RÚBRICA DE CALIFICACIÓN PARCIAL UNO

Total	**5,0**
Primer Punto	**1,50**
El estudiante identifica las clases y la interfaz que hacen parte del modelo (Producto,Medicamento,Formulado,SuplementoAlimenticio, Controlable) y la única clase que hace parte del control (Ejecutable)	0,10
El estudiante tiene claro que la interfaz Controlable es una interfaz bandera y por esta razón está bien definida	0,30
El estudiante define un método abstracto en la clase padre para calcular el costo y lo redefine en las clases hijas (Medicamento y SuplementoAlimenticio)	0,30
El estudiante identifica las entradas: código de producto	0,10
El estudiante identifica las salidas: mensaje solicitando fórmula médica y precio del producto	0,10
El estudiante busca el producto en la colección y lo recupera en una variable de tipo Producto	0,10
El estudiante usa la interfaz para verificar el control de fórmula médica	0,20
Código Java	0,30
Segundo Punto	2,00
Diagrama de clases	**1,50**
El estudiante identifica una clase material y 6 clases hijas (Vidrio, Papel, Cartón, Plástico, Metal y orgánico)	0,50
La clase material tiene los atributos nombre y proveedor	0,05
la clase vidrio tiene el atributo que permite identificar el tipo de contenido	0,05
la clase papel tiene los atributos gramaje y uno que permite identificar si está limpio o no	0,05
La clase cartón tiene un atributo que permite identificar si es cartón paja o no	0,05
La clase plástico tiene el atributo color	0,05
la clase metal tiene el atributo peso	0,05
La clase orgánico tiene el atributo peso	0,05
El estudiante identifica la interfaz reciclable y el método que permite mostrar las indicaciones	0,10
La interfaz sólo es implementada por las clases: vidrio, papel, cartón, plástico y metal. La relación está bien construida	0,15
Las clases que implementan la interfaz sobreescriben el método	0,20
La clase ejecutable tiene dos arraylist, uno para almacenar los materiales reciclables y otro para almacenar los materiales orgánicos o uno para los materiales	0,10
La clase ejecutable tiene el método que permite registara el material recolectado y muestra las indicaciones	0,10
Descripción de métodos	**0,50**
El estudiante solicita todas las entradas y salidas: Solicitar el tipo de material, el nombre y el proveedor. Dependiendo del tipo de material: • Si es papel: solicitar el gramaje y si viene limpio • Si es cartón: preguntar si es cartón paga • Si es vidrio: Preguntar si el contenido fue consumible • Si es plástico: solicitar el color • Si es metal o desechos orgánicos: solicitar el peso salidas: El método debe desplegar un mensaje para el caso de los productos reciclables que indique el color de la bolsa a utilizar.	0,10

El proceso realiza lo solicitado:Solicitar las entradas Crear el objeto material Si es un objeto reciclable mostrar en pantalla el mensaje que indique el color de la bolsa a utilizar y adicionar el material en el arraylist de productos naturales	
Si el producto es orgánico se debe almacenar en un arraylist especial	0,30
El estudiante redefine el método de la interfaz en todas las clases que la implementan	0,10
Tercer Punto	**1,50**
El estudiante identifica las 3 excepciones No comprobadas que se presentan en el método nextInt() en el main	0,30
El estudiante identifica las 3 excepciones Comprobadas que se presentan en el método InstanciaDeLaClase	0,30
El estudiante indentifica que la excepción no comprobada NullPointerExeption no se presenta	0,10
El estudiante controla las exepciones no comprobadas en el método main enviando mensajes apropiados al usuario por cada una de ellas. Tiene en cuenta el orden de las exepciones, respetando la herencia que hay entre ellas.	0,30
El estudiante se recupera de las exepciones no comprobadas sin terminar el programa, usando una estructura repetitiva	0,10
El estudiante controla las exepciones comprobadas en el método instanciaDeLaClase() o las propaga y las controla en el main. Para esto, envía para una mensajes adecuados al usuario.	0,30
El estudiante corrije la propagación de la excepción NullPointerExeption	0,10

