

UNIVERSIDAD DE LAS FUERZAS ARMADAS (ESPE)

CARRERAS TECNOLOGIA DE LA INFORMACIÓN

PRIMER PARCIAL

COMPUTACIÓN DIGITAL

“TEMA: SISTEMAS DE NUMERACIÓN

AUTOR:

Guamán Ortega

Cristina Alexandra

PARALELO:

NRC-5404

DOCENTE:

Luis Enrique Jaramillo Montaña

PERÍODO:

Octubre – marzo 2024

FECHA ENTREGA:

08 de diciembre del 2024

SANGOLQUI – ECUADOR

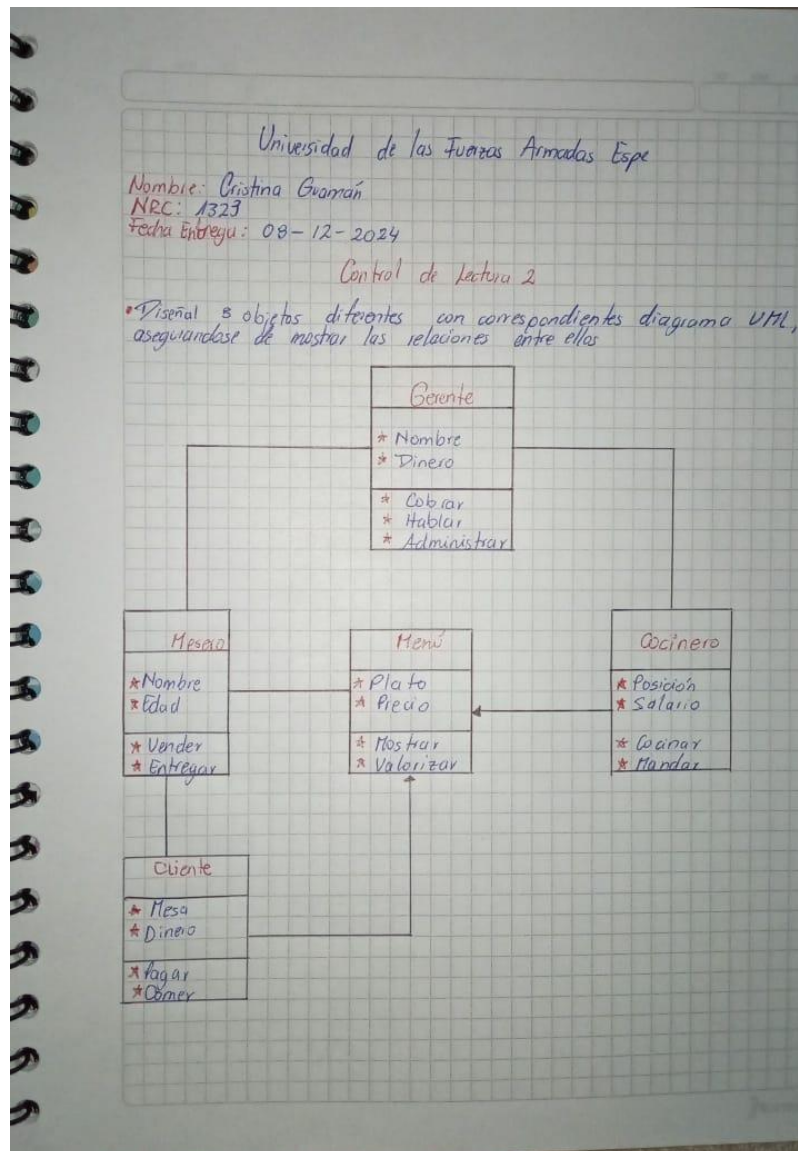
Qué es Java

Java es un lenguaje de programación utilizado para crear software compatible con una gran diversidad de sistemas operativos. Este lenguaje tiene la particularidad de ser compilado e interpretado al mismo tiempo; esto significa que es un lenguaje simplificado que convierte automáticamente el código en instrucciones de máquina.

