

DIVISÃO DE TAREFAS - CareHub

Proposta Integrada de Alocação para 3 Desenvolvedores - 4 Semanas

Data: 4 de Fevereiro de 2026 **Status:** Proposta Híbrida Otimizada **Objetivo:** MVP Production-Ready em 4 Semanas **Abordagem:** Especialização de Papéis + Roadmap Detalhado + Mitigação de Risco





1. ESTRUTURA DE PAPÉIS & ESPECIALIZAÇÃO

Dev 1: Arquiteto Backend & Infraestrutura

Foco Principal: Segurança, Performance de Dados, Logística de API

Tecnologias: - NestJS (framework) - PostgreSQL (database) - Prisma (ORM) - JWT (autenticação) - Docker (containerização)

Responsabilidades: 1. **Design Database** - Modelagem schema PostgreSQL completo 2. **Autenticação & Role Based Access Control - RBAC** - Sistema auth com guards de proteção 3. **API Development** - Todos endpoints CRUD (Cuidadores, Pacientes, Plantões, Pagamentos) 4. **Integrações** - Payment gateway, logging, audit trails 5. **Segurança** - Criptografia, rate limiting, security headers 6. **DevOps Colaboração** - Coordenar com Dev 3 em deployment

Success Metrics: -  Todos 30+ endpoints implementados -  Testes integração 70%+ -  Zero vulnerabilidades críticas -  Response time < 200ms média

Dev 2: Especialista Frontend & UX

Foco Principal: Estética Premium, Componentização, User Experience

Tecnologias: - React 18 (framework) - TypeScript (type safety) - Tailwind CSS (styling) - Shadcn/UI (components) - Zustand (state management) - Recharts (data visualization)

Responsabilidades: 1. **Design System** - Biblioteca de componentes (50+ componentes) 2. **Páginas Core** - Login, Dashboard, Admin, Cuidador, Familiar views 3. **Responsividade** - Compatibilidade Cross-browser. 4. **Performance UI** - Lighthouse score > 80, lazy loading, code splitting 5. **Animações & UX** - Micro-animations, feedback visual, smooth transitions 6. **Testing Componentes** - Unit tests 70%+ coverage

Success Metrics: - ✅ Todas 5+ páginas implementadas - ✅ Lighthouse > 80 - ✅
Responsivo em todos devices - ✅ Component test coverage 70%+

🔧 Dev 3: Full-stack & Lógica de Negócio + DevOps/QA

Foco Principal: Integração Front-to-Back, Regras de Negócio, Infraestrutura & QA

Tecnologias: - React Hook Form (form handling) - Zod (validation) - Axios (HTTP client) - Docker (infraestrutura) - GitHub Actions (CI/CD) - Jest & Vitest (testing)

Responsabilidades: 1. **Lógica de Negócio** - Cálculo horas, plantões, pagamentos 2. **Integração Frontend-Backend** - Conectar APIs com UI 3. **Infraestrutura** - Docker, GitHub Actions, CI/CD pipeline 4. **Testing** - Testes integração, E2E, QA manual 5. **Segurança Operacional** - Security audits, OWASP checks 6. **Deployment** - Staging & production deployment

Success Metrics: - ✅ Todos flows integrados funcionando - ✅ E2E tests críticos passando - ✅ Security audit passed - ✅ Deployment sem incidentes

17 2. ROADMAP 4 SEMANAS - ENTREGAS

SEMANA 1: Fundação & Setup

Tema: Todos rodando, APIs contratadas, Data Base Pronta.

Dev 1 (Backend)

- ☐ Dia 1: PostgreSQL schema design + Docker setup
- ☐ Dia 1: Prisma ORM configurado + migrations
- ☐ Dia 2-3: Auth service completo (login/register/refresh)
- ☐ Dia 3-4: API contracts documentadas (Swagger)
- ☐ Dia 4-5: Error handling framework + logging
- **Entrega:** Database pronto, Auth APIs funcionando

Dev 2 (Frontend)

