DEVELOPMENT OF AN APPLICATION SERVER FOR POINT OF SALE TERMINAL

INDICE

INTRODUCTION

The realization of this final project will be jointly run between Manuel Ariza Calpe and Omar Peset Gregorio, which will be delivered at the University Vilniaus Gedimino Universitetas Technik (VGTU) in Vilnius (Lithuania) during the ERASMUS scholarship second semester of the academic year 2009/2010.

The director in charge of the project is Dr. Mindaugas VGTU Rybokas, co-director of the same in the UPV is Dr. Juan Carlos Ruiz Garcia

APROACH OF THE PROBLEM

It is intended to improve as far as possible the operation currently being used in a Valencia company to manage sales data that they do. This catering company that has 4 head offices in difference locations around Valencia and in each site has several POS terminals, the number varies depending on the head office. Each POS terminal stored in a database Access all movements relating to sales, such as: code, time of the sale price, units, seller, payment, etc.. All these data are stored in each terminal point sales locally with records in a table in the database. This database contains more tables, but other data relating to internal configuration of software on the POS terminal, the sales data is only stored in a one table.

The problem we are having now is that the company has grown considerably and are not able to view global data, only have access to summaries of "Cash Closing" local to each POS terminal, and a small sales report made by the POS terminal when it is "Cash Closing". In addition it is now impossible to see a summary of global sales, as each database is independent, to make an overall summary must move physically across all head offices and POS terminal collecting individual closures for after add them, all manually.

DESCRIPTION OF THE SOLUTION

Due to the diversification and large volume of data, the company has now almost impossible to manage the information, we will develop as a final project-server software to attempt to solve the possible shortcomings of the system Current.

The software will be installed on a computer and will be responsible for being listen to permanently and concurrently, that when a POS terminal makes any cash transaction closing, or any specific request, the server collects this request, process and be able to save to a global database if it is a cash closing or sending data if it is a personal service.

Once have identified the type of request and stored in the database or data used, the server software will leave a file type Access database global gathering in the same database and share information in different tables, both the sales information from the POS terminals, such as information services offered by the server to the POS terminals.

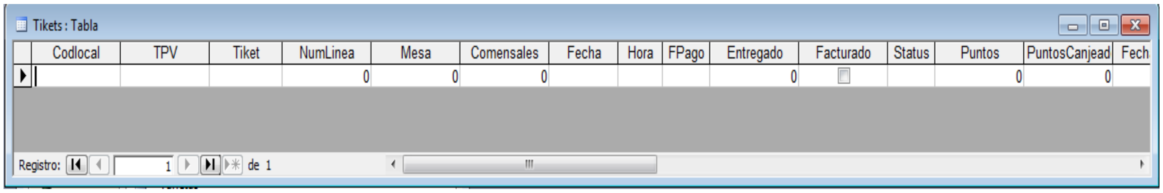
The final objective is that the company can access data, run reports, or any questions they may have, in a fast and comfortable from its head offices and in a real time, without having to make any movement and full automation.

The programming language used to implement the application will be Visual Basic. We chose this language to further enrich our knowledge in the various existing programming languages.

Communication between the different points of sales terminals and the server software will be through TCP sockets, which will be able to securely send online data without being altered in any way communication thus ensuring the integrity of the data.

DETAILS OF THE DATABASE

Just go into detail of the database server because it is what we are going to implement and will be responsible for storing all the data from the various points of sale terminals in each site globally.



Tikets : Tabla

Codlocal	TPV	Tiket	NumLinea	Mesa	Comensales	Fecha	Hora	FPago	Entregado	Facturado	Status	Puntos	PuntosCanjead	Fech
			0	0	0				0	☐		0	0	

Registro: 1 de 1

- Z Tickets:

This table will be the most important of all because here will reflect all sales of the company, it will appear the following fields.

- Code of the local: local identification code.
- POS: Code that identifies each POS within its site.
- Ticket: Code to identify each ticket.
- LineNumber: The number that identifies the various items within the same ticket.
- Table: number that identifies whether the item is sold on a table or bar.
- Table: number that identifies whether the item is sold on a table or bar.
- Date: Date on which the sale was made.
- Time: Time in which the sale was made.
- Payment: number that identifies the payment method.
- Checked: Boolean indicating whether the ticket has been invoiced to which it is associated.
- Status: Text indicating any kind of incident.
- Points: Possible generation of customer points.
- PointsExchanged: the possible points exchanged of the client.

