UNIVERSIDAD DE CARTAGENA

Facultad de Ingeniería

Programa de Ingeniería de Software

Diseño de una Base de Datos para la Automatización de Procesos Comerciales en una Empresa de Productos Informáticos

II Semestre

Presentado por:

Bella Samai Botello Meza

Docente:

John carlos arrieta arrieta

Fecha:

**Cartagena de Indias, 20 de mayo de 2025**

TABLA DE CONTENIDOS

1. **Introducción** ......................................................................................................... 3
2. **Objetivos** .............................................................................................................. 4  
     2.1. Objetivo General  
     2.2. Objetivos Específicos
3. **Justificación** ....................................................................................................... 5
4. **Clases Identificadas en el Ejercicio** .................................................................. 6
5. **Requerimientos Funcionales** ............................................................................. 7  
     5.1. Gestión de Productos en Venta  
     5.2. Gestión de Proveedores  
     5.3. Gestión de Productos para Alquiler  
     5.4. Gestión de Productos de Alta Tecnología  
     5.5. Gestión del Soporte Técnico entre Fabricantes  
     5.6. Gestión de Servicios Técnicos para Impresoras
6. **Modelado de Clases** ..........................................................................................9
7. **Relaciones de las Clases** .................................................................................. 39
8. **Diagrama de Clase con sus Relaciones** .......................................................... 45
9. **Bibliografía** .................................................................................................. 46

Introducción

Actualmente, la tecnología se ha vuelto un elemento fundamental en todas las áreas, incluyendo el comercio. Una tienda informática que desea modernizar la gestión de sus procesos comerciales puede beneficiarse enormemente del diseño de una base de datos que facilite la administración del inventario, las relaciones con proveedores, el alquiler de equipos y los servicios técnicos ofrecidos a los clientes.

Esto no solo brindará a la tienda la oportunidad de automatizar y optimizar sus procesos internos, sino que también proporcionará una plataforma robusta para gestionar relaciones comerciales, mejorar la atención al cliente y aumentar los ingresos mediante la diversificación de productos y servicios.

**Objetivos**

**Objetivo General:**

Diseñar una base de datos que permita a una tienda informática automatizar y optimizar la gestión de su inventario, sus relaciones con proveedores, el alquiler de equipos y los servicios técnicos ofrecidos a los clientes.

**Objetivos Específicos:**

* Analizar los procesos comerciales de la tienda relacionados con ventas, alquiler y soporte técnico.
* Identificar las entidades y relaciones clave para el diseño de la base de datos.
* Diseñar un modelo relacional que integre y organice la información de forma eficiente.
* Mejorar el control interno y la atención al cliente mediante el uso de la base de datos.

**Justificación**

En la actualidad, una tienda informática necesita manejar muchas cosas a la vez, como el inventario, los proveedores, los alquileres y el servicio técnico. Si todo se hace manualmente, es fácil que haya errores o que se pierda información importante, lo que puede afectar el rendimiento del negocio.

Por eso, diseñar una base de datos que organice toda esta información ayudará a que los procesos sean más rápidos y claros. Además, facilitará el control del inventario, gestionará las relaciones comerciales, mejorará la atención al cliente y aumentará los ingresos a través de la diversificación de productos y servicios.

Esta base de datos no solo hará que la tienda funcione mejor, sino que también permitirá ofrecer un mejor servicio y aumentar las ventas, ayudando a que el negocio crezca y se adapte a lo que hoy exige el mercado.

**Clases identificadas en el ejercicio**

1. Producto.
2. Monitor.
3. Impresora.
4. CPU.
5. DiscoDuro.
6. OtroProducto.
7. ProductoProveedor.
8. Proveedor.
9. ProductoAlquiler.
10. ProductoAltaTecnologia.
11. SoporteTecnicoFabricante.
12. EmpresaFabricante.
13. ServicioTecnico.
14. PrecioServicioTecnico.
15. Cliente.
16. ClienteImpresora

**Requerimientos funcionales**

A continuación, se presenta un análisis detallado de los requerimientos funcionales del sistema. Estos requerimientos describen las funciones esenciales que deberá cumplir la base de datos para automatizar y optimizar los procesos de la tienda informática.

