

Disseny i programació orientats a objectes

Pràctica del segon semestre
Centralized Restaurant

Departament d'Enginyeria
La Salle – Universitat Ramon Llull
12 de març de 2018

Índex

1. Descripció general	3
2. Especificació de requeriments	4
2.1. Aplicació Servidor	4
2.1.1. Configuració	5
2.1.2. Gestionar reserves	5
2.1.3. Gestionar les taules	6
2.1.4. Gestionar la carta de plats del restaurant.....	6
2.1.5. Gestionar les comandes de cada Reserva	6
2.1.6. Mostrar el top 5 de plats de la setmana i totals.....	7
2.2. Aplicació Reserva	7
2.2.1. Configuració	7
2.2.2. Autenticar-se al sistema	8
2.2.3. Veure la carta i realitzar comanda	8
2.2.4. Veure l'estat actual de la comanda	8
2.2.5. Pagar i marxar	8
2.3. Aplicació Entrada	9
2.3.1. Configuració	9
2.3.2. Demanar / Reservar taula	9
3. Consideracions	10

1. Descripció general

Alguns restaurants comencen a modernitzar la seva forma de treballar i a automatitzar i centralitzar algunes tasques. Des de l'assignatura volem demostrar que la gestió d'un restaurant totalment automatitzat i centralitzat és possible.

L'arquitectura del sistema *Centralized Restaurant* és client-servidor. Tal com es pot observar a la figura següent, en aquesta hi ha un programa servidor que s'executa en una màquina en concret (el servidor ubicat a la cuina del restaurant) i un conjunt de màquines que executen el programa client (el taulell de recepció i les aplicacions que es troben en les taules).

El programa servidor, entre d'altres, emmagatzemarà les dades d'accés, els productes disponibles, la carta i les taules. Els programes client, seran el programari que utilitzaran els usuaris mitjançant el qual es podran demanar o reservar una taula, veure la carta o fer comandes.

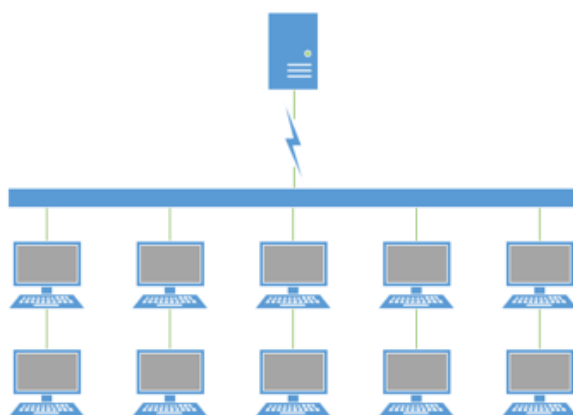


Figura 1: Representació de l'arquitectura client – servidor de *Centralized Restaurant*.

Els apartats següents descriuen els detalls i el comportament que s'espera de les funcionalitats concretes que ha d'implementar cadascuna de les parts identificades en aquest escenari: servidor i client.

2. Especificació de requeriments

Aquest apartat recull el funcionament i el comportament que s'espera de l'aplicació *Centralized Restaurant* per tal de satisfer les especificacions requerides. En primer lloc les del programa servidor i tot seguit les dels programes "Recepció" i "Reserva".

2.1. Aplicació Servidor

El servidor ha de ser un programa que pugui ser executat en una màquina independent de la resta dels clients, que ofereixi una interfície gràfica amb diverses opcions i capaç de proveir unes funcionalitats mitjançant la recepció i enviament de dades per una xarxa. Concretament ha d'oferir funcionalitats per:

1. Gestionar reserves (no accessible mitjançant la interfície gràfica, opció remota)
2. Gestionar les taules del restaurant
3. Gestionar la carta de plats del restaurant
4. Gestionar les comandes de cada Reserva
5. Mostrar el top 5 de plats de la setmana i totals

La primera funcionalitat és d'accés remot i té a veure amb la comunicació remota amb els programes Recepció i Reserva en execució. En el sentit que el programa servidor ha de tenir la capacitat de gestionar les peticions corresponents mitjançant una connexió TCP (amb *sockets*) des del programa client. En canvi, les funcionalitats 3, 4 i 5, són opcions que ha d'oferir la interfície gràfica del propi programa servidor. Els detalls d'aquestes funcionalitats s'expliquen al llarg dels propers apartats.

