

DISSENY I PROGRAMACIÓ ORIENTADA A OBJECTES  
CURS 2017/2018

# PROJECTE: RESTAURANT

## **Grup C5:**

Marc Llorc  
Paula Rodríguez  
Alex Almansa  
Manel López  
Àlex Vicente

# ÍNDIX

1. Resum de les especificacions del projecte	p. 3
2. Disseny de la interfície gràfica	p. 4
3. Model de disseny	p. 9
3.1 Diagrama de classes	p. 9
3.2 Descripció	p. 11
4. Metodologia de desenvolupament	p. 13
5. Dedicació	p. 14
6. Conclusions	p. 15
7. Bibliografia	p. 16

# 1. RESUM DEL PROJECTE

Es tracta de crear un projecte que generi un *Centralized Restaurant*. Aquest projecte consta de tres grans programes, el programa "Servidor", el programa "Reserva" i el programa "Recepció".

El programa Servidor, gestiona tot el que te a veure amb les peticions del client (validar credencials de cada reserva, proveir una taula a la reserva, generar codi d'accés a la nova reserva, ...). El programa Servidor també emmagatzema les reserves, gestiona les taules del restaurant, la carta (afegeix i elimina plats), les comandes dels clients (mostra en una taula la informació de totes les comandes realitzades per les reserves des del programa Reserva) i a més, els top 5 de plats de la setmana i totals. Aquest programa també consta d'una interfície gràfica, que serveix per veure les estadístiques, i les gestions de les comandes i de la pròpia carta del restaurant. Al ser necessari l'intercanvi de informació entre els altres dos programes (Reserva i Entrada), es realitza mitjançant una comunicació TCP. A més, la informació referent a la carta, les taules, i a la informació rebuda sobre les reserves i comandes, s'emmagatzema tota en una base de dades, el qual el Servidor es l'únic programa que té accés a aquesta.

El programa d'Entrada s'encarrega de que el client sigui capaç de demanar o realitzar una reserva al restaurant. Quan es demana, es per aquell mateix dia i hora, mentre que si l'usuari vol reservar una taula, es realitzarà per a una data i hora futura.

Per tal de demanar o reservar una taula, el client haurà de introduir el nom al qual es farà la reserva, el nombre de comensals, i en cas de que sigui una reserva, la data i la hora a la qual es vol reservar. La petició s'enviarà al programa servidor, i aquest validarà si s'ha pogut o no realitzar la petició. En cas de que la petició hagi sigut un èxit, es proporciona un codi d'accés per tal d'accedir al programa Reserva.

Finalment, el programa Reserva. Aquest implementa totes les funcionalitats que poden realitzar els clients quan es troben a la taula assignada de la seva reserva. Per tal de poder realitzar aquestes funcionalitats que ofereix el programa, primer cal autenticar-se al sistema. El programa Servidor serà l'encarregat de verificar aquestes dades. Un cop acceptades, el client podrà veure la carta, afegir plats a la seva comanda, veure l'estat de la seva comanda (llistat amb tots els plats demanats i veure si han estat servits o no), editar la comanda abans de ser enviada al programa servidor (modificar el numero de plats o eliminar plats), i, per últim, pagar i marxar. L'usuari podrà pagar i marxar en qualsevol moment, encara que no s'hagin servit tots els plats.

## 2. DISSENY DE LA INTERFÍCIE GRÀFICA

### 2.1. Interfície gràfica programa Entrada

La vista principal consta d'un CardLayout, el qual ens permet fàcilment canviar de panell segons el que ens interessi. Aquest CardLayout, consta de tres panells diferents, un per tal d'escollir si es vol demanar o reservar taula, un panell per reservar taula, i per últim, un per demanar taula

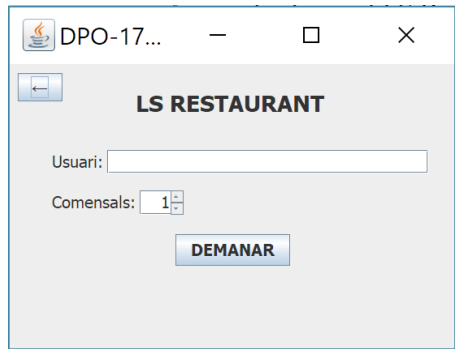
#### Panell per Reservar o demanar



Aquest panell consta de:

- 2 JButtons, un per demanar i un altre per reservar.
- JLabel, el qual ens diu que ens trobem al LS Restaurant.