- ☐ Dia 1: React 18 + Vite + Tailwind + TypeScript setup
- ☐ Dia 1-2: Design System (5 componentes base)
- ☐ Dia 2-3: Login page + Register page (UI completo)
- ☐ Dia 3-4: Protected routing + Auth context
- ☐ Dia 4-5: API client setup + TypeScript types
- **Entrega:** Auth UI funcionando, components base prontos

Dev 3 (DevOps/QA)

- ☐ Dia 1: Docker Compose setup (local dev)
- ☐ Dia 1-2: GitHub repo + branch strategy (Git Flow)
- ☐ Dia 2: GitHub Actions CI setup
- ☐ Dia 2-3: Testing framework setup (Jest + Vitest)
- ☐ Dia 3-4: Development documentation
- ☐ Dia 4-5: Security checklist criado
- **Entrega:** Dev environment pronto, CI rodando

Gate da Semana 1: ☒ - [] Todos devs conseguem rodar docker-compose up - []
Contratos API documentados - [] Auth endpoints testando - [] GitHub Actions passa

SEMANA 2: CRUD Core & Integração

Tema: “Todas operações CRUD, integração começada, RBAC funcionando”

Dev 1 (Backend)

- ☐ Dia 1-2: Users + Cuidadores endpoints (CRUD + approve/reject)
- ☐ Dia 2-3: Pacientes endpoints (CRUD completo)
- ☐ Dia 3-4: Plantões endpoints (CRUD + approval workflow)
- ☐ Dia 4: Cálculo horas lógica implementado
- ☐ Dia 4-5: RBAC guards + testes básicos
- **Entrega:** 20+ endpoints prontos, RBAC funcionando

Dev 2 (Frontend)

- ☐ Dia 1: Dashboard layout (sidebar + top nav)
- ☐ Dia 1-2: Cuidadores list + add/edit/delete modals
- ☐ Dia 2-3: Pacientes list + modals
- ☐ Dia 3-4: Plantões list + modals (com date picker)
- ☐ Dia 4-5: Fix integração com Dev 1, responsividade
- **Entrega:** 4 páginas CRUD implementadas, dados reais fluindo

Dev 3 (DevOps/QA)

- ☐ Dia 1-2: Test templates criados (backend + frontend)
- ☐ Dia 2: Code coverage setup + reporting
- ☐ Dia 2-3: Manual QA testing endpoints
- ☐ Dia 3-4: Integration tests básicos criados
- ☐ Dia 4-5: Bug tracking, defects documentados
- **Entrega:** Testing framework pronto, 1ª rodada QA completa

Gate da Semana 2: ☒ - [] Todos endpoints CRUD funcionando - [] Frontend integrado com backend - [] RBAC validado - [] Sem bugs bloqueadores

SEMANA 3: Dashboard, Reports & Hardening

Tema: “Funcionalidades avançadas, segurança aplicada, staging pronto”

Dev 1 (Backend)

- ☐ Dia 1: Pagamentos service + endpoints
- ☐ Dia 1-2: Reports endpoints (monthly, per-person)
- ☐ Dia 2-3: Dashboard metrics endpoints
- ☐ Dia 3: CSV export implementado
- ☐ Dia 3-4: Field-level encryption (CPF, bank data)
- ☐ Dia 4-5: Security headers + rate limiting
- **Entrega:** APIs avançadas completas, dados sensíveis criptografados

Dev 2 (Frontend)

- ☐ Dia 1: Dashboard com métricas reais
- ☐ Dia 1-2: Charts/visualizações (Recharts)
- ☐ Dia 2-3: Pagamentos/Reports página
- ☐ Dia 3: Export CSV functionality
- ☐ Dia 3-4: Mobile responsiveness audit + fixes
- ☐ Dia 4-5: Polish UI, animations refinement
- **Entrega:** Dashboard funcional, reports working, mobile-ready

Dev 3 (DevOps/QA)

- ☐ Dia 1-2: Security audit checklist execução
- ☐ Dia 2: OWASP scanning + vulnerability fixes
- ☐ Dia 2-3: Staging environment setup (production-ready)
- ☐ Dia 3: Monitoring setup (logs + metrics)
- ☐ Dia 3-4: Load testing basic scenario
- ☐ Dia 4-5: Deployment runbook criado
- **Entrega:** Staging pronto, security audit passed, monitoring live

