ARQUITETURA_ERP_UNIFICADO_COMPLETA

🏗️ ARQUITETURA ERP UNIFICADO UZZAI v3.0 — O SANTO GRAAL

> **DOCUMENTO MASTER DE ARQUITETURA**
>
> Este documento consolida a arquitetura completa do **ERP Unificado UzzAI**, que combina:
>
> - 🏢 **Gestão Interna** (Projetos, Reuniões, Atas, Decisões, Ações, Sprints, RAG)
> - 🛒 **ERP Comercial** (Vendas, Estoque, Produtos, Clientes, Fornecedores)
> - 📊 **Financeiro/Fiscal** (Fluxo de Caixa, DRE, Impostos, Notas Fiscais)
> - 🤖 **IA Multi-Agente** (Extração automática, RAG, Automações)
>
> **Visão:** "Um sistema único que gerencia toda a empresa — da reunião à nota fiscal."

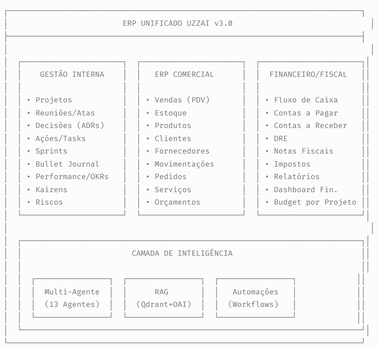
📑 ÍNDICE

---

1. VISÃO GERAL

1.1 O Que É Este Sistema

O **ERP Unificado UzzAI** é um sistema completo de gestão empresarial que integra:



1.2 Problema Resolvido

Cenário	ANTES (Fragmentado)	DEPOIS (Unificado)
Reunião → Ata	4-6 horas manuais	5 min automáticos
Venda → Estoque	Planilhas separadas	Atualização automática
Projeto → Budget	Desconectados	Integração total
Cliente → Histórico	Sistemas separados	Visão 360°
Relatório Financeiro	Export manual	Dashboard real-time
Decisões duplicadas	Frequentes	RAG detecta 100%

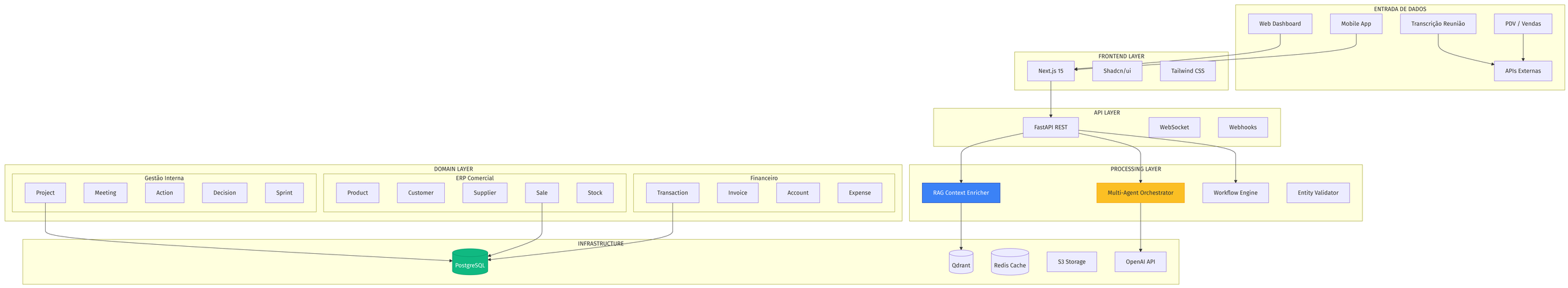
1.3 Princípios Arquiteturais

Princípio	Descrição	Implementação
Clean Architecture	Separação de camadas	Domain → Application → Infrastructure
Domain-Driven Design	Modelagem por domínio	Bounded Contexts por módulo
Event-Driven	Comunicação desacoplada	Domain Events + Webhooks
Multi-Tenancy	Isolamento por organização	`organization_id` global
RAG-First	Contexto histórico	Qdrant + OpenAI Embeddings
Mobile-First	Responsividade	Next.js + PWA

