



Escola Universitària
Politécnica de Mataró

Enginyeria Tècnica en Informàtica de Gestió

(Restaurant 2.0)

(Autor : Xavier Pardo Montsant)

(Ponent : Xavier Font Aragonés)

(Tardor) (2009)

Resum / Resumen / Abstract

L'objectiu del projecte és el d'aconseguir realitzar una solució per a un restaurant que contingui tres aplicacions, una pàgina web, una aplicació de gestió per al gerent del restaurant i una aplicació per a PDA per als treballadors.

La idea és que l'usuari que utilitza la pàgina web, pugui d'una manera fàcil consultar informació sobre el restaurant, registrar-se per a rebre promocions, rebre una atenció més personalitzada i fer reserves on-line. Que el gerent pugui utilitzar l'aplicació de gestió per controlar d'una manera senzilla una part de la informació del restaurant i com que la base de dades a la que accedirà la web i la que controlarà el programa de gestió serà la mateixa, els canvis que realitzi el gerent des de l'aplicació es reflecteixin ràpidament en la informació que es mostrarà en la web. El gerent també podrà gestionar les reserves que els usuaris facin a través de la web. Per últim tindrem la aplicació de la PDA, aquesta aplicació permetrà als treballadors recollir les comandes dels clients, passar-les a la cuina perquè les realitzin i efectuar el cobrament d'aquestes tant en efectiu com en targeta de crèdit. El treballador per realitzar aquestes tasques disposarà d'una PDA i una impressora portàtil que aniran lligades per "bluetooth".

Per dur a terme aquesta solució m'he decantat per a utilitzar varies tecnologies, que m'han semblat les més adients d'acord amb les que ofereix el mercat. He decidit utilitzar Flex Builder (potent dissenyador de pàgines web) per a realitzar el disseny de la pàgina, c# .Net per a realitzar un Web Service per que la web obtingui la informació de la base de dades, el mateix llenguatge (c# .Net) per a realitzar el software de gestió per al gerent i per l'aplicació de la PDA. Per la base de dades he utilitzat SQLExpress (gratuït i potent gestor de bases de dades).

El objetivo del proyecto es el de conseguir realizar una solución para un restaurante que contenga tres aplicaciones, una página web, una aplicación de gestión para el gerente del restaurante i una aplicación para PDA para los trabajadores.

La idea es que el usuario que utiliza la página web, pueda de una manera fácil consultar información sobre el restaurante, registrarse para recibir promociones, recibir una atención más personalizada y hacer reservas on-line. Que el gerente pueda utilizar la aplicación de gestión para controlar de una manera sencilla una parte de la información del restaurante y como la base de datos a la que accederá la web y la que controlará el programa de gestión será la misma, los cambios que realice el gerente desde la aplicación de gestión se vean reflejado rápidamente en la información que se mostrará en la web. Por último tendremos la aplicación de la PDA, esta aplicación permitirá a los trabajadores recoger los pedidos de los clientes, pasarlos a la cocina para que los realicen y efectuar el cobro de estos ya sea en efectivo o tarjeta de crédito. El trabajador para realizar estas tareas tendrá una PDA i una impresora portátil que irán unidas por "bluetooth".

Resum

Para llevar a término esta solución me he decantado por utilizar varias tecnologías, que me han parecido las más adecuadas de acuerdo con las que ofrece el mercado. He decidido utilizar Flex Builder (potente diseñador de páginas web) para realizar el diseño de la página, c#.Net para realizar un Web Service para que la web obtenga la información de la base de datos, el mismo lenguaje (c# .Net) para realizar el software de gestión para el gerente y para la aplicación de la PDA. Para la base de datos he utilizado SQLExpress (gratuito y potente gestor de bases de datos).

The idea of the project is to realize a solution for a restaurant containing three applications, a web page, a management software for the restaurant manager and a PDA software for the workers.

The idea is that the user who uses the web page, could in an easy way consult the information about the restaurant, register to receive promotions, receive a more personalized attention and to do on-line reserves. That the restaurant manager could use the management application to control a part of the restaurant information in an easy way and , because the database would be the same for the web and management software, the changes that the manager made to the database will be quickly shown in the web information. At last we have the PDA software, this software will let the workers take the notes from the clients, give it to the kitchen and finally get paid by the customers, the payment would be in hand or by credit card. The worker will have a PDA and a printer that will be joined by "bluetooth".

To take to term this solution i have decided to use some technologies, which have seemed to me the best ones that offers the market. I have decided to use Flex Builder (powerful web page designer) to realize the design of the page, c#. Net to realize a Web Service in order that the web obtains the information of the database, the same language (c #.Net) to realize the management software and the PDA software. For the database i have used SQLExpress (free and powerful database manager).

Índex

1. Introducció	pàg. 1
2. Objectius	pàg. 3
2.1 Requeriments de l'aplicació	pàg. 4
2.1.1 Requeriments no funcionals	pàg. 4
2.1.2 Requeriments funcionals	pàg. 5
3. Avaluació tecnològica	pàg. 7
3.1 Perquè Adobe Flex Builder 3.0?	pàg. 7
3.2 Perquè C# .Net?	pàg. 7
3.3 Perquè un Web Service?	pàg. 8
3.4 Arquitectura utilitzada	pàg. 9
3.4.1 Servidor	pàg.10
3.4.2 Client	pàg.10
3.4.3 Base de dades SQL Express	pàg.11
3.4.4 HTML	pàg.11
4. Planificació inicial	pàg.13
5. Model conceptual	pàg.15
5.1 Glossari de classes	pàg.16
5.2 Dades aplicació PDA	pàg.17
6. Casos d'ús	pàg.19
6.1 Diagrama de casos d'ús (Web)	pàg.19
6.2 Diagrama casos d'us (Programa Gestió)	pàg.20
6.3 Diagrama casos d'ús (Aplicació PDA)	pag.21
6.4 Casos d'ús de la solució	pàg.22
6.4.1 Actors	pàg.22
6.4.2 Cas d'ús registrar usuari	pàg.22

6.4.3	Cas d'ús realitzar reserva	pàg.23
6.4.4	Cas d'ús fer menú	pàg.23
6.4.5	Cas d'ús escollir menú	pàg.24
6.4.6	Cas d'ús consultar menú	pàg.25
6.4.7	Cas d'ús modificar menú	pàg.25
6.4.8	Cas d'ús eliminar menú	pàg.26
6.4.9	Cas d'ús consultar plats	pàg.26
6.5.0	Cas d'ús alta plats	pàg.27
6.5.1	Cas d'ús eliminar plat	pàg.28
6.5.2	Cas d'ús identificar	pàg.28
6.5.3	Cas d'ús enviar informació	pàg.29
6.5.4	Cas d'ús realitzar comanda	pàg.29
6.5.5	Cas d'ús imprimir comanda	pàg.30
6.5.6	Cas d'ús autenticar-se	pàg.30
7.	Diagrames de Seqüència	pàg.31
7.1	Registrar dades	pàg.31
7.2	Consultar dades	pàg.31
7.3	Esborrar dades	pàg.32
7.4	Identificar	pàg.32
7.5	Modificar dades	pàg.33
8.	Disseny	pàg.35
8.1	Capas	pàg.35
8.1.1	Capa de presentació	pàg.35
8.1.2	Capa d'aplicació	pàg.36
8.1.3	Capa de dades	pàg.36
8.2	Disseny de les dades	pàg.37

8.2.1 Model conceptual	pàg.37
8.2.2 Entitats del model de dades	pàg.38
8.2.3 Relacions del model de dades	pàg.41
8.3 Disseny de navegació	pàg.43
9. Conclusions	pàg.47
10. Bibliografia	pàg.49
Annex	pàg.51

1. Introducció

Avui en dia l'objectiu de qualsevol solució és (hauria de ser) millorar i facilitar les tasques a l'usuari/s d'aquesta solució i també s'ha d'aconseguir que les tasques de manteniment i administració siguin fàcils per tal que el client pugui realitzar-les sense la necessitat de comptar amb el creador de la solució o amb un especialista en la matèria.

Resumint, les solucions han de facilitar la realització de les tasques que fins ara es realitzaven d'una altre manera.

Per aquest motiu, aquest projecte esta encaminat a oferir un servei fàcil i innovador a l'usuari, un servei que es diferenciï del que hi ha actualment al mercat, que ofereixi una sèrie d'avantatges que l'usuari trobi positius i el facin decantar-se a utilitzar-lo en front d'altres. Alhora també es vol aconseguir que el client que administri la solució ho faci de manera intuïtiva i que li resulti fàcil manipular totes les eines i informacions que contindrà la solució.

El projecte en si, una pàgina web per a un restaurant, contindrà totes les funcionalitats que pot oferir qualsevol pàgina web per a restauració actualment, això és : informació de la "casa", localització, com arribar

La importància del projecte radica en les funcionalitats "no habituals" que contindrà la pàgina. Aquestes seran, la possibilitat de poder-te registrar i així obtenir ofertes a través del mail i altres promocions que el restaurant pot realitzar per als clients registrats. La possibilitat de fer reserves on-line en temps real i que aquestes siguin acceptades o rebutjades al moment. També alhora de fer una reserva, podrà escollir el menú (entre els menús del restaurant) o podrà confeccionar el seu propi menú seleccionant entre tots els plats de la carta que se li mostraran en pantalla.

Tota aquesta operativa es realitzarà de forma automàtica sense la intervenció del gerent del restaurant, aquest només rebrà les reserves que s'hagin realitzat i s'encarregarà de gestionar-les. Serà la web conjuntament amb la base de dades i un web service d'enllaç qui s'encarregarà de gestionar tota la operativa de la web.

Paral·lelament a la web, hi haurà una altre aplicació que només tindrà el gerent del restaurant. Aquesta aplicació servirà per a administrar i mantenir la web. El gerent disposarà d'una sèrie d'eines amb les que podrà realitzar l'entrada dels diferents plats del restaurant (opcional: amb les seves fotos) a la base de dades de la web, podrà consultar les reserves que s'hagin realitzat així com els menús que s'hagin escollit, també podrà veure la llista de clients que s'hagin registrat i possiblement també podrà realitzar alguna altre operació com la inclusió promocions a la web ...

Introducció

Lligat al programa de gestió tindrem una aplicació per a PDA que ens permetrà que els treballadors puguin realitzar la seva feina d'una manera més ràpida i informatitzada. Amb l'aplicació de la PDA, el treballador podrà prendre nota a les taules, generar les comandes per a la cuina (o podrà fer via wireless o mitjançant una nota impresa), també podrà realitzar el cobrament de les comandes ja siguin en efectiu o amb targeta de crèdit (sempre i quan disposi de connexió GPRS).

2. Objectius i requeriments

L'objectiu és el de dissenyar una solució que contingui tres mòduls un serà la pàgina web d'un restaurant, que ofereixi una sèrie de serveis i informacions a l'usuari que aquest trobi interessants i atractius i que alhora pugui ser mantinguda pel gerent del restaurant d'una forma fàcil i ràpida, l'altre serà un programa de gestió des d'on es puguin gestionar diverses dades del dia a dia del restaurant i a ser possible consultar i modificar dades que s'han entrat des de la pàgina web (que existeixi interacció entre el programa de gestió i la pàgina web). Per últim crear una aplicació que funcioni sobre PDA i que permeti al treballador realitzar les seves funcions fàcilment.

Fites a aconseguir :

- Construir una pàgina Web per al restaurant.
- Realitzar un programa de gestió per al gerent del restaurant.
- Confeccionar una base de dades per al programa de gestió.
- Crear un web service mitjançant el qual la web pugui consultar les dades que s'hagin modificat des del programa de gestió. (Així mantenim la web al dia de forma automàtica).
- Realitzar una aplicació per a PDA que permeti realitzar les comandes i el cobrament d'aquestes en efectiu i targeta de crèdit.
- Que el programa de PDA pugui obtenir dades en local i per wireless. Cobrament per targeta utilitzant GPRS.
- Integrar varies tecnologies per a aconseguir un fi comú, utilitzar Flex per a dissenyar la pàgina web i interactuar amb el web service per accedir a la base de dades. Realitzar una aplicació Windows en c#.net per a gestionar la base de dades i realitzar una aplicació PDA que pugui mitjançant wireless utilitzar el web service per actualitzar les seves dades.

2.1 Requeriments de l'aplicació

Requeriment:

- Condició o necessitat d'un usuari per resoldre un problema o arribar a un objectiu.

Els requeriments poden ser de dos tipus :

- Requeriment no funcionals.
- Requeriments funcionals.

2.1.1 Requeriments no funcionals

Els requeriments no funcionals tenen a veure amb les característiques que d'una o altre forma poden limitar l'aplicació, com la interfície de l'usuari, la fiabilitat del sistema, el manteniment, la seguretat, la portabilitat ...

Requeriments no funcionals de la solució :

1- Recursos humans

Els usuaris que interactuen amb la web, amb el programa de gestió i amb la PDA són usuaris finals, que no tenen perquè conèixer el món de l'informàtica ni han de saber com estan funcionant les aplicacions que utilitzen.

Per tant hem de fer que la utilització dels programes i les seves característiques, siguin fàcils i intuïtius.

2- Interfície d'usuari

La interfície ha de ser clara, entenedora i robusta, així protegirem l'usuari de cometre errors d'ús incorrecte i farem que li sigui agradable utilitzar la nostre solució.

Els perifèrics que utilitzarà l'usuari són ratolí, teclat, monitor i (opcional: impressora) pel que fa al programa de gestió i a la web. Respecte a l'aplicació de PDA l'usuari utilitzarà la PDA, l'accessori corresponent per a la lectura de les bandes magnètiques de les targetes de crèdit i una petita impressora portable.

L'usuari de la web haurà de tenir una connexió Internet per tant també estarà utilitzant de forma indirecte un mòdem o "router". L'usuari de l'aplicació de gestió, haurà de tenir també Internet perquè des del Web Service s'accedeixi a les seves dades. Si es vol utilitzar la solució de PDA en wireless també haurà de tenir un (o més) punt d'accés i per a realitzar el cobrament amb targeta haurà de disposar d'una SIM per utilitzar la connexió GPRS.

