

SCMV Nemosine Nous ©

Technical Report

Mapeamento da Engenharia de Software Aplicada a um Sistema Cognitivo Modular Vivo

Autor:

Edervaldo José de Souza Melo

Registro de Programa de Computador (INPI – Brasil):

BR512025003335-4

Tipo de Documento:

Documento Técnico de Engenharia de Software (Technical Report)

Versão:

v0.1

Status:

Documento Técnico (Technical Document)

Data:

Dezembro de 2025

Resumo

Este Relatório Técnico apresenta a concepção estrutural, conceitual e técnica do SCMV Nemosine Nous ©, um sistema cognitivo modular destinado a apoiar a reflexão individual, a organização do pensamento e a modelagem cognitiva pessoal. O documento descreve o escopo do sistema, seu ciclo de vida, princípios arquiteturais, limites de governança e diretrizes de implementação, com ênfase em modularidade, responsabilidade ética e evolução controlada.

Em vez de definir uma implementação fechada ou executável, este relatório estabelece um arcabouço técnico de referência, documentando decisões de projeto, limitações, riscos e exclusões deliberadas, preservando flexibilidade para desenvolvimento futuro. O SCMV é explicitamente caracterizado como um sistema não social, não clínico e não prescritivo, operando como um ambiente cognitivo instrumental, e não como autoridade decisória.

Este relatório constitui um artefato fundacional para orientar o desenvolvimento contínuo, processos de validação e atividades de pesquisa relacionadas ao sistema Nemosine.

Palavras-chave: Sistema Cognitivo Modular; Arquitetura de Software; Modelagem Cognitiva; Engenharia de Software; Governança de IA; Sistemas Baseados em IA; Projeto de Sistemas; Privacidade e Ética em IA

Abstract

This Technical Report presents the structural, conceptual, and technical design of the *SCMV Nemosine Nous* ©, a modular cognitive system intended to support individual reflection, organization, and cognitive modeling. The document describes the system's scope, lifecycle, architectural principles, governance boundaries, and implementation guidelines, emphasizing modularity, ethical responsibility, and controlled evolution.

Rather than defining a closed or executable implementation, this report establishes a technical reference framework that documents design decisions, limitations, risks, and deliberate exclusions, while preserving flexibility for future development. The SCMV is explicitly positioned as a non-social, non-clinical, and non-prescriptive system, operating as an instrumental cognitive environment rather than a decision-making authority.

This report serves as a foundational artifact to guide ongoing development, validation, and research activities related to the Nemosine system.

Keywords: Modular Cognitive System; Software Architecture; Cognitive Modeling; Software Engineering; AI Governance; AI-Based Systems; System Design; Privacy and Ethics in AI

Aviso Legal e de Propriedade Intelectual

Este documento descreve, de forma técnica e metodológica, o mapeamento da Engenharia de Software aplicada ao SCMV Nemosine Nous © (Sistema Cognitivo Modular Vivo), sistema registrado como programa de computador junto ao Instituto Nacional da Propriedade Industrial (INPI), sob o número BR512025003335-4.

O presente Technical Report possui caráter descritivo, metodológico e arquitetural, tendo como objetivo documentar decisões de engenharia, ciclos de vida, práticas de projeto, implementação e validação adotadas no contexto específico do SCMV Nemosine Nous ©.

A publicação deste documento não implica renúncia de direitos autorais, patrimoniais ou morais, nem constitui cessão automática de direitos sobre eventuais implementações derivadas, produtos comerciais, serviços, aplicações institucionais ou módulos proprietários associados ao sistema.

Salvo indicação explícita em contrário, este documento pode ser disponibilizado em regime de open access, respeitando-se o modelo de open core, no qual a documentação metodológica é pública, enquanto implementações completas, aplicações comerciais e extensões estratégicas podem estar sujeitas a licenciamento específico.

Nota de Escopo Editorial

Este documento não é:

1. um artigo científico tradicional (paper) destinado a periódicos indexados;
2. um whitepaper de caráter mercadológico ou promocional;
3. um manual de usuário final;
4. um framework genérico ou universal de engenharia de software.

Este documento é:

1. um Technical Report;
2. um artefato de engenharia aplicada;

3. um registro estruturado do mapeamento do ciclo de vida de software, conforme práticas clássicas e contemporâneas da Engenharia de Software;
4. uma documentação incremental e versionada das decisões técnicas, modelagens, ferramentas, fluxos e critérios de validação adotados no desenvolvimento e evolução do SCMV Nemosine Nous ©.

O conteúdo aqui apresentado reflete o estado do sistema no momento da versão indicada, estando sujeito a revisões, expansões e refinamentos em versões posteriores, de acordo com a evolução técnica, experimental e operacional do projeto.

Convenções Editoriais

1. Linguagem técnica, objetiva e não promocional;
 2. Ênfase em clareza, rastreabilidade e justificativa de decisões;
 3. Estrutura orientada ao ciclo de vida do software: Especificação → Projeto → Implementação → Testes → Manutenção;
 4. Versionamento explícito do documento;
 5. Alinhamento com práticas aceitas de Engenharia de Software, sem dependência de formalismos acadêmicos estritos.
-

1. Introdução

O **SCMV Nemosine Nous © (Sistema Cognitivo Modular Vivo)** é um sistema cognitivo modular concebido para apoiar processos de reflexão, organização mental, tomada de decisão e mediação simbólica por meio de interações estruturadas entre módulos conceituais, funcionais e narrativos.

Diferentemente de aplicações tradicionais de software orientadas exclusivamente à automação de tarefas ou ao processamento de dados objetivos, o SCMV opera em um **território sensível**, no qual aspectos cognitivos, emocionais e simbólicos do usuário fazem parte do contexto de uso do sistema. Essa característica impõe **exigências adicionais de engenharia**,

especialmente no que se refere à segurança, à previsibilidade