

Exceptions – Sistemas de Trazas

Programación Orientada a Objetos

San Salvador de Jujuy

UNJu – Facultad de Ingeniería
Ing. José Zapana



Introducción

- A menudo usamos `System.out.println(...)` para escribir en la consola el valor de una variable
- O bien para mostrar un mensaje “pasó por aquí”, como seguimiento de la ejecución del programa.
- Inconvenientes:
 - Si no se controlan pueden causar demora en los procesos
 - Los mensajes enviados NO indican en donde se produjo el error
 - En ambiente de producción deben ser controlados



Qué es el loggin de las aplicaciones

- El "logging" (registro o registro de eventos) en aplicaciones de programación se refiere al proceso de registrar información significativa o eventos relevantes que ocurren durante la ejecución de una aplicación
- Este registro puede realizarse en un archivo o en otro medio de registro.
- Los registros son una herramienta esencial para el desarrollo, el mantenimiento y la operación de aplicaciones de software.



Propósitos del logging

- **Depuración y diagnóstico:** ver datos del programa
- **Auditoría y seguridad:** transacciones bancarias
- **Rendimiento y optimización:** tiempos de respuesta
- **Trazabilidad y Monitorización:** seguir el flujo de ejecución a través de diferentes componentes y microservicios
- **Registro de Actividades:** acciones realizadas por los usuarios en una aplicación
- **Cumplimiento de Normativas:** registros detallados para cumplir con regulaciones y normativas específicas



Logging en Java Spring Boot

- Spring Boot utiliza SLF4J (Simple Logging Facade for Java) como una fachada de logging.
- Logback es el proveedor de logging predeterminado en Spring Boot, pero puede integrarse con otros como Log4j.



Niveles de log

- Cuadro comparativo

		Detail Included in Logs				
		Debug statements	Informational messages	Warning statements	Error statements	Fatal statements
Logging Level	ALL	Included	Included	Included	Included	Included
	DEBUG	Included	Included	Included	Included	Included
	INFO	Excluded	Included	Included	Included	Included
	WARN	Excluded	Excluded	Included	Included	Included
	ERROR	Excluded	Excluded	Excluded	Included	Included
	FATAL	Excluded	Excluded	Excluded	Excluded	Included
	OFF	Excluded	Excluded	Excluded	Excluded	Excluded



application.properties

```
logging.level.root=INFO  
logging.level.com.example=DEBUG  
logging.file.name=logs/spring-boot-app.log  
logging.pattern.console=%d{yyyy-MM-dd HH:mm:ss} - %msg%n
```



¿Qué es @Slf4j ?

- **@Slf4j** es una anotación de Lombok que provee una instancia de SLF4J Logger de manera automática.
- Simplifica el uso de logging al eliminar la necesidad de crear manualmente instancias de Logger.

Ejemplo

```
import lombok.extern.slf4j.Slf4j;
import org.springframework.stereotype.Service;

@Slf4j
@Service
public class UserService {

    public void createUser(String name) {
        log.info("Creando usuario: {}", name);

        if (name == null) {
            log.error("El nombre no puede ser nulo");
        } else {
            log.debug("Usuario creado con éxito: {}", name);
        }
    }
}
```