3- Tractament d'errors

La solució informarà a l'usuari dels diferents errors i informacions que es considerin rellevants mitjançant pantalles que aquest haurà d'acceptar, o contestar si o no segons el que se li estigui preguntant.

4- Funcionament de l'aplicació

La web ha de poder funcionar d'una forma correcta amb els diferents navegadors que existeixen al mercat, l'aplicació de gestió funcionarà sobre el sistema operatiu Windows XP, ja que esta pensada per funcionar sobre aquest sistema, es pot provar d'utilitzar sobre un altre sistema operatiu però no s'assegurarà el seu correcte funcionament. L'aplicació PDA estarà programada sobre el sistema Windows Mobile 6.1, i es sobre aquest sistema sobre el que s'assegura que funcionarà correctament.

5- Factors de qualitat

Les solucions han de ser fiables, de forma que tant els usuaris, el gestor com els treballadors que les utilitzin tinguin confiança en elles.

La tolerància a les errades hauria de ser 0.

6- Manteniment

La solució ha de ser flexible, amb això el que vull dir es que ha de permetre , modificacions i ampliacions de les seves funcionalitats. Aquestes modificacions i ampliacions s'han de poder fer d'una manera fàcil (el manteniment ha de ser fàcil!), es a dir que realitzar millores després de detectar errors ha de ser possible i ràpid.

2.1.2 Requeriments funcionals

Els requeriments funcionals defineixen les funcions que podrà realitzar la nostra solució. Descriuen els diferents processos a les que estan sotmeses les entrades per produir sortides.

La nostra solució esta dividida en dues parts clarament diferenciades :

- Utilització de la web per part de l'usuari.
- Gestió de la base de dades amb l'aplicació Windows per part del client.
- Utilització de la PDA per al treball diari del restaurant.

1- Utilització web per part de l'usuari

Objectius i requeriments

Un cop dins la web, l'usuari es podrà registrar, consultar informació referent al restaurant (plats, historia, localització...), realitzar reserves on-line i confeccionar un menú o bé consultar els disponibles.

2- Gestió per part del client

Aquí també podrem diferenciar varies parts, per una part tindrem la gestió dels continguts de la web, diferents plats, menús, plats de temporada... Aquesta part serà la d'actualització dels continguts de les diferents taules de la base de dades. Per una altre part tindrem la de consulta de clients, aquí veurem els diferents clients que s'han registrat a la base de dades i podrem utilitzar aquesta informació per a enviar propostes i ofertes i per oferir un servei més personalitzat. Per finalitzar tindrem la part de consulta de les reserves, aquí veurem les reserves que s'hagin fet a través de la web i els menús que s'han escollit. Així controlarem l'ocupació del restaurant i podrem planificar la feina.

3 – Utilització PDA

Amb el programa de la PDA el treballador podrà atendre les peticions bàsiques dels clients com són prendre nota del que aquests volen i realitzar el cobrament de les seves comandes. Així mateix també podrà informar a la cuina de quines són les comandes a preparar.

3. Avaluació tecnològica

Aquí quedarà reflectit el perquè hem escollit les diferents solucions tecnològiques que hem utilitzat en front de les altres que hi ha al mercat sense comparar-les, simplement definint el perquè de la nostra decisió.

3.1 Perquè Adobe Flex Builder 3.0?

Flex builder permet la creació tant de webs com d'aplicacions d'una manera fàcil e intuïtiva. Podem dissenyar webs i que es mostrin d'una manera professional sense que realitzar aquest disseny sigui molt costos i ens requereixi de grans coneixements. Això no vol dir que sigui una eina poc potent perquè alhora podem crear el codi font de la pàgina tant complicat com vulguem, servint-nos d'"scripts", "actionscripts", "mxml". Mitjançant el codi, podem enllaçar amb servidors de bases de dades, o amb web Services creats amb altres llenguatges i crear així aplicacions i webs de gran potencial.

L'IDE en qüestió és molt semblant a qualsevol editor HTML amb entorn WYSIWYG: Té una pestanya per veure el codi MXML que integra elements com títols o camps de formulari a la pel·lícula (com ho faria l'XHTML), l'"ActionScript" que controla els esdeveniments, pot modificar l'MXML (com ho faria JavaScript), i una altra pestanya, aquesta mostrant els elements tal com es veuran al navegador, i es tindran les possibilitats de modificació amb uns simples clics de ratolí i teclat.

3.2 Perquè C# .Net?

Llenguatge de programació orientat a objectes desenvolupat per Microsoft i estandarditzat, com part de la seva plataforma .NET.

La seva sintaxis bàsica deriva de C/C++ i utilitza el model d'objectes de la plataforma .NET el qual és similar al de Java però inclou millores derivades d'altres llenguatges. C# fou dissenyat per a combinar el control a nivell baix de llenguatges com C i la velocitat de programació de llenguatges com Visual Basic.

Avantatges enfront de C i C++

- Compila a codi intermedi (CIL) independent del llenguatge en què hagi estat escrita l'aplicació i independent de la màquina on es vagi a executar
- Recol·lecció d'escombraries automàtica
- Capacitats de reflexió.
- No cal preocupar-se per arxius de capçalera ". H"
- No importa l'ordre en que hagin estat definides les classes ni les funcions
- No hi ha necessitat de declarar funcions i classes abans de definir
- No existeixen les dependències circulars
- Suporta definició de classes dins d'altres
- No existeixen funcions, ni variables globals, tot pertany a una classe

- Tots els valors són inicialitzat abans de ser usats (automàticament s'inicialitzen al valor estandarditzat, o manualment es poden inicialitzar des de constructors estàtics)
- No es poden utilitzar valors no booleans (sencers, coma flotant ...) per condicionals. És molt més net i menys propens a errors

Avantatges enfront de Java

- El rendiment és, en general, molt millor
- CIL (el llenguatge intermedi de .NET) està estandarditzat, mentre que els byte de Java no ho estan
- Suporta força més tipus primitius (*value types*), incloent tipus numèrics sense signe
- Indexats que permeten accedir a qualsevol objecte com si es tractés d'una matriu
- Compilació condicional
- Aplicacions multiprocés fill simplificades
- Suporta la sobrecàrrega d'operadors, que encara que poden complicar el desenvolupament són opcionals i algunes vegades molt útils

3.3 Perquè un Web Service?

És una col·lecció de protocols i estàndards que serveix per intercanviar dades entre aplicacions. Diferents aplicacions de programari desenvolupades en llenguatges de programació diferents i executades sobre qualsevol plataforma poden utilitzar els serveis web per l'intercanvi de dades en una xarxa com Internet. Aquesta gran interoperabilitat s'aconsegueix gràcies a l'adopció d'estàndards oberts.

Avantatges del Web Service:

- Aporten interoperabilitat entre aplicacions molt diverses, en diverses plataformes, diversos llenguatges de programació i de diversos propietaris.
- Els serveis web fomenten l'ús dels estàndards i protocols basats en text, cosa que fa més fàcil accedir al seu contingut i entendre'n el seu funcionament.
- Al funcionar sobre HTTP, poden aprofitar-se de tots els sistemes de seguretat *firewall* existents sense la necessitat de canviar les regles de filtrat.
- Permeten fàcilment oferir serveis integrats que es basin en serveis de diferents propietaris i distribuïts geogràficament.
- Permeten interoperar de forma transparent diferents plataformes.

En el nostre cas, el web service creat amb c# ens facilita tota la interacció amb la base de dades i ens permet que la web l'utilitzi segons les seves necessitats i que el software de la PDA també l'utilitzi sense la necessitat de tenir informacions duplicades, el programa de gestió també el podria utilitzar però com que treballa directament sobre la base de dades no es necessari. Això vol dir que el canvis que es realitzin des de l'aplicació Windows, immediatament es veuran reflectits en la web o en l'aplicació de la PDA depenent de l que es modifiqui, i a l'inrevés passarà el mateix. (per exemple: reserves).

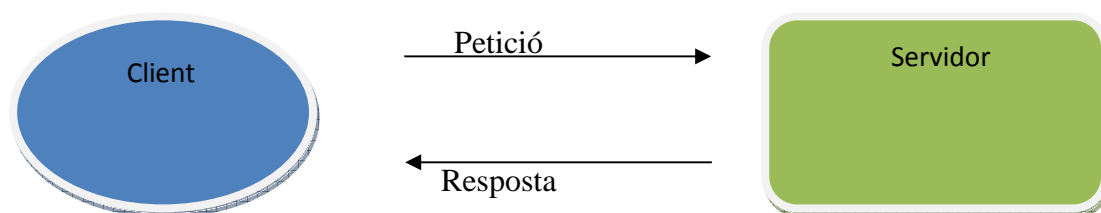
3.4 Arquitectura utilitzada

Quan es parla d'una arquitectura al context de les tecnologies d'informació, ens referim a una combinació d'elements hardware i software que escollim per solucionar un problema, a més de les formes en que es comuniquen aquests elements.

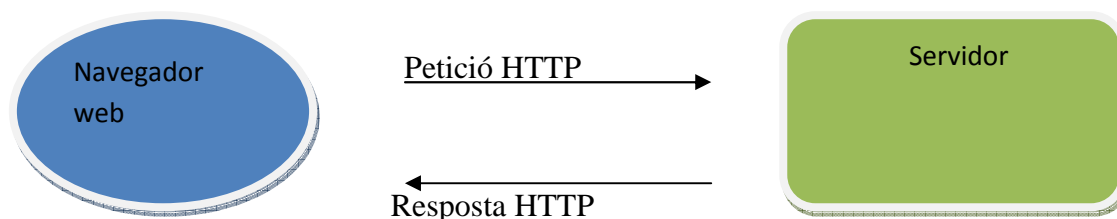
- Arquitectura Client/Servidor

La arquitectura client/servidor és una forma d'organització de les xarxes informàtiques en les que tenim dos tipus de programes: els que fan les peticions dels clients i els que les serveixen. Normalment, aquests programes estan en màquines diferents, però podrien estar a la mateixa maquina.

Quan els dos elements es comuniquin hauran de fer-ho utilitzant un llenguatge conegut pels dos. Aquest llenguatge que utilitzen per comunicar-se es coneix com a protocol.



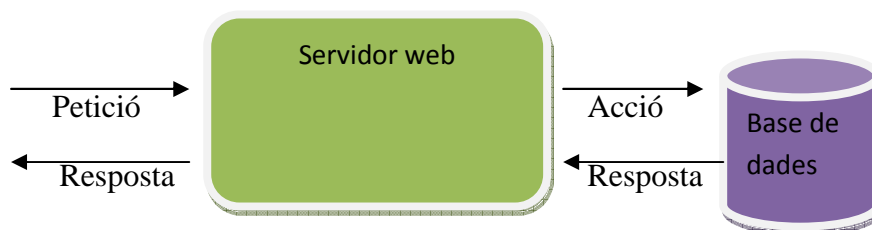
La manera en que funciona el web és un exemple clar d'aquesta arquitectura. Quan estem navegant per Internet, el nostre navegador és el client i la pàgina web a la que ens estem connectant executa el programa servidor.



3.4.1 Servidor

El servidor es un programa que des d'un ordinador, serveix les pàgines web que conté guardades en el seu disc dur. En funció del que demani l'usuari el servidor respondrà a aquestes peticions i així mitjançant aquest servidor se'ns permet de construir la pagina dinàmicament.

La nostra solució pel que fa al servidor respon al següent esquema :



Enviem la petició al servidor web, aquest ens serveix la pagina web i si ha de realitzar alguna acció ja sigui consulta, inserció o modificació, la realitza i ens serveix la resposta a la petició que hem realitzat.

Per exemple: Obrim la pagina web que conté els diferents menús del restaurant, el servidor rep la petició d'obrir la pagina "x" en rebre aquesta petició executa el codi de la pagina i en aquest codi es realitza la consulta a la BBDD, executa l'acció sobre la base de dades i ens retorna la pagina web totalment processada. Així es processarà qualsevol petició que li fem al servidor.

En el cas de l'aplicació de la PDA, aquesta treballarà sempre que es pugui de forma local, per tant tindrà tant la capa d'aplicació com la de les dades en el mateix lloc. Això es realitzarà així per que la funcionalitat sigui més ràpida i no haguem d'estar a tothora pendents de la comunicació. Així mateix però hi haurà subaplicacions que requeriran de connectar-se a un servidor i esperar una resposta d'aquest (Aquestes però seran opcionals).

3.4.2 Client

Aquest simplement s'ocupa d'enviar les peticions i espera les respostes del servidor. El client (navegador web) demana les pàgines al servidor, que processa la consulta, i li envia la resposta que espera, també comunica amb el Web Service que després de consultar la base de dades li respon el que aquest a demanat o el missatge d'error corresponent.

A la PDA, l'aplicació exercirà de client quan comuniqui per wireless per a l'actualització de les dades i quan faci la petició de pagament per targeta que serà el servidor el qui ens haurà d'autoritzar.

3.4.3 Base de dades SQL Express

SQL Express es la versió "Lite" del Microsoft SQL Server, Aquesta serveix per a ús lliure i distribuïble als desenvolupadors de programari. Amb el SQL Express management studio permet la gestió de la base de dades per part del programador d'una manera ràpida i fàcil. SQL Express ens dota de gran robustesa pel que fa a la base de dades i controla les diferents transaccions que es realitzen en les diferents taules de la base de dades. Pel que fa a la aplicació de gestió treballarem sobre una base de dades creada amb el SQL express i ho farem connectant-nos al servidor de bases de dades que pot estar instal·lat al mateix pc que l'aplicació o en un pc remot. El web service també treballarà connectant-se al servidor de bases de dades.

Pel que fa a la aplicació PDA crearem un arxiu de base de dades SQL Express compact for Windows Mobile, llavors no necessitarem una connexió a un servidor de base de dades si no que treballarem directament sobre aquest arxiu (com si es tractes d'un fitxer pla) utilitzant, però, la mateixa sintaxis SQL que utilitzaríem per a treballar sobre una base de dades. Els arxius de bases de dades del SQL compact Mobile són per a Windows Mobile o CE i utilitzen la terminació ".sdf".

