

# Actividad 01

## Ingeniería de Software y CMMI

### Información Estudiante

Nombre	Edgar Arturo
Apellido	Jiménez Centeno
Carnet	19001704
Curso	Metodología de Desarrollo de Software
Fecha límite de entrega	09-Jun- 2025
Total de páginas	10

# Planteamiento del Problema

## Problema “Sistema de pedidos”

El proyecto consiste en una aplicación que permite realizar operaciones básicas sobre pedidos: agregar, editar, listar y eliminar. La gestión de estas acciones se realiza mediante sesiones de usuario; sin embargo, cualquier persona, incluso sin autenticarse, podrá crear (colocar) un pedido. Para tener acceso a todas las funciones (editar, listar y eliminar), el usuario deberá registrarse proporcionando los siguientes datos:

- Name
  - UserName
- Password
  - Email

Cada pedido contendrá la siguiente información:

- ID del pedido
  - nombre de quien envía el pedido
  - tipo de pedido
  - descripción
- fecha de recolección
  - fecha de entrega
  - cantidad de artículos
  - costo

## Instrucciones

☒ **Fase 1: Análisis.**

Realizar el análisis correspondiente para la realización del proyecto, presentar propuesta y estimar tiempos y aproximado de costos.

☒ **Fase 2: Diseño**

Crear un diseño con algún diagrama UML:

- diagrama de casos de uso,
- diagrama de clases, o
- diagrama de secuencias.

☒ **Fase 3: Código**

Realizar el código de la aplicación mediante un proyecto de Java Standard Edition, con interfaz de línea de comandos. Puede utilizar el IDE de su preferencia para escribir el código (Netbeans, Eclipse, JDeveloper, entre otros). Adjuntar el código en el mismo archivo de la tarea a entregar.

☒ La actividad debe entregarse en formato PDF.

# Solución

## Fase 1: Análisis

### Problema: “Agenda de Contactos”

Ámbito	Solución Propuesta
Determinar el ámbito del proyecto	Aplicación de gestión de pedidos a través de consola, con funciones CRUD (Crear, Leer, Actualizar, Eliminar). Uso de sesiones para controlar el acceso a funcionalidades.
Definir recursos preliminares	Recurso humano (desarrollador). Recursos de software y herramientas.
Obtener recursos principales	Visual Studio Code (IDE), Java JDK, sistema operativo actualizado, desarrollador.

### Análisis y requisitos del software

Actividad	Descripción
Especificaciones preliminares del software	CRUD completo para la entidad "Pedido". Control de usuarios mediante sesiones.
Realizar análisis de necesidades	Se requiere un sistema que permita a cualquier usuario colocar pedidos y que los usuarios registrados puedan gestionarlos completamente.
Revisar las especificaciones del software	Registro de usuario con los siguientes campos: Name, UserName, Password, Email.  Datos del pedido: ID del pedido, nombre de quien envía el pedido, tipo de pedido, descripción, fecha de recolección, fecha de entrega, cantidad de artículos, costo.

### Análisis y requisitos del hardware

Requerimiento	Descripción
Equipo a necesitar para desarrollo	Computadora con mínimo 4 GB de RAM, procesador Dual Core o superior. Idealmente verificar cuales son los requisitos mínimos del sistema elegido por el equipo de desarrollo.
Equipo a necesitar para implementación	Cualquier equipo capaz de ejecutar Java con mínimo 3 GB de RAM, sistema operativo actualizado.

### Tiempos y costos estimados

Actividad	Estimación
Tiempo de desarrollo	5 días (8 horas de análisis y diseño, 24 horas de desarrollo de funcionalidades, 8 horas de pruebas y ajustes)
Costo estimado del desarrollo	Aproximadamente entre Q1,500 y Q2,500, dependiendo del nivel de detalle requerido.
Herramientas necesarias	Visual Studio Code(agregar la Extension Pack for java, Java JDK), sistema operativo actualizado.