RÚBRICA DE CALIFICACIÓN PARCIAL DOS

Total	**5,0**
Primer punto	**2,00**
Diagrama de clases	**1,00**
Maneja MVC. Modelo:competencia y Paloma.Control:hilo y ejecutable. Vista: ventanas	0,2
utilización correcta de sintaxis UML (dirección de flechas, atributos privados, herencia, métodos públicos y estáticos)	0,1
El estudiante identifica la clase Paloma y sus atributos: código, propietario, edad. Adicionalmente, atributos para dibujar la paloma	0,1
El estudiante identifica la clase competencia y representa un atributo Paloma para conocer la paloma ganadora. Los métodos incluidos en la clase deben permitir registrar una paloma y consultar todas las palomas.	0,1
El estudiante identifica correctamente la relación entre competencia y Paloma	0,2
El estudiante identifica la clase hilo que hereda de Thread	0,1
La clase hilo tiene relación con la clase Paloma	0,2
Descripción de métodos	**1,00**
método para registrar palomas en la clase ejecutable	
Solicita los datos de los atributos, crea el objeto paloma e invoca el método registra Paloma de la clase competencia	0,2
Método que permite iniciar una competencia o puede estar en el main	
Se recupera el objeto fuente	0,1

Se registra el objeto escucha en la fuente	0,1
se describe la clase interna que permite iniciar la carrera, inicializa todos los objetos Paloma e inicia los hilos	0,2
método run	
el run sólo invoca el método de dsplazamiento en paloma	0,2
Método main	
Se crea un objeto competencia	0,05
Se invoca el método que permite iniciar competencia	0,05
Se invoca el método que permite registrar palomas	0,05
La aplicación muestra la información de la paloma que primero llegue	0,05
Segundo punto	**2,00**
Análisis	**1,50**
Identificación y descripción de la Clase Interna para el ActionListener de cboPlato. - Obtiene el índice del producto seleccionado a través del comboBox de Platos. - Obtiene el objeto que corresponde al producto Plato usando el índice del valor seleccionado en el combo y la lista de productos del Restaurante. - Obtiene el valor boolean para cada uno de los tamaños disponibles del producto. - Habilitar o deshabilitar los redioButtons de los tamaños, dependiendo del valor boolean que representa la disponibilidad para cada tamaño. - Consume el método calcularPrecio():double del producto y lo retorna para incluirlo en la vista en el textField correspondiente. - Se actualiza el valor del IVA, el valor del servicio y el total.	0,30
Identificación y descripción de la Clase Interna para el ActionListener de cboBebida. - Obtiene el índice del producto seleccionado a través del comboBox de Bebidas. - Obtiene el objeto que corresponde al producto Bebida usando el índice del valor seleccionado en el combo y la lista de productos del Restaurante. - Consume el método calcularPrecio(): y lo retorna para incluirlo en la vista en el textField correspondiente. - Se actualiza el valor del IVA, el valor del servicio teniendo en cuenta el chck y el total.	0,30
Identificación y descripción de la Clase Interna para el ActionListener de chckServicio - Si el check box de servicio no está seleccionado Asignar el valor cero a la propiedad text del textField de valorDelServicio. - Sino Asigna a la propiedad text del textField de valorDelServicio, el resultado de multiplicar el valor de la propiedad Text de valor del pedido por el 10%. - Actualizar el valor de la propiedad text del textField del Total del pedido, sumando el valor del pedido, el IVA y el servicio.	0,30
Identificación y descripción de la Clase Interna para el ActionListener de btnRegistrar - Buscar el cliente que corresponde a la cédula ingresada por el usuario. - Si no existe Mostrar un mensaje al usuario - Sino Buscar el producto Plato a través del Restaurante Buscar el producto Bebida a través del Restaurante Crear un objeto Pedido pasando por parámetro el Cliente, el producto Plato, el producto Bebida y un boolean que indica si el cliente selección o no la opción de pagar el servicio. o Agregar el objeto Pedido a la lista de pedidos del restaurante.	0,30

