

Test BC-HLR-5

" Modularidade

O software deve ser organizado em módulos distintos: Wifi, TFTP, gerência de estados, protocolo ARINC615A, armazenamento, autenticação e handler para o botão."

Procedimento realizado

Análise de código

Resultados obtidos

Como podemos observar,

```
▽ components
  > arinc
  > auth
  > button_handler
  > state_machine
  > storage
  > tftp
  > wifi
```

os módulos foram corretamente implementados como **components** do framework **Espressif ESP-IDF**, cada um com responsabilidade bem definida.

O mecanismo de modularização nativo do ESP-IDF permite que cada módulo possua seu próprio arquivo `CMakeLists.txt`, cabeçalhos (`include/`) e código-fonte (`src/`), garantindo **independência, encapsulamento e baixo acoplamento** entre as partes do sistema.

Essa estrutura modular reflete boas práticas de arquitetura de software embarcado, facilitando testes unitários, manutenção futura e integração contínua no projeto B/C.

Destaque para o módulo `state_machine`, que gerencia a lógica de transição entre os estados, através de `fsm.c` e `state_table.c`, enquanto cada um dos estados, contém seu próprio arquivo, para gerenciar o que cada estado tem como responsabilidade de forma independente.

```
▽ SRC
C fsm.c
C state_error.c
C state_init.c
C state_maint_wait.c
C state_operational.c
C state_save.c
C state_table.c
C state_teardown.c
C state_upload_prep.c
C state_uploading.c
C state_verify.c
```