

# TechFood- Sistema Gerenciador de Pedidos em Restaurante

---

## Plano de Manutenção Preventiva

## **Garçom**

- Acessa o sistema via celular (navegador web);
- Abre a conta da mesa e anota os pedidos;
- Fecha a conta;

## **Cozinha**

- Recebe os pedidos em um computador.
- Gerencia os pedidos por ordem de chegada.
- Imprime as comandas para o cozinheiro.

## **Caixa**

- Baixa a conta da mesa quando o cliente finaliza o pedido.
- Gera relatórios de faturamento diário e mensal.
- Controla o gasto de insumos com base nos pedidos.

# Stacks

## Front-End

Javascript +  
Vue.js +  
Bootstrap

## Back-End

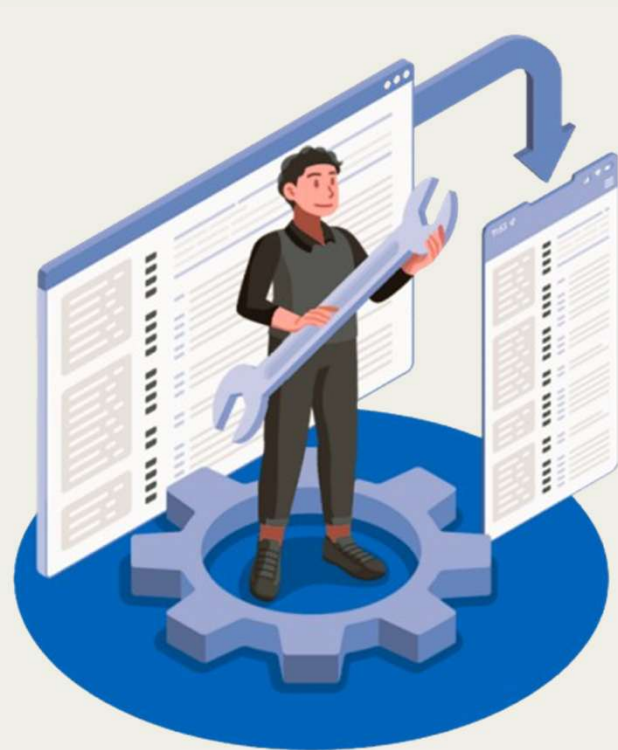
C#/.NET +  
API (Pedidos,  
Cardápio e BD) +  
Postgres

## Possíveis problemas:

- Lentidão,
- Falhas no banco de dados,
- Erros no processamento de pedidos
- Indisponibilidades do sistema,
- Erros no cadastro de informações

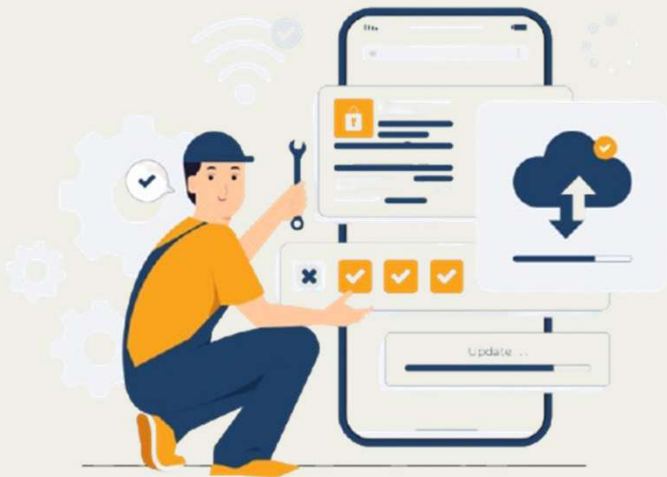
# Plano de Manutenção Preventiva

# Plano de Manutenção Preventiva



- Monitoramento de Desempenho:
  - Prometheus + Grafana para métricas e alertas.
- Atualizações Regulares:
  - Dependabot para atualizações de dependências.
- Testes Automatizados:
  - xUnit/NUnit (backend) + Jest/Cypress (frontend).

# Plano de Manutenção Preventiva



- Backup de dados:
  - Backups diários e incrementais no AWS S3.
- Segurança:
  - Autenticação básica, HTTPS e varreduras com OWASP ZAP.

# Ferramentas utilizadas:

- Monitoramento: Prometheus, Grafana;
- CI/CD: GitHub Actions;
- Backup: AWS S3;
- Testes: xUnit/NUnit, Jest/Cypress;
- Segurança: OWASP ZAP, HTTPS.





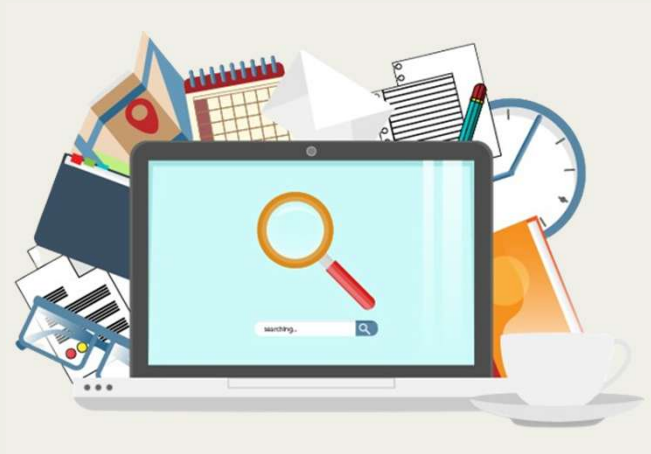
## **Resultados Esperados**

- Sistema estável e sem falhas durante o horário de pico.
- Dados seguros e recuperáveis em caso de problemas.
- Relatórios de faturamento e insumos gerados corretamente.

## **Importância**

- Melhoria na eficiência do negócio.
- Redução de custos com problemas evitáveis.
- Satisfação da equipe e dos clientes.

# A Pesquisa:



- **Objetivos:**
  - Entender como gerenciar mudanças e garantir a estabilidade do sistema.
- **Fontes Utilizadas:**
  - Documentação, tutoriais, artigos da internet.
- **Conceitos estudados:**
  - Controle de versão (Git).
  - Integração contínua e entrega contínua (CI/CD).
  - Monitoramento e alertas.
  - Backup e recuperação de dados.
  - Prevenção de falhas e segurança.

# Conclusão

- Resumo: Sistema estável, seguro e eficiente;
- Próximos passos: Implementar o plano e colher feedback;

FIM