UNIVERSIDAD ORT
Uruguay

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

EVALUACION	Obligatorio	GRUPO	TODOS	**FECHA**	Marzo de 2023
MATERIA	**Algoritmos y Estructuras de Datos 1**				
CARRERA	Analista Programador / Analista en Tecnologías de la información				
CONDICIONES	**- Lectura de Obligatorio:** 27/03/2023 **- Fecha máxima de entrega 1**: 04/05/2023 hasta las 21 hs. Puntaje: Máximo 15, Mínimo 0 **- Fecha máxima de entrega 2**: 14/06/2023 hasta las 21 hs. Puntaje: Máximo 40, Mínimo 0 LA ENTREGA SE REALIZA EN FORMA ONLINE EN ARCHIVO NO MAYOR A 40MB EN FORMATO ZIP O PDF. **IMPORTANTE** - Los grupos deben estar conformados por hasta 2 personas. - Inscribirse y subir el trabajo a Gestión antes de la hora indicada, ver hoja al final del documento: "RECORDATORIO" - Se debe entregar un conjunto de pruebas (utilizando la clase Prueba y Retorno provista durante el curso). Estas deben cubrir todos los métodos, tanto los casos con resultado OK, como los de ERROR.				

Información importante

- **La selección adecuada de las estructuras para modelar el problema y la eficiencia encada una de las operaciones es muy importante.**
- Se deben implementar todos los TADs utilizadas, no se permite el uso de Listas, pilas y colas de JAVA.
- Los obligatorios se realizan por equipos de hasta 2 **estudiantes.**
- Se deberán respetar los formatos de impresión dados para las operaciones que imprimen en consola.
- El resto de las operaciones no deben imprimir nada en consola.
- El sistema no debe requerir ningún tipo de interacción con el usuario por consola.
- Es obligación del estudiante mantenerse al tanto de las aclaraciones que se realicen en clase o a través del foro de aulas.
- Deberá aplicar la metodología vista en el curso.
- Para la presentación de la documentación se publicará en aulas.ort.edu.uy un template. (El uso deeste template es obligatorio).
- El proyecto será implementado en lenguaje JAVA sobre una interfaz que se publicará en el sitio de la materia en aulas.ort.edu.uy (El uso de esta interfaz es obligatorio).

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

Obligatorio:

Contenido

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

1. Introducción

Se desea implementar un sistema para la autogestión de pedidos en una importante cadena de comercios de comidas rápidas. Los clientes - al llegar al comercio - pueden interactuar con un "Tótem de pedidos", donde especifican los productos que desea comparar y sus cantidades. Una vez "cerrado" y procesado el pedido, podrá ser retirado por el cliente.

El sistema debe ser capaz de gestionar clientes, productos y gestionar el stock de la mercadería. Asimismo, debe brindar funcionalidades de reporte de seguimiento y para la toma de decisiones de la empresa. Los mismos deben ser resueltos a nivel de arquitectura y estructuras del sistema de la forma más eficiente posible.

Se proveen los siguientes tipos de datos que deberán ser respetados.

Sistema	public class Sistema{
	/*Aquí introduzca la información que estime conveniente*/
	}

Retorno	public class Retorno{
	enum Resultado{OK,ERROR_1,ERROR_2,ERROR_3,ERROR_4, ERROR_5, NO_IMPLEMENTADA};
	boolean valorBooleano
	int valorEntero;
	String valorString;
	Resultado resultado;
	}

Pueden definirse tipos de datos (clases) auxiliares.

2. Registros de Clientes y Productos

2.1. Crear Sistema de Autoservicio

Firma: `Retorno crearSistemaDeAutoservicio(int maxUnidadesDePedido);`

Descripción: Crea la estructura necesaria para representar el sistema de Autoservicio. Se deberá indicar cuántas unidades máximas se permiten en total por pedido.