Gate da Semana 3: ☒ - [] Dashboard + Reports funcionando - [] Security audit passed - [] Staging environment pronto - [] Performance aceitável

SEMANA 4: Testing, Polish & Production Launch

Tema: “Tudo testado, seguro, launched em produção”

Dev 1 (Backend)

- ☐ Dia 1-2: Integração testing completa (70%+)
- ☐ Dia 2: Performance optimization
- ☐ Dia 2-3: Bug fixes críticos
- ☐ Dia 3-4: Final security review
- ☐ Dia 4-5: Production deployment com Dev 3
- **Entrega:** Backend pronto para produção

Dev 2 (Frontend)

- ☐ Dia 1: Fix QA bugs relatados
- ☐ Dia 1-2: Component testing (React Testing Library)
- ☐ Dia 2: Lighthouse optimization final
- ☐ Dia 2-3: E2E critical flows testing
- ☐ Dia 3-4: Browser/device compatibility verify
- ☐ Dia 4-5: Production deployment
- **Entrega:** Frontend pronto para produção

Dev 3 (DevOps/QA)

- ☐ Dia 1-2: Full QA testing (todas features)
- ☐ Dia 2-3: Performance testing final
- ☐ Dia 3: Production environment final setup
- ☐ Dia 4: Production deployment execution
- ☐ Dia 4-5: Post-launch monitoring e fixes
- **Entrega:** Aplicação live em produção, monitorada

Gate da Semana 4: ☒ - [] Production deployment successful - [] Zero critical vulnerabilities - [] Monitoring operational - [] Team trained on support

3. PROCESSOS & WORKFLOW

Daily Standup (9:00 AM - 15 min máx)

Formato:

Dev 1: Qual API shipei? Blockers?
Dev 2: Qual UI shipei? Blockers?
Dev 3: Qual infraestrutura? Blockers?







Se alguém bloqueado → Huddle de 15 min imediatamente

Git Flow Strategy

```
main
  ↑ (merge com tag release)
  |
staging
  ↑ (merge após QA pass)
  |
dev (integração branch)
  ↑ (merge de features após PR review)
  |
feature/auth-system (individual branches)
feature/crud-cuidadores
feature/dashboard
...
```

Regras: 1. Nada entra em dev sem PR 2. Mínimo 1 code review (outro dev) 3. CI/CD deve passar antes de merge 4. Commit messages claros (feat: login API, fix: dashboard bug)

Definition of Done

Uma task é “done” quando: -  Código escrito -  Código testado (unit tests min 70%) -  Code review aprovado -  CI/CD pipeline passou -  Integrado com outras partes -  Documentado

PR Review Checklist





- ☐ Código segue padrões
 - ☐ Tests inclusos + passando
 - ☐ Sem console.logs/debug code
 - ☐ Performance considerado
 - ☐ Security implications revisado
 - ☐ Documentação atualizada
-

4. GESTÃO DE RISCO & CONTINGÊNCIA

Riscos Identificados

Risco	Probabilidade	Impacto	Mitigação	Trigger
API Contract churn	30%	ALTO	Dev 1 publica contratos EOD W1	Se mudanças requisitadas > 3x
DB Performance	20%	ALTO	Indexing strategy D1 W1	If queries > 1s
Payment integration	10%	MÉDIO	Skip em MVP, W4 se tempo	If progress < 80% W3
Security audit failure	5%	CRÍTICO	Daily security checklist	If vuln crítica encontrada
Deploy issues	10%	MÉDIO	Test staging W3	If deploy W4 > 30 min

Plano B: Redução de Escopo (se atraso)

Se Semana 3 completion < 80%, cut: 1.  Payment processing (mostrar só números) 2.  Document uploads (UI apenas) 3.  Mobile optimization (desktop-first) 4.  Keep: CRUD + Auth + Dashboard + Reports

5. MÉTRICAS DE SUCESSO

Por Semana

Semana	Métrica	Target	Real
1	Endpoints Auth	3	?
1	Auth UI Pages	2	?
1	CI/CD Passing	100%	?
2	CRUD Endpoints	20	?
2	CRUD Pages	4	?
2	Integration Tests	10	?
3	Advanced Endpoints	10	?
3	Dashboard Metrics	100%	?
3	Security Audit	Pass	?

