

**TAREA INTEGRADORA 1: SISTEMA DE PEDIDOS**

**LAURA DANIELA MARTÍNEZ ORTIZ**

**ARIEL EDUARDO PABÓN BOLAÑOS**

**29 DE MARZO DE 2021**

**ALGORITMOS Y PROGRAMACIÓN II**

**UNIVERSIDAD ICESI**

## **1. REQUERIMIENTOS FUNCIONALES**

### **RF 1. Gestionar los productos del restaurante.**

- **RF 1.1. Agregar un producto.** Se le pedirá al usuario el nombre, el tipo de producto -plato principal, adicional o bebida-, el conjunto de ingredientes -podrán ser cero (0) o más-, el tamaño del producto y el precio para agregar un nuevo producto dentro del programa. Deben poderse seleccionar de ingredientes previamente agregados al programa.
- **RF 1.2. Actualizar el producto.** Se le permitirá al usuario cambiar el nombre, el tipo de producto, el conjunto de ingredientes, el tamaño del producto y el precio para actualizar un producto dentro del programa.
- **RF 1.3. Eliminar el producto.** Se le preguntará al usuario si quiere eliminar el producto existente para borrarlo dentro del programa, siempre y cuando este no esté referenciado desde otro objeto.
- **RF 1.4. Deshabilitar el producto.** Se le preguntará al usuario si quiere deshabilitar el producto existente para hacerlo inaccesible -no disponible- dentro del programa.

### **RF 2. Gestionar la información del restaurante.**

- **RF 2.1. Gestionar los tipos de productos.** El usuario debe poder crear nuevos tipos de productos y cambiar los que existen dentro del programa.
- **RF 2.2. Gestionar los ingredientes.** El usuario debe poder crear nuevos ingredientes y cambiar los que existen dentro del programa.
- **RF 2.3. Gestionar a los clientes.** El usuario debe poder manejar la información de los clientes como su nombre, sus apellidos, su número de identificación -opcional-, su dirección, su teléfono y las observaciones que pueda tener. La lista de clientes tiene que estar ordenada de forma alfabética

descendente por apellido y por nombre, y cada vez que se agregue un nuevo elemento se debe insertar de forma ordenada.

- **RF 2.4. Gestionar a los empleados.** El programa debe manejar la información de los empleados como su nombre, sus apellidos y su número de identificación.
- **RF 2.5. Gestionar el usuario.** El usuario debe poder manejar su información dentro del programa, como su nombre, sus apellidos, su número de identificación, su nombre de usuario y su contraseña del sistema. **Todo usuario es empleado del restaurante.**
- **RF 2.6. Gestionar los pedidos.** El usuario debe poder gestionar la información de los pedidos, como su código -el cual es autogenerado-, su estado -debe poder ser cambiado entre **SOLICITADO, EN PROCESO, ENVIADO** y **ENTREGADO** de forma progresiva únicamente-, un listado de productos y la cantidad de cada uno, el cliente que los solicita, el empleado que lo entrega, la fecha y hora de la solicitud y las observaciones que se puedan tener.

**RF 3. Visualizar la información del restaurante.** El programa debe tener la posibilidad de visualizar un listado de los elementos del restaurante en columnas, mostrando sus principales campos.

- **RF 3.1. Visualizar los productos.** Mostrar nombre, el tipo de producto, el conjunto de ingredientes, el tamaño del producto y el precio. Se deben poder listar en orden de precio ascendente.
- **RF 3.2. Visualizar los ingredientes.** Se deben poder listar en orden alfabético descendente.
- **RF 3.3. Visualizar a los clientes.** Mostrar nombre, apellidos, número de identificación -opcional-, dirección, teléfono y las observaciones que pueda tener.

- **RF 3.4. Visualizar a los empleados.** Mostrar nombre, sus apellidos y su número de identificación.
- **RF 3.5. Visualizar a los usuarios.** Mostrar nombre, sus apellidos, su número de identificación y su nombre de usuario.
- **RF 3.6. Visualizar los pedidos.** Mostrar el código de pedido, el estado, el listado de productos y la cantidad de cada uno, el cliente que los solicita, el empleado que lo entrega, la fecha y hora de la solicitud y las observaciones que se puedan tener.
- **RF 3.7. Visualizar la modificación de los elementos.** Mostrar quién creó el elemento y quién fue la última persona que lo modificó a la hora de cambiar sus datos.

**RF 4. Guardar toda la información.** Cada vez que se registre o actualice la información, el programa lo guardará en archivos serializados.

**RF 5. Exportar un reporte de los pedidos.** El usuario debe poder generar un archivo .csv de pedidos, con una fila por cada uno de ellos que contenga el nombre, la dirección, el teléfono del cliente que lo solicitó, el nombre del empleado que lo entrega, tres columnas por cada producto del pedido con el nombre, la cantidad y el valor unitario del producto, además de los demás datos del pedido como la fecha y la hora, y las observaciones.

- **RF 5.1. Organizar por rango de horas.** El reporte se debe generar de acuerdo con el rango de horas establecidas por el usuario -fecha y hora inicial y final del reporte, por defecto 00:00 y 23:59-, y debe estar ordenado por fecha y hora del pedido ascendente.
- **RF 5.2. Preguntar por separador.** Se le debe preguntar al usuario cuál es el separador que desea usar para organizar la información. Por defecto este es el punto y coma.