Fase 2: Diseño

Diagrama de Casos de Uso

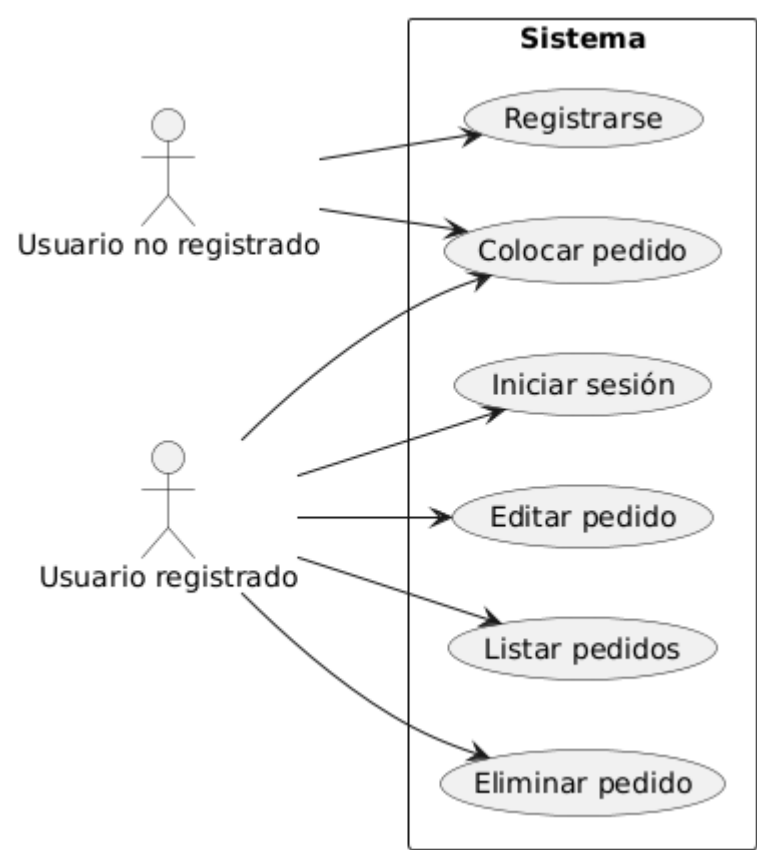
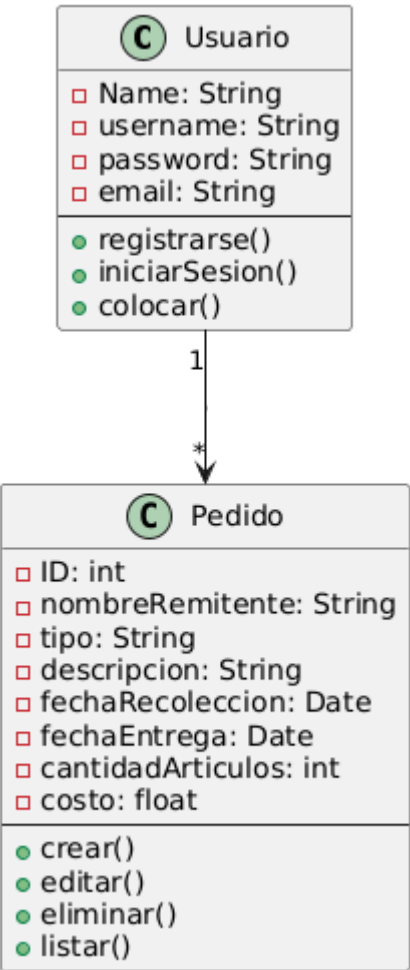
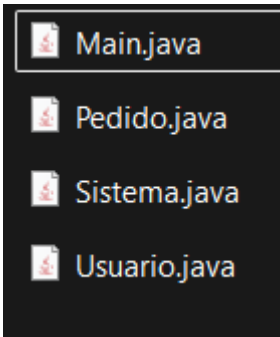


Diagrama de clases:



## Fase 3: Código

El siguiente código está dividido en



Se puede clonar desde el siguiente enlace

[https://github.com/EdgarJimenez19001704/Metodologia\\_Actividad01](https://github.com/EdgarJimenez19001704/Metodologia_Actividad01)

o leer a continuación:

### Main.java

```
package SistemaPedidos;

public class Main {
    public static void main(String[] args) {
        Sistema sistema = new Sistema();
        sistema.iniciar();
    }
}
```

