

1er Proyecto - Tienda de Ciclismo

Un sistema de gestión es usado para obtener información, realizar transacciones y hacer pedidos para algún servicio, como por ejemplo la gestionar los trabajos en un taller mecánico.

Este sistema se encargará en gestionar la venta de productos relacionado al ciclismo como también ofrecer servicios de taller.

El desarrollo de este programa debe ser Orientado a Objetos por lo que las diferentes clases que forman parte de este enunciado han de desarrollarse. La información debe de guardarse en un archivo plano con separadores (CSV), JSON o XML

1. Software que desarrollar

Su trabajo consiste en crear una aplicación para la gestión de una tienda de bicicletas, Java será el lenguaje de programación utilizado, haciendo uso de los conceptos visto relacionado a la orientación de objetos.

1.1. Control de acceso

Para acceder a estas funcionalidades el **usuario** deberá ingresar una **contraseña** y al ingresar se deben habilitar una ventana con las siguientes funcionalidades:

1. Registro de bicicletas y accesorios para ciclismo
2. Registro de cliente
3. Registro de servicio de taller
4. Facturación de productos

Los usuarios deben de estar guardados en un archivo llamado usuarios.acc, el formato interno será de la siguiente manera:

- pperez,12345,
- ccastro,castro123,

1.2. Registro de Productos

El sistema debe permitir dar mantenimiento a los diferentes conceptos relacionados con una bicicleta, se debe permitir **buscar, agregar, modificar y eliminar**. Primero debe existir un módulo para crear los diferentes **tipos de productos**, sus atributos serían;

- Código: numérico y se incrementa automáticamente de 1 en 1
- Nombre: una cadena de texto no vacía y sin espacios en blanco en sus extremos

Creado los tipos de productos, se procede a crear cada uno de los **artículos** a vender en la tienda, sus atributos serían:

- Código del artículo: numérico y se incrementa automáticamente de 1 en 1
- Código del tipo de producto: valor numérico existente en tipo de productos
- Nombre del artículo: una cadena de texto no vacía y sin espacios en blanco en sus extremos
- Tipo: valor a seleccionar entre Bicicleta, Accesorios, Suplementos alimenticios
- Tamaño: Si se selecciona bicicleta, los tamaños son de 12, 16, 22, 26, 27, 27.5 29. Si se seleccionó Accesorios o Suplemento, este atributo no se despliega y se guardará el valor de cero.
- Marca del producto: una cadena de texto no vacía y sin espacios en blanco en sus extremos
- Precio: valor numérico sin decimales
- Cantidad: valor numérico sin decimales.

Además, debe considerar al momento de programar este módulo lo siguiente:

- Se podrá buscar por **código y/o nombre**.
- Al momento de modificar, no podrá alterarse el **código principal**
- Al eliminar, debe mostrar al usuario una ventana de confirmación antes de realizar el borrado. Si el artículo ya está facturado no podrá ser eliminado.
- Después de cada acción debe mostrar al usuario un mensaje de éxito o error
- Todos los atributos son obligatorios para crear o modificar

1.3. Registro de clientes

Este módulo debe permitir dar mantenimiento a los diferentes conceptos relacionados con el cliente, se debe permitir **buscar, agregar, modificar y eliminar**.

Los atributos por registrar de un cliente son lo siguiente:

- Código: numérico y se incrementa automáticamente de 1 en 1
- Nombre: una cadena de texto no vacía y sin espacios en blanco en sus extremos
- Apellidos: una cadena de texto no vacía y sin espacios en blanco en sus extremos
- Teléfono: numérico, largo de solo 8 dígitos, los dígitos a iniciar son 2, 4, 6 y 8
- Correo electrónico: una cadena de texto no vacía y sin espacios en blanco en sus extremos. Además, debe validarse que tenga el **formato válido**
- Provincia: valor a seleccionar: San José, Alajuela, Cartago, Heredia, Puntarenas, Guanacaste y Limón. Desplegarse en orden alfabético.
- Cantón: una cadena de texto no vacía y sin espacios en blanco en sus extremos
- Distrito: una cadena de texto no vacía y sin espacios en blanco en sus extremos
- Fecha de nacimiento: Formato correcto de fecha

Además, debe considerar al momento de programar este módulo lo siguiente:

- Se podrá buscar por código, nombre y apellidos.
- Al momento de modificar, no podrá alterarse el código principal
- Al eliminar, debe mostrar al usuario una ventana de confirmación antes de realizar el borrado. Si el cliente ya tiene facturas no podrá ser eliminado.
- Después de cada acción debe mostrar al usuario un mensaje de éxito o error
- Todos los atributos son obligatorios para crear o modificar

1.4. Registro de Servicio de Mantenimiento

Este módulo debe permitir gestionar los diferentes conceptos relacionados con el servicio de mantenimiento, se debe permitir **buscar, agregar, modificar y eliminar**.

