# Planejamento da Interface de Usuário do Sistema LangNet

## 1. Visão Geral da Interface

A interface do usuário do LangNet será projetada para guiar o usuário por todo o ciclo de criação de aplicações baseadas em agentes, desde a concepção inicial até a implementação e monitoramento. A interface será estruturada em torno das principais etapas do fluxo de trabalho, com foco na usabilidade, visualização clara e edição interativa.

## 2. Componentes Principais da Interface

### 2.1 Dashboard Principal

#### 2.1.1 Layout e Organização

* Design moderno com sistema de grid responsivo
* Menu lateral para navegação principal entre seções
* Área central para conteúdo principal
* Barra superior para ações globais, busca e perfil de usuário

#### 2.1.2 Elementos do Dashboard

* Visão geral de projetos recentes com miniaturas e status
* Métricas e estatísticas sobre projetos ativos
* Acesso rápido a modelos e templates predefinidos
* Notificações sobre atualizações, alertas e tarefas pendentes

#### 2.1.3 Ações Principais

* Botão de destaque para "Novo Projeto"
* Acesso rápido à documentação e tutoriais
* Filtros para organizar e localizar projetos existentes
* Opções de importação e exportação de projetos

### 2.2 Criação e Gestão de Projetos

#### 2.2.1 Fluxo de Criação de Projeto

* Modal/página de criação com formulário inicial
* Campos para nome, descrição, domínio, e objetivo do projeto
* Seleção de templates ou início em branco
* Opções para configurações avançadas (LLMs a utilizar, frameworks, etc.)
* Configuração de colaboradores e permissões

#### 2.2.2 Visão do Projeto

* Header com nome do projeto, status e ações principais
* Exibição em forma de pipeline das etapas do projeto
* Indicadores visuais de progresso e completude
* Sidebar para navegação rápida entre seções do projeto
* Área de anotações e comentários colaborativos

#### 2.2.3 Gerenciamento de Recursos

* Interface para upload e gestão de documentos de requisitos
* Biblioteca de componentes e templates reutilizáveis
* Gestão de chaves de API e integrações externas
* Controle de versões e histórico de alterações
* Configurações de backup e sincronização

### 2.3 Interface para Etapas do Fluxo de Trabalho

#### 2.3.1 Leitura e Análise de Documentação

* Área de upload com suporte a drag-and-drop
* Visualizador integrado para diferentes formatos de documentos
* Interface para adicionar anotações e highlights
* Painel de extração automática de requisitos com edição
* Visualização de entidades e conceitos identificados em forma de grafo

#### 2.3.2 Geração e Edição de Especificação Funcional

* Editor rich-text para visualização e edição da especificação
* Organização em seções expansíveis/retráteis
* Ferramentas de formatação e estilo
* Histórico de versões com diff visual
* Painel lateral para revisão e comentários

#### 2.3.3 Definição de Agentes e Tarefas

* Interface visual tipo "card" para criação e edição de agentes
* Editor estruturado para cada componente do agente (role, goal, backstory, tools)
* Visualização das relações entre agentes
* Interface similar para tarefas, com destaque para fluxos e entradas/saídas
* Validação em tempo real de definições

#### 2.3.4 Geração e Edição de Arquivos YAML

* Editor de código com syntax highlighting para YAML
* Alternância entre visualização estruturada e código raw
* Validação em tempo real com feedback visual de erros
* Previsualização de como o sistema interpretará as definições
* Ferramentas de formatação automática e refatoração

### 2.4 Editor de Redes de Petri

#### 2.4.1 Área de Design Visual

* Canvas amplo com suporte a zoom, pan e seleção
* Paleta de componentes (places, transitions, arcs)
* Grid opcional para alinhamento
* Opções de layout automático
* Miniaturas para navegação em redes complexas

#### 2.4.2 Componentes e Interações

* Criação de lugares (places) através de drag-and-drop
* Adição de transições com vinculação a agentes/tarefas
* Desenho de arcos com definição de peso
* Tokens iniciais configuráveis
* Propriedades avançadas acessíveis via painel lateral

#### 2.4.3 Validação e Simulação

* Feedback visual para problemas na rede (deadlocks, etc.)
* Modo de simulação passo-a-passo
* Animação de fluxo de tokens
* Histórico de estados durante simulação
* Ferramenta de análise formal de propriedades

#### 2.4.4 Ferramentas de Edição Avançada

* Agrupamento de elementos para organização
* Templates para padrões comuns de rede
* Comentários e anotações no diagrama
* Modos de visualização (simplificado, detalhado, analítico)
* Exportação em vários formatos (PNG, SVG, JSON, PNML)

### 2.5 Geração e Visualização de Código

#### 2.5.1 Interface de Geração de Código

* Painel de configuração para opções de geração
* Visualização de estrutura do projeto a ser gerado
* Opções para inclusão/exclusão de componentes
* Configuração de destino (local, GitHub, etc.)

#### 2.5.2 Editor de Código Integrado

* Editor com syntax highlighting para Python e outros formatos
* Estrutura de arquivos navegável
* Terminal integrado para execução de comandos
* Depuração básica e visualização de logs
* Suporte a versões e branches

#### 2.5.3 Visualização de Resultados

* Visualização da saída da execução
* Explorador de estado do sistema em tempo de execução
* Monitoramento de uso de APIs e recursos
* Ferramentas de debugging e diagnóstico
* Captura e visualização de logs estruturados

### 2.6 Monitoramento e Analytics

#### 2.6.1 Dashboard de Monitoramento

* Visualização em tempo real de métricas chave
* Gráficos de atividade de agentes
* Estatísticas de uso de recursos (tokens, API calls, etc.)
* Visão de estado atual do sistema
* Alertas e notificações sobre anomalias

#### 2.6.2 Integração com Langfuse

* Visualização de traces e spans dentro da plataforma
* Filtros e busca para localizar eventos específicos
* Drill-down em operações de LLM para análise detalhada
* Anotação e compartilhamento de insights
* Visualizações agregadas e tendências ao longo do tempo

## 3. Recursos de Interface Específicos

### 3.1 Componentes Interativos

#### 3.1.1 Chat com Agentes

* Interface tipo chat para comunicação direta com agentes
* Histórico de conversas organizadas por sessão
* Ferramentas de prompt engineering integradas
* Visualização do "pensamento" do agente em tempo real
* Feedback e correção interativa

#### 3.1.2 Visualização de Fluxo de Execução

* Representação visual do fluxo de trabalho em execução
* Estado atual destacado em tempo real
* Navegação entre estados anteriores
* Detalhes de contexto e variáveis em cada ponto
* Opções para intervenção manual em pontos críticos

### 3.2 Personalização e Acessibilidade

#### 3.2.1 Opções de Customização

* Temas claro e escuro
* Personalização de layout e componentes visíveis
* Atalhos de teclado configuráveis
* Configurações de exibição (densidade, tamanho de fonte, etc.)
* Perfis de configuração salvos

#### 3.2.2 Recursos de Acessibilidade

* Suporte completo a screen readers
* Contraste adequado para todas as informações visuais
* Alternativas textuais para elementos visuais
* Navegação via teclado para todas as funções
* Compatibilidade com tecnologias assistivas

