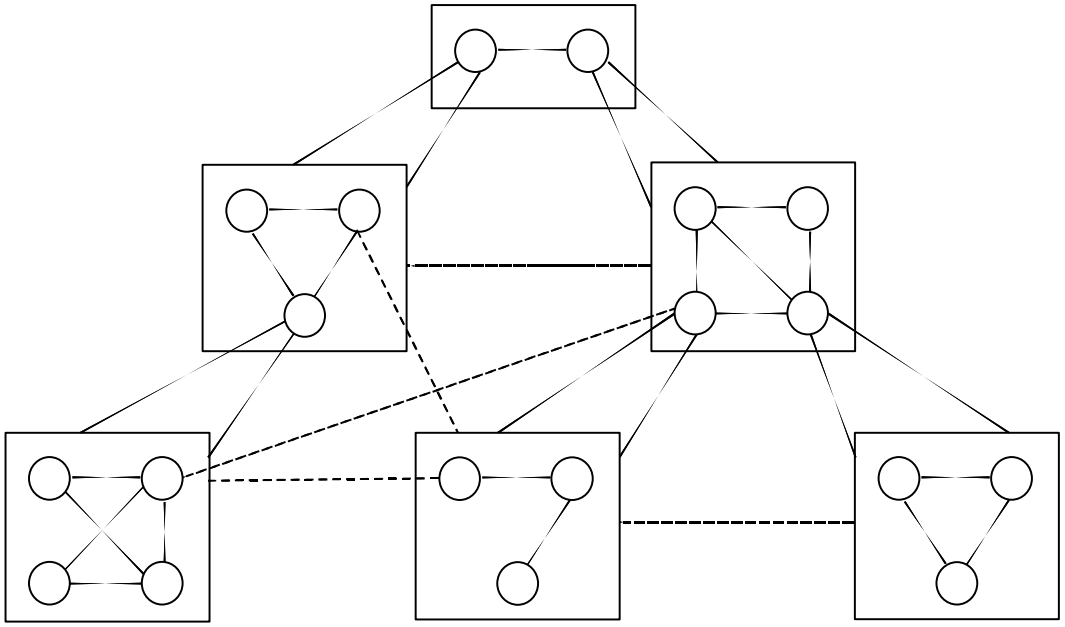
Engenharia de Software

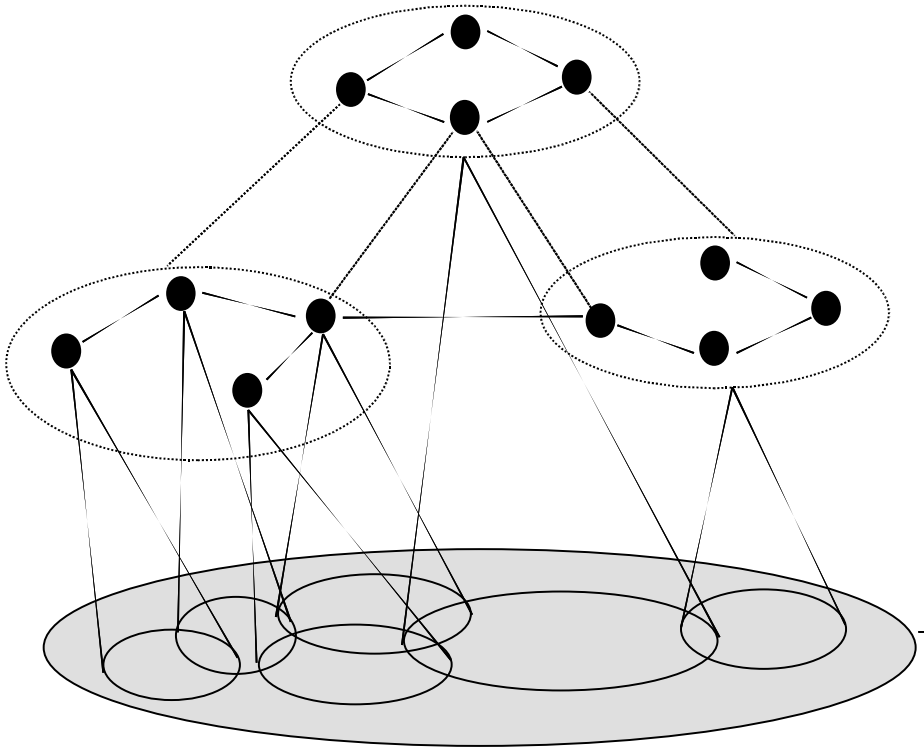
- Engenharia de Software
  - É a aplicação de uma abordagem sistemática, disciplinada e quantificada para o desenvolvimento, operação e manutenção de software.
- Engenharia de Software Orientada a Agente
  - É a aplicação de agentes na engenharia de software em termos de fornecer meios para analisar, projetar e construir sistemas de software.
  - abordagem recente na Engenharia de Software;
  - algumas metodologias estão propostas;
  - ideal para o desenvolvimento de sistemas complexos e distribuídos.

Sistemas Complexos



- Subsistema
- Componente do subsistema
- Composto de
- Interação freqüente
- Interação não-freqüente

Sistemas Multi-Agentes



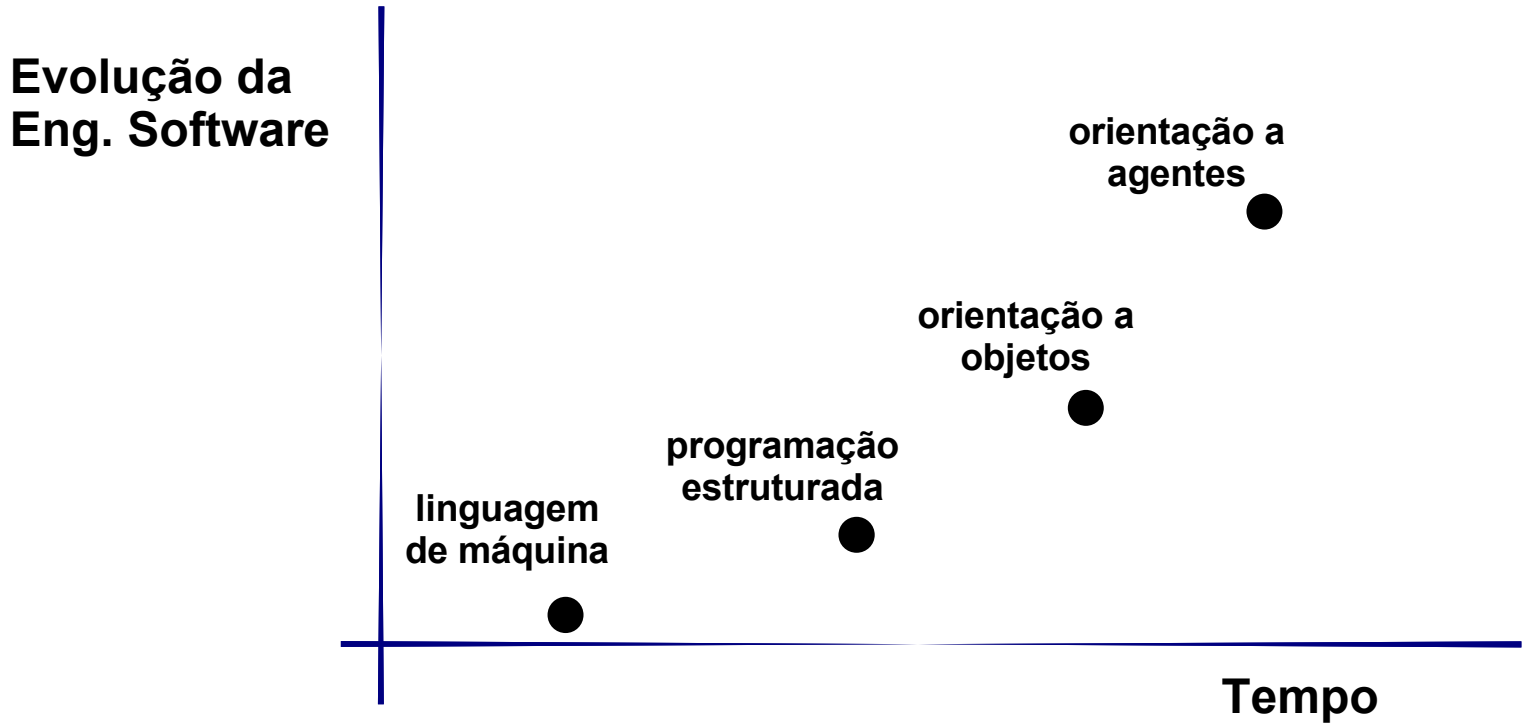
Agente

Interação

Relacionamento Organizacional

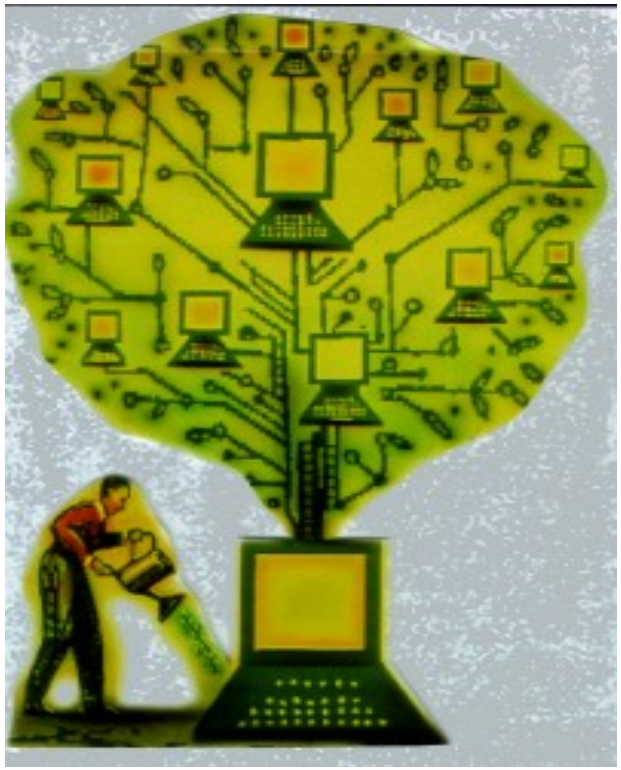
Visão do ambiente / esfera de influência

Agentes x Engenharia de Software



Evolução da Engenharia de Software

Engenharia de Software Orientada a Agente



Abordagem baseada em agente é uma evolução natural e lógica de várias abordagens contemporâneas da Eng. de Software.

Unidade de abstração → Agente

Metodologias

- Metodologias
  - Meios fornecidos pela Engenharia de Software (conceitos, diretrizes, atividades) para facilitar o processo de desenvolvimento de software.
- Metodologias Orientadas a Agentes:
  - Extensões das Metodologias OO: capturam similaridades entre agentes e objetos.
  - Extensões das Técnicas da Engenharia do Conhecimento: útil para modelar o conhecimento do agente.

