

Definir estándares de codificación de acuerdo con la plataforma de desarrollo elegida.

Actividad: GA7-220501096-AA1-EV02

Wendy González López CC 1022406984

Grupo: 2879641

Servicio Nacional de Aprendizaje

Centro de Formación Minero Ambiental

Tecnología en Análisis y Desarrollo de Software

2025

## **Introducción**

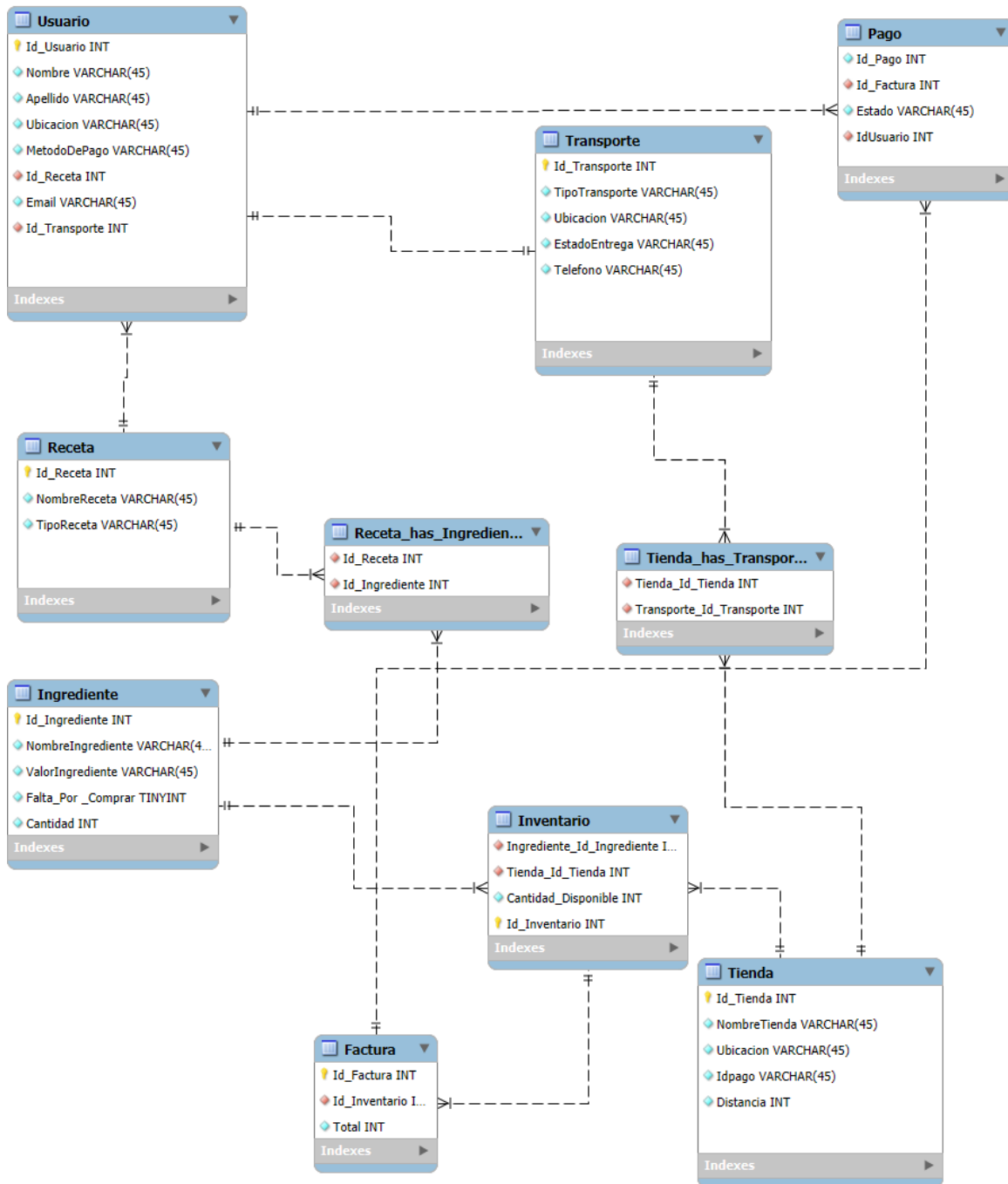
En siguiente documento se plantean los principales elementos de la aplicación web de VeciCocina, teniendo en cuenta el planteamiento de los requerimientos y los diagramas anteriormente realizados. Se tendrá en cuenta el paradigma orientado a objetos en el programa de java, se comenzara por plantear cada uno de las clases, los métodos y los objetos de esta aplicación.

## **Aplicación de POO en VeciCocina**

En el lenguaje de programación Java se estructura en clases, que son los componentes básicos del lenguaje. Cada clase contiene datos y métodos, que se utilizan para manipular esos datos. Java también incluye una serie de tipos de datos incorporados, como enteros, números en coma flotante y caracteres. Estos tipos de datos pueden combinarse para crear estructuras de datos más complejas, como matrices y objetos. Java es un lenguaje de programación orientado a objetos, lo que significa que está diseñado para trabajar con objetos. Los objetos son instancias de clases, que contienen datos y métodos para manipular esos datos.