- Interacció entre navegador i la base de dades:
 1. L'usuari carrega una pàgina HTML omple les dades i les envia al servidor web.
 2. Per mitja del TCP/IP les dades arriben al servidor.
 3. El servidor recull les dades i inspecciona la pagina a retornar.
 4. Executa les diferents funcions que hi ha al codi de la pàgina accedint al servidor SQL Express.
 5. SQL Express processa la petició i retorna el resultat.
 6. Retorna la pagina web processada al client.
 7. El client rep la pagina HTML mitjançant la xarxa TCP/IP.
- Interacció entre aplicació PDA i arxiu de base de dades:
 1. Connectem amb el fitxer de base de dades local.
 2. Utilitzem sentències SQL.
 3. Analitzem i utilitzem com ens convingui el que ens retornen les consultes.

3.4.4 HTML

Una part important de qualsevol aplicació web és el llenguatge HTML. HTML, Hipertext Mark up Language (*llenguatge de marques de Hipertext*), és el llenguatge que predomina

per a la construcció de pàgines web. És usat per descriure la estructura i el contingut en forma de text, com també per complementar el text amb objectes com ara imatges. HTML s'escriu en forma de "etiquetes", rodejades per claudàtors angulars (<,>). HTML també pot descriure, fins un cert punt, l'aparença d'un document, i pot incloure *scripts* (per exemple Javascript), els quals poden afectar el comportament de navegadors web i altres processadors HTML.

Al nostre projecte com a totes les aplicacions d'aquesta tipus tenim l'estructura bàsica de les pàgines fetes amb HTML amb codi MXML(de FLEX) incrustat dins d'aquest.

4. Planificació inicial

La planificació inicial del projecte estableix les següents directives de treball i temps d'execució de les diferents parts de què consta, amb una durada determinada. En aquest sentit, les tasques principals que s'han definit per realitzar són les següents:

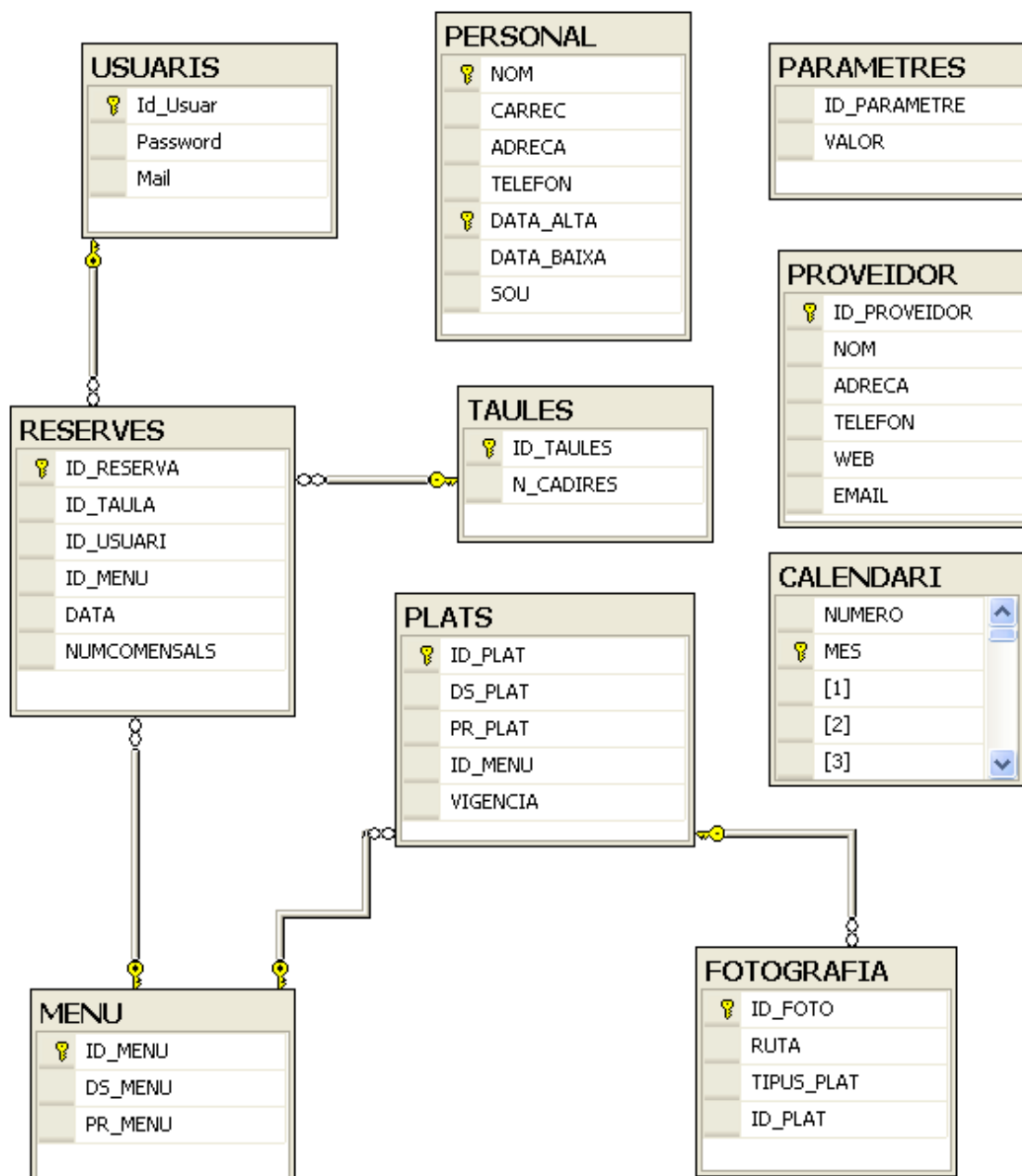
- Estudi de l' informació a processar i de les eines més adequades per a fer-ho.
- Desenvolupament d'una aplicació web per a un restaurant on es permetin registres, reserves on-line i altres opcions.
- Aplicació per gestionar les dades del restaurant, actualitzar-les, modificar-les... i així quan les altres aplicacions accedeixin a les dades aquestes estiguin actualitzades.
- Confecció d'una base de dades per a l' informació del restaurant.
- Aplicació per la PDA per poder realitzar comandes , imprimir-les i gestionar el cobrament.
- Confecció d'una base de dades per a la PDA on tinguem les dades per que el personal pugui validar-se.

Aquest són a grans trets els tres punts que ens marquen la línia de treball a seguir. Amb aquestes dades s'ha realitzat el següent esquema marcant els temps a dedicar en les diferents parts del projecte.

- | | |
|---|-------------------------|
| - Anàlisi del problema | 40 hores (1 setmana). |
| - Recerca i estudi de les tecnologies a utilitzar | 80 hores (2 setmanes). |
| - Disseny | 80 hores (2 setmanes). |
| - Implementació | 240 hores (6 setmanes). |
| - Fase de Test | 40 hores (1 setmana). |
| - Documentació | 60 hores (1'5 setmana). |

5. Model conceptual Base de dades (Gestió i web service)

El model conceptual mostra el diagrama UML d'aquest model.



5.1 Glossari de classes

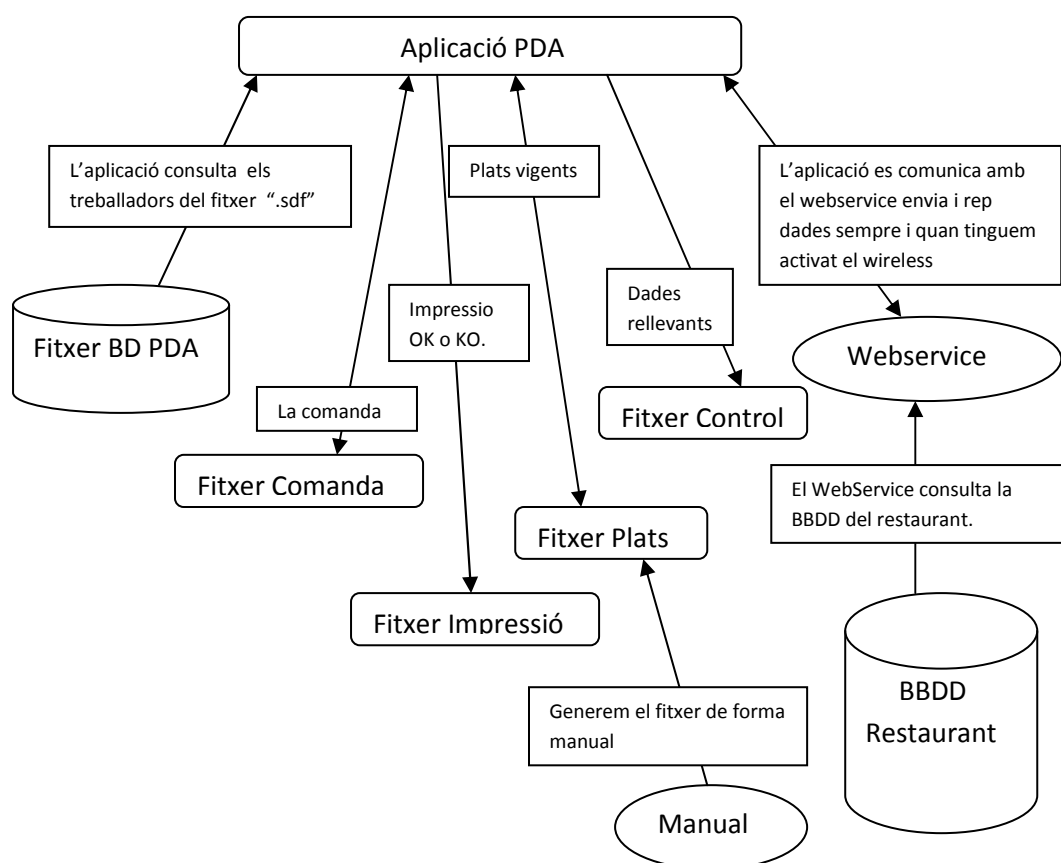
- **Usuaris:** Conté els usuaris que es registrin a la base de dades i els que el gerent entri a través del programa de gestió, del Usuaris registrats tindrem un identificador (pot ser el seu nom), una contrasenya amb la que es poden validar a la pàgina web i el seu mail, que ens pot interessar per a enviar informació, ofertes...
- **Personal:** Conté la informació del personal, aquí tindrem el seu nom, el càrrec que ocupen, la seva adreça, telèfon, sou i la data d'alta i baixa. El camp de la data de baixa ens servirà per saber els treballadors que ja han plegat.
- **Proveïdor:** Conté les dades que el gerent vagi entrant sobre els proveïdors que utilitza, d'aquests tindrem un identificador, el nom, l'adreça, el telèfon, i si en té la web i el correu electrònic.
- **Reserves:** Emmagatzema tant les reserves que s'han realitzat on-line per part del usuaris com les que pot realitzar el gerent des del programa de gestió.
- **Taules:** Conté cada taula que hi ha al restaurant i el número de cadires que aquesta té.
- **Plats:** Conté els diferents plats que es fan al restaurant i que el gerent s'encarrega d'anar entrant a la base de dades. Els camps seran els següents, identificador del plat, descripció del plat, preu d'aquest, identificador del menú al que pertany (si no pertany a cap menú serà 0). I per últim tindrem un camp anomenat vigència que estarà a "true" o "false" depenent si el plat aquell s'està fent o no en aquell moment.
- **Menú:** Conté els diferents menús del restaurant que haurà creat el gerent des de l'aplicació de gestió.
- **Paràmetres:** Conté diferents paràmetres configurables en l'aplicació de gestió.
- **Fotografia:** Conté les fotografies dels plats. El gerent des del programa de gestió pot posar fotografies per cada plat que hi ha a la base de dades. Aquí ha d'establir un identificador de la foto i la ruta d'aquesta foto.
- **Calendari:** Aquí es defineix el nom da cada mes amb el seu identificador i després es marcarà cada dia segons sigui festiu o no.

5.2 Dades Aplicació PDA

Les dades que utilitza l'aplicació de la PDA, les emmagatzemarà de dues maneres diferents, amb un fitxer de SQL Express compact Mobile edition (.sdf) i amb fitxers plans de text. Dins el fitxer ".sdf" tindrem una taula usuaris i hi haurà la dos camps un amb la identificador dels treballadors i l'altre amb la seva contrasenya associada. Així cada cop un treballador utilitzi el programa s'haurà de validar.