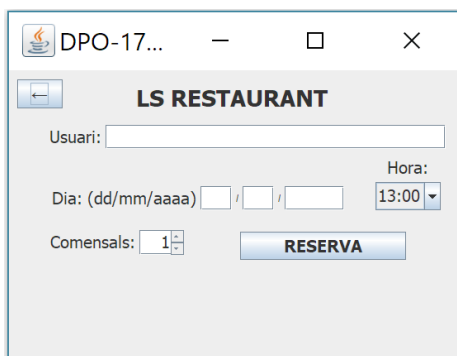
#### Panell per demanar taula



Aquest panell consta de:

- 2 JButtons, un de tornar al panell de Reservar o Demanar, i un altre per demanar la taula.
- 2 JLabels, que indiquen l'usuari i el nombre de comensals per demanar la taula.
- 1 JTextField, per a que el client pugui introduir el nom de reserva desitjat.
- 1 JSpinner, per tal d'escollir el nombre de comensals.

#### Panell per reservar taula



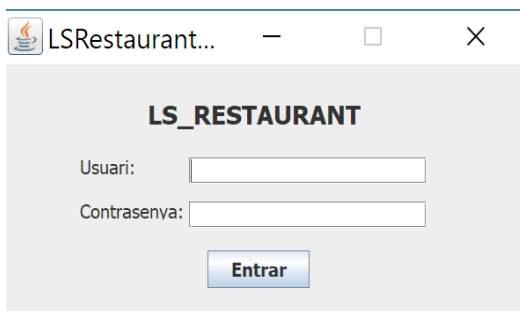
Aquest panell consta de:

- 2 JButtons, un de tornar al panell de Reservar o Demanar, i un altre per reservar la taula.
- 4 JLabels, que indiquen on s'ha de posar l'usuari, el dia que es vol la reserva, l'hora desitjada i els comensals.
- 4 JTextFields, per tal d'introduir l'usuari, el dia, el mes, i l'any de la reserva.
- 1 JComboBox per les hores de la reserva.
- 1 JSpinner per el nombre de comensals.

## 2.2 Interfície gràfica programa Reserva

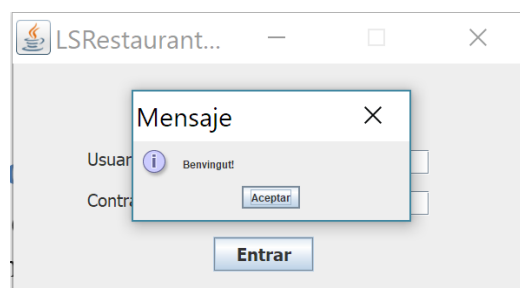
Aquesta vista principal també consta d'un CardLayout, amb el qual fàcilment podem canviar de panell. Aquest CardLayout consta de 4 panells diferents. Un panell d'accés, un panell per visualitzar la carta, un altre per veure l'estat de la comanda, i per últim un altre per pagar i marxar. Quan ja s'ha obtingut accés, quan les credencials introduïdes siguin correctes, la vista principal esta constituïda per un JMenuBar, amb el qual fàcilment podrem accedir a les diferents funcionalitats del programa, i a la part central de la vista, el panell indicat per cada ítem del menú. A l'hora d'obrir el editor de la comanda, apareix una altra vista.

### Panell d'Accés



Aquest panell consta de:

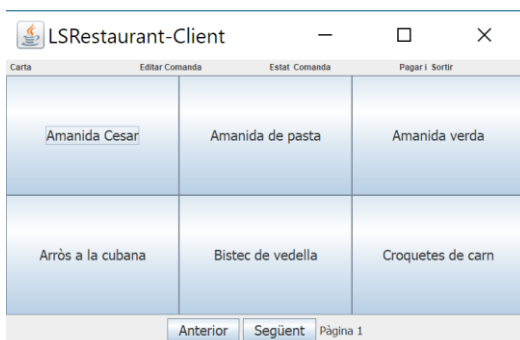
- 1 JButton, un boto per entrar, el qual permet a l'usuari entrar al programa.
- 3 JLabels, els quals indiquen on s'ha de posar el nom d'usuari i la contrasenya donada anteriorment al realitzar la reserva.
- 2 JTextFields, els quals serveixen per posar les dades per tal d'accedir al sistema.



Ja siguin les credencials correctes o no, apareix un MessageDialog que ens indica en un missatge si hem pogut o no entrar.

