

**PRÁCTICA 1**

**PROGRAMACIÓN ORIENTADA A OBJETOS**

**PROGRAMACIÓN EN JAVA**

Integrantes:

Erik Alexander González Cardona

Emilio Porras Mejía

Esteban García Carmona

Felipe Miranda Arboleda

Profesor

Jaime Alberto Guzmán Luna

Universidad Nacional de Colombia sede Medellín  
Facultad de Minas  
Medellín, Colombia  
Diciembre 2021

## Descripción general de la solución:

Generic IT necesita de una aplicación para que el Administrador lleve el control de sus empleados y productos, así como las interacciones entre ellos y sus clientes. Proponemos una solución orientada a objetos en la cual cada actor involucrado en el sistema estará representado. En él, el administrador podrá ordenar a sus empleados para que lleven a cabo la cadena de acciones que llevan a completar con éxito un servicio. También, generará los elementos que se derivan de estas (facturación, entrega de productos, etc.).

## Características de programación orientada a objetos:

- **Clase abstracta y métodos abstractos:** Empleado es una clase abstracta de la cual heredan las clases Tecnico y Dependiente. Empleado, a su vez, implementa los métodos abstractos quitarServicio, asignarServicio y cobrarServicio de la interfaz Personal. Estos métodos son utilizados por los objetos de tipo Tecnico y Dependiente.
- **Interfaz:** La interfaz Personal, como fue indicado anteriormente, contiene métodos abstractos requeridos por la clase Empleado.
- **Herencia:** Las clases Tecnico y Dependiente heredan de la clase Empleado, puesto que hay comportamientos similares en ambas.
- **Ligaduras dinámicas:** En la funcionalidad “Liquidar” se utiliza la ligadura dinámica en dos oportunidades. Cuando se itera sobre la lista de todos los empleados, se desambigua si se utiliza el método cobrarSalario de Tecnico o de Dependiente a través de la ligadura dinámica. Similarmente, se utiliza ligadura para discriminar entre el toString de Tecnico y de Dependiente a la hora de definir el mensaje de recepción del salario.
- **Atributo y método de clase:** La lista componentes de la clase Bodega, que almacena todos los componentes disponibles en la Bodega, es de clase. Similarmente, tanto los métodos agregarComponente como quitarComponente de Bodega son de clase.
- **Constante:** el atributo MARGEN\_GANANCIA de la clase Dependiente es una constante que contiene el factor por el que se debe multiplicar el costo de un servicio para obtener lo que se debe cobrar.
- **Encapsulamiento:** El método buscarComponente de la clase Tecnico, al ser un método auxiliar del método reparar, es private. Similarmente, el mencionado método reparar de Tecnico es public y es fundamental para las funcionalidades. Además, la lista servicios de Empleado es protected, ya que se requiere esta visibilidad con tal de ser llamada desde los constructores de Dependiente y Tecnico.
- **Sobrecarga de método:** El método sacarComponente de la clase Bodega, si recibe una String con el nombre del componente, devuelve un objeto componente correspondiente al nombre. Por otro lado, si recibe un objeto de tipo Componente, lo retira de la lista de componentes de la bodega y lo retorna nuevamente.
- **Sobrecarga de constructores:** Para la generación de clientes se utiliza un constructor cuando es generado automáticamente y uno diferente cuando se ingresan sus datos de forma manual. Adicionalmente, para crear un Componente del Producto de un Cliente, se utiliza un constructor diferente al utilizado para la Bodega, puesto que del primero no se lleva cuenta de su precio.
- **Manejo de this:** En ambos constructores de Producto, se utiliza this para diferenciar entre los atributos de instancia que se están inicializando y los parámetros que recibe el constructor.
- **Manejo de this():** En uno de los constructores de cliente y de producto, se utiliza el this() para llamar al otro constructor de la respectiva clase.
- **Enumeración:** PrecioComponente corresponde a una enumeración de la cual se obtienen los precios de los diferentes componentes a través de su nombre

