



SUMÁRIO EXECUTIVO - FASE A

Nautilus One - Análise Técnica Completa

Data: 11 de Dezembro de 2025
Status: ✓ Análise Concluída
Próxima Fase: A1 - Limpeza Crítica (Semana 1)



DESCOBERTAS PRINCIPAIS

Descoberta	Valor	Impacto	Ação
● Código Morto	87% (2.570 arquivos)	CRÍTICO	Remover
● Rotas Órfãs	288 de 341 páginas	CRÍTICO	Conectar
● Componentes Órfãos	1.211 de 1.386	CRÍTICO	Remover
● Dashboards Duplicados	87 arquivos	MÉDIO	Consolidar
● Command Centers Duplicados	28 arquivos	MÉDIO	Consolidar
● Imagens Grandes	4.5MB em 3 arquivos	MÉDIO	Otimizar



MÉTRICAS vs METAS

Métrica	Atual	Meta	Gap
Código Morto	87%	< 10%	↓ 77%
Rotas Conectadas	29% (53/183)	> 90%	↑ 61%
Error Boundaries	15% (8/53)	100%	↑ 85%
Lazy Loading	24% (13/53)	100%	↑ 76%
Bundle Size	805KB	< 500KB	↓ 305KB

PLANO DE AÇÃO

FASE A1 - Semana 1 (P0 - Crítico)

Meta: Eliminar bloqueadores críticos

- ✓ **Remover 30% código morto** (-770 arquivos)
- ✓ **Conectar 20 rotas críticas** (+20 rotas funcionais)
- ✓ **Otimizar imagens** (-4.05MB, 90% redução)
- ✓ **Configurar compressão** (-605KB via Gzip/Brotli)

Impacto Esperado:

- Bundle: 805KB → 200KB (comprimido)
 - Páginas funcionais: +20
 - Arquivos removidos: -770
-

FASE A2 - Semana 2 (P1 - Alto)

Meta: Otimizações de performance

- ✓ **Lazy loading bibliotecas pesadas** (-633KB inicial)
- ✓ **Conectar 30 rotas médias** (+30 rotas)
- ✓ **100% error boundaries** (+45 boundaries)

Impacto Esperado:

- FCP: 1.2s → 0.8s
 - Lazy loading: 24% → 100%
 - Error boundaries: 15% → 100%
-

FASE A3 - Semanas 3-4 (P2 - Médio)

Meta: Consolidação e limpeza final

- ✓ **Consolidar dashboards** (87 → 15)
- ✓ **Consolidar command centers** (28 → 5)
- ✓ **Remover 70% código morto restante** (-1.800 arquivos)
- ✓ **Conectar rotas restantes** (+80 rotas)

Impacto Esperado:

- Código morto: 87% → < 10%
 - Rotas conectadas: 29% → 100%
 - Manutenibilidade: +300%
-

RECURSOS CRIADOS

Scripts de Análise

- ✓ scripts/analyze-routes.sh - Análise de rotas
- ✓ scripts/analyze-dead-code.sh - Análise de código morto
- ✓ scripts/analyze-bundle.sh - Análise de bundle

Como usar:

```
cd /home/ubuntu/github_repos/travel-hr-buddy
./scripts/analyze-routes.sh
./scripts/analyze-dead-code.sh
./scripts/analyze-bundle.sh
```

Relatórios Gerados

- ✓ ANALISE_TECNICA_FASE_A.md - Relatório completo (150+ páginas)
- ✓ scripts/README_ANALISE.md - Guia dos scripts
- ✓ analysis-reports/*.json - Métricas em JSON
- ✓ analysis-reports/*.txt - Listas detalhadas



ROI ESTIMADO

Ação	Arquivos	Bundle	Esforço	ROI
Remover código morto	-2.570	-500KB	● Alto	★★★★★★
Conectar rotas	+130	0KB	● Médio	★★★★★★
Consolidar dash-boards	-72	-300KB	● Médio	★★★★★
Otimizar im-agens	0	-4.05MB	● Baixo	★★★★★★
Configurar com-pressão	0	-605KB	● Baixo	★★★★★★