1. **Gestión de productos en venta**
   * El sistema debe permitir registrar cada producto con un código único que lo identifique.
   * Debe almacenar la información específica según el tipo de producto:
     + CPU: modelo y memoria principal.
     + Impresoras: modelo y velocidad de impresión.
     + Monitores: modelo y resolución máxima.
     + Otros productos: modelo.
2. **Gestión de proveedores**
   * El sistema debe permitir registrar información de proveedores: NIF, dirección y fecha de adquisición.
   * Permitir la asociación de varios productos a múltiples proveedores.
   * Permitir consultar qué productos suministra cada proveedor.
3. **Gestión de productos para alquiler**
   * Registrar los productos que pueden alquilarse.
   * Almacenar el precio por hora de alquiler para estos productos.
   * Registrar características adicionales relevantes para alquiler (resolución en monitores, capacidad en discos duros).
   * No permitir el alquiler de productos que no sean monitores o discos duros.
4. **Gestión de productos de alta tecnología**
   * Identificar productos de alta tecnología.
   * Registrar información adicional para estos productos: país de origen y fecha de fabricación.
   * Registrar datos de la empresa fabricante (nombre, dirección, número de empleados).
5. **Gestión del soporte técnico entre fabricantes**
   * Registrar los precios del soporte técnico que ofrecen las empresas fabricantes para sus productos y productos de otras empresas.
   * Permitir comparar precios para seleccionar el proveedor de soporte más adecuado.
6. **Gestión de servicios técnicos para impresoras**
   * El sistema debe permitir registrar datos de clientes que compran impresoras (DNI, teléfono, dirección, fecha de compra).
   * El sistema debe permitir registrar detalles de los servicios técnicos realizados (detalle del servicio y total facturado).

**Modelado de clases**

1. **Producto**.

Esta clase representa a todos los objetos de tipo producto que forman parte del sistema de gestión de tienda informática. Es una clase abstracta, lo que significa que no puede ser instanciada directamente, sino que nos servirá como base para tomar ya sea sus atributos y métodos en otras clases.   
Esta clase define atributos generales y comunes que comparten todos los productos sin importar su especialización.

Propiedades de la clase

* **código**. Tipo: Caracteres. ¿Es requerido? Sí
* **modelo**. Tipo: Caracteres. ¿Es requerido? Sí

Métodos

● **Producto (codigo, modelo)**

* Objetivo: Inicializar un objeto de una subclase de Producto, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: Producto
* Parámetros: Cadena, Cadena

● **getCodigo()**

* Objetivo: Retornar el valor de la propiedad codigo.
* Ámbito: Público
* Tipo de retorno: Cadena
* Parámetros: Ninguno

● **getModelo()**

* Objetivo: Retornar el valor de la propiedad modelo.
* Ámbito: Público
* Tipo de retorno: Cadena
* Parámetros: Ninguno

1. **Monitor**.

Monitor es una subclase de la clase producto, esta clase se encargará de instanciar o crear todos los objetos de tipo monitor, cada objeto de esta clase está relacionada con la clase ProductoAlquiler, lo que indica que los monitores pueden estar disponibles para alquiler.

Propiedades de la clase

* **resolucionMaxima**. Tipo: Caracteres. ¿Es requerido?: Sí
* **alquiler**. Tipo: Objeto (ProductoAlquiler). ¿Es requerido?: Sí

Métodos

● **Monitor(codigo, modelo, resolucionMaxima, alquiler)**

* Objetivo: Crear un objeto de tipo Monitor, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: Monitor
* Parámetros: Cadena, Cadena, Cadena, Objeto ProductoAlquiler

● **getResolucionMaxima()**

* Objetivo: Retornar el valor de la propiedad resolucionMaxima.
* Ámbito: Público
* Tipo de retorno: Cadena
* Parámetros: Ninguno

● **setResolucionMaxima(valor)**

* Objetivo: Cambiar el valor de la propiedad resolucionMaxima.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Cadena

● **getAlquiler()**

* Objetivo: Retornar el valor de la propiedad alquiler.
* Ámbito: Público
* Tipo de retorno: Objeto ProductoAlquiler
* Parámetros: Ninguno

● **setAlquiler(valor)**

* Objetivo: Cambiar el valor de la propiedad alquiler.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto ProductoAlquiler

1. **Impresora**   
   La clase impresora es una subclase de la clase producto. La usaremos como plantilla para crear a todos los objetos de tipo impresora. Está asociada obligatoriamente a un objeto de tipo ServicioTecnico, lo que permite gestionar el soporte técnico relacionado con el producto.

Propiedades de la clase

* **velovidadImpresora**. Tipo: Caracteres. ¿Es requerido?: Sí
* **servicioImpresora**. Tipo: Objeto (ServicioTecnico). ¿Es requerido?: Sí

Métodos

● **Impresora(codigo, modelo, velovidadImpresora, servicioImpresora)**

* Objetivo: Crear y retornar un objeto de tipo Impresora, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: Impresora
* Parámetros: Cadena, Cadena, Cadena, Objeto ServicioTecnico

● **getVelovidadImpresora()**

* Objetivo: Retornar el valor de la propiedad velovidadImpresora.
* Ámbito: Público
* Tipo de retorno: Cadena
* Parámetros: Ninguno

● **setVelovidadImpresora(valor)**

* Objetivo: Cambiar el valor de la propiedad velovidadImpresora
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Cadena

● **getServicioImpresora()**

* Objetivo: Retornar el valor de la propiedad servicioImpresora
* Ámbito: Público
* Tipo de retorno: Objeto ServicioTecnico
* Parámetros: Ninguno

● **setServicioImpresora(valor)**

* Objetivo: Cambiar el valor de la propiedad servicioImpresora.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto ServicioTecnico

1. **CPU**

La clase CPU es una subclase de la clase Producto esta se usara como plantilla para crear a todos los objetos de tipo CPU. Esta clase relacionada con ProductoAltaTecnologia, lo que indica que este producto puede pertenecer a una categoría especializada de alta tecnología.