## Descripción de las 5 funcionalidades:

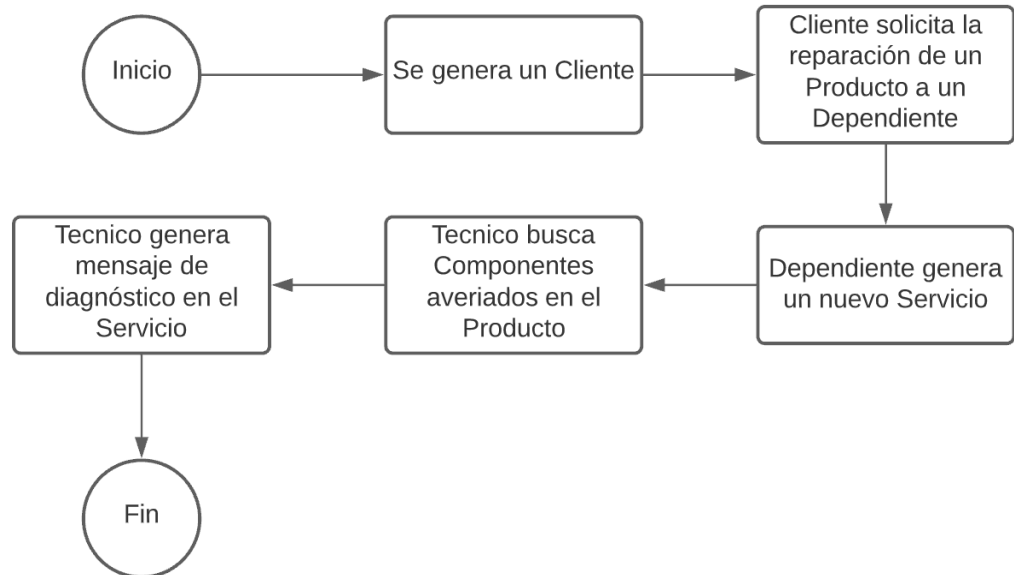
1. **Diagnosticar:** Funcionalidad que describe el proceso desde que llega un cliente hasta que su producto está próximo a ser reparado.

Objetos que intervienen en la funcionalidad:

- un Cliente

- un Producto
- un Dependiente
- un Servicio
- un Tecnico
- varios Componentes

Secuencia del proceso:



Resultados al usuario:

