



# **Universidad del Valle**

**Facultad de Ingeniería  
Escuela de Ingeniería de Sistemas y Computación  
Fundamentos de Programación Orientada a Objetos**

## **NORMAS PARA LA ENTREGA DEL PROYECTO**

- **Lugar y Medio de Entrega:** *Todos los archivos que se soliciten se deben subir en la plataforma Moodle en el respectivo enlace del proyecto.*
- **Forma de entrega:** *Subir un archivo comprimido en formato ZIP con todos los elementos del proyecto y con un tamaño máximo de 10 MB.*
- **Plazo:** *Los estudiantes deben subir los archivos antes del día y la hora establecida por el profesor, el vínculo para esta actividad se deshabilitará automáticamente una vez se cumpla el plazo.*
- *Durante el curso no se recibirán proyectos por correo electrónico y tampoco se podrá entregar al monitor.*

## **Proyecto**

**Clases y objetos, Vectores de objetos, Herencia, Polimorfismo, Refactoring**  
**Fecha de entrega: 09 de diciembre de 2024**

### **Objetivos:**

- Desarrolla proyectos de programación colaborativamente en equipo, desempeñando unas tareas específicas individuales y comunicando sus ideas, participa proactivamente en la ejecución del proyecto.
- Desarrolla proyectos de programación buscando nueva información, aportando soluciones para el cumplimiento del programa y utilizando el paradigma orientado a objetos.

# Problema por resolver

## Comercio electrónico

Desarrolle un proyecto en C++ para una tienda de comercio electrónico, considerando los siguientes requerimientos:

1. Presentar un informe con la metodología de programación orientada a objetos aplicada en el desarrollo de la solución al problema.
2. Desarrollar la lógica de la aplicación utilizando Clases y objetos, Vectores de objetos, Herencia, Polimorfismo, Refactoring.
3. El programa debe gestionar la información de productos, marcas, facturas de ventas, facturas de compras, stocks, empleados, clientes y proveedores.
4. La información de cada clase es la siguiente:
  - a. **Producto:** idProducto, descripcionProducto, idMarca.
  - b. **Marca:** idMarca, nombreMarca.
  - c. **Factura:** idFactura, fechaFactura, horaFactura, tipoFactura (compra, venta), idEmpleado, idCliente, idProveedor.
  - d. **DetalleFactura:** idDetalleFactura, idFactura, idProducto, valorProducto, cantidadProductos.
  - e. **Stock:** idStock, idProducto, existencias, stockMinimo.
  - f. **Empleado:** id, nombre, direccion, telefono, email, salario, horario.
  - g. **Cliente:** id, rut, nombre, direccion, telefono, email, profesión, tipoCliente (pensionado, empleado público, empresario, militar).
  - h. **Proveedor:** id, rut, nombre, direccion, telefono, email, banco, nroCuenta, tipoCuenta (ahorros, corriente).
5. Validar los datos de entrada, garantizando que los usuarios ingresen valores permitidos (por ejemplo, cantidades y valores positivos y dentro de rangos razonables, nombres, teléfonos y correos electrónicos válidos, etc.). Mostrar mensajes de error si los valores son incorrectos.
6. Construir las interfaces gráficas de usuario para cada entidad (**Opcional**).
7. Los campos idProducto, idMarca, idFactura y idDetalleFactura, idStock, idEmpleado, idCliente, idProveedor son valores numéricos únicos, es decir, que no deben admitir valores duplicados.

8. Cuando se realice una factura de venta, la cantidad de productos vendidos se debe restar a las existencias en la entidad stock. En el caso de una factura de compra, la cantidad se debe sumar en las existencias de stock.
9. Desarrollar las siguientes funcionalidades:
  - a. Calcular el total de ventas mensuales.
  - b. Mostrar el o los meses con mayores ventas.
  - c. Imprimir todos los datos de los productos de una marca seleccionada por el usuario.
  - d. Mostrar los datos de la marca más vendida.
  - e. Presentar la cantidad y los valores totales de las facturas de venta y las facturas de compra con el objetivo de conocer las ganancias.
  - f. Mostrar los datos de todas las facturas de ventas de mayo: idFactura, fechaFactura, horaFactura, valorTotal, idEmpleado, idCliente.
  - g. Buscar y mostrar los datos de una factura por idFactura: idFactura, fechaFactura, horaFactura, tipoFactura, idEmpleado, idCliente, idProveedor, descripcionProducto, cantidadProductos, valorProducto, subtotalFactura, valorTotalFactura.
  - h. Mostrar un listado de los productos que tienen una existencia menor al stock mínimo.
  - i. Presentar los datos de los tres mejores clientes considerando el valor de las compras.
  - j. Imprimir los datos del empleado con más ventas por mes.
  - k. Mostrar la cantidad de facturas de compra por cada proveedor.
  - l. Mostrar el promedio de ventas por mes.
10. Utilizar la herencia en las entidades empleado, cliente y proveedor, aplicar el polimorfismo en alguna parte del proyecto y considerar el refactoring para mejorar las funcionalidades del sistema.

### **Observaciones:**

- Se debe desarrollar en grupos de **4** estudiantes.
- La solución debe incluir toda la documentación y el código fuente.
- La entrega realizada se debe sustentar en la fecha indicada.
- Los grupos deben escribir el código fuente con buenas prácticas y con el menor número de líneas de código.

**Retos:**

- Los grupos que construyan interfaces gráficas de usuario, recibirán un bono de **0.5** en la nota final de la asignatura.
- Los grupos que incluyan la opción de guardar y cargar datos históricos en archivos planos, recibirán un bono de **0.5** en la nota final de la asignatura.
- Los bonos serán válidos sólo si el proyecto cumple con todos los demás requerimientos.

**Entregables del proyecto:**

1. Metodología.
2. Cabecera de los archivos.
3. Algoritmos en C++.
4. Interfaz gráfica.
5. Pruebas funcionales de la interfaz gráfica.