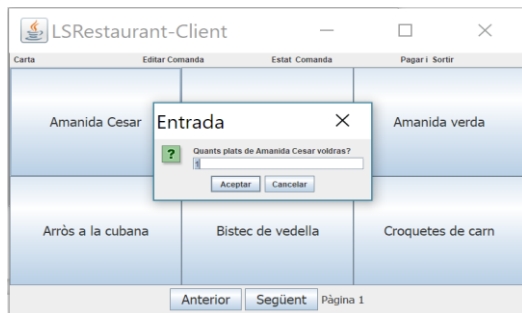
### Panell Carta

Aquest panell esta format per un altre panell, que està format per els botons on es troben els plats. Cada un d'aquests panells son les diferents pagines que formen la carta, formades per 6 botons, on cada boto fa referència a un plat diferent.



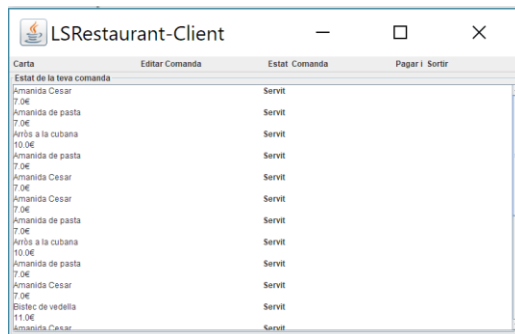
Com podem veure, el panell esta constituït per:

- La pagina de la carta, que constitueix en menú del restaurant.
- 2 JButtons, un per anar a la pagina anterior i un altre per passar de pàgina.
- 1 JLabel que ens indica en quina pàgina ens trobem.



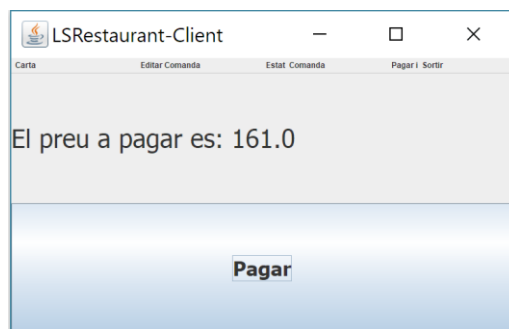
A l'hora de afegir un plat, surt un `OptionPane`, el qual demana al client la quantitat de plats que es vol afegir.

## Panell Estat Comanda



Aquest panell que el forma, com es pot veure es tracta d'un `JScrollPane`. A part, el forma un `JTextArea` on surt escrita tota la teva comanda, i, també si ja ha estat servit o no el plat.

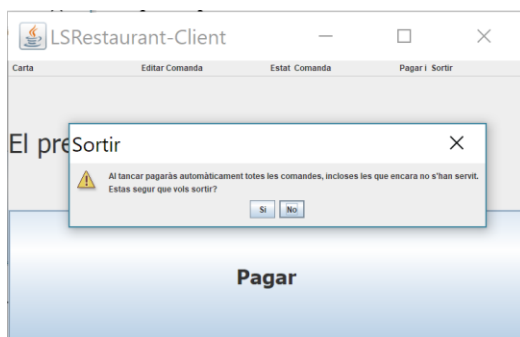
## Panell Pagar i Sortir



Aquest panell el forma:

1 `JLabel`, que ens indica el preu a pagar.

1 `JButton` per pagar.



Quan es prem el botó de pagar, apareix un `OptionPane` que indica al client si està segur de marxar. En funció de si queden plats encara per servir o no, sortirà un missatge o un altre.

## 2.3 Interfície gràfica programa Servidor

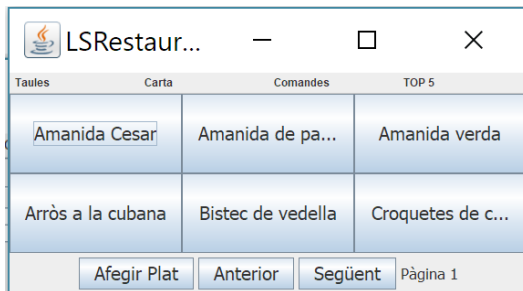
La vista del servidor també consta amb un JMenuBar, el qual ens porta als diferents panells que es poden consultar. Aquestes funcionalitats són veure les taules, on es pot veure l'estat de les taules (si es troben reservades o no, i si ho està, veure l'estat de la reserva), eliminar o afegir taules, veure la carta, veure l'estat de les comandes de cada reserva, i per últim, mostrar el top 5 setmanal o totals.



### Panell Taules

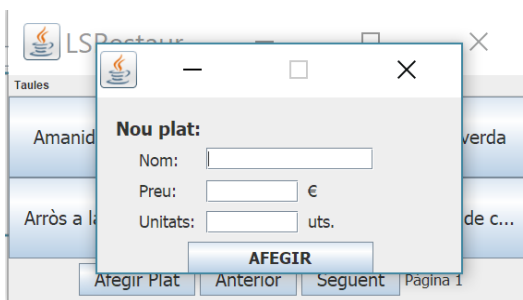
Aquest panell està format per un JSplitPane. A la part esquerra del panell es troba un llistat de les taules i dos botons per poder actualitzar o eliminar taules.