AUML – Agent Unified Modeling Language

- Proposta por Odell, Parunak e Bauer
  - Fornece extensões baseados nos diagramas do UML (OO)
    - Mensagens → atos comunicativos
    - Agentes são representados por papéis e não por nomes
  - Modela protocolos para interações entre agentes, em 3 camadas:
    1. Definições gerais dos protocolos de interação
    2. Representa as interações particulares entre agentes
    3. Detalha o comportamento interno no agente:

- 1. Definições gerais dos protocolos de interação
  - pacotes: representam abstrações de diagramas quaisquer
  - templates: é um modelo parametrizado (papéis, restrições e comunicações)

AUML – Agent Unified Modeling Language

- 2. Representa as interações particulares entre agentes
  - Diagrama de Seqüência:
    - Especifica o papel exercido pelo agente.
    - Um mesmo agente pode aparecer em diferentes caixas, exercendo papéis diferentes.
    - Estende comunicação assíncrona e concorrente do UML→ threads
  - Diagrama de Colaboração:
    - Um mesmo agente exercendo papéis distintos pode aparecer em lugares distintos e a transição de um papel para o outro → arcos estereotipados.
  - Diagrama de Atividades:
    - As atividades exercidas pelos agentes são indicadas em diferentes swinlanes.
    - Permite a eliminação dos swinlanes, indicando embaixo de cada atividade qual é o agente que a executa.
  - Diagrama de Estados:
    - Recomendado para enfatizar as restrições de protocolo
    - Indica o nome do agente relacionado ao estado

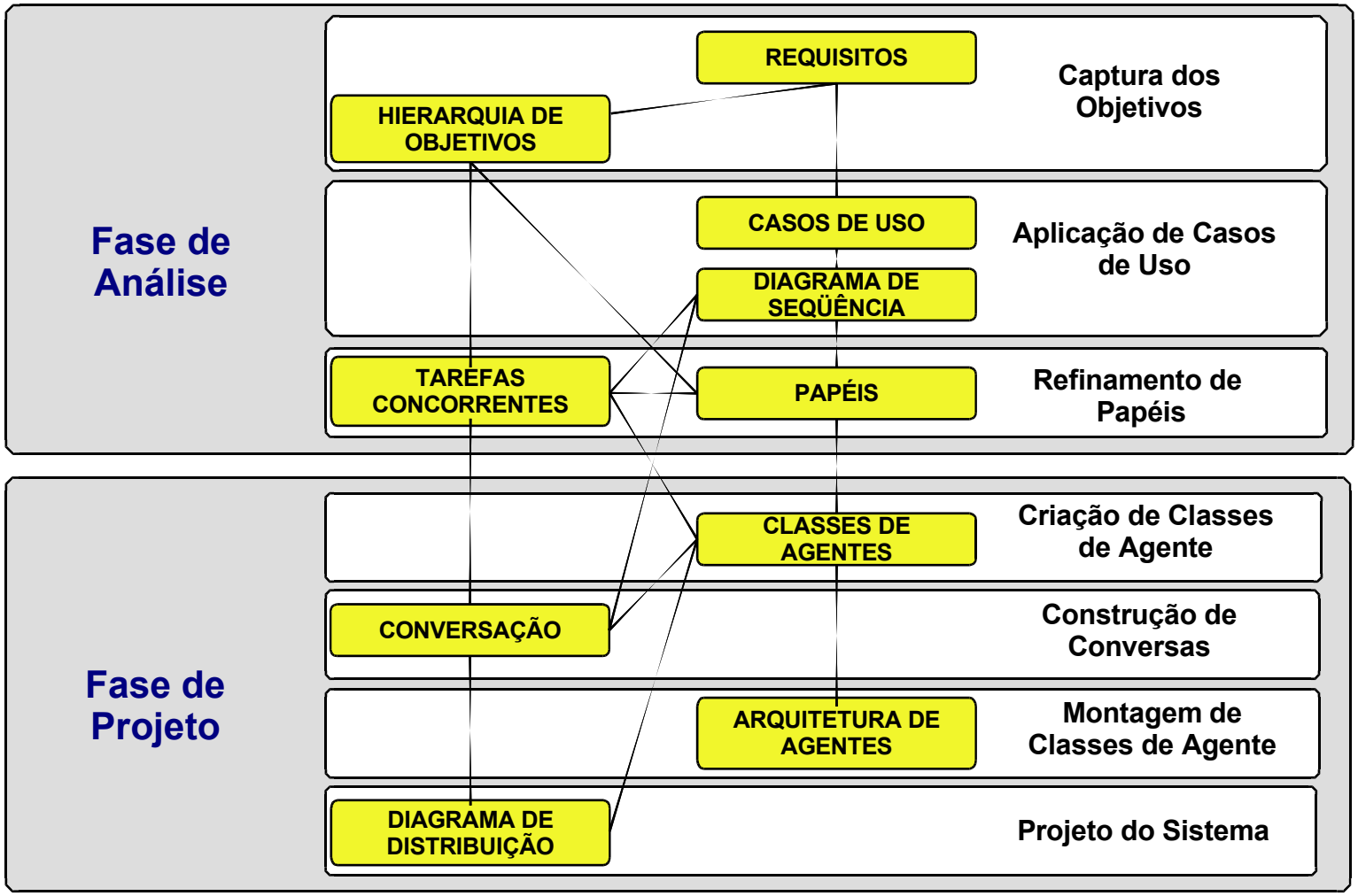
AUML – Agent Unified Modeling Language

- 3. Detalha o comportamento interno no agente:
  - Diagrama de Atividades
  - Diagrama de Estados

MaSE – Multiagent Systems Engineering

- Características
  - Air Force Institute of Technology (DeLoach e Wood)
  - Agentes não necessariamente autônomos e pró-ativos
    - processos de software simples que interagem para atingir o objetivo geral do sistema.
  - Assume a existência prévia da especificação dos requisitos para o início do desenvolvimento da metodologia.
  - AgentTool: gera códigos em Java baseado nos modelos do projeto.
- 2 Fases
  - **Fase de Análise:** Captura os objetivos do sistema, aplica os casos de uso e executa o refinamento dos papéis
  - **Fase de Projeto:** Cria as classes de agentes, construção da conversa entre os agentes, monta as classes de agente e projeta o sistema.

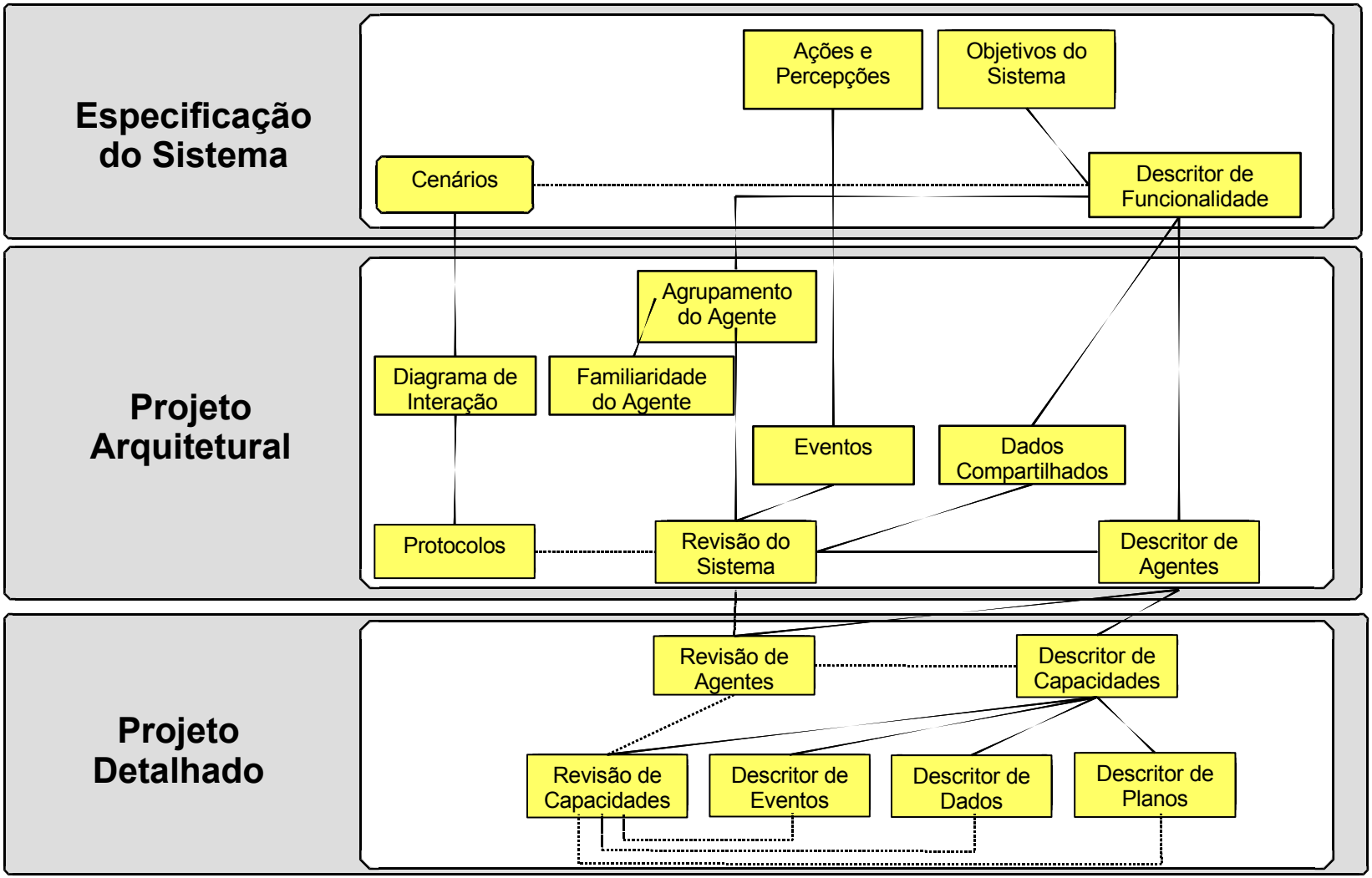
Metodologia MaSE



Metodologia Prometheus

- Desenvolvida pelo Agent Oriented Software Group (AOS – Estados Unidos, Reino Unido e Austrália)
- Nome: referência a um Titã da Mitologia Grega → protetor e benfeitor do homem
- Especificação do Sistema:
  - Identificar as funcionalidades básicas, objetivos
  - Determinar interfaces externas: percepções (entradas), ações (saídas) e dados compartilhados.
- Projeto Arquitetural:
  - Determinar os agentes do sistema e como irão interagir.
- Projeto Detalhado:
  - Focalizar no interior de cada agente
  - Definir as capacidades, planos, crenças e eventos.
  - Como o agente irá realizar suas tarefas no sistema.

Metodologia Prometheus



Metodologia Tropos

- Criado por pesquisadores de vários países (Brasil, Canadá e Itália)
- Integra as idéias das tecnologias de sistemas multi-agentes e da Engenharia de Requisitos
- Idéias Básicas:
  - a noção de agente e todas as noções relacionadas à mente (crença, objetivos, ações e planos) são usadas em todas as fases de desenvolvimento do software
  - adota uma abordagem de refinamento de passos, utilizando um conjunto específico de operadores de transformação
  - Requisitos Iniciais, Requisitos Finais, Projeto Arquitetural, Projeto Detalhado e Implementação