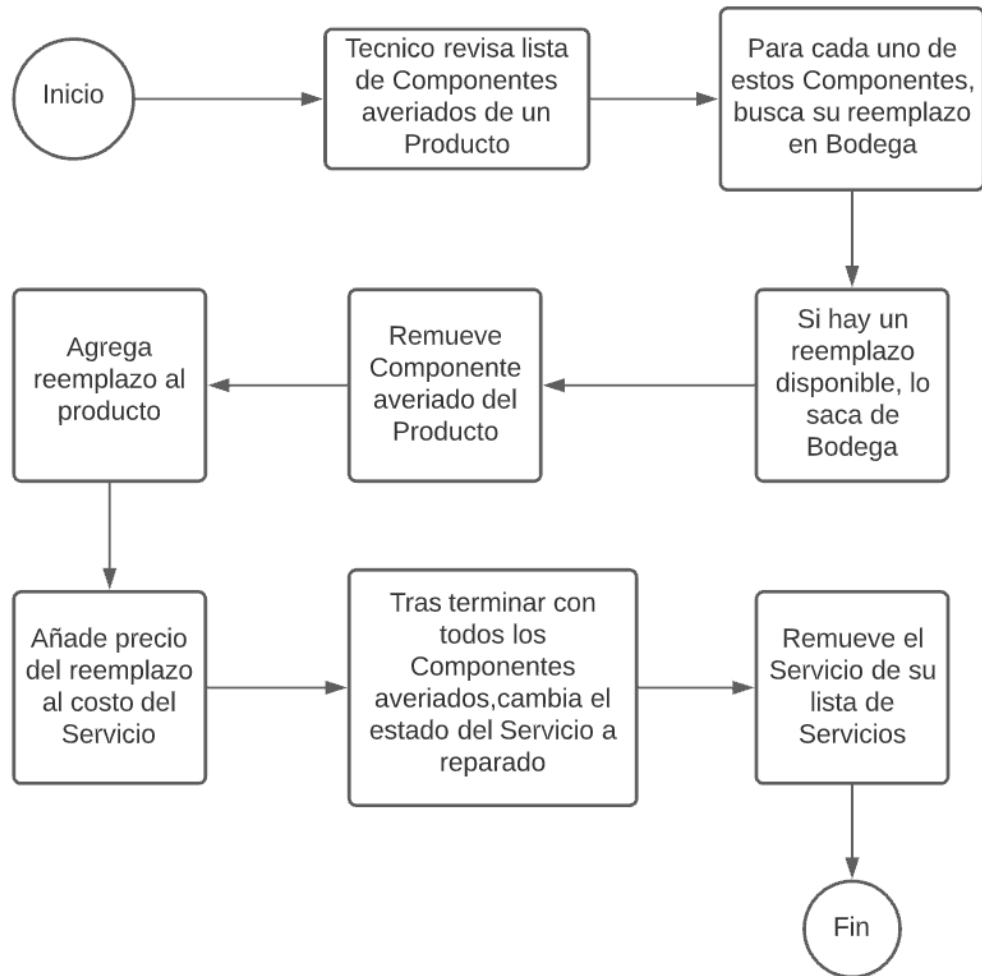
Se genero el cliente ID:0 Nombre: Esteban CC: 9699183 Cartera: 812551.0  
 El cliente fue atendido exitosamente por Camila!  
 Ya puede consultar en los servicios para iniciar su diagnostico o reparacion.  
 Se encontraron problemas en los siguientes componentes del producto:  
 [Memoria 4g Kingsgton, Display 17 pulgadas]  
 Ya puede volver al menu principal para solicitar reparacion

## 2. **Reparar:** Funcionalidad que lleva a cabo detalladamente la reparación de un producto.

Objetos que intervienen en la funcionalidad:

- un Tecnico
- un Producto
- varios Componentes
- (static) Bodega
- un Servicio

Secuencia del proceso:



Resultados al usuario:

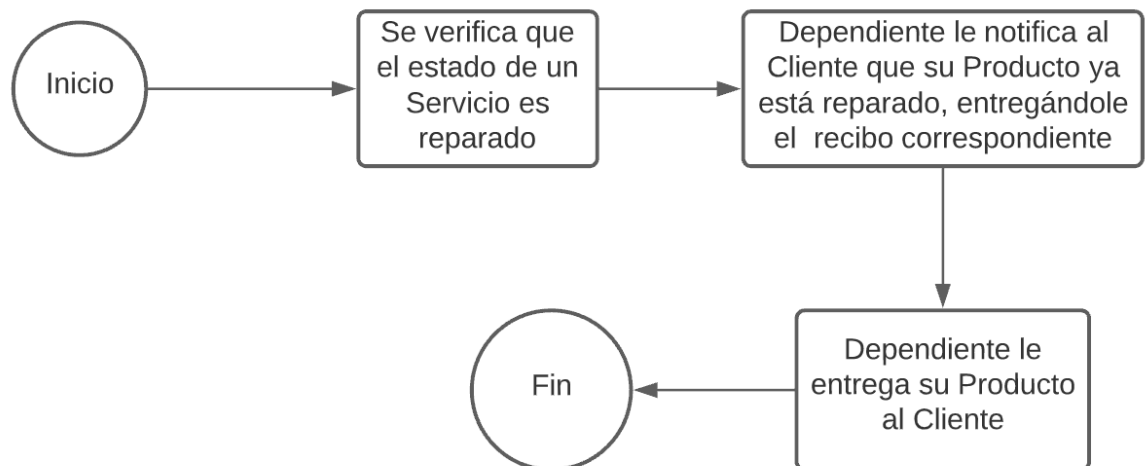
El servicio de Esteban fue arreglado por Tecnico:  
Emilio y tuvo un costo para la empresa de 140000.0

3. **Finalizar Servicio:** Funcionalidad que realiza los últimos pasos de un servicio sin incluir su cobro.

