

Funcionalidad de Logging

El código de la práctica incluye una funcionalidad para recoger logs de la ejecución. Los logs os ayudarán a depurar el código cuando haya errores, especialmente cuando ejecutáis a la herramienta DSLab puesto que no podréis utilizar ninguna debugador ni ver el resultado de los `System.out.println`.

Cuando incluyáis un log, este se verá tanto si el programa se ejecuta en local como si se hace a la herramienta DSLab.

En las ejecuciones en local los logs se guardaran en el fichero que hayáis indicado. También tenéis la opción que los logs se vea por el stdout.

En las ejecuciones a DSLab los logs serán accesibles desde la pantalla donde podéis consultar el resultado de la ejecución.

Añadir una entrada al log

Para añadir una entrada en el log:

```
LSimLogger.log(nivell, missatge);
```

Hay 6 niveles de log

- **FATAL**: Recommended for high critical events that will lead to abort the application instance.
- **ERROR**: An error event that can affect the program but might still allow to continue running the application instance.
- **WARN**: An event that might have an adverse effect.
- **INFO**: Informational messages about the progress of the application instance at coarse-grained level.
- **DEBUG**: Informational messages useful to debug your code (more fine-grained than INFO level.). This level must be deactivated when you finish the debugging process.
- **TRACE**: Informational messages more detailed than the DEBUG level (finest-grained)

Además de estos niveles,

- **OFF**: To turn off the log. This log will not be written.

Algunos ejemplos de uso:

```
LSimLogger.log(Level.INFO, "Inici ");
```

```
LSimLogger.log(Level.INFO, "Total: " + total);
```

```
try{  
    ...  
}catch (Exception e){
```

```
        LSimLogger.log(Level.ERROR, "Exception: " + e.getMessage());  
    }  
}
```

Uso de la funcionalidad de logging en local

En las ejecuciones en local hay que indicar donde se tienen que guardar los logs y si queréis que también se muestren por el stdout. Hay que hacerlo al inicio del programa (antes de hacer el primer llamamiento a LSimLogger; recomendamos hacerlo justo después del main()):

```
LSimLogger.setLoggerAsLocalLogger(logging_id, path_folder_logs,  
_print_logs_in_stdout);
```

- logging_id: per saber quin programa ha generat el log
- path_folder_logs: camí de la carpeta on volem que es desin els logs.
- _print_logs_in_stdout: per indicar si, a més de desar-se en un fitxer, volem que els logs s'escriguin a l'stdout.

Ejemplo de uso:

```
LSimLogger.setLoggerAsLocalLogger("servidor", "./logs/", true);
```

Los logs se guardaran en la carpeta logs y se mostrarán por la stout. El fichero se llamará una cosa del estilo:

```
29-09-2020 08h25m23s179ms_servidor.log
```