



**CÓDIGO FUENTE GENERADO A PARTIR DE LAS CLASES
GA4-220501095-AA23EV01**

**Aprendiz:
Paula Andrea Vargas Moreno**

**Instructor:
Jorge Callejas**

**PROGRAMACIÓN DE APLICACIONES Y SERVICIOS PARA LA NUBE
(2879881)
SERVICIO NACIONAL DE APRENDIZAJE SENA
CENTRO DE LA CONSTRUCCIÓN - CALI
24 JULIO DE 2024**



INTRODUCCIÓN

El informe estará enfocado en el desarrollo de código fuente generado desde el diagrama de clases del proyecto a desarrollar al finalizar el programa técnico, tendremos en cuenta conceptos básicos que se maneja en la programación orientada a objeto, debido a este requerimiento el código se desarrolló en el lenguaje de JAVA, teniendo en cuenta las sesiones sincrónicas que nos brindó el área técnica del SENA y también el material de estudio suministrado por el técnico.



- El código debe cumplir con todas las convenciones para las clases, atributos y métodos utilizadas en el lenguaje Java y el código debe tener comentarios.

CLASE USUARIO

```
user-class.java X
user-class.java
1  public class Usuarios {
2      // definir atributos
3      public String name;
4      public String email;
5      private String password;
6
7      //Constructor
8      public Usuarios(String name, String email, String password){
9          this.name = name;
10         this.email = email;
11         this.password = password;
12     }
13
14     //Crear metodo sin retorno por el void
15     public void obtenerDatosUser() {
16         System.out.println("Nombre: " + name);
17     }
18
19     //Crear metodo de la clase Usuarios con retorno
20     public String datosUser (String name, String email){
21         return this.name + email
22     }
23
24     //Principal - Objeto a partir del constructor
25     public static void main(String[] args){
26         Usuarios usuario1 = new Usuarios ("Andrea", "pav@gmail.com");
27
28         //Llamar el método para mostrar el datos del usuario.
29         usuario1.datosUser();
30     }
31
32 }
```



buy-class.java X

buy-class.java

```
1  public class Compra {
2      // definir atributos
3      public double id_buy ;
4      public String pay_method;
5      public double buy_date ;
6      public Int value;
7
8      //Constructor
9      public Compra (double id_buy, String pay_method, double buy_date, Int value){
10         this.id_buy = id_buy;
11         this.pay_method = pay_method;
12         this.buy_date = buy_date;
13         this.value = value;
14     }
15     //Crear metodo sin retorno por el void
16     public void obtenerValueCompra() {
17         System.out.println("Valor: " + value);
18     }
19
20     //Creamos metodos de la clase Compra con retorno
21     public String obtenerDatosCompra (double id_buy, Int value){
22         return this.id_buy + value
23     }
24
25     //Principal - Objeto a partir del constructor
26     public static void main(Int[] args){
27         Compra compra1 = new Compra (200, "abc123");
28
29         //Llamar el método para mostrar valor compra.
30         compra1.obtenerValueCompra();
31     }
32 }
```



CLASE CONSULTORIAS

```
consulting-class.java M X
consulting-class.java
1  public class Consultorias {
2      // definir atributos
3      public String name_contact;
4      public String name_company;
5      public String email;
6      public Int mobile;
7      public String message;
8      public StringBoolean type_service; // si el input es de select el dato como lo debo poner
9      public Boolean documents;
10
11     //Constructor
12     public Consultorias (String name_contact, String name_company, String email, Int mobile,
13     String message, Boolean type_service, Boolean documents){
14         this.name_contact = name_contact;
15         this.name_company = name_company;
16         this.email = email;
17         this.mobile = mobile;
18         this.message = message;
19         this.type_service = type_service;
20         this.documents = documents;
21     }
22
23     //Crear metodo sin retorno por el void
24     public void obtenerDatosConsultoria() {
25         System.out.println("Contacto: " + name_contact + ", Empresa: " + name_company +
26         ", Tipo Servicio: " + type_service);
27     }
28
29     // Creamos metodo de la clase Consultoria con retorno
30     public String datosConsultoria (String name_company, String type_service){
31         return this.name_company + type_service;
32     }
33
34     //Principal-Objeto a partir del constructor
35     public static void main (String[] args){
36         Consultorias consultoria1 = new Consultorias ("Emermedica");
37
38         //Llamar el método para mostrar datos de la consultoria
39         consultoria1.obtenerDatosConsultoria();
40     }
```



SUBCLASE SERVICIOS

servicios-class.java X

servicios-class.java

```
1  public class Servicios extends Consultorias {
2      // definir atributos de la subclase
3      public String descripcion_servicio;
4
5      //Constructor
6      public Servicios (String descripcion_servicio){
7          this.descripcion_servicio = descripcion_servicio;
8      }
9
10     //Crear metodo sin retorno
11     public void obtenerServicio(){
12         System.out.println("El servicio de la consultoria es: " + descripcion_servicio);
13     }
14
15     // Getters y Setters
16     public String getServicio() {
17         return descripcion_servicio;
18     }
19
20     public void setServicio(String descripcion_servicio) {
21         this.descripcion_servicio = descripcion_servicio;
22     }
23 }
```