- **RF 5.3. Definir nombre de las columnas.** La primera línea del archivo debe tener los nombres de las columnas separadas también por dicho separador, menos las columnas de los productos.

**RF 6. Exportar un reporte de los empleados.** El usuario debe poder generar un archivo .csv de empleados, consolidando el número de pedidos entregados y la suma de los valores de dichos pedidos.

- **RF 6.1. Organizar por rango de horas.** El reporte se debe generar de acuerdo con el rango de horas establecidas por el usuario -fecha y hora inicial y final del reporte, por defecto 00:00 y 23:59-, y debe estar ordenado por fecha y hora del pedido ascendente.
- **RF 6.2. Preguntar por separador.** Se le debe preguntar al usuario cuál es el separador que desea usar para organizar la información. Por defecto este es el punto y coma.
- **RF 6.3. Agregar una fila que totalice las columnas numéricas.**

**RF 7. Exportar un reporte de los productos.** El usuario debe poder generar un archivo .csv de productos, consolidando el número de veces que se pidió y la cantidad de dinero total que se pagó por todos los productos, es decir, el número de veces por el precio del producto.

- **RF 7.1. Organizar por rango de horas.** El reporte se debe generar de acuerdo con el rango de horas establecidas por el usuario -fecha y hora inicial y final del reporte, por defecto 00:00 y 23:59-, y debe estar ordenado por fecha y hora del pedido ascendente.
- **RF 7.2. Preguntar por separador.** Se le debe preguntar al usuario cuál es el separador que desea usar para organizar la información. Por defecto este es el punto y coma.
- **RF 7.3. Agregar una fila que totalice las columnas numéricas.**

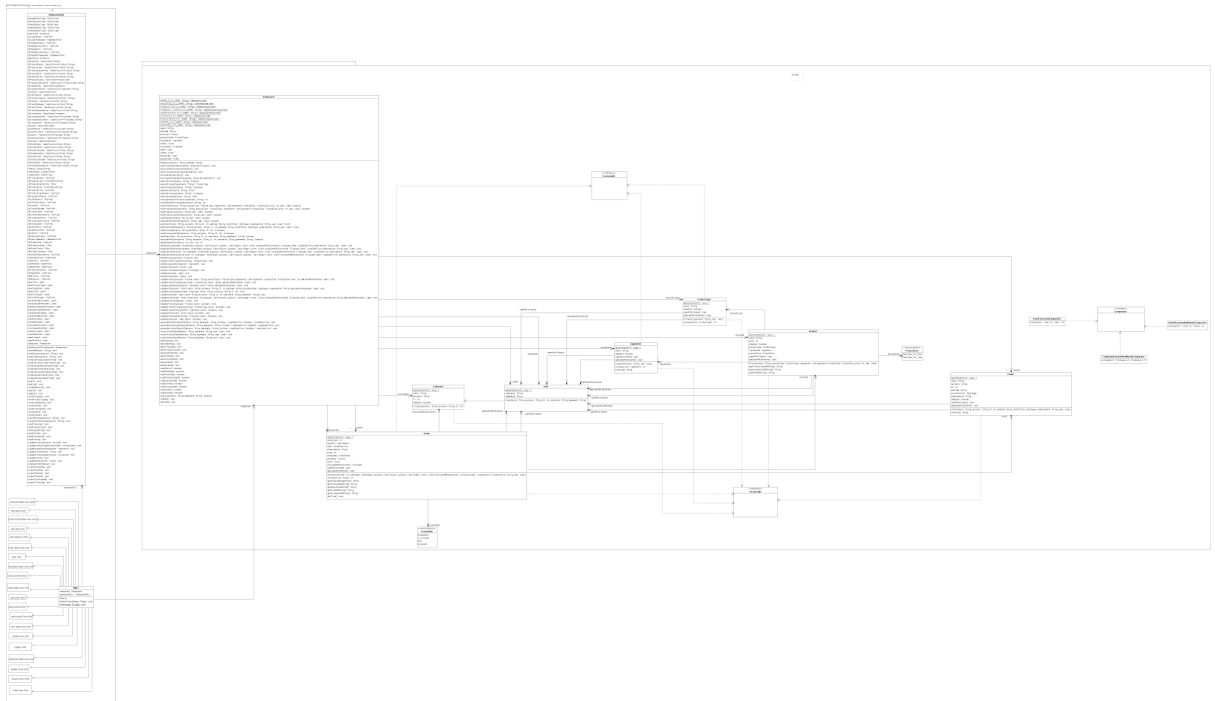
**RF 8. Buscar a un cliente.** Dado el nombre del cliente, realizar una búsqueda eficiente e indicar el tiempo que tardó la búsqueda.

**RF 9. Importar datos de un archivo.** El programa debe permitir importar datos de un archivo .csv.

- **RF 9.1** Importar datos con información de clientes.
- **RF 9.2** Importar datos con información de productos.
- **RF 9.3** Importar datos con información de pedidos.

**RF 10. Modificar la información del restaurante.** Debe tener la posibilidad de modificar la información del restaurante, a través de la GUI, si se hace doble clic en la fila en la que se encuentra.

## 2. DIAGRAMA DE CLASES



La imagen completa está en la carpeta docs dentro del proyecto.