Propiedades de la clase

* **memoriaPrincipal**. Tipo: Caracteres. ¿Es requerido?: Sí
* **altaTecnologia**. Tipo: Objeto (ProductoAltaTecnologia). ¿Es requerido?: Sí

Métodos

● **CPU(codigo, modelo, memoriaPrincipal, altaTecnologia)**

* Objetivo: Crear y retornar un objeto de tipo CPU, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: CPU
* Parámetros: Cadena, Cadena, Cadena, Objeto ProductoAltaTecnologia

● **getMemoriaPrincipal()**

* Objetivo: Retornar el valor de la propiedad memoriaPrincipal.
* Ámbito: Público
* Tipo de retorno: Cadena
* Parámetros: Ninguno

● **setMemoriaPrincipal(valor)**

* Objetivo: Cambiar el valor de la propiedad memoriaPrincipal.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Cadena

● **getAltaTecnologia()**

* Objetivo: Retornar el valor de la propiedad altaTecnologia.
* Ámbito: Público
* Tipo de retorno: Objeto ProductoAltaTecnologia
* Parámetros: Ninguno

● **setAltaTecnologia(valor)**

* Objetivo: Cambiar el valor de la propiedad altaTecnologia.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto ProductoAltaTecnologia

1. **DiscoDuro**

Es una subclase de la clase producto, Esta clase se usará como plantilla para crear todos los objetos de tipo disco duro. Tiene una relación con las clases ProductoAltaTecnologia y ProductoAlquiler. Lo que indica que los objetos creados de tipo disco duro pueden ser alta tecnología como estar disponible para alquiler.

Propiedades de la clase

* **capacidadEspacio**. Tipo: Caracteres. ¿Es requerido?: Sí
* **altaTecnologia**. Tipo: Objeto (ProductoAltaTecnologia). ¿Es requerido?: Sí
* **alquiler**. Tipo: Objeto (ProductoAlquiler). ¿Es requerido?: Sí

Métodos

● **DiscoDuro(codigo, modelo, capacidadEspacio, altaTecnologia, alquiler)**

* Objetivo: Crear y retornar un objeto de tipo DiscoDuro, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: DiscoDuro
* Parámetros: Cadena, Cadena, Cadena, Objeto ProductoAltaTecnologia, Objeto ProductoAlquiler

● **getCapacidadEspacio()**

* Objetivo: Retornar el valor de la propiedad capacidadEspacio.
* Ámbito: Público
* Tipo de retorno: Cadena
* Parámetros: Ninguno

● **setCapacidadEspacio(valor)**

* Objetivo: Cambiar el valor de la propiedad capacidadEspacio.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Cadena

● **getAltaTecnologia()**

* Objetivo: Retornar el valor de la propiedad altaTecnologia
* Ámbito: Público
* Tipo de retorno: Objeto ProductoAltaTecnologia
* Parámetros: Ninguno

● **setAltaTecnologia(valor)**

* Objetivo: Cambiar el valor de la propiedad altaTecnologia.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto ProductoAltaTecnologia

● **getAlquiler()**

* Objetivo: Retornar el valor de la propiedad alquiler.
* Ámbito: Público
* Tipo de retorno: Objeto ProductoAlquiler
* Parámetros: Ninguno

● **setAlquiler(valor)**

* Objetivo: Cambiar el valor de la propiedad alquiler.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto ProductoAlquiler

1. **OtroProducto**

Es una subclase de la clase Producto, que sirve como categoría general para aquellos productos que no entran dentro de las categorías específicas como monitores, impresoras, CPUs o discos duros. Representa productos variados dentro del inventario de la tienda informática.

