



Daniel Ivan Anaya Alvarez

## Documentación Singleton

En esta actividad decidí hacer un logger que básicamente mantiene un registro cada vez que se quiere generar un log en el programa loggenerador.java se declararon las variables privadas para encapsular y también se hizo un constructor privado estático para que no se pudiera acceder desde afuera y hacer que la variable perteneciera a la clase y evitar que se creen más objetos. La manera en que se puede tener acceso al loggenerador es a través del `getInstance` del constructor que definimos. Dentro del principal es donde se crean los loggers y hacen el método de `getInstance` el cual va hacer que apunten al mismo objeto sin generar otros, al final verificamos con un `print` si los 3 están pasando por el constructor y también si están apuntando al mismo objeto.

### Principal.java

```
package singleton;

public class Principal {
    public static void main(String[] args) {

        // Generador de Log 1
        Loggenerador logger1 = Loggenerador.getInstance();

        // Generador de Log 2
        Loggenerador logger2 = Loggenerador.getInstance();

        // Generador de Log 3
        Loggenerador logger3 = Loggenerador.getInstance();

        // Registrar los mensajes en el metodo log
        logger1.log("Este es el primer mensaje de log.");
        logger2.log("Este es el segundo mensaje de log.");
        logger3.log("Este es el tercer mensaje de log.");

        // Verificar que las tres referencias apuntan al mismo objeto
        if (logger1 == logger2 && logger1 == logger3) {
            System.out.println("logger1, 2 y 3 son la misma instancia.");
        }
    }
}
```

```

    } else {
        System.out.println("logger1, 2 y 3 son diferentes instancias.");
    }
}
}

```

## Loggenerador.java

```

package singleton;

public class Loggenerador {

    // Se encapsula para que no se puedan crear varias instancias del objeto
    private static Loggenerador instanciaUnica;

    // Constructor privado para evitar la creación de múltiples instancias por
    // otros programas
    private Loggenerador() {

    }

    // Método estático para obtener la única instancia de Logger en la clase,
    // solo se accede por getInstance()
    public static Loggenerador getInstance() {
        if (instanciaUnica == null) {
            instanciaUnica = new Loggenerador(); // Crear la instancia si no
            // existe debido al lazy inicialización
        }
        return instanciaUnica; // Devolver la instancia única a los demás si es
        // que ya existe una instanciaUnica
    }

    // String que se va imprimir cuando se utilice el metodo
    public void log(String mensaje) {
        System.out.println("Log: " + mensaje); // Imprimir el mensaje en la
        // consola
    }
}

```

## Foto Del Print