Objetos que intervienen en la funcionalidad:

- un Servicio
- un Dependiente
- un Cliente
- un Producto

Secuencia del Proceso:



Resultados al usuario:

Factura #0

Cliente: Esteban con cedula 9699183

Recibir el producto: HP XPS

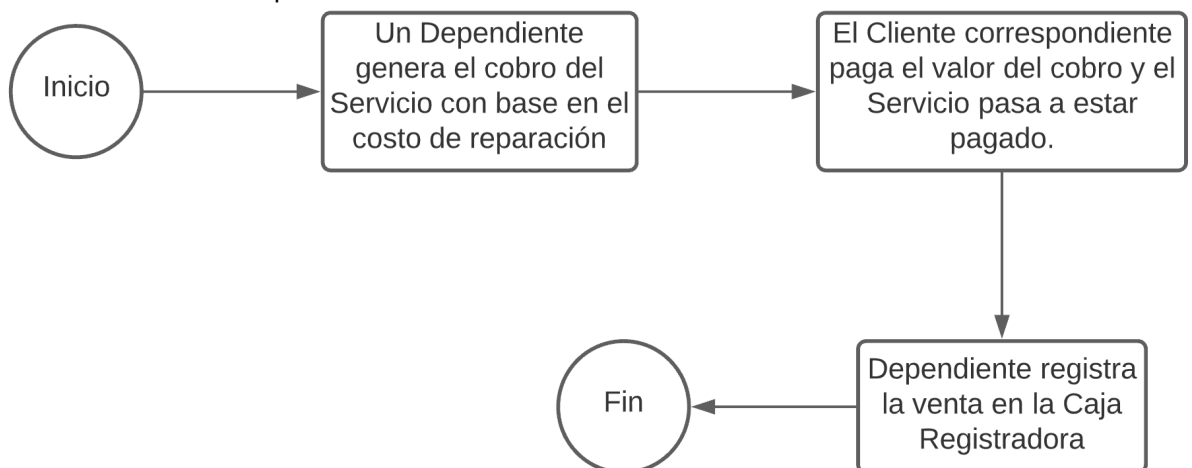
El servicio ya esta listo para ser cobrado

4. **Cobrar Servicio:** Funcionalidad que hace todo el proceso de cobro al cliente y de registro de pagos.

Objetos que intervienen en la funcionalidad:

- un Dependiente
- un Servicio
- un Cliente
- una CajaRegistradora

Secuencia del proceso:



Resultados al usuario:

Se cobra el servicio por un total de 210000.0

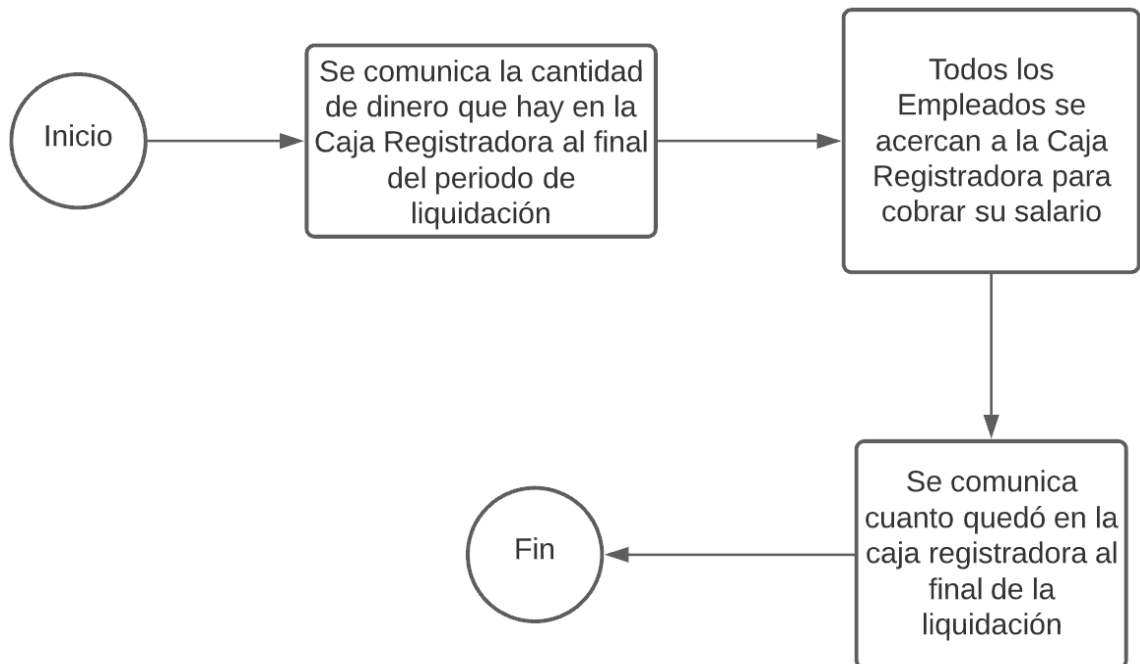
En la caja registradora ahora hay 210000.0 pesos

