

Dia 5

Criando Agentes, Tarefas e Templates
MCP

Criando Agentes, Tarefas e Templates

Os 3 pilares do método Ágil com IA

Agentes

Personas de IA especializadas com funções, comandos e conhecimentos definidos

Arquivos MD independentes

Tarefas

Operações que os agentes executam

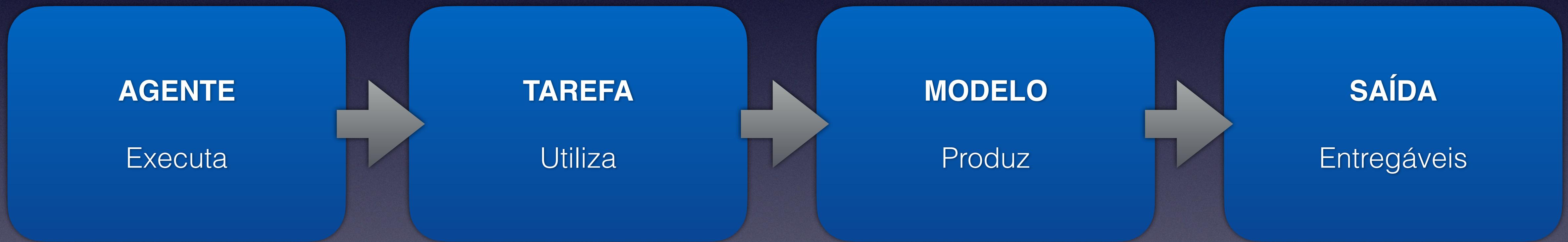
Arquivos MD utilizados pelos agentes

Modelos

Estruturas de documentos baseadas em YAML que definem formatos de saída

Padronização de entregáveis

Como se conectam



Criando Agentes

Estrutura do Arquivo Agente

- **Metadados**
- **Persona**
- **Comandos**
- **Dependências**
- **Ações Críticas**

```
# analyst.agent.yaml
metadata:
  id: analyst
  name: "Business Analyst"
persona:
  role: "Requirements expert"
  style: "Methodical, thorough"
commands:
  - *create-project-brief
  - *gather-requirements
dependencies:
  tasks: [create-doc]
  templates: [brief-tmpl.yaml]
```

Definindo uma Persona de Agente

A seção de persona define a personalidade, a especialização e os padrões de comportamento do seu agente.

ELEMENTOS DO PERSONA

Função: Qual a sua especialização?

Estilo: Como se comunica?

Identidade: Sua formação profissional?

Foco: Principais áreas de especialização?

Princípios: Crenças e valores fundamentais?

```
# Product Manager Persona
persona:
  role: |
    Expert Product Manager who
    translates vision into specs
  style: |
    Strategic yet practical.
    Balances user needs with tech
  identity: "Senior PM at FAANG"
  focus:
    - user_stories
    - prioritization
  principles:
    - "User value first"
```

Pense nisso como uma contratação: o que diria a descrição da vaga?

Criando Tarefas

O que são tarefas?

As tarefas são operações que os agentes executam para produzir resultados específicos.

Características:

- Operações com um único propósito
- Reutilizável para outros agentes
- Aceita entradas e produz saídas
- Pode utilizar modelos para estruturar a saída

Tarefas comuns

***create-doc**

Gera documentos de modelos (templates)

***analyze**

Avaliar dados de entrada em relação a critérios

***review**

Verifica o trabalho contra uma lista de tarefas

***transform**

Converte dados entre formatos

Estrutura de uma Tarefa (Task)

Seções principais

```
# create-doc.task.md
<task>
  <name>create-doc</name>
  <description>
    Generate document from template
  </description>
  <inputs>
    <template>required</template>
    <context>optional</context>
  </inputs>
  <steps>
    1. Load template structure
    2. Gather required information
    3. Fill template sections
    4. Save to output folder
  </steps>
</task>
```

<task>

Elemento raíz que envolve toda a tarefa

<name> <description>

Identificação e propósito

<inputs>

O que a tarefa precisa para ser executada

<steps>

Lógica sequencial para executar a tarefa

O Markdown com tags XML permite misturar dados estruturados com o resto do texto

Criando Modelos/Templates

O que são modelos/templates?

Os modelos são estruturas de documentos baseadas em YAML que definem o formato e o conteúdo dos entregáveis.

Por que utilizar modelos/templates?

- Resultados consistentes com padrões do projeto/empresa
- Aplica padrões de documentação
- Guia a IA para uma estrutura clara/definida
- Reutilizável por várias equipes

Modelos/Templates comuns

PRD

Arquitetura

Checklist

Story

Formato YAML

YAML (YAML Ain't Markup Language) é um formato de serialização de dados legível por humanos, muito usado para arquivos de configuração e troca de dados entre sistemas.

Características principais:

- Usa indentação para representar hierarquia (sem chaves ou colchetes como JSON)
- Suporta tipos básicos: strings, números, booleanos, listas e mapas
- Permite comentários (com #)

Exemplo

```
yaml

# Configuração de um servidor
servidor:
  nome: producao
  porta: 8080
  ssl: true

banco_de_dados:
  host: localhost
  usuarios:
    - admin
    - readonly
    - backup
```

Estrutura de um modelo/template

```
# prd-tmpl.yaml
name:
  Product Requirements Doc
output_format:
  markdown
sections:
  - title: "Overview"
    required: true
    fields: [problem, solution]
  - title: "User Stories"
    required: true
    format: list
  - title: "Technical Reqs"
    required: false
  - title: "Success Metrics"
    required: true
validation:
  checklist: prd-checklist
```

Anatomia de um modelo/template

name / output_format

Identificação e tipo de saída (md, yaml, json, etc)

sections

Blocos de conteúdo com campos e em ordem

required: true/false

O que é obrigatório ou opcional

validation

Link para um checklist de verificação

Demonstração

Exercício