### Sistema.java

```
package SistemaPedidos;

import java.util.*;

public class Sistema {
    private Scanner scanner = new Scanner(System.in);
    private Map<String, Usuario> usuarios = new HashMap<>();
    private List<Pedido> pedidos = new ArrayList<>();
    private Usuario usuarioActual = null;

    public void iniciar() {
        int opcion;
        do {
            mostrarMenu();
            opcion = Integer.parseInt(scanner.nextLine());
            switch (opcion) {
                case 1 -> colocarPedido();
                case 2 -> registrarse();
                case 3 -> iniciarSesion();
                case 4 -> {
                    if (usuarioActual != null) {
```

```
        editarPedido();
    } else {
        System.out.println("Debe iniciar sesión.");
    }
}

case 5 -> {
    if (usuarioActual != null) {
        eliminarPedido();
    } else {
        System.out.println("Debe iniciar sesión.");
    }
}

case 6 -> {
    if (usuarioActual != null) {
        listarPedidos();
    } else {
        System.out.println("Debe iniciar sesión.");
    }
}

case 0 -> System.out.println("Saliendo del sistema...");
default -> System.out.println("Opción no válida.");
}

} while (opcion != 0);
}

private void mostrarMenu() {
    System.out.println("\n--- MENÚ ---");
    System.out.println("1. Colocar pedido");
    System.out.println("2. Registrarse");
    System.out.println("3. Iniciar sesión");
    System.out.println("4. Editar pedido");
    System.out.println("5. Eliminar pedido");
    System.out.println("6. Listar pedidos");
    System.out.print("Seleccione una opción: ");
}

private void registrarse() {
    String Name = Validador.leerTexto(scanner, "Name: ");
    System.out.print("Username: ");
    String username = scanner.nextLine();
    System.out.print("Password: ");
    String password = scanner.nextLine();
    String email = Validador.leerEmail(scanner, "Email: ");
    usuarios.put(username, new Usuario(Name, username, password, email));
    System.out.println("Registro exitoso.");
}

private void iniciarSesion() {
    System.out.print("Username: ");
    String username = scanner.nextLine();
    System.out.print("Password: ");
    String password = scanner.nextLine();
    Usuario usuario = usuarios.get(username);
```



```
        if (usuario != null && usuario.getPassword().equals(password)) {
            usuarioActual = usuario;
            System.out.println("Inicio de sesión exitoso.");
        } else {
            System.out.println("Credenciales incorrectas.");
        }
    }

    private void colocarPedido() {
        Pedido pedido = new Pedido();
        pedido.capturarDatos(scanner);
        pedidos.add(pedido);
        System.out.println("Pedido colocado correctamente.");
    }

    private void editarPedido() {
        System.out.print("ID del pedido a editar: ");
        int id = Integer.parseInt(scanner.nextLine());
        for (Pedido pedido : pedidos) {
            if (pedido.getId() == id) {
                pedido.capturarDatos(scanner);
                System.out.println("Pedido editado.");
                return;
            }
        }
        System.out.println("Pedido no encontrado.");
    }

    private void eliminarPedido() {
        System.out.print("ID del pedido a eliminar: ");
        int id = Integer.parseInt(scanner.nextLine());
        pedidos.removeIf(pedido -> pedido.getId() == id);
        System.out.println("Pedido eliminado (si existía).");
    }

    private void listarPedidos() {
        if (pedidos.isEmpty()) {
            System.out.println("No hay pedidos.");
        } else {
            for (Pedido pedido : pedidos) {
                System.out.println(pedido);
            }
        }
    }
}
```

## Usuario.java

```
package SistemaPedidos;

public class Usuario {
    private String name;
```



```
private String username;
private String password;
private String email;

public Usuario(String name, String username, String password, String email)
{
    this.name = name;
    this.username = username;
    this.password = password;
    this.email = email;
}

public String getUsername() {
    return username;
}

public String getPassword() {
    return password;
}
}
```

Pedido.java

```
package SistemaPedidos;

import java.util.Scanner;

public class Pedido {
    private static int contador = 1;
    private int id;
    private String nombreRemitente;
    private String tipo;
    private String descripcion;
    private String fechaRecoleccion;
    private String fechaEntrega;
    private int cantidad;
    private double costo;

    public Pedido() {
        this.id = contador++;
    }

    public void capturarDatos(Scanner scanner) {
        this.nombreRemitente = Validador.leerTexto(scanner, "Nombre del remitente: ");
        this.tipo = Validador.leerTexto(scanner, "Tipo de pedido: ");
        System.out.print("Descripción: ");
        this.descripcion = scanner.nextLine();
        System.out.print("Fecha de recolección: ");
        this.fechaRecoleccion = scanner.nextLine();
        System.out.print("Fecha de entrega: ");
        this.fechaEntrega = scanner.nextLine();
    }
}
```



```
        this.cantidad = Validador.leerEntero(scanner, "Cantidad de artículos: ");
    };

    this.costo = Validador.leerDouble(scanner, "Costo: ");
}

public int getId() {
    return id;
}

@Override
public String toString() {
    return "Pedido #" + id + " - " + nombreRemitente + " - " + tipo + " - " + descripcion + " - Recolección: " + fechaRecoleccion + ", Entrega: " + fechaEntrega + ", Cantidad: " + cantidad + ", Costo: Q" + costo;
}
}
```

### Validador.java

```
package SistemaPedidos;

import java.util.regex.Pattern;
import java.util.Scanner;

public class Validador {

    public static String leerTexto(Scanner scanner, String mensaje) {
        String input;
        do {
            System.out.print(mensaje);
            input = scanner.nextLine();
        } while (!input.matches("[a-zA-ZáéíóúÁÉÍÓÚñÑ ]+"));
        return input;
    }

    public static int leerEntero(Scanner scanner, String mensaje) {
        String input;
        while (true) {
            System.out.print(mensaje);
            input = scanner.nextLine();
            try {
                return Integer.parseInt(input);
            } catch (NumberFormatException e) {
                System.out.println("Debe ingresar un número entero.");
            }
        }
    }

    public static double leerDouble(Scanner scanner, String mensaje) {
        String input;
        while (true) {
            System.out.print(mensaje);
            input = scanner.nextLine();
        }
    }
}
```

```
        try {
            return Double.parseDouble(input);
        } catch (NumberFormatException e) {
            System.out.println("Debe ingresar un número decimal.");
        }
    }
}

public static String leerEmail(Scanner scanner, String mensaje) {
    String input;
    Pattern pattern = Pattern.compile("^[\\w.-]+@[\\w.-]+\\. [a-zA-Z]{2,}$");
    do {
        System.out.print(mensaje);
        input = scanner.nextLine();
    } while (!pattern.matcher(input).matches());
    return input;
}
}
```

Fin.