Qué es UML

El Lenguaje Unificado de Modelado (UML) fue creado para forjar un lenguaje de modelado visual común y semántica y sintácticamente rico para la arquitectura, el diseño y la implementación de sistemas de software complejos, tanto en estructura como en comportamiento.

Uml grafica



Codigo uml

El gerente tiene la relacion de asociacion con el mesero y el cocinero y el mesero tiene la relacion de asociacion con el cliente y el menú el cocinero y el cliente dependen del menú.

Gerente

Constructor para Inicialización: Considera añadir constructores para inicializar los atributos de las clases



```
1- /*****
2
3- Welcome to GDB Online.
4- GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,
5- C#, OCaml, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.
6- Code, Compile, Run and Debug online from anywhere in world.
7
8- *****/
9- // Clase Gerente
10- class Gerente {
11-     private String nombre;
12-     private double dinero;
13
14-     public void cobrar() {
15-         System.out.println("El gerente está cobrando.");
16-     }
17
18-     public void hablar() {
19-         System.out.println("El gerente está hablando.");
20-     }
21
22-     protected void administrar() {
23-         System.out.println("El gerente está administrando.");
24-     }
25- }
```

Mesero

Un cliente podría interactuar con un mesero o el gerente podría supervisar a un cocinero. Considera implementar relaciones entre clases, como agregar un atributo <Mesero>



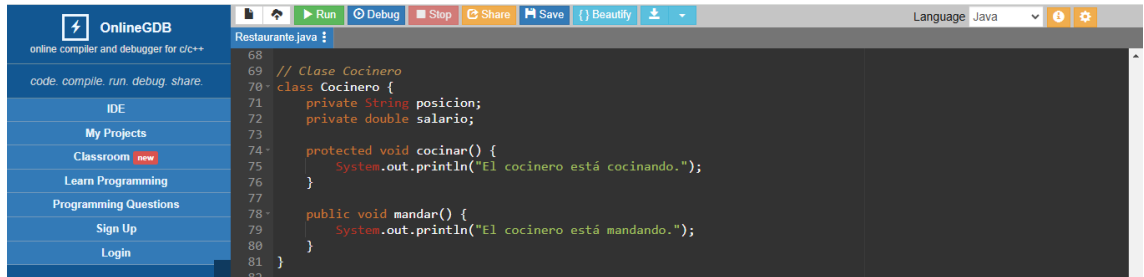
```
26
27 // Clase Mesero
28 class Mesero {
29     private String nombre;
30     private int edad;
31
32     public void vender() {
33         System.out.println("El mesero está vendiendo.");
34     }
35
36     protected void entregar() {
37         System.out.println("El mesero está entregando.");
38     }
39 }
```

Menú



```
54
55 // Clase Menu
56 class Menu {
57     private String plato;
58     private double precio;
59
60     public void mostrar() {
61         System.out.println("Mostrando el menú.");
62     }
63
64     protected void valorizar() {
65         System.out.println("Valorizando los precios del menú.");
66     }
67 }
68
```