Identificación y descripción de la Clase Interna para el ActionListener de btnLimpiar. - Asignar espacio en blanco a los campos cédula, valor del pedido, valor Iva, valor del servicio y total.	0,30
Código	**0,50**
Creación del objeto Listener y su registro a través de la clase interna, para manejar el evento ActionEvent asociado al JComboBox de Bebida	0,10
Implementación del método ActionPerformed	0,10
Obtención del objeto fuente del evento (el ComboBox)	0,10
Obtención de la bebida a partir del valor seleccionado en el Combo	0,10
Actualización de los campos requeridos	0,10
Tercer punto	**1,00**
Pregunta 3.1: El estudiante identifica que el estado en el que se encuentra un hilo luego de instanciarlo es el estado nuevo o new	0,10
Pregunta 3.2: El estudiante identifica que los hilos que esperan por el procesador con la misma prioridad son atendidos de acuerdo al orden de llegada o de forma aleatoria dependiendo del mecanismo de atención que maneje el sistema (no juegan las prioridades)	0,20
Pregunta 3.3: El estudiante indica que cuando se interrumpe un hilo este si termina su ejecución y pierde los recursos que se le habían asignado.	0,20
Pregunta 3.4: El estudiante propone una situación en la que se evidencia la necesidad de sincronizar el uso de un recurso común a varios procesos con el fin de evitar problemas de acceso concurrente.	0,20
Pregunta 3.5 A: Evento (ActionEvent) - Obj fuente (botónIniciar) - Listener (un objeto de la clase ActionListener).	0,10
Pregunta 3.5 B: Evento (MouseEvent) - Obj fuente (panelVaso, panelVasoUno) - Listener (un objeto de tipo MouseControlador).	0,10
Pregunta 3.5 C: Evento (KeyEvent) - Obj fuente (campo) - Listener (un objeto de la clase interna KeyAdapter).	0,10

RÚBRICA DE CALIFICACIÓN PARCIAL TRES

Total	**5,0**
Primer punto	**2,00**
1.Completar Diagrama de Clases	**0,80**
El estudiante implementa la interfaz Comparable para la clase Producto	0,10
El estudiante implementa el método compareTo(Object):int en la clase Producto	0,10
El estudiante completa el criterio de igualdad de la clase Producto sobrescribiendo el método hashCode():int	0,10
El estudiante define la clase Comparadora para el criterio de comparación por costo de fabricación. La clase implementa el método compareTo(Object,Object):int	0,20
El estudiante implementa la interfaz Comparator para la nueva clase Comparadora	0,20

El estudiante identifica la necesidad de un método que permita a la Fabrica retornar una colección con los productos ordenados por costoDeFabricación.	0,10
2.Análisis y código del método	**0,80**
El estudiante identifica las entradas, salidas, parámetros y retorno del método para escribir la información de los productos.	0,10
El estudiante identifica en el análisis que debe implementar un flujo de escritura de bytes.	0,10
El estudiante identifica en el análisis los campos de los productos que deben incluirse en el archivo para que puedan ser procesados por el sistema del cliente (código,nombre,precio de venta).	0,20
El estudiante crea correctamente el flujo de escritura de bytes	0,10
El estudiante implementa correctamente el recorrido del set a través de un iterador	0,10
El estudiante implementa correctamente la escritura de los datos de cada producto en el archivo de bytes.	0,10
El estudiante sigue las buenas prácticas cerrando el flujo.	0,05
El estudiante implementa un adecuado manejo de excepciones	0,05
3.Descripción de requisitos para serializar	**0,40**
El estudiante identifica que la clase Fabrica y Producto deben implementar la interfaz bandera Serializable	0,20
El estudiante indica que es necesario la definición de métodos para escribir y leer la información del sistema a través de flujos de objetos.	0,20
Segundo punto	**3,00**
El estudiante implementa adecuadamente los paquetes modelo y control	0,10
El estudiante identifica todas las clases con sus atributos y métodos básicos.	1,40
El estudiante implementa la relación entre Curso y Postre a través de un Map	0,10
El estudiante implementa la relación entre Escuela y Postre a través de un Map	0,10
El estudiante implementa la relación entre Escuela y Curso a través de un set o una lista	0,10
El estudiante implementa la relación entre Escuela y Profesor a través de un Set	0,10
El estudiante implementa correctamente la relación entre curso y Profesor	0,10
El estudiante implementa la relación entre Curso y Estudiante a través de un List/Set	0,10
El estudiante sobreescribe el método toString en la clase Receta para facilitar su consulta.	0,10
Si se implementa un TreeMap la clase Postre debe implementar Comparable y sobreescribir el método equals y el hashcode. Si utiliza un hashmap debe sobrescribir el equals y el hashcode.	0,20
El estudiante identifica los metodos de la clase Escuela	0,20
El estudiante identifica los metodos y atributos de la Ejecutable	0,20
El estudiante sobreescribe el hashCode y equals en la clase Profesor	0,10
El estudiante implementa en la clase Postre un constructor que recibe el nombre para las búsquedas de los postres.	0,10