Propiedades de la clase

* **tipoProducto**. Tipo: Caracteres. ¿Es requerido?: Sí

Métodos

● **OtroProducto(codigo, modelo, tipoProducto)**

* Objetivo: Crear y retornar un objeto de tipo OtroProducto, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: OtroProducto
* Parámetros: Cadena, Cadena, Cadena

● **getTipoProducto()**

* Objetivo: Retornar el valor de la propiedad tipoProducto.
* Ámbito: Público
* Tipo de retorno: Cadena
* Parámetros: Ninguno

● **setTipoProducto(valor)**

* Objetivo: Cambiar el valor de la propiedad tipoProducto.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Cadena

1. **ProductoProvedor**

Esta clase representa la relación entre los objetos de tipo Producto y los objetos de tipo Proveedor. Su propósito principal es almacenar información detallada sobre los suministros realizados por los proveedores a la tienda. Esta clase no representa un producto ni un proveedor en sí, sino el **acto de suministro** o **historial de abastecimiento**, y cumple una función informativa y de control logístico dentro del sistema.

Propiedades de la clase

* **id**. Tipo: Caracteres. ¿Es requerido?: Sí
* **fechaAdquisicion**. Tipo: Dato. ¿Es requerido?: Sí
* **producto**. Tipo: Objeto (Producto). ¿Es requerido?: Sí
* **proveedor**. Tipo: Objeto (Proveedor). ¿Es requerido?: Sí
* **cantidadSuministrada**. Tipo: Entero. ¿Es requerido?: Sí

Métodos

● **ProductoProveedor(id, fechaAdquisicion, producto, proveedor, cantidadSuministrada)**

* Objetivo: Crear y retornar un objeto de tipo ProductoProveedor, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: ProductoProveedor
* Parámetros: Cadena, Fecha, Objeto Producto, Objeto Proveedor, Entero

● **getId()**

* Objetivo: Retornar el valor de la propiedad id.
* Ámbito: Público
* Tipo de retorno: Cadena
* Parámetros: Ninguno

● **setId(valor)**

* Objetivo: Cambiar el valor de la propiedad id.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Cadena

● **getFechaAdquisicion()**

* Objetivo: Retornar el valor de la propiedad fechaAdquisicion.
* Ámbito: Público
* Tipo de retorno: Fecha
* Parámetros: Ninguno

● **setFechaAdquisicion(valor)**

* Objetivo: Cambiar el valor de la propiedad fechaAdquisicion.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Fecha

● **getProducto()**

* Objetivo: Retornar el valor de la propiedad producto.
* Ámbito: Público
* Tipo de retorno: Objeto Producto
* Parámetros: Ninguno

● **setProducto(valor)**

* Objetivo: Cambiar el valor de la propiedad producto.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto Producto

● **getProveedor()**

* Objetivo: Retornar el valor de la propiedad proveedor.
* Ámbito: Público
* Tipo de retorno: Objeto Proveedor
* Parámetros: Ninguno

● **setProveedor(valor)**

* Objetivo: Cambiar el valor de la propiedad proveedor.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto Proveedor

● **getCantidadSuministrada()**

* Objetivo: Retornar el valor de la propiedad cantidadSuministrada.
* Ámbito: Público
* Tipo de retorno: Entero
* Parámetros: Ninguno

● **setCantidadSuministrada(valor)**

* Objetivo: Cambiar el valor de la propiedad cantidadSuministrada.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Entero

1. **Proveedor**

Esta clase representa a los objetos de tipo proveedor, quienes son los encargados de suministrar productos a la tienda informática.

Propiedades de la clase

* **IdProveedor**. Tipo: Caracteres. ¿Es requerido?: Sí
* **Nit**. Tipo: Caracteres. ¿Es requerido?: Sí
* **nombre**. Tipo: Caracteres. ¿Es requerido?: Sí
* **direccion**. Tipo: Caracteres. ¿Es requerido?: Sí

Métodos

● **Proveedor(idProveedor, nit, nombre, direccion)**

* Objetivo: Crear y retornar un objeto o instancia de tipo Proveedor, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: Proveedor
* Parámetros: Caracteres, Caracteres, Caracteres, Caracteres

● **getIdProveedor()**

* Objetivo: Retornar el valor de la propiedad IdProveedor.
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setIdProveedor(valor)**

* Objetivo: Cambiar el valor de la propiedad IdProveedor.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

● **getNit()**

* Objetivo: Retornar el valor de la propiedad Nit.
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setNit(valor)**

* Objetivo: Cambiar el valor de la propiedad Nit.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

● **getNombre()**

* Objetivo: Retornar el valor de la propiedad nombre.
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setNombre(valor)**

* Objetivo: Cambiar el valor de la propiedad nombre.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

● **getDireccion()**

* Objetivo: Retornar el valor de la propiedad direccion.
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setDireccion(valor)**

* Objetivo: Cambiar el valor de la propiedad direccion.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

1. **ProductoAlquiler.**

En esta clase se va registrar la información del precio por hora de los productos que han de ser alquilados

Propiedades de la clase

* **precioHora**. Tipo: Decimal (Double). ¿Es requerido?: Sí

Métodos

● **ProductoAlquiler(precioHora)**

* Objetivo: Crear y retornar un objeto o instancia de tipo ProductoAlquiler, asignando a su propiedad precioHora el valor recibido por parámetro.
* Ámbito: Público
* Tipo de retorno: ProductoAlquiler
* Parámetros: Double