### 3.3 Colaboração e Compartilhamento

#### 3.3.1 Ferramentas Colaborativas

* Edição colaborativa em tempo real
* Sistema de comentários e feedback
* Controle de versões e histórico de alterações
* Resolução de conflitos de edição
* Atribuição de tarefas e responsabilidades

#### 3.3.2 Compartilhamento e Exportação

* Geração de links compartilháveis para diferentes níveis de acesso
* Exportação de projetos em formatos abertos
* Publicação de templates e componentes reutilizáveis
* Integração com repositórios de código (GitHub, GitLab, etc.)
* Exportação de documentação em formatos padrão

## 4. Fluxo de Interação do Usuário

### 4.1 Criação de um Novo Projeto

1. Iniciar Projeto
   * Usuário acessa dashboard principal
   * Clica em "Novo Projeto"
   * Preenche informações básicas (nome, descrição, domínio)
   * Seleciona template ou começa em branco
   * Configura integrações iniciais (APIs, ferramentas externas)
2. Carregar Documentação
   * Acessa a seção "Documentação"
   * Faz upload de arquivos relevantes
   * Visualiza documentos carregados
   * Solicita análise automática
   * Revisa e ajusta entidades e requisitos extraídos
3. Revisar Especificação Funcional
   * Navega para seção "Especificação"
   * Visualiza especificação gerada automaticamente
   * Edita conteúdo conforme necessário
   * Adiciona detalhes ou esclarecimentos
   * Valida completude da especificação
4. Definir Agentes e Tarefas
   * Acessa seção "Agentes & Tarefas"
   * Revisa agentes sugeridos e adiciona/remove conforme necessário
   * Edita detalhes de cada agente (role, goal, backstory)
   * Configura ferramentas disponíveis para cada agente
   * Define e refina tarefas com seus respectivos passos
5. Revisar Arquivos YAML
   * Navega para seção "YAML"
   * Examina arquivos gerados
   * Faz ajustes manuais se necessário
   * Valida sintaxe e estrutura
   * Confirma para prosseguir

### 4.2 Modelagem da Rede de Petri

1. Acessar Editor de Rede
   * Navega para seção "Rede de Petri"
   * Visualiza rede inicial gerada automaticamente
   * Acessa ferramentas de edição
2. Personalizar Rede
   * Adiciona/remove places conforme necessário
   * Configura transições e vincula a agentes/tarefas
   * Define arcos e seus pesos
   * Posiciona tokens iniciais
   * Organiza visualmente a rede para clareza
3. Validar e Simular
   * Executa validação automática da rede
   * Corrige problemas identificados
   * Inicia simulação para verificar comportamento
   * Ajusta configurações com base nos resultados
   * Salva versão final da rede
4. Customizar Visualização
   * Ajusta layout para melhor clareza
   * Agrupa elementos relacionados
   * Adiciona anotações explicativas
   * Configura diferentes modos de visualização
   * Exporta visualizações para documentação

### 4.3 Geração e Teste de Código

1. Configurar Geração
   * Acessa seção "Geração de Código"
   * Seleciona opções de implementação
   * Configura componentes a incluir
   * Define destino para o código gerado
2. Revisar Código
   * Examina código gerado
   * Faz ajustes manuais se necessário
   * Valida estrutura e organização
   * Adiciona comentários ou documentação
3. Executar e Testar
   * Inicia execução do sistema
   * Monitora logs e saídas
   * Interage com sistema via interface de teste
   * Verifica comportamento esperado
   * Identifica e corrige problemas
4. Análise de Performance
   * Revisa métricas de execução
   * Analisa uso de recursos
   * Identifica gargalos ou ineficiências
   * Ajusta configurações para otimização
   * Documenta resultados e recomendações

## 5. Considerações de Implementação da Interface

### 5.1 Tecnologias Recomendadas

* Frontend Principal:
  + React.js com TypeScript para componentes complexos
  + Material-UI ou Chakra UI como biblioteca de componentes
  + Redux ou Context API para gerenciamento de estado
  + React Router para navegação
* Editor de Rede de Petri:
  + D3.js ou Cytoscape.js para visualização de grafo
  + React Flow para componentes de diagrama interativo
  + Canvas API para animação de alta performance
* Interface Simplificada:
  + Streamlit para prototipagem rápida e casos de uso mais simples
  + Gradio para interfaces de demonstração
* Editor de Código:
  + Monaco Editor (mesmo usado no VS Code)
  + Prismjs para syntax highlighting
  + Integração com LSP (Language Server Protocol)

### 5.2 Considerações de Responsividade

* Design adaptativo para diferentes tamanhos de tela
* Layout fluido com breakpoints bem definidos
* Versão móvel com funcionalidades essenciais
* Interface adequada para tablets e dispositivos touch
* Priorização de conteúdo para diferentes formatos

### 5.3 Performance e Otimização

* Carregamento lazy de componentes pesados
* Virtualização de listas e tabelas grandes
* Otimização de imagens e assets
* Caching inteligente de dados
* Feedback visual para operações demoradas
* Throttling e debouncing para interações frequentes

### 5.4 Integração com Backend

* API REST para operações principais
* WebSockets para atualizações em tempo real
* GraphQL para queries complexas e personalizadas
* Autenticação via JWT ou OAuth
* Estratégias de retry e circuit breaking para resiliência

## 6. Wireframes e Fluxos Principais

### 6.1 Dashboard Principal

+---------------------------------------------------------------+

| LANGNET [User] [Settings]|

+---------------+---------------------------------------------+

| | |

| NAVIGATION | DASHBOARD |

| - Dashboard | |

| - Projects | +------------------+ +------------------+ |

| - Templates | | Recent Project 1 | | Recent Project 2 | |

| - Settings | +------------------+ +------------------+ |

| - Help | |

| | +------------------+ +------------------+ |

| | | Recent Project 3 | | + NEW PROJECT | |

| | +------------------+ +------------------+ |

| | |

| | SYSTEM METRICS |

| | [CPU/Memory Usage] [API Usage] [Storage] |

| | |

| | RECENT ACTIVITIES |

| | - Project X updated 2h ago |

| | - Generated code for Project Y |

| | - Completed training for Agent Z |

| | |

+---------------+---------------------------------------------+

### 6.2 Criação de Projeto

+---------------------------------------------------------------+

| CREATE NEW PROJECT [X] [Cancel]|

+---------------------------------------------------------------+

| |

| PROJECT NAME: [ ] |

| |

| DESCRIPTION: [ ] |

| [ ] |

| |

| DOMAIN: [ Select domain ▼ ] |

| |

| START FROM: ○ Blank Project |

| ○ Template [ Select template ▼ ] |

| |

| ADVANCED OPTIONS: |

| ┌─────────────────────────────────────────────────────────┐ |

| │ Default LLM: [ OpenAI GPT-4 ▼ ] │ |

| │ Framework: [ CrewAI ▼ ] │ |

| │ Memory System: [ LangChain ▼ ] │ |

| └─────────────────────────────────────────────────────────┘ |

| |

| [ BACK ] [ CREATE ▶ ] |

+---------------------------------------------------------------+

### 6.3 Editor de Rede de Petri

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+---------------+---------------------------------------------+