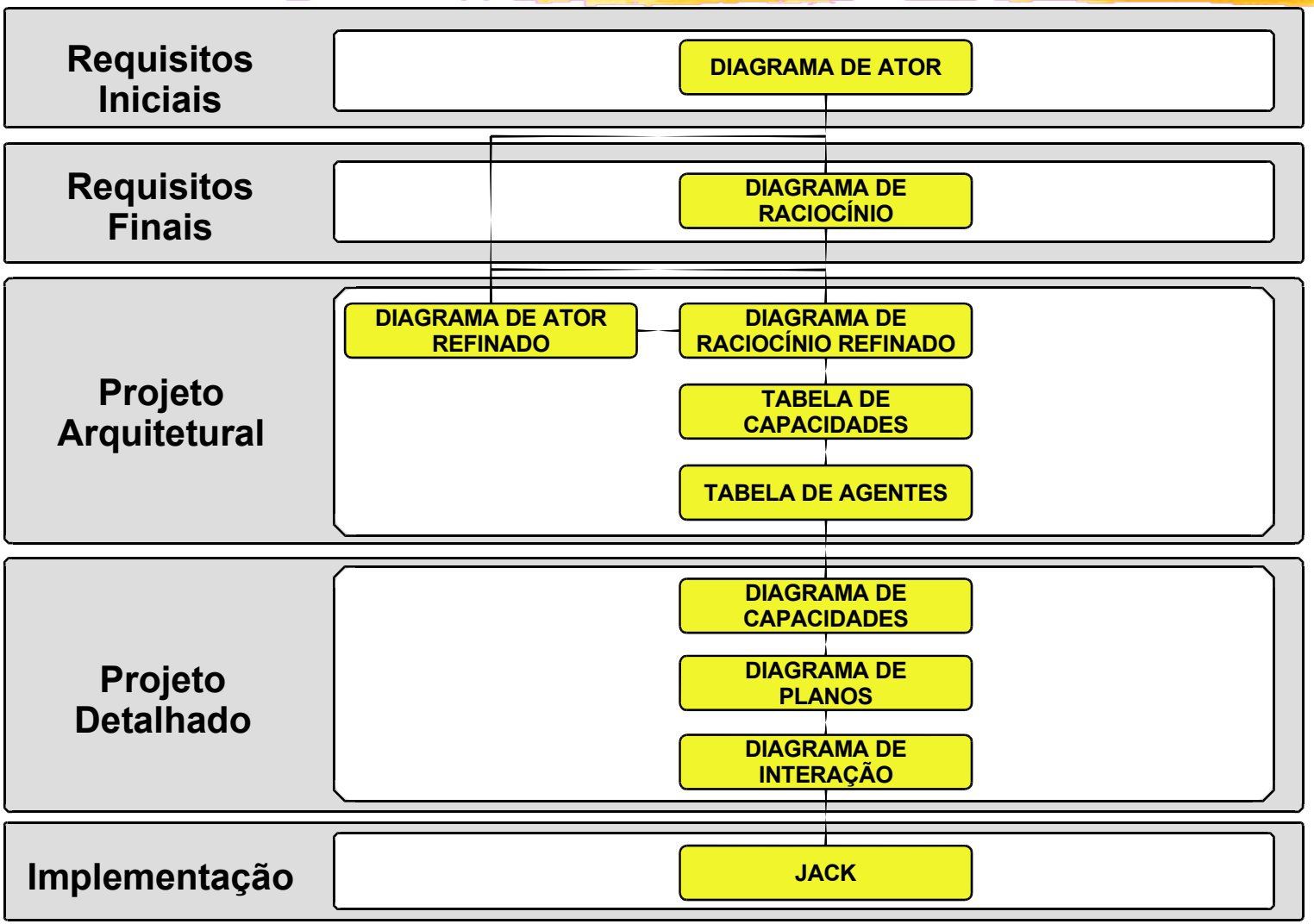
Metodologia Tropos

- Requisitos Iniciais:
  - identificação dos stakeholders relevantes do sistema (representados por atores) e suas intenções (metas). Cada meta é analisada do ponto de vista de seu ator resultando em um conjunto de dependências entre pares de atores.
    - Diagrama de ator → dependência estratégica
    - Diagrama de raciocínio → razão estratégica
- Requisitos Finais:
  - introduz o sistema a ser desenvolvido como um outro ator no diagrama de ator e o relaciona aos atores sociais em termos de dependências.
  - Refinamento dos diagramas.

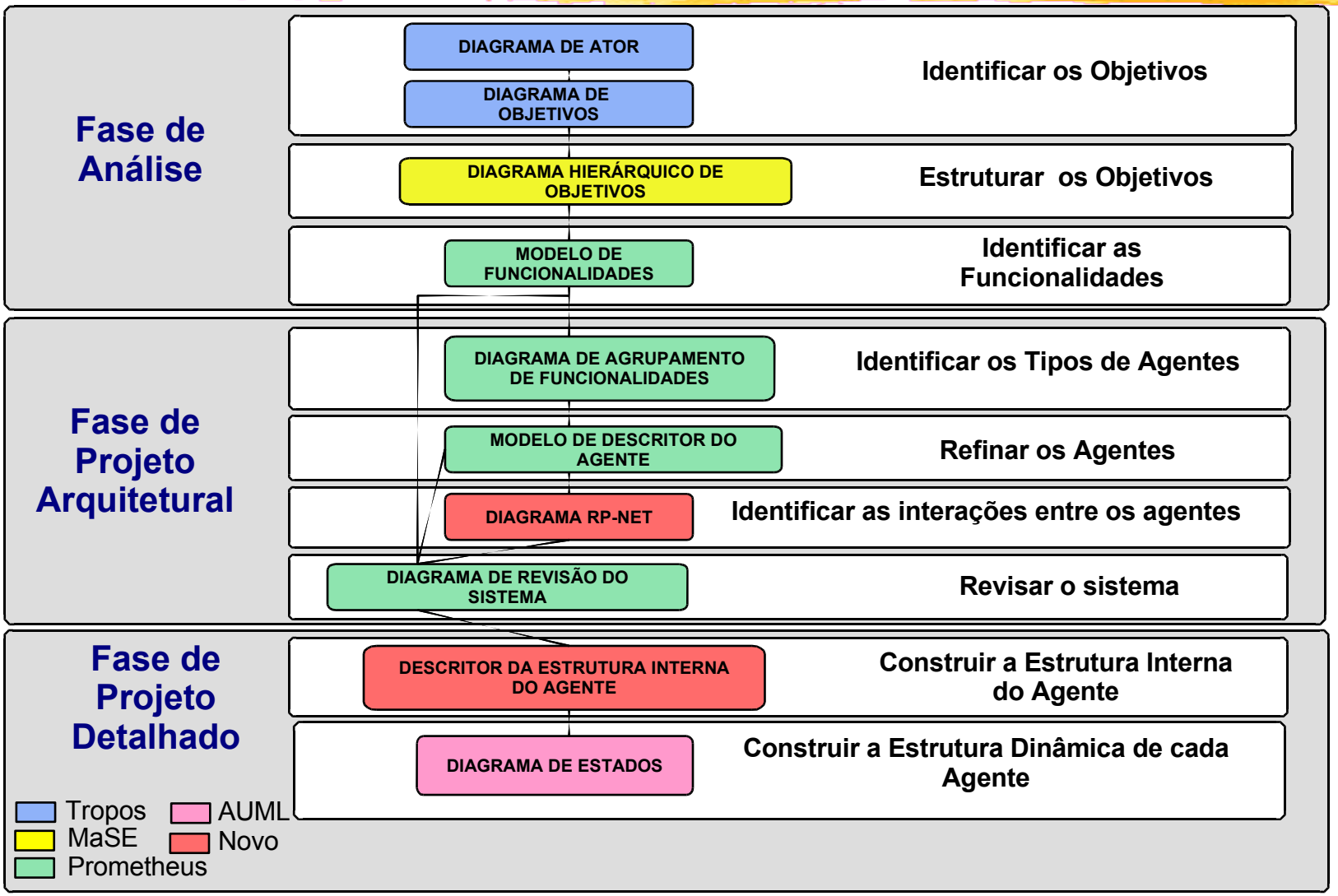
Metodologia Tropos

- Projeto Arquitetural:
  - define a arquitetura global do sistema em termos de sub-sistemas (atores), interconectados através de fluxos de controle de dados (dependências de atores) → Diagrama de Ator Estendido.
    - São identificadas as capacidades necessárias para os atores realizarem seus objetivos (Tabela de Capacidades).
    - Define-se um conjunto de agentes, atribuindo as capacidades aos agentes específicos (Tabela de Tipos de Agentes).
- Projeto Detalhado:
  - nível micro do agente, detalhando as capacidades (Diagrama de Capacidades), os planos (Diagrama de Planos), bem como a interação do agente (Diagrama de Interação).
- Implementação:
  - mapeamento entre os conceitos do Tropos e os elementos da plataforma de interação - Jack Intelligent Agents (Java)

