

# Test GSE-HLR-40

" Interpretar arquivos de resposta LUI e LUS

O software DEVE interpretar corretamente os arquivos binários LUI (Load Unit Information) e LUS (Load Unit Status), extraíndo todos os campos definidos pela norma — incluindo comprimento, versão do protocolo, status ARINC, descrição ASCII e progresso percentual."

## Procedimento realizado

Análise do código

## Resultados obtidos

Esse requisito foi dividido em requisitos de baixo nível, e implementados através da função

```
def parse_lui_response(data: bytes) -> Dict[str, Any]:
    """
    Analisa a resposta de um arquivo LUI (Load Unit Information).
    Formato: [file_length(4)][protocol_version(2)][status_code(2)][desc_length(1)][description]
    """

```

que faz os tratamentos corretos, e retorna o objeto com os seguintes campos

```
    return {
        "file_length": file_length,
        "protocol_version": protocol_version,
        "status_code": f"0x{status_code:04x}",
        "status_name": ARINC_STATUS_MAP.get(status_code, "Desconhecido"),
        "desc_length": desc_length,
        "description": description,
    }

except Exception as e:
    # =====
    # REQ: GSE-LLR-42 - Padronizar erro em exceções de LUI
    # Tipo: Requisito Funcional
    # Descrição: Em qualquer exceção durante o parsing de LUI, a interface DEVE
    #             retornar exatamente {"error": f"Erro ao analisar LUI: {msg}"}.
    # Autor: Julia | Revisor: Fabrício
    # =====
    return {"error": f"Erro ao analisar LUI: {e}"}
```