| | PETRI NET EDITOR [🔍] [⚙️] |

| PROJECT | |

| NAVIGATION | +-------+ +------------+ +-----+ |

| - Documents | | | | | | | |

| - Spec | | Place |----->| Transition |----->|Place| |

| - Agents | | P1 | | T1 | | P2 | |

| - Tasks | +-------+ +------------+ +-----+ |

| - YAML | | | |

| > Petri Net | | +------------+ | |

| - Code | +------------->| Transition |<-----+ |

| - Deploy | | T2 | |

| | +------------+ |

+---------------+ |

| | |

| COMPONENTS | |

| 🔵 Place | |

| ⬛ Transition | |

| ➡️ Arc | |

| ⚫ Token | |

| | |

+---------------+---------------------------------------------+

| PROPERTIES |

| Selected: Transition T1 |

| Linked Agent: Customer Service Agent |

| Linked Task: Process Customer Query |

| Guard Condition: [ ] |

| Execution Time: [ Estimated: 2-5s ] |

+---------------------------------------------------------------+

### 6.4 Visualização de Agentes

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+---------------+---------------------------------------------+

| | AGENTS [+ Add] |

| PROJECT | |

| NAVIGATION | +------------------------------------------+

| - Documents | | ✨ CUSTOMER SERVICE AGENT |

| - Spec | +------------------------------------------+

| > Agents | | Role: Primary support interface |

| - Tasks | | Goal: Understand and resolve customer |

| - YAML | | queries effectively |

| - Petri Net | | |

| - Code | | Backstory: |

| - Deploy | | You are an experienced customer service |

| | | professional with deep knowledge of... |

+---------------+ | |

| | | Tools: |

| AGENT LIST | | - KnowledgeBaseTool |

| ✅ Customer | | - TicketCreationTool |

| Service | | - CustomerHistoryTool |

| ✅ Technical | | |

| Support | | [Edit] [Delete] [Duplicate] |

| ✅ Billing | +------------------------------------------+

| Assistant | |

| ✅ Supervisor | AGENT RELATIONSHIPS |

| | [Graph visualization of agent interactions]|

+---------------+---------------------------------------------+

| VALIDATION |

| ✅ All required fields completed |

| ⚠️ Consider adding examples for better agent performance |

| ℹ️ This agent is used in 3 transitions in the Petri Net |

|  |
| --- |

# Wireframes Complementares - Sistema LangNet

## 6.3 Leitura e Análise de Documentação

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+-------------+-----------------------------------------------------+

| | DOCUMENTATION ANALYSIS [↻] [⚙️] |

| PROJECT | |

| NAVIGATION | UPLOAD AREA |

| - Documents | ┌─────────────────────────────────────────────────┐ |

| - Spec | │ 📁 Drag & Drop files here or click to browse │ |

| - Agents | │ │ |

| - Tasks | │ Supported: PDF, DOC, DOCX, TXT, MD │ |

| - YAML | │ Max size: 50MB per file │ |

| - Petri Net | └─────────────────────────────────────────────────┘ |

| - Code | |

| - Deploy | UPLOADED DOCUMENTS [+ Add] |

| | ┌─────────────────────────────────────────────────┐ |

| DOCUMENT | │ ✅ requirements.pdf [View] [Delete] │ |

| STATUS | │ ✅ business\_rules.docx [View] [Delete] │ |

| 📁 3 files | │ ✅ user\_stories.md [View] [Delete] │ |

| 📊 Analyzed | └─────────────────────────────────────────────────┘ |

| ⚠️ 2 Issues | |

| | ANALYSIS RESULTS |

+-------------+ ┌─────────────────────────────────────────────────┐ |

| | │ EXTRACTED ENTITIES │ |

| ACTIONS | │ • Actors: Customer, Agent, Supervisor │ |

| [📤 Export] | │ • Systems: CRM, Ticketing, Knowledge Base │ |

| [📋 Report] | │ • Processes: Query Resolution, Escalation │ |

| [🔄 Rerun] | │ • Rules: 24h response, Priority handling │ |

| | └─────────────────────────────────────────────────┘ |

| | |

| | REQUIREMENTS IDENTIFIED [Edit] [Export]|

| | ┌─────────────────────────────────────────────────┐ |

| | │ FUNCTIONAL REQUIREMENTS (12) │ |

| | │ FR001: System must handle customer queries │ |

| | │ FR002: Route tickets to appropriate agents │ |

| | │ FR003: Escalate unresolved issues │ |

| | │ [View All] │ |

| | └─────────────────────────────────────────────────┘ |

| | |

| | ISSUES & RECOMMENDATIONS |

| | ⚠️ Ambiguous requirement in business\_rules.docx |

| | ℹ️ Consider adding SLA specifications |

| | ✨ Suggested: Add multilingual support |

+-------------+-----------------------------------------------------+

| PROGRESS: Analysis Complete - 3 documents processed successfully |

+---------------------------------------------------------------+

## 6.4 Geração e Edição de Especificação Funcional

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+-------------+-----------------------------------------------------+

| | FUNCTIONAL SPECIFICATION [📝] [📊] |

| PROJECT | |

| NAVIGATION | DOCUMENT OUTLINE |

| - Documents | ┌─────────────────────────────────────────────────┐ |

| - > Spec | │ 1. Introduction [Collapse ▲] │ |

| - Agents | │ 2. System Overview [Expand ▼] │ |

| - Tasks | │ 3. Functional Requirements [Expand ▼] │ |

| - YAML | │ 4. Non-Functional Requirements [Expand ▼] │ |

| - Petri Net | │ 5. Data Model [Expand ▼] │ |

| - Code | │ 6. User Interface Requirements [Expand ▼] │ |

| - Deploy | │ 7. Integration Requirements [Expand ▼] │ |

| | └─────────────────────────────────────────────────┘ |

| VERSION | |

| CONTROL | CONTENT EDITOR [Rich Text] |

| v1.0 | ┌─────────────────────────────────────────────────┐ |

| [History] | │ ## 1. Introduction │ |

| [Compare] | │ │ |

| [Restore] | │ This document specifies the functional │ |

| | │ requirements for the Customer Support System │ |

+-------------+ │ designed to handle customer inquiries through │ |

| | │ intelligent routing and automated responses. │ |

| COMMENTS | │ │ |

| (3) | │ ### 1.1 Purpose │ |

| 💬 User1 | │ The purpose of this system is to... │ |

| "Review | │ │ |

| section 3" | │ ### 1.2 Scope │ |

| | │ The system will encompass: │ |

| 💬 Admin | │ - Automated query classification │ |

| "Approved | │ - Intelligent agent assignment │ |

| v1.0" | │ - Response tracking and escalation │ |

| | └─────────────────────────────────────────────────┘ |

+-------------+ |

| | VALIDATION STATUS |

| | ✅ Grammar and spelling check passed |

| | ✅ Requirements coverage: 95% |

| | ⚠️ Missing: Error handling specifications |

| | ℹ️ Estimated completeness: 85% |

+-------------+-----------------------------------------------------+

| ACTIONS: [Generate PDF] [Export Word] [Share Link] [Request Review] |

+---------------------------------------------------------------+

## 6.5 Definição e Edição de Tarefas

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+-------------+-----------------------------------------------------+