5. **Liquidar:** Funcionalidad que lleva a cabo las acciones referentes a la liquidación de un periodo.

Objetos que intervienen en la funcionalidad:

- una CajaRegistradora
- varios Empleados
- varios Tecnicos
- varios Dependientes

Secuencia del proceso:

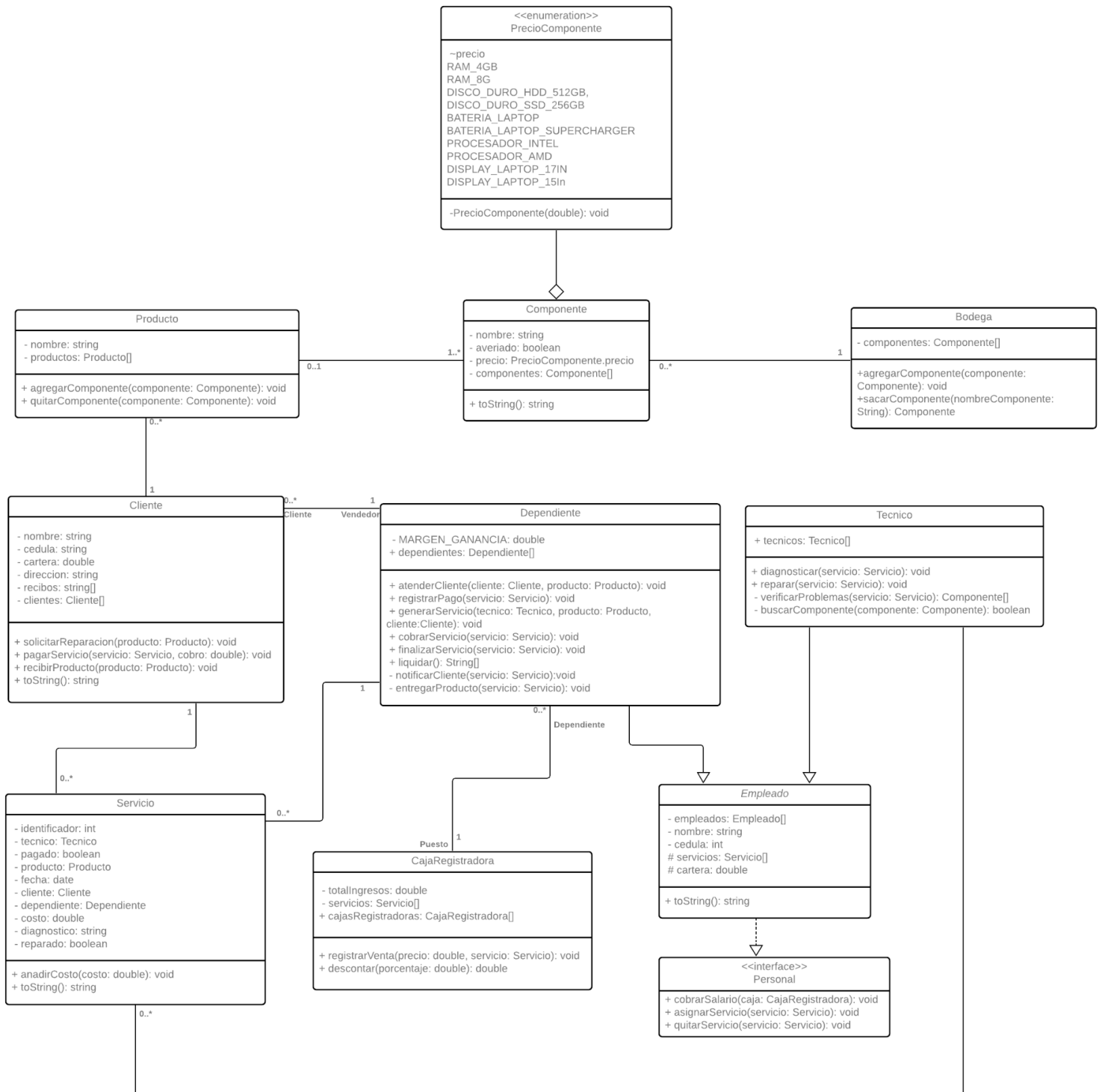


Resultados al usuario:

En la caja registradora hay 210000 antes de liquidar.  
El Dependiente: Camila ha recibido 2100 por su trabajo.  
El Tecnico: Emilio ha recibido 4200 por su trabajo.  
En la caja registradora quedan 203700

## Descripción del diseño estático del sistema en la especificación UML:

### Diagrama de Clases



## Objetos

- **Producto:** Cada producto representará la solución técnica que un Cliente requiere su reparación. La lista productos es estática.
- **Cliente:** Un Cliente será la persona que se recibe “en tiempo real”, de la que se requiere guardar su información con datos como nombre, cédula, productos a reparar. La lista clientes es estática
- **PrecioComponente:** Es un enum que guardará la información de los precios con los que se podrá interactuar en el sistema, es decir, el precio de los componentes que se venden en Generic IT.
- **Componente:** Un componente es cada parte que puede ser reparada de un producto. Este puede ser varias cosas: pantallas, procesadores, cámaras, etc. Cuando se evalúa un componente, debe decidirse si este está averiado o no. La venta (a modo de reparación) de estos es lo que le genera ingresos a la compañía. La lista componentes es estática