A la part dreta del panell, es troba un JTable que conte les reserves d'aquella taula. Finalment, tenim un JTextField i un JButton per tal d'afegir una nova taula de tants comensals.



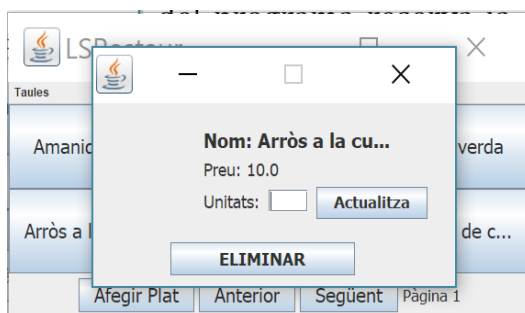
### Panell Carta

Aquesta vista és molt semblant a la vista de la carta del programa reserva ja explicada anteriorment. La diferència més notable és la de afegir un plat.



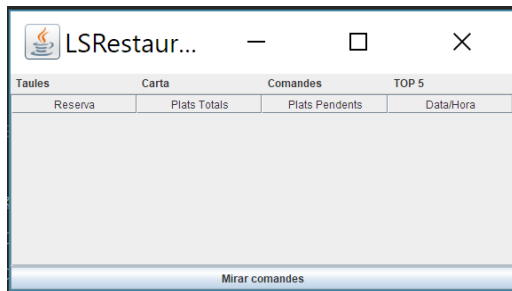
Quan es vol afegir un plat, apareix un JDialog, amb el qual introduïm les dades referents al plat per afegir-lo a la carta. Consta de 6 JLabels, 3 JTextFields per posar les dades, i per últim, un JButton per afegir-lo. En cas de que el plat ja existeixi o les dades no siguin correctes, apareix missatge d'error.

En cas contrari, surt un missatge d'èxit.

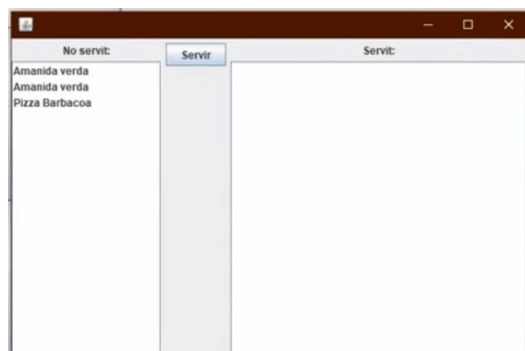


Quan es prem un boto de plat, apareix un JDialog. Aquest JDialog et dona dues opcions, eliminar un plat de la carta, o canviar les unitats disponibles d'aquell plat.

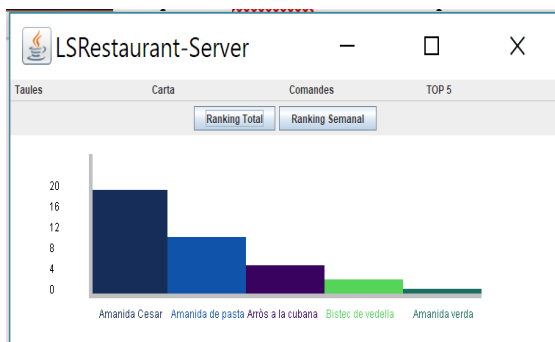
## Panell Comandes



Aquest panell, de tipus JScrollPane, consta d'una JTable, la qual ens mostra totes les reserves, i quants plats pendents te aquest. El JButton de sota, serveix per mirar les comandes de la reserva desitjada, i així poder servir els plats d'aquella comanda.