● **getPrecioHora()**

* Objetivo: Retornar el valor de la propiedad precioHora.
* Ámbito: Público
* Tipo de retorno: Double
* Parámetros: Ninguno

● **setPrecioHora(valor)**

* Objetivo: Cambiar el valor de la propiedad precioHora.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Double

1. **ProductoAltaTecnologia.**

Esta clase representa a los objetos de tipo **producto que son considerados de alta tecnología.** Los productos que se clasifican en esta categoría son aquellos que cumplen con estándares avanzados de tecnología, como ciertos modelos de CPUs, monitores de última generación

Propiedad de la clase

* **paisOrigen**. Tipo: Caracteres. ¿Es requerido?: Sí
* **fechaFabricacion**. Tipo: Fecha. ¿Es requerido?: Sí
* **fabricante**. Tipo: Objeto (EmpresaFabricante). ¿Es requerido?: Sí

Métodos

● **ProductoAltaTecnologia(paisOrigen, fechaFabricacion, fabricante)**

* Objetivo: Crear y retornar un objeto o instancia de tipo ProductoAltaTecnologia, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: ProductoAltaTecnologia
* Parámetros: Caracteres, Fecha, Objeto EmpresaFabricante

● **getPaisOrigen()**

* Objetivo: Retornar el valor de la propiedad paisOrigen.
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setPaisOrigen(valor)**

* Objetivo: Cambiar el valor de la propiedad paisOrigen.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

● **getFechaFabricacion()**

* Objetivo: Retornar el valor de la propiedad fechaFabricacion.
* Ámbito: Público
* Tipo de retorno: Fecha
* Parámetros: Ninguno

● **setFechaFabricacion(valor)**

* Objetivo: Cambiar el valor de la propiedad fechaFabricacion.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Fecha

● **getFabricante()**

* Objetivo: Retornar el valor de la propiedad fabricante.
* Ámbito: Público
* Tipo de retorno: Objeto EmpresaFabricante
* Parámetros: Ninguno

● **setFabricante(valor)**

* Objetivo: Cambiar el valor de la propiedad fabricante.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto EmpresaFabricante

1. **SoporteTecnicoFabricante**.

Esta clase representa los servicios de soporte técnico ofrecidos directamente por una empresa fabricante para productos de alta tecnología. Cada objeto de esta clase está vinculado a una **empresa fabricante específica** y a un **producto de alta tecnología.**

Propiedades de la clase

* **precioSoporte**. Tipo: Decimal (Double). ¿Es requerido?: Sí
* **empresa**. Tipo: Objeto (EmpresaFabricante). ¿Es requerido?: Sí
* **producto**. Tipo: Objeto (ProductoAltaTecnologia). ¿Es requerido?: Sí

Métodos

● **SoporteTecnicoFabricante(precioSoporte, empresa, producto)**

* Objetivo: Crear y retornar un objeto o instancia de tipo SoporteTecnicoFabricante, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: SoporteTecnicoFabricante
* Parámetros: Decimal, Objeto EmpresaFabricante, Objeto ProductoAltaTecnologia

● **getPrecioSoporte()**

* Objetivo: Retornar el valor de la propiedad precioSoporte.
* Ámbito: Público
* Tipo de retorno: Decimal
* Parámetros: Ninguno

● **setPrecioSoporte(valor)**

* Objetivo: Cambiar el valor de la propiedad precioSoporte.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Decimal

● **getEmpresa()**

* Objetivo: Retornar el valor de la propiedad empresa.
* Ámbito: Público
* Tipo de retorno: Objeto EmpresaFabricante
* Parámetros: Ninguno

● **setEmpresa(valor)**

* Objetivo: Cambiar el valor de la propiedad empresa.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto EmpresaFabricante

● **getProducto()**

* Objetivo: Retornar el valor de la propiedad producto.
* Ámbito: Público
* Tipo de retorno: Objeto ProductoAltaTecnologia
* Parámetros: Ninguno

● **setProducto(valor)**

* Objetivo: Cambiar el valor de la propiedad producto.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto ProductoAltaTecnologia

1. **EmpresaFabricante**

Esta clase representa a las empresas que fabrican productos de alta tecnología. Los objetos de esta clase contienen información esencial sobre la empresa

Propiedades de la clase

* **nombre**. Tipo: Caracteres. ¿Es requerido?: Sí
* **direccion**. Tipo: Caracteres. ¿Es requerido?: Sí
* **numEmpleados**. Tipo: Entero. ¿Es requerido?: Sí