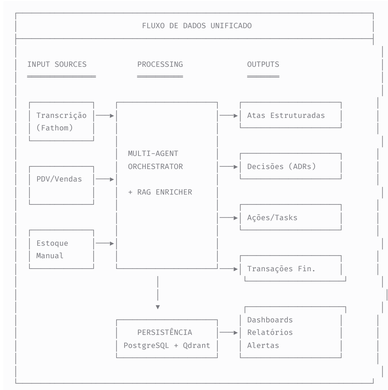
---

2. ARQUITETURA DE ALTO NÍVEL

2.1 Diagrama de Componentes



2.2 Fluxo de Dados Principal



---

3. MÓDULOS DO SISTEMA

3.1 Mapa de Módulos

```
ERP-UNIFICADO-UZZAI/
│
├── 🏢 GESTÃO INTERNA (Core UzzAI)
│   ├── Projetos
│   │   ├── Dashboard por Projeto
│   │   ├── Sprints Semanais
│   │   ├── Roadmap Visual (Gantt)
│   │   └── Métricas de Velocity
│   │
│   ├── Reuniões
│   │   ├── Atas Automáticas
│   │   ├── Extração Multi-Agente
│   │   └── Efetividade Score
│   │
│   ├── Decisões (ADRs)
│   │   ├── Catálogo de Decisões
│   │   ├── Anti-Duplicação RAG
│   │   └── Impacto Tracking
│   │
│   ├── Ações/Tasks
│   │   ├── Kanban Board
│   │   ├── Atribuição Automática
│   │   └── Deadlines Inteligentes
│   │
│   ├── Bullet Journal
│   │   ├── Daily Logs
│   │   ├── Weekly Reviews
│   │   └── Monthly Reports
│   │
│   ├── Pessoas
│   │   ├── Perfis Unificados
│   │   ├── Alocação de Carga
│   │   └── Performance Score
│   │
│   ├── Performance/OKRs
│   │   ├── Avaliação 360°
│   │   ├── KPIs por Pessoa
│   │   └── OKRs Trimestrais
│   │
│   └── Base de Conhecimento
│       ├── Frameworks
│       ├── Garimpos/Insights
│       └── Metodologias
│
├── 🛒 ERP COMERCIAL (Core MeguisPet)
│   ├── Cadastros
│   │   ├── Pessoas (Cliente/Fornecedor/Ambos) ← UNIFICADO
│   │   ├── Produtos
│   │   ├── Categorias
│   │   └── Serviços
│   │
│   ├── Vendas
│   │   ├── PDV (Ponto de Venda)
│   │   ├── Orçamentos
│   │   ├── Pedidos
│   │   └── Histórico de Vendas
│   │
│   ├── Estoque
│   │   ├── Movimentações (Entrada/Saída/Ajuste)
│   │   ├── Inventário
│   │   ├── Preço Médio Ponderado
```

```
│   │   └── Alertas de Mínimo
│   │
│   └── Compras
│       ├── Pedidos de Compra
│       ├── Cotações
│       └── Recebimento
│
├── 🔥 FINANCEIRO/FISCAL
│   ├── Contas a Pagar
│   │   ├── Agendamento
│   │   ├── Parcelamentos
│   │   └── Baixa Automática
│   │
│   ├── Contas a Receber
│   │   ├── Faturamento
│   │   ├── Cobrança
│   │   └── Inadimplência
│   │
│   ├── Fluxo de Caixa
│   │   ├── Previsão
│   │   ├── Realizado
│   │   └── Runway
│   │
│   ├── DRE (Demonstrativo)
│   │   ├── Por Período
│   │   ├── Por Projeto
│   │   └── Por Centro de Custo
│   │
│   ├── Notas Fiscais
│   │   ├── NFe (Produtos)
│   │   ├── NFSe (Serviços)
│   │   └── Cancelamentos
│   │
│   └── Budget por Projeto
│       ├── Planejado vs Realizado
│       ├── Alertas de Estouro
│       └── Integração Reunião→Despesa
│
└── 🧠 INTELIGÊNCIA ARTIFICIAL
    ├── Multi-Agent System
    │   ├── Extraction Agents (5)
    │   ├── Enrichment Agents (6)
    │   └── Validator Agent (1)
    │
    ├── RAG System
    │   ├── Decisions Index
    │   ├── Actions Index
    │   └── Kaizens Index
    │
    └── Automações
        ├── Workflows Customizáveis
        ├── Triggers Inteligentes
        └── Notificações
```