Cocinero



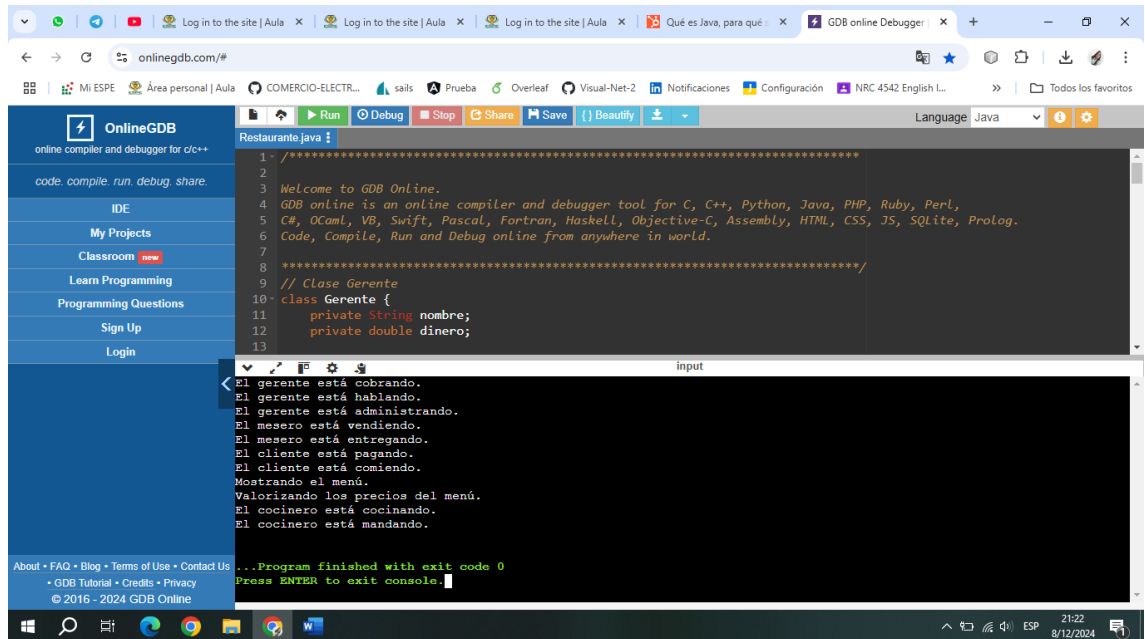
```
68
69 // Clase Cocinero
70 class Cocinero {
71     private String posicion;
72     private double salario;
73
74     protected void cocinar() {
75         System.out.println("El cocinero está cocinando.");
76     }
77
78     public void mandar() {
79         System.out.println("El cocinero está mandando.");
80     }
81 }
82
```

Clase principal para probar el código



```
83 // Clase Principal para probar el programa
84 public class Restaurante {
85     public static void main(String[] args) {
86         // Crear instancias
87         Gerente gerente = new Gerente();
88         Mesero mesero = new Mesero();
89         Cliente cliente = new Cliente();
90         Menu menu = new Menu();
91         Cocinero cocinero = new Cocinero();
92
93         // Invocar métodos
94         gerente.cobrar();
95         gerente.hablar();
96         gerente.administrar();
97
98         mesero.vender();
99         mesero.entregar();
100
101         cliente.pagar();
102         cliente.comer();
103
104         menu.mostrar();
105         menu.valorizar();
106
107         cocinero.cocinar();
108         cocinero.mandar();
109     }
110 }
```

EJECUCION DEL CODIGO DERESTAURANTE



The screenshot shows the OnlineGDB IDE interface. The left sidebar contains navigation links: OnlineGDB, code.compile.run.debug.share., IDE, My Projects, Classroom, Learn Programming, Programming Questions, Sign Up, and Login. The main editor displays a Java file named 'Restaurante.java' with the following code:

```
1- /*****
2-
3- Welcome to GDB Online.
4- GDB online is an online compiler and debugger tool for C, C++, Python, Java, PHP, Ruby, Perl,
5- C#, OCaml, VB, Swift, Pascal, Fortran, Haskell, Objective-C, Assembly, HTML, CSS, JS, SQLite, Prolog.
6- Code, Compile, Run and Debug online from anywhere in world.
7-
8- *****/
9- // Clase Gerente
10- class Gerente {
11-     private String nombre;
12-     private double dinero;
13- }
```

The output window at the bottom shows the execution results:

```
El gerente está cobrando.
El gerente está hablando.
El gerente está administrando.
El mesero está vendiendo.
El mesero está entregando.
El cliente está pagando.
El cliente está comiendo.
Mostrando el menú.
Valorizando los precios del menú.
El cocinero está cocinando.
El cocinero está mandando.
...Program finished with exit code 0
Press ENTER to exit console.
```

Resumen

El código modela un restaurante usando programación incluye clases para los roles principales (Gerente, Mesero, Cliente, Cocinero) y una para el Menú. Cada clase tiene atributos privados y métodos que representan sus acciones específicas como *cobrar* (Gerente), *vender* (Mesero), *cocinar* (Cocinero), y *pagar* (Cliente). Ya que en la clase principal Restaurante se crean instancias de estas clases y se ejecutan sus métodos para simular las actividades de un restaurante.