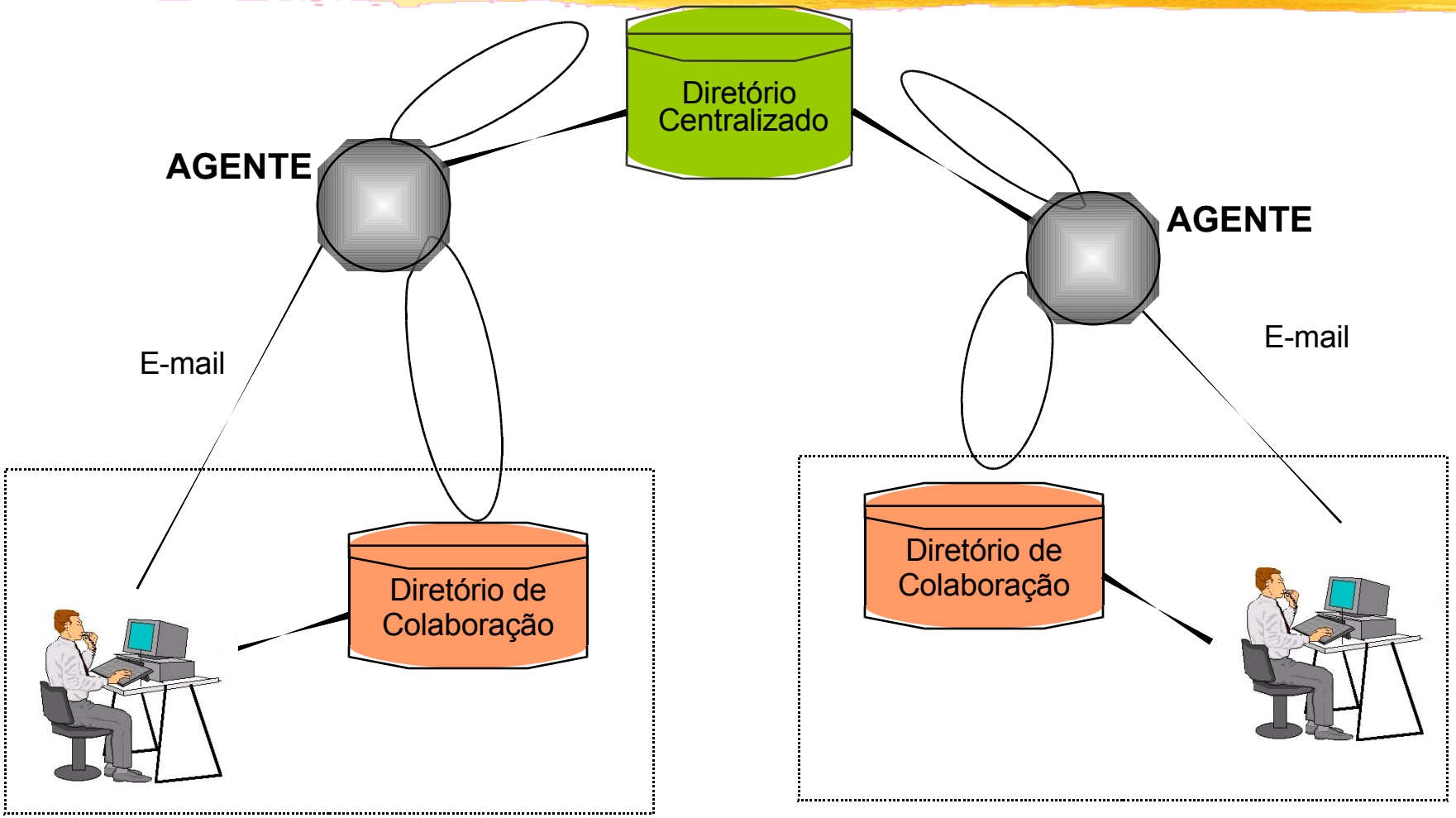
Metodologia Tropos



Metodologia Unificada



Estudo de Caso



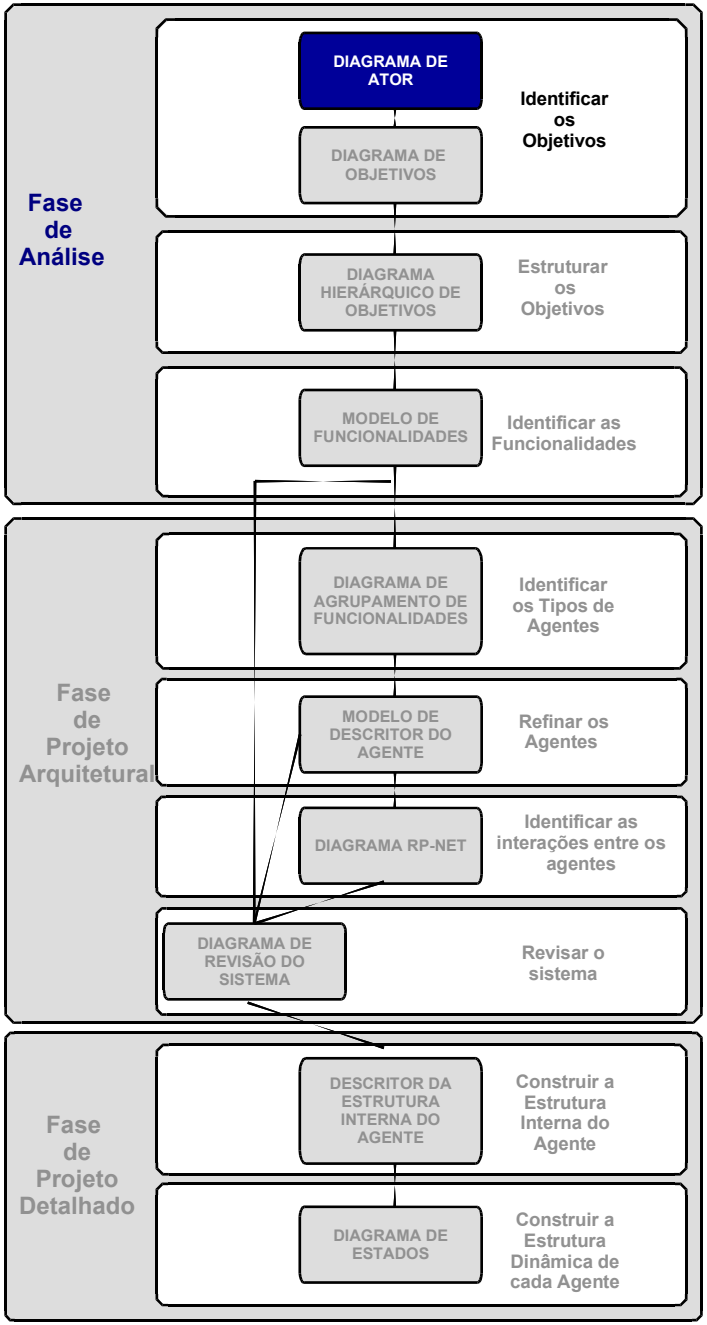
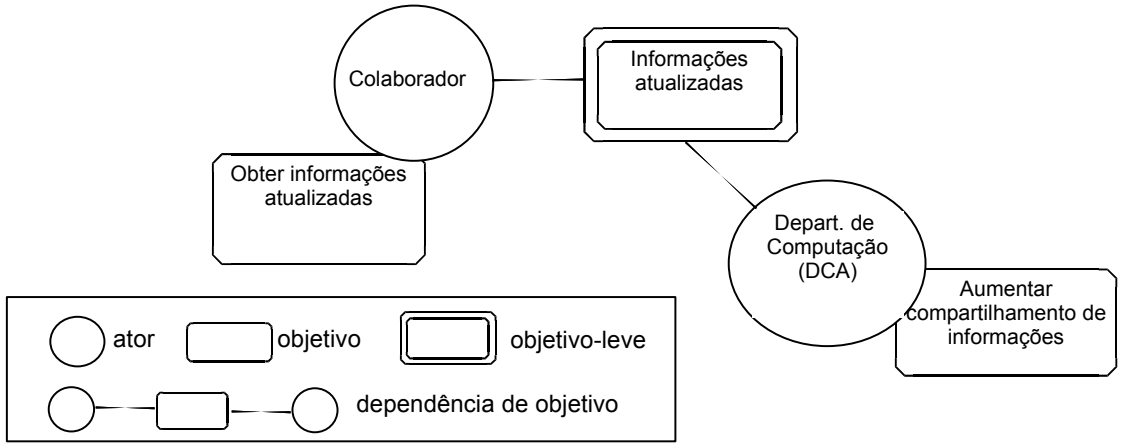


DIAGRAMA DE ATOR



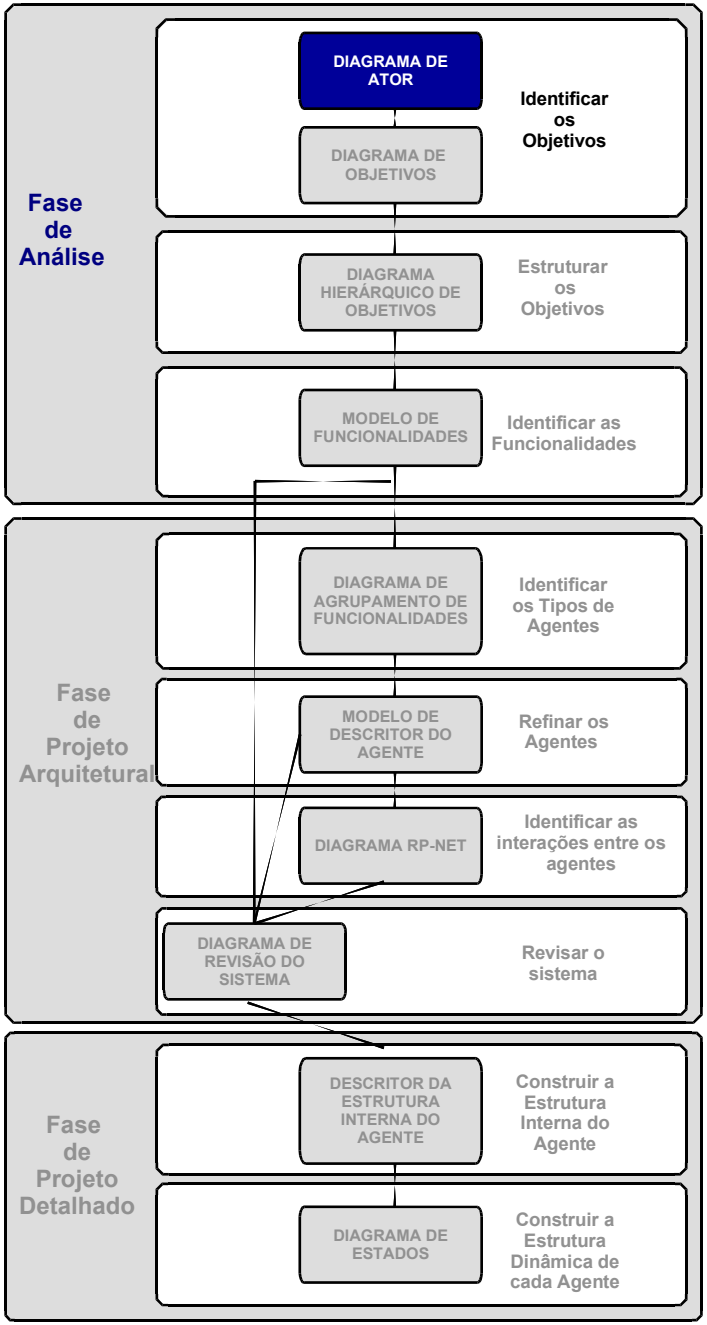
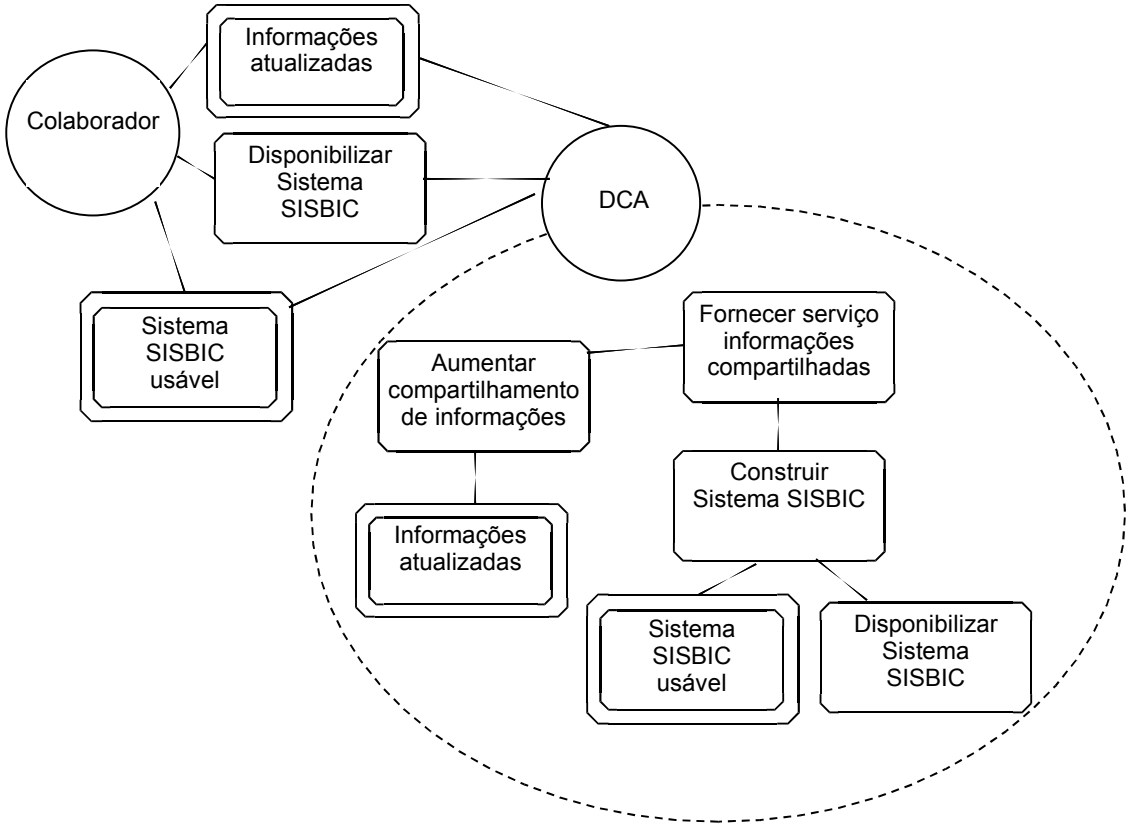
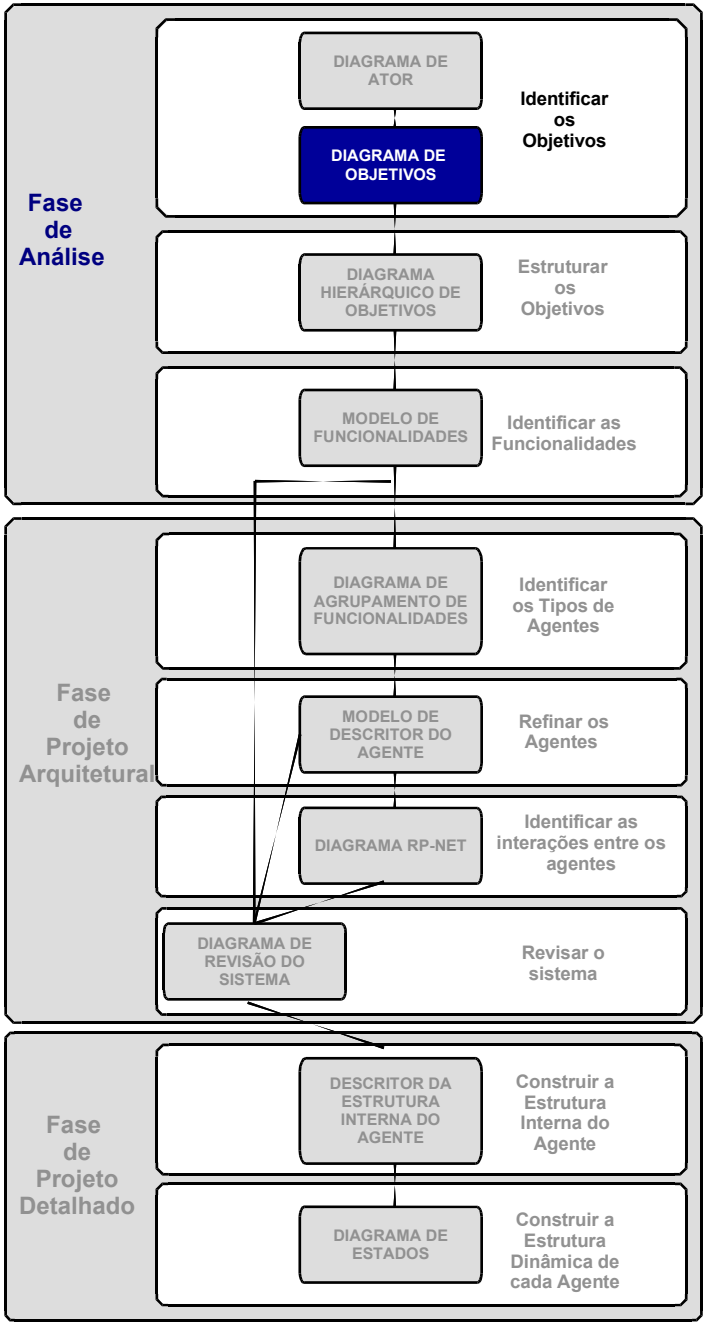


