



INDEX

1. OBJECTIUS	2
2. DESENVOLUPAMENT	3
2.1 <i>Comunicació dels dispositius</i>	<i>5</i>
2.2 <i>Aprofitament del codi</i>	<i>5</i>
3. CONCLUSIONS.....	7



1. Objectius

Existeixen, des de telèfons mòbils amb moltes funcionalitats informàtiques com agenda electrònica o possibilitat de fer conferències en directe, fins a petits ordenadors que caben en el palmell de la mà anomenats PDA. Aquests son els més utilitzats avui en dia per demanar comandes en el mon de l'hostaleria i transport de productes en general. També els podem utilitzar com si fos un bloc de notes de paper o fins i tot, els podem utilitzar com a GPS o per enviar i rebre correu, entre moltes altres funcionalitats.

Aquesta evolució tecnològica a aportat molts avenços sobretot cares als negocis i a la obtenció de la màxima comoditat i rendiment del mateix. Això ha provocat un ràpid desenvolupament de software per aquests dispositius portàtils.

Tot això pot ésser totalment aprofitable en el mon de la restauració. Sobretot en restaurants de gran envergadura amb terrasses lluny de les barres i llocs de preparació dels productes els cambrers on han de realitzar varis viatges abans de servir les taules. En aquests casos un dispositiu portàtil com una PDA seria de gran ajuda a la hora de recollir les comandes , optimitzant així el temps de resposta.

Aquesta aplicació s'emmarca en el mon de la restauració. Ja existeixen aplicacions d'aquest tipus encara que crec convenient plantejar-ne una per formalitzar completament la gestió entre tauler i cuina de forma més còmoda ràpida i intuïtiva.

El projecte consisteix en un sistema informatitzat per restaurants centralitzat en un servidor que permet recollir comandes als clients a través de PDA's i de pantalles tàctils situades a la barra per enviar-les als corresponents llocs d'elaboració:

- Terminal de la cuina on s'elaboraran tots els menjars.
- Terminal de la bodega on es serviran els vins i caves.
- Terminal barra on es serviran els licors, refrescos (entre d'altres productes totalment configurables).
- Terminal de postres on es reben les línies de comanda relacionades amb els postres.

L'objectiu acadèmic del projecte serà adquirir coneixement a nivell de disseny i gestió de dades, així com la comunicació de varis dispositius en temps real. S'utilitzarà la tecnologia .net, concretament el llenguatge de programació visual c# .net. i com a gestor de bases de dades el mysql 5.0.19. A més, treballaré amb la metodologia UML, utilitzant també varis patrons.

Per fer del sistema un sistema centralitzat en un servidor s'utilitzarà la comunicació per terminal service de windows el qual ens brinda la possibilitat



d'establir varies sessions d'usuaris concurrents. Cada sessió estarà destinada a un cambrer, per tant, hi haurà una sessió per PDA.

