

# PROJETO INTEGRADOR — GEMEO DIGITAL DO CONTROLE DE TANQUES (DTS200)

## I. Objetivo

Criar um gêmeo digital do sistema de controle de tanques, onde os níveis reais sejam lidos por um ESP32, enviados a um banco de dados MySQL (XAMPP) e visualizados em tempo real em um supervisório web. O sistema também deve manter histórico de medições e permitir análise básica de variações de nível.

## II. Divisão da equipe (7 integrantes)

| Grupo                          | Integrantes | Função principal  |
|--------------------------------|-------------|---|
| 1 - Firmware / IoT (ESP32)     | Dupla       | Leitura e envio dos dados do controlador; integração futura com o sistema real.       |
| 2 - Backend + Banco de Dados   | Dupla       | API para receber e validar dados; gravação e consulta no MySQL (XAMPP).               |
| 3 - Frontend / Dashboard Web   | Dupla       | Interface de supervisão com tanques e níveis; histórico e atualizações em tempo real. |
| 4 - QA / Documentação / Gestão | Solo        | Padronização, cronograma, testes, documentação e integração final.                    |

## III. Etapas e responsabilidades de cada grupo

### 1 - Firmware / IoT (ESP32)

- Ler os sinais reais do controlador dos tanques (nível, status de válvulas, etc.) e enviar periodicamente para o banco de dados.
- Testar leitura dos sensores conectados ao ESP (análogicos ou via comunicação serial).
- Estruturar dados em formato JSON.
- Enviar via HTTP POST ou MQTT.
- Implementar verificações de conexão e reenvio.
- Testar fluxo completo com backend e dashboard.

### 2 - Backend + Banco de Dados

- Garantir que os dados recebidos do ESP sejam armazenados corretamente e disponibilizados para o dashboard.
- Criar tabelas no MySQL (tanques, medicos, eventos).
- Desenvolver API de recebimento (PHP/Python/Node.js).
- Criar endpoints de consulta (último valor, histórico, alertas).
- Implementar validação e segurança.
- Testar integração com ESP e Frontend.

### 3 - Frontend / Dashboard Web

- Exibir os tanques, níveis e histórico em uma interface visual responsiva.
- Criar layout dos tanques e tela de supervisão.
- Desenvolver página HTML/CSS/JS com gráficos e indicadores.
- Integrar com API do backend.
- Adicionar alertas visuais e indicadores de status.
- Entregar painel funcional e atualizado em tempo real.

#### 4 - QA / Documentação / Gestão

- Garantir que o projeto siga o cronograma, mantenha padrão e esteja bem documentado.
- Organizar reuniões semanais e acompanhar tarefas.
- Definir estrutura de pastas e convenções.
- Criar README principal e documentação.
- Testar integração entre módulos.
- Preparar relatório e materiais de apresentação.

#### IV. Estrutura padrao do projeto (ASCII)

```
Projeto-Integrador-Tanques/
  +- firmware/
  +- backend/
  +- database/
  +- frontend/
  +- shared_data/
    +- dados_simulados.json
    +- docs/
      +- README.md
      +- test/
```

#### V. Linha do tempo sugerida

| Semana | Foco                    | Entregas principais                                 |
|--------|-------------------------|---|
| 1      | Organização e simulação | Dados simulados funcionando entre todos os módulos. |
| 2      | APIs e banco de dados   | Inserção e consulta funcionando no MySQL.           |
| 3      | Dashboard funcional     | Interface exibe dados e histórico corretamente.     |
| 4      | Integração real         | ESP lendo dados reais do controlador.               |
| 5      | Ajustes e testes finais | Sistema completo, com relatório e apresentação.     |