DIAGRAMA DE ATOR




Fase de Análise
DIAGRAMA DE ATOR
DIAGRAMA DE OBJETIVOS
Identificar os Objetivos
DIAGRAMA HIERÁRQUICO DE OBJETIVOS
Estruturar os Objetivos
MODELO DE FUNCIONALIDADES
Identificar as Funcionalidades
Fase de Projeto Arquitetural
DIAGRAMA DE AGRUPAMENTO DE FUNCIONALIDADES
Identificar os Tipos de Agentes
MODELO DE DESCRITOR DO AGENTE
Refinar os Agentes
DIAGRAMA RP-NET
Identificar as interações entre os agentes
DIAGRAMA DE REVISÃO DO SISTEMA
Revisar o sistema
Fase de Projeto Detalhado
DESCRITOR DA ESTRUTURA INTERNA DO AGENTE
Construir a Estrutura Interna do Agente
DIAGRAMA DE ESTADOS
Construir a Estrutura Dinâmica de cada Agente

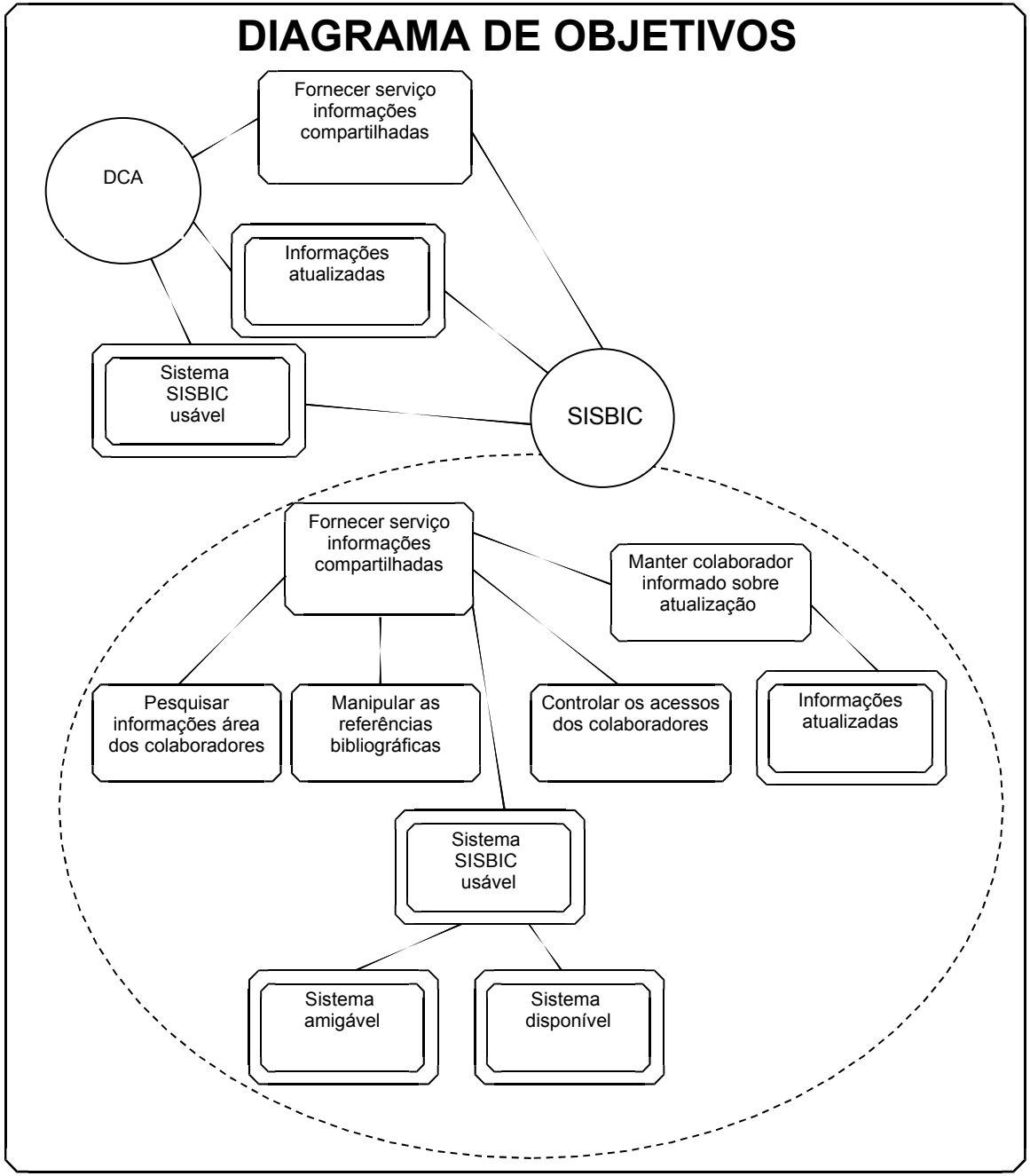


DIAGRAMA DE OBJETIVOS
Fornecer serviço informações compartilhadas
DCA
Informações atualizadas
Sistema SISBIC usável
SISBIC
Fornecer serviço informações compartilhadas
Manter colaborador informado sobre atualização
Pesquisar informações área dos colaboradores
Manipular as referências bibliográficas
Controlar os acessos dos colaboradores
Informações atualizadas
Sistema SISBIC usável
Sistema amigável
Sistema disponível

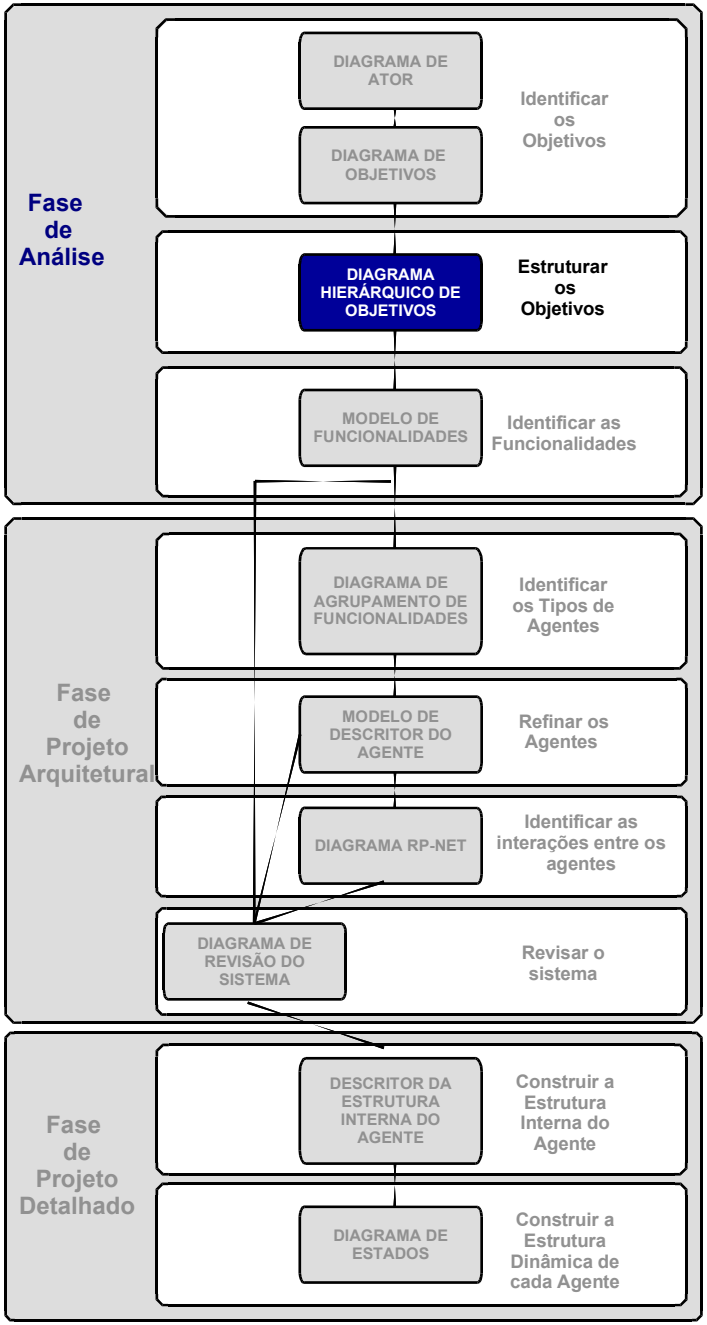
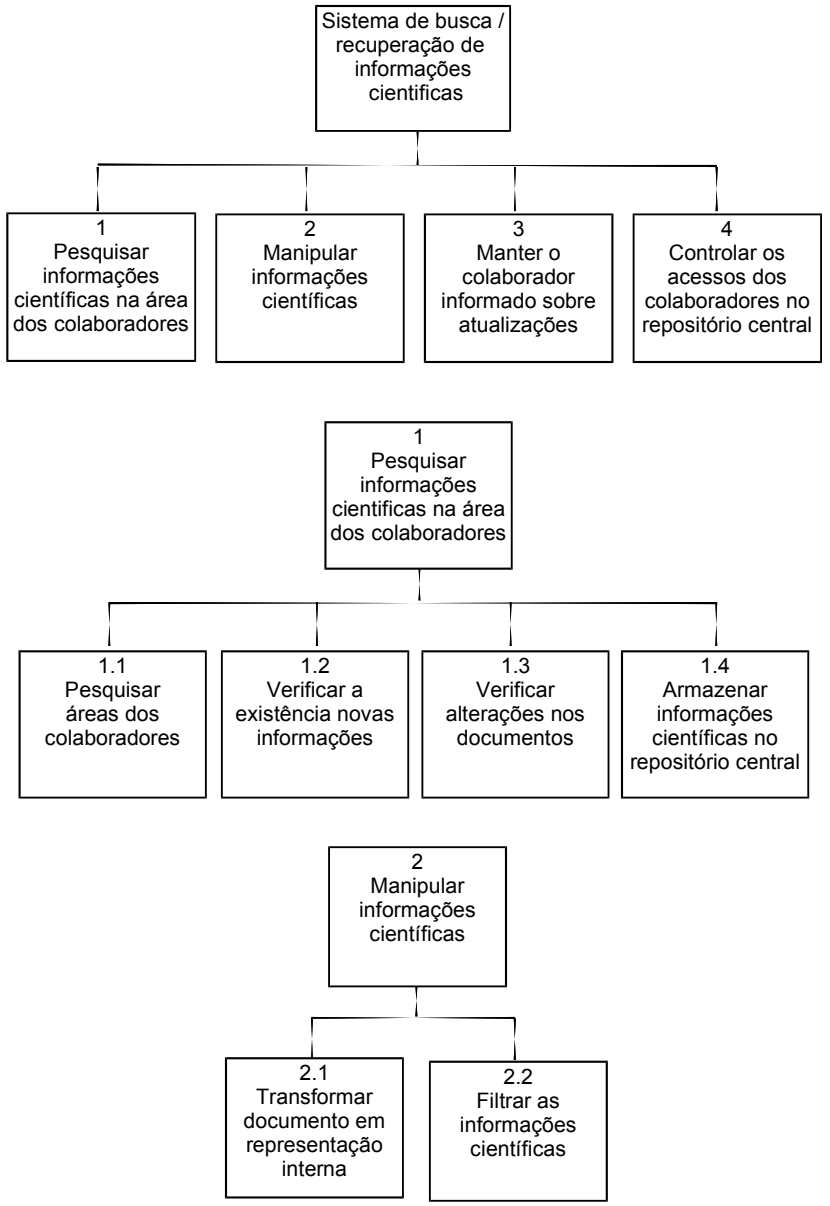
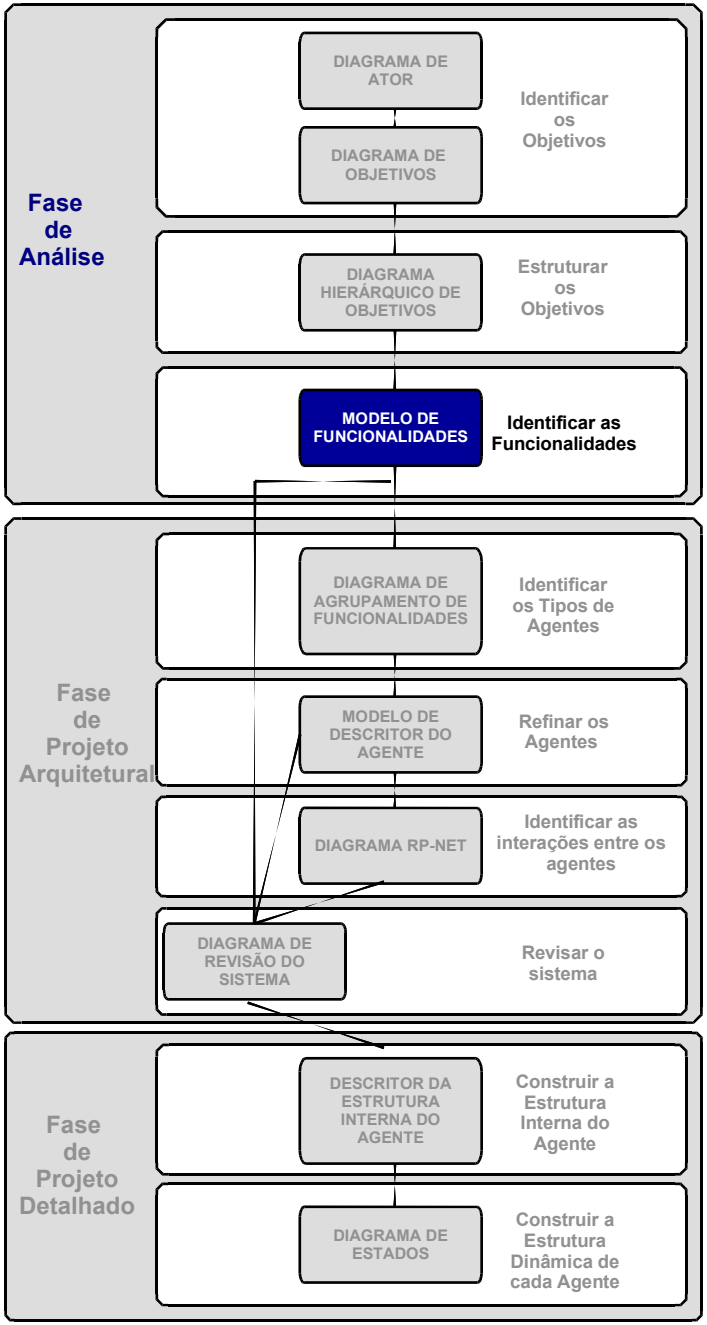