Retornos posibles	
OK	Si pudo inicializar el sistema correctamente.
ERROR	Si las unidades máximas de pedidos son <= 3
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.2. Crear Cliente

Firma: `Retorno agregarCliente(String nombre,String ci,int tel);`

Descripción: Registra al cliente, siempre y cuando ya no exista en el sistema.

Retornos posibles	
OK	Si pudo registrar al cliente
ERROR	1.- Si ya existe un cliente con igual ci
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.3. Eliminar Cliente

Firma: `Retorno eliminarCliente(String ci);`

Descripción: Elimina el cliente con el ci indicado, siempre y cuando no tenga pedidos registrados (abiertos o cerrados).

Retornos posibles	
OK	Si pudo eliminar el cliente.
ERROR	1. En caso de que no exista un cliente con esa ci. 2. Si tiene pedidos registrados (no se elimina el cliente).
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.4. Agregar Producto

Firma: `Retorno agregarProducto(String nombre, String descripcion);`

Descripción: Crear el producto en el sistema. El. El sistema generará un número de producto en forma automática y correlativo comenzando del 1.

Retornos posibles	
OK	Si se pudo crear el producto.
ERROR	1. En caso de que ya exista un producto con igual nombre
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.5. Eliminar Producto

Firma: `Retorno eliminarProducto(String nombre);`

Descripción: Elimina el producto del sistema.

Retornos posibles	
OK	Si pudo eliminar el producto.
ERROR	1.- En caso de que no exista el nombre de producto. 2.- En caso de que el producto se encuentre en un pedido (abierto o cerrado)
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

2.6. Alta de stock a producto

Firma: `Retorno altaStockProducto(int nroProducto, int unidades);`

Descripción: Se registra el stock del producto en el sistema (acumulándose al ya existente).

Retornos posibles	
OK	Si pudo dar de alta del stock al producto
ERROR	1.- En caso de que no exista un producto con el número indicado. 2.- En caso de que las unidades sean <= 0
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

3. Gestión de pedidos

3.1. Abrir un nuevo pedido

Firma: `Retorno aperturaDePedido(String ciCliente);`

Descripción: Se realiza la apertura de un nuevo pedido. Dicho pedido queda "abierto" para que se puedan seleccionar e ingresar productos con sus unidades correspondientes. Un mismo cliente no puede tener más de un pedido abierto en forma simultánea. El sistema generará un número de pedido en forma automática y correlativo comenzando del 1.

Retornos posibles	
OK	Se realiza el despacho correctamente
ERROR	1.- En caso de que no exista el cliente con la ci indicada 2.- En caso de que el cliente ya tenga un pedido abierto.
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

3.2. Agregar productos a pedido

Firma: `Retorno agregarProductoAPedido(String ciCliente, int nroProducto, int unidades);`

Descripción: Se agrega una cantidad especificada de unidades de un determinado producto al pedido. No hay restricciones en lo que refiere a la cantidad de productos. Se deberá controlar que la cantidad de unidades totales del producto no supere al máximo definido.

Retornos posibles	
OK	Si se pudo realizar el retiro en forma total o parcial.
ERROR	1.- En caso de que no exista la ci del cliente 2.- En caso de que no exista el nro de producto. 3.- En caso de que, con la cantidad de unidades indicadas, se supere el máximo de unidades totales permitidas para el pedido.
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

3.3. Deshacer agregado de productos

Firma: `Retorno deshacerPedido(String ciCliente, int cantAccionesDesacer);`

Descripción: El cliente podrá indicar deshacer las últimas acciones de agregado de productos (con sus unidades) especificando cuantas acciones desea deshacer. Siempre se comienza deshaciendo el último producto agregado con todas sus unidades.

