

Requerimientos Funcionales
Tarea Integradora I

Integrantes:

Jhorman Germán Mera Escobar

Paula Andrea Trujillo Mejía

Curso:

Algoritmos y Programación II

Docente:

Juan Manuel Reyes García

Universidad Icesi
Cali - Valle del Cauca

19 abril 2021

Requerimientos Funcionales - Tarea Integradora I

El sistema debe estar en la capacidad de:

1. **Req1. Gestionar** los tipos de productos e ingredientes con su nombre. Todo producto tiene un nombre, un tipo (plato principal, adicional o bebida -pero estos pueden variar-), un conjunto de ingredientes, y tamaños con sus respectivos precios.

Req1.1. Crear un nuevo producto con nombre, tipo (plato principal, adicional o bebida u algún otro), conjunto de ingredientes, tamaño y precio. Los ingredientes podrán ser cero (0) o más, y deben poderse seleccionar de ingredientes previamente agregados al programa.

Req1.2. Actualizar un producto que se ha creado con anterioridad. Se permite hacer cambios en su nombre, tipo (plato principal, adicional o bebida u algún otro), conjunto de ingredientes, tamaño o precio.

Req1.3. Eliminar un objeto (producto, ingrediente, tipo de producto) de los que se ha creado con anterioridad, siempre y cuando este objeto no esté asociado con ningún otro objeto.

Req1.4. Deshabilitar un objeto (producto, ingrediente, tipo de producto). Esto implica que ese objeto no estará disponible de ahí en adelante, se desaparecerá de la lista de objetos disponibles.

2. **Req2. Gestionar** los clientes, estos deben tener nombre, apellidos, un número de identificación (opcional), dirección, teléfono, y un campo de observaciones. No se permite que haya clientes con el mismo número de identificación, el programa no dejará registrar a ninguna persona si este es el caso.

Req2.1. Agregar un nuevo cliente.

Req2.2. Actualizar la información de clientes existentes.

Req2.3. Eliminar un cliente.

Req2.4. Deshabilitar un cliente.

3. **Req3. Gestionar** los empleados, estos deben tener nombre, apellidos y número de identificación. No se permite que haya empleados con el mismo número de identificación, el programa no dejará registrar a ninguna otra persona si este es el caso.

Req3.1. Agregar un nuevo empleado.

Req3.2. Actualizar la información de empleados existentes.

Req3.3. Eliminar un empleado.

Req3.4. Deshabilitar un empleado.

4. **Req4. Gestionar** los usuarios, todo usuario también es un empleado, por consecuente, estos deberán tener igualmente un nombre, apellidos y número de identificación. No se permite que haya usuarios con el mismo número de identificación, el programa no dejará registrar a ninguna persona si este es el caso.

Req4.1. Agregar un nuevo usuario.

Req4.2. Actualizar la información de usuarios existentes.

Req4.3. Eliminar un usuario.

Req4.4 Deshabilitar un usuario

5. **Req5. Organizar** la lista de clientes siempre alfabéticamente descendente por apellido y nombre, por tanto, cada vez que se agrega un nuevo cliente, este debe insertarse de forma ordenada (se debe tener en cuenta que insertar de forma ordenada no es lo mismo que agregar y luego ordenar).
6. **Req6. Registrar** un pedido, el cual debe tener un código (autogenerado) de pedido, un estado, un listado de productos, la cantidad por cada uno, el cliente que los solicita, el empleado que lo entrega, la fecha y hora de la solicitud (de tipo Date), y las observaciones que pueda tener.
7. **Req7. Tomar** la fecha y hora de la solicitud, de la fecha y hora actual del sistema al momento de haber registrado un pedido.
8. **Req8. Cambiar** el estado de un pedido entre SOLICITADO, EN PROCESO, ENVIADO y ENTREGADO. También es importante tener en cuenta que se puede cambiar el estado del pedido hacia adelante (por ejemplo: de SOLICITADO a EN PROCESO, o de SOLICITADO a ENVIADO) pero no hacia atrás.
9. **Req9. Visualizar**, para cualquier objeto (productos, pedidos, clientes, usuarios, etc.) en cada formulario un listado de ellos, con columnas mostrando sus principales campos.
10. **Req10. Modificar** un registro si se hace doble click en la fila en la que se encuentra.