Métodos

● **EmpresaFabricante(nombre, direccion, numEmpleados)**

* Objetivo: Crear y retornar un objeto o instancia de tipo EmpresaFabricante, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: EmpresaFabricante
* Parámetros: Caracteres, Caracteres, Entero

● **getNombre()**

* Objetivo: Retornar el valor de la propiedad nombre.
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setNombre(valor)**

* Objetivo: Cambiar el valor de la propiedad nombre.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

● **getDireccion()**

* Objetivo: Retornar el valor de la propiedad direccion.
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setDireccion(valor)**

* Objetivo: Cambiar el valor de la propiedad direccion.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

● **getNumEmpleados()**

* Objetivo: Retornar el valor de la propiedad numEmpleados.
* Ámbito: Público
* Tipo de retorno: Entero
* Parámetros: Ninguno

● **setNumEmpleados(valor)**

* Objetivo: Cambiar el valor de la propiedad numEmpleados.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Entero

1. **ServicioTecnico**

Esta clase representa los servicios técnicos realizados a productos del sistema, especialmente impresoras.

Propiedades de la clase

* **detalleServicio**. Tipo: Caracteres. ¿Es requerido?: Sí
* **totalFacturado**. Tipo: Decimal (Double). ¿Es requerido?: Sí
* **fecha**. Tipo: Fecha. ¿Es requerido?: Sí
* **precioAplicado**. Tipo: Objeto (PrecioServicioTecnico). ¿Es requerido?: Sí
* **clienteImp**. Tipo: Objeto (ClienteImpresora). ¿Es requerido?: Sí
* **impresora**. Tipo: Objeto (Impresora). ¿Es requerido?: Sí

Métodos

● **ServicioTecnico(detalleServicio, totalFacturado, fecha, precioAplicado, clienteImp, impresora)**

* Objetivo: Crear y retornar un objeto o instancia de tipo ServicioTecnico, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: ServicioTecnico
* Parámetros: Caracteres, Decimal (Double), Fecha, Objeto PrecioServicioTecnico, Objeto ClienteImpresora, Objeto Impresora

● **getDetalleServicio()**

* Objetivo: Retornar el valor de la propiedad detalleServicio.
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setDetalleServicio(valor)**

* Objetivo: Cambiar el valor de la propiedad detalleServicio.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

● **getTotalFacturado()**

* Objetivo: Retornar el valor de la propiedad totalFacturado.
* Ámbito: Público
* Tipo de retorno: Decimal (Double)
* Parámetros: Ninguno

● **setTotalFacturado(valor)**

* Objetivo: Cambiar el valor de la propiedad totalFacturado.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Decimal (Double)

● **getFecha()**

* Objetivo: Retornar el valor de la propiedad fecha.
* Ámbito: Público
* Tipo de retorno: Fecha
* Parámetros: Ninguno

● **setFecha(valor)**

* Objetivo: Cambiar el valor de la propiedad fecha.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Fecha

● **getPrecioAplicado()**

* Objetivo: Retornar el valor de la propiedad precioAplicado.
* Ámbito: Público
* Tipo de retorno: Objeto PrecioServicioTecnico
* Parámetros: Ninguno

● **setPrecioAplicado(valor)**

* Objetivo: Cambiar el valor de la propiedad precioAplicado.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto PrecioServicioTecnico

● **getClienteImp()**

* Objetivo: Retornar el valor de la propiedad clienteImp.
* Ámbito: Público
* Tipo de retorno: Objeto ClienteImpresora
* Parámetros: Ninguno

● **setClienteImp(valor)**

* Objetivo: Cambiar el valor de la propiedad clienteImp.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto ClienteImpresora

● **getImpresora()**

* Objetivo: Retornar el valor de la propiedad impresora.
* Ámbito: Público
* Tipo de retorno: Objeto Impresora
* Parámetros: Ninguno

● **setImpresora(valor)**

* Objetivo: Cambiar el valor de la propiedad impresora.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto Impresora

1. **PrecioServicioTecnico.**

Esta clase almacena los precios aplicables a los servicios técnicos

Propiedades de la clase

* **precio**. Tipo: Decimal (Double). ¿Es requerido?: Sí
* **fechaAplicacion**. Tipo: Fecha. ¿Es requerido?: Sí

Métodos

● **PrecioServicioTecnico(precio, fechaAplicacion)**

* Objetivo: Crear y retornar un objeto o instancia de tipo PrecioServicioTecnico, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: PrecioServicioTecnico
* Parámetros: Decimal (Double), Fecha

● **getPrecio()**

* Objetivo: Retornar el valor de la propiedad precio.
* Ámbito: Público
* Tipo de retorno: Decimal (Double)
* Parámetros: Ninguno

● **setPrecio(valor)**

* Objetivo: Cambiar el valor de la propiedad precio.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Decimal (Double)

● **getFechaAplicacion()**

* Objetivo: Retornar el valor de la propiedad fechaAplicacion.
* Ámbito: Público
* Tipo de retorno: Fecha
* Parámetros: Ninguno

● **setFechaAplicacion(valor)**

* Objetivo: Cambiar el valor de la propiedad fechaAplicacion.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Fecha

1. **Cliente**

Esta clase se usará como plantilla para crear todos los objetos de tipo cliente, quienes pueden adquirir productos o solicitar servicios técnicos (especialmente para impresoras).