Ejemplo: Si se indica deshacer 1 acción, se elimina el último producto agregado con sus unidades
Si se indica deshacer 3 acciones se eliminan los 3 últimos productos agregados con sus unidades

Retornos posibles	
OK	Si se pudo deshacer la cantidad de acciones especificadas
ERROR	1.- En caso de que no exista la ci del cliente 2.- En caso de que el nro de acciones sea <=0 3.- En caso de que la cantidad de acciones solicitadas supere la cantidad de productos que fueron agregados.
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

3.4. Cerrar pedido

Firma: `Retorno cerrarPedido(String ciCliente);`

Descripción: Se cierra el pedido. En este momento, dicho pedido queda a la espera de que sea procesado por el personal del comercio para su entrega.

Retornos posibles	
OK	Si se pudo cerrar el pedido
ERROR	1.- En caso de que no exista la ci del cliente
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

3.5. Tomar pedido para procesamiento

Firma: `Retorno procesarPedido(int cantiPedidos);`

Descripción: Los pedidos cerrados serán procesados en orden de llegada. Dado que existe una capacidad finita del personal que debe realizar el procesamiento, se indicará la cantidad de pedidos a procesar. Los mismos deben pasar a estado de "Prontos para entregar"

Retornos posibles	
OK	Si se pudo deshacer la cantidad de acciones especificadas
ERROR	1.- En caso de que la cantidad de pedidos sea <=0
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

4. Listados y reportes

4.1. Listar Clientes

Firma: `Retorno listarClientes();`

Descripción: Se listan los clientes ordenados por orden alfabético. Se muestran todos sus datos.

Retornos posibles	
OK	Si se muestran todos los clientes
ERROR	No hay errores
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

NOTA: Esta operación debe ser realizada recorriendo una única vez la estructura escogida.

4.2. Listar Productos

Firma: `Retorno listarProductos();`

Descripción: Se muestran todos los productos en el orden que fueron registrados. Se muestra sus datos y el stock disponible de dicho producto.

Retornos posibles	
OK	Si se muestran todos los clientes
ERROR	No hay errores
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno No por defecto.

NOTA: Esta operación debe ser realizada en forma recursiva.

4.3. Listar pedidos abiertos

Firma: `Retorno listarPedidosAbiertos();`

Descripción: Lista todos los pedidos con sus productos y unidades solicitadas. Se debe mostrar la cantidad total de unidades que tiene cada pedido.

Retornos posibles	
OK	Si se muestran los pedidos con sus productos con sus unidades. También si se muestra la cantidad de unidades totales por cada pedido.

ERROR	No hay errores
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

4.4. Listar pedidos cerrados de Cliente

Firma: `Retorno pedidosCerradosDeCliente(int ci);`

Descripción: Lista todos los productos cerrados para dicho cliente.

Retornos posibles	
OK	Si se listan todos los pedidos cerrados del cliente indicado
ERROR	1.- En caso de que no exista la ci del cliente
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

4.5. Listar pedidos listos para la entrega

Firma: `Retorno productosParaEntregar ();`

Descripción: Se deben mostrar - similar a un "panel informativo" - los pedidos que están prontos para la entrega. Se debe mostrar de cada pedido únicamente su número y nombre del cliente.

Retornos posibles	
OK	Si se listan todos los pedidos cerrados del cliente indicado
ERROR	1.- En caso de que no exista la ci del cliente
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

NOTA: Esta operación debe ser realizada en forma recursiva.

4.6. Reporte de envíos por productos

Firma: `Retorno reporteDeProductosSolicitadosXCliente();`

Descripción: Se muestra una matriz donde, para cada pedido (filas) se muestran la cantidad de unidades ingresadas para cada producto (columnas).

Retornos posibles	
OK	Si se muestra el reporte correctamente.
ERROR	No hay errores
NO_IMPLEMENTADA	Cuando aún no se implementó. Es el tipo de retorno por defecto.

NOTA: Esta operación debe estar soportada en una matriz de información.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

PRIMERA ENTREGA

Para la primera entrega se solicita:

1) Representación gráfica de las estructuras seleccionadas para resolver TODO el problema planteado (ver ejemplo disponible en Aulas).
2) Especificación de pre y post condiciones de TODAS las operaciones que conforman la interfaz “Obligatorio”
3) Implementar las operaciones 2.1, 2.2, 2.3, 2.4 y 2.5.
4) Realizar y entregar juego de pruebas que evidencien el correcto funcionamiento de dichas operaciones.