| | TASKS DEFINITION [+ Add] |

| PROJECT | |

| NAVIGATION | TASK OVERVIEW [Grid] [List]|

| - Documents | ┌───────────────┐ ┌───────────────┐ ┌──────────────┐ |

| - Spec | │🎯 PROCESS │ │🔍 ANALYZE │ │📊 ESCALATE │ |

| - Agents | │CUSTOMER QUERY │ │SENTIMENT │ │COMPLEX ISSUE │ |

| - > Tasks | │ │ │ │ │ │ |

| - YAML | │Agent: Service │ │Agent: Analysis│ │Agent: Senior │ |

| - Petri Net | │Priority: High │ │Priority: Med │ │Priority: High│ |

| - Code | │Status: Active │ │Status: Active │ │Status: Draft │ |

| - Deploy | └───────────────┘ └───────────────┘ └──────────────┘ |

| | |

| TASK LIST | ┌───────────────┐ ┌───────────────┐ ┌──────────────┐ |

| ✅ Process | │📋 GENERATE │ │📝 UPDATE │ │✉️ NOTIFY │ |

| Query | │RESPONSE │ │KNOWLEDGE BASE │ │CUSTOMER │ |

| ✅ Analyze | │ │ │ │ │ │ |

| Sentiment | │Agent: Writer │ │Agent: KB Mgr │ │Agent: Comms │ |

| 🔄 Generate | │Priority: Med │ │Priority: Low │ │Priority: High│ |

| Response | │Status: Ready │ │Status: Pending│ │Status: Ready │ |

| ⏳ Escalate | └───────────────┘ └───────────────┘ └──────────────┘ |

| Issue | |

| | |

+-------------+ SELECTED TASK: Process Customer Query [Edit] [Del]|

| | ┌─────────────────────────────────────────────────┐ |

| TASK | │ TASK DETAILS │ |

| PROPERTIES | │ Name: Process Customer Query │ |

| Description | │ Description: Analyze incoming customer query │ |

| Expected | │ and determine appropriate response path │ |

| Output | │ │ |

| Tools | │ Expected Output: Structured classification │ |

| Context | │ with priority level and routing information │ |

| Human Input | │ │ |

| | │ Agent Assignment: Customer Service Agent │ |

+-------------+ │ │ |

| | │ PROCESSING STEPS: │ |

| VALIDATION | │ 1. Parse incoming query text │ |

| ✅ Required | │ 2. Identify key entities and intent │ |

| fields | │ 3. Classify urgency level │ |

| ✅ Output | │ 4. Determine routing category │ |

| format | │ 5. Generate structured response │ |

| ⚠️ Missing | │ │ |

| context | │ Tools Required: │ |

| | │ - Text Analysis Tool │ |

| | │ - Classification Model │ |

| | │ - Priority Assessment Tool │ |

| | └─────────────────────────────────────────────────┘ |

+-------------+-----------------------------------------------------+

| DEPENDENCIES: Show task relationships [Dependency Graph] [Matrix] |

+---------------------------------------------------------------+

## 6.6 Geração e Revisão de Arquivos YAML

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+-------------+-----------------------------------------------------+

| | YAML CONFIGURATION [Generate] [Validate]|

| PROJECT | |

| NAVIGATION | FILE STRUCTURE [Tree] [Tabs]|

| - Documents | ┌─────────────────────────────────────────────────┐ |

| - Spec | │ 📁 project\_config/ │ |

| - Agents | │ 📄 agents.yaml [Modified] │ |

| - Tasks | │ 📄 tasks.yaml [Modified] │ |

| - > YAML | │ 📄 tools.yaml [Current] │ |

| - Petri Net | │ 📄 config.yaml [Current] │ |

| - Code | └─────────────────────────────────────────────────┘ |

| - Deploy | |

| | YAML EDITOR [Raw] [Visual]|

| YAML | ┌─────────────────────────────────────────────────┐ |

| STATUS | │ # agents.yaml - Generated from specifications │ |

| ✅ Valid | │ │ |

| Syntax | │ customer\_service\_agent: │ |

| ✅ Schema | │ role: > │ |

| Valid | │ You are a customer service specialist │ |

| ⚠️ 3 | │ responsible for handling customer queries │ |

| Warnings | │ goal: > │ |

| | │ Understand customer needs and provide │ |

+-------------+ │ appropriate solutions or routing │ |

| | │ backstory: > │ |

| QUICK | │ You have 5 years of experience in │ |

| ACTIONS | │ customer service and deep knowledge of │ |

| [Format] | │ company products and policies │ |

| [Validate] | │ tools: │ |

| [Download] | │ - knowledge\_base\_tool │ |

| [Copy] | │ - ticket\_creation\_tool │ |

| | │ - customer\_history\_tool │ |

+-------------+ │ verbose: true │ |

| | │ allow\_delegation: false │ |

| VALIDATION | │ │ |

| RESULTS | │ technical\_support\_agent: │ |

| ✅ Syntax | │ role: > │ |

| OK | │ Technical support specialist for complex │ |

| ⚠️ Line 23: | │ technical issues and troubleshooting │ |

| Consider | └─────────────────────────────────────────────────┘ |

| shorter | |

| backstory | VALIDATION PANEL [Auto-Fix] |

| ⚠️ Line 45: | • Line 23: Backstory text may be too verbose |

| Missing | • Line 45: Missing 'allow\_code\_execution' field |

| field | • Line 67: Tool 'advanced\_analytics' not defined |

| | • Suggestion: Add examples for better performance |

+-------------+-----------------------------------------------------+

| ACTIONS: [Auto-Generate from Specs] [Compare Versions] [Export All] |

+---------------------------------------------------------------+

## 6.7 Modelagem de Rede de Petri (Vista Expandida)

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+-------------+-----------------------------------------------------+

