

Objetivos

Unidad 1: Construcción de la interfaz gráfica

OE1.1. Utilizar una arquitectura de tres capas para el desarrollo de un programa de computador, repartiendo de manera adecuada las responsabilidades entre la interfaz de usuario, el control de la interfaz y el modelo. El estudiante deberá poder explicar la importancia de mantener separadas las clases de estos tres dominios.

OE1.2. Construir las clases que implementan una interfaz de usuario.

OE1.3. Aplicar la técnica de descomposición de requerimientos para cumplir con la funcionalidad de un programa de computador.

Unidad 2: Persistencia, Manejo de Archivos de Texto y Algoritmos de Ordenamiento y Búsqueda

OE2.1 Manipular archivos de texto para implementar requerimientos del cliente relacionados con persistencia.

OE2.2 Leer entradas e imprimir salidas de programas que interactúan directamente con archivos de texto y no con usuarios finales.

OE2.3 Hacer persistir el estado del modelo de solución del problema durante la ejecución de un programa y restaurarlo cuando se requiera usando la técnica de serialización.

OE2.4 Implementar algoritmos clásicos de ordenamiento de datos en estructuras de datos lineales y aplicarlos en la solución de un problema.

OE2.5 Implementar algoritmos clásicos de búsqueda de información en estructuras de datos lineales y aplicarlos en la solución de un problema.

OE2.6 Hacer uso de las interfaces Comparable y Comparator para definir relaciones de orden total sobre objetos.

OE2.7 Calcular el tiempo de ejecución de un algoritmo por medio de las operaciones de tiempo del sistema

OE2.8 Implementar métodos que permitan generar muestras con datos aleatorios.



1. DESCRIPCIÓN

La cucharita es un restaurante con varias sedes en la ciudad de Cali que es conocido por ofrecer el popular corrientazo. Este restaurante quiere incluir TI en su negocio para administrar sus negocios dado que esperan un crecimiento en 2022 por la fundación de dos sedes más del restaurante.

Desde su apertura, los administradores han tenido que registrar sus ventas de forma manual en un cuaderno de contabilidad, lo que genera inconvenientes en términos de agilidad en la operación, consolidación de reportes, validación de los datos y seguridad, entre otros, ocasionando que todos los procesos de la compañía sean demorados.

Por lo tanto, se hace indispensable el desarrollo de un sistema de software que permita llevar a cabo la administración de pedidos y generación de reportes básicos del restaurante de forma inmediata.

2. ROLES

El sistema a implementar agrega valor a la compañía en la medida en la que se puedan agilizar **los procesos de administración de empleados, de inventario, de pedidos y la carta**. Por lo tanto el programa será manejado por el personal del restaurante permitiéndoles administrar justamente los empleados, el inventario, los pedidos y la carta.

El programa debe tener un usuario administrador que es el primer empleado registrado en el sistema, debe estar desde la primera ejecución y que puede usar todos los módulos del programa, entre ellos el módulo de empleados para registrar a todo el personal.

3. MÓDULOS DEL PROGRAMA

El programa se puede dividir en 4 módulos: módulo de autenticación, módulo de personal, módulo de inventario y el módulo de pedidos. Cada uno de los módulos nos va a permitir visualizar, agregar y eliminar elementos propios de cada módulo.

A. Módulo de autenticación

Permite al empleado usar los demás módulos. El empleado abre el programa y se va a encontrar con una ventana de inicio de sesión. El empleado deberá usar su cédula y contraseña. Si la cédula y contraseña concuerdan, el programa le dará acceso a los demás módulos para poder administrar personal, inventarios o pedidos.

B. Módulo de personal



Este módulo muestra listados todos los empleados del restaurante.

Adicionalmente va a permitir al programa registrar a un empleado usando su nombre, cédula, fecha de nacimiento y una contraseña. Una vez que la persona esté agregada como empleado podrá ingresar al sistema usando el módulo de autenticación.

El módulo adicionalmente le va a permitir cambiar la contraseña del usuario que inició sesión.

C. Módulo de inventario

Este módulo es muy importante para el modelo de negocio. Aquí usted debe poder ver en una tabla todos los ingredientes con los que cuenta el restaurante en su inventario y además debe poder administrarlo, esto es, debe poder agregar los ingredientes que quiera, puede aumentar o disminuir la cantidad de cualquier ingrediente de la tabla y también puede eliminar el ingrediente.

Para agregar ingredientes debe registrar el nombre del ingrediente, las unidades en las que mide la cantidad (mililitros para bebidas, gramos para alimentos como lentejas, unidad para alimentos como papas o pan, kilogramos para las carnes).



Tenga en cuenta que en el inventario se basan los pedidos que los clientes puedan hacer. Cada vez que un empleado registra un pedido, los ingredientes que se usan para ese pedido se restan del inventario.

D. Módulo de carta



Una vez haya un inventario, el usuario podrá crear una carta registrando platillo por platillo. Por simpleza puede usar Combo 1, Combo 2, Combo 3, etc como nombres de los platillos de la carta.

El usuario puede entonces definir cada uno de los platillos. Para eso, el usuario crea uno usando el nombre del platillo, la lista de ingredientes que tiene el platillo con sus respectivas cantidades y el precio. Estos ingredientes DEBEN estar registrados en el inventario antes de crear el platillo.