- **Bodega:** la bodega es la clase en la que se almacenan todos los componentes disponibles para las reparaciones. Sirve para mantener una idea de qué está disponible o no. La lista componentes es estática. El método sacarComponente es sobrecargado.
- **Servicio:** un servicio es un acto en asociación de la empresa y el cliente, donde se guarda la información de todo lo que se debe hacer y revisar del producto del cliente. Este servicio, realizado por un dependiente específico, debe luego ser atendido por un técnico, que es un empleado que corrige las complicaciones que puede tener el producto del cliente.
- **Personal:** Interfaz que define métodos relevantes para la clase Empleado.
- **Empleado:** Clase abstracta que implementa la interfaz Personal y representa los requerimientos organizacionales y humanos que debe tener cada dependiente y técnico de la compañía. El método toString de empleado es un método abstracto. La lista de empleados es estática.
- **Técnico:** Hereda de la clase Empleado, está en capacidad de arreglar productos.. La lista tecnicos es estática.
- **CajaRegistradora:** Clase encargada de mover y almacenar el dinero, así como de registrar las ventas. La lista cajasRegistradoras es estática.
- **Dependiente:** Hereda de la clase Empleado. Atenderá a los clientes, diligenciará los servicios y será el responsable del flujo de caja. La lista dependientes es estática.

