

Programación Orientada a Objetos

Nombre: Peisy Abigail Quillupangui Tupe

NRC: 1323

C. Lectura N° 2

Fecha: /11/2024

C.I: 1753918653

Actividad

Diseña 5 objetos diferentes con su correspondiente diagrama UML asegurándose de mostrar las relaciones entre ellos

Ejemplo

- **Motocicleta:** con atributos como marca, modelo, año, kilometraje, color, número de chasis y métodos como reparar(), diagnosticar()
- **Cliente:** como atributos como nombre, apellido, teléfono, dirección y método como solicitar Servicio()
- **Mecánico:** con atributos como nombre, especialidad y años de experiencias y método como realizar reparación(), diagnosticar()
- **Repuesto:** con atributos como marca, código, precio, cantidad en stock
- **Orden de trabajo:** con atributos como fecha de ingreso, fecha de salida, motocicleta, cliente, mecánico, lista de repuesto, proveedores, costo total y métodos con abrir(), cerrar()

Relaciones

Motocicleta y orden de trabajo.

Cliente y orden de trabajo.

Mecánico y orden de trabajo.

Orden de trabajo y repuesto.

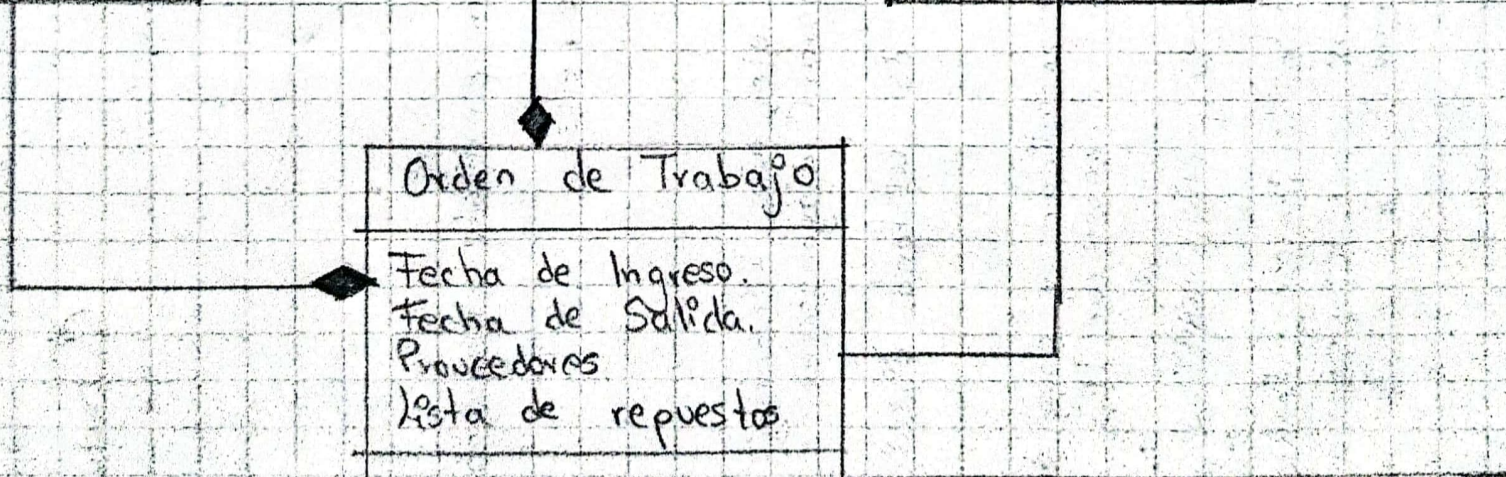
Motocicleta
Nº Motor Marca Modelo Año Color
Reparar

Cliente
Nombre Apellido Teléfono Dirección
Solicitar servicio

Mecánico
Nombre Especialidad Año experiencia
Reparación Diagnosticar

Repuesto
Marca Codigo Precio

Orden de Trabajo
Fecha de Ingreso Fecha de Salida Procedimientos Lista de repuestos
Abrir Cerrar



Código.

```
1  Main.java :
2  public class Motocicleta {
3      public String Numero de Motor;
4      public String Marca;
5      public String Modelo;
6      int Año;
7      public String color;
8
9      public void Reparar ( ) {
10         System.out.println ("Motocicleta" + "Numero de Motor" + "Marca" +
11         "Modelo" + "Año" + "Color.");
12     }
13 }
14 public class Cliente {
15     public String Nombre;
16     public String Apellido;
17     public String Dirección;
18     int numero;
19
20     public void Solicitar Servicio ( ) {
21         System.out.println ("Cliente" + "Nombre" + "Apellido" + "Dirección" +
22         "Numero.");
23     }
24 }
25 public class Mecanico {
26     public String Nombre;
27     public String Especialidad;
28     int Año Experiencia;
29
30     public void Reparación() {
31         System.out.println ("Mecanico" + "Nombre" + "Especialidad" +
32         "Año Experiencia.");
33     }
34
35     public void Diagnosticar ( ) {
36         System.out.println ("Mecanico" + "Nombre" + "Especialidad" +
37         "Año experiencia.");
38     }
39 }
40 public class Repuesto {
41     public String Marca;
42     public String Código;
43     public String Precio;
44 }
45
46 public class Orden de trabajo {
47     public String Proveedores;
48     LocalDate = Fecha de Ingreso;
49     LocalDate = Fecha de Salida;
50     ArrayList = Lista de Repuestos;
```



```

49 public void Abrir () {
50     System.out.println ("Proveedores" + "Fecha de Ingreso" + "Fecha de Salida" +
51     "Lista de Repuestos.");
52 }
53 public void Cerrar () {
54     System.out.println ("Proveedores" + "Fecha de Ingreso" + "Fecha de Salida" +
55     "Lista de Repuestos.");
56 }
57 }
58
59
60
61
62
63

```

Resumen de resolución

Objetivo.

Crear un taller mecánico de motos utilizando el UML para obtener la representación visual y estructurada del sistema.

Desarrollo

Se crean las principales entidades del taller (motoциcletas, clientes, mecánicos, repuestos y ordenes de trabajo con sus respectivas relaciones.

Se ha utilizado el diagrama de clases UML para representar los 5 objetos (atributos, métodos y relaciones entre ellos).

Utilizamos el código para plasmarlo en java y verificar el resultado.

Como resultados hemos obtenido un diagrama de clases que refleja la estructura del taller mecánico, permitiendo una mejor comprensión de las interacciones entre los diferentes componentes.

Conclusión

El uso de la escritura, el diagrama uml y código ha facilitado la visualización y comprensión del sistema, lo que es de gran utilidad para el desarrollo de un software de gestión para el taller.