2. Desenvolupament

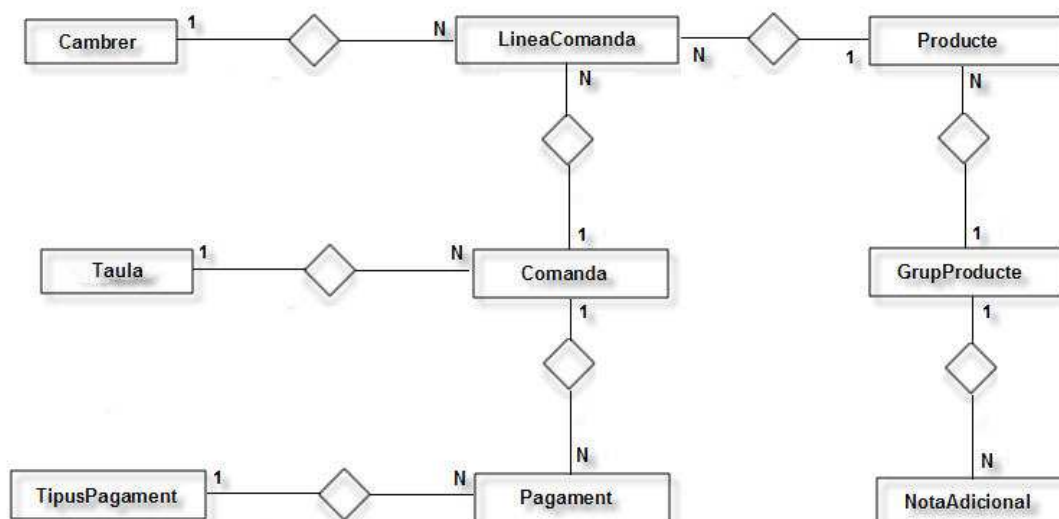
Cal destacar que el que farà d'aquest projecte un projecte diferent i sobretot eficient és la utilització de un motor de persistència anomenat nhibernate utilitzat per comunicar les dades del model relacional de base de dades amb els objectes de la programació orientada a objectes.

Al llarg del projecte es poden diferenciar les tres fases per les que ha passat, encara que el desenvolupament del software en UML és un procés iteratiu i incremental i algunes fases s'han treballat quasi paral·lelament.

1. Anàlisi de requeriments i del sistema: En aquesta fase s'ha estudiat la forma més viable per informatitzar les dades de un restaurant i la manera amb la que es farà tota la gestió de comandes i de tots els altres objectes. A partir d'aquí sorgeixen els primers casos d'ús i classes de dades principals.
2. Disseny del sistema: En aquesta etapa de desenvolupament s'estudia la millor forma amb la que es comunicaran els objectes entre ells, s'estudia la manera en que es disposen els paquets de la aplicació, i s'aportaran solucions de disseny del software amb patrons Gof i GRASP com per exemple el factory method, Expert, Singleton, Façade, high cohesion, low coupling entre molts altres.
3. . Una decisió important ha estat utilitzat un motor de persistència per objectes i model relacional anomenat nhibernate.
4. Implementació: Encara que aquesta etapa es desenvolupa paral·lelament amb el disseny, és una part que requereix molt de temps i que es distingeix totalment de les altres. Un cop ha estat realitzat tota la implementació de les classes inferiors que conformen la base, procedeix a implementar totes les interfícies.

Podríem parlar també de una quarta fase, que seria el refinament i proves del software, però les proves i refinaments les he anat realitzant al mateix temps que el desenvolupament.

Crec que es important veure el model de dades sobre el que s'ha construït l'aplicació, ja que tot volta en torn aquest model que ha anat evolucionant durant tot el projecte. Es mostra a continuació:



Aquest model de dades ha estat representat de la mateixa manera en objectes i en el model relacional. A més cada una d'aquestes classes a estat mapejada en fitxers XML per fer funcionar el motor de persistència utilitzat.

Aquest motor, nhibernate, ha de tenir informació del model de dades per tal de fer la relació entre els objectes i les dades del model relacional de la base de dades. D'aquesta manera es configura aquest framework, que permetrà entre moltes altres coses, accedir i guardar dades de una manera transparent al tipus de base de dades que sigui i deslligar el codi d'accés a dades del codi de la capa de presentació (les interfícies).

A més, aquesta manera de treballar ens dona eficiència en els accesos a dades ja que nhibernate treballa amb memòries caché i permet obtenir dades obtingudes recentment sense necessitat d'accedir a la base de dades.



2.1 Comunicació dels dispositius

Tot el sistema informàtic està centralitzat a un servidor principal via wifi. Aquest servidor es qui contindrà el servidor de la base de dades i el que donarà sessions remotes als dispositius portàtils PDA's.

El fet de utilitzar la comunicació a escriptori remot dona unes certes avantatges respecte la programació de un programa fet amb un llenguatge específic per una pda.

