

Ingeniería en Sistemas de Información – Programación III

Profesor Máster Carlos Carranza Blanco

Proyecto Programado (Valor 50%)

Objetivo:

Aplicar de manera práctica los conocimientos adquiridos sobre la plataforma JAVA, mediante la confección de aplicación de facturación para restaurantes, poniendo en práctica los conceptos estudiados de programación orientada a objetos, Java, JavaFX, CSS, JPA y servicios web, así como la investigación de nuevas tecnologías.

Descripción del proyecto:

Los puntos de ventas para restaurantes son unos de los sistemas más requeridos en el ámbito comercial, si bien son muchos los puntos de venta de este tipo que se ofrecen en el mercado, cada vez es mayor la exigencia de soluciones más visuales, flexibles y prácticas.

Es por ello por lo que se requiere desarrollar el proyecto con una arquitectura que impulse la escalabilidad, y le permita a la aplicación aumentar sus posibilidades en un ambiente tan variable como es el comercio. La aplicación a desarrollar debe quedar preparada para soportar, mediante una fácil adaptación, diferentes métodos de ventas con distintas tecnologías como la móvil o la web.

La idea es crear una aplicación **multilenguaje** para la venta de comidas y bebidas de un restaurante, en la cual me permita configurar los distintos salones o aposentos que posea el restaurante con las distintas distribuciones de las mesas de cada uno de estos aposentos, **sin dejar por fuera las ventas hechas en la barra**.

Para ello se debe contar con una interfaz diseñada para el uso en dispositivos táctiles, que sea muy dinámica y configurable, sin dejar de lado la robustez con la que debe contar el sistema.

Para la realización del proyecto se deberán seguir los siguientes lineamientos: el desarrollo se debe realizar en JavaFX 11 por medio de Netbeans 12.0, la cual por medio de servicios web REST debe establecer una conexión a un servidor de aplicaciones Payara, el cual a su vez debe conectarse a una base de datos Oracle 18c XE, por medio de un pool de conexión y el uso de JPA. EN CASO DE NO SEGUIR ESTAS INDICACIONES NO SE REVISARÁ EL PROYECTO.

A continuación, se especifican los requerimientos definidos de una vez en funcionales y no funcionales:

Requisitos Funcionales:

Los requisitos funcionales son aquellos que describen la operatividad del sistema, entre ellos están:

Requisitos de base de datos:

1. Se requiere definir un esquema de base de datos con el nombre de la aplicación (RestUNA) donde se creen todos los objetos de BD necesarios para el funcionamiento de la aplicación.

Requisitos del servidor de aplicaciones:

2. Se debe definir un pool de conexión con el nombre “RestPool” para gestionar la conexión con la base de datos.
3. Se debe definir un recurso jdbc llamado “jdbc/RestUNA” para darle acceso a la aplicación al pool de conexiones.
4. Se debe desarrollar una aplicación web llamada “WsRestUNA” donde se encuentre alojado un web service REST con todos los métodos necesarios para el funcionamiento de la aplicación SigeceUNA.

Requisitos generales del sistema:

5. Se requiere un módulo donde los usuarios se registren con su nombre, usuario (debe ser único), contraseña y rol (Administrativos, Cajeros y Saloneros). En cuanto a los roles de seguridad un salonero sólo podrá ingresar pedidos de los clientes, el cajero podrá realizar la labor del salonero además de facturar los pedidos, y por último el administrador tendrá acceso total al sistema.
6. El sistema debe incluir una forma de ingresar a la aplicación mediante el usuario y contraseña, según el rol del usuario autenticado el sistema debe mostrarle solo los accesos permitidos, en caso de ser salonero cuando se le muestre la pantalla de los salones o aposentos del restaurante y se seleccione la mesa a la cual se le incluirá el pedido del cliente, se debe pedir de nuevo la autenticación del salonero, quedando registrado a su usuario la orden ingresada.
7. Se deberá crear un menú para acceder a las funcionalidades del sistema, funcionalidades que estarán limitadas según rol del usuario.
8. El sistema de incluir una opción para la creación y modificación de los salones o secciones con las que cuenta el restaurante, se debe poder incluir el nombre de la sección así como la imagen de la mesa (debe poder subir cualquier imagen) con la que se quiere diseñar la distribución visual de las mesas en dicho aposento, además se debe indicar si la sección corresponde a un salón el cual requiere el diseño de la distribución de mesas o si es una sección de venta directa como una barra y se debe indicar si en dicha sección se cobra o no el impuesto de servicio.
9. El sistema debe contar con una sección donde se muestren los salones del restaurante, debe haber una forma sencilla de cambiar entre salones, un usuario administrador debe poder activar un modo de edición de los salones y agregar de forma sencilla la distribución de las mesas en dicho aposento, para ello de forma gráfica se deberá visualizar la imagen seleccionada para las mesas de dicho salón, y arrastrando dicha imagen deberán poder realizar el diseño de las mesas de cada salón, a cada mesa debe poder asignarle un nombre o número, una vez finalizado el diseño del salón debe desactivar el modo edición.
10. Una vez diseñados los salones, los saloneros deben poder seleccionar una mesa e ingresar o modificar una orden, incluyendo productos solicitados o bien eliminando alguno en caso de error. En las mesas de los salones se debe poder identificar gráficamente el estado en el que se encuentra la mesa (libre u ocupada), una mesa se debe poner como ocupada en el momento que se le agregue el primer pedido, y debe quedar libre en el momento que se facture.
11. El sistema debe contar con un mantenimiento de tipos o grupos de productos (bebidas calientes, bebidas frías, platos fuertes, entradas...), para poder realizar agrupaciones de productos para facilitar su posterior búsqueda en la pantalla de facturación o de ingreso de pedidos, además de indicar si se debe incluir en el menú de productos de acceso rápido.
12. El sistema debe contar con un mantenimiento de los productos que vende el restaurante, para ello debe indicar un nombre del producto, un nombre corto utilizado para la sección de selección de

productos en las órdenes, el precio, el grupo al que pertenece el producto, además de indicar si se debe incluir en el menú de productos de acceso rápido.

13. El sistema debe contar con una pantalla de parámetros donde se indiquen aspectos generales del sistema como idioma, porcentajes de impuesto de ventas y de servicio, porcentaje máximo de descuento que puede aplicar un cajero, nombre del restaurante que se debe indicar en las facturas emitidas, números de teléfono y cualquier otra información importante que se considere para la impresión de tiquetes o funcionamiento general del sistema.
14. El sistema debe incluir una pantalla de creación de órdenes, que se accederá cuando un salonerero seleccione una mesa en la que desea incluir una orden de un cliente, en esta pantalla se debe mostrar el salón y la mesa seleccionada, de forma opcional se puede indicar el cliente, se deben poder seleccionar los productos del menú de productos de acceso rápido, o de una búsqueda general en caso de no estar incluidos en dicho menú, e indicar la cantidad solicitada. **Esta pantalla debe poder ser accedida también desde un menú adicional, para poder ingresar las ordenes de los clientes ubicados en las secciones del restaurante que no son salones, como la barra.**
15. El menú de acceso rápido de productos se debe generar a modo de jerarquía basado en los grupos de productos creados, debe ser generado para una interfaz táctil al igual que todo el sistema, a la hora de tocar un grupo se deben desplegar los productos asociados.
16. El sistema debe contar con una pantalla para facturar los pedidos de los clientes, para ingresar a esta pantalla se puede realizar mediante tres formas, una es arrastrando una mesa de un salón hasta una sección de la pantalla destinada a la caja de facturación, otra es desde la pantalla de pedidos mediante una opción de facturar, y por último desde un menú de facturación. En caso de ingresar mediante el menú se debe dar la opción de seleccionar la orden a facturar, o bien se puede realizar una facturación rápida directa en caja. En esta pantalla se deben mostrar los productos a facturar, cantidades, precio y subtotal por producto, se debe incluir el cliente, se debe precargar si se cobra impuesto de venta y de servicio según parámetros pero puede ser modificado, se debe poder indicar un descuento, agregar o eliminar un producto, y finalmente generar el cobro, en el cual se debe indicar el monto total a cobrar, el monto recibido (en efectivo y tarjetas) y el vuelto a dar, por último se debe imprimir la factura con todos los datos de la compra, incluido el nombre del restaurante y del cliente. Se debe dar la funcionalidad de enviar por correo electrónico la factura al cliente, para lo cual se deberá solicitar al cliente el correo y enviar la factura, el correo debe tener un atractivo diseño en html con la información de la factura, además de adjuntar el pdf de la factura.
17. El sistema debe permitir controlar los cierres de caja de cada cajero, para ello cuando un cajero realice la primer factura del día se debe crear una apertura de caja a su nombre, de ahí en adelante todas las facturas que realice en el día estarán ligadas a esa apertura de caja, al finalizar su turno el cajero deberá generar un cierre de caja, donde el cajero indicará el monto en efectivo y en tarjetas con el que terminó el turno, y procederá a aplicar el cierre, una vez aplicado el cierre, el sistema imprimirá un reporte con todos los movimientos del cierre e indicará en un resumen el monto en efectivo y tarjetas que indicó el cajero y lo comparará con lo registrado en el sistema, e indicará si hay algún faltante o sobrante.
18. El sistema debe contar con un módulo de reportes, los cuales deberán ser diseñados en Jasper Report, donde se puedan generar listados de facturas entre un rango de fechas donde se visualice toda la información de dichas facturas, **otro reporte a generar es el de los cierres de caja, el cual debe permitir seleccionar el cierre de un determinado cajero para un día en particular, por último un reporte donde se pueda visualizar la cantidad de veces que se han vendido los productos del restaurante entre un rango de fechas, ordenado del producto más vendido al menos vendido.**