Com es veurà a continuació, el programa servidor haurà d'emmagatzemar dades de les reserves (nom de la taula, paraula de pas, comandes que ha fet, etcètera), informació referent a la carta de plats (nom del plat, unitats disponibles, etcètera) i informació sobre les taules existents (identificador de la taula, nombre de cadires, si està reservada o no, etcètera) en una base de dades MySQL. Referent a això, és important notar el fet de que el programa servidor és l'únic programa que té accés a la base de dades, i que la lògica de funcionament del sistema (com per exemple la distribució de les taules per reserves) comprèn l'intercanvi d'informació entre els programes servidor, recepció i reserva mitjançant la comunicació TCP amb *sockets*.

Aquestes característiques es poden apreciar en el següent diagrama de desplegament UML.

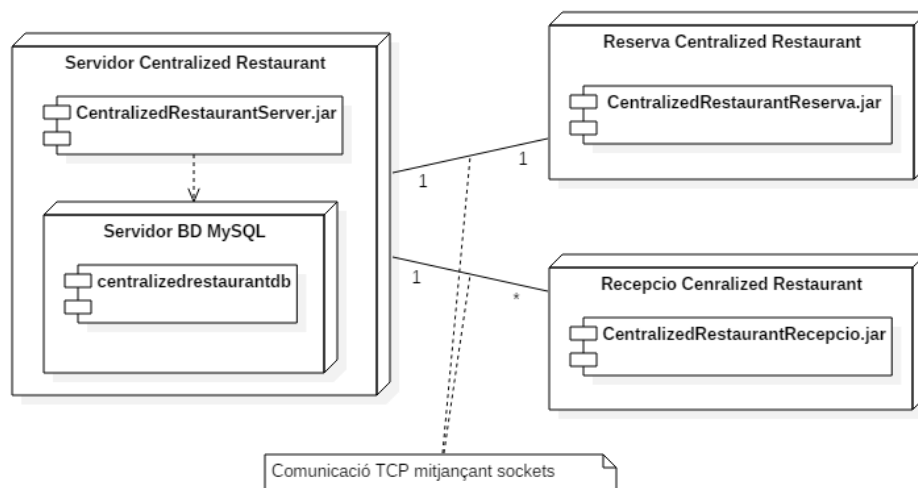


Figura 2: Diagrama de desplegament UML de Centralized Restaurant

2.1.1. Configuració

En el moment d'engegar el servidor aquest haurà de llegir un fitxer de configuració anomenat *config.json* ubicat en el directori arrel del projecte on hi haurà la següent informació:

- Port de connexió amb la Base de dades
- Direcció IP del servidor de la base de dades (normalment "localhost")
- Nom de la Base de dades
- Usuari d'accés a la Base de dades
- Contrasenya d'accés a la Base de dades
- Port de comunicació on escoltar les peticions del programa Recepcio
- Port de comunicació on escoltar les peticions dels programes Reserva

2.1.2. Gestionar reserves

La primera funcionalitat ha de permetre realitzar noves reserves en el sistema. Cal indicar que una nova reserva serà, tan una reserva feta per una data futura, com una petició de taula per assegurar-se al moment.

Tota reserva ha de ser caracteritzada per: un nom de reserva, un *password* auto-generat pel servidor, el nombre de comensals i la data de la reserva (futura o la data actual segons necessitat). La reserva es farà de forma remota, de forma que aquesta opció ha d'estar activa en tot moment mentre el servidor és en execució. Serà l'encarregada de rebre la informació enviada, mitjançant *sockets*, per la Recepció en la seva pròpia opció de fer una reserva.

La tasca del servidor serà: validar les dades rebudes, auto-generar un codi d'accés i comprovar si hi ha una taula disponible per la data sol·licitada.

El servidor haurà de ser qui comprovi que les dades de la reserva són vàlides en aquelles situacions en què cal consultar la base de dades, és a dir, si l'usuari introdueix un camp buit en el formulari es pot gestionar des del programa Recepció, però que el nom de la reserva no estigui repetit, ha de ser verificat pel programa servidor, ja que és aquest qui té accés a la base de dades. Tanmateix, totes aquelles comprovacions que faci el programa Recepció també haurà de fer el programa servidor.

El servidor serà l'encarregat de generar un codi d'accés per a la nova reserva. El codi d'accés haurà de ser sempre de 6 caràcters i podrà contenir lletres (majúscules i minúscules) i números.

