

Clean Code II

Bloque IV: objetos y estructuras de datos

- Mala práctica:

```
public class MalaPractica {  
    public void procesar(String[] datos) {  
        for (int i = 0; i < datos.length; i++) {    Use enhanced for loop to  
            if (datos[i] != null && !datos[i].isEmpty()) {  
                System.out.println("Procesando: " + datos[i]);  
            }  
        }  
    }  
}
```

- Buena práctica:

```
import java.util.List;  
  
public class BuenaPractica {  
    public void procesar(List<String> datos) {  
        for (String dato : datos) {  
            if (esValido(dato)) {  
                procesarDato(dato);  
            }  
        }  
    }  
  
    private boolean esValido(String dato) {  
        return dato != null && !dato.isEmpty();  
    }  
  
    private void procesarDato(String dato) {  
        // Lógica clara y concisa...  
        System.out.println("Procesando: " + dato);  
    }  
}
```

Bloque V: manejo de errores

- Mala práctica:

```
public class MalaPractica {    The declared package "" does not exist
    public int dividir(int a, int b) {
        try {
            return a / b;
        } catch (Exception e) {
            System.out.println(x:"Ocurrió un error.");
            return 0;
        }
    }
}
```

- Buena práctica:

```
package BloqueV;    Incorrect Package

public class BuenaPractica {
    public int dividir(int a, int b) {
        try {
            return a / b;
        } catch (ArithmeticException e) {
            System.out.println(x:"Error: No se puede dividir por cero.");
            return 0;
        }
    }
}
```

Bloque VI: test unitarios

- Mala práctica:

```
package BloqueVI;      Incorrect Package

public class MalaPractica {
    public int sumar(int a, int b) {
        return a + b;
    }

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        MalaPractica calc = new MalaPractica();
        int resultado = calc.sumar(a:2, b:3);

        if (resultado == 5) {
            System.out.println(x:"Test aprobado");
        } else {
            System.out.println(x:"Test fallido");
        }
    }
}
```

- Buena práctica:

```
package BloqueVI;      Incorrect Package

import static org.junit.jupiter.api.Assertions.assertEquals;   package org.junit.jupi
import org.junit.jupiter.api.Test;   package org.junit.jupiter.api does not exist

class Calculadora {
    public int sumar(int a, int b) {
        return a + b;
    }
}

public class BuenaPractica {
    @Test      cannot find symbol🔗 symbol: class Test🔗 location: class BuenaPractica
    void testSumar() {      testSumar is never used
        Calculadora calc = new Calculadora();
        assertEquals(5, calc.sumar(a:2, b:3), () -> "La suma de 2 y 3 debe ser 5");
    }
}
```

Bloque VII: clases

- Mala práctica:

```
package BloqueVII;    Incorrect Package

public class MalaPractica {
    public void crearUsuario(String nombre, String email) {
        System.out.println("Usuario creado: " + nombre);
        System.out.println("Email enviado a: " + email);
        System.out.println("Actividad registrada: Usuario creado - " + nombre);
    }
}
```

- Buena práctica:

```

package BloqueVII;

public class BuenaPractica {

    public static class ServicioEmail {
        public void enviarEmailBienvenida(String email) {
            System.out.println("Email enviado a: " + email);
        }
    }

    public static class RegistroActividad {
        public void registrarActividad(String usuario, String actividad) {
            System.out.println("Actividad registrada: " + actividad + " - " + usuario);
        }
    }

    public static class Usuario {
        private final String nombre;
        private final String email;

        public Usuario(String nombre, String email) {
            this.nombre = nombre;
            this.email = email;
        }

        public String getNombre() {
            return nombre;
        }

        public String getEmail() {
            return email;
        }
    }

    private final ServicioEmail servicioEmail;
    private final RegistroActividad registroActividad;

```

```

    public BuenaPractica(ServicioEmail servicioEmail, RegistroActividad registroActividad) {
        this.servicioEmail = servicioEmail;
        this.registroActividad = registroActividad;
    }

    public void crearUsuario(Usuario usuario) {
        System.out.println("Usuario creado: " + usuario.getNombre());
        servicioEmail.enviarEmailBienvenida(usuario.getEmail());
        registroActividad.registrarActividad(usuario.getNombre(), actividad:"Usuario creado");
    }

    Run main | Debug main | Run | Debug
    public static void main(String[] args) {
        ServicioEmail servicioEmail = new ServicioEmail();
        RegistroActividad registroActividad = new RegistroActividad();
        BuenaPractica gestorUsuarios = new BuenaPractica(servicioEmail, registroActividad);
        Usuario usuario = new Usuario(nombre:"Juan", email:"juan@example.com");
        gestorUsuarios.crearUsuario(usuario);
    }
}

```