## Manual de usuario:

El programa expone una serie de opciones, las cuales se seleccionan escribiendo el numeral y luego apretando “Enter”. Por ejemplo, si se desea entrar a “Menu diagnosticar un producto”, tan solo se debe escribir el número 1, y apretar enter. Esto aplicará para todas las operaciones que se realizan en el sistema. Tras completar una función o navegar entre menús, si aparece el mensaje “Presione Enter para continuar”, se debe actuar como el programa impera. Presione “Enter” y será redireccionado a la siguiente pantalla.

Al abrir el programa, se encuentra las siguientes opciones:

“Que desea hacer?”

1. Menu diagnosticar un producto
2. Reparar un producto



3. Finalizar un servicio
  4. Cobrar un servicio
  5. Liquidación del periodo
- Opciones alternativas
6. Mostrar clientes
  7. Mostrar servicios
  8. Guardar y cerrar
- Elija una opción: “

Si se desea usar la primera funcionalidad, se debe apretar “1”. Este nos mostrará la siguiente lista:

1. Crear cliente manualmente
2. Generar cliente
3. Solicitar servicio
4. Diagnosticar producto
5. Volver al menu principal

Ya estando en el submenú, se puede apretar la primera opción para generar a mano un cliente. Se solicitará su nombre, cédula y cartera (presupuesto). Si por ejemplo, se fuera a ingresar los datos de una persona llamada Luna, con cédula 1000454243, dirección en la calle 34 sur y un presupuesto de 1000000, se debería ingresar:

Luna, y apretamos enter. 1000454243, y apretamos enter, calle 34 sur, y apretamos enter, 1000000, y apretamos enter.

También se puede crear automáticamente un cliente, con generar cliente. Solamente es elegir la opción y no hay que hacer nada más.

Tras generar un cliente de alguna de las dos maneras, se debe continuar solicitando un servicio. Se selecciona la opción “Solicitar un servicio”. Si no se conoce cuál es la id del cliente para el servicio, se tendrá que volver a la pantalla principal, apretando “5” y en submenú y luego, ya estando en la pantalla principal, apretando “Mostrar servicios” y buscando a la persona en cuestión. Si en alguna de estas elecciones elige un número incorrecto, no pasa nada, dado que el código no falla, simplemente informa de que se ha elegido una opción inválida.

Tras solicitar el servicio, se deberá diagnosticar. Para esto, en el submenú elegir la opción “Diagnosticar producto”. Este informará qué componentes están fallando. Ya podemos volver al menú principal.

Ahora debemos elegir la opción 2 en el menú principal “Reparar un producto”. Este nos pedirá que escribamos el nombre del producto a reparar. Si se desconoce, puede volver a la pantalla principal e ir a “mostrar servicios”, donde se encuentran enumerados. Continuando, esta funcionalidad nos dirá el nombre del técnico y el costo de reparación para la empresa (no es el lo que se le cobrará al cliente).

Ya reparado el producto, se puede ir en el menú a la opción 3, Finalizar un servicio. Se debe escribir el id del servicio, este nos imprimirá la boleta o factura, que tiene al cliente, el costo total y el producto a recibir.

Completado el servicio, se debe cobrar el servicio. Se aprieta la opción “4”, cobrar un servicio. Se elige el servicio con el id. Se dice si podrá pagar el servicio el cliente. Dice el precio que paga el cliente por el servicio y el dinero total que queda en la caja registradora después del proceso.

La última funcionalidad es la liquidación del período, que se accede usando la opción 5 “Liquidación del periodo”. Esta se hace cada vez que se le desee pagar a los empleados. Se debe tener en cuenta que su salario depende en cuánto se vende. No es un salario fijo. Al apretar la opción, nos mostrará en pantalla cuánto dinero había antes de la liquidación, se le da a cada empleado su respectiva paga, y luego se muestra cuánto dinero queda en la caja tras haberles pagado.

Para ver clientes que ha atendido anteriormente, se debe apretar la opción "6" . Para ver servicios en proceso y terminados, se debe apretar la opción "7" .  
Al terminar una operación, se debe apretar la opción "8", para guardar el trabajo realizado y cerrar el programa, de otro modo, se perderá el trabajo realizado.