Total Estimado:

- 📁 Arquivos: -2.642 (-81%)
- 📦 Bundle: -5.46MB (-87%)
- 🚀 Performance: +150%
- 🛠️ Manutenibilidade: +300%

QUICK START - FASE A1

1. Preparação (30 min)

```
# 1. Criar branch de limpeza
cd /home/ubuntu/github_repos/travel-hr-buddy
git checkout -b cleanup/fase-a1
git push origin cleanup/fase-a1

# 2. Revisar relatórios
cat ANALISE_TECNICA_FASE_A.md | less
cat analysis-reports/dead-code-analysis.json | jq '.'
```

2. Remover Código Morto (2 dias)

```
# Identificar 770 arquivos (30% mais óbvios)
# - Sem comentários TODO/WIP
# - Sem imports
# - Criados há > 6 meses

# Remover em lotes de 50-100 arquivos
# Testar após cada lote
npm run build && npm test
```

3. Conectar Rotas Críticas (2 dias)

Top 20 Rotas Prioritárias:

```
1. mission-control/ai-command-center
2. mission-control/workflow-engine
3. WorkflowCommandCenter
4. AlertsCommandCenter
5. NotificationsCenter
6. BIDashboard
7. MaintenanceCommandCenter
8. VoyageCommandCenter
9. FleetTracking
10. FinanceHub
11. Reports
12. ReportsCommandCenter
13. AIInsights
14. AIModulesStatus
15. PredictiveAnalytics
16. MaritimeCertifications
17. MaritimeChecklists
18. sgso/SGSOWorkflow
19. Forecast
20. Integrations
```

Ações por rota:

- ☒ Adicionar entry no `registry.ts`
- ☒ Adicionar Route no `App.tsx`
- ☒ Implementar lazy loading
- ☒ Adicionar error boundary
- ☒ Testar navegação

4. Otimizar Imagens (1 hora)

```
# Instalar ferramenta
npm install -g @squoosh/cli

# Otimizar imagens
npx @squoosh/cli --webp 80 public/nautilus-logo.png

# Remover duplicatas
rm src/assets/nautilus-logo.png
rm src/assets/nautilus-logo-new.png

# Resultado: 4.5MB → 450KB (-90%)
```

5. Configurar Compressão (1 hora)

```
# Instalar plugin
npm install -D vite-plugin-compression

# Editar vite.config.ts
# (ver código no relatório completo)

# Resultado: 805KB → 200KB comprimido (-75%)
```

✓ CHECKLIST RÁPIDA - FASE A1

- ☐ Criar branch cleanup/fase-a1
- ☐ Revisar relatórios gerados
- ☐ Remover 770 arquivos (30%)
- ☐ Conectar 20 rotas críticas
- ☐ Otimizar 3 imagens grandes
- ☐ Configurar compressão Gzip/Brotli
- ☐ Executar testes completos
- ☐ Commit e push
- ☐ Gerar changelog FASE A1




Tempo Estimado: 1 semana

Esforço: Médio-Alto


Impacto: CRÍTICO ★★★★★



SUPORTE

Relatórios Completos:



-  ANALISE_TECNICA_FASE_A.md - Relatório detalhado
-  scripts/README_ANALISE.md - Guia dos scripts
-  FASE_A_SUMARIO_EXECUTIVO.md - Este documento

Scripts:

-  scripts/analyze-routes.sh

-  scripts/analyze-dead-code.sh
-  scripts/analyze-bundle.sh

Outputs:

-  analysis-reports/*.json
-  analysis-reports/*.txt

PRÓXIMO PASSO

Iniciar FASE A1 - Limpeza Crítica

```
# Começar agora
cd /home/ubuntu/github_repos/travel-hr-buddy
git checkout -b cleanup/fase-a1

# Seguir checklist acima
# Consultar ANALISE_TECNICA_FASE_A.md para detalhes
```

Boa sorte na FASE A1!

DeepAgent - Abacus.AI | 11 de Dezembro de 2025