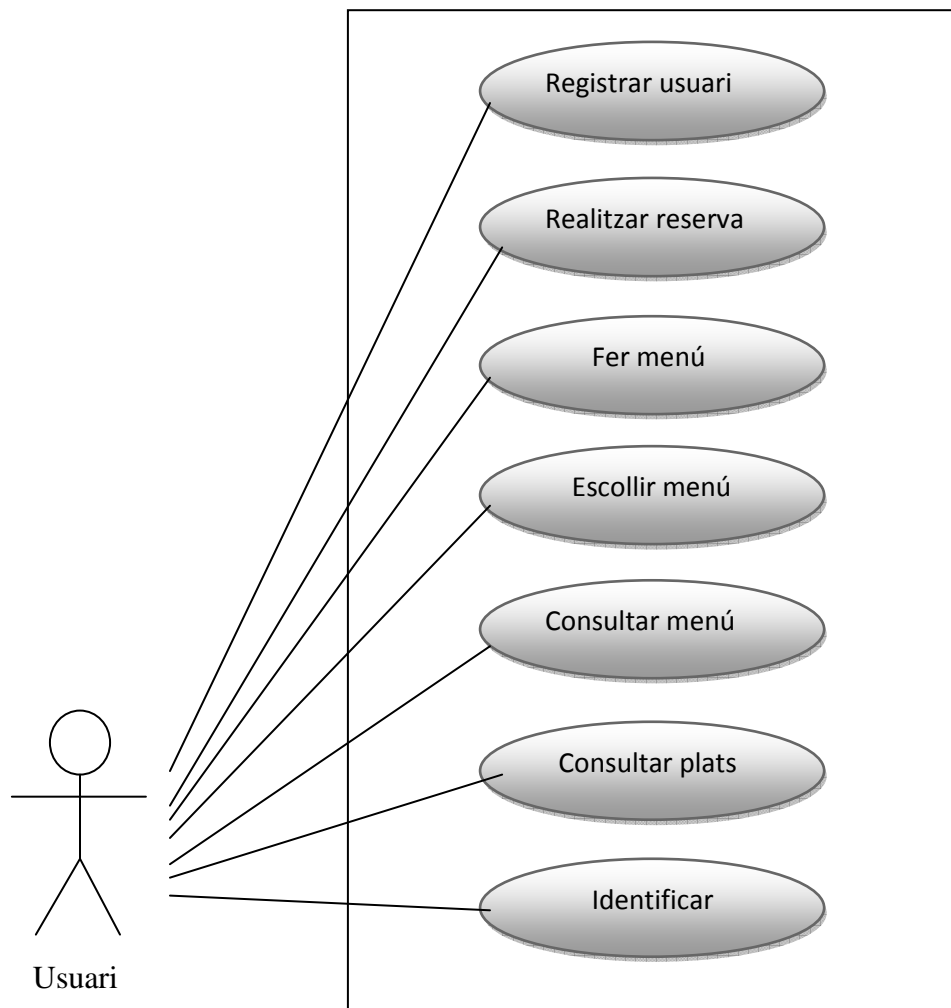
Als fitxers de text i guardarem les dades que anem generant durant l'ús de l'aplicació, algunes d'aquestes simplement seran de control, d'altres seran d'us consultiu i algun per a guardar la informació sobre la que estem treballant.

Tindrem un fitxer en el que anirem guardant la comanda que estem fent, i així la podem recuperar en qualsevol moment. Un altre fitxer ens indicarà si la impressió ha estat correcte o no així, si no s'imprimeix es podrà consultar aquest fitxer. El fitxer que contindrà tota la llista de plats i preus per als clients, podrà ser introduït de manera manual en el directori de l'aplicació o si tenim connexió wireless, serà el programa el que s'encarregarà de preguntar mitjançant el web service quins son els plats vigents i el generarà.

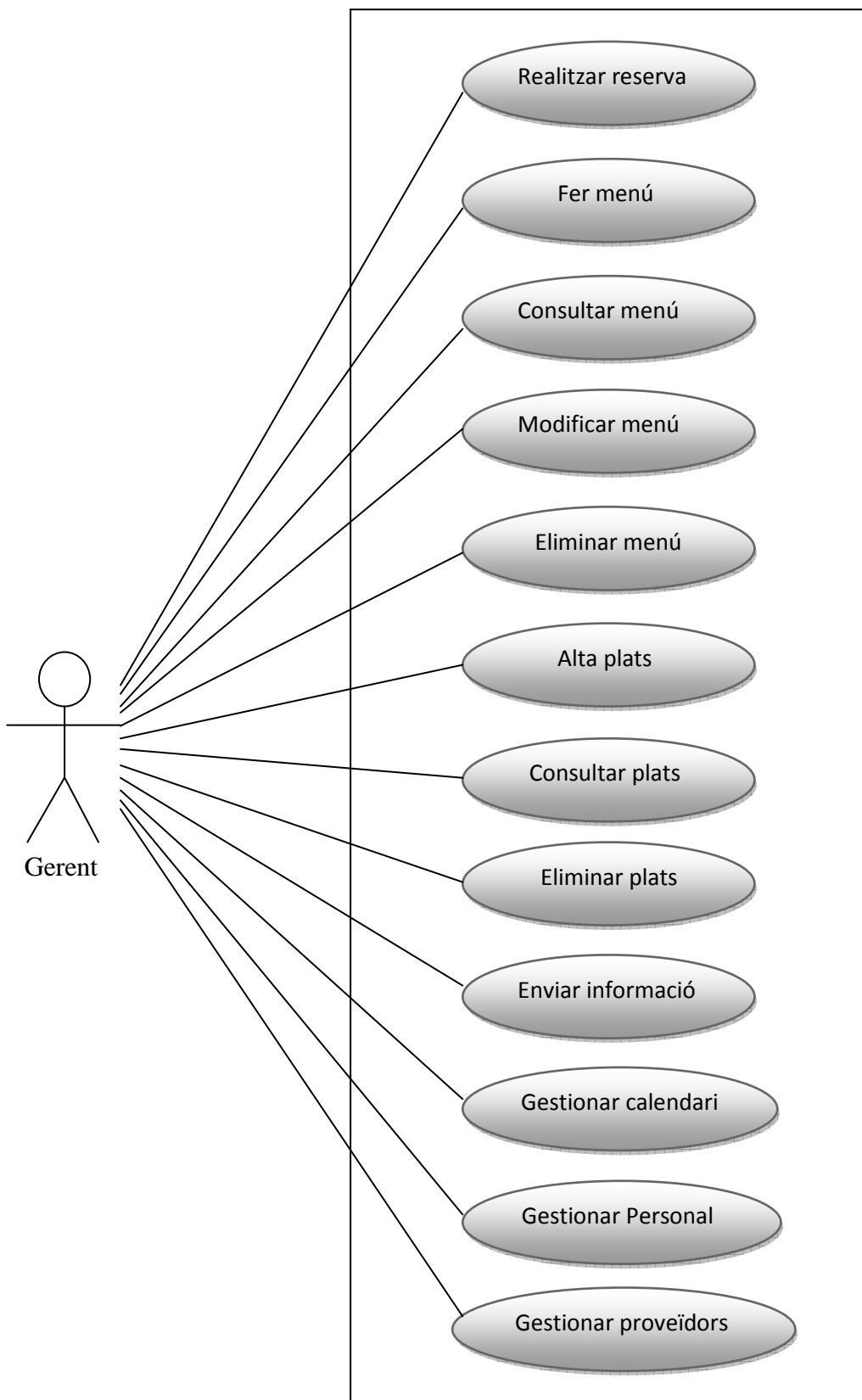


6. Casos d'ús

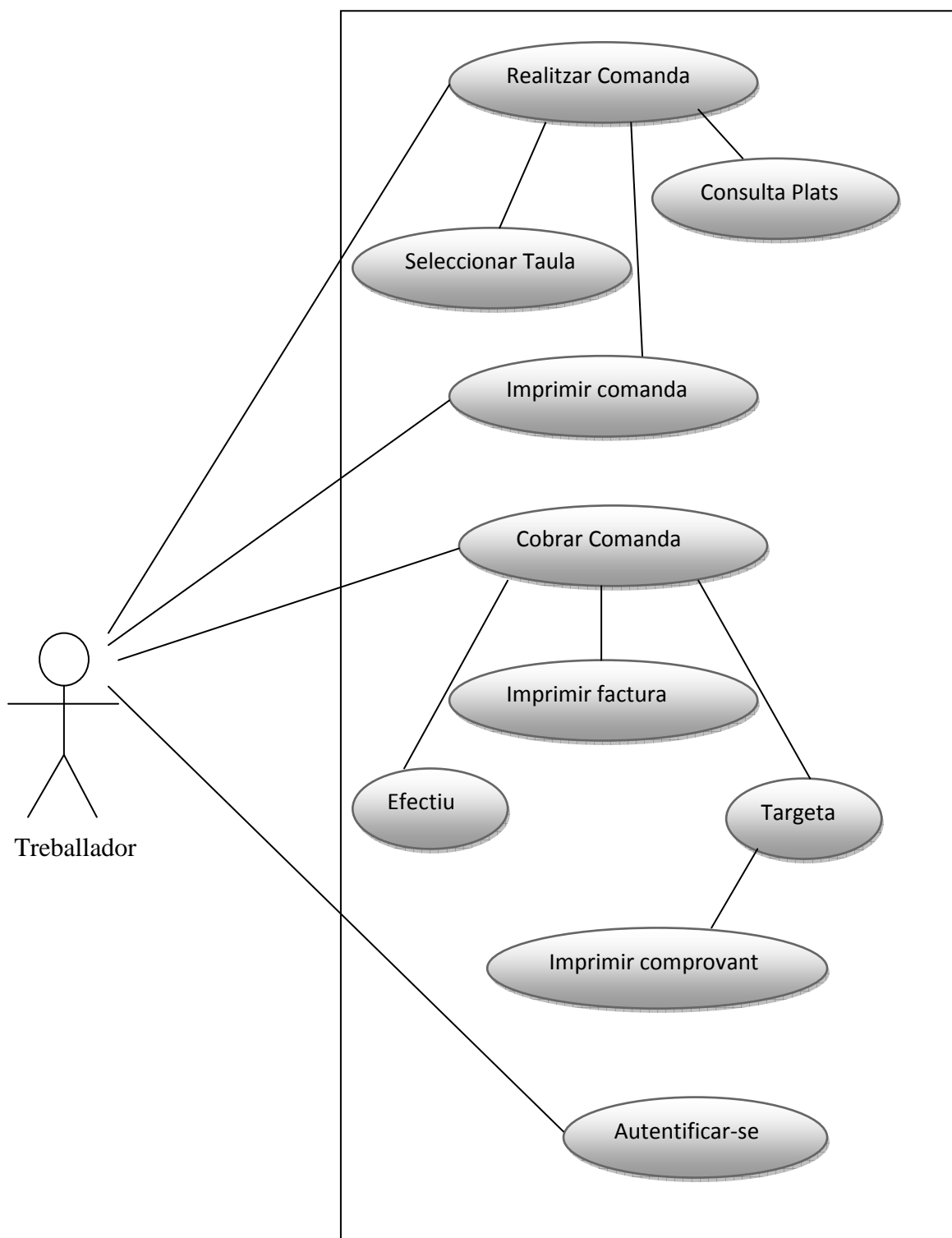
6.1 Diagrama de casos d'ús (Web)



6.2 Diagrama de Casos d'ús (Programa gestió)



6.3 Diagrama de Casos d'ús (Aplicació PDA)



6.4 Casos d'ús de la solució

6.4.1 Actors

Hi han tres actors que interactuen amb la nostra solució. Un és l'usuari que és el que es registra, consulta i utilitza les diferents eines de les que disposa a la web. Un altre és el gerent que gestiona plats, menús, envia informació als usuaris, s'encarrega de consultar les reserves i manté les dades internes del restaurant. Per últim el treballador que utilitza la PDA per a realitzar les comandes que li demanen el clients, imprimir-les i cobrar-les.

6.4.2 Cas d'ús registrar usuari

Cas d'ús : Registrar usuari.

Actor : Usuari.

Descripció : L'usuari entra a la web i es registra omplint el formulari corresponent.

Requeriments: Cap.

Curs esdeveniments :

1. L'usuari entra a la web i apreta el botó registrar-se.
2. L'usuari omple el formulari per registrar-se.
 - L'usuari s'equivoca al omplir el formulari.
 - El sistema informa l'usuari i el retorna a 2.
3. El sistema registra l'usuari a la base de dades i ho confirma a l'usuari.
4. L'usuari continua navegant per la web o surt.

6.4.3 Cas d'ús realitzar reserva

Cas d'ús : Realitzar reserva.

Actor : Usuari.

Descripció : L'usuari omple el formulari de reserva i el sistema li confirma o no la reserva.

Requeriments: Estar registrat.

Curs esdeveniments :

1. L'usuari entra a la web s'identifica i apreta el botó de reserves.
 - L'usuari s'equivoca en l'identificació.
 - El sistema li comunica i el retorna a 1.
2. L'usuari omple el formulari de registre.
 - L'usuari s'equivoca al omplir el formulari o no es possible fer la reserva.
 - El sistema informa l'usuari i el retorna a 2.
3. El sistema registra registra la reserva a la base de dades i ho confirma a l'usuari.
4. L'usuari continua navegant per la web o surt.

6.4.4 Cas d'ús fer menú

Cas d'ús : fer menú.

Actor : Usuari, Administrador.

Descripció : L'usuari estar omplint el formulari de reserves i vol confeccionar el seu propi menú amb els diferents plats de la carta. L'administrador des del programa de gestió selecciona l'opció de donar d'alta menú i confecciona el menú en funció dels plats existents.

Requeriments: L'usuari ha d'estar realitzant una reserva. L'administrador ha d'haver donat d'alta els plats primer.

Curs esdeveniments Usuari:

1. L'usuari pitja confeccionar menú durant la reserva.
2. L'usuari confecciona el menú a partir dels plats que se li presenten.
3. El sistema registra el nou menú a la base de dades i ho confirma a l'usuari.
 - Si el registre a la base de dades no es possible s'informa a l'usuari i retorna a 2.

Curs esdeveniments Administrador:

1. L'administrador selecciona donar d'alta menú.
2. confecciona el menú a partir dels plats que se li presenten.
3. El sistema registra el nou menú a la base de dades i ho confirma a l'administrador.
 - Si el registre a la base de dades no es possible s'informa a l'administrador i retorna a 2.
4. L'administrador continua dins el programa.

6.4.5 Cas d'ús escollir menú

Cas d'ús : Escollir menú.

Actor : Usuari.

Descripció : L'usuari esta realitzant la reserva i escull un dels menús predefinits.

Requeriments: Estar realitzant una reserva.

Curs esdeveniments :

1. L'usuari escull un dels menús predefinits mentre estar fent la reserva.
2. L'usuari continua amb la reserva.

6.4.6 Cas d'ús consultar menú

Cas d'ús : Consultar menú.

Actor : Usuari, Administrador.

Descripció : L'usuari clica el botó de consulta de menús i visualitza els menús vigents. L'administrador selecciona l'opció de visualitzar els menús i el sistema els hi presenta en pantalla.

Requeriments: Cap.

Curs esdeveniments Usuari:

1. L'usuari clica el botó de consulta de menús.
2. Se li mostren els menús vigents en pantalla.
3. L'usuari continua navegant per la web o surt.