Se deberá entregar un .ZIP con el proyecto (implementación, pre y post condiciones y pruebas) y un archivo .PDF con la representación de la arquitectura escogida. Se valorará la eficiencia y adecuación de esta al contexto planteado.

SEGUNDA ENTREGA

Para la segunda entrega se solicita:

1) Representación gráfica ajustada final.
2) Implementar de todas las operaciones de la interfaz “Obligatorio”
3) Realizar y entregar juego de pruebas definitivo que evidencie el correcto funcionamiento de TODAS las operaciones.

Se deberá entregar un .ZIP con el proyecto (implementación, pre y post condiciones y pruebas) y un archivo .PDF con la representación de la arquitectura final escogida. Se valorará la eficiencia y adecuación de esta al contexto planteado.

Facultad de Ingeniería
Bernard Wand-Polak
Cuareim 1451
11.100 Montevideo, Uruguay
Tel. 2902 15 05 Fax 2908 13 70
www.ort.edu.uy

RECORDATORIO: IMPORTANTE PARA LA ENTREGA

- **Obligatorios**

La entrega de los obligatorios será en formato digital online, a excepción de algunas materias que se entregarán en Bedelía y en ese caso recibirá información específica en el dictado de esta.

Los principales aspectos a destacar sobre la **entrega online de obligatorios** son:

1. Ingresá al sistema de Gestión.
2. En el menú, seleccioná el ítem "Evaluaciones" y la instancia de evaluación correspondiente, que figura bajo el título "Inscripto".
3. Para iniciar la entrega hacé clic en el ícono:
4. Ingresá el número de estudiante de cada uno de los integrantes y hacé clic en "Agregar". El sistema confirmará que los integrantes estén inscriptos al obligatorio y, de ser así, mostrará el nombre y la fotografía de cada uno de ellos. Una vez agregados todos los integrantes, hacé clic en "Crear equipo".

   Cualquier integrante podrá:

   • Modificar la integración del equipo.

   • Subir el archivo de la entrega.

5. Seleccioná el archivo que deseas entregar. Verificá el nombre del archivo que aparecerá en la pantalla y hacé clic en "Subir" para iniciar la entrega. Cada equipo (hasta 2 estudiantes) debe entregar **un único archivo en formato zip o rar** (los documentos de texto deben ser pdf, y deben ir dentro del zip o rar). El archivo a subir debe tener **un tamaño máximo de 40mb**

   Cuando el archivo quede subido, se mostrará el nombre generado por el sistema (1), el tamaño y la fecha en que fue subido.
6. El sistema enviará un e-mail a todos los integrantes del equipo informando los detalles del archivo entregado y confirmando que la entrega fue realizada correctamente.
7. Podés cerrar la pestaña de entrega y continuar utilizando Gestión o salir del sistema.
8. La **hora tope para subir el archivo será las 21:00** del día fijado para la entrega.
9. La entrega se podrá realizar desde cualquier lugar (ej. hogar del estudiante, laboratorios de la Universidad, etc).
10. Aquellos de ustedes que presenten alguna dificultad con su inscripción o tengan inconvenientes técnicos, por favor contactarse con el Coordinador o Coordinación adjunta **antes de las 20:00hs.** del día de la entrega, o enviando mail a las tres siguientes direcciones: gervaz@ort.edu.uy ,alamon@ort.edu.uy y terra@ort.edu.uy, o telefónicamente al 29021505 - int 1156 (de 8:00 a 20:00 hs) .

Si tuvieras una situación particular de fuerza mayor, debes contactarte con suficiente antelación al plazo de entrega, al Coordinador de Cursos (gervaz@ort.edu.uy) o Secretario Docente (paulos@ort.edu.uy).