| | PETRI NET DESIGNER [🔍] [🎨] |

| PROJECT | ┌─────────────────────────────────────────────────┐ |

| NAVIGATION | │ CANVAS CONTROLS [Zoom: 100%] [Grid: On] │ |

| - Documents | │ [🔍+] [🔍-] [📐] [⚙️] [🎯] [💾] [↶] [↷] │ |

| - Spec | └─────────────────────────────────────────────────┘ |

| - Agents | |

| - Tasks | MAIN CANVAS |

| - YAML | ┌─────────────────────────────────────────────────┐ |

| > Petri Net | │ │ |

| - Code | │ ┌─────┐ ┌──────────────┐ ┌─────┐ │ |

| - Deploy | │ │ P1 │───▶│ T1: Process │───▶│ P2 │ │ |

| | │ │Start│ │ Query │ │Queue│ │ |

| COMPONENTS | │ │ ● │ │ [Agent1] │ │ ●● │ │ |

| PALETTE | │ └─────┘ └──────────────┘ └─────┘ │ |

| 🔵 Place | │ │ │ |

| ⬛ Transition| │ ▼ │ |

| ➡️ Arc | │ ┌─────┐ ┌──────────────┐ ┌─────┐ │ |

| ⚫ Token | │ │ P3 │◀───│ T2: Analyze │◀───│ P4 │ │ |

| 🔗 Reference | │ │Route│ │ Sentiment │ │Hold │ │ |

| | │ │ │ │ [Agent2] │ │ │ │ |

| TOOLS | │ └─────┘ └──────────────┘ └─────┘ │ |

| ✂️ Cut | │ │ │ │ │ |

| 📋 Copy | │ ▼ ▼ ▼ │ |

| 📌 Paste | │ ┌─────┐ ┌──────────────┐ ┌─────┐ │ |

| 🗑️ Delete | │ │ P5 │───▶│ T3: Generate │───▶│ P6 │ │ |

| 🎯 Select | │ │Resp │ │ Response │ │Send │ │ |

| | │ │ │ │ [Agent3] │ │ │ │ |

+-------------+ │ └─────┘ └──────────────┘ └─────┘ │ |

| | │ │ |

| MINIMAP | └─────────────────────────────────────────────────┘ |

| ┌─────────┐ | |

| │ ●──■ │ | PROPERTIES PANEL [📊] [🔧]|

| │ │ │ │ | Selected: T1 (Process Query) |

| │ ■──● │ | Agent: Customer Service Agent |

| │ [■] │ | Task: Process Customer Query |

| └─────────┘ | Guard: token\_count > 0 |

| | Priority: High |

| VALIDATION | Execution Time: 2-5 seconds |

| ✅ Structure| Input Places: P1 (Start) |

| ✅ Reachable| Output Places: P2 (Queue), P3 (Route) |

| ⚠️ Potential| Tokens Required: 1 |

| Deadlock | Success Rate: 95% |

| | |

+-------------+-----------------------------------------------------+

| SIMULATION: [▶️ Start] [⏸️ Pause] [⏹️ Stop] [🔄 Reset] [📈 Analysis] |

+---------------------------------------------------------------+

## 6.8 Geração e Visualização de Código

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+-------------+-----------------------------------------------------+

| | CODE GENERATION & REVIEW [🔧] [📋] |

| PROJECT | |

| NAVIGATION | GENERATION OPTIONS [Configure] |

| - Documents | ┌─────────────────────────────────────────────────┐ |

| - Spec | │ Framework: CrewAI v0.70.1 [Change] │ |

| - Agents | │ Memory System: LangChain Full [Change] │ |

| - Tasks | │ LLM Provider: OpenAI GPT-4o [Change] │ |

| - YAML | │ Output Format: Python Package [Change] │ |

| - Petri Net | │ Include Tests: ✅ [Config] │ |

| > Code | │ Include Docs: ✅ [Config] │ |

| - Deploy | └─────────────────────────────────────────────────┘ |

| | |

| FILE | PROJECT STRUCTURE [Generate] |

| EXPLORER | ┌─────────────────────────────────────────────────┐ |

| 📁 src/ | │ customer\_support\_system/ │ |

| 📄 main.py| │ ├── src/ │ |

| 📄 agents/| │ │ ├── main.py [Modified] │ |

| 📄 tasks/ | │ │ ├── agents/ │ |

| 📄 tools/ | │ │ │ ├── \_\_init\_\_.py [Generated]│ |

| 📁 tests/ | │ │ │ ├── customer\_service.py [Generated]│ |

| 📄 test\_\* | │ │ │ └── technical\_support.py [Generated]│ |

| 📁 docs/ | │ │ ├── tasks/ │ |

| 📁 config/ | │ │ │ ├── \_\_init\_\_.py [Generated]│ |

| | │ │ │ └── process\_query.py [Generated]│ |

+-------------+ │ │ └── tools/ │ |

| | │ │ └── knowledge\_base.py [Generated]│ |

| CODE | │ ├── tests/ │ |

| EDITOR | │ │ └── test\_integration.py [Generated]│ |

| [Python] | │ ├── docs/ │ |

| [YAML] | │ │ └── README.md [Generated]│ |

| [Markdown] | │ └── requirements.txt [Generated]│ |

| [JSON] | └─────────────────────────────────────────────────┘ |

| | |

+-------------+ CODE PREVIEW [Format] |

| | ┌─────────────────────────────────────────────────┐ |

| EXECUTION | │ # main.py - Generated Petri Net Implementation │ |

| STATUS | │ from frameworkagentsadapterv4 import \* │ |

| 🟢 Running | │ from pathlib import Path │ |

| CPU: 45% | │ │ |

| Memory: 2GB | │ class CustomerSupportNet: │ |

| | │ def \_\_init\_\_(self): │ |

| LOGS | │ self.state\_class = CustomerState │ |

| [Clear] | │ self.graph = Graph(self.state\_class) │ |

| [Filter] | │ │ |

| [Export] | │ def setup\_agents(self): │ |

| | │ # Customer Service Agent │ |

| Terminal | │ self.service\_agent = HybridAgentAdapter(│ |

| > python | │ name="customer\_service\_agent", │ |

| main.py | │ role="Customer service specialist", │ |

| | │ goal="Handle customer queries", │ |

| > Test | │ tools=[knowledge\_tool, ticket\_tool] │ |

| passed ✅ | │ ) │ |

| | └─────────────────────────────────────────────────┘ |

+-------------+-----------------------------------------------------+

| ACTIONS: [Run Tests] [Deploy] [Download Package] [Generate Docs] |

+---------------------------------------------------------------+

## 6.9 Monitoramento e Analytics (Dashboard Langfuse)

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+-------------+-----------------------------------------------------+