Curs esdeveniments Administrador:

1. L'administrador clica el botó de consulta de menús.
2. Se li mostren tots els menús en pantalla.
3. L'administrador continua dins de l'aplicació.

6.4.7 Cas d'ús modificar menú

Cas d'ús : Modificar menú.

Actor : Administrador.

Descripció : L'administrador selecciona l'opció de modificar menús actuals.

Requeriments: Hi ha d'haver menús registrats al sistema.

Curs esdeveniments :

1. L'administrador selecciona l'opció modificar menú.
 - No hi ha menús registrats, s'informa a l'usuari.
 - L'administrador continua dins de l'aplicació.
2. Es mostren els menús registrats per a escollir el que s'ha de modificar.
3. Es modifica el menú escollit.
4. Es registren les modificacions a la base de dades.

6.4.8 Cas d'ús eliminar menú

Cas d'ús : Eliminar menú.

Actor : Administrador.

Descripció : L'administrador selecciona l'opció d'eliminar menú, tria el menú a eliminar i l'esborra de la base de dades.

Requeriments: Hi ha d'haver menús registrats a la base de dades.

Curs esdeveniments :

1. L'administrador selecciona eliminar menú.
 - Si no hi ha menús, s'informa a l'administrador.
 - Se'l manté dins de l'aplicació.
2. L'administrador selecciona el menú a eliminar.
3. El sistema elimina el menú de la base de dades i ho confirma a l'administrador.
4. L'administrador continua dins de l'aplicació.

6.4.9 Cas d'ús consultar plats

Cas d'ús : Consultar plats.

Actor : Usuari, Administrador.

Descripció : L'usuari clica el boto de consulta de plats. L'administrador selecciona l'opció de consulta de plats.

Requeriments: Hi ha d'haver plats registrats o no es mostrarà res.

Curs esdeveniments usuari :

1. L'usuari clica el botó de consulta de plats
2. Se li mostren els plats per pantalla.
3. L'usuari continua navegant per la web o surt.

Curs esdeveniments administrador :

1. L'administrador selecciona consulta de plats
2. Se li mostren els plats per pantalla.
3. L'administrador continua dins de l'aplicació.

6.5.0 Cas d'ús alta plats

Cas d'ús : Alta plats.

Actor : Administrador.

Descripció : L'administrador dona d'alta un plat a la base de dades.

Requeriments: Cap.

Curs esdeveniments :

1. L'administrador selecciona l'opció donar d'alta plat.
2. Omple les dades per a donar d'alta el plat.
3. El sistema registra el plat a la base de dades i ho confirma a l'administrador.
 - Error registrant el plat a la base de dades. S'informa al administrador.
 - Es retorna al punt 2.
4. L'administrador continua dins de l'aplicació.

6.5.1 Cas d'ús eliminar plat

Cas d'ús : Eliminar plat.

Actor : Administrador.

Descripció : L'Administrador elimina un plat de la base de dades.

Requeriments: Hi ha d'haver plats donats d'alta.

Curs esdeveniments :

1. L'administrador selecciona eliminar plat.
2. Selecciona el plat a eliminar d'entre els que se li mostren.
3. El sistema elimina el plat de la base de dades i ho confirma a l'administrador.
4. L'administrador continua dins de l'aplicació.

6.5.2 Cas d'ús identificar

Cas d'ús : Identificar.

Actor : Usuari.

Descripció : L'usuari entra a la web i s'identifica omplint les dades d'usuari i contrasenya.

Requeriments: ha d'estar registrat.

Curs esdeveniments :

1. L'usuari omple els camps d'usuari i password i clica identificar-se.
2. El sistema comproba les dades amb la base de dades
 - Error d'identificació. Se li notifica a l'usuari.
 - Retornem al punt 1.
3. L'usuari continua navegant ara identificat.

6.5.3 Cas d'ús enviar informació

Cas d'ús : Enviar informació.

Actor : Administrador.

Descripció : L'administrador envia informació a l'usuari.

Requeriments: L'usuari ha d'estar registrat.

Curs esdeveniments :

1. L'administrador selecciona l'opció d'enviar informació.
2. Omple les dades que li vol fer arribar a l'usuari.
3. Les dades s'envien a l'usuari.
4. L'administrador continua dins de l'aplicació.

6.5.4 Cas d'ús realitzar comanda

Cas d'ús : Realitzar una comanda a partir del que demanen els clients.

Actor : Treballador.

Descripció : El treballador pren nota amb la PDA del que li demanen els clients.

Requeriments: El treballador s'ha d'haver autenticat a l'aplicació.

Curs esdeveniments :

1. El treballador selecciona l'opció Comanda del menú de la PDA.
2. Selecciona la taula a la que esta prenent nota.
3. Escull entre els diferents plats, i es van introduint en els grups de "primers", "segons" i "postres".
4. Finalitza la comanda o la imprimeix.
5. La comanda queda guardada en un fitxer perquè després pugui ser recuperada.
6. El treballador es queda a la pantalla de menú de l'aplicació.

6.5.5 Cas d'ús imprimir comanda

Cas d'ús : Imprimim una comanda (per a la cuina).

Actor : Treballador.

Descripció : El treballador imprimeix la comanda seleccionada.

Requeriments: Hi ha d'haver alguna comanda realitzada.

Curs esdeveniments :

1. El treballador selecciona la taula.
2. Selecciona l'opció Imprimir comanda.
 - Error, mostrem missatge error i tornem a menú.
3. La comanda s'imprimeix.

6.5.6 Cas d'ús Autenticar-se

Cas d'ús : Autenticació.

Actor : Treballador.

Descripció : L'usuari entra a l'aplicació i s'identifica omplint les dades d'identificació i contrasenya.

Requeriments: ha d'estar al fitxer de base de dades.

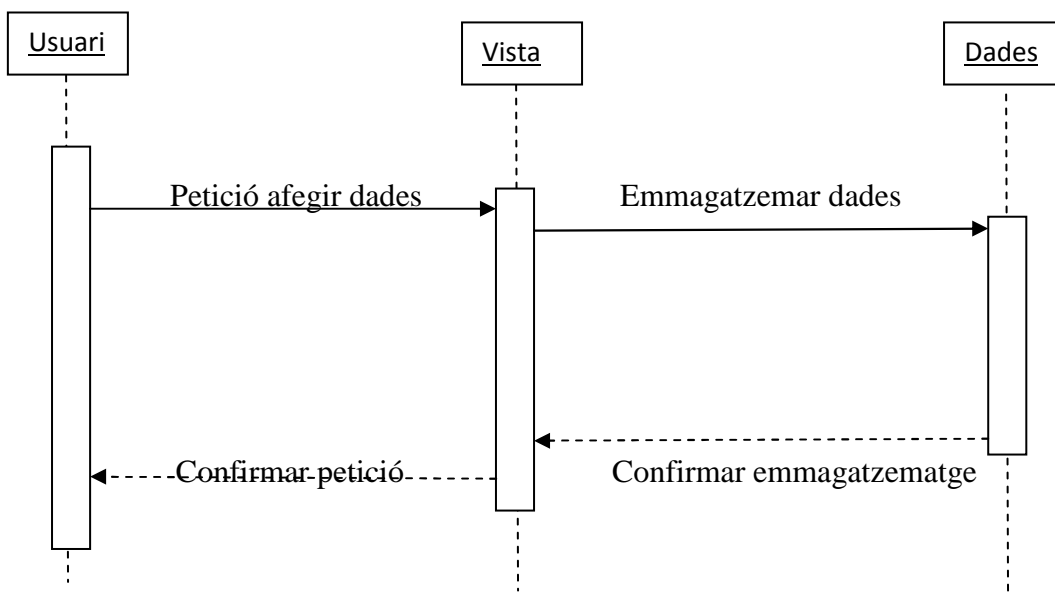
Curs esdeveniments :

1. El treballador omple els camps de Identificador i Contrasenya, selecciona el mode de treball i clica accepta.
2. El sistema comproba les dades amb el fitxer de base de dades
 - Error d'identificació. Se li notifica a l'usuari.
 - Retornem al punt 1.

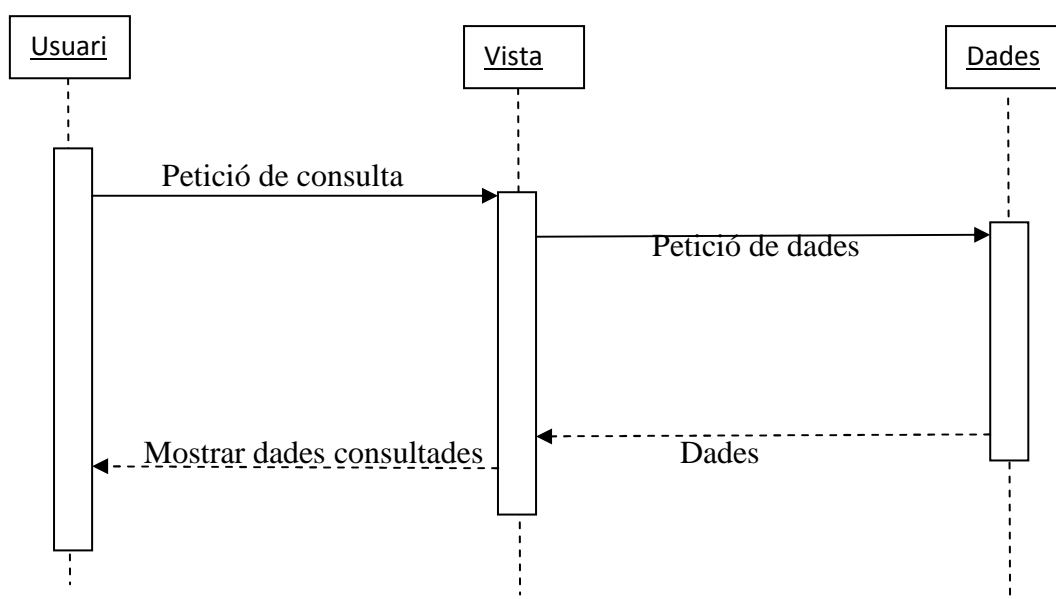
7 Diagrames de Seqüència

Els diagrames de seqüència s'utilitzen per mostrar com interactuen l'usuari i l'administrador amb la vista i el model de dades.

