

ALGORITMOS Y ESTRUCTURAS DE DATOS

EVALUACIÓN 2

Competencia asociada:

Desarrollar back end de aplicaciones utilizando estándares de la industria.

Unidad de Competencia:

1. Construye clases agrupadas dinámicas en el lenguaje Java que presentan un comportamiento complejo de flujo de control
2. Construye clases agrupadas de acuerdo a "buenas practicas" definidas por la industria

Consideraciones importantes

1. La evaluación tiene solo un intento permitido.
2. Usted posee un periodo en envío limitado para poder realizar el encargo. Luego de ello se entrega la retroalimentación de su código
3. La copia o plagio de algún encargo y/o evaluación será sancionada con nota 1.

Instrucciones

El dueño de un taller mecánico de reparación de automóviles necesita llevar un control de sus ventas de repuestos, por lo que lo contrata a usted para desarrollar un módulo de registro.

Las clases a desarrollar son:

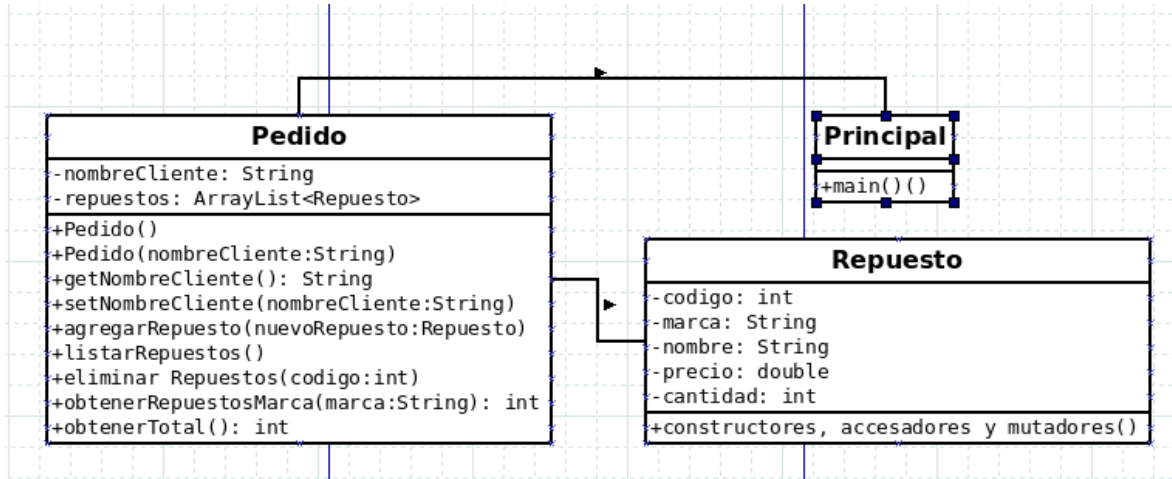
Repuesto: Del que se almacena su código, nombre, precio, cantidad y marca.

Pedido: Se almacena el nombre del cliente y la lista de repuestos utilizados.

Reglas de negocio:

- ⑩ nombre marca: no puede estar vacío
- ⑩ código: no puede estar vacío
- ⑩ cantidad: mayor que cero

Para desarrollar esta aplicación debe crear un proyecto en NetBeans con JAVA con la siguiente estructura:



Métodos solicitados:

La clase Repuesto debe contar con constructores, accesadores, mutadores y método para listar todos sus atributos.

Para la clase Pedido:

1. **agregarRepuesto** : Agregar un nuevo Repuesto a la lista sólo si no existe su código. En caso de existir, aumenta en 1 la cantidad.
2. **listarRepuestos**: Imprimir en pantalla los Repuestos que se encuentran en la lista.
3. **eliminar Repuestos**: Eliminar Repuestos de la lista por su código.
4. **obtenerRepuestosMarca**: Obtener el total de repuestos que se encuentran en la lista de pedido que correspondan a una misma marca (ingresando el nombre como parámetro del método)
5. **obtenerTotal**: Obtener el total del pedido

Desde la clase donde se encuentra el main:

⑩ Agregar dos repuestos distintos al pedido, enviando un mensaje de información del ingreso exitoso.

⑩ Mostrar todos los repuestos del pedido

⑩ Ingresar un código de repuesto verificar si existe enviando mensaje adecuado.

⑩ Eliminar alguno de los repuestos de la lista por su código y listar nuevamente para verificar que se eliminó el repuesto.

⑩ Mostrar el total de Repuestos que correspondan con la marca que ud. ingrese.

Criterios de evaluación

Escala de apreciación

Indicadores de logro	COMPLETAMENTE LOGRADO	MEDIANAMENTE LOGRADO	NO LOGRADO
	2 PUNTOS	1 PUNTO	0 PUNTO
Define las clases con sus atributos, constructores, getter y setter			
Utiliza ArrayList			
Implementa regla de negocio nombre marca: no puede estar vacío			
Implementa regla de negocio código: no puede estar vacío			
Implementa regla de negocio cantidad: mayor que cero			
método agregarRepuesto			
método listarRepuestos			
método eliminarRepuestos			
método obtenerRepuestosMarca			
método obtenerTotal			
En el main() crea todos los objetos solicitados			
En el main() agrega repuestos al pedido y los lista			
En el main() ingresa un código y verifica			
En el main() elimina repuestos del pedido			
En el main() muestra los repuestos que corresponden con la marca solicitada			

AUTOEVALUACIÓN

Estimado estudiante:

La autoevaluación es la valoración y/o reflexión que usted realiza sobre su proceso de aprendizaje. Este ejercicio es fundamental para identificar tanto debilidades como fortalezas de la evaluación realizada.

Para realizar la autoevaluación usted deberá contestar las siguientes preguntas, las cuales se encontrarán en la pestaña **AUTOEVALUACIÓN**, donde deberá completar cada una de ellas. *(No se deberán contestar en esta planilla)*

Conteste con toda la sinceridad posible, ya que será un insumo para la retroalimentación que realizará posteriormente el docente.

- ¿Qué hice bien?
- ¿En qué necesito ayuda?
- ¿Sobre qué quiero saber más?
- ¿Qué aprendí en esta evaluación?
- ¿Qué debo mejorar?
- ¿Qué acciones realizaré para mejorar?