DIAGRAMA HIERÁRQUICO DE OBJETIVOS





MODELO DE FUNCIONALIDADES

Nome: Descobridor

Descrição: Pesquisar informações científicas nas áreas dos colaboradores

Objetivos:
Pesquisar áreas dos colaboradores
Verificar a existência de novas informações científicas
Verificar as alterações nos documentos armazenados no repositório central
Armazenar as informações científicas, coletadas das áreas dos colaboradores, no repositório central

Percepções:
Usuário grava novas informações científicas no seu repositório

Ações:
Coletar as informações científicas no repositório de cada colaborador
Armazenar as informações cientificas coletadas no repositório central

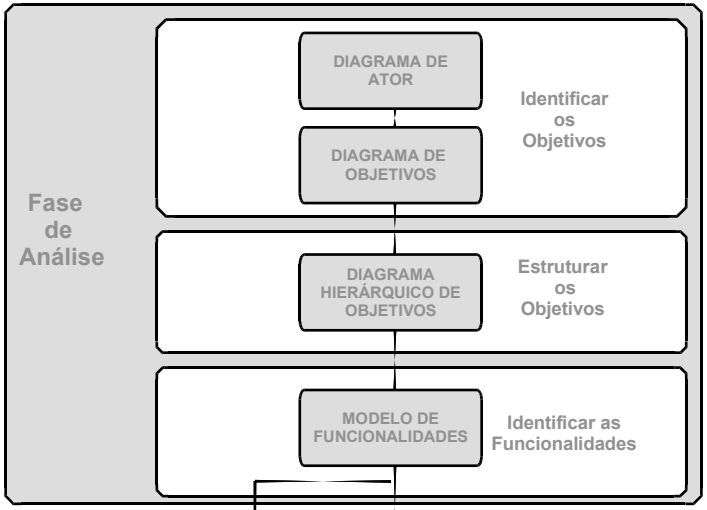
Mensagens Coletadas:
InformaPerfil_Descobridor

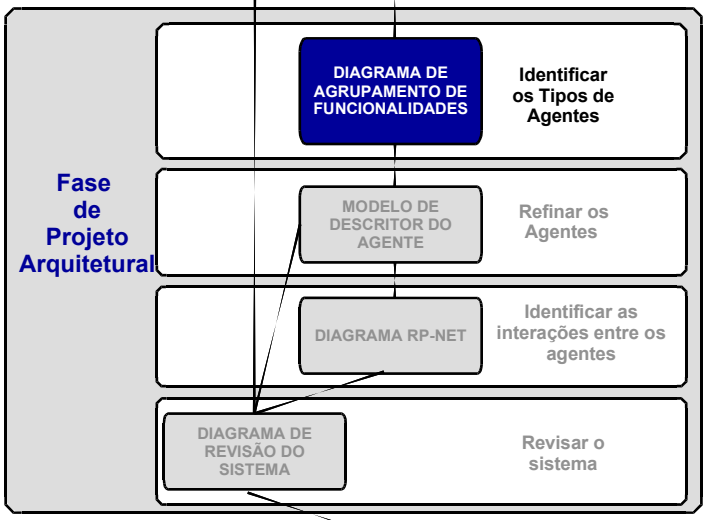
Mensagens Enviadas:
InformaDescobridor_Classificador

Intersecções:
buffer_Monitor_Perfil_Descobridor, buffer_Descobridor_Classificador

Dados lidos:
referencias_bibli_colaborador_db

Dados gerados:
referencias_bibli_db





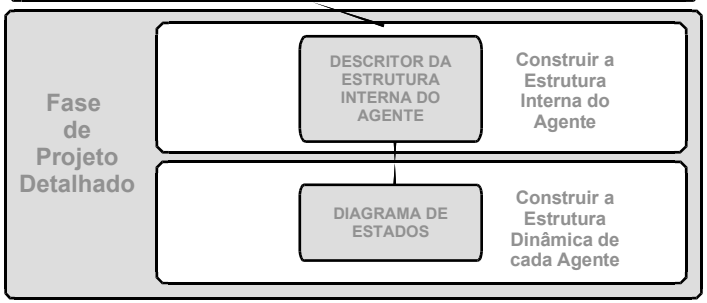
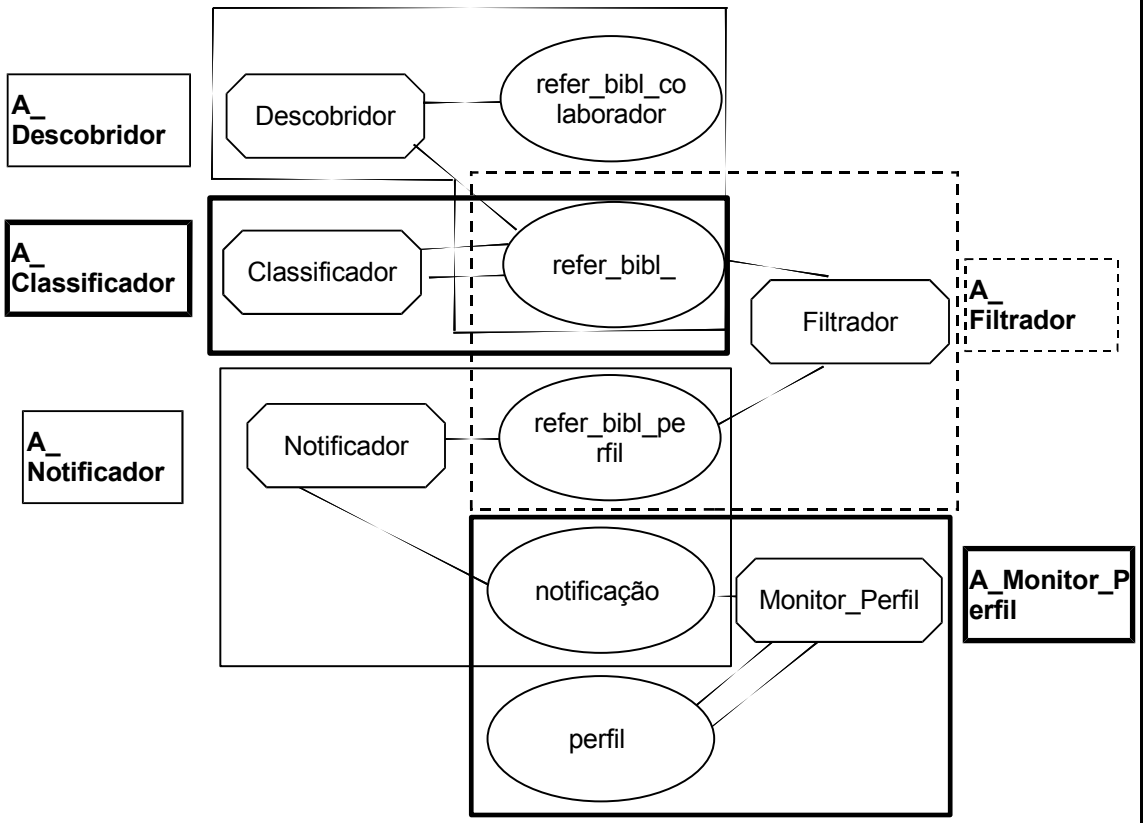
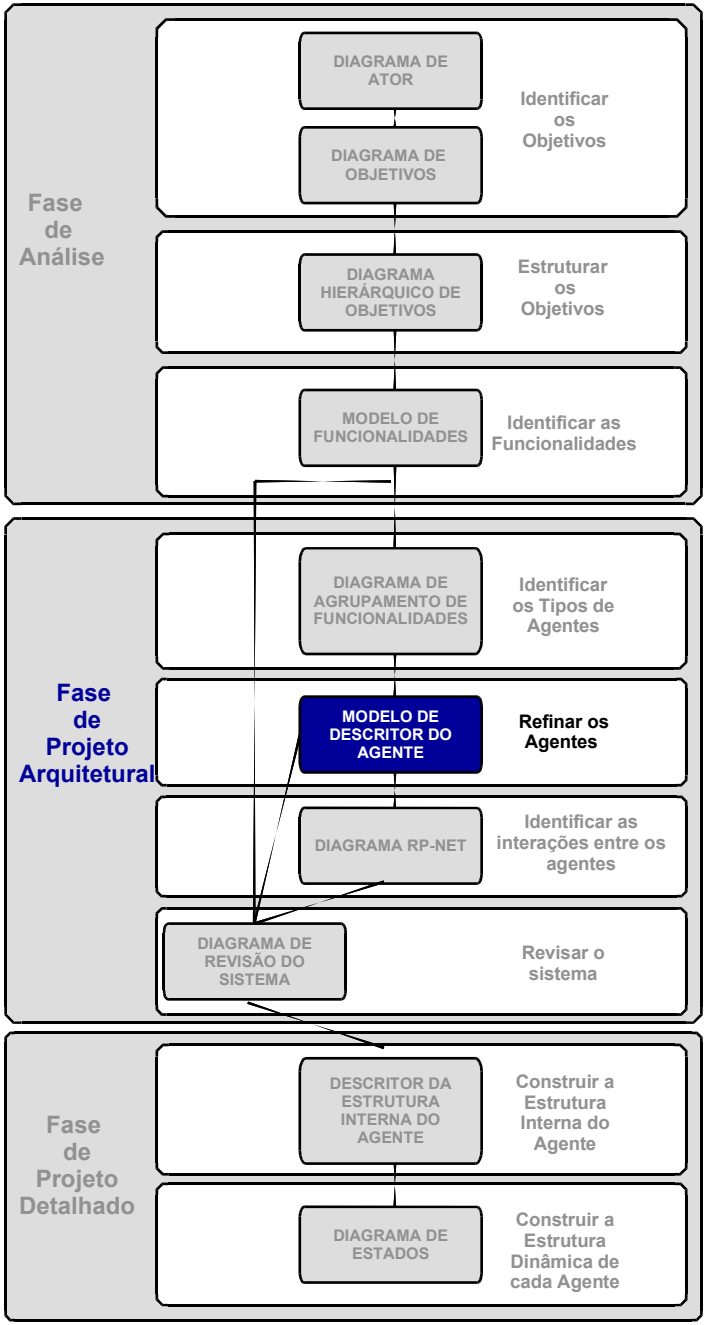