Propiedades de la clase

* **codigo**. Tipo: Caracteres. ¿Es requerido?: Sí
* **id**. Tipo: Caracteres. ¿Es requerido?: Sí
* **nombre**. Tipo: Caracteres. ¿Es requerido?: Sí
* **telefono**. Tipo: Caracteres. ¿Es requerido?: Sí
* **direccion**. Tipo: Caracteres. ¿Es requerido?: Sí
* **gmail**. Tipo: Caracteres. ¿Es requerido?: Sí

Métodos

● **Cliente(codigo, id, nombre, telefono, direccion, gmail)**

* Objetivo: Crear y retornar un objeto o instancia de tipo Cliente, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: Cliente
* Parámetros: Caracteres, Caracteres, Caracteres, Caracteres, Caracteres, Caracteres

● **getCodigo()**

* Objetivo: Retornar el valor de la propiedad código
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setCodigo(valor)**

* Objetivo: Cambiar el valor de la propiedad código
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

● **getId()**

* Objetivo: Retornar el valor de la propiedad id
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setId(valor)**

* Objetivo: Cambiar el valor de la propiedad id
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

● **getNombre()**

* Objetivo: Retornar el valor de la propiedad nombre
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setNombre(valor)**

* Objetivo: Cambiar el valor de la propiedad nombre
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

● **getTelefono()**

* Objetivo: Retornar el valor de la propiedad teléfono
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setTelefono(valor)**

* Objetivo: Cambiar el valor de la propiedad teléfono
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

● **getDireccion()**

* Objetivo: Retornar el valor de la propiedad dirección
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setDireccion(valor)**

* Objetivo: Cambiar el valor de la propiedad direccion
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

● **getGmail()**

* Objetivo: Retornar el valor de la propiedad Gmail
* Ámbito: Público
* Tipo de retorno: Caracteres
* Parámetros: Ninguno

● **setGmail(valor)**

* Objetivo: Cambiar el valor de la propiedad Gmail
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Caracteres

**ClienteImpresora**

La clase ClienteImpresora es una subclase de la clase Cliente. La usaremos como plantilla para crear objetos de clientes que están relacionados directamente con una impresora. Esto permite gestionar servicios técnicos personalizados o seguimiento especializado según el equipo que utilizan.

**Propiedades de la clase**  
• impresoraAsociada. Tipo: Objeto (Impresora). ¿Es requerido?: Sí

**Métodos**  
● ClienteImpresora(codigoCliente, nombre, id, email, telefono, direccion, impresoraAsociada)

* Objetivo: Crear y retornar un objeto de tipo ClienteImpresora, asignando a sus propiedades los valores recibidos por parámetro.
* Ámbito: Público
* Tipo de retorno: ClienteImpresora  
  o Parámetros: Cadena, Cadena, Cadena, Cadena, Cadena, Cadena, Objeto Impresora

● getImpresoraAsociada()

* Objetivo: Retornar el valor de la propiedad impresoraAsociada.
* Ámbito: Público
* Tipo de retorno: Objeto Impresora
* Parámetros: Ninguno

● setImpresoraAsociada(valor)

* Objetivo: Cambiar el valor de la propiedad impresoraAsociada.
* Ámbito: Público
* Tipo de retorno: Nada
* Parámetros: Objeto Impresora

## Relaciones de las clases

### **1. Herencia**

La herencia se utiliza para expresar que una clase deriva de otra, heredando sus atributos y comportamientos. En este modelo:

* Producto es una clase base de la cual derivan varias clases:
  + CPU
  + Impresora
  + Monitor
  + DiscoDuro
  + OtroProducto

Estas subclases heredan las propiedades y métodos generales de Producto

* ClienteImpresora hereda de Cliente, indicando que es un tipo específico de cliente asociado al uso de impresoras.

**2. Agregación / Composición**

Estas relaciones expresan que una clase contiene o se compone de otra. En este caso, se utilizan asociaciones con multiplicidad y el símbolo o-- para indicar una **agregación** (relación débil):

* **CPU** y **DiscoDuro** pueden tener un ProductoAltaTecnologia (0 o 1).  
  Esto indica que no todos los CPUs o discos duros tienen esta característica, pero es posible.
* **Monitor** y **DiscoDuro** también pueden estar asociados con un ProductoAlquiler, indicando que pueden estar disponibles para alquiler.
* **Impresora** puede tener un ServicioTecnico, lo cual representa una relación de soporte técnico exclusivo entre estos objetos.

