



Definição de Pronto (DoD)

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.

Critérios de Pronto

1. Qualidade do Código:

- Conformidade com os padrões de codificação: O código deve seguir os padrões definidos para o projeto, como boas práticas de escrita, nomenclatura consistente e organização das funções.
- Ausência de erros e warnings: Todo o código deve estar livre de erros e warnings após testes e validações realizadas no Arduino IDE ou outra plataforma relevante.
- Documentação do código: O código principal, funções e trechos críticos devem conter comentários claros, explicando a lógica e a finalidade de cada parte.

2. Funcionalidade:

- Requisitos atendidos: A funcionalidade atende completamente aos requisitos especificados, como a simulação de peso, frequência respiratória, frequência cardíaca e temperatura utilizando potenciômetros.
- Integração com dispositivos IoT: O código está sendo testado para funcionar conforme o esperado nos dispositivos IoT utilizados (ex.: ESP32, display LCD, LEDs e potenciômetros).
- Integração com sensores e atuadores: Todos os potenciômetros, LEDs e o display LCD funcionam corretamente e respondem conforme o esperado.
- Execução de testes automatizados: Os testes foram implementados para validar a funcionalidade e garantir uma cobertura mínima de 80% dos casos.

3. Testes:

- Testes unitários: Todas as funções e componentes individuais foram testados de forma isolada, garantindo o funcionamento básico.
- Testes de integração: O sistema foi testado como um todo, verificando a interação entre os potenciômetros, LEDs, display LCD e o código.
- Testes de hardware: O hardware foi testado em diferentes condições (ex.: variação de valores nos potenciômetros) para garantir robustez e precisão.



- Revisão de QA (Qualidade): A funcionalidade deve ser revisada por um membro da equipe, garantindo conformidade com os critérios estabelecidos.

4. Documentação:

- Documentação técnica: Deve incluir explicações sobre a arquitetura do sistema, fluxos de dados e a configuração dos componentes de hardware (ex.: mapeamento de pinos do ESP32, conexões do LCD e LEDs).
- Manual do usuário (se aplicável): Deve haver um guia para operação do sistema, com instruções para conectar os potenciômetros e interpretar os dados exibidos no LCD.
- Atualização de logs: Alterações no código devem ser registradas no histórico, explicando o motivo das mudanças e quem as realizou.

5. Segurança e Privacidade:

- Verificação de segurança: Embora o projeto seja um simulador, a integridade do sistema deve ser garantida, minimizando riscos de falhas críticas (ex.: valores fora do intervalo esperado).
- Proteção de dados: Dados simulados devem ser manipulados de forma consistente e protegida, mesmo que sejam gerados localmente.
- Autenticação e autorização: Caso o projeto seja estendido para integrar comunicação remota (ex.: via Wi-Fi), recursos de autenticação e autorização devem ser implementados.

6. Aprovação:

- Revisão por pares: O código foi revisado por um membro da equipe para garantir qualidade e aderência aos critérios de pronto.
- Feedback incorporado: Todo o feedback relevante da equipe e stakeholders foi considerado e incorporado.
- Pull Request/Commit: Antes de ser integrado ao repositório principal, a funcionalidade deve ser submetida com uma descrição clara do que foi implementado..