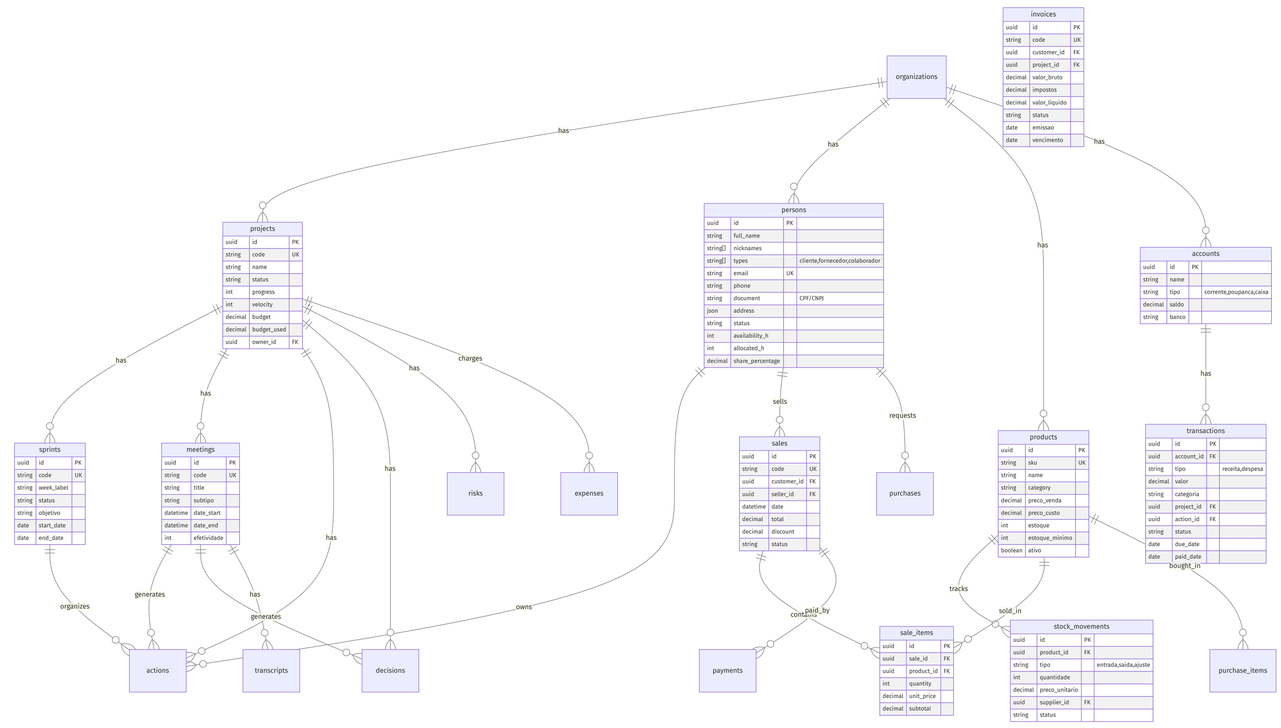
3.2 Matriz de Integração entre Módulos

Origem	Destino	Integração	Exemplo
Reunião	**Decisão**	Auto-extração	"D-2025-042: Usar Capacitor"
Reunião	**Ação**	Auto-extração	"A-2025-123: Implementar login"
Reunião	**Despesa**	Classificação	"Contratar designer R$2.500"
Venda	**Estoque**	Baixa automática	Produto vendido → Estoque-1
Venda	**Contas Receber**	Faturamento	Venda a prazo → Parcelas
Compra	**Estoque**	Entrada automática	Compra recebida → Estoque+N
Compra	**Contas Pagar**	Agendamento	Compra → Parcelas a pagar
Projeto	**Budget**	Monitoramento	Despesas vs Planejado
Cliente	**Vendas**	Histórico 360°	Todas compras do cliente
Produto	**Estoque**	Status real-time	Quantidade disponível

---

4. MODELO DE DOMÍNIO UNIFICADO

4.1 Entidades Principais (Diagrama ER)



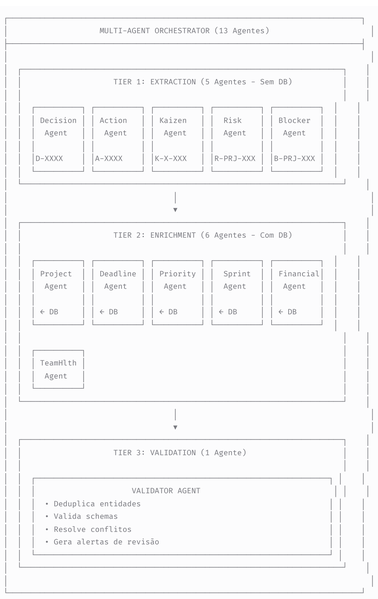
4.2 Schema de IDs Unificados

Entidade	Formato	Exemplo
Decisão	`D-{YYYY}-{seq}`	`D-2025-042`
Ação	`A-{YYYY}-{seq}`	`A-2025-123`
Kaizen	`K-{tipo[0]}-{seq}`	`K-T-015`
Risco	`R-{projeto}-{seq}`	`R-CHATBOT-003`
Meeting	`MTG-{YYYY-MM-DD}-{projeto}`	`MTG-2025-11-24-CHATBOT`
Sprint	`Sprint-{YYYY}-W{nn}`	`Sprint-2025-W48`
Venda	`VND-{YYYY}-{seq}`	`VND-2025-00456`
NF	`NF-{YYYY}-{seq}`	`NF-2025-00123`
Produto	`SKU-{categoria}-{seq}`	`SKU-ELET-001`

---

5. ARQUITETURA MULTI-AGENTE

5.1 Visão Geral



5.2 Financial Agent (NOVO)

```python
class FinancialAgent(BaseSpecializedAgent):
    """Agente que classifica ações com impacto financeiro."""

    FINANCIAL_VERBS = [
        "contratar", "comprar", "pagar", "assinar", "adquirir",
        "investir", "gastar", "desembolsar", "fechar plano"
    ]

    CATEGORY_MAPPING = {
        "ferramenta": "ferramentas",
        "design": "design",
        "dev": "desenvolvimento",
        "marketing": "marketing",
        "infra": "infraestrutura"
    }

    async def classify(self, action: ActionDTO) -> Optional[ExpenseDraft]:
        """Classifica se ação gera despesa."""
        text = action.description.lower()

        # Detecta verbos financeiros
        is_financial = any(v in text for v in self.FINANCIAL_VERBS)

        # Extrai valor se mencionado
        value_match = re.search(r'R\$\s*([\d.,]+)', action.description)
        value = parse_currency(value_match.group(1)) if value_match else None

        if is_financial:
            return ExpenseDraft(
                description=action.description,
                project_code=action.project,
                action_id=action.id,
                gross_total=value,
```

```
                category=self.infer_category(text),
                status="draft"  # Requer aprovação
            )
        return None
```

