

Instituto Tecnológico de Costa Rica (ITCR)
Ingeniería en Computación, Alajuela (ICAL)
Lenguajes de programación
II Semestre 2023
Actividad de estudio de caso

Tema: Programación orientada a objetos

Forma de trabajo: en grupos de máximo 2 personas.

Entrega por medio del TecDigital y presentación de solución en clase.

Aspectos básicos para la planificación del Caso

Objetivo específico (del curso al que corresponde el contenido):

Aprender un lenguaje ejemplo para cada uno de los cuatro paradigmas principales de programación, así como conocer varios otros lenguajes para representar conceptos que no tengan los lenguajes ejemplo.

Contenido:

Tema 4. Programación orientada a objetos: Principios de diseño y programación OO

Cantidad de sesiones (o trabajo extraclase):

2 horas de trabajo en clase para finalizar la programación y presentar la tarea al grupo en la clase siguiente.

Modelo de caso (Justificación):

Modelo de entrenamiento en la Resolución de Situaciones.

Se aplica el modelo de caso de **Resolución de Situaciones** para que el grupo de estudiantes genere las capacidades requeridas para desarrollar sistemas de información a la medida. Los estudiantes deben poner en práctica metodologías de desarrollo de sistemas para solventar las necesidades de un grupo de usuarios o una institución, proceso al que se van a enfrentar en muchas ocasiones cuando estén ejerciendo como profesionales en Computación.

Tipo de caso (Justificación):

Casos de resolución de problemas.

Se aplica el tipo de caso de resolución de problemas porque se requiere que los estudiantes tomen decisiones de diseño e implementación de las funcionalidades

del sistema a desarrollar a partir del conjunto de necesidades expresadas por los usuarios.

Objetivo de aprendizaje:

Al finalizar el ejercicio las y los estudiantes deben contar con más capacidad para aplicar las técnicas de modelización de problemas y desarrollo de sistemas utilizando orientación a objetos para la creación de programas computacionales y proponer mecanismos de implementación que aprovechen toda la funcionalidad de la OO en el lenguaje Java.

Ejercicio

Para el caso que se presenta a continuación:

1. Utilice orientación a objetos, para construir un modelo que plantee una solución completa al problema. Documenten el modelo utilizando artefactos del Lenguaje Unificado de Modelado (UML) y utilice una herramienta de software para presentar el modelo formalmente (10 puntos)
2. Seleccione un conjunto reducido de clases que implementen algunas de las funcionalidades del modelo propuesto que cumpla con lo siguiente.
 - Utilice el concepto de herencia en su modelo e implementación (5 puntos)
 - Defina una clase abstracta con al menos un método abstracto (5 puntos).
 - Defina una variable y un método de clase (static) (5 puntos)
 - Defina un bloque static (5 puntos).
 - Muestre cómo se implementa el polimorfismo por herencia (5 puntos) y por sobrecarga de métodos (5 puntos).
 - Utilice herencia de interfaces con al menos una definición de método por clase y que al menos tres clases del modelo de datos implementen (5 puntos).
- La implementación del modelo debe realizarse en Java y debe ser fiel al modelo UML (5 puntos).
- Cree una clase Test que permita probar y mostrar la funcionalidad desarrollada (5 puntos).
- Presente su solución en clase (5 puntos)

Delimitación:

Se desea brindar los servicios en Internet de manejo de listas de compras con la siguiente funcionalidad:

- La lista de compra (clase LengShoppingList) ofrece la siguiente funcionalidad:
 - Una lista de compras contiene cero o más productos.

- Los productos están asociados a un área de ventas, que puede ser zapatos, electrónicos, ropa y juegos. Asigne los atributos y comportamiento a las clases que usted considere convenientes.
- Cada producto tiene los siguientes atributos asociados: código (id), nombre (name), precio (price), cantidad de producto en existencia (quantityStock) y porcentaje de oferta asociado al producto (offer).
- Además cada producto en la lista de compras presenta la cantidad de unidades a comprar por el cliente (purchasedAmount).
- La funcionalidad debe poder agregar cantidad de items a la existencia de un producto (buyProduct).
- Cuando se agrega un producto a una lista de compras se debe indicar la cantidad a comprar. Esa cantidad se debe disminuir de la existencia del producto.
- La lista de compra está asociada a un cliente (LengClient). Cada cliente tiene asociado: cédula (id), nombre (firstName), apellido (lastName) y dirección (address).
- Se Debe poder consultar:
 - si la **lista** está vacía (empty()).
 - ¿cuántos productos tiene la **lista**? (countProducts()).
 - el total de la compra: suma del valor de los productos * la cantidad a comprar * oferta (totalPurchased()).
 - los datos del **lengCliente** asociado (toString()).
 - todos los datos de la lista (displayList()).