Tarea Integradora 1

Requerimientos:

* **Gestionar** (crear, actualizar, eliminar y deshabilitar) todos los productos que ofrece.

**Productos:**

**Nombre**

**Tipo-Enum**

**Ingredientes-Array de String**

**Tamaños -Enum**

**Precio según el tamaño-Array Int**

**Disponibilidad Boolean**

* Todo objeto (producto, ingrediente, tipo de producto) en el sistema podrá ser eliminado siempre que no esté referenciado desde otro objeto.

**Customer:**

* **Nombre**
* **Apellido**
* **ID (Opcional)**
* **dirección**
* **Teléfono**
* **String de observaciones**

**Employee:**

* **Nombre**
* **Apellido**
* **ID**

**User extends Employee**

* **User**
* **Password**
* La lista de clientes debe estar ordenada siempre alfabéticamente descendente por apellido y nombre, por tanto, cada vez que se agrega un nuevo cliente, este debe insertarse de forma ordenada (tenga en cuenta que insertar de forma ordenada no es lo mismo que agregar y luego ordenar). LIST CONVERSE REVERSE (SORT) COMPARATOR-Primero apellido y luego nombre String.compare(l1,l2), if son iguales comparar por nombre.

**Pedido:**

* Id Automatic
* Status- enum SOLICITADO, EN PROCESO, ENVIADO y ENTREGADO
* Listado de productos, List Detalle de pedido.
* Customer
* Employee
* Date-Hora actual del sistema.
* Observaciones- agrega las observaciones del customer.

**Para hacer pedido botón para buscar cliente**

**Y si no crear.**

* Usted tendrá una pequeña bonificación si muestra permanentemente actualizada la fecha y hora actual en alguna parte de la ventana.

**Detalle pedido**

* Producto PRODUCTO
* Cantidad Int
* El programa debe permitir cambiar el estado de un pedido entre SOLICITADO, EN PROCESO, ENVIADO y ENTREGADO. **Botón en la pantalla del pedido**
* Table view por cada producto,pedido,cliente y usuarios.
* 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.
* Serialización List de Users, List de Productos, Lits de clientes, List de pedidos
* Clase controladora List de Users, List de Productos, Lits de clientes, List de pedidos
* Exportar -CSV de cada pedido 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, 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 hora, y las observaciones.

El reporte debe estar en un intervalo (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 organizado por más reciente.

Preguntar el separador, pero por defecto es (;)

La primera línea del archivo debe tener los nombres de las columnas separadas también por dicho separador.

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 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. (b) 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). 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. En los dos reportes se debe agregar una fila al final que totalice las columnas numéricas.

El programa debe tener una opción que permita listar en pantalla todos los productos en orden de precio ascendente, otra opción que permita listar en pantalla todos los ingredientes en orden alfabético descendente. El programa debe 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.

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 y 1 ordenamiento utilizando Comparator, utilizando en ambos casos el sort de Collections o de Arrays.