El servidor també serà l'encarregat de proveir d'una taula per la reserva tenint en compte l'ocupació de les taules, les reserves fetes per a un futur en aquelles taules i el nombre de comensals de la reserva.

A l'igual que per fer la reserva, el servidor és l'encarregat, quan un programa Reserva es vulgui autenticar, de verificar que el nom de la reserva existeix i que el codi d'accés introduït és l'associat a aquesta reserva.

Quan el servidor rebí una petició de reserva ha de procedir a efectuar-lo i donar resposta al programa client segons sigui el resultat, satisfactori o insatisfactori, de la petició.

2.1.3. Gestionar les taules

El servidor ha d'oferir controls per tal de gestionar les taules existents en el restaurant. En concret, caldrà oferir controls per afegir una nova taula al sistema, mostrar un llistat de totes les taules existents en el sistema, mostrar les reserves fetes per a cada una de les taules del sistema i eliminar una taula del sistema.

2.1.4. Gestionar la carta de plats del restaurant

El servidor haurà d'oferir controls per tal de poder gestionar la carta de plats que s'ofereix en el restaurant. En concret, caldrà poder afegir un nou plat a la carta, mostrar tots els plats existents actualment en la carta i eliminar un plat de la carta.

Respecte als plats, aquests hauran de tenir, com a mínim, un nom, un preu i el nombre d'unitats que el restaurant és capaç de servir segons els recursos del magatzem.

Finalment, caldrà afegir també una funcionalitat que permeti actualitzar el nombre d'unitats de les quals es disposa per a cada plat.

2.1.5. Gestionar les comandes de cada Reserva

El servidor haurà de permetre mostrar una taula (JTable) amb la informació sobre totes les comandes realitzades per les diferents Reserves. La informació sobre cada comanda s'haurà de rebre des de l'aplicació Reserva, però la informació s'haurà d'emmagatzemar a la base de dades. Per cada una de les comandes caldrà indicar, com a mínim, el nom de la reserva que ha fet la comanda, el nombre de plats existents a la comanda, el nombre de plats pendents de servir de la comanda i la data de realització de la comanda.

El llistat de comandes sempre haurà d'estar ordenat per la data de la comanda, de forma que a la part superior de la taula sempre s'haurà de mostrar la comanda que porta més temps esperant a ser servida. El llistat de comandes s'haurà de refrescar en tot moment en temps real.

Respecte a les comandes, una Reserva podrà actualitzar en tot moment la seva comanda afegint noves peticions. En aquest cas caldrà actualitzar tota la informació corresponent a la comanda i caldrà actualitzar també la data de realització de la comanda (i reordenar-la en la taula).

En el moment que l'usuari cliqui en una de les comandes de la taula, caldrà oferir els controls per mostrar tota la informació sobre la comanda, un llistat amb tots els plats pendents de servir i un llistat amb tots els plats que ja s'han servit. També caldrà oferir un control per tal de

servir un plat, de forma que s'actualitzin les dues llistes i es notifiqui en temps real a la Reserva corresponent que el plat ja està en camí.

2.1.6. Mostrar el top 5 de plats de la setmana i totals

El servidor haurà d'oferir la possibilitat de mostrar un conjunt de gràfics amb estadístiques sobre el sistema. En concret, l'aplicació haurà de permetre mostrar un gràfic de barres on es mostrin els 5 plats més demanats de la setmana i un segon gràfic de barres on es mostrin els 5 plats més demanats des de l'obertura del restaurant.

Els gràfics hauran de ser generats des del programa, utilitzant les eines de dibuix que ofereix la llibreria AWT/SWING de Java i escalat de forma conseqüent en funció dels valors mínim i màxim de l'eix de les y o ordenades on es mostri el número de plats demanats. No es poden utilitzar llibreries externes.