La programación orientada a objetos permite un alto grado de abstracción y encapsulación, lo que facilita la escritura de sistemas de software complejos. Java incluye una serie de características que lo convierten en una potente herramienta para la programación orientada a objetos, como la herencia y el polimorfismo. La herencia permite a los desarrolladores crear nuevas clases que heredan propiedades y métodos de clases existentes. Esto facilita la reutilización del código y la escritura de programas más eficaces. El polimorfismo permite que los objetos adopten múltiples formas. Por ejemplo, una subclase puede utilizarse en lugar de su superclase, lo que permite una mayor flexibilidad en la programación.

Para tener claridad en cada uno de los elementos de la aplicación se anexa el diagrama de clases de la aplicación



A continuación, definimos los elementos de la aplicación VeciCocina:

Cada tabla del modelo se convierte en una clase:

- Usuario
- Receta

- Ingrediente
- Inventario
- Tienda
- Transporte
- Factura
- Pago

Atributos: según las columnas del diagrama se pueden definir los siguientes

- Clase Usuario

```
public class Usuario {  
    private int idUsuario;  
    private String nombre;  
    private String apellido;  
    private String ubicacion;  
    private String metodoDePago;  
    private String email;  
  
    // Relaciones  
    private Receta receta;  
    private Transporte transporte;  
}
```

- Clase Ingrediente

```
public class Ingrediente {  
    private int idIngrediente;  
    private String nombreIngrediente;  
    private String valorIngrediente;  
    private boolean faltaPorComprar;  
    private int cantidad;  
}
```

- Clase Inventario

```
public class Inventario {  
    private int idInventario;  
    private Ingrediente ingrediente;  
    private Tienda tienda;  
    private int cantidadDisponible;  
}
```

- Clase Tienda

```
public class Tienda {  
    private int idTienda;  
    private String nombreTienda;  
    private String ubicacion;  
    private String idPago;  
    private int distancia;  
}
```

- Clase Transporte

```
public class Transporte {  
    private int idTransporte;  
    private String tipoTransporte;  
    private String ubicacion;  
    private String estadoEntrega;  
    private String telefono;  
}
```

- Clase Factura

```
public class Factura {  
    private int idFactura;  
    private int total;  
    private List<Pago> pagos;  
}
```

- Clase Pago

```
public class Pago {  
    private int idPago;  
    private Factura factura;  
    private Usuario usuario;  
    private String estado;  
}
```

Métodos generales: los métodos que tendremos en cada una de las clases serán:

```
// Getters y Setters  
// Constructor vacío y con parámetros  
// toString()
```

Los métodos específicos para cada clase están definidos así:

- En Usuario

```
public void realizarPago(Pago pago) { ... }  
public void agregarReceta(Receta receta) { ... }
```

- En Receta

```
public void agregarIngrediente(Ingrediente ingrediente) { ... }  
public int calcularCostoTotal() { ... }
```

- En Inventario

```
public boolean verificarDisponibilidad(int cantidad) { ... }  
public void actualizarStock(int nuevaCantidad) { ... }
```

- En Tienda

```
public void registrarTransporte(Transporte transporte) { ... }  
public void mostrarInventario() { ... }
```

- En Transporte

```
public void actualizarEstadoEntrega(String nuevoEstado) { ... }
```

- En Factura

```
public void calcularTotal() { ... }  
public void agregarPago(Pago pago) { ... }
```

## **Conclusiones**

En el anterior ejercicio se pudo observar que la manera mas sencilla de determinar los diferentes elementos para trabajarlos en java fue utilizar el diagrama de clases y con este darse cuenta de los atributos de cada una de las clases. Adicional se debe tener en cuenta que el hecho de usar el paradigma orientado a objetos permite que se pueda reciclar algunos de los métodos ya que se pueden usar en varias partes del código sin afectar la calidad del programa y permitiendo ahorrar tiempo. Es importante determinar cuales son los objetos y como se van a ligar con los métodos para poder tener un sistema funcionando sin interrupción.



## Referencias

- Aulab Hakcademy. (s.f.). *Diferencia entre lenguajes de programación compilados e interpretados*. Recuperado el 2 de 12 de 2024, de <https://aulab.es/noticia/18/diferencia-entre-lenguajes-de-programacion-compilados-e-interpretados>
- Ceballos, J. (2000). Programacion orientada a objetos . En J. Ceballos, *Curso de programacion* (págs. 23-27). Ra-Ma.
- Moraguez, E. R. (s.f.). *Desarrollo Tecnológico*. Recuperado el 19 de 10 de 2025, de ¿Qué es Java: cómo funciona y para qué sirve?: <https://lovtechnology.com/que-es-java-como-funciona-y-para-que-sirve/>
- SENA. (s.f.). *Conceptos basicos de programacion orientada a objetos*. Recuperado el 01 de 08 de 2025, de [https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF16/index.html#/curso/tema1#t\\_1\\_5](https://zajuna.sena.edu.co/Repositorio/Titulada/institution/SENA/Tecnologia/228118/Contenido/OVA/CF16/index.html#/curso/tema1#t_1_5)