- Si la bateria de la pda s'acaba en un moment donat, no es perd cap dada ja que la sessió queda en l'últim estat que estava abans d'apagar-se. Només falta tornar a connectar-se a la sessió i seguir per allà on ho havíem deixat.
- Descarreguem tota la feina de la pda, cada acció, mètode, càlcul es realitza en el servidor on està connectada remotament. La pda només ens proporciona la interfície.
- Finalment, des de el punt de vista de programació, es un estalvi enorme de temps el fet d'aprofitar en un 75% la interfície creada primerament per una tpv, simplement s'ha de reduir i casi no es toca res de codi.

2.2 Aprofitament del codi

Aquest ha estat un concepte que he utilitzat molt i he tingut en compte al llarg del projecte. El cas més clar, és la interfície destinada a recollir comanda per pantalles tàctils i la interfície per fer el mateix però en una PDA.

Si hagués fet una aplicació amb un llenguatge específic per la pda no podia haver utilitzat res de codi, però d'aquesta manera s'ha utilitzat més de un 90%. Simplement s'ha de canviar l'aspecte gràfic i desglossar tot el que cap en una pantalla en varies de més reduïdes per la PDA.



A continuació un exemple d'aquestes dos interfícies:



Frontal de vendes per pantalla tàctil, tota la informació en una pantalla



Frontal vendes per PDA, la mateixa informació repartida en una seqüència de petites pantalles.



3. Conclusions

Els objectius marcats inicialment han estat assolits satisfactòriament, aconseguint informatitzar per complet el sistema de comandes de restaurants i facilitant la feina a tots els treballadors que disposin d'aquest.

Un dels principals objectius acadèmics era el de saber desenvolupar software en varis llenguatges i tecnologies de programació. Finalment s'ha utilitzat el llenguatge `c#` de la tecnologia `.net` que tant d'èxit està tenint en el món del desenvolupament del software actual.

He aprofitat els recursos d'aquesta tecnologia per tractar el llenguatge XML, destinat sobretot a fitxers de configuració de l'aplicació.

A més, l'ús de `nhibernate`, motor de persistència per programació orientada a objectes i bases de dades relacionals, ha estat una metodologia fortament utilitzada al llarg de tota la implementació, ja que la base del sistema informàtic resideix sobre aquest.

Un altre objectiu principal era el de aprofitar tots els recursos i tècniques apreses durant la formació acadèmica i aplicar-ho en aquest projecte. Em refereixo a la orientació a objectes, tècniques de modelatge i estandardització del software i el model relacional entre molts altres.

Hi ha una fort component de enginyeria del software basat en el modelat de un llenguatge unificat (UML) amb el que es pretén crear un software que segueixi uns estàndards i sobretot, un software el qual pugui ser totalment reutilitzable, ampliable i fàcil de mantenir. Aquests tres conceptes són la base principal sobre la que es fonamenta una bona enginyeria del programari, uns conceptes recalcats al llarg de tots els anys de formació acadèmica i totalment aplicats en aquest projecte.

Per realitzar l'anàlisi del software he utilitzat eines CASE com `Poseidon` la qual ha ajudat molt en la autogeneració de codi de les classes principals.

Amb aquest projecte he après a aplicar a la pràctica els patrons de disseny que ens ofereix el UML. Així doncs he solucionat molts problemes de estandardització i optimització de codi amb patrons `Gof` i `GRASP`.

El fet d'haver treballat utilitzant tots els coneixements adquirits i utilitzant eines i llenguatges que desconeixia m'ha ajudat a fer-me una imatge sobre l'entorn professional en el àmbit informàtic.

La realització d'aquest projecte ha estat una satisfacció personal, degut a que gràcies als coneixements aplicats he pogut materialitzar una idea que tenia en ment des de feia temps. A més el fet de saber-me moure amb moltes tecnologies, llenguatges o tècniques és una cosa que em satisfà com a estudiant i futur professional.