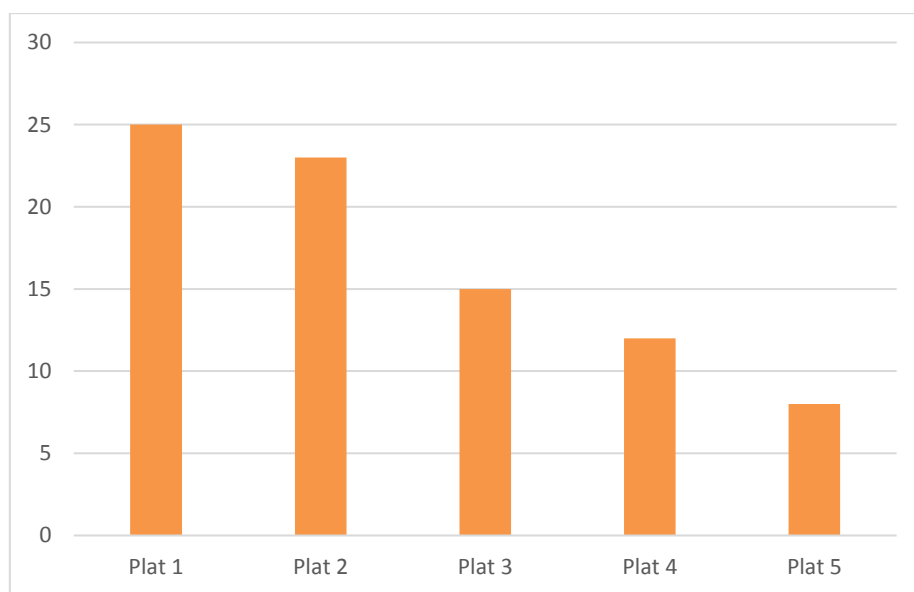


Figura 3: Top 5 plats setmanals

Un cop descrites les funcionalitats del servidor, l'apartat següent exposa les funcionalitats dels clients, aquells programes que seran utilitzats pels usuaris i que enviaran informació, mitjançant *sockets*, al programa servidor.

2.2. Aplicació Reserva

En aquest apartat es descriuen les opcions que ha d'implementar el programa Reserva. És a dir, del programa que mitjançant la comunicació amb el servidor implementa les funcionalitats que poden executar els clients un cop es trobin asseguts a la taula assignada per la seva reserva.

2.2.1. Configuració

En engegar el programa, aquest haurà de carregar el fitxer de configuració *config.json* amb la direcció IP i el port de connexió amb el servidor que està obert per a les reserves. Aquest fitxer ha d'estar ubicat en el directori arrel del projecte del programa Reserva.

Un cop llegit el fitxer s'obrirà la finestra gràfica, la qual haurà de permetre:

1. Autenticar-se al sistema

2. Veure la carta i realitzar comanda
3. Veure l'estat actual de la comanda
4. Pagar i marxar

Els propers apartats expliquen les característiques de cadascuna d'aquestes opcions.

2.2.2. Autenticar-se al sistema

Per tal d'autenticar-se al sistema l'usuari haurà d'introduir el nom de la reserva i la contrasenya associada. Un cop s'envia el formulari al servidor, aquest el validarà i garantirà accés si les credencials són correctes i, en cas contrari, mostrarà un error genèric tal que "Les credencials introduïdes són incorrectes" en qualsevol cas.

Pel que fa al tancament de la sessió, tot usuari que ha estat autenticat ha de poder tancar sessió en qualsevol moment mitjançant l'opció de pagar i sortir.

2.2.3. Veure la carta i realitzar comanda

Un cop l'usuari es trobi autenticat en el sistema, l'aplicació haurà d'oferir la possibilitat de veure la carta de plats que ofereix el restaurant. Respecte a la carta, s'hauran de mostrar tots els plats que estiguin disponibles, és a dir, aquells plats registrats al sistema que tinguin unitats disponibles. En aquesta funcionalitat també s'haurà de permetre afegir un plat a la comanda actual i triar el nombre d'unitats que es volen afegir.

L'aplicació també haurà de permetre visualitzar quins plats es troben afegits a la comanda pendent de realitzar (aquells plats que s'han triat, però que encara no s'han enviat al servidor). Per cada un d'aquests plats, caldrà afegir controls per tal d'esborrar-los de la comanda o modificar el nombre d'unitats.

L'aplicació haurà d'oferir controls per tal d'enviar una comanda al servidor. El servidor serà l'encarregat de gestionar que tota la informació de la comanda és correcta (per exemple: el nombre d'unitats de tots els plats està disponible al magatzem). En el cas que la Reserva ja hagués realitzat una comanda prèviament, la nova comanda afegirà els nous plats a la comanda ja realitzada.

2.2.4. Veure l'estat actual de la comanda

Aquesta opció ha de permetre veure l'estat actual de la comanda que s'ha enviat al servidor. En concret, s'haurà de mostrar una llista de tots els plats compresos en la comanda indicant quins d'ells ja s'han servit i quins encara estan pendents de servir.

