

---

**Primer Parcial – Versión 1**

29/04/2022

**Tener en cuenta que:**

- La duración del parcial es de 2:30 horas.
- Se debe realizar de forma **individual**, recordar el código de honor el cual acordaron respetar al entrar en la Universidad, ítem 3:
  - *El estudiante asume una participación honesta en las actividades de la Universidad, que se manifiesta en su comportamiento en clase y en el trabajo bien hecho. Por ello renuncia al fraude académico en cualquiera de sus modalidades pautadas en el reglamento del Alumno.*
  - La referencia al reglamento de alumno es el Artículo 80 que indica lo siguiente:
    - Entre ejemplos de conducta inapropiada grave se encuentra: “La simulación, el engaño, el plagio o cualquier otra manifestación de deslealtad que se advierta en las formas de evaluación, sean éstas escritas u orales.”
- Se puede usar material para la parte práctica, la parte teórica es sin material (primeros 30 minutos).
- Se entrega un proyecto que debe usarse para realizar la evaluación.
- Se deben usar la librería de TADs entregada en el proyecto, NO se puede usar los TADs propios ni de terceros a no ser que la letra lo indique.
- La entrega se debe realizar en el Moodle en el horario indicado al comienzo de la evaluación.
- Tenga en cuenta que no se tomarán en consideración para la corrección clases que no compilen.

**Preguntas Teóricas :**

1. Suponga que dado un árbol binario de búsqueda vacío se ingresan los siguientes elementos {35, 12, 34, 1, 80, 100}. Realice un dibujo de cómo quedaría el árbol y simule la salida del algoritmo de recorrida inOrder. Por último, explique brevemente como realizaría su implementación.
  2. Indique cuando sería conveniente usar un árbol binario de búsqueda en vez de una lista enlazada en un algoritmo.
-

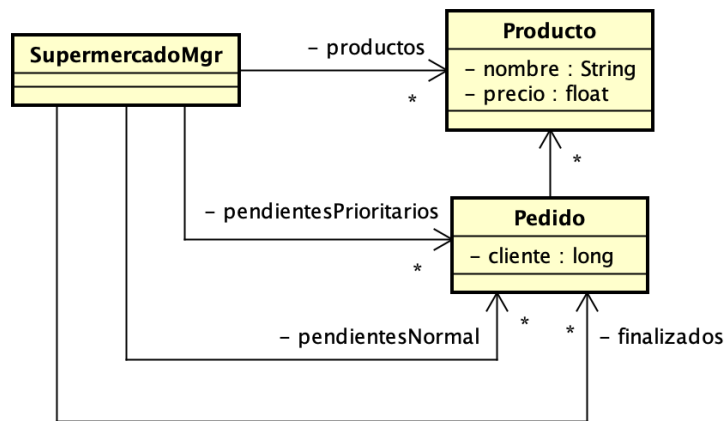
### 3. Ejercicios Práctico

#### Ejercicio 1

Un supermercado desea digitalizar su proceso de recepción y atención de pedidos con entrega a domicilio. El super maneja una lógica particular de atención que cumple las siguientes reglas:

- Todo pedido que tenga menos de 3 artículos es un pedido prioritario, si tiene 3 o mas artículos es un pedido normal.
- Los pedidos prioritarios se deben atender antes que los pedidos normales.
- Si se tiene mas de un pedido prioritario se debe atender de forma LIFO.
- Si se tiene mas de un pedido normal se debe atender de forma FIFO.
- No puede haber mas de un pedido por cliente pendiente.

Pensando en esta realidad se propone el siguiente diagrama de clases para representar la realidad:



Se debe implementar una interfaz SupermercadoMgt y su respectiva implementación SupermercadoMgr que disponga las siguientes operaciones:

- **void agregarProducto(String nombre, float precio) throws EntidadYaExiste**
  - Registra un producto en el supermercado identificado por su nombre.
  - Lanza la excepción EntidadYaExiste si el producto ya esta registrado.
- **void ingresarPedido(long cedula, MyList<String> productos) throws EntidadNoExiste, InformacionInvalida**
  - Agrega un pedido de productos de cierto cliente. Dentro de estos pedidos se permite que un mismo producto este repetido. En caso que el cliente ya disponga de un pedido sin procesar se debe lanzar un error.
  - Lanza la excepción EntidadNoExiste si algún producto no existe.
  - Lanza InformacionInvalida si ya existe un pedido pendiente para el cliente.
- **Pedido procesarProximoPedido() throws InformacionInvalida**
  - Retorna el próximo pedido pendiente para procesarse dado las reglas indicadas al comienzo de la letra. Esta operación debe a su vez archivar el pedido como finalizado para analisis posteriores.
  - Lanza la excepción InformacionInvalida si no existe próximo pedido
- **void cancelarPedido(long cedula) throws EntidadNoExiste**
  - Cancela un pedido sin procesar de la cédula indicada.
  - Lanza la excepción EntidadNoExiste si no se encontró pedido pendiente de la cédula indicada.
- **Cliente obtenerClienteConTicketMasCaro()**

- Retorna el cliente que tuvo el ticket con importe total (suma de los precios de sus productos) de mayor valor. En caso de no existir retorna null.

Consideraciones:

- Para realizar la operación cancelarPedido puede necesitarse agregar una operación a Queue y Stack.
  - Para probar el correcto funcionamiento del sistema se debe crear pruebas en la clase principal de todo el flujo.
  - Existe libertad en indicar tanto excepciones como otras entidades no declaradas para representar la realidad.
  - Tenga en cuenta que para esta instancia si no se selecciona TAD's acordes a lo solicitado se considera error.
-