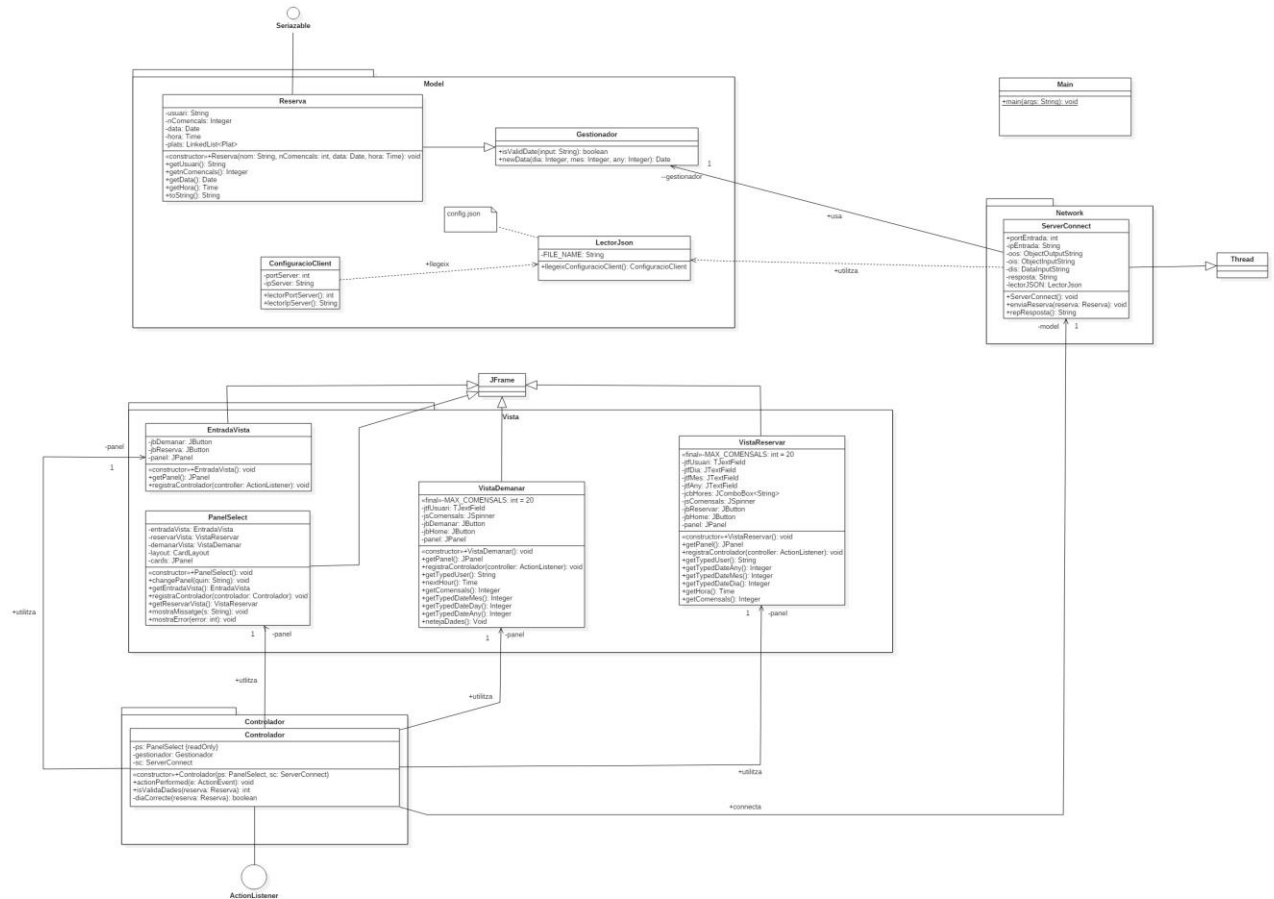
Aquest JFrame, serveix per editar l'estat de les comandes de cada reserva. Consta de dos JLists, un per plats servits, i un altre per els que encara estan pendents. També consta d'un JButton per servir el plat. Quan es serveix el plat, aquest canvia de JList, i la informació s'envia al gestor, el qual envia aquesta informació a la base de dades, i també s'envia al programa reserva, per tal d'actualitzar la vista de l'estat de la comanda.



