

| | |
|--|------------------------|
| EPIC: Fundação Técnica (SCRUM-16) | |
| 🔗 [SCRUM-27] Setup CI/CD GitLab Criado: 28/set/25 Atualizado(a): 28/set/25 | |
| Status: | A fazer |
| Projeto: | Le Menu |
| Componentes: | Nenhum |
| Versões afetadas: | Nenhum |
| Versões corrigidas: | Nenhum |
| Pai: | EPIC: Fundação Técnica |

| | | | |
|------------------------------------|-------------------|----------------------------------|-------------------|
| Tipo: | História | Prioridade: | Medium |
| Relator: | Jeferson Ferreira | Responsável: | Jeferson Ferreira |
| Resolução: | Não resolvido(s) | Votos: | 0 |
| Categorias: | tipo:story | | |
| Σ Estimativa de trabalho restante: | Desconhecido | Estimativa de trabalho restante: | Desconhecido |
| Σ de Tempo Gasto: | Desconhecido | Tempo gasto: | Desconhecido |
| Σ da Estimativa Original: | Desconhecido | Estimativa original: | Desconhecido |

| Subtarefas: | Chave | Resumo | Tipo | Status | Responsável |
|-----------------------|-------------------------|---------------------------------------|---------|---------|-------------------|
| | SCRUM-44 | T1: Configurar arquivo .gitlab-ci.yml | Subtask | A fazer | Jeferson Ferreira |
| | SCRUM-45 | T2: Setup Docker build stage | Subtask | A fazer | Jeferson Ferreira |
| | SCRUM-46 | T3: Configurar test stage | Subtask | A fazer | Jeferson Ferreira |
| | SCRUM-47 | T4: Setup deploy automatico | Subtask | A fazer | Jeferson Ferreira |
| Sprint: | Sprint 1: Setup inicial | | | | |
| Story point estimate: | 4 | | | | |
| Rank: | 0 j0001f: | | | | |

Descrição

Resumo

Configuração de um pipeline completo de CI/CD no GitLab para automatizar builds, testes, qualidade e deploy do projeto Le Menu.

Contexto

A implementação do pipeline visa automatizar o processo de integração contínua e entrega contínua, garantindo controle de qualidade e deploy automático.

Critérios de Aceitação

- DADO QUE preciso automatizar o processo de deploy
- QUANDO configuro o pipeline no GitLab
- ENTÃO o pipeline executa em menos de 5 minutos
- E a cobertura de testes fica acima de 80%
- E o Quality Gate do SonarQube passa
- E o deploy automático funciona em staging
- E o rollback automático funciona em caso de falha
- E o pipeline multi-stage está configurado

Outras informações

Escopo Técnico

- Pipeline multi-stage (build, test, quality, deploy)
- Ambientes: development, staging, production
- Docker containers para consistência
- Integração com SonarQube para qualidade
- Deploy automático em staging
- Rollback automático em falhas

Definition of Ready (DoR)

- ☐ Descrição clara no formato Como/Quero/Para
- ☐ Critérios de aceitação em Gherkin definidos
- ☐ Pipeline multi-stage detalhado (build, test, quality, deploy)
- ☐ Ambientes definidos (development, staging, production)
- ☐ Docker containers e consistência planejados
- ☐ Integração com SonarQube especificada
- ☐ Métricas de performance definidas (<5 minutos, >80% cobertura)
- ☐ Deploy automático e rollback especificados
- ☐ Acesso ao GitLab e SonarQube configurado
- ☐ Dependências mapeadas (GitLab, Docker, SonarQube)
- ☐ Responsável designado
- ☐ Sem bloqueios conhecidos para refinamento e estimativa

Definition of Done

- Pipeline CI/CD completo funcionando
- Todos os stages executando corretamente
- Cobertura de testes >80% atingida
- Quality Gate SonarQube passando
- Deploy automático em staging validado
- Rollback testado e funcionando
- Documentação de pipeline criada

Gerado em Mon Sep 29 01:01:24 UTC 2025 por Jeferson Ferreira usando JIRA 1001.0.0-SNAPSHOT#100290-
rev:70c7a4e063e24c24167aa537a0ce01c348bf87b2.