

EVALUACION	OBLIGATORIO	FECHA ENTREGA	10/12/2018
MATERIA	Diseño y Desarrollo de Aplicaciones		
CARRERA	Analista Programador		
CONDICIÓN	Grupo de hasta dos alumnos.		

SISTEMA PARA ATENCIÓN DE RESTAURANTS

Precarga de información.

NO es necesario implementar una interfaz de usuario para el mantenimiento de la información. El sistema deberá tener pre-cargada la información, de modo que al iniciarse ya cuente con un conjunto de datos definido.

Usuarios: Información básica que se desea manejar sobre los usuarios: nombre de usuario (que debe ser único), contraseña y nombre completo. Existen usuarios que atienden mesas (en adelante mozos) y usuarios que gestionan los pedidos: cocineros, barmans, etc (en adelante gestores).

Unidades procesadoras de pedidos: De cada una se sabe su nombre. Deben definirse mínimo dos, una de nombre "Cocina" y otra de nombre "Bar".

Mesas: De cada mesa se debe registrar el número de mesa y a que mozo esta asignada.

Productos: Los productos tienen código, nombre, precio unitario, cantidad disponible en stock y la unidad procesadora de los pedidos del producto.

Clientes: De cada cliente se deberá conocer su id, su nombre y su email. Los comensales del Restaurant pueden estar registrados como clientes o no. Si están registrados pueden llegar a tener algún beneficio al pagar el servicio. Existen 3 tipos de clientes. Comunes: Pagan \$0 por todos los cafés consumidos en el servicio. Preferenciales: Pagan \$0 por todas las aguas minerales consumidas en el servicio y si el monto total del servicio supera los \$2000 tienen un 5% de descuento sobre el total. De la casa: Tienen \$500 de consumo gratis. Cuando se cierra el servicio, si el cliente está registrado, se aplica el beneficio correspondiente. Un cliente puede cambiar de tipo en cualquier momento. Si bien aún no se implementa un caso de uso para el cambio de tipo de un cliente esto tiene que ser considerado en la solución.

Aplicación de atención de mesas.

Esta aplicación es utilizada solo por los mozos. Los casos de uso disponibles en esta aplicación son los siguientes:

- **Ingreso:** El mozo indica su nombre de usuario y contraseña. El sistema muestra el nombre completo del mozo en el título de la ventana principal y el conjunto de mesas que el mozo tiene asignadas. El conjunto de mesas asignadas debe ser el mismo que tenía en su último ingreso al sistema. Un mismo mozo no podrá ingresar dos veces al mismo tiempo. Las mesas deben mostrarse gráficamente (se sugiere JTable), no como texto. Las mesas libres (o cerradas) en un color y las ocupadas (o abiertas) en otro color diferente. Debe mostrarse además el número de cada mesa. Este conjunto de mesas deberá estar siempre visible en la pantalla principal de la aplicación. Cada mesa debe poder seleccionarse haciendo clic sobre la misma. Cada vez que se selecciona una mesa, si la mesa está abierta, deben mostrarse todos los datos del servicio de la mesa. (**)
- **Abrir una mesa:** El mozo selecciona una mesa e indica que desea abrirla. La mesa queda abierta. Se denomina servicio al conjunto de artículos consumidos en una mesa desde que se abre la mesa (llegan los comensales) hasta que se cierra (se retiran los comensales). Cada Servicio tiene un conjunto de ítems. En cada ítem el mozo indica el artículo consumido, la cantidad y una descripción opcional.
- **Agregar un artículo al servicio:** El mozo selecciona una mesa abierta e indica que desea agregar un artículo al servicio. El sistema muestra la lista de artículos con stock disponible, el mozo selecciona un artículo e ingresa la cantidad y opcionalmente una descripción. Si la cantidad es menor a 1 el sistema mostrara el mensaje “cantidad invalida”. Si la cantidad es mayor a la cantidad en stock el sistema mostrara el mensaje “sin stock”. Si la cantidad es correcta el sistema descontara el stock del artículo ingresado, enviara el pedido a la unidad procesadora correspondiente al artículo (cocina, bar, etc) y mostrara los datos actualizados del servicio (**). No es posible agregar artículos a una mesa cerrada.
- **Cerrar una mesa.** El mozo selecciona una mesa abierta e indica que desea cerrarla. El mozo confirma el cierre de la mesa. No es posible cerrar una mesa cerrada ni una mesa que tenga pedidos pendientes de finalización.

Antes de cerrar una mesa opcionalmente el mozo ingresa un id de cliente e indica buscar el cliente. Si no encuentra un id que corresponda se mostrara el mensaje “Cliente no valido”. Si el sistema encuentra el cliente, aplica el beneficio correspondiente y muestra nuevamente los datos actualizados del servicio. Se deberá mostrar, además: el nombre del cliente, el monto total original, un texto que describe el beneficio aplicado (“Café invitación” p.ej.) el monto del beneficio aplicado y el monto total a pagar.

