

**CENTRO UNIVERSITÁRIO UNIFECAF**

**GRADUAÇÃO TECNOLÓGICA EM INTELIGÊNCIA ARTIFICIAL E AUTOMAÇÃO  
DIGITAL**

**ALUNO: MARCOS DETRANO RODRIGUES JUNIOR**

**PROJETO COM BUBBLE: DO BANCO DE DADOS À PUBLICAÇÃO**

São Paulo

2026

## **DO BANCO DE DADOS À PUBLICAÇÃO**

### **Desenvolvimento do OpsTask Manager**

Trabalho prático apresentado à disciplina de Fundamentos de No-Code e Low-Code do curso de Graduação Tecnológica em Inteligência Artificial e Automação Digital do Centro Universitário UniFECAF, como requisito para avaliação.

São Paulo

2026

## SUMÁRIO

1. INTRODUÇÃO
  2. OBJETIVOS
  3. FUNDAMENTAÇÃO TEÓRICA E JUSTIFICATIVA
  4. ARQUITETURA E ESTRUTURA DO SISTEMA
  5. FUNCIONAMENTO E LÓGICA DE NEGÓCIOS
  6. PUBLICAÇÃO E MANUTENÇÃO
  7. CONCLUSÃO
  8. REFERÊNCIAS
- 

### 1. INTRODUÇÃO

No cenário de operações de tecnologia e infraestrutura, a descentralização da comunicação representa um gargalo crítico para a produtividade. Solicitações de serviços, chamados e atualizações de tarefas frequentemente se perdem em e-mails e plataformas de mensageria instantânea, comprometendo a rastreabilidade, os acordos de nível de serviço (SLA) e a qualidade das entregas.

O presente trabalho documenta o desenvolvimento do **OpsTask Manager**, um portal de gerenciamento de tarefas e comunicação interna. O projeto visa resolver esse problema real de gestão, permitindo que times colaborem, criem e atualizem demandas de forma centralizada. A solução foi inteiramente construída utilizando a plataforma Bubble, aplicando os fundamentos de no-code e low-code para garantir a entrega de um Produto Mínimo Viável (MVP) robusto e escalável.

### 2. OBJETIVOS

**Objetivo Geral:** Planejar, construir e publicar uma aplicação web funcional de gestão de tarefas utilizando desenvolvimento no-code.

#### Objetivos Específicos:

- Validar rapidamente uma aplicação funcional sem a necessidade de escrever código tradicional.
- Testar o modelo de negócio com usuários reais em um prazo de 30 dias, mitigando altos custos iniciais de desenvolvimento.
- Estruturar um banco de dados relacional e uma interface de usuário (UI) responsiva com navegação intuitiva.

- Implementar *workflows* para automação de regras de negócio.
- Executar a publicação em ambiente de testes e planejar a manutenção contínua.

### **3. FUNDAMENTAÇÃO TEÓRICA E JUSTIFICATIVA**

A escolha estratégica da plataforma Bubble para o desenvolvimento deste portal fundamenta-se na necessidade de agilidade e qualidade na validação de produtos digitais. O desenvolvimento tradicional de software exige a configuração de servidores, criação de APIs (Back-end) e codificação de interfaces (Front-end), o que inviabilizaria o ciclo rápido de 30 dias.

As plataformas no-code/low-code abstraem a complexidade da infraestrutura, permitindo que o foco seja direcionado inteiramente à modelagem da regra de negócio e à experiência do usuário. O Bubble, especificamente, atua como uma solução *Full-stack*, unindo o banco de dados e a interface visual de forma nativa. Essa abordagem reduz drasticamente o tempo de desenvolvimento e o custo operacional, entregando uma aplicação completa e pronta para uso profissional.