Semana	Métrica	Target	Real
4	Staging Deployment	Success	?
4	Production Deployment	Success	?
4	Monitoring Live	Yes	?

Qualidade Geral

- **Test Coverage:** Target 70%+
- **Lighthouse Score:** Target > 80
- **Response Time:** Target < 200ms média
- **Security Vulnerabilities:** Target 0 críticas
- **Uptime:** Target 99.5%

6. COMUNICAÇÃO & ESCALAÇÃO

Canais de Comunicação

- **Daily Sync:** Slack #carehub-dev (async updates)
- **Standup:** Discord/Teams (9:00 AM live)
- **PRs:** GitHub (code reviews)
- **Decisions:** Wiki/Confluence
- **Issues:** Jira/GitHub Issues

Escalação Path

Dev → Standup mention → Tech Lead → Project Manager
 ↓
 30 min huddle
 ↓
 Resolved or escalated up

Decision Making

- **Technical:** Dev Lead (Dev 1 for backend, Dev 2 for frontend)
- **Process:** Majority vote (2/3 devs)
- **Scope:** Project Manager + Tech Lead

7. ENTREGÁVEIS FINAIS (SEMANA 4)

Backend Deliverables

- ☐ NestJS aplicação completa
- ☐ PostgreSQL database com seed data
- ☐ 30+ endpoints funcionando

- ☐ Authentication/RBAC system
- ☐ Business logic (horas, pagamentos)
- ☐ Testes integração (70%+)
- ☐ API documentation (Swagger)
- ☐ Deployment guide


Frontend Deliverables


- ☐ React aplicação completa
- ☐ 50+ components
- ☐ 5+ páginas completas
- ☐ Responsivo (mobile/tablet/desktop)
- ☐ Component tests (70%+)
- ☐ Lighthouse > 80
- ☐ Accessibility WCAG 2.1 AA
- ☐ User documentation


DevOps/QA Deliverables

- ☐ Docker setup (local + prod)
- ☐ GitHub Actions CI/CD
- ☐ Testing framework + test suite
- ☐ Production environment
- ☐ Monitoring & alerting
- ☐ Security audit report
- ☐ Deployment runbook
- ☐ Support documentation

8. SUCCESS CRITERIA - Checklist Final

Backend Ready:  - ☐ Todos endpoints implementados - ☐ Autenticação funcionando - ☐ RBAC validado - ☐ Tests 70%+ - ☐ Security audit passed - ☐ Deployment tested

Frontend Ready:  - ☐ Todas páginas completas - ☐ Responsivo verificado - ☐ Performance OK (Lighthouse > 80) - ☐ Tests 70%+ - ☐ Integração backend OK - ☐ Deployment tested

DevOps/QA Ready:  - ☐ Infraestrutura pronta - ☐ CI/CD passando - ☐ Staging testado - ☐ Produção ready - ☐ Monitoring live - ☐ Team trained

QUICK START CHECKLIST (Dia 1)

- ☐ Reunião kickoff (15 min cada dev)
- ☐ Compartilhar este documento
- ☐ Setup Slack/GitHub/Jira
- ☐ Primeiro Docker compose up
- ☐ Primeira standup (confirmação)
- ☐ Configurar 1ª semana tasks

Contatos & Responsabilidades

Item	Responsável	Backup
Backend Architecture	Dev 1	Tech Lead
Frontend Architecture	Dev 2	Tech Lead
DevOps/Infrastructure	Dev 3	Tech Lead
Project Management	PM	Tech Lead
Client Communication	PM	Tech Lead

POST-MVP: Phase 2 (Semanas 5-8)

Imediatamente após launch: 1. **Payment Integration** (Stripe/Mercado Pago) - W5-6 2. **Document Upload** (S3 integration) - W6-7 3. **Mobile Optimization** (React Native) - W7-8 4. **SMS/Email Notifications** - W7-8

Versão: 1.0 (Proposta Híbrida) **Status:** Ready for Approval **Timeline:** 4 Semanas **Objetivo:** MVP Production-Ready + Feliz Equipe 🎉