

Portal Love to Fly - Procedimentos de Desenvolvimento e Status

Gerado: 10 de Janeiro de 2026

Status Projeto: 95% Completo

Lançamento Alvo: 23 Fevereiro 2026

ÍNDICE

1. [PROCEDIMENTOS PARA NOVAS MUDANÇAS](#)
 2. [REGRAS DE INDEPENDÊNCIA DE FUNCIONALIDADES](#)
 3. [ÚLTIMAS 3 IMPLANTAÇÕES](#)
 4. [PRIORIDADES ATUAIS](#)
 5. [PROCEDIMENTOS DE MIGRAÇÃO DE BANCO](#)
 6. [CHECKLIST DE TESTES](#)
-

PROCEDIMENTOS PARA NOVAS MUDANÇAS

Protocolo de 5 Passos Para Toda Mudança de Código

PASSO 1: Reconhecimento e Planejamento

1. Avise sempre sobre ações que serão tomadas
2. Aguarde permissão antes de executar
3. Liste arquivos que serão modificados
4. Estime tempo/complexidade

PASSO 2: Independência de Funcionalidades

CRÍTICO: Todas as funcionalidades devem permanecer independentes

PERMITIDO:

- Importar dados de outra funcionalidade (somente leitura)
- Ler schemas de banco de outras funcionalidades
- Usar componentes UI compartilhados

PROIBIDO:

- Modificar banco de dados de outra funcionalidade
- Alterar código-fonte de outra funcionalidade
- Criar dependências cruzadas entre funcionalidades

PASSO 3: Workflow Git

```
# 1. Verificar branch atual  
git status  
git branch
```

```
# 2. Criar branch feature
```

```

git checkout -b feature/nome-descritivo

# 3. Fazer mudanças incrementais
# ... editar arquivos ...

# 4. Commit com mensagens claras
git add <arquivos>
git commit -m "feat: descrição clara do que foi feito"

# 5. Push e PR
git push origin feature/nome-descritivo
# Criar Pull Request no GitHub

```

PASSO 4: Procedimento de Mudança de Código ✖

1. Identificar arquivos afetados
2. Ler contexto completo (não assumir)
3. Fazer mudanças incrementais (não tudo de uma vez)
4. Testar cada mudança antes de prosseguir
5. Documentar mudanças em .md se necessário
6. Verificar erros com: npm run lint
7. Rodar testes: npm test

PASSO 5: Validação e Testes ✓

ANTES DE COMMIT:

- npm run lint (sem erros)
- npm run build (sucesso)
- npm test (todos passam)
- Testar funcionalidade manualmente
- Verificar não quebrou outras funcionalidades
- Atualizar documentação se necessário

🔒 REGRAS DE INDEPENDÊNCIA DE FUNCIONALIDADES

Funcionalidades Principais (Independentes):

1. **HangarShare** - Marketplace de hangares
2. **Career** - Vagas e currículo
3. **Classifieds** - Anúncios de aeronaves
4. **Profile** - Perfil do usuário
5. **Tools** - Ferramentas de voo (E6B, etc)

Matriz de Interação Permitida:

De → Para	HangarShare	Career	Classifieds	Profile	Tools
HangarShare	<input checked="" type="checkbox"/> Modificar	<input type="checkbox"/> Ler	<input type="checkbox"/> Ler	<input type="checkbox"/> Ler	<input type="checkbox"/> Não
Career	<input type="checkbox"/> Ler	<input checked="" type="checkbox"/> Modificar	<input type="checkbox"/> Ler	<input type="checkbox"/> Ler	<input type="checkbox"/> Não

Classifieds	Ler	Ler	Modificar	Ler	Não
Profile	Ler	Ler	Ler	Modificar	Não
Tools	Não	Não	Não	Ler	Modificar

Legenda:

- = Pode modificar (própria funcionalidade)
 - = Pode ler/importar dados (somente leitura)
 - = Sem interação
-

ÚLTIMAS 3 IMPLANTAÇÕES

Implantação 1: Carreira Fase 2 + ANAC Logbook (10 Janeiro 2026)

Status: 100% COMPLETO E NO AR

Features Implementadas:

- Fase 1: CRUD vagas, aplicações, interface básica
- Fase 2: Perfil ANAC completo (25+ campos)
 - Dados pessoais (nome, CPF, CMA, CHT)
 - Licenças e habilitações
 - Experiência de voo detalhada
 - Formação acadêmica
 - Cursos e certificações
 - Upload documentos (PDF/imagens)
 - Validação ANAC-compliant

Arquivos Documentação:

- CAREER_IMPLEMENTATION_INDEX.md
- CAREER_PHASES_1_2_SUMMARY.md
- CAREER_PHASE2_COMPLETE.md
- CAREER_PHASE2_TESTING_GUIDE.md

Resultado: Sistema pronto para recrutadores de aviação, compliant com requisitos ANAC

Implantação 2: Tratamento de Erros & UI Padronizada (6 Janeiro 2026)

Status: IMPLANTADO NO NETLIFY

Features Implementadas:

- Página 404 customizada (aviação-themed)
- Página 500 customizada (error handling)
- UI consistente em toda plataforma
- Componentes reutilizáveis padronizados

Arquivos Documentação:

- ERROR_HANDLING_COMPLETE.md

Resultado: UX profissional, tratamento de erros robusto

Implantação 3: Classificados Fase 2 (6 Janeiro 2026)

Status: ✓ FASE 2 COMPLETA

Features Implementadas:

- ✓ Listagem anúncios com busca/filtros
- ✓ Detalhes de aeronaves
- ✓ Formulário contato vendedor
- ✓ Sistema de favoritos
- ✓ Dashboard do vendedor

Arquivos Documentação:

- CLASSIFIEDS_PHASE2_COMPLETE.md
- AIRCRAFT_CLASSIFIEDS_PHASE1_COMPLETE.md

Resultado: Marketplace funcional para compra/venda de aeronaves

🎯 PRIORIDADES ATUAIS

Tarefas Críticas (Para Lançamento)

#	Tarefa	Esforço	Status	Prazo
1	Mock Data → Banco Real	5-8h	● Não iniciado	13 Jan
2	Upload de Fotos	4-6h	● Não iniciado	15 Jan
3	Edição de Anúncios	3-4h	● Não iniciado	17 Jan
4	Upload Documentos	4-5h	● Não iniciado	20 Jan
5	Gestão de Reservas	6-8h	● Não iniciado	22 Jan

Total Estimado: 22-31 horas

Detalhes Tarefas Críticas:

1 Mock Data → Banco Real

Problema: APIs HangarShare usam dados hardcoded Solução:

```
// Substituir em:  
// - /api/hangarshare/airport/search  
// - /api/hangarshare/owners  
  
// De mock:  
const mockData = [ ... ]  
  
// Para queries reais:  
const result = await pool.query(  
  'SELECT * FROM airport_icao WHERE icao_code = $1',
```

```
[icao]
);
```

2 Upload de Fotos

Problema: Não há sistema de upload de imagens **Solução:** Implementar AWS S3 ou storage local

```
# Opções:
- AWS S3 (recomendado para produção)
- Vercel Blob Storage
- Local storage (desenvolvimento)
```

3 Edição de Anúncios

Problema: Usuários não podem editar hangares publicados **Solução:** Criar endpoint PATCH /api/hangershare/listings/:id

4 Upload Documentos

Problema: Upload documentos não implementado **Solução:** Integrar com S3, validar PDFs/imagens

5 Gestão de Reservas

Problema: Não há sistema de reservas **Solução:** Criar tabela bookings, API de agendamento

PROCEDIMENTOS DE MIGRAÇÃO DE BANCO

Comando de Migração:

```
# Ver status
npm run migrate

# Aplicar próxima migration
npm run migrate:up

# Reverter última migration
npm run migrate:down

# Criar nova migration
npm run migrate:create nome-descritivo
```

Padrão de Migration:

```
-- migrations/00X_description.sql

-- UP migration
CREATE TABLE IF NOT EXISTS table_name (
    id SERIAL PRIMARY KEY,
    user_id INTEGER REFERENCES users(id),
    created_at TIMESTAMP DEFAULT NOW()
);
```

```
CREATE INDEX IF NOT EXISTS idx_table_user
ON table_name(user_id);

-- DOWN migration (comentado)
-- DROP TABLE IF EXISTS table_name;
```

Regras:

- Sempre usar `IF NOT EXISTS`
- Criar índices para foreign keys
- Migrations sequenciais: 001, 002, 003...
- Atualizar `src/types/db.d.ts` após

✓ CHECKLIST DE TESTES E VALIDAÇÃO

Antes de Cada Commit:

- `npm run lint` # ESLint check
- `npm run build` # TypeScript + build
- `npm test` # Jest tests
- Teste manual da feature
- Verificar `console.log/errors`
- Testar em mobile (responsive)

Antes de Implantação:

- Rodar testes E2E (Playwright)
- Verificar variáveis de ambiente
- Testar build de produção `local`
- Revisar migrations pendentes
- Atualizar documentação
- Tag Git release

Após Implantação:

- Verificar logs Netlify
- Testar site em produção
- Monitorar erros (Sentry se houver)
- Verificar performance
- Atualizar `DEPLOYMENT_COMPLETE.md`

📊 MÉTRICAS DO PROJETO

Infraestrutura:

- **Framework:** Next.js 16.1.1 (App Router)

- **React:** 19.2.3 (mais recente)
- **TypeScript:** Strict mode
- **Banco:** Neon PostgreSQL
- **Migrations:** 29 aplicadas
- **Hosting:** Netlify
- **Testes:** Jest + Playwright

Status Funcionalidades:

Funcionalidade	Status	Completude
Auth & Login	✅ Completo	100%
Career Fase 1 & 2	✅ Completo	100%
Classifieds Fase 2	✅ Completo	100%
HangarShare Core	🟡 Parcial	70%
Profile Config	✅ Completo	100%
Tools (E6B)	✅ Completo	100%
Error Pages	✅ Completo	100%
Payments (Stripe)	🟡 Integrado	80%
Emails (Resend)	✅ Completo	100%

Status Geral: 95% Completo

Estimativa Para Lançamento:

- **Tarefas Restantes:** 5 críticas
- **Tempo Estimado:** 22-31 horas
- **Data Realista:** 23 Fevereiro 2026
- **Data Agressiva:** 9 Fevereiro 2026

🎯 PRÓXIMOS PASSOS IMEDIATOS

1. **Hoje:** Implementar Mock Data → Banco Real (5-8h)
2. **Amanhã:** Implementar Upload de Fotos (4-6h)
3. **Esta Semana:** Completar 3 tarefas críticas restantes

📞 CONTATOS E RECURSOS

Documentação Completa: Ver INVENTARIO_ARQUIVOS_MD_PT.md

Guia Rápido: Ver COMECE_AQUI_GUIA_RAPIDO_PT.md

Líder Projeto: Edson (edsonassumpcao@)

Auto-gerado: 10 Janeiro 2026