Requerimientos funcionales: Tarea Integradora 1

El programa debe estar en capacidad de:

RF1: ***Crear*** un objeto (producto, ingrediente, tipo de producto). El programa no dejará crear ningún objeto si algún campo está vacío.

RF2: ***Actualizar*** la información que provee los objetos (producto, ingrediente, tipo de producto).

RF3: ***Eliminar*** un objeto (producto, ingrediente, tipo de producto, etc.) siempre que no esté referenciado desde otro objeto. Por ejemplo, un ingrediente podrá ser eliminado siempre que no haya un producto que esté asociado a dicho ingrediente.

RF4: ***Deshabilitar*** un objeto siempre será posible y esto implicará que ese producto no estará disponible, para ser referenciado desde otro objeto. Por ejemplo, si se deshabilita un producto, este ya no se mostrará entre los productos disponibles al momento de hacer un pedido.

RF5: ***Gestionar tipos productos*** en donde todo tipo producto tiene un nombre (String), un creador(User), un editor(User) y un estado(Boolean).

RF5.1: ***Crear*** un tipo de producto que ofrece el restaurante.

RF5.2: ***Actualizar*** un tipo de producto de los que ofrece el restaurante, por si hay algún cambio.

RF5.4: ***Deshabilitar*** un tipo de producto por si no está disponible.

RF5.3: ***Eliminar*** un tipo producto por si ya no se seguirá ofreciendo.

RF6: ***Gestionar productos*** en donde todo producto tiene un nombre(String) , un tipo de producto (ProductType) , un conjunto de ingredientes, y tamaños (ProductSize) con sus respectivos precios(Double), un creador(User), un editor(User) y un estado(Boolean).

RF6.1: ***Crear*** un producto que ofrece el restaurante.

RF6.2: ***Actualizar*** un producto de los que ofrece el restaurante, por si hay algún cambio.

RF6.4: ***Deshabilitar*** un producto por si no está disponible.

RF6.3: ***Eliminar*** un producto por si ya no se seguirá ofreciendo.

RF7: ***Gestionar ingredientes*** en donde todo ingrediente tiene un nombre(String), un creador(User), un editor(User) y un estado(Boolean).

RF7.1: ***Crear*** los ingredientes que ofrece el restaurante.

RF7.2: ***Actualizar*** los ingredientes de los que ofrece el restaurante, por si hay algún cambio.

RF7.3: ***Deshabilitar*** los ingredientes por si no están disponibles.

RF7.4: ***Eliminar*** los ingredientes por si ya no se seguirán ofreciendo.

RF8: ***Gestionar clientes*** ellos deben tener nombres(String), apellidos(String), un número de identificación (int), dirección(String), teléfono(String), un campo de observaciones(String) al igual que un creador(User), un editor(User) y un estado(Boolean).

RF8.1: ***Crear*** un cliente. El número de identificación no puede ser repetido en ningún otro cliente o persona, el programa no dejará registrar ninguna persona si este es el caso. Al agregar un cliente, este se agrega ordenadamente.

RF8.2: ***Actualizar*** la información del cliente. Si actualiza el ID, y hay otra persona con el mismo identificador, el programa no le dejará editar la información. SI se actualiza el nombre o apellido, el programa ordenará de nuevo y meterá al cliente ordenadamente.

RF8.3*: Deshabilitar* un cliente.

RF8.4: ***Eliminar*** un cliente.

RF9: ***Gestionar empleados*** ellos al igual que los clientes deben tener nombres(String), apellidos(String), un número de identificación(int) un creador(User), un editor(User) y un estado(Boolean).

RF9.1: ***Crear*** un empleado por si fue contratado para trabajar en el restaurante. El número de identificación no puede ser repetido en ningún otro cliente o persona, el programa no dejará registrarlo si este es el caso.

RF9.2: ***Actualizar*** la información de un empleado. Si actualiza el ID, y hay otra persona con el mismo identificador, el programa no le dejará editar la información.

RF9.3: ***Deshabilitar*** un empleado por si no se encuentra trabajando actualmente.

RF9.4: ***Eliminar*** un empleado por si fue despedido.

RF10: ***Gestionar usuarios*** todo usuario es empleado, por lo tanto, también tienen nombres (String) , apellidos(String), un número de identificación(int) , un nombre de usuario(String), una contraseña para ingresar al sistema (String) , un creador(User), un editor(User) y un estado (Boolean).