---

6. SISTEMA RAG

6.1 Coleções Qdrant

```
COLLECTIONS = {
    "decisions": {
        "vector_size": 3072,  # text-embedding-3-large
        "payload": ["decision_id", "project_code", "title", "status"]
    },
    "actions": {
        "vector_size": 3072,
        "payload": ["action_id", "project_code", "description", "success_score"]
    },
    "kaizens": {
        "vector_size": 3072,
        "payload": ["kaizen_id", "lesson", "category", "roi_impact"]
    },
    "products": {
        "vector_size": 3072,
        "payload": ["sku", "name", "category", "description"]
    },
    "customers": {
        "vector_size": 3072,
        "payload": ["person_id", "name", "preferences", "purchase_history"]
    }
}
```

6.2 Fluxo Anti-Duplicação

```
Nova Decisão → Embed → Buscar Similares → Threshold > 0.75?
                                    │
                    ┌───────────────┼───────────────┐
                    │               │               │
                    ▼               ▼               ▼
               SIM (>0.75)      TALVEZ (0.5-0.75)   NÃO (<0.5)
                    │               │               │
                    ▼               ▼               ▼
              Marca requires_review  Sugere similar   Indexa nova
              Alerta para usuário    para verificar   decisão
```

---

7. BANCO DE DADOS

7.1 Índices Críticos

```
-- Performance para Gestão Interna
CREATE INDEX idx_actions_project_status ON actions(project_id, status);
CREATE INDEX idx_actions_owner_due ON actions(owner_id, due_date);
CREATE INDEX idx_meetings_project_date ON meetings(project_id, date_start DESC);
CREATE INDEX idx_decisions_project ON decisions(project_id);

-- Performance para ERP Comercial
CREATE INDEX idx_products_sku ON products(sku);
CREATE INDEX idx_products_category ON products(category);
CREATE INDEX idx_sales_customer ON sales(customer_id);
CREATE INDEX idx_sales_date ON sales(date DESC);
CREATE INDEX idx_stock_product ON stock_movements(product_id);

-- Performance para Financeiro
CREATE INDEX idx_transactions_account_date ON transactions(account_id, due_date);
CREATE INDEX idx_transactions_project ON transactions(project_id);
CREATE INDEX idx_invoices_customer ON invoices(customer_id);

-- Full Text Search
CREATE INDEX idx_products_name_fts ON products USING gin(to_tsvector('portuguese', name));
CREATE INDEX idx_actions_desc_fts ON actions USING gin(to_tsvector('portuguese', description));
```

7.2 Views Materializadas

```
-- View: Estoque com Valores
CREATE MATERIALIZED VIEW estoque_com_valores AS
SELECT
    p.id,
    p.sku,
    p.name,
    p.estoque,
    p.preco_custo,
    p.preco_venda,
    (p.estoque * p.preco_custo) as valor_total_custo,
    (p.estoque * p.preco_venda) as valor_total_venda,
    (p.preco_venda - p.preco_custo) as margem_unitaria,
    CASE
        WHEN p.estoque <= 0 THEN 'sem_estoque'
        WHEN p.estoque <= p.estoque_minimo THEN 'estoque_baixo'
        ELSE 'normal'
    END as status_estoque
FROM products p
WHERE p.ativo = true;

-- View: Métricas de Projeto
CREATE MATERIALIZED VIEW project_metrics AS
SELECT
    p.id AS project_id,
    p.code,
    COUNT(DISTINCT a.id) FILTER (WHERE a.status = 'open') AS open_actions,
    COUNT(DISTINCT a.id) FILTER (WHERE a.status = 'done') AS done_actions,
    COALESCE(SUM(e.valor), 0) AS budget_used,
    MAX(r.severity) AS max_risk_severity
FROM projects p
LEFT JOIN actions a ON a.project_id = p.id
LEFT JOIN transactions e ON e.project_id = p.id AND e.tipo = 'despesa'
LEFT JOIN risks r ON r.project_id = p.id AND r.status = 'open'
GROUP BY p.id, p.code;
```