### **4. ARQUITETURA E ESTRUTURA DO SISTEMA**

O sistema baseia-se em uma arquitetura de banco de dados relacional construída no próprio Bubble, desenhada para garantir a integridade da informação:

- **UserProfile (Usuários):** Entidade responsável pelo controle de acesso e perfil. Armazena dados essenciais como Nome, E-mail, Função e Foto de Perfil.
- **Task (Tarefas):** Entidade principal de operação. Registra as demandas por meio dos campos: Título, Descrição, Status (Pendentes, Em Andamento, Concluídas), Prioridade (Baixa, Média, Alta, Urgente) e Data de Vencimento. Possui vínculo direto com a entidade *UserProfile* para definição de responsáveis.
- **Comment (Comentários):** Entidade relacional para comunicação. Armazena a mensagem de texto e possui chaves estrangeiras vinculando o autor (*UserProfile*) e, obrigatoriamente, a tarefa de origem (*Task*).

(Nota: Recomenda-se a inserção de prints do banco de dados relacional nesta seção).

### **5. FUNCIONAMENTO E LÓGICA DE NEGÓCIOS**

A aplicação possui uma interface responsiva dividida em visões estratégicas, como o *Board de Tarefas* (Kanban geral) e as *Minhas Demandas* (visão individual).

O motor da aplicação é regido por *workflows* que automatizam as regras de negócio:

- **Criação e Fluxo de Tarefas:** Ao submeter o formulário de nova tarefa, um *workflow* cadastra o registro na entidade *Task*, definindo os parâmetros iniciais. A alteração de status via interface interativa movimenta o card dinamicamente nas colunas do Kanban, atualizando o banco de dados em tempo real.
- **Comunicação Contextualizada:** A inserção de um novo comentário dispara uma automação que atrela a mensagem exclusivamente à tarefa que está em evidência na tela. Isso constrói um histórico de resolução (troubleshooting) auditável e organizado, eliminando ruídos de comunicação.

(Nota: Recomenda-se a inserção de prints das lógicas de workflow nesta seção ).

## 6. PUBLICAÇÃO E MANUTENÇÃO

Seguindo as melhores práticas de operações, a aplicação utiliza a separação nativa de ambientes fornecida pela plataforma. O ambiente de version-test (Homologação) foi utilizado para o desenvolvimento, validação de usabilidade e testes de fluxos de dados sem impacto ao usuário final.

O *deploy* para o ambiente de produção foi realizado visando a entrega do MVP funcional. O plano de manutenção contínua apoia-se no monitoramento de logs nativos do servidor Bubble para rastreio de anomalias e na utilização de rotinas automáticas de backup (*Point-in-time recovery*) para assegurar a resiliência e a proteção dos dados.

## 7. CONCLUSÃO

O desenvolvimento do **OpsTask Manager** cumpriu integralmente o desafio proposto, entregando um portal de gerenciamento de tarefas funcional, com interface responsiva e banco de dados integrado. A adoção da plataforma no-code Bubble comprovou-se altamente eficaz, permitindo que as complexidades da programação tradicional fossem substituídas pela estruturação lógica de regras de negócio.

O resultado é uma aplicação de alto valor operacional, capaz de otimizar a comunicação das equipes e controlar prazos com agilidade e qualidade. O projeto não apenas validou a viabilidade do MVP em prazo recorde, como também consolidou o aprendizado prático e a autonomia técnica em tecnologias no-code, resultando em um material robusto para portfólio profissional.

## **8. REFERÊNCIAS**

**BUBBLE. Bubble Academy.** Disponível em: <https://bubble.io/academy>. Acesso em: 27 fev. 2026.

**BUBBLE. Bubble Forum - Comunidade e boas práticas.** Disponível em: <https://forum.bubble.io>. Acesso em: 27 fev. 2026.

**BUBBLE. Bubble Manual.** Disponível em: <https://manual.bubble.io>. Acesso em: 27 fev. 2026.

**MAKERPAD. Makerpad - No-Code Learning.** Disponível em: <https://www.makerpad.co>. Acesso em: 27 fev. 2026.

**NOCODE.TECH. Nocode.tech - Ferramentas e exemplos.** Disponível em: <https://nocode.tech>. Acesso em: 27 fev. 2026.

**ROCKETSEAT. Fundamentos de No-Code e Low-Code.** Disponível em: <https://ftr.rocketseat.com.br/jornada/fundamentos-de-no-code-e-low-code/conteudos>. Acesso em: 27 fev. 2026.