- DateOfExchanged: Date of points exchanged of the client.
- CodCliente: Possible code number of each client.
- ReqArti: Item code required.
- CodCama: Code waiter who makes the sale.
- Description: Possible manual description of the item.
- Department: Department Code that owns the item.
- Fares: Number indicating the fare applied.
- Quantity: Number of items sold.
- PvpArti: price of the item.
- IvaArti: I.V.A applied.
- CosteArti: Cost price of the item.
- NumberZ: Number of closing cash which included the sale.

SHIPPING PROCEDURE AND DATA PROCESSING

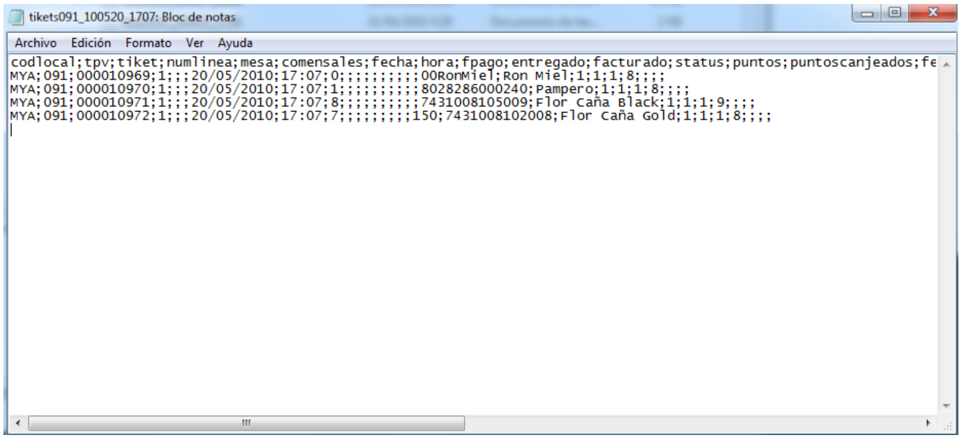
- CUSTOMER PART

When the sales program to the cash closing comes the procedure which makes it possible to unify the central database of all records of all TPV.

This procedure begins by copying all records from table "Tickets" in the local database. These records are placed on the table "TicketsZ" local and in addition to these registers generates a plain text file delimited by semicolons.

This file contains in the first line the names of the fields in the table "Tickets" delimited by semicolons, and the following data lines of the table also delimited by semicolons.

The program sends this file to the server program, it receives the file and interprets it (later we shall see in detail).