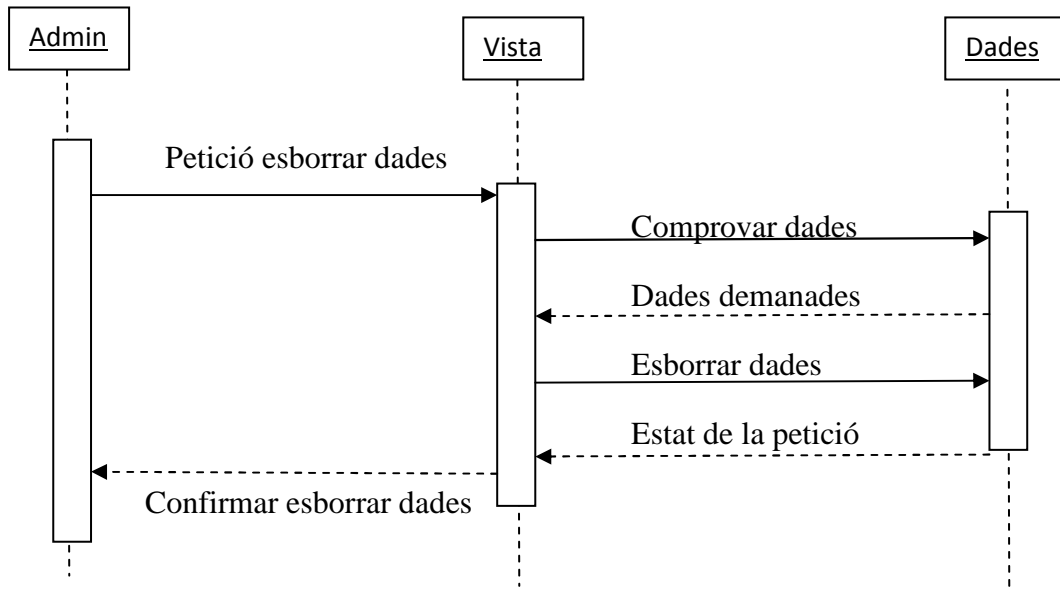
7.4 Registrar dades



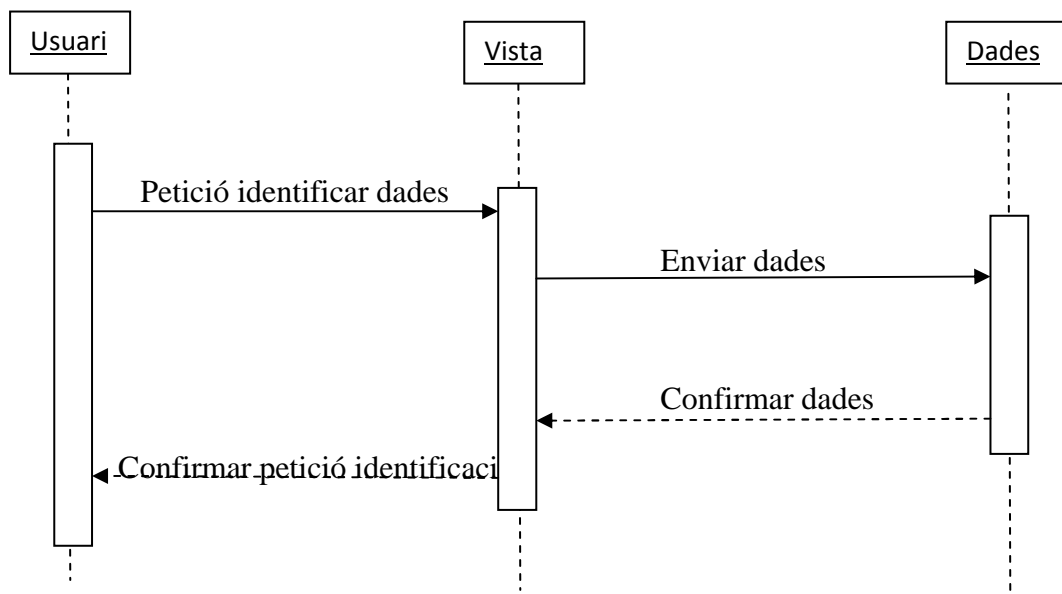
7.5 Consultar dades



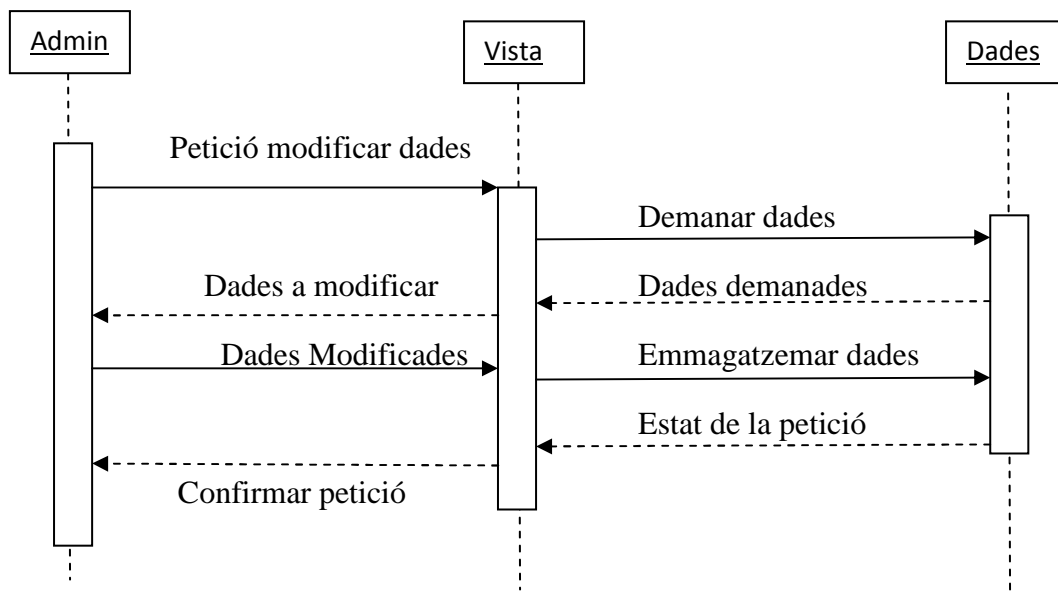
7.6 Esborrar dades



7.7 Identificar



7.8 Modificar dades



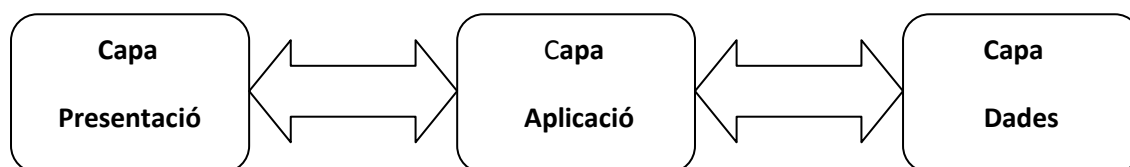
8 Disseny

8.1 Capes

Un dels dissenys per aplicacions Web més utilitzats és el de realitzar una divisió de l'arquitectura de les aplicacions en nivells. El patró arquitectònic orientat a capes és aquella que divideix una aplicació en diferents capes funcionals independents. De manera particular, es pot parlar del patró arquitectònic de 3 capes, que és aquell que es divideix en 3 capes o nivells funcionals. Cadascuna d'aquestes capes s'encarrega d'implementar un grup de funcions independents. Partint del nostre sistema, dividirem l'aplicació en les 3 capes esmentades:

- Capa de Presentació o Interfície. La capa de presentació és l' encarregada de interaccionar amb l'usuari recollint l' informació processada per aquest.
- Capa d'Aplicació, és on es localitza la lògica del negoci. La capa d'aplicació rep la petició de l'usuari mitjançant la capa de presentació.
- Capa de Dades, s'encarrega de donar resposta atenent on tenim la informació de l'aplicació, en el nostre cas una base de dades.

En el següent esquema es mostra l'arquitectura esmentada:



El disseny d'una aplicació, d' acord amb aquesta arquitectura, de capes presenta el gran avantatge de produir codi modular en el que la modificació d'un dels seus components, per exemple l' interfície d'usuari (la forma en la que l'usuari navega per la Web o la forma en la que es presenten les dades) no requereix modificar cap dels elements de l'altre capa.

8.1.1 Capa de Presentació

El nivell de presentació s'encarrega de mostrar les dades a l'usuari. Ha de contenir funcions que mostrin l' informació al usuari, segons l' interfície que oferim.

La totalitat de funcions que permeten accedir a aquest nivell implementen l' interfície entre el nivell de presentació i el nivell d'aplicació.

En aquesta capa, encara que es mostren pantalles d' inici de sessió amb contrasenya, la capa no és responsable de l'autenticació i de l'autorització, ja què, aquestes funcions són pròpies de la capa d'aplicació.

En aquesta capa es generen tots els formularis i capes Web dinàmiques amb l'ajuda de la capa d'aplicació, però que l'usuari no veu, ja que, és totalment transparent per ell, ja que no l'interessa saber la quantitat de capes que hi ha darrera.

8.1.2 Capa d'Aplicació

De manera general, aquest nivell ha d'estar format per crides al nivell de dades i crides al nivell de presentació, a més de l'implementació dels algorismes adients depenent de l'acció del usuari.

La funcionalitat principal de aquesta capa es pot dividir en tres parts:

- Obtenir i enviar les dades a la capa de base de dades.
- Obtenir i rebre les dades de la capa de presentació
- Realitzar els càlculs i manipulacions de dades necessàries

La capa d'aplicació pren les dades que obté de la base de dades i les utilitza segons les necessitats del usuari, tal i com es defineix a la capa de presentació.

Aquesta capa també pren la informació subministrada per l'usuari mitjançant la capa de presentació i la formatarà segons les necessitats de la base de dades pel seu emmagatzematge.

La capa d'aplicació normalment s'utilitza per definir els límits de les transaccions per la creació, lectura, actualització i eliminació d'informació. Un altre funció és la de configurar i enviar una transacció SQL a la capa de base de dades i després, quan la capa de base de dades completa la transacció, rebre la resposta generada per aquesta capa.

8.1.3 Capa de dades

El nivell de dades és l'encarregat de rebre les entrades del usuari, i de cridar a les funcions d'accés a les dades, realitzar el procediment o càlcul demanat i tornar l'informació al nivell d'aplicació per que siguin visualitzades per l'usuari.

La base de dades d'aquesta capa normalment es carrega i s'accedeix des d'un ordinador propi, servidor de base de dades. En sistemes Web de major grandària tenen sovint una còpia reflexa instal·lada en un altre equip per ser utilitzada en cas de problemes amb la còpia mestra de la base de dades o amb el propi servidor

8.2 Disseny de les dades

Especificarem un model de dades que representi els aspectes estàtics i dinàmics del model de domini de la nostra solució. El model de dades que utilitzarem serà el model conceptual perquè ens sembla el més òptim per definir la nostra solució.

8.2.2 Entitats del model de dades

Descripció de les diferents entitats del nostre model.

- Entitat Administrador

Administrador		
Nom	String	Nom de l'administrador
Cognoms	String	Cognoms de l'administrador

- Entitat usuari

Usuari		
Id_usuari	Alfanumèric	Codi d'usuari
Password	Alfanumèric	Contrasenya de l'usuari
Mail	Alfanumèric	Mail de l'usuari

- Entitat Plats

Plats		
Id_Plats	Alfanumèric	Codi de plat
Descripció	String	Descripció del plat
Preu	Double	Import del plat
Id_Menu	Alfanumèric	Codi del menú al que esta associat el plat

- **Entitat Menú**

Menú		
Id_menu	Alfanumèric	Codi de menú
Descripció	String	Descripció del menú
Preu	Double	Import del menú

- **Entitat Taula**

Taula		
Id_Taula	Alfanumèric	Codi de taula
Cadires	Integer	Número de cadires

- **Entitat Reserva**

Usuari		
Id_reserva	Alfanumèric	Codi de reserva
Id_menu	Alfanumèric	Codi del menú associat a la reserva
Id_taula	Alfanumèric	Codi de la taula associada a la reserva
Id_usuari	Alfanumèric	Codi de l'usuari que fa la reserva
Comensals	Integer	Número de persones
Data	Date	Data de la reserva

- **Entitat Proveïdor**

Usuari		
Id_Proveïdor	Alfanumèric	Codi del proveïdor
Nom	Alfanumèric	Nom del proveïdor
Adreca	Alfanumèric	Adreça
Telèfon	Alfanumèric	Telèfon
Web	Alfanumèric	Web
Email	Alfanumèric	Correu electrònic

- **Entitat Calendari**

Usuari		
Número	Integer	Identificador del mes
Mes	Alfanumèric	Nom del mes
1	Integer	Dia del mes (1-normal, 2-festiu)
2	Integer	Dia del mes (1-normal, 2-festiu)
...	Integer	Dia del mes (1-normal, 2-festiu)...

- **Entitat Fotografia**

Usuari		
Id_Foto	Alfanumèric	Identificador de la fotografia
Ruta	Alfanumèric	Director on es troba la fotografia
Tipus_plat	Integer	Tipus de plat(1-primer, 2-segon, 3-postre)
Id_Plat	Alfanumèric	Codi del plat associat a la foto

- **Entitat Personal**

Usuari		
Nom	Alfanumèric	Identificador del treballador
Carrec	Alfanumèric	Càrrec que ocupa
Adreca	Alfanumèric	Adreça del treballador
Telèfon	Alfanumèric	Telèfon
Data_Alta	Datetime	Data d'entrada a la empresa
Data_Baixa	Datetime	Si ha plegat, la data de baixa
Sou	Money	Sou del treballador

8.2.3 Relacions del model de dades

Dona d'alta

- Un Administrador pot donar d'alta tants plats com vulgui.
- Tots els plats són donats d'alta per un administrador.

Realitza

- Un administrador realitza tants menús com vulgui.
- Tots els menús són realitzats per un administrador.

Conté

- Varis menús contenen varis plats.

Té

- Una reserva té 1 taula o varies.
- Varies taules estan en una reserva.
- Un plat té una o més fotografies.

Fa

- Un Usuari fa varies reserves.

Té

- Una reserva té un usuari.

Confecciona

- Un administrador confecciona varies taules.
- Les taules son confeccionades per un administrador.

Escull

- Varis usuaris escullen 1 menú.
- Varis menús son escollits per un usuari.

Conté

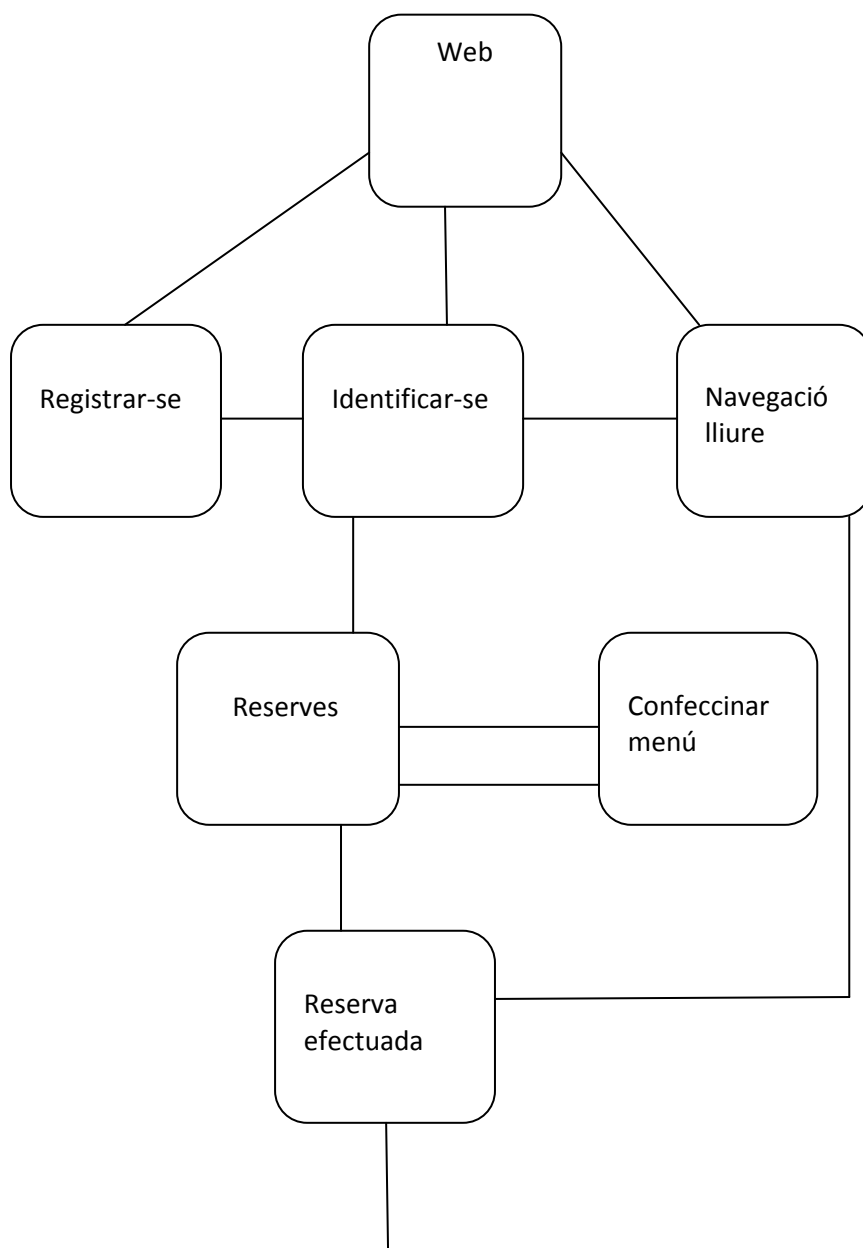
- Varies reserva contenen un menú.
- Un menú esta en varies reserves.

