta.
- Quan s'elimini un usuari el sistema hauria de donar la possibilitat que l'administrador elimini també les seves fitxes.
- Quan l'usuari vulgui contrastar la informació o qualsevol concepte aparegut a les preguntes i respostes dels tests, tingui la possibilitat d'accedir directament a l'apartat corresponent de la teoria on s'explica el concepte o dubte.
- Donar la possibilitat a l'usuari de realitzar tests enfocats a capítols determinats de la teoria.

Així mateix, possibles línies de seguiment per al desenvolupament d'altres projectes futurs relacionats amb aquest són les següents:

- Es podria desenvolupar una aplicació on s'aprengués a circular amb vehicle per espais simulats d'una zona com una ciutat, amb l'objectiu de familiaritzar-se amb les senyals i normes de tràfic i poder practicar tant temes teòrics de l'examen de conduir com temes per preparar-se i familiaritzar-se amb la pràctica.