11. **Req11. Incluir** en todo objeto de las clases del modelo, dos campos internos referenciados a un objeto usuario, uno al usuario que lo creó y otro al último usuario que lo modificó.
12. **Req11. Guardar** toda la información del programa 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, esta se guardará en los archivos serializados.
13. **Req13. Generar** un archivo csv de pedidos, con una fila por cada pedido, con los datos del nombre, dirección y teléfono del cliente que lo solicitó, el nombre del empleado que lo entrega, estado del pedido y además de los demás datos del pedido como la fecha y hora, y las observaciones, debe tener tres columnas por cada producto del pedido con el nombre, la cantidad y el valor unitario del producto.

Req13.1. Organizar las columnas de los productos al final porque cada pedido tiene una cantidad diferente de productos.

Req13.2. Generar el reporte en un rango de fechas y hora, es decir, al momento de generarlo se pregunta la fecha y hora inicial, y la fecha y hora final del reporte. (Por defecto, el valor en la fecha y hora inicial son las 00:00 del día actual, y la fecha final debe ser por defecto 23:59 del día actual).

Req13.3. Ordenar el reporte por fecha y hora del pedido ascendente.

Req13.4. Preguntar al usuario en el momento en que se genera el reporte cuál separador debe usar. Por defecto debe estar en punto y coma (;).

Req13.5. Colocar un encabezado con los nombres de las columnas, divididos por el separador elegido con anterioridad por el usuario. El encabezado no debe incluir las columnas de los productos

14. **Req14. Generar** un listado de empleados consolidando el número de pedidos entregados y la suma de los valores de dichos pedidos.

Req14.1. Clasificar este listado en un rango de fecha y hora, inicial y final. Por defecto, el valor en la fecha y hora inicial son las 00:00 del día actual, y la fecha final debe ser por defecto 23:59 del día actual.

Req14.2. Agregar al final del reporte una fila que totalice las columnas numéricas.

15. **Req15. Generar** un listado de los productos consolidando el número de veces que se pidió y la cantidad de dinero total que se pagó por todos los productos (número de veces por precio del producto).

Req15.1. Clasificar este listado en un rango de fecha y hora, inicial y final. Por defecto, el valor en la fecha y hora inicial son las 00:00 del día actual, y la fecha final debe ser por defecto 23:59 del día actual.

Req15.2. Agregar al final del reporte una fila al final que totalice las columnas numéricas.

16. **Req16. Tener** una opción que permita listar en pantalla todos los productos en orden de precio ascendente.

17. **Req17. Tener** una opción que permita listar en pantalla todos los ingredientes en orden alfabético descendente.

18. **Req18. Tener** una opción que permita buscar eficientemente un cliente dado un nombre e indicar el tiempo que tardó la búsqueda. Esta opción debe estar en el campo que permite realizar los pedidos.

19. **Req19. Implementar** al menos 2 algoritmos de ordenamiento de los 3 vistos en clase (burbuja, selection e insertion).

20. **Req20. Implementar** por lo menos 1 ordenamiento utilizando Comparable y 1 ordenamiento utilizando Comparator, empleando en ambos casos el sort de Collections o de Arrays.

21. **Req21. Importar** datos de un archivo csv con información de clientes.

22. **Req22. Importar** datos de un archivo csv con información de productos.

23. **Req23. Importar** datos de un archivo csv con información de pedidos.

24. **Req24. Generar** un archivo csv con al menos 1000 datos para probar los productos y con los pedidos.

25. **Req25. Generar** un archivo csv con más de 1000 datos para probar los clientes.