Al crearse un usuario, el programa lo crea directamente en la tabla de empleados.

RF10.1: ***Crear*** un nuevo usuario con sus datos personales (nombre, apellidos y el ID) además con un nombre de usuario único, foto y una contraseña. Si el sistema no tiene ningún usuario, se le habilitará la opción de registrar usuario, después de el primer usuario creado, esta opción estará deshabilitada y solo podrá crear un usuario desde el CRUD. El número de identificación y el nombre de usuario no pueden ser repetidos en ningún usuario o persona, el programa no dejará registrar ninguna persona si este es el caso.

RF10.2: ***Actualizar*** la información del usuario, por si cambió su nombre, su apellido (el número de identificación no se puede actualizar) su nombre de usuario o su contraseña. Si actualiza el ID o el nombre de usuario, y hay otra persona con el mismo identificador y/o nombre de usuario, el programa no le dejará editar la información. Si un usuario edita su información, el mismo programa actualizará esa información en todas las partes donde ha sido referenciado.

RF10.3: ***Eliminar*** un usuario por si ha dejado de pertenecer al restaurante. No se puede eliminar un usuario si está activo, es decir, está logeado en la aplicación.

RF10.4*: Deshabilitar* un usuario mientras esté inactivo del restaurante.

RF11: ***Ordenar*** la lista de clientes alfabéticamente descendente por apellido y nombre, por lo tanto, cada vez que se agrega un nuevo cliente, este debe insertarse de forma ordenada.

RF12: ***Registrar pedidos*** que tienen un código, 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 , y las observaciones que pueda tener. El programa debe tomar la fecha y hora de la solicitud, de la fecha y hora actual del sistema

RF13: ***Cambiar el estado de un pedido*** entre SOLICITADO, EN PROCESO, ENVIADO y ENTREGADO. Es importante tener en cuenta, que solo se puede ir hacia adelante, es decir, solo se puede pasar de "solicitado" a "en proceso" y no hacia atrás.

RF14: ***Listar*** cualquier objeto (productos, pedidos, clientes, usuarios, etc.), donde el programa debe tener la posibilidad, de visualizar un listado de ellos, con columnas mostrando sus principales campos

RF15*: Modificar un registro* si se hace doble clic en la fila en la que se encuentra. Todo objeto, de las clases del modelo, debe tener dos campos internos referenciados a un objeto usuario, uno al usuario que lo creó y otro al último usuario que lo modificó.

RF16: ***Guardar 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, esta se guardará en los archivos serializados

RF17: ***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 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. Las columnas de los productos van al final porque cada pedido tiene una cantidad diferente de productos

RF18: ***Ordenar archivo csv***, debe poder generarse 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. El reporte debe estar ordenado por fecha y hora del pedido ascendente

RF19: ***Exportar información*** que irá al archivo csv generado, para eso también se le preguntará al usuario cuál es el separador que se utilizará, aunque por defecto debe tener puesto en ese campo el valor punto y coma (;) . La primera línea del archivo debe tener los nombres de las columnas separadas también por dicho separador, menos las columnas de los productos.

RF20: ***Generar listado de empleados*** consolidando el número de pedidos entregados y la suma de los valores de dichos pedidos. Esto 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. Al final se debe agregar una fila que totalice las columnas numéricas.

RF21: ***Generar listado de 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). Este reporte también debe solicitar un rango de fechas inicial y final igual que el anterior, con los mismos valores por defecto. Estas fechas y horas siempre se podrán cambiar. Al final se debe agregar una fila que totalice las columnas numéricas.

RF22: ***Listar*** en pantalla todos los productos en orden de precio ascendente.

RF23: ***Listar*** en pantalla todos los ingredientes en orden alfabético descendente. Existe un botón que permite ordenar la lista cuando el usuario lo desee, y a la vez que se organiza el ingrediente, el código autogenerado también lo hará.

RF24: ***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. Esta opción está disponible de manera indirecta en la interfaz del cliente con un search filter.

RF25: ***Importar*** datos de un archivo csv con información de clientes.

RF26: ***Importar*** datos de un archivo csv con información de productos.

RF27: ***Importar*** datos de un archivo csv con información de pedidos.

RF28: ***Generar*** un archivo csv de prueba con al menos 1000 datos para probar los productos

RF29: ***Generar*** un archivo csv de prueba con al menos 1000 datos para probar los pedidos

RF30: ***Generar*** un archivo csv de prueba para los clientes.