

" Interface Interna (Handler de Log) GSE-LLR-162

Deve existir uma interface de logging central (handler de log) que aceite uma message (str).

Lógica (Handler de Log - UI) GSE-LLR-163

A interface de logging central DEVE emitir o sinal logMessage (GSE-LLR-153) com a message recebida.

Lógica (Handler de Log - Arquivo) GSE-LLR-164

A interface de logging central DEVE invocar a interface de escrita do GSELogger com a message recebida.

Lógica (Handler de Log - Proteção) GSE-LLR-165

A interface de logging central DEVE ser encapsulada em um bloco de tratamento de exceções para garantir que uma falha no logging não interrompa a aplicação.

"

Procedimento Realizado

Metodologia de Análise estática de código

Análise no arquivo [gse/backend/controllers/upload_controller.py](#)

Resultados obtidos

Os requisitos estão rastreados e estão implementados através do método privado load_handler da classe UploadController do arquivo referido no procedimento

```
121
122     # =====
123     # REQ: GSE-LLR-162: Interface Interna (Handler de Log)
124     # Descrição: DEVE existir uma interface de logging central (handler de log)
125     # que aceite uma `message` (str).
126     #
127     # ---#
128     # REQ: GSE-LLR-163: Lógica (Handler de Log - UI)
129     # Descrição: A interface de logging central DEVE emitir o sinal `logMessage`
130     # (GSE-LLR-153) com a `message` recebida.
131     #
132     # ---#
133     # REQ: GSE-LLR-164: Lógica (Handler de Log - Arquivo)
134     # Descrição: A interface de logging central DEVE invocar a interface
135     # de escrita do `GseLogger` com a `message` recebida.
136     #
137     # ---#
138     # REQ: GSE-LLR-165: Lógica (Handler de Log - Proteção)
139     # Descrição: A interface de logging central DEVE ser encapsulada em um
140     # bloco de tratamento de exceções para garantir que uma falha no
141     # logging não interrompa a aplicação.
142     #
143     # Autor: Julia
144     # Revisor: Fabrício
145     # =====
146
147     def _log_handler(self, message: str):
148         """
149             Um handler privado que envia logs para DOIS lugares:
150             1. Para a UI (QML) através do sinal logMessage.
151             2. Para o arquivo de log (GseLogger).
152             Implementa: GSE-LLR-162, 163, 164, 165
153         """
154
155         # GSE-LLR-165
156         try:
157             if "[erro]" in message.lower() or "[auth-erro]" in message.lower():
158                 print(message)
159
160             # GSE-LLR-163
161             self.logMessage.emit(message)
162
163             # GSE-LLR-164
164             if self.file_logger:
165                 self.file_logger.write_log(message)
166
167         except Exception as e:
168             print(f"ERRO NO LOG HANDLER: {e}")
```