DIAGRAMA DE AGRUPAMENTO DE FUNCIONALIDADES





MODELO DE DESCRITOR DO AGENTE

Nome: A_Descobridor
Descrição: Busca referência bibliográfica da área de cada colaborador
Funcionalidades: Descobridor
Instâncias: n, onde n é igual ao número de colaboradores.
Interações: buffer_Monitor_Perfil_Descobridor (buffer 6), buffer_Descobridor_Classificador (buffer 1)
Mensagens coletadas: InformaPerfil-Desc
Mensagens produzidas: InformaDescobridor-Class
Inicialização do agente: colaborador
Destruição do agente: colaborador
Dados lidos: referencias_bibli_colaborador_db
Dados gravados: referencias_bibli_db

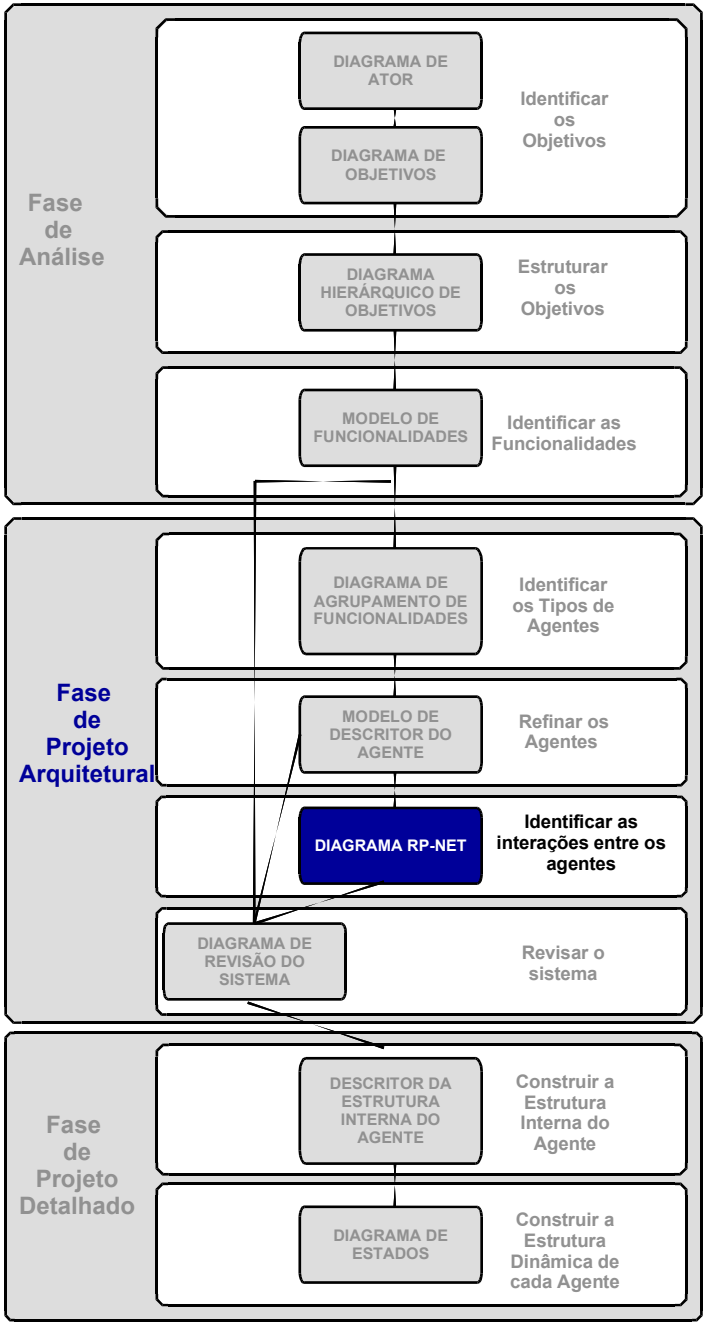
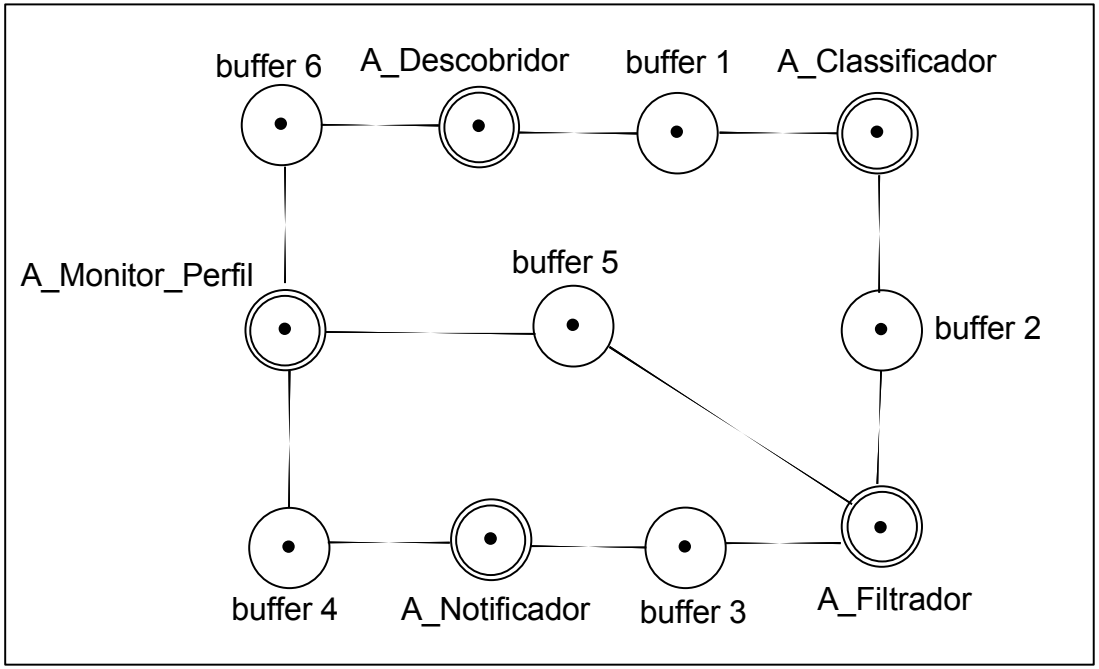
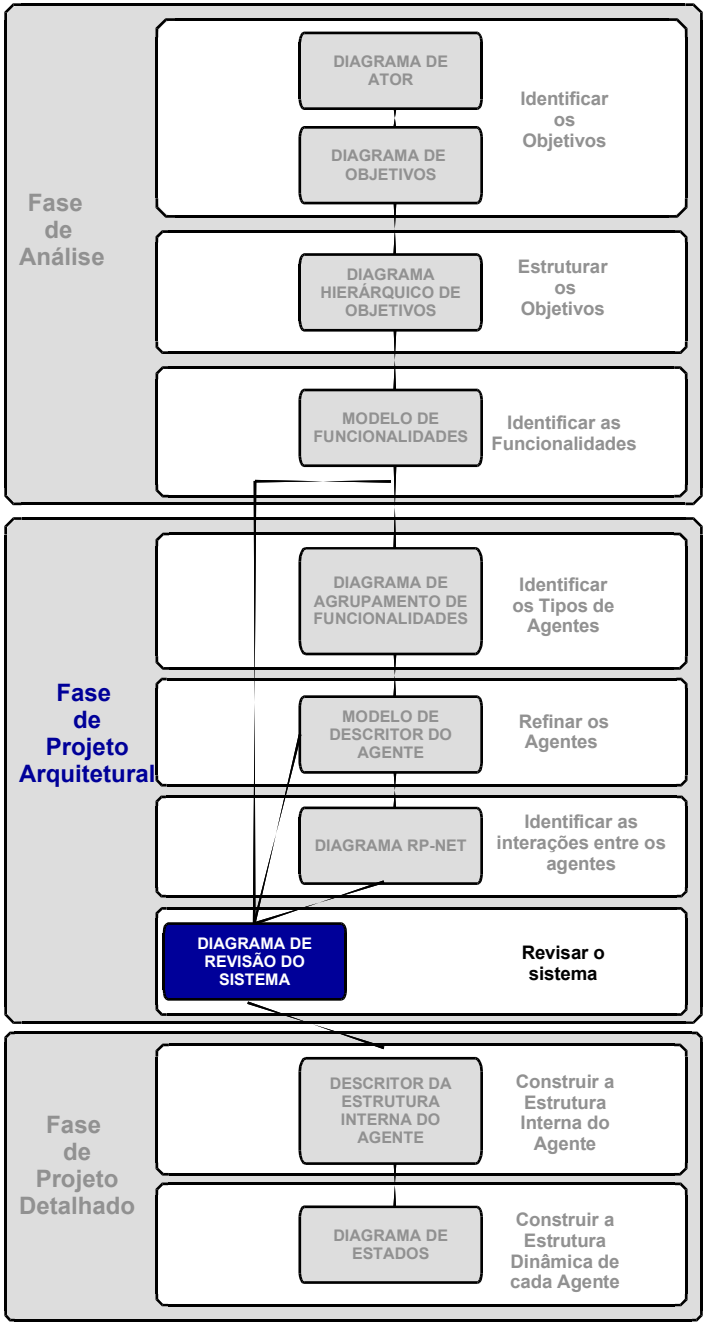


DIAGRAMA RP-NET



Buffer	Descrição
buffer 1	Buffer_Descobridor_Classificador
buffer 2	Buffer_Classificador_Filtrador
buffer 3	Buffer_Filtrador_Notificador
buffer 4	Buffer_Notificador_Monitor_Perfil
buffer 5	Buffer_Monitor_Perfil_Filtrador
buffer 6	Buffer_Monitor_Perfil_Descobridor