Gestiona

- Un administrador gestiona els proveïdor.
- Un administrador gestiona el calendari.
- Un administrador gestiona les fotografies.

8.3 Disseny de Navegació

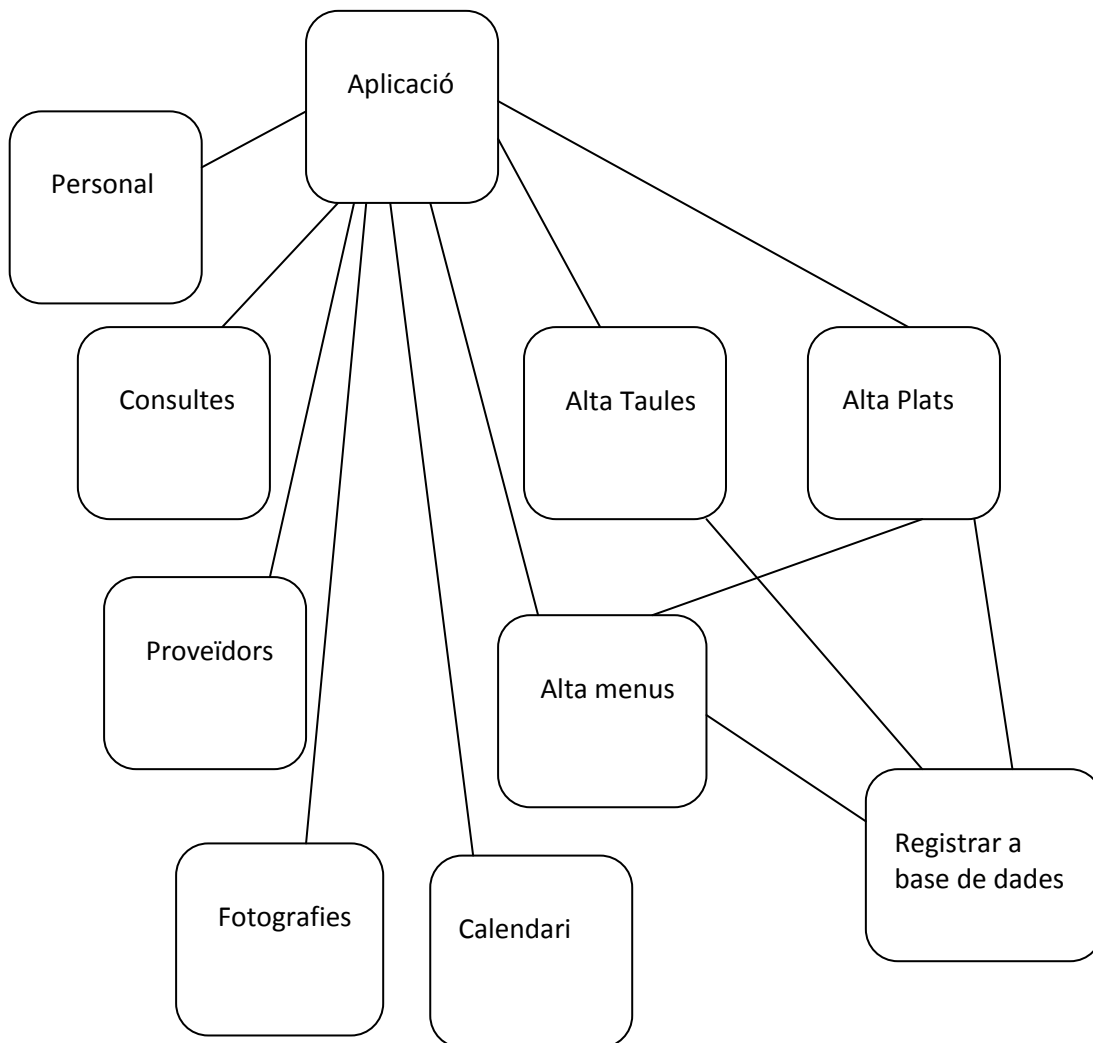
- Navegació usuari



- Descripció Navegació Usuari

L'usuari entra a la web , si és la primera vegada te les opcions de navegar lliurement per la web o registrar-se, si ja s'ha registrat pot identificar-se. Un cop identificat pot seguir navegant lliurement per la web i se li obren altres opcions com la de realitzar les reserves. Quan realitza una reserva pot simplement efectuar la reserva o pot confeccionar-se el menú, llavors realitzarà una reserva amb menú. Al final, la reserva quedarà efectuada. Sempre podrà cancel·lar la reserva mentre la està realitzant. Però un cop realitzada ja no es podrà modificar ni cancel·lar des de la web.

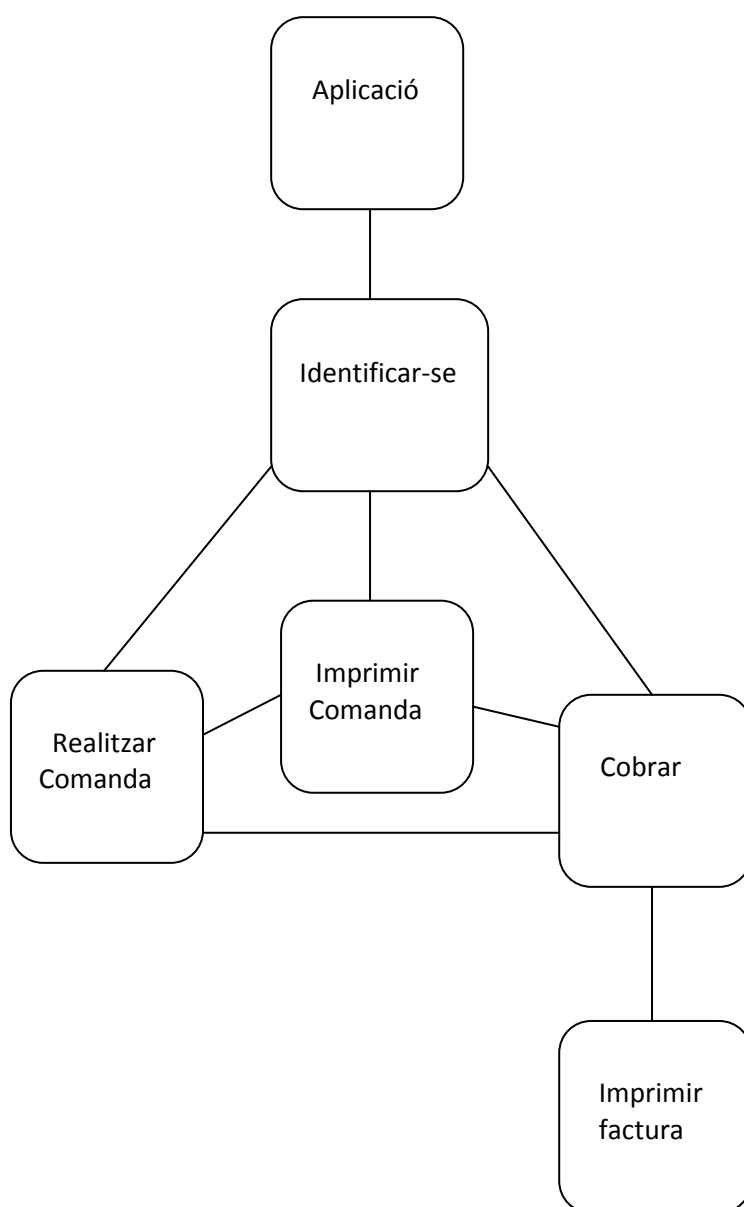
- Navegació administrador



- Descripció Navegació administrador

Des del programa de gestió, es té accés directe a la base de dades i es on donem d'alta i modifiquem el que necessitem. Podem dona d'alta el personal i els proveïdors, confeccionar el calendari laboral i realitzar les consultes que tinguem operatives. Però per introduir els altres elements hem de seguir un ordre i tenir en compte les limitacions. Primer hauríem d'introduir el Plats, per poder introduir els menús i després hauríem d'introduir les taules per a poder realitzar les reserves. Per poder introduir les fotografies, primer hem d'haver introduït els plats. Tenint en compte tots aquests factors podem moure'ns lliurement per tot el programa de gestió.

- Navegació Aplicació PDA



- **Descripció Navegació Aplicació PDA**

Primerament el treballador s'haurà d'identificar amb un nom i una contrasenya, Un cop identificat té tres opcions, realitzar comanda, Imprimir comanda o cobrar. Per poder imprimir o cobrar una comanda primer l'haurà de realitzar, si la comanda ja ha estat realitzada pot anar directament a qualsevol d'aquestes dues opcions. Podria cobrar la comanda sense necessitat d'imprimir-la. Un cop cobrada la comanda, s'imprimirà la factura per al client.

9 Conclusions

L'experiència que ha suposat la realització d'aquest projecte ha estat totalment enriquidora, en molts aspectes ha suposat l'anàlisi i l'estudi de tecnologies desconegudes per a mi.

L'intent d'innovar en un camp tant explotat com es el de la web, crec que és complicat per tant s'ha intentat realitzar un projecte que sigui senzill i que alhora permeti amb una sola solució, resoldre totes les mancances que tinguem. La dificultat radica sobretot en aconseguir lligar entre si els tres móns sobre els que hem treballat, el món de la web, el de les aplicacions de sobretaula (Windows XP) i el de la PDA.

Pel que fa a la tecnologia, s'ha treballat amb Flex que era una tecnologia nova per a mi i per tant ha suposat una dificultat afegida. Pel que fa a les solucions de sobretaula i de PDA s'ha treballat amb C# .NET, però s'ha de tenir en compte que moltes opcions que se'ns permeten quan treballem sobre Windows XP, no estan disponibles quan treballem sobre Windows Mobile. Per tant algunes coses que s'haurien pogut programar de la mateixa manera, s'han hagut d'introduir canvis al canviar de plataforma.

Hi ha hagut coses concretes de l'aplicació de PDA que ens ha suposat utilitzar DLL's del sistema i externes (per exemple la de zebra d'impressió), per poder realitzar el que volíem.

Finalment, només afegiria que s'ha aconseguit una aplicació senzilla i bastant simple en alguns aspectes, però que podria ser una solució econòmica i que cobris en gran part les necessitats del restaurant. Evidentment es podrien afegir millores...

10 Bibliografia

- David Gassner. Adobe Flex 3 Bible. Wiley publishing inc.
ISBN : 978-0-470-28764-4
- http://gsraj.tripod.com/dotnet/webservices/webservice_server.html
Web Services and the .net framework. Gopalan Suresh Raj.
- Labs.adobe.com
- <http://blog.flexexamples.com/>
- http://fleksray.org/Flex_skin.html
Adobe Flex skins and themes.
- <http://kanuwadhwa.wordpress.com/examples/>
Sharp Mind one more blog for people who like to work in flex
- www.Gencat.cat
Traductor.gencat.cat
- es.Wikipedia.org
Enciclopèdia de consulta on-line.
- <http://www.microsoft.com/SqlServer/2005/en/us/express.aspx>
Lloc web del Microsoft SQL Express 2005, des d'on descarregar i veure les característiques del producte.
- [http://msdn.microsoft.com/es-es/library/ms171938\(SQL.90\).aspx](http://msdn.microsoft.com/es-es/library/ms171938(SQL.90).aspx)
Lloc web on hi ha informació sobre el SQL Express compact Mobile edition.

Annex

Descripció del Webservice Restaurant 2.0

Decembre 2009

SUMARI

1. INTRODUCCIÓ

1.1. Objectiu del document

1.2. Variables d'entorn

2. DESCRIPCIÓ DEL SERVEI

2.1. Protocol de comunicació entre el Client del servei i el Servidor

2.2. Aspectes de connexió del servei

2.3. Serveis del webservice

2.3.1. Funcions

2.3.2. Camps del missatge

2.3.3. Missatges de resposta

2.3.4. Missatge d'exemple

3. SEGURETAT

3.1. Autenticació

4. CÓDI DEL SERVEI

1. Introducció

Aquest document s'inclou dins el projecte Restaurant 2.0.

1.1. Objectiu del document

L'objectiu del document es descriure en detall l'ús del servei per a realitzar les diverses consultes a la base de dades des de la web i també la utilització de les diferents funcions per a actualitzar, modificar, inserir... registres a la base de dades ja sigui des de l'aplicació web o des de l'aplicació Windows.

Aquest servei esta basat en tecnologies web service c# .net (ASP.net) mitjançant HTTP.

1.2. Variables d'entorn

En aquest apartat es defineixen els valors de les variables d'entorn dels sistemes que s'utilitzen per al funcionament del servei.

Variable	Descripció	Valor
SERVIDOR	Direcció IP o <i>Hostname</i> del servidor on es troba desplegat el servei.	Local host
PORT	Port TCP del servidor on es troba desplegat el servei.	2358
Nom del usuari	Nom d'usuari d'autenticació HTTP per accedir al servei.	
Contrasenya	Contrasenya d'autenticació HTTP per accedir al servei.	

Taula 1 Variables d'entorn

2. Descripció del servei

En els següents punts es descriu com s'ha d'utilitzar el servei ofert de forma correcta.

2.1. Protocol de comunicació entre el Client del servei i el Servidor

Per minimitzar possibles errors de comunicació degut a problemes de connexió entre el client i el servidor s'ha establert una funció que ens retornarà "true" si te connexió amb la base de dades i "false" en cas que no pugui.

Aquesta funció en principi serà interna del web service i no podrà ser cridada des de cap aplicació (web, Windows), però cada cop que es demani alguna cosa al web service, aquest cridarà la funció de connexió i així comprovarà el correcte accés a la BD.

2.2. Aspectes de connexió del servei

Per invocar el servei que es defineix en aquest document fa falta utilitzar la següent direcció, coneguda com *Target End Point Reference*:

URL: <http://SERVIDOR:PORT/WebSite2>

Els valors del camps SERVIDOR I PORT (a falta de ser publicats) es poden consultar en l'apartat de variables d'entorn ([1.3](#)).

