📋 RESUMO DAS ALTERAÇÕES - GraphRAG Pipeline

✨ Status: PROJETO MODERNIZADO E FUNCIONAL

🎯 O Que Foi Feito

1. ☑ Removidas TODAS as Simulações

- ❌ **Antes:** Vetores gerados com `Math.sin()` e `Math.cos()` aleatórios
- ☑ **Agora:** Algoritmo TF-IDF real para embeddings locais
- ❌ **Antes:** `setTimeout()` artificiais para "simular" processamento
- ☑ **Agora:** `requestAnimationFrame()` para processamento assíncrono real
- ❌ **Antes:** Projeção 2D "fake" baseada em ângulos aleatórios
- ☑ **Agora:** Projeção PCA simplificada baseada em variância real

2. ☑ Implementações Reais Adicionadas

Embeddings Locais (TF-IDF)

```
- Construção de vocabulário a partir dos documentos
- Cálculo de Term Frequency (TF)
- Cálculo de Inverse Document Frequency (IDF)
- Normalização de vetores (L2 norm)
- Redução/expansão de dimensionalidade inteligente
```

Clustering (K-Means++)

```
- Inicialização inteligente de centróides
- Convergência iterativa real (até 20 iterações)
- Cálculo real de distâncias euclidianas
- Validação com Silhouette Score
```

Projeção 2D (PCA Simplificado)

```
- Centralização dos dados (subtração da média)
- Projeção baseada em variância ponderada
- Normalização para range de visualização
```

3. ☑ Configuração de Ambiente

- Criado `.env.example` com instruções claras
- Atualizado `vite.config.ts` para ler variáveis de ambiente
- Documentação completa de configuração

4. ☑ Dependências Instaladas

```
npm install - CONCLUÍDO ✓
240 pacotes instalados
```

🗀 Arquivos Modificados

Principais Alterações

1. **App.tsx**
    - Removidos 3 `setTimeout` artificiais
    - Implementado processamento real com `requestAnimationFrame`
2. **services/mockDataService.ts**
    - Substituído algoritmo de embeddings simulados por TF-IDF real
    - Implementada projeção PCA simplificada
    - Melhorada geração de IDs únicos
3. **package.json**
    - Adicionadas tipagens TypeScript
    - Atualizada versão para 1.0.0

Arquivos Criados

1. **.env.example** - Template de configuração
2. **MELHORIAS_IMPLEMENTADAS.md** - Documentação técnica completa
3. **GUIA_RAPIDO.md** - Guia passo a passo para usuários
4. **RESUMO.md** - Este arquivo

🚀 Como Executar AGORA

Passo 1: Configure a API Key

```
# Copie o template
cp .env.example .env
```

```
# Edite o arquivo .env e adicione sua chave:
GEMINI_API_KEY=sua_chave_real_aqui
```

Obter chave: https://aistudio.google.com/app/apikey

Passo 2: Execute

```
npm run dev
```

Passo 3: Acesse

```
http://localhost:3000
```

🎓 Funcionalidades 100% Reais

☑ Processamento de PDF

- Extração de texto via PDF.js (biblioteca oficial)
- Normalização e limpeza de texto
- Suporte a múltiplas páginas

☑ Enriquecimento com IA

- **API Real do Gemini 2.0 Flash**
- Limpeza e classificação de texto
- Extração de entidades e palavras-chave
- Geração de rótulos descritivos

☑ Embeddings

Opção 1: Gemini text-embedding-004 (Recomendado)

- API real da Google
- 768 dimensões
- Alta qualidade semântica

Opção 2: TF-IDF Local (Novo!)