Por ejemplo, el usuario quiere registrar el combo 1 que será una hamburguesa con papas y gaseosa. Entonces inscribe el combo 1 y va agregando uno a uno los ingredientes y cantidades: 250 gramos de carne, 50 gramos de lechuga, 0.25 und de

tomate, 1 und de pan de hamburguesa, 2 und de papa y 100 mililitros de gaseosa. Estos ingredientes fueron registrados antes de la creación del combo 1. Finalmente, se pone que costará \$14.900 y el platillo de la carta va a quedar listo.

NO es necesario que haya una cantidad mayor a cero en el inventario para poder registrar el platillo, pero sí tiene que estar presente en el inventario

E. Módulo de Pedidos

Este **módulo le permite ver TODOS los pedidos**, no sólo los que están pendientes.

Adicionalmente le **permite al empleado registrar un pedido**. El cliente tendrá a **disposición una carta** y podrá escoger los platillos que él desee.

Cuando el usuario registra un pedido, este queda con un código (autogenerado UUID), la lista de platillos que pidió con su respectiva cantidad, un estado del pedido y la *fecha del pedido*.

Por ejemplo si el cliente pidió 2 “combo 1” y se efectúa el pedido, del inventario se restan 2 veces los ingredientes de ese combo: se restan 500 gramos de carne, 10 gramos de lechuga, 0.5 unidades de tomate, 2 unidades de pan, 4 unidades de papa y 200 mililitros de gaseosa.

Si al momento de hacer el pedido, los ingredientes son insuficientes, el programa lo debe decir. NO se admite que el inventario pueda tener números negativos.

El programa debe **permitir cambiar el estado de un pedido** entre PENDIENTE, EN PROCESO y ENTREGADO.

4. ALMACENAMIENTO

El programa debe guardar toda su información a través de la serialización de sus objetos en archivos. Este guardado debe ser transparente para el usuario del programa, es decir, cada vez que se registre o actualice información, ésta se guardará en los archivos serializados.

5. REPORTES

Se requiere que el programa genere también los siguientes dos reportes:

- A. **Un listado de empleados consolidando el número de pedidos entregados y la suma de los valores de dinero de dichos pedidos.**
- B. **Un listado de los platillos de la carta consolidando el número de veces que se pidió y la cantidad de dinero total que se pagó por ellos.**

Para crear ambos reportes, el programa le permite seleccionar una fecha de inicio y una fecha final. Por defecto la fecha de inicio y final será la del día actual por lo cual el reporte mostrará la información del día.

6. ORDENAMIENTO

Su programa debe implementar al menos 2 algoritmos de ordenamiento de los 3 vistos en clase (burbuja, selección e inserción). Debe hacer por lo menos 1 ordenamiento utilizando Comparable o Comparator. Use esos algoritmos para ordenar la lista de ingredientes, empleados, carta o pedidos.

Por ejemplo, es buena idea que la lista de pedidos quede organizada mostrando primero los que están en estado SOLICITADO, EN PROCESO y luego ENTREGADO.

Entregables

1. **[10%]** Especificación de Requerimientos Funcionales. Liste los requerimientos funcionales del programa. Para esto puede dividir los requerimientos por secciones (sección de pedidos, cliente, etc). Numere adecuadamente los requerimientos. Cada uno debe ser un párrafo donde se describa el requerimiento, por ejemplo:
AUTH1: El empleado debe poder identificarse en el sistema por medio del inicio de sesión. El empleado tuvo que ser registrado desde antes para que el programa le permita el ingreso.
Aquí la nomenclatura AUTH hace referencia al módulo de autenticación y el 1 a la numeración.
2. **[5%]** Luego de levantar los requerimientos realice una propuesta gráfica simple, que incluya todas las ventanas que usted va a realizar con la posición de los elementos gráficos que van en cada una de las ventanas. Esta fase de diseño preliminar le ayudará a pensar en cómo quedará la aplicación antes de comenzar a desarrollarla. Por ejemplo, [aquí](#) puede ver la propuesta simple o wireframe de la página web de Facebook
3. **[15%]** Diseño completo del diagrama de clases, incluyendo el paquete del modelo y de la interfaz con el usuario. Este debe ser diseñado utilizando Visual Paradigm o Lucidchart. El diagrama de clase debe ser legible y en blanco y negro.
4. **[60%]** Implementación completa y correcta del modelo y la ui.
5. **[10%]** Usted debe entregar el enlace del repositorio en GitHub o GitLab con los elementos anteriores. Su repositorio debe corresponder con un proyecto de eclipse. Cada uno de los integrantes del grupo que la desarrolla debe tener al menos 10 commits con diferencia de mínimo 1 hora entre cada uno de ellos. En el repositorio o proyecto de eclipse debe haber un directorio llamado **docs/** en el cual deberán ir cada uno de los documentos del diseño. **El último commit de la tarea debe tener un tag que diga: V1.0.0.**

Los requerimientos funcionales, el diagrama de clases y el diseño de casos de prueba deben entregarse en un mismo archivo en formato **pdf**, bien organizado por secciones y títulos, con hoja de portada.