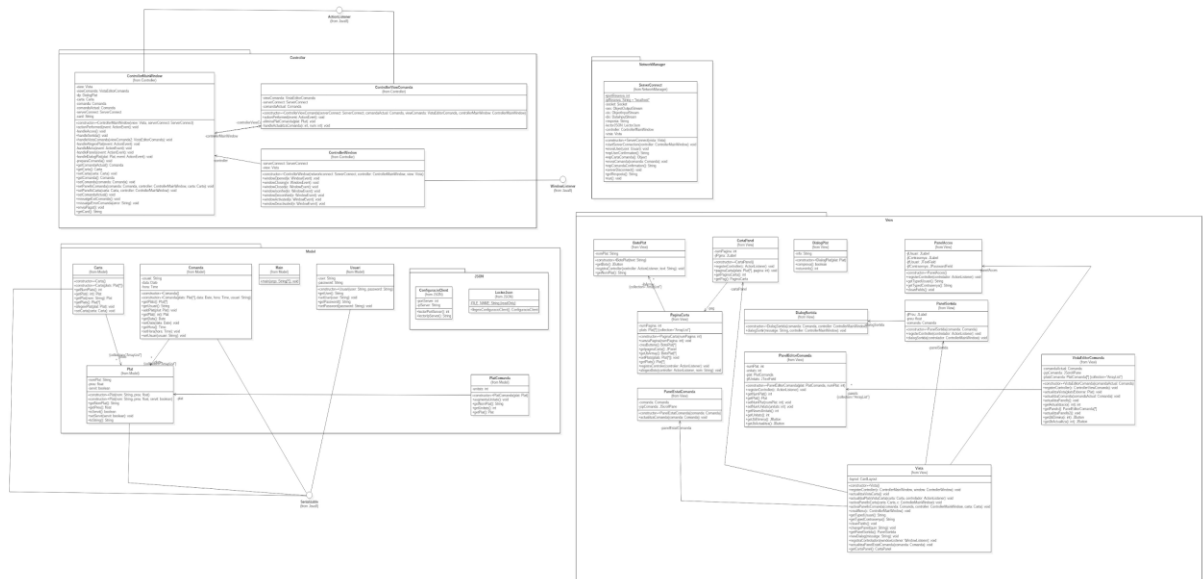
## Panell Top5

Per últim, tenim el panell dels Top5. Aquest panell consta d'una classe gràfic, la qual es un panell, que ens mostra els 5 plats més demanats, ja siguin d'aquesta setmana, o els plats més demanats totals. Aquest panell consta de 2 JButtons, els quals ens permeten canviar de gràfic segons es vulgui, al top 5 setmanal, o al top 5 total.

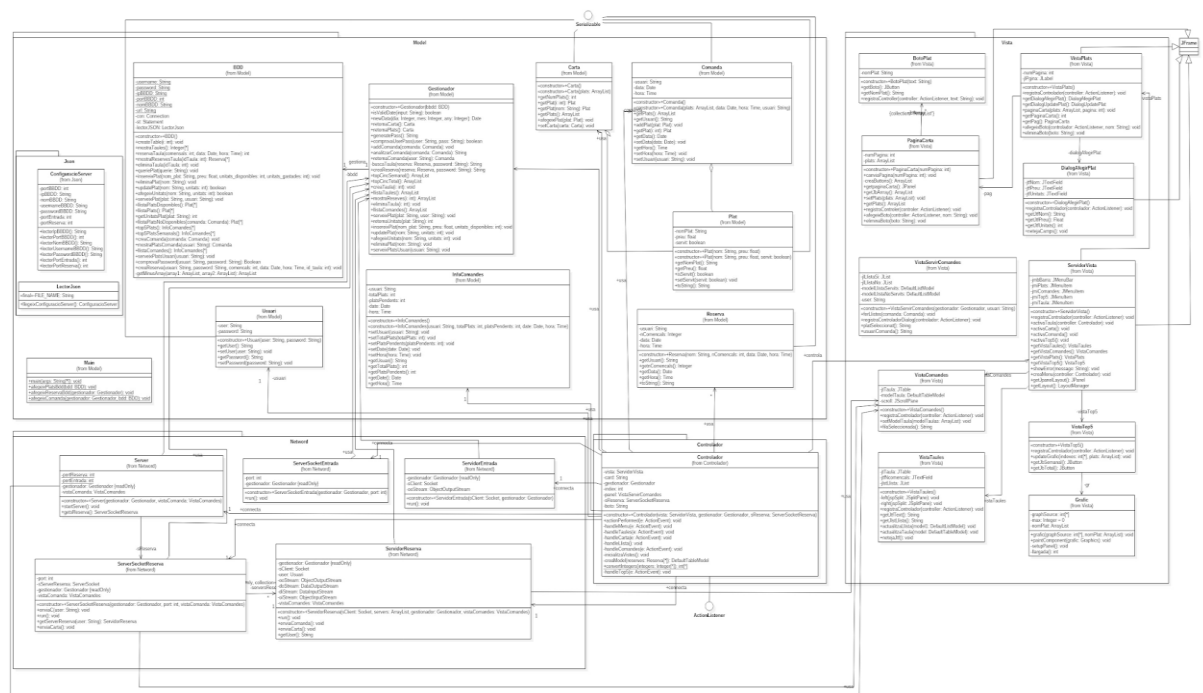




## b) Diagrama de Reserva



## c) Diagrama del servidor



## 3.2 DESCRIPCIÓ

Per a l'aplicació del patró de disseny Model-Vista-Controlador, hem seguit la següent estructura:

### a) Client Entrada

Les classes de Vista capten les accions realitzades per l'usuari i els envien directament al Controlador, que implementa l'ActionListener i connecta amb la Network que a la vegada connecta amb el Model i les seves classes Gestionador i LectorJson.