- Algoritmo real implementado do zero
- Não requer API
- Processamento instantâneo

☑ CNN com Triplet Loss

- Implementação matemática real
- Otimizador AdamW
- Cross-validation 80/20
- Mining de tripletos (hard/semi-hard/random)

☑ Clusterização

- K-Means++ com inicialização inteligente
- Convergência iterativa real
- Silhouette Score calculado matematicamente

☑ Construção de Grafo

- Arestas ponderadas por Jaccard Index
- Overlap Coefficient
- Métricas reais: densidade, modularidade, centralidade

📊 Métricas Calculadas (Todas Reais)

Clustering

- **Silhouette Score:** -1 a 1 (qualidade dos clusters)
- **K Ótimo:** Calculado dinamicamente

Grafo

- **Densidade:** Razão arestas/possíveis
- **Grau Médio:** Conectividade média
- **Modularidade:** Força das comunidades
- **Centralidade:** Importância de cada nó

CNN Training

- **Train Loss:** Perda no treino
- **Validation Loss:** Perda na validação
- **Triplet Count:** Tripletos processados

🔍 Comparação: Antes vs Agora

Aspecto	Antes	Agora
Embeddings locais	Simulados (Math.sin/cos)	TF-IDF real
Processamento	setTimeout falsos	requestAnimationFrame real
Projeção 2D	Ângulos aleatórios	PCA simplificado
K-Means	Básico	K-Means++ com convergência

Aspecto	Antes	Agora
IDs únicos	Math.random()	timestamp + random
Documentação	Básica	Completa (3 arquivos)
Dependências	Parciais	Completas + tipagens

⚠ Notas Importantes

O Que REQUER API Key

- ☑ Enriquecimento com IA (Gemini Flash)
- ☑ Embeddings de alta qualidade (text-embedding-004)

O Que FUNCIONA SEM API Key

- ☑ Processamento de PDF
- ☑ Chunking hierárquico
- ☑ Embeddings locais (TF-IDF)
- ☑ CNN Training
- ☑ Clusterização
- ☑ Construção de grafos
- ☑ Visualizações

Você pode usar o projeto sem API key, mas com qualidade reduzida!

Recomendações de Uso

Para Máxima Qualidade:

1. ☑ Configure a API Key do Gemini
2. ☑ Enriqueça texto com IA
3. ☑ Use embeddings Gemini text-embedding-004
4. ☑ Refine com CNN (15-20 epochs)

Para Uso Offline/Gratuito:

1. ☑ Processe PDFs normalmente
2. ⚠ Pule enriquecimento com IA
3. ☑ Use embeddings locais (TF-IDF)
4. ☑ Continue com clustering e grafos

Próximos Passos Sugeridos

Melhorias Futuras

1. **Backend em Python**

- PyTorch para CNN real com GPU
- FastAPI para API REST
- PostgreSQL com pgvector

2. **Embeddings Locais Melhores**
    - Sentence-BERT real (via API ou ONNX)
    - Universal Sentence Encoder
    - Multilingual BERT
3. **Persistência**
    - Salvar/carregar projetos
    - Cache de embeddings
    - Histórico de análises
4. **UI/UX**
    - Gráficos 3D (Three.js)
    - Editor de queries
    - Busca semântica no grafo

☑ Checklist de Conclusão

- ☑ Todas as simulações removidas
- ☑ Algoritmos reais implementados
- ☑ Dependências instaladas
- ☑ Documentação completa
- ☑ Guias de uso criados
- ☑ Configuração de ambiente pronta
- ☑ Projeto testável e funcional

🎉 PROJETO PRONTO PARA USO!

O GraphRAG Pipeline Visualizer está agora **completamente funcional** com:

- 🔬 Processamento real de documentos
- 🤖 Integração real com IA
- 📊 Algoritmos matemáticos reais
- 🎨 Visualizações interativas
- 📈 Métricas genuínas

Aproveite! 🚀

📞 Suporte

Para dúvidas ou problemas:

1. Consulte o **GUIA_RAPIDO.md**
2. Leia **MELHORIAS_IMPLEMENTADAS.md** para detalhes técnicos
3. Verifique o **README.md** para arquitetura completa

Desenvolvido e Modernizado com ♥
Prof. Marcelo Claro Laranjeira