2.2.5. Pagar i marxar

Aquesta opció ha de permetre a l'usuari visualitzar el preu total pendent de pagar segons la comanda realitzada al servidor. En qualsevol moment l'usuari podrà decidir pagar l'import i tancar la sessió. En cas de disposar de plats que encara no s'han servit, s'haurà de cancel·lar la comanda al servidor (tot i que l'import a pagar inclourà els plats encara no servits).

2.3. Aplicació Entrada

En aquest apartat es descriuen les opcions que ha d'implementar el programa Entrada. És a dir, del programa que mitjançant la comunicació amb el servidor implementa les funcionalitats que poden executar els clients per tal de realitzar reserves al restaurant.

2.3.1. Configuració

En engegar el programa, aquest haurà de carregar el fitxer de configuració *config.json* amb la direcció IP i el port de connexió amb el servidor que està obert per a l'aplicació Entrada. Aquest fitxer ha d'estar ubicat en el directori arrel del projecte del programa Reserva.

Un cop llegit el fitxer s'obrirà la finestra gràfica, la qual haurà de permetre:

1. Demanar / Reservar taula

2.3.2. Demanar / Reservar taula

L'aplicació Entrada ha d'oferir la possibilitat de demanar una taula o reservar una taula. La principal diferència entre les dues funcionalitats és que demanar una taula és una reserva per a la data (dia i hora) actual, mentre que reservar una taula és una reserva per a una data futura.

Per tal de demanar / reservar una taula, caldrà introduir el nom de la reserva, el nombre de comensals i (només per a reserves) la data de la reserva.

La petició s'haurà d'enviar al servidor, que validarà la disponibilitat de la reserva (nom de la reserva no existent i disponibilitat de taules per a la data i nombre de comensals) i, en cas de poder gestionar la petició, generar un codi d'accés que es mostrarà a l'usuari.

3. Consideracions

De forma addicional a les especificacions anteriors, cal tenir en compte les restriccions següents.

Estructures de dades

No és necessari que codifiqueu les estructures de dades, com per exemple una llista dinàmica. Podeu usar les classes del paquet `java.util` de l'API de Java, com les classes `LinkedList<E>`, `ArrayList<E>`, `Vector<E>` i/o d'altres.

D'aquesta manera obtindreu, "gratuïtament", estructures de dades amb funcionalitats interessants, lliures d'errors i ràpides, amb el que agilitzareu molt el procés de codificació de la pràctica.

Dibuix de gràfiques

Per tal de dibuixar les gràfiques no es pot fer ús de cap llibreria externa al SDK de Java (JDK). Cal utilitzar algun dels mecanismes que ofereix la llibreria AWT/SWING de Java i esclar els eixos en funció dels valors màxim i mínim del conjunt de dades que es representen.

Disseny gràfic

L'aspecte visual tan del programa servidor com client és totalment lliure, sigueu creatius. Investigueu el potencial que us ofereix Java, utilitzeu menús, separadors, desplegable, colors, creeu imatges personalitzades pels botons, etcètera.

Procés de desenvolupament

Abans de posar les mans sobre el teclat i començar a codificar, penseu. Analitzeu què cal implementar i dissenyeu quines classes i quines relacions tindrà el sistema, agafeu paper i llapis o utilitzeu l'eina StarUML per tal de crear un diagrama de classes UML base.

Segurament aquest diagrama no serà el definitiu, i anirà canviant a mida que avanceu el desenvolupament, però d'aquesta manera assentareu unes bases sòlides pel desenvolupament del sistema i us estalviareu hores de codificació i molts maldecaps.

El desenvolupament d'un projecte informàtic real típicament comprèn les fases: presa de requeriments, especificació, anàlisi, disseny, implementació, proves i implantació. Podeu pensar que les dues fases inicials d'aquesta seqüència ja han estat realitzades per l'equip docent i que ara us passem el relleu per tal que acabeu el sistema, que continueu amb l'anàlisi, disseny, etcètera.

El no compliment d'algun dels punts especificats en aquest enunciat suposarà la no acceptació de la pràctica i la devolució de la mateixa.

La detecció de copia comportarà suspendre tots el integrants dels grups implicats, tant els que han copiat com els que han estat copiats, i perdre l'opció de presentar-se a la pròxima convocatòria ordinària. (veure normativa de la universitat)