Descripció de les classes més importants:

Controller

**Controlador:** Classe que controla totes les accions de botons i qualsevol Action Listener. Implementa Action Listener per poder escoltar els botons.

Model

**Gestionador:** Classe que té les funcions de comprovar la data, i altres comprovacions a fer..

**Reserva:** Tipus reserva que conté el usuari, data, hora i nombre de comensals. Implementa Serializable per poder enviar-la al servidor.

### b) Client Reserva

Té un funcionament molt semblant al client Entrada que hem mencionat anteriorment. En aquest cas enviem classes comanda i usuari al server.

Descripció de les classes més importants:

Model

**Carta:** Arraylist de plats de la carta amb els sus getters i setters. Implementa Serializable per poder enviar-la per el servidor.

**Comanda:** Classe que conté els plats de la comanda, el usuari, la data i hora. Implementa Serializable per poder enviar-la per el servidor.

**Plat:** Classe que conte la informació del plat(nom, preu), implementa Serializable per poder enviarlo per el servidor.

**Usuari:** Classe usuari que conté el nom d'usuari i password.

### c) Server

Les classes del package Vista del servidor són panels i ens movem entre ells fent us de un CardLayout. El Controlador (registrat a ServidorVista) interactua amb diferents classes tant de Network com de Model i Vista. Les classes de Network, estableixen associacions amb les de Model i, com hem esmentat anteriorment, les de Network.

#### Vista

**ServidorVista:** Classe que té la vista general del servidor, principalment s'encarrega d'intercanviar les pestanyes i inicialitzar els diferents panels que formen la vista.

#### Model

**ConfiguracioServer:** Classe que conte les dades que llegeix del fitxer json.

**LectorJson:** Lector de el json config.

**BDD:** Classe que s'encarrega de realitzar totes les queries i actualitzacions d'informació de la base de dades.

**Gestionador:** Classe que conte totes les funcions per fer diferents tasques referents a la base de dades.

**InfoComandes:** Classe que conte la informació de els plats demanats i pendents. Ens serveix per realitzar els gràfics.

**Reserva:** Classe Reserva que conté la informació (usuari, nombre de comensals, data i hora) de la reserva.

#### Network

**Server:** Classe servidor que es comunicarà amb els "dos" clients (reserva, entrada)

**Main:** Inicialitza el servidor i la seva base de dades. Creem la vista i el seu controlador. Durant la primera execució del server afegeix plats i taules a la bbdd, afegeix reserves i comandes.

## 4. METODOLOGIA DE DESENVOLUPAMENT

En quant a la metodologia que hem seguit, en primer lloc, vam crear i preparar tot el projecte organitzat en els tres programes diferents al github, després ens vam centrar sobre tot en implementar la base de dades i la seva connexió amb el servidor, és a dir, crear totes les funcions per omplir, eliminar i modificar tota la informació de la base de dades.

Tot seguit, vam començar a implementar les funcions que farien funcionar el servidor, totes les funcions de crear usuaris, crear plats, servir plats, etc.

Un cop vam tenir la base del servidor preparada amb la base de dades, vam començar a fer unes vistes molt bàsiques per provar el funcionament del servidor, a la vegada que vam començar a fer la part de reserva.

Un cop vam tenir les vistes bàsiques, vam començar a implementar-les i fer els controladors de totes, a la vegada que vam començar a fer la connexió entre servidor i client.

Un cop vam tenir la connexió bàsica, vam començar a implementar la connexió per tal que la vista s'actualitzés automàticament al rebre la informació del servidor sense haver de prémer el botó. Això ens va portar problemes sobre tot per que es refresqués la vista i especialment, la part de la carta, que al haver moltes pàgines, vam tenir algun problema per fer que s'actualitzés correctament.

Un cop vam tenir la connexió i les vistes que s'actualitzessin automàticament vam acabar de fer les vistes i els gràfics del servidor. Un cop vam tenir això acabat, vam començar a canviar petites coses i controlar alguns errors.

