



Modelo de Definição de Pronto (DoD)

NOME DA EQUIPE: Lumon

PARTICIPANTES: Felipe Rubens de Sousa Borges, Marcos Vinícius Tenacol Coêlho, Thiago Thomáz Santana do Nascimento

Esse documento assegura que todos os critérios mínimos de qualidade e funcionalidade foram atendidos antes de uma tarefa ou funcionalidade ser considerada "pronta".

Objetivo: Estabelecer critérios claros e mensuráveis para que uma tarefa ou funcionalidade seja considerada "pronta" no contexto do projeto de IoT, garantindo qualidade, consistência e conformidade com os requisitos.

Crítérios de Pronto Obrigatórios

1. Qualidade do Código:

- Ausência de erros:
 - O código deve ser escrito de forma que não haja falhas óbvias ou erros de sintaxe. Nenhuma falha de compilação ou execução deve ocorrer antes da execução dos testes.
- Documentação do código:
 - O código deve ser comentado de forma clara e explicativa, especialmente em partes que envolvem a lógica de integração com sensores, atuadores e comunicação via Wi-Fi.
 - Cada função/método importante deve ter uma descrição de seu propósito, parâmetros de entrada e saída.

2. Funcionalidade:

- Requisitos atendidos:
 - O código deve estar implementado de acordo com os requisitos descritos nas histórias de usuário e nos requisitos funcionais do projeto (conforme especificado nos RF1, RF2, RF3, RF4, RF5 e RF6).
 - RF1 (Coleta de Dados): Implementação da coleta de dados dos sensores (PIR, Acelerômetro) com carimbo de data e hora.
 - RF2 (Armazenamento de Dados): Implementação de armazenamento dos dados coletados em um banco de dados (local ou nuvem), com formato adequado.
 - RF3 (Controle Remoto dos Atuadores): Implementação do controle dos atuadores (LED, Buzzer) por meio do painel web ou aplicativo móvel.
 - RF4 (Análise de Dados): Implementação da análise dos dados de sono para detectar distúrbios (como apneia ou agitação excessiva).



- RF5 (Notificações de Alertas): Implementação do envio de notificações para os responsáveis sempre que um distúrbio for identificado.
- RF6 (Comunicação de Dados): Implementação da comunicação entre sensores e servidor via Wi-Fi ou Bluetooth para transmissão de dados em tempo real.
- Integração com dispositivos IoT:
 - O código deve estar preparado para ser testado em dispositivos IoT reais, como Arduino, Raspberry Pi e ESP32, com interfaces para comunicação com sensores e atuadores do projeto.
- Integração com sensores e atuadores:
 - O sistema deve integrar corretamente todos os sensores e atuadores, incluindo o PIR, acelerômetro, DHT11, LEDs e buzzer, de acordo com os requisitos do projeto.

3. Aprovação:

- Pull Request/Commit:
 - O código está pronto para ser submetido em um pull request ou commit, com a descrição completa das funcionalidades implementadas, incluindo:
 - Quais requisitos foram atendidos.
 - Quais componentes do sistema foram modificados ou integrados.
 - Quais funcionalidades foram concluídas (como controle dos atuadores, análise de dados, notificações, etc.).
 - As mensagens de commit devem ser claras, indicando o progresso no desenvolvimento das funcionalidades do projeto.