---

8. API REST UNIFICADA

8.1 Estrutura de Endpoints

```
/api/v1/
│
├── /auth/
│   ├── POST   /login
│   ├── POST   /logout
│   └── POST   /refresh
│
├── /organizations/
│   ├── GET    /
│   └── GET    /{id}
│
├── /projects/
│   ├── GET    /
│   ├── POST   /
│   ├── GET    /{id}
│   ├── PUT    /{id}
│   └── GET    /{id}/dashboard
│
├── /meetings/
│   ├── GET    /
│   ├── POST   /ingest          ← PROCESSA TRANSCRIÇÃO
│   ├── GET    /{id}
│   └── GET    /{id}/entities
│
├── /actions/
│   ├── GET    /
│   ├── POST   /
│   ├── PUT    /{id}
│   ├── POST   /{id}/complete
│   └── GET    /kanban
│
├── /decisions/
│   ├── GET    /
│   ├── GET    /{id}
│   └── GET    /similar         ← RAG SEARCH
│
├── /sprints/
│   ├── GET    /
│   ├── GET    /{id}
│   └── POST   /{id}/close
│
├── /products/
│   ├── GET    /
│   ├── POST   /
│   ├── PUT    /{id}
│   ├── DELETE /{id}
│   └── GET    /{id}/stock
│
├── /sales/
│   ├── GET    /
│   ├── POST   /                ← CRIA VENDA + BAIXA ESTOQUE
│   ├── GET    /{id}
│   └── POST   /{id}/cancel
│
├── /stock/
│   ├── GET    /movements
│   ├── POST   /entry           ← ENTRADA ESTOQUE
│   ├── POST   /exit            ← SAÍDA ESTOQUE
│   └── POST   /adjust          ← AJUSTE/INVENTÁRIO
│
├── /persons/
│   ├── GET    /
│   ├── POST   /
│   ├── PUT    /{id}
│   ├── GET    /{id}/360        ← VISÃO 360° (cliente+colaborador)
│   ├── GET    /customers
│   ├── GET    /suppliers
│   └── GET    /team
│
├── /financial/
│   ├── GET    /accounts
│   ├── GET    /transactions
│   ├── POST   /transactions
│   ├── GET    /cashflow
│   ├── GET    /dre
│   └── GET    /budget/{project_id}
│
├── /invoices/
│   ├── GET    /
│   ├── POST   /
│   ├── GET    /{id}
│   └── POST   /{id}/emit       ← EMITE NF
│
└── /reports/
    ├── GET    /sales
    ├── GET    /stock
    ├── GET    /financial
    └── GET    /performance
```

8.2 Endpoint Principal: Ingest Meeting

```
// POST /api/v1/meetings/ingest
{
  "transcript": {
    "raw_text": "Reunião de alinhamento ... ",
    "source": "fathom",
    "language": "pt-BR"
  },
  "metadata": {
    "title": "Reunião Chatbot - Sprint 48",
    "date_start": "2025-11-24T14:00:00Z",
    "project_code": "CHATBOT",
    "sprint_code": "Sprint-2025-W48"
  },
  "options": {
    "auto_extract": true,
    "generate_minutes": true,
    "detect_financial": true,    // ← NOVO: Detecta despesas
    "update_dashboard": true
  }
}

// Response 201
{
  "meeting": { "id": " ... ", "code": "MTG-2025-11-24-CHATBOT" },
  "extracted": {
    "decisions": 3,
    "actions": 7,
    "risks": 2,
```

```
    "kaizens": 1,
    "expenses_draft": 2  // ← NOVO: Despesas detectadas
  },
  "files_generated": {
    "ata": "40-Reunioes/2025-11-24-Reuniao-Chatbot.md"
  }
}
```

---

9. FRONTEND UNIFICADO

9.1 Estrutura de Navegação

```
┌─────────────────────────────────────────────────────────────┐
│                    SIDEBAR UNIFICADA                        │
├─────────────────────────────────────────────────────────────┤
│                                                             │
│  🚀 ERP-UZZAI                                               │
│  "Think Smart, Think Uzz.Ai"                                │
│                                                             │
│  ─────────────────────────                                  │
│                                                             │
│  📊 DASHBOARD EXECUTIVO                                     │
│                                                             │
│  📂 GESTÃO INTERNA                                          │
│    ├── 📁 Projetos                                          │
│    │     ├── CHATBOT                                       │
│    │     ├── SITE-BUILDER                                  │
│    │     └── NUTRI-TRAIN                                   │
│    ├── 🗓️ Reuniões                                          │
│    ├── ✅ Ações (Kanban)                                    │
│    ├── 📓 Bullet Journal                                    │
│    ├── 👥 Equipe                                            │
│    └── 📈 Performance                                       │
│                                                             │
│  🛒 ERP COMERCIAL                                           │
│    ├── 🏷️ Produtos                                          │
│    ├── 📦 Estoque                                           │
│    ├── 💳 Vendas (PDV)                                      │
│    ├── 📄 Pedidos                                           │
│    └── 👤 Clientes                                          │
│                                                             │
│  🔥 FINANCEIRO                                              │
│    ├── 💵 Fluxo de Caixa                                    │
│    ├── 📋 Contas a Pagar                                    │
│    ├── 📋 Contas a Receber                                  │
│    ├── 🧾 Notas Fiscais                                     │
│    └── 📊 Relatórios                                        │
│                                                             │
│  🧠 INTELIGÊNCIA                                            │
│    ├── 🔍 RAG Insights                                      │
│    └── 📚 Base de Conhecimento                              │
│                                                             │
│  ⚙️ CONFIGURAÇÕES                                           │
│                                                             │
│  ─────────────────────────                                  │
│                                                             │
│  👤 Pedro Vitor                                             │
│     CEO & Founder                                           │
└─────────────────────────────────────────────────────────────┘
```