| | MONITORING DASHBOARD [📊] [⚙️] |

| PROJECT | |

| NAVIGATION | SYSTEM OVERVIEW [Real-time] [1h]|

| - Documents | ┌─────────────────────────────────────────────────┐ |

| - Spec | │ 🟢 System Status: Online Uptime: 99.5% │ |

| - Agents | │ 📈 Requests/min: 45 ⚡ Avg Response: 2.3s │ |

| - Tasks | │ 🤖 Active Agents: 8/12 💰 Token Usage: 15.2K │ |

| - YAML | │ ⚠️ Alerts: 2 warnings 🔄 Queue Depth: 12 │ |

| - Petri Net | └─────────────────────────────────────────────────┘ |

| - Code | |

| > Monitoring| AGENT PERFORMANCE [Details] |

| | ┌─────────────────────────────────────────────────┐ |

| TIME RANGE | │ Agent Name │Status│Success│Avg Time│Tokens│ |

| ⚪ Live | │ Customer Service │ 🟢 │ 96% │ 2.1s │ 4.2K │ |

| ⚪ 1 Hour | │ Technical Support│ 🟢 │ 94% │ 4.7s │ 6.8K │ |

| ⚫ 1 Day | │ Sentiment Analyzer│🟢 │ 99% │ 0.8s │ 1.2K │ |

| ⚪ 1 Week | │ Response Generator│🟡 │ 87% │ 3.2s │ 3.0K │ |

| | │ Escalation Mgr │ 🟢 │ 100% │ 1.5s │ 0.8K │ |

| FILTERS | └─────────────────────────────────────────────────┘ |

| 🔍 Agent | |

| 🔍 Task | EXECUTION TRACES [View All] |

| 🔍 Error | ┌─────────────────────────────────────────────────┐ |

| 🔍 Warning | │ 14:23:15 Query: "Billing issue resolution" │ |

| | │ ├─ 📥 Input processed (0.1s) │ |

+-------------+ │ ├─ 🤖 Agent: Customer Service (2.1s) │ |

| | │ ├─ 🔍 Sentiment: Negative (-0.8) (0.3s) │ |

| LANGFUSE | │ ├─ 🎯 Route: Billing Specialist (0.2s) │ |

| INTEGRATION | │ └─ ✅ Response sent (0.1s) Total: 2.8s │ |

| 🟢 Connected| │ │ |

| Project: | │ 14:22:48 Query: "Product information" │ |

| CSSupport | │ ├─ 📥 Input processed (0.1s) │ |

| API Key: | │ ├─ 🤖 Agent: Customer Service (1.8s) │ |

| \*\*\*\*-\*\*\*\* | │ ├─ 🔍 Sentiment: Neutral (0.0) (0.2s) │ |

| | │ └─ ✅ Response sent (0.1s) Total: 2.2s │ |

| ALERTS | └─────────────────────────────────────────────────┘ |

| 🔔 2 Active | |

| ⚠️ Response | METRICS CHARTS [Customize] |

| time spike | [📈 Response Times] [📊 Success Rates] [💰 Costs] |

| ⚠️ Token | [🔄 Queue Depth] [🤖 Agent Load] [⚡ Throughput] |

| limit near | |

+-------------+-----------------------------------------------------+

| QUICK ACTIONS: [Export Report] [Set Alert] [View Logs] [Refresh] |

+---------------------------------------------------------------+

## 6.10 Configuração de Deployment

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+-------------+-----------------------------------------------------+

| | DEPLOYMENT CONFIGURATION [🚀] [⚙️] |

| PROJECT | |

| NAVIGATION | DEPLOYMENT TARGET [Configure] |

| - Documents | ┌─────────────────────────────────────────────────┐ |

| - Spec | │ Environment: Production [Change] │ |

| - Agents | │ Platform: Docker + Kubernetes [Change] │ |

| - Tasks | │ Cloud Provider: AWS [Change] │ |

| - YAML | │ Region: us-east-1 [Change] │ |

| - Petri Net | │ Scaling: Auto (2-10 instances) [Change] │ |

| - Code | └─────────────────────────────────────────────────┘ |

| - Monitoring| |

| > Deploy | INFRASTRUCTURE [Generate] |

| | ┌─────────────────────────────────────────────────┐ |

| ENV | │ COMPUTE RESOURCES │ |

| SETTINGS | │ • CPU: 2 vCPU per instance │ |

| 🔧 Dev | │ • Memory: 4GB RAM per instance │ |

| 🔧 Staging | │ • Storage: 20GB SSD │ |

| 🔧 Prod | │ • Network: VPC with private subnets │ |

| | │ │ |

| SECURITY | │ SERVICES │ |

| 🔐 API Keys | │ • Load Balancer: Application LB │ |

| 🔐 Secrets | │ • Database: PostgreSQL (RDS) │ |

| 🔐 Certs | │ • Cache: Redis (ElastiCache) │ |

| | │ • Monitoring: CloudWatch + Langfuse │ |

+-------------+ │ • Queue: SQS for async processing │ |

| | └─────────────────────────────────────────────────┘ |

| DEPLOYMENT | |

| HISTORY | ENVIRONMENT VARIABLES [Add] [Import] |

| ✅ v1.2.1 | ┌─────────────────────────────────────────────────┐ |

| ✅ v1.2.0 | │ OPENAI\_API\_KEY: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* [Hidden] │ |

| │ LANGFUSE\_PUBLIC\_KEY: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* [Hidden] │ |

| │ LANGFUSE\_SECRET\_KEY: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* [Hidden] │ |

| │ DATABASE\_URL: postgresql://... [Hidden] │ |

| │ REDIS\_URL: redis://... [Hidden] │ |

| │ LOG\_LEVEL: INFO [Edit] │ |

| │ MAX\_WORKERS: 4 [Edit] │ |

| └─────────────────────────────────────────────────┘ |

| | |

| ❌ v1.1.9 | DEPLOYMENT STATUS [Refresh] |

| ❌ v1.1.8 | ┌─────────────────────────────────────────────────┐ |

| | │ Current: v1.2.1 (Production) 🟢 Healthy │ |

| ROLLBACK | │ Instances: 3/3 Running [Scale] │ |

| [Select | │ Health Check: Passing [Logs] │ |

| Version] | │ Last Deploy: 2024-03-15 14:30 UTC [Details] │ |

| | │ Uptime: 99.97% [Monitor] │ |

+-------------+ └─────────────────────────────────────────────────┘ |

| | |

| CI/CD | PIPELINE STATUS [Configure] |

| PIPELINE | Build: ✅ Passed (2m 34s) [View Log] |

| ✅ Build | Tests: ✅ 127/127 passed (1m 12s) [View Log] |

| ✅ Test | Security: ✅ No vulnerabilities [View Log] |

| ✅ Security | Deploy: ✅ Completed (45s) [View Log] |

| ✅ Deploy | |

+-------------+-----------------------------------------------------+

| ACTIONS: [Deploy Now] [Schedule Deploy] [Rollback] [View Metrics] |

+---------------------------------------------------------------+

## 6.11 Interface de Chat e Controle de Agentes

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+-------------+-----------------------------------------------------+