- **Transferir una mesa:** El mozo selecciona una de sus mesas (abierta o cerrada) e indica que desea transferirla a otro mozo. El sistema muestra una lista con los mozos que están actualmente en el sistema (logueados), el mozo selecciona uno e inicia la transferencia. El sistema muestra una notificación al mozo destino, que debe aceptar o rechazar la transferencia de la mesa. La notificación debe indicar el número de mesa, si está abierta o cerrada y el nombre del mozo que inicio la transferencia. Si el mozo destino rechaza la transferencia la operación no tiene efecto y el sistema avisa al mozo de origen que la transferencia fue rechazada. Si acepta la transferencia, el sistema asigna la mesa al mozo destino, la muestra en su lista de mesas asignadas y la quita de la lista de mesas del mozo que inicio la transferencia y le avisa que la transferencia fue aceptada.
- **Límite de tiempo en transferencias.** Los mozos tendrán un tiempo límite para responder una transferencia. El tiempo límite comienza a contar regresivamente desde que el mozo de origen realiza la transferencia. Si el tiempo límite llega a cero, se considera que el mozo rechaza la transferencia. El tiempo límite deberá implementarse en la lógica de la aplicación, no en la interfaz de usuario. El valor del tiempo límite debe

mostrarse en todo momento en el mozo destino. El valor inicial del tiempo límite debe ser fácilmente modificable desde el código del programa.

- **Logout.** El mozo sale del sistema. No es posible salir del sistema si tiene mesas abiertas.

(**) Cuando se muestra el servicio (siempre que se selecciona una mesa abierta) se debe mostrar el monto total del servicio y cada uno de sus ítems: nombre del artículo, cantidad, precio unitario y cantidad*precio unitario. Además se debe mostrar el estado de proceso del ítem: si está en espera de ser tomado por un gestor, si ha sido tomado y por cual gestor, o si está finalizado y por cual gestor.

Aplicación procesadora de pedidos.

Esta aplicación es utilizada solo por los gestores. Los casos de uso disponibles en esta aplicación son los siguientes:

- **Ingreso:** El gestor indica su nombre de usuario y contraseña. El sistema muestra una lista con los nombres de las unidades procesadoras de pedidos definidas en el sistema (Cocina, bar, etc). El gestor selecciona una unidad procesadora. El sistema ejecuta el caso de uso gestionar pedidos.
- ***Gestionar pedidos:** Cada vez que algún mozo agrega un ítem a un servicio, deberá aparecer un nuevo pedido para todos los gestores que estén trabajando con la unidad procesadora correspondiente al artículo del ítem. En el pedido debe aparecer el nombre del artículo, la cantidad, la descripción, la mesa y el mozo que la atiende.
- **Tomar un pedido:** El gestor selecciona el pedido e indica que lo está procesando, el sistema marca que ese pedido está siendo procesado por ese gestor y lo quita de la lista de pedidos del resto de los gestores.
- **: El gestor selecciona un pedido de la lista de pedidos tomados e indica que está finalizado.** El sistema marca el pedido como finalizado y lo quita de la lista de pedidos tomados del gestor y muestra una notificación al mozo que atiende la mesa correspondiente, indicando que el pedido está finalizado. Se debe mostrar artículo, cantidad, y número de mesa.

Aplicaciones distribuidas

Pueden haber varias instancias de las aplicaciones en distintas terminales. Deben comunicarse mediante RMI.

Interfaz Gráfica

Debe usarse Swing.

Importante: La información de todas las ventanas debe obligatoriamente actualizarse de manera automática, sin necesidad de que el usuario indique que desea actualizar la información.

Requerimientos de diseño:

- 1) Maximizar la modularidad y claridad del código. Para esto utilice la métrica que dice que ningún método debería tener más código que el que se puede visualizar en una pantalla.
- 2) Minimizar la duplicación de código. Evitar métodos o porciones de código que realizan la misma tarea.
- 3) División física de las clases en al menos 2 paquetes: Clases para la interfaz de usuario y clases de la lógica del negocio.
- 4) División lógica:
 - a) Interfaz de usuario: Se encarga de capturar la información que el usuario debe proveer y presentar la información que “el sistema” provee. Evite codificar partes de la lógica del sistema (reglas de negocio) en las clases que se encargan de realizar la interfaz con el usuario.
 - b) Lógica de negocio o modelo: Se encarga de validar y almacenar la información provista por los usuarios y de implementar la lógica necesaria para proveer de información a los mismos. Evite codificar la presentación de la información en las clases que se encargan de la lógica del sistema.
- 5) La información entre la capas o subsistemas se comunicará utilizando objetos de las clases del dominio del problema.
- 6) Uso del patrón de diseño “Fachada”. La política propuesta para este patrón es ocultar a las clases de la interfaz de usuario, todas las clases de la lógica del negocio o modelo que no pertenezcan al dominio del problema.
- 7) Maximizar la aplicación del GRASP Experto para asignar las responsabilidades.
- 8) Utilizar una arquitectura M.V.C.

SE PIDE ENTREGAR

- Implementación del sistema cumpliendo con todos los requerimientos funcionales y de diseño solicitados.
- Diagramas:
 - Un diagrama de clases conceptual modelando el dominio del problema. O
 - Un diagrama de clases de diseño (o más de uno si lo considera apropiado) que incluya a todas las entidades que participan en la solución.
 - Diagrama de Paquetes.
 - Diagrama Despliegue
- Auto Evaluación: Descripción breve de aquellos requerimientos funcionales o de diseño que faltan o no funcionan correctamente y/o pueden ser mejorados y el motivo.
- Justificación de las decisiones de diseño.
- Datos de prueba: Listado impreso con el nombre de usuario, contraseña pre-cargados en el sistema.

ATENCIÓN: Los requerimientos pueden variar en el futuro. El docente le notificará en el foro de la plataforma si los hubiera.