9.2 Páginas por Módulo

Módulo	Páginas	Componentes Principais
Dashboard	`/`	KPIs, Charts, Alertas, Atividade Recente
Projetos	`/projetos`, `/projetos/[id]`	ProjectCard, SprintBoard, GanttChart
Reuniões	`/reunioes`, `/reunioes/[id]`	MeetingCard, AtaViewer, EntityList
Ações	`/acoes`	KanbanBoard, ActionCard, FilterBar
Bujo	`/bullet-journal`	DailyView, WeeklyView, MonthlyView
Equipe	`/equipe`, `/equipe/[id]`	PersonCard, PerformanceChart, 360View
Produtos	`/produtos`, `/produtos/[id]`	ProductCard, ProductForm, StockBadge
Estoque	`/estoque`	MovimentacaoForm, StockTable, AlertBadge
Vendas	`/vendas`, `/pdv`	PDVScreen, VendaForm, SaleHistory
Clientes	`/clientes`, `/clientes/[id]`	PessoaForm, CustomerHistory, 360View
Financeiro	`/financeiro`	CashFlowChart, DRETable, BudgetChart
NF	`/notas-fiscais`	InvoiceList, InvoiceEmit, NFViewer
RAG	`/rag-insights`	SimilaritySearch, DedupeAlert
KB	`/conhecimento`	KBCard, CategoryFilter, SearchBar

---

10. FLUXOS DE NEGÓCIO

10.1 Fluxo: Reunião → Ata → Despesa

```
┌──────────────┐    ┌──────────────┐    ┌──────────────┐    ┌──────────────┐
│  Transcrição │ ──▶│  Multi-Agent │ ──▶│  Extração    │ ──▶│     Ata      │
│   (Fathom)   │    │ Orchestrator │    │  Entidades   │    │ Estruturada  │
└──────────────┘    └──────────────┘    └──────────────┘    └──────────────┘
                                                                   │
                                                                   ▼
                    ┌──────────────────────────────────────────────┐
                    │              ENTIDADES EXTRAÍDAS             │
                    ├──────────────────────────────────────────────┤
                    │ D-2025-042: Usar Capacitor                   │
                    │ A-2025-123: Implementar login @Luis 📅 01/12 │
                    │ A-2025-124: Contratar designer R$2.500 @Pedro│ ← FINANCEIRO
                    │ K-T-015: Commits pequenos evitam conflitos   │
                    │ R-CHATBOT-003: LOI atrasada                  │
                    └──────────────────────────────────────────────┘
                                                │
                                                ▼
                    ┌──────────────────────────────────────────────┐
                    │              FINANCIAL AGENT                 │
                    ├──────────────────────────────────────────────┤
                    │ Detecta: "Contratar designer R$2.500"        │
                    │ Verbo: "contratar" ✓                         │
                    │ Valor: R$ 2.500,00 ✓                         │
                    │ Categoria: design                            │
                    │ → Cria ExpenseDraft (status: draft)          │
                    └──────────────────────────────────────────────┘
                                                │
                                                ▼
                    ┌──────────────────────────────────────────────┐
                    │              EXPENSE DRAFT                   │
                    ├──────────────────────────────────────────────┤
                    │ ID: EXP-DRAFT-2025-001                       │
                    │ Descrição: Contratar designer                │
                    │ Projeto: CHATBOT                             │
                    │ Valor: R$ 2.500,00                           │
                    │ Categoria: design                            │
                    │ Status: AGUARDANDO APROVAÇÃO                 │
                    │ Origem: A-2025-124 (Reunião MTG-2025-11-24)  │
                    └──────────────────────────────────────────────┘
                                                │
                                    ┌───────────┼───────────┐
                                    ▼           ▼           ▼
                              ┌──────────┐ ┌──────────┐ ┌──────────┐
                              │ APROVAR  │ │ EDITAR   │ │ REJEITAR │
                              └──────────┘ └──────────┘ └──────────┘
                                    │
                                    ▼
                    ┌──────────────────────────────────────────────┐
                    │              EXPENSE CONFIRMED               │
                    ├──────────────────────────────────────────────┤
                    │ → Cria Transação em Contas a Pagar           │
                    │ → Atualiza Budget do Projeto CHATBOT         │
                    │ → Agenda pagamento                           │
                    └──────────────────────────────────────────────┘
```