| | AGENT INTERACTION CENTER [🎯] [⚙️] |

| PROJECT | |

| NAVIGATION | ACTIVE AGENTS [Refresh] |

| - Documents | ┌─────────────────────────────────────────────────┐ |

| - Spec | │ 🟢 Customer Service │Load: 67% │Queue: 5 │ |

| - Agents | │ 🟢 Technical Support │Load: 23% │Queue: 1 │ |

| - Tasks | │ 🟢 Sentiment Analyzer │Load: 45% │Queue: 8 │ |

| - YAML | │ 🟡 Response Generator │Load: 89% │Queue: 12 │ |

| - Petri Net | │ 🟢 Escalation Manager │Load: 12% │Queue: 0 │ |

| - Code | └─────────────────────────────────────────────────┘ |

| - Monitoring| |

| - Deploy | CHAT INTERFACE [Agent: All] |

| > Live Chat | ┌─────────────────────────────────────────────────┐ |

| | │ 💬 CONVERSATION HISTORY │ |

| AGENT | │ │ |

| SELECTOR | │ [14:25] User: Test the customer query processing│ |

| 🤖 All | │ [14:25] 🤖 Customer Service: I'll process a │ |

| 🤖 Customer | │ sample customer query. Please provide the text. │ |

| Service | │ │ |

| 🤖 Technical| │ [14:26] User: "My billing shows wrong amount" │ |

| Support | │ [14:26] 🤖 Customer Service: I've analyzed the │ |

| 🤖 Sentiment| │ query. Detected: Billing issue, Negative │ |

| Analyzer | │ sentiment (-0.6). Routing to Billing Specialist│ |

| 🤖 Response | │ │ |

| Generator | │ [14:26] 🤖 Technical Support: Query received │ |

| 🤖 Escalation| │ from routing. Category: Billing/Account. │ |

| Manager | │ Estimated resolution time: 5-10 minutes. │ |

| | │ │ |

+-------------+ │ [14:27] 🤖 Response Generator: Draft response │ |

| | │ prepared: "I understand your billing concern. │ |

| QUICK | │ Let me review your account details..." │ |

| COMMANDS | │ │ |

| /status | │ [14:27] User: Approve and send response │ |

| /restart | │ [14:27] ✅ System: Response sent to customer │ |

| /debug | └─────────────────────────────────────────────────┘ |

| /metrics | |

| /help | MESSAGE INPUT [Send] [📎] |

| | ┌─────────────────────────────────────────────────┐ |

| TEMPLATES | │ Type your message or command... │ |

| 📝 Test | └─────────────────────────────────────────────────┘ |

| Query | |

| 📝 Debug | AGENT CONTROLS [Emergency] |

| Session | [🔄 Restart Agent] [⏸️ Pause] [📊 Details] [🔧 Config]|

| 📝 Load | |

| Test | SYSTEM STATUS |

| | Net State: Processing | Active Transitions: 3 |

+-------------+ Current Place: P2 (Queue) | Tokens: 5 | Next: T2 |

+---------------------------------------------------------------+

## 6.12 Visualização de Execução em Tempo Real

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+-------------+-----------------------------------------------------+

| | REAL-TIME EXECUTION MONITOR [⏸️] [🔄] |

| PROJECT | |

| NAVIGATION | NETWORK VISUALIZATION [Controls] |

| - Documents | ┌─────────────────────────────────────────────────┐ |

| - Spec | │ ┌─────┐ ⚡ ┌──────────────┐ ⚡ ┌─────┐ │ |

| - Agents | │ │ P1 │──▶│ T1: Process │──▶│ P2 │ │ |

| - Tasks | │ │Start│ │ Query │ │Queue│ │ |

| - YAML | │ │ ● │ │ [Agent1] │ │ ●●● │ │ |

| - Petri Net | │ └─────┘ └──────────────┘ └─────┘ │ |

| - Code | │ │ │ |

| - Monitoring| │ ▼ │ |

| - Deploy | │ ┌─────┐ 🔥 ┌──────────────┐ ┌─────┐ │ |

| - Live Chat | │ │ P3 │◀───│ T2: Analyze │◀───│ P4 │ │ |

| > Execution | │ │Route│ │ Sentiment │ │Hold │ │ |

| | │ │ ● │ │ [Agent2] │ │ │ │ |

| LEGEND | │ └─────┘ └──────────────┘ └─────┘ │ |

| ⚡ Active | │ │ │ │ │ |

| 🔥 Processing│ ▼ ▼ ▼ │ |

| ⏸️ Waiting | │ ┌─────┐ ┌──────────────┐ ┌─────┐ │ |

| ❌ Error | │ │ P5 │───▶│ T3: Generate │───▶│ P6 │ │ |

| ✅ Complete | │ │Resp │ │ Response │ │Send │ │ |

| | │ │ │ │ [Agent3] │ │ ● │ │ |

| SPEED | │ └─────┘ └──────────────┘ └─────┘ │ |

| [🐌] [🐰] | └─────────────────────────────────────────────────┘ |

| | |

+-------------+ EXECUTION TIMELINE [Expand] |

| | ┌─────────────────────────────────────────────────┐ |

| ACTIVE | │ 14:30:15 │T1 Fired │Customer Service Agent │ |

| PROCESSES | │ 14:30:17 │T2 Started │Sentiment Analysis │ |

| 🔥 Query#127| │ 14:30:18 │T2 Complete │Result: Negative (-0.7) │ |

| 🔥 Query#128| │ 14:30:18 │T3 Fired │Response Generation │ |

| ⏸️ Query#129| │ 14:30:20 │P6 Updated │Token added to Send │ |

| | │ 14:30:21 │T4 Fired │Notification Sent │ |

| QUEUE | └─────────────────────────────────────────────────┘ |

| Pending: 12 | |

| Processing:5| PERFORMANCE METRICS [Details] |

| Completed:89| ┌─────────────────────────────────────────────────┐ |

| | │ Throughput: 45 queries/min ↗️ +15% from avg │ |

| ALERTS | │ Avg Processing: 2.3s ↘️ -0.2s improved │ |

| 🟡 High | │ Queue Depth: 12 ⚠️ Above threshold │ |

| queue depth | │ Success Rate: 96.5% ✅ Within target │ |

| 🟢 All | │ Active Agents: 8/12 🔄 Auto-scaling │ |

| agents up | └─────────────────────────────────────────────────┘ |

+-------------+-----------------------------------------------------+

| CONTROLS: [⏸️ Pause All] [🛑 Emergency Stop] [📊 Export Data] [🔧 Tune] |

+---------------------------------------------------------------+

## 6.13 Configurações e Administração do Sistema

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+-------------+-----------------------------------------------------+

| | SYSTEM ADMINISTRATION [🔧] [👥] |

| PROJECT | |

| NAVIGATION | GENERAL SETTINGS [Apply] |

| - Documents | ┌─────────────────────────────────────────────────┐ |

| - Spec | │ Project Name: Customer Support System │ |

| - Agents | │ Description: Automated customer query handling │ |

| - Tasks | │ Domain: Customer Service │ |

| - YAML | │ Version: 1.2.1 │ |

| - Petri Net | │ Created: 2024-03-01 Modified: 2024-03-15 │ |

| - Code | │ Owner: Admin User │ |

| - Monitoring| └─────────────────────────────────────────────────┘ |

| - Deploy | |

| - Live Chat | LLM CONFIGURATION [Test] |

| - Execution | ┌─────────────────────────────────────────────────┐ |

| > Settings | │ Primary LLM: OpenAI GPT-4o [Change] │ |

| | │ Fallback LLM: Claude 3 Sonnet [Change] │ |

| SETTINGS | │ Temperature: 0.3 [0.0-1.0] │ |

| MENU | │ Max Tokens: 4000 [1-8000] │ |

| 🔧 General | │ Timeout: 30s [5-120s] │ |

| 🤖 LLMs | │ Retry Attempts: 3 [1-5] │ |

| 🛡️ Security | │ │ |

| 🔌 Integrations│ API Keys: │ |

| 👥 Users | │ OpenAI: sk-\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* [Edit] │ |

| 📊 Analytics| │ Anthropic: \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* [Edit] │ |