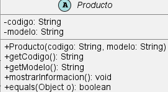
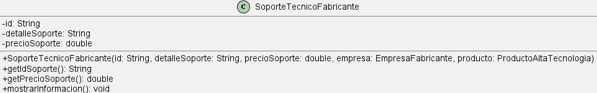
### **3. Asociación (Relación general entre clases)**

Las asociaciones representan relaciones más generales entre clases:

* Un ProductoAltaTecnologia es fabricado por exactamente una EmpresaFabricante.
* ProductoProveedor es una clase que relaciona un Producto con un Proveedor, permitiendo modelar la entrega de productos por diferentes proveedores.
* Un ServicioTecnico se relaciona con:
  + Un ClienteImpresora
  + Una Impresora
  + Un PrecioServicioTecnico

Esto modela el servicio técnico específico brindado a una impresora de un cliente, con su precio respectivo.

* SoporteTecnicoFabricante se asocia tanto con un ProductoAltaTecnologia como con una EmpresaFabricante, modelando el soporte especializado que un fabricante brinda para sus productos avanzados.

Clases sin relaciones:

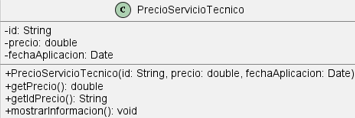
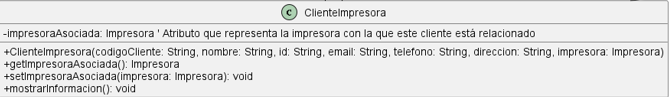
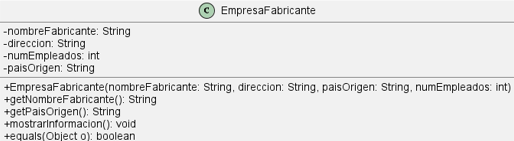
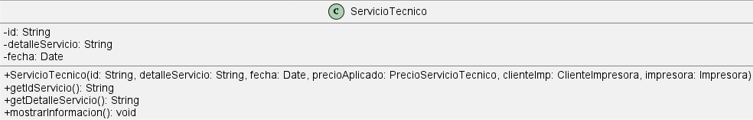
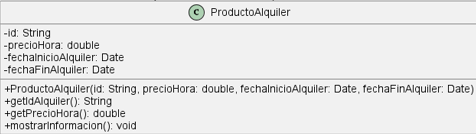
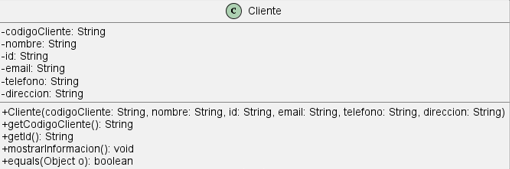
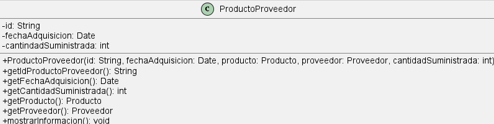
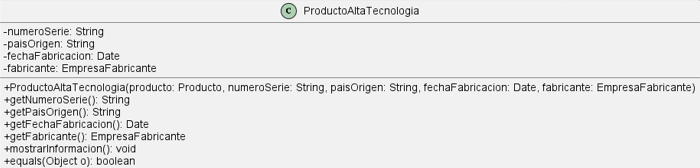


Diagrama de clase con sus relaciones

[ActividadUnidad2/co/edu/udc/poo/bellabotello/tiendaDeInformatica/Documentacion/DiagramaClasesNew.puml at main · Bella200625/ActividadUnidad2 · GitHub](https://github.com/Bella200625/ActividadUnidad2/blob/main/co/edu/udc/poo/bellabotello/tiendaDeInformatica/Documentacion/DiagramaClasesNew.puml)

Enlace de GitHub:

<https://github.com/Bella200625/ActividadUnidad2.git>

Video:

<https://youtu.be/TAvTHZTqfqA?feature=shared>

BIBLIOGRAFIAS

 **Deitel, P., & Deitel, H. M. (2020).**  
*Java: How to Program (11th Edition)*. Pearson Education.

 **Sierra, K., & Bates, B. (2019).**  
*Head First Java (2nd Edition)*. O'Reilly Media.

 **Larman, C. (2005).**  
*Applying UML and Patterns: An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition)*. Prentice Hall.

 **Gamma, E., Helm, R., Johnson, R., & Vlissides, J. (1994).**  
*Design Patterns: Elements of Reusable Object-Oriented Software*. Addison-Wesley.

 **Gaddis, T. (2021).**  
*Starting Out with Java: From Control Structures through Objects (8th Edition)*. Pearson.