10.2 Fluxo: Venda → Estoque → Financeiro

```
┌──────────────┐    ┌──────────────┐    ┌──────────────┐
│     PDV      │ ──▶│    Venda     │ ──▶│    Itens     │
│  (Frontend)  │    │  Registrada  │    │  da Venda    │
└──────────────┘    └──────────────┘    └──────────────┘
                                               │
                          ┌────────────────────┼────────────────────┐
                          ▼                    ▼                    ▼
                   ┌──────────────┐     ┌──────────────┐     ┌──────────────┐
                   │   ESTOQUE    │     │  FINANCEIRO  │     │   CLIENTE    │
                   │   -Qtd       │     │   +Receita   │     │  Histórico   │
                   └──────────────┘     └──────────────┘     └──────────────┘
                          │                    │
                          ▼                    ▼
                   ┌──────────────┐     ┌──────────────┐
                   │  Trigger     │     │  À Vista?    │
                   │  Estoque     │     │              │
                   │  Mínimo?     │     │  Sim ──▶ Baixa automática
                   └──────────────┘     │  Não ──▶ Contas a Receber
                          │             └──────────────┘
                          ▼
                   ┌──────────────┐
                   │   ALERTA     │
                   │  Reposição   │
                   └──────────────┘
```

---

11. STACK TECNOLÓGICO

11.1 Backend

Componente	Tecnologia	Versão
Linguagem	Python	3.11+
Framework API	FastAPI	0.104+
ORM	SQLAlchemy	2.0+
Validação	Pydantic	2.0+
LLM	OpenAI API	gpt-4o-mini
Embeddings	OpenAI	text-embedding-3-large
Vector DB	Qdrant	1.7+
Database	PostgreSQL	15+
Cache	Redis	7+
Migrations	Alembic	1.12+
CLI	Typer	0.9+
Templates	Jinja2	3.1+

11.2 Frontend

Componente	Tecnologia	Versão
Framework	Next.js	15+
UI Library	React	19+
Components	Shadcn/ui	latest
Styling	Tailwind CSS	3.4+
Language	TypeScript	5.3+
Charts	Chart.js	4.4+
Forms	React Hook Form	7+
Validation	Zod	3.22+
State	Zustand	4+

11.3 Infraestrutura

Componente	Tecnologia
Container	Docker + Docker Compose
CI/CD	GitHub Actions
Hosting API	Vercel / Railway
Hosting DB	Supabase / Neon
Vector DB	Qdrant Cloud
Storage	S3 / Cloudflare R2
Monitoring	Sentry

11.4 Docker Compose

```yaml
version: '3.8'

services:
  api:
    build: ./backend
    ports:
      - "8000:8000"
```

```
    environment:
      - DATABASE_URL=postgresql://user:pass@postgres:5432/erpuzzai
      - QDRANT_URL=http://qdrant:6333
      - REDIS_URL=redis://redis:6379
      - OPENAI_API_KEY=${OPENAI_API_KEY}
    depends_on:
      - postgres
      - qdrant
      - redis

  frontend:
    build: ./frontend
    ports:
      - "3000:3000"
    environment:
      - NEXT_PUBLIC_API_URL=http://api:8000

  postgres:
    image: postgres:15
    environment:
      POSTGRES_USER: user
      POSTGRES_PASSWORD: pass
      POSTGRES_DB: erpuzzai
    volumes:
      - postgres_data:/var/lib/postgresql/data

  qdrant:
    image: qdrant/qdrant:v1.7.0
    ports:
      - "6333:6333"
    volumes:
      - qdrant_data:/qdrant/storage

  redis:
    image: redis:7-alpine
    ports:
      - "6379:6379"

volumes:
  postgres_data:
  qdrant_data:
```

---

12. ESTRUTURA DE CÓDIGO

```
erp-uzzai-unified/
├── backend/
│   ├── src/
│   │   ├── domain/
│   │   │   ├── entities/
│   │   │   │   ├── __init__.py
│   │   │   │   ├── project.py
│   │   │   │   ├── meeting.py
│   │   │   │   ├── action.py
│   │   │   │   ├── decision.py
│   │   │   │   ├── person.py           # Unificado: cliente+fornecedor+colaborador
│   │   │   │   ├── product.py
│   │   │   │   ├── sale.py
│   │   │   │   ├── stock_movement.py
│   │   │   │   ├── transaction.py
│   │   │   │   └── invoice.py
│   │   │   ├── repositories/
│   │   │   │   └── ...
│   │   │   └── events/
│   │   │       └── domain_events.py
│   │   │
│   │   ├── application/
│   │   │   ├── use_cases/
│   │   │   │   ├── ingest_meeting.py
│   │   │   │   ├── create_sale.py
│   │   │   │   ├── process_stock.py
│   │   │   │   └── emit_invoice.py
│   │   │   └── services/
│   │   │       ├── rag_context_enricher.py
│   │   │       ├── financial_classifier.py
│   │   │       └── stock_calculator.py
│   │   │
│   │   ├── infrastructure/
│   │   │   ├── persistence/
│   │   │   │   ├── sql/
│   │   │   │   │   ├── database.py
│   │   │   │   │   ├── models.py
│   │   │   │   │   └── repositories/
│   │   │   │   └── file/
│   │   │   │       └── markdown_assembler.py
│   │   │   ├── rag/
│   │   │   │   ├── qdrant_client.py
│   │   │   │   └── embedding_service.py
│   │   │   ├── llm/
│   │   │   │   └── openai_client.py
│   │   │   └── agents/
│   │   │       ├── base_agent.py
│   │   │       ├── decision_agent.py
│   │   │       ├── action_agent.py
│   │   │       ├── financial_agent.py  # NOVO
│   │   │       └── orchestrator.py
│   │   │
│   │   └── interfaces/
│   │       ├── api/
│   │       │   ├── main.py
│   │       │   ├── routes/
│   │       │   │   ├── meetings.py
│   │       │   │   ├── projects.py
│   │       │   │   ├── products.py
│   │       │   │   ├── sales.py
│   │       │   │   ├── stock.py
│   │       │   │   ├── financial.py
│   │       │   │   └── invoices.py
│   │       │   └── schemas/
│   │       └── cli/
│   │
│   ├── migrations/
│   ├── tests/
│   ├── Dockerfile
│   └── pyproject.toml
│
├── frontend/
│   ├── src/
│   │   ├── app/
│   │   │   ├── layout.tsx
│   │   │   ├── page.tsx                # Dashboard
│   │   │   ├── projetos/
│   │   │   ├── reunioes/
│   │   │   ├── acoes/
│   │   │   ├── bullet-journal/
│   │   │   ├── equipe/
│   │   │   ├── produtos/
│   │   │   ├── estoque/
│   │   │   ├── vendas/
│   │   │   ├── pdv/
│   │   │   ├── clientes/
│   │   │   ├── financeiro/
│   │   │   ├── notas-fiscais/
│   │   │   ├── rag-insights/
│   │   │   └── conhecimento/
│   │   │
│   │   ├── components/
│   │   │   ├── ui/                     # Shadcn/ui
│   │   │   ├── layout/
│   │   │   │   ├── Sidebar.tsx
│   │   │   │   └── TopBar.tsx
│   │   │   ├── dashboard/
│   │   │   ├── projects/
│   │   │   ├── meetings/
│   │   │   ├── actions/
│   │   │   ├── products/
│   │   │   ├── sales/
│   │   │   ├── stock/
│   │   │   ├── financial/
│   │   │   └── forms/
│   │   │       ├── PessoaForm.tsx       # Unificado
│   │   │       ├── ProductForm.tsx
│   │   │       ├── VendaForm.tsx
│   │   │       ├── EstoqueOperacaoForm.tsx
│   │   │       └── TransacaoForm.tsx
│   │   │
│   │   ├── lib/
│   │   │   ├── api.ts
│   │   │   ├── utils.ts
│   │   │   └── hooks/
│   │   │
│   │   └── types/
│   │
│   ├── public/
│   ├── Dockerfile
│   └── package.json
│
├── docs/
│   └── architecture/
│
├── docker-compose.yml
└── README.md
```

---

13. ROADMAP DE IMPLEMENTAÇÃO

13.1 Gantt Chart

13.2 Checklist por Fase

Fase 0: Foundation

- [x] ~~Estrutura de pastas~~
- [x] ~~Modelo de domínio~~
- [ ] Setup PostgreSQL + Qdrant
- [ ] Multi-Agent Orchestrator
- [ ] RAG Context Enricher
- [ ] Frontend base (Next.js + Shadcn)

Fase 1: Gestão Interna

- [ ] CRUD Projetos
- [ ] Sprints + Dashboard
- [ ] Ingestão de Reuniões
- [ ] Extração Multi-Agente
- [ ] Ações + Kanban Board
- [ ] Bullet Journal

Fase 2: ERP Comercial

- [ ] PessoaForm unificado
- [ ] CRUD Produtos
- [ ] PDV / Vendas
- [ ] Estoque + Movimentações
- [ ] Preço Médio Ponderado

Fase 3: Financeiro

- [ ] Contas a Pagar
- [ ] Contas a Receber
- [ ] Fluxo de Caixa
- [ ] DRE
- [ ] Emissão NF

Fase 4: SaaS

- [ ] Multi-tenancy
- [ ] Billing
- [ ] Onboarding
- [ ] Integrações

MÉTRICAS DE SUCESSO

Métrica	Target
Extração Recall	≥ 85%
Extração Precision	≥ 80%
Deduplicação	100%
Latência API	≤ 200ms
Processamento Reunião	≤ 60s
Uptime	≥ 99.5%

REFERÊNCIAS

- `ARQUITETURA_ERP_UZZAI_COMPLETA.md` — Gestão Interna
- `ARQUITETURA_WEB_COMPLETA.md` — Stack Web
- `DIAGRAMAS_INTERLIGACOES.md` — MeguisPet
- `MAPA_INTERLIGACOES_SISTEMA.md` — Dependências
- `FORMULARIOS_REDESIGN.md` — Unificação

Última Atualização: 2025-11-29T16:00
Autor: Sistema de Documentação ERP-UzzAI
Versão: 3.0.0
Próxima Revisão: Após conclusão da Fase 0