Los atributos por registrar son lo siguiente:

- Código del servicio: numérico y se incrementa automáticamente de 1 en 1
- Código del cliente: valor numérico existente en Clientes
- Marca de la bicicleta: una cadena de texto no vacía y sin espacios en blanco en sus extremos
- Descripción de la bicicleta: una cadena de texto no vacía y sin espacios en blanco en sus extremos
- Precio: valor numérico sin decimales
- Fecha de recibido: Formato correcto de fecha
- Fecha de entrega: Formato correcto de fecha
- Observaciones: una cadena de texto no vacía y sin espacios en blanco en sus extremos
- Estado: Valor a seleccionar Abierto o Cerrado. Cuando este se cierre, pasará al módulo de facturación para realizar el respectivo cobro

Además, debe considerar al momento de programar este módulo lo siguiente:

- Se podrá buscar por código y nombre del cliente.
- Al momento de modificar, no podrá alterarse el código principal
- Al eliminar, debe mostrar al usuario una ventana de confirmación antes de realizar el borrado. Si el servicio ya está facturado no podrá ser eliminado.
- Después de cada acción debe mostrar al usuario un mensaje de éxito o error
- Todos los atributos son obligatorios para crear o modificar

1.5. Facturación

Este módulo debe permitir gestionar los diferentes conceptos relacionados con la facturación, se debe permitir **buscar, agregar y anular**.

Los atributos por registrar en el encabezado de una factura son lo siguiente:

- Número de factura: numérico y se incrementa automáticamente de 1 en 1. Inicia desde 1
- Código del cliente: valor numérico existente en Clientes
- Fecha de recibido: Formato correcto de fecha
- Estado: Valor a seleccionar Válido o Anulado. Este valor no debe ser manipulado por el usuario, sino que al momento de crear su estado será **Válido** y solo podrá ser cambiado cuando este se **anula**.
- Subtotal: valor numérico sin decimales
- Impuesto: valor numérico sin decimales. Este será el 13% de la suma de los artículos que se facture
- Total: valor numérico sin decimales

Los atributos por registrar en el detalle de una factura son los siguiente:

- Código de artículo: numérico y se incrementa automáticamente de 1 en 1. Inicia desde 1
- Número de factura: numérico y existente
- Cantidad: valor numérico sin decimales
- Precio unitario: valor numérico sin decimales
- Total: valor numérico sin decimales

Además, debe considerar al momento de programar este módulo lo siguiente:

- Se podrá buscar por número de factura, fecha y nombre del cliente.
- Al anular la factura, debe mostrar al usuario una ventana de confirmación.
- Después de cada acción debe mostrar al usuario un mensaje de éxito o error
- Todos los atributos son obligatorios para crear

2. Aspectos administrativos

- Trabajo en grupo de 3 personas
- Usar Netbeans
- Entrega en el Github Classroom (<https://classroom.github.com/a/3HYWdOJH>), lunes 6 de enero del 2025 hasta las 8 am.
- En cada módulo debe de tener su propio archivo para almacenar cada uno de los elementos que van registrando.
- **Tomar en cuenta toda explicación o aclaración de la explicación del proyecto por parte del profesor durante la clase, para que el estudiante no asuma requerimientos equivocado.**
- Al momento de **guardar, modificar o eliminar un elemento**, estos deben ser guardados en los archivos independientes por cada módulo, como también la lista de **objetos** que tiene en memoria.

- Toda búsqueda o selección de estos objetos, debe realizarse en memoria, es decir, debe existir una lista de objetos. No se permite hacer estas búsquedas en el archivo.
- La aplicación debe contar con interfaz gráfica, además de la funcionalidad se evaluará la estética de este.
- Al momento de seleccionar un cliente, producto o artículo, este debe ser por su valor descriptivo y no por su código. Se recomienda hacer del correcto uso del componente de selección o **combobox**
- Tomar en cuenta las validaciones de los formularios, y la documentación interna (código)

3. Documentación

La documentación es un aspecto de gran importancia en el desarrollo de programas, especialmente en tareas relacionadas con el mantenimiento de estos. Para la documentación interna, deberán incluir comentarios descriptivos para cada clase y funciones **principales**, con sus entradas, salidas, restricciones y objetivo (**Seguir javadoc**).

La documentación externa deberá incluir:

1. Portada.
2. Manual de usuario: **screenshot de las pantallas desarrolladas** y su explicación.
3. Pruebas de funcionalidad: incluir el enlace de *un video YouTube. Modo público o al menos que sea visible por el profesor. Máximo 15 minutos*
4. Descripción del problema.
5. Diseño del programa: **Descripción de las clases y las relaciones entre ellas.**
6. Librerías usadas: creación de archivos, etc.
7. Análisis de resultados: objetivos alcanzados, objetivos no alcanzados, y razones por las cuales no se alcanzaron los objetivos (en caso de haberlos).

4. Evaluación

4.1. Rubro para el programa

Descripción	Valor
Documentación interna (javadoc)	5%
Diseño Orientado a Objetos	40%
Control de acceso	5%
Registro Productos	10%
Registro de clientes	10%
Registro de servicios	15%

Descripción	Valor
Facturación	15%
TOTAL	100%

4.2. Rubros del proyecto en total

Descripción	Valor
Documentación	15%
Video explicativo YouTube *	5%
Funcionalidad del programa	80%
TOTAL	100%

- (*) Debe seguir las instrucciones
- **El diseño de la interfaz gráfica tiene un peso del 20%**