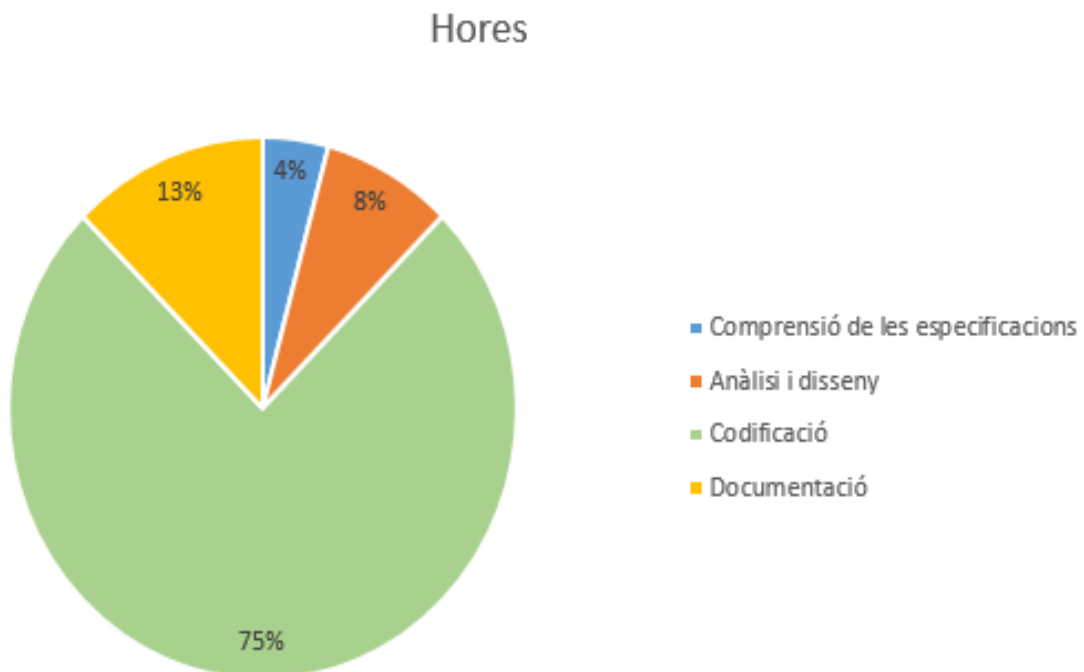
Finalment, un cop acabada tota la fase de prova vam afegir algunes últimes funcionalitats i provar-les i ja vam tenir el projecte acabat.

## 5. DEDICACIÓ

Sense cap dubte podem dir que aquest projecte ha estat el treball més extens de tot el curs. Si bé això, els grups eren de 5, pel que ens hem pogut repartir la feina i no ha sigut tant pesat com ho hauria sigut per una sola persona.

No sabem exactament quant hem trigat exactament, però hem fet uns càlculs aproximats i imaginem que els números deuen anar entre 230 - 250 h.

1. Comprensió de les especificacions: 10h
2. Anàlisi i disseny: 20h
3. Codificació: 180h
4. Documentació: 30h



## 6. CONCLUSIONS

Després de gairebé tot el semestre desenvolupant el projecte, i trobar molts errors, vam aconseguir un programa que ens funcionés correctament i realitzés totes les funcions requerides per el enunciat.

Ara, un cop acabat, ens adonem de que hem après molt de com treballar en un entorn de treball amb més de 2 persones, com estem acostumats. Això ens ha forçat a fer us de eines com github, per poder portar un control sobre les diferents versions que hem anat tenint. Ens ha sigut molt útil per trobar errors que sense adonar-nos, havíem introduït, i tornar a poder treballar des de la versió que ens funcionava correctament.

Un altre punt important ha sigut el fer us del pòster per en tot moment saber en quin punt de el procés estàvem i així saber si ens era necessari fer més o menys feina al següent sprint.

Ens ha semblat molt interessant treballar amb mysql dins de java, ja que ens hem adonat de que és una forma molt més optima de treballar amb grans quantitats de dades i de aconseguir, amb queries, aquella informació que necessitem en cada moment.

Alhora de organitzar el codi i fer el javadoc, ens ha sigut bastant senzill adaptar-nos a les convencions i tenir-lo ordenat i organitzat de forma clara.

En general, ha sigut un projecte amb el que hem après molt sobre les connexions client-servidor, sobre la interacció entre java i mysql, i a treballar en equip per poder acabar el treball a temps.

## 7. BIBLIOGRAFIA CONSULTADA

- Campus online de La Salle: Universitat Ramón Llull - eStudy [en línia] [dates de consulta: del març del 2018 al maig del 2018].

Disponible en: <https://estudy.salle.url.edu/course/view.php?id=13184>

- Web per desenvolupadors StackOverflow [en línia] [dates de consulta: del març del 2018 al maig del 2018].

Disponible en: <https://stackoverflow.com/>

- Java Documentation [en línia] [dates de consulta: del març del 2018 al maig del 2018].

Disponible en: <https://docs.oracle.com/javase/7/docs/api/>