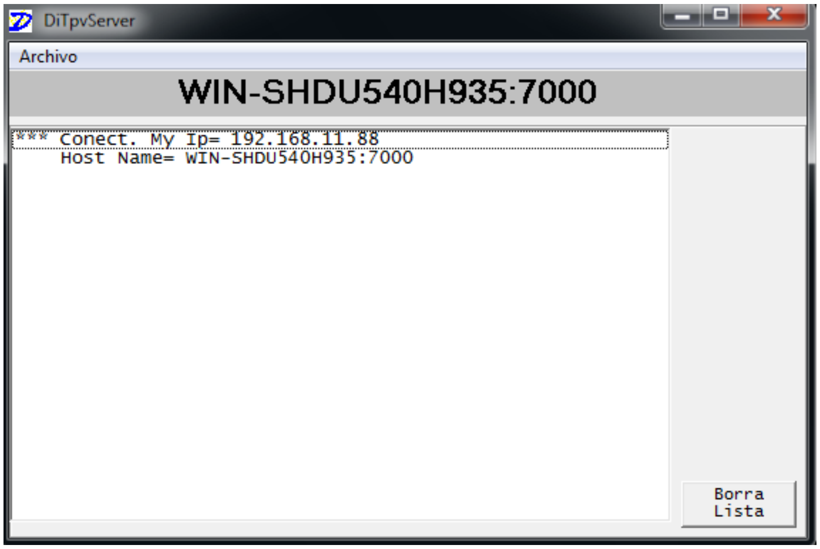

tikets091_100520_1707: Bloc de notas
Archivo Edición Formato Ver Ayuda

```
codlocal;tpv;tiket;numlinea;mesa;comensales;fecha;hora;fpago;entregado;facturado;status;puntos;puntoscanjeados;fe
MYA;091;000010969;1;;;20/05/2010;17:07;0;;;;;;;;;;00RonMiel;Ron Miel;1;1;1;8;;;;
MYA;091;000010970;1;;;20/05/2010;17:07;1;;;;;;;;;;8028286000240;Pampero;1;1;1;8;;;;
MYA;091;000010971;1;;;20/05/2010;17:07;8;;;;;;;;;;7431008105009;Flor Caña Black;1;1;1;9;;;;
MYA;091;000010972;1;;;20/05/2010;17:07;7;;;;;;;;;150;7431008102008;Flor Caña Gold;1;1;1;8;;;;
```

At the end of sending the file is just clear the table "Tickets" to not create duplicates.

- SERVER SIDE

When the program starts it opens port 7000 and let him listen. When this port receives any packet, the program is analyzed by the first 4 characters received.

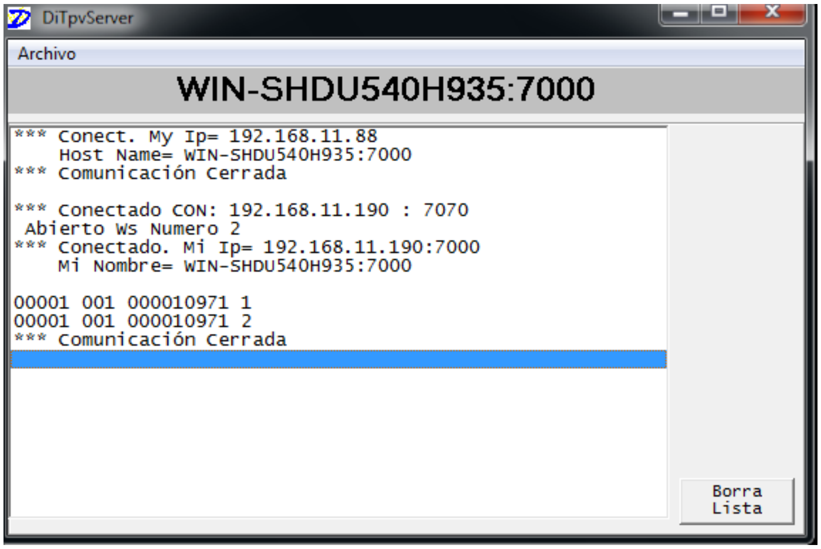


```
*** Conect. My Ip= 192.168.11.88
    Host Name= WIN-SHDU540H935:7000
```

If these caractecteres match "file", then you get the plain text file and start to save in the table "TicketsZ" the server. If ever you receive another text string that is not encoded by the program automatically discards the packet.

As in the first line of text file are the names of the fields in the table the program reads this line and only have to associate the file name field with name in the table. Once the association, and if there has been no problema, begins to insert each line of text file in a row in the table "TiketsZ" the server, until the end of the file.

To provide the user the task of verifying whether the data are recorded in the database, it is painted on the screen of Server-program a short summary of the record is being recorded, so you can assess whether is processing the text file correctly.



```
*** Conect. My Ip= 192.168.11.88
    Host Name= WIN-SHDU540H935:7000
*** Comunicación Cerrada

*** Conectado CON: 192.168.11.190 : 7070
 Abierto Ws Numero 2
*** Conectado. Mi Ip= 192.168.11.190:7000
    Mi Nombre= WIN-SHDU540H935:7000

00001 001 000010971 1
00001 001 000010971 2
*** Comunicación Cerrada
```

If during the process generates some error insertion, also be reflected in the display, for further arrangement.

Once finished reading the file, closes the connection.

CONCLUSION

With the development of this application has achieved the main objective of the project, to automate the sending process sales and manage all information in one global database.