2.3. Serveis del webservice

Aquests serveis permeten la interacció amb la Base de Dades. Aquesta interacció es realitza mitjançant les funcions que especificarem en els següents apartats.

2.3.1. Funcions/Operacions

- ValidarUsuari(usuari, password)
- RegistrarUsuari(usuari, password)
- Reservar(usuari, comensals, data, menú)
- InsertarReserva(usuari, comensals, data, menú)
- FerMenu(plats, id_menu)
- TotPlats()
- Menu()
- Plats()

2.3.2. Camps de les funcions

Per a invocar les funcions es necessari passar una sèrie de paràmetres que dependran del tipus de funció que vulguem executar, per exemple si demanem una validació d'usuari amb ValidarUsuari, li haurem de passar l'usuari i el password, perquè el Web Service o verifiqui amb la base de dades i ens retorni un "true" o "false" depenent de la correctesa o no dels paràmetres que li hem passat.

2.3.3. Missatges de resposta

Rebrem diferents respostes segons el que demanem tindrà una estructura o una altre.

3. Seguretat

Aquí s'explica com utilitzar el sistema de seguretat del servei per accedir a ell i utilitzar-lo.

3.1. Autenticación

En aquest servei no es necessari cap mena d'autenticació, contestem a qualsevol petició que estigui ben realitzada.

4. CODI del servei

El Codi del servei es mostra a continuació. (pot variar del contingut final)

```
using System;
using System.Web;
using System.Web.Services;
using System.Web.Services.Protocols;
using System.Data;
using System.Data.OleDb;
using System.Collections.Generic;
using System.IO;

[WebService(Namespace = "http://tempuri.org/")]
[WebServiceBinding(ConformsTo = WsiProfiles.BasicProfile1_1)]
public class Service : System.Web.Services.WebService
{
    public string SQL_BaseDatos = "Restaurant";
    public string SQL_Servidor = "XAVI\\SQLEXPRESS";
    public string ComputerName = "XAVI";
    public OleDbConnection Con;
    public int maxcomensals = 50;

    public Service ()
    {
        //Eliminar la marca de comentario de la línea siguiente si utiliza los
        //componentes diseñados
        //InitializeComponent();
    }
}
```

Annex - Webservice

```

    /// <summary>
    /// Funció que valida l'usuari contra la BD
    /// </summary>
    /// <param name="usuari">Usuari a validar</param>
    /// <param name="pass">Password de l'usuari</param>
    /// <returns>True- si l'usuari i el password són correctes. False - si no ho
son</returns>
    [WebMethod]
    public bool ValidarUsuari(string usuari, string pass)
    {
        OleDbDataAdapter oda;
        DataSet qry;
        string SQL;
        bool Ret = false;

        if (Conexió())
        {
            SQL =
                "SELECT * FROM" +
                " Usuaris" +
                " WHERE ID_USUAR = '" + usuari + "' " +
                " AND PASSWORD = '" + pass + "'";

            oda = new OleDbDataAdapter(SQL, Con);
            qry = new DataSet();
            oda.Fill(qry, "Usuaris");

            if (qry.Tables["Usuaris"].Rows.Count > 0)
            {
                Ret = true;
            }
        }
        return Ret;
    }

    /// <summary>
    /// Creem la connexió amb la BD
    /// </summary>
    /// <returns>True si es pot connectar. False si no.</returns>
    private bool Conexió()
    {
        bool Ret = true;

        try
        {
            //Creem la connexió
            Con = new OleDbConnection();

            Con.ConnectionString =
                "Provider=SQLOLEDB.1;" +
                "Integrated Security=SSPI;" +
                "Persist Security Info=False;" +
                "Initial Catalog=" + SQL_BaseDatos + ";" +
                "Data Source=" + SQL_Servidor + ";" +
                "Use Procedure for Prepare=1;" +
                "Auto Translate=True;" +
                "Packet Size=4096;" +
                "Workstation ID=" + ComputerName + ";" +
                "Use Encryption for Data=False;" +
                "Tag with column collation when possible=False";
            Con.Open();
        }
        catch (Exception Err)
        {
            Ret = false;
        }
    }

```



```

    return Ret;
}

/// <summary>
/// Registrar l'usuari a la BD
/// </summary>
/// <param name="usuari">Nom de l'usuari</param>
/// <param name="pass">Password de l'usuari</param>
/// <returns>True si es registra correctament. False si no pot.</returns>
[WebMethod]
public bool RegistrarUsuari(string usuari, string pass, string mail)
{
    OleDbCommand cmd;
    string SQL;
    bool Ret = false;

    if (Conexió())
    {
        SQL =
            "INSERT INTO USUARIS " +
            "(ID_USUAR, PASSWORD, MAIL) VALUES " +
            "('" + usuari + "', '" + pass + "', '" + mail + "')";
        cmd = new OleDbCommand(SQL, Con);
        cmd.ExecuteNonQuery();

        Ret = true;
    }
    return Ret;
}

/// <summary>
/// Funció que comprova que es pugui fer la reserva i crida a la funció de registre
/// </summary>
/// <param name="usuari">Usuari (registrat o no) que fa la reserva </param>
/// <param name="comensals">Número de persones que vindran</param>
/// <param name="Data">Día de la reserva</param>
/// <param name="menu">Menú escollit (serà creat per l'usuari o un dels existents)</param>
/// <returns>String indicant si s'ha realitzat o no la reserva</returns>
[WebMethod]
public string reservar(string usuari, string comensals, string Data, string menu)
{
    OleDbDataAdapter oda;
    DataSet qry;
    string SQL;
    string RET = "";

    if (Convert.ToInt32(comensals) > maxcomensals)
    {
        RET = "El nombre màxim de comensals es de: " + maxcomensals.ToString();
    }
    else
    {
        if (Conexió())
        {
            SQL =
                "SELECT SUM(NUMCOMENSALS) AS NCOM " +
                "FROM RESERVES " +
                "WHERE DATA = '" + Convert.ToDateTime(Data) + "'";
            oda = new OleDbDataAdapter(SQL, Con);
            qry = new DataSet();

            oda.Fill(qry, "RESERVES");

```

Annex - Webservice

```

    if (qry.Tables["RESERVES"].Rows.Count > 1)
    {
        int suma = Convert.ToInt32(qry.Tables["RESERVES"].Rows[0]["NCOM"]);
        if ((suma + Convert.ToInt32(comensals)) > maxcomensals)
        {
            RET = "No es possible fer la reserva, masses comensals";
        }
        else
        {
            if (InsertarReserva(usuari, Data, comensals, menu) == "OK")
            {
                RET = "Reserva efectuada pel dia: " + Data + " per a " + comensals
+ " persones";
            }
            else
            {
                RET = "No ha estat possible efectuar la reserva. Truqi al
xxxxxxxxxxx. Gràcies.";
            }
        }
    }
    else
    {
        if (InsertarReserva(usuari, Data, comensals, menu) == "OK")
        {
            RET = "Reserva efectuada pel dia: " + Data + " per a " + comensals +
" persones";
        }
        else
        {
            RET = "No ha estat possible efectuar la reserva. Truqi al xxxxxxxxxxxx.
Gràcies.";
        }
    }
}
}
return RET;
}

/// <summary>
/// Funció que inserta la reserva a la base de dades
/// </summary>
/// <param name="Usuari">Usuari</param>
/// <param name="Data">Data</param>
/// <param name="comensals">Número persones</param>
/// <param name="menu">Menú</param>
/// <returns>string OK si ha anat tot be o cadena d'error</returns>
public string InsertarReserva(string Usuari, string Data, string comensals,
string menu)
{
    OleDbDataAdapter oda;
    OleDbCommand cmd;
    DataSet qry;
    string SQL;
    string RET;
    string id_taula;

    try
    {
        SQL =
            "SELECT ID_TAULES " +
            "FROM TAULES " +
            "WHERE N_CADIRES >= ' " + Convert.ToInt32(comensals) + "' " +
            "AND ID_TAULES <> " +
            "(SELECT ID_TAULES " +
            "FROM RESERVES " +
            "WHERE DATA = ' " + Convert.ToDateTime(Data) + "')";
    }
}

```

```

oda = new OleDbDataAdapter(SQL, Con);
qry = new DataSet();
oda.Fill(qry, "TAULES");

if (qry.Tables["TAULES"].Rows.Count > 0)
{
    id_taula = qry.Tables["TAULES"].Rows[0]["ID_TAULES"].ToString();
}
else
{
    id_taula = "000";
}

string id_reserva = comensals + id_taula + Data;

SQL =
    "INSERT INTO RESERVES " +
    "(ID_RESERVA, ID_TAULA, ID_USUARI, ID_MENU, DATA, NUMCOMENSALS)" +
    "VALUES ('" + id_reserva + "', '" + id_taula + "', '" + Usuari + "', '" +
menu + "', '" + Convert.ToDateTime(Data) + "', '" + Convert.ToInt32(comensals) +
"')";
cmd = new OleDbCommand(SQL, Con);
cmd.ExecuteNonQuery();
RET = "OK";
}
catch (Exception Err)
{
    RET = "Error insertant la reserva a la base de dades." + Err.Message;
}
return RET;
}

/// <summary>
/// Funció que permet realitzar el teu menú personal
/// </summary>
/// <param name="plats">Llista amb els plats que componen aquest menu</param>
/// <param name="id_menu">Nom que li donem al menu</param>
/// <returns>String dient si s'ha pogut confeccionar el menú o no</returns>
[WebMethod]
public string fermenu(List<string> plats, string id_menu)
{
    OleDbDataAdapter oda;
    OleDbCommand cmd;
    DataSet qry;
    string SQL;
    string RET;
    double PreuMenu = 0;

    try
    {
        if (Conexió())
        {
            for (int i = 0; i < plats.Count; i++)
            {
                SQL =
                    "SELECT PR_PLAT " +
                    "FROM PLATS " +
                    "WHERE ID_PLAT = '" + plats[i] + "'";
                oda = new OleDbDataAdapter(SQL, Con);
                qry = new DataSet();

                oda.Fill(qry, "PLATS");

                PreuMenu += Convert.ToDouble(qry.Tables["PLATS"].Rows[0]["PR_PLAT"]);
            }
        }
    }
}

```

Annex - Webservice

```

        SQL =
            "INSERT INTO MENU " +
            "(ID_MENU, DS_MENU, PR_MENU) " +
            "VALUES ('" + id_menu + "', 'MENU CREAT PER USUARI', '" + PreuMenu +
            "')";
        cmd = new OleDbCommand(SQL, Con);
        cmd.ExecuteNonQuery();
        RET = "Menu generat correctament";
    }
    else
    {
        RET = "Impossible conectar amb la base de dades.";
    }
}
catch (Exception Err)
{
    RET = "Error generant el menu, el menu no s'ha guardat" + Err.Message;
}
return RET;
}

/// <summary>
/// Funció que retorna la taula amb els plats de la carta
/// </summary>
/// <returns>Taula dels plats</returns>
[WebMethod]
public DataTable TotPlats()
{
    OleDbDataAdapter oda;
    DataSet qry = new DataSet();
    string SQL;

    try
    {
        if (Conexió())
        {
            SQL =
                "SELECT ID_PLAT, DS_PLAT, PR_PLAT FROM " +
                "PLATS";

            oda = new OleDbDataAdapter(SQL, Con);

            oda.Fill(qry, "PLATS");

        }
    }
    catch (Exception Err)
    {
        string x = Err.Message;
    }
    return qry.Tables[0];
}

/// <summary>
/// Funció que retorna la taula amb els menús del restaurant
/// </summary>
/// <returns>Taula dels menús</returns>
[WebMethod]
public DataTable TotMenus()
{
    OleDbDataAdapter oda;
    DataSet qry = new DataSet();
    string SQL;

    try
    {
        if (Conexió())

```

```

    {
        SQL =
            "SELECT * FROM " +
            "MENU";

        oda = new OleDbDataAdapter(SQL, Con);
        oda.Fill(qry, "MENU");
    }
}
catch (Exception Err)
{
    string x = Err.Message;
}
return qry.Tables[0];
}

/// <summary>
/// Funció que retorna els menús del restaurant
/// </summary>
/// <returns> menús</returns>
[WebMethod]
public List<string> Menu()
{
    OleDbDataAdapter oda;
    DataSet qry = new DataSet();
    string SQL;
    List<string> LMenus = new List<string>();

    try
    {
        if (Conexió())
        {
            SQL =
                "SELECT DS_MENU FROM " +
                "MENU";

            oda = new OleDbDataAdapter(SQL, Con);

            oda.Fill(qry, "MENU");
            if (qry.Tables["MENU"].Rows.Count > 0)
            {
                for (int i= 0; i< qry.Tables["MENU"].Rows.Count;i++)
                {
                    LMenus.Add(qry.Tables["MENU"].Rows[i][0].ToString());
                }
            }
        }
    }
    catch (Exception Err)
    {
        string x = Err.Message;
    }
    //return qry.Tables[0];
    return LMenus;
}

#region WebservicePDA

/// <summary>
/// Funció que retorna una llista amb els plats i els preus d'aquests
/// </summary>
/// <returns>Plats i preu</returns>
[WebMethod]
public List<String> Plats()
{
    OleDbDataAdapter oda;

```

Annex - Webservice

```
DataSet qry = new DataSet();
string SQL;
List<string> lista = new List<string>();

try
{
    if (Conexió())
    {
        SQL =
            "SELECT DS_PLAT, PR_PLAT FROM " +
            "PLATS";
        oda = new OleDbDataAdapter(SQL, Con);
        oda.Fill(qry, "plats");

        if (qry.Tables["plats"].Rows.Count > 0)
        {
            for (int i = 0; i <= qry.Tables["plats"].Rows.Count; i++)
            {
                lista[i] = qry.Tables["plats"].Rows[i][0].ToString();
                lista[i] = qry.Tables["plats"].Rows[i][1].ToString();
            }
        }
    }
}
catch (Exception Err)
{
    string x = Err.Message;
}
return lista;
}
#endregion WebservicePDA
}
```