Fase de Análise
DIAGRAMA DE ATOR
DIAGRAMA DE OBJETIVOS
Identificar os Objetivos
DIAGRAMA HIERÁRQUICO DE OBJETIVOS
Estruturar os Objetivos
MODELO DE FUNCIONALIDADES
Identificar as Funcionalidades
Fase de Projeto Arquitetural
DIAGRAMA DE AGRUPAMENTO DE FUNCIONALIDADES
Identificar os Tipos de Agentes
MODELO DE DESCRITOR DO AGENTE
Refinar os Agentes
DIAGRAMA RP-NET
Identificar as interações entre os agentes
DIAGRAMA DE REVISÃO DO SISTEMA
Revisar o sistema
Fase de Projeto Detalhado
DESCRITOR DA ESTRUTURA INTERNA DO AGENTE
Construir a Estrutura Interna do Agente
DIAGRAMA DE ESTADOS
Construir a Estrutura Dinâmica de cada Agente

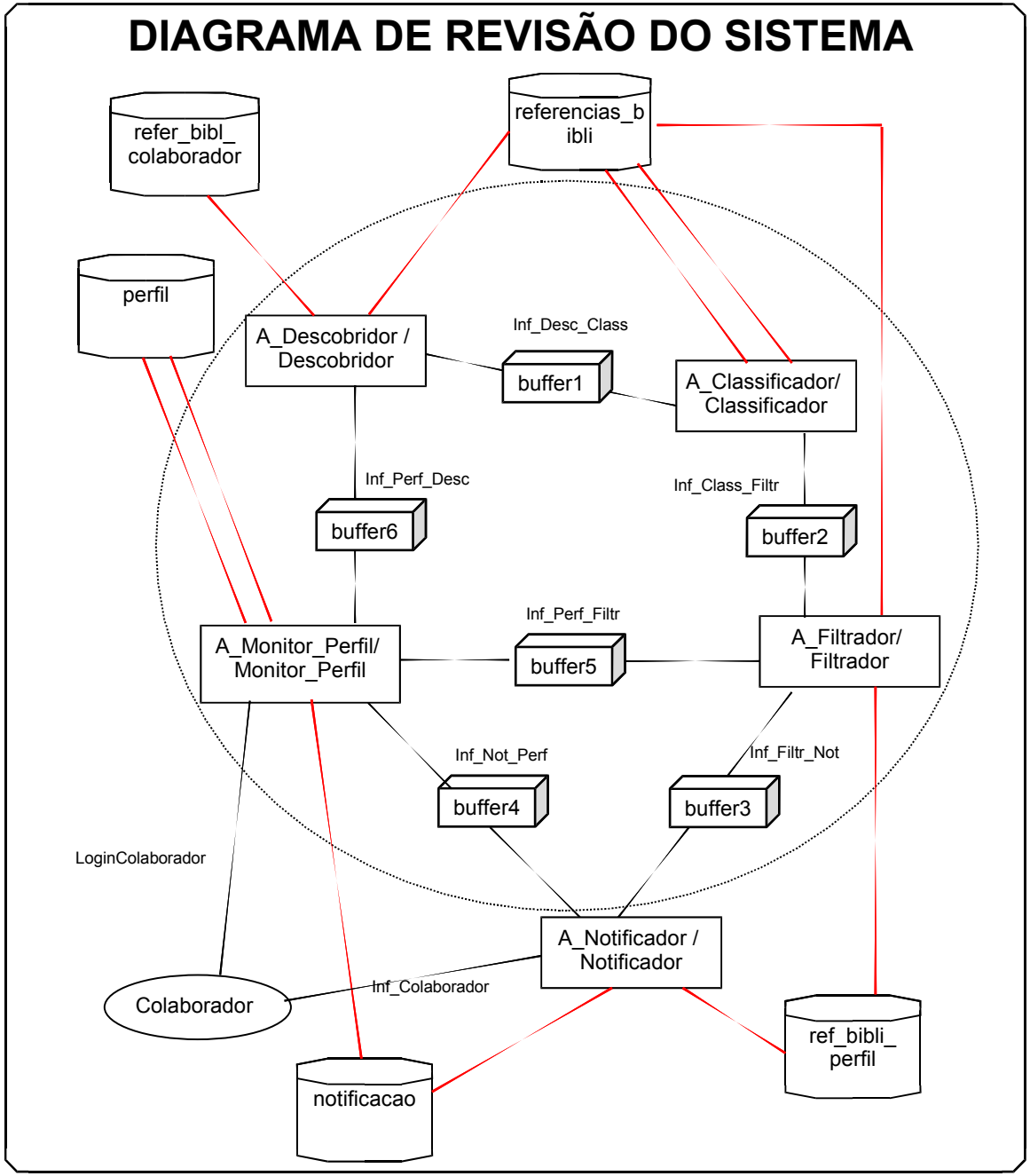
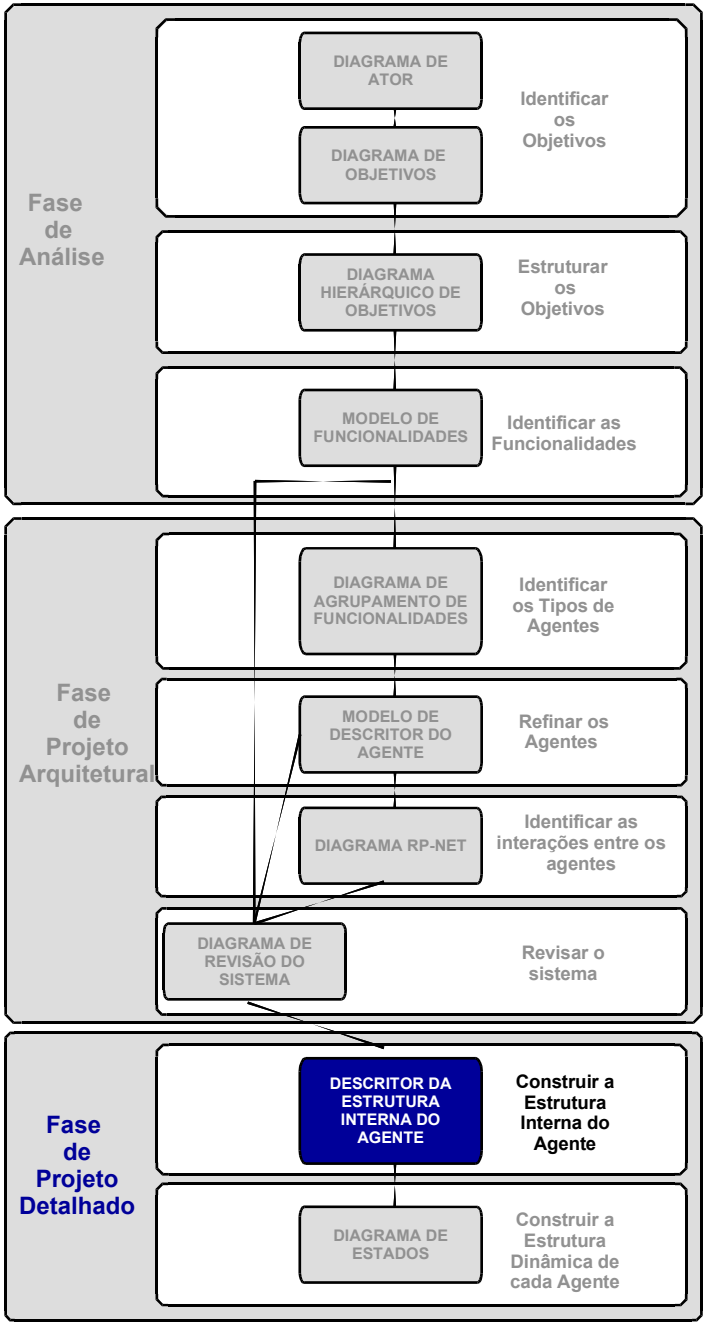


DIAGRAMA DE REVISÃO DO SISTEMA
refer_bibl_ colaborador
referencias_b ibli
perfil
A_Descobridor / Descobridor
Inf_Desc_Class
buffer1
A_Classificador/ Classificador
Inf_Perf_Desc
buffer6
Inf_Class_Filtr
buffer2
A_Monitor_Perfil/ Monitor_Perfil
Inf_Perf_Filtr
buffer5
A_Filtrador/ Filtrador
Inf_Not_Perf
buffer4
Inf_Filtr_Not
buffer3
LoginColaborador
A_Notificador / Notificador
Colaborador
Inf_Colaborador
ref_bibli_ perfil
notificacao



DESCRITOR DA ESTRUTURA INTERNA DO AGENTE

Nome: A_Descobridor

Planos envolvidos:

01 – Ler o repositório específico de cada colaborador

02 – Gravar no repositório central as informações científicas lidas
no repositório específico de cada colaborador

03 – Verificar se a referência já não foi gravada no repositório central

***Contingência*:**

no. plano: 01

causa da falha: colaborador não está logado / acessível

reação: tentar novamente em intervalos de tempo

***Contingência*:**

no. plano: 02 e 03

causa da falha: repositório central não está disponível (desligado / sobrecarregado)

reação: informar colaborador / administrador sobre indisponibilidade do servidor e tentar novamente

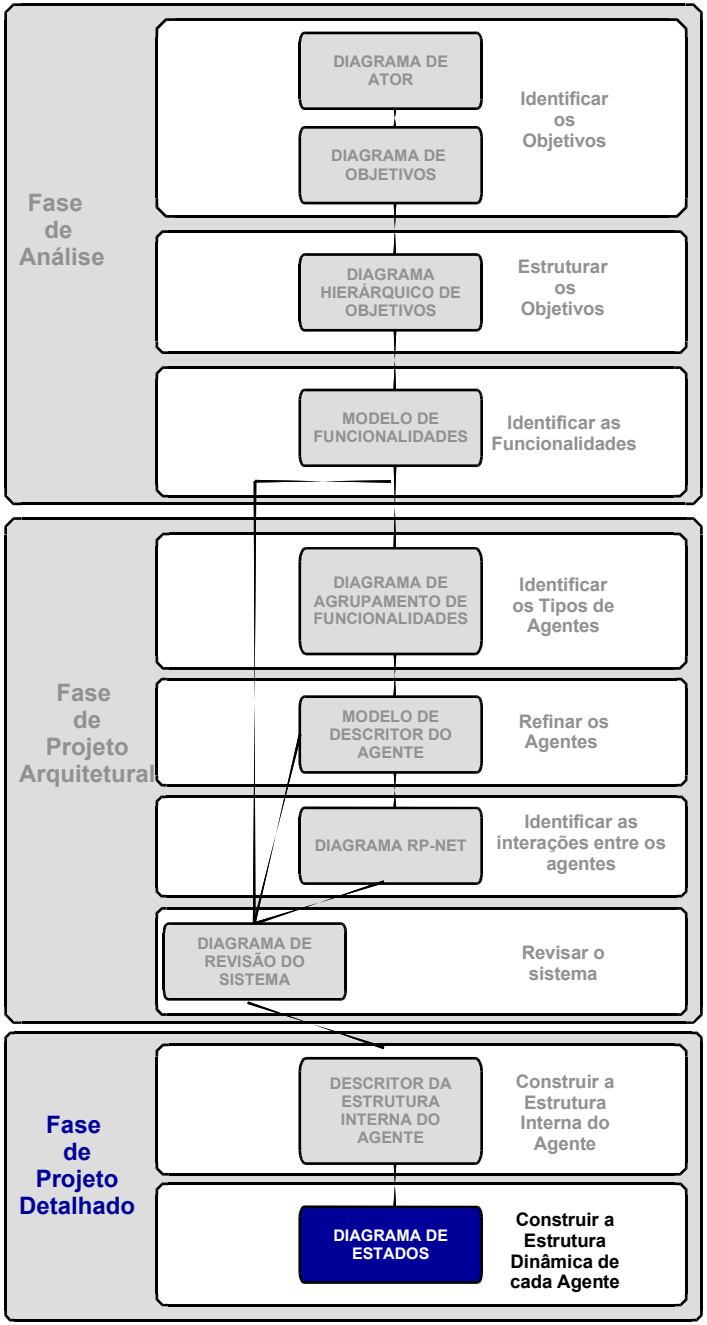


DIAGRAMA DE ESTADOS