| 💾 Backup | └─────────────────────────────────────────────────┘ |

| 🚨 Alerts | |

| | SECURITY SETTINGS [Configure] |

+-------------+ ┌─────────────────────────────────────────────────┐ |

| | │ Authentication: OAuth 2.0 + JWT [Change] │ |

| USER | │ Session Timeout: 4 hours [1-24h] │ |

| MANAGEMENT | │ Password Policy: Strong [Edit] │ |

| 👤 Admin | │ 2FA Required: ✅ [Toggle] │ |

| 👤 Dev Team | │ API Rate Limiting: 1000/hour [Edit] │ |

| 👤 Operators| │ Audit Logging: ✅ [Configure]│ |

| 👤 Viewers | │ │ |

| | │ Access Permissions: │ |

| ROLES & | │ • Admins: Full access │ |

| PERMISSIONS | │ • Developers: Code, Deploy, Monitor │ |

| [Manage] | │ • Operators: Monitor, Chat, Control │ |

| | │ • Viewers: Read-only access │ |

+-------------+ └─────────────────────────────────────────────────┘ |

| | |

| INTEGRATIONS| INTEGRATION STATUS [Add New] |

| STATUS | ┌─────────────────────────────────────────────────┐ |

| 🟢 Langfuse | │ Langfuse: 🟢 Connected [Configure]│ |

| 🟢 OpenAI | │ OpenAI: 🟢 Connected (Quota: 78%) [Details] │ |

| 🟡 Slack | │ Slack: 🟡 Partial (Alerts only) [Fix] │ |

| ❌ Teams | │ MS Teams: ❌ Disconnected [Setup] │ |

| 🟢 GitHub | │ GitHub: 🟢 Connected (Repo sync) [Manage] │ |

| | │ Email SMTP: 🟢 Connected [Test] │ |

| WEBHOOKS | └─────────────────────────────────────────────────┘ |

| [Configure] | |

+-------------+-----------------------------------------------------+

| SYSTEM INFO: Version 2.1.0 | Uptime: 15d 7h | Users: 24 | Load: 67% |

+---------------------------------------------------------------+

## 6.14 Relatórios e Analytics Avançados

+---------------------------------------------------------------+

| PROJECT: Customer Support System [Save] [Export] [?]|

+-------------+-----------------------------------------------------+

| | ANALYTICS & REPORTING [📊] [📈] |

| PROJECT | |

| NAVIGATION | REPORT DASHBOARD [Customize] |

| - Documents | ┌─────────────────────────────────────────────────┐ |

| - Spec | │ TIME PERIOD: Last 30 Days [Change] │ |

| - Agents | │ ┌─────────────┐ ┌─────────────┐ ┌─────────────┐ │ |

| - Tasks | │ │📈 QUERIES │ │⚡ AVG TIME │ │✅ SUCCESS │ │ |

| - YAML | │ │ 15,847 │ │ 2.3s │ │ 96.5% │ │ |

| - Petri Net | │ │ ↗️ +23% │ │ ↘️ -0.2s │ │ ↗️ +1.2% │ │ |

| - Code | │ └─────────────┘ └─────────────┘ └─────────────┘ │ |

| - Monitoring| └─────────────────────────────────────────────────┘ |

| - Deploy | |

| - Live Chat | PERFORMANCE TRENDS [View All] |

| - Execution | ┌─────────────────────────────────────────────────┐ |

| - Settings | │ 📈 Query Volume (Daily) │ |

| > Analytics | │ 800│ ● │ |

| | │ 600│ ● ● │ |

| REPORT | │ 400│ ● │ |

| TYPES | │ 200│ ● │ |

| 📊 Overview | │ 0└────────────────────────────────────────── │ |

| 📈 Performance│ Mon Tue Wed Thu Fri Sat Sun │ |

| 🤖 Agents | └─────────────────────────────────────────────────┘ |

| 📝 Tasks | |

| 💰 Costs | AGENT PERFORMANCE BREAKDOWN [Details] |

| 🔄 Flows | ┌─────────────────────────────────────────────────┐ |

| 📋 Custom | │ Agent Name │Queries│Success│Avg Time│Cost│ |

| | │ Customer Service │ 8,234 │ 97.2% │ 2.1s │$124│ |

| TIME | │ Technical Support │ 3,156 │ 95.8% │ 4.2s │$89 │ |

| PERIODS | │ Sentiment Analyzer│12,847 │ 99.1% │ 0.8s │$45 │ |

| ⚫ 30 Days | │ Response Generator│ 7,892 │ 94.3% │ 3.1s │$156│ |

| ⚪ 7 Days | │ Escalation Mgr │ 234 │100.0% │ 1.2s │$12 │ |

| ⚪ 24 Hours | └─────────────────────────────────────────────────┘ |

| ⚪ Custom | |

| | COST ANALYSIS [Optimize] |

+-------------+ ┌─────────────────────────────────────────────────┐ |

| | │ Total Cost (30d): $426 │ |

| EXPORT | │ ┌─────────────────────────────────────────────┐ │ |

| OPTIONS | │ │ 💰 Cost Breakdown by Component │ │ |

| 📄 PDF | │ │ • LLM API Calls: $312 (73%) │ │ |

| 📊 Excel | │ │ • Compute Resources: $89 (21%) │ │ |

| 📈 CSV | │ │ • Storage & Network: $18 (4%) │ │ |

| 📧 Email | │ │ • Third-party Tools: $7 (2%) │ │ |

| 🔗 Share | │ └─────────────────────────────────────────────┘ │ |

| | │ │ |

| FILTERS | │ 📈 Cost Trend: ↗️ +15% vs last month │ |

| 🔍 Agent | │ 💡 Optimization Suggestion: Consider caching │ |

| 🔍 Date | │ for repeated queries (Est. saving: $45/mo) │ |

| 🔍 Task | └─────────────────────────────────────────────────┘ |

| 🔍 Error | |

| | BUSINESS METRICS [Configure] |

+-------------+ Customer Satisfaction: 4.7/5.0 (↗️ +0.2) |

| | First Response Time: 1.8min (↘️ -0.3min) |

| SCHEDULED | Resolution Rate: 94.2% (↗️ +2.1%) |

| REPORTS | Escalation Rate: 8.3% (↘️ -1.2%) |

| [Setup] | |

+-------------+-----------------------------------------------------+

| ACTIONS: [Generate Report] [Schedule] [Export Data] [Share Dashboard] |

+---------------------------------------------------------------+

Estes wireframes complementam o sistema LangNet com todas as telas necessárias para o fluxo completo de trabalho. Cada tela mantém a consistência visual e funcional estabelecida nos exemplos 6.1 e 6.2, incluindo:

Recursos Visuais Consistentes:

* Layout com navegação lateral e área principal
* Status indicators e progress tracking
* Botões de ação padronizados
* Painéis de informação organizados
* Validação em tempo real

Funcionalidades Principais Cobertas:

1. Upload e análise de documentação (6.3)
2. Edição de especificação funcional (6.4)
3. Definição detalhada de tarefas (6.5)
4. Geração e edição de YAML (6.6)
5. Editor avançado de Petri Net (6.7)
6. Geração e review de código (6.8)
7. Monitoramento com Langfuse (6.9)
8. Configuração de deployment (6.10)
9. Chat com agentes (6.11)
10. Execução em tempo real (6.12)
11. Administração do sistema (6.13)
12. Analytics e relatórios (6.14)

Cada tela segue a progressão lógica do processo, mantendo o usuário orientado sobre onde está no fluxo e permitindo navegação fácil entre as diferentes etapas do desenvolvimento da aplicação baseada em agentes.

Estes wireframes representam os componentes principais da interface, focando na clareza, usabilidade e integração natural com o fluxo de trabalho. A implementação final deve seguir princípios de design consistentes e considerar feedback de usuários em ciclos iterativos de desenvolvimento.