Requisitos No Funcionales:

1. **Un requisito indispensable es que la interfaz del sistema sea moderna y atractiva, además de ser funcional e intuitiva.**
2. Se debe de utilizar la base de datos Oracle 18c XE.
3. Como servidor de aplicaciones es indispensable el uso de Payara 5.2020.3.
4. Documentación incluyendo requerimientos y su especificación en alto nivel.
5. Se requiere que la documentación de la base de datos cuente con el modelo entidad relación.
6. Se requiere que las conexiones externas se realicen por medio de Web Services Rest.
7. Los códigos que se utilicen se deben realizar por medio de secuencias de la base de datos.
8. **El sistema debe ser multilenguaje, al menos español e inglés.**
9. Todos los correos que se envíen deben llevar una estructura atractiva en HTML.
10. Los reportes deben ser generados en el servidor y enviados al cliente para su visualización.
11. **Todos los archivos deberán ser almacenados en la base de datos.**
12. **La interfaz debe estar diseñada para dispositivos táctiles.**

Grupos de trabajo:

Estarán formados por 3 estudiantes, NO se permitirán proyectos individuales. Si queda algún estudiante sin grupo, el profesor le asignará el grupo correspondiente.

Fecha de entrega.

La documentación del proyecto y el diseño de la base de datos deberán ser entregados el 20/09/2021
Se realizará la entrega del proyecto según fecha indicada en el cronograma de curso (15/11/2021).

Material que entregar al profesor

Cada grupo deberá subir al aula virtual el código fuente, ejecutables del programa y la documentación.

Notas:

Los puntos del desarrollo del sistema se calificarán según se cumplan los requisitos propuestos del sistema.

El código del programa debe estar debidamente comentado.

El día de la entrega se presentará al profesor el sistema en ejecución.

Cada miembro del grupo deberá defender el código del programa de forma individual.

La defensa se calificará de forma individual a cada miembro del grupo.

Se tomará en cuenta la apariencia gráfica del proyecto.

En caso de similitudes evidentes entre dos o más proyectos, estos serán anulados en su totalidad.

Si se comprueba por parte del profesor que alguna parte del proyecto ha sido sustraída de internet o algún otro medio, está será anulada.

Calificación:

Aspecto	Valor	Desglose	%	Nota	Observaciones
Documentación	7	Redacción, Presentación y Ortografía	2		
		Planteamiento del problema y solución adoptada	2		
		Requerimientos	3		
Base de datos	5	Estándares aplicados a la base de datos	1		
		Modelo entidad relación	3		
		Script de Base de Datos	1		
Desarrollo del sistema	78	Innovación en el desarrollo	5		
		Manejo de validaciones y mensajes	2		
		Registro y autenticación de usuarios(req. 5 y 6)	4		
		Menú principal (req. 7)	1		
		Mantenimiento de salones (req. 8)	3		
		Mantenimiento de grupos de productos (req. 11)	3		
		Mantenimiento de productos (req. 12)	3		
		Parámetros generales (req. 13)	3		
		Diseño y funcionalidad de los salones (rea. 9 y 10)	8		
		Generación de órdenes (req. 14 y 15)	11		
		Facturación (req. 15 y 16)	14		
		Cierres de caja (req. 17)	8		
		Reportes (req. 18)	8		
		Apariencia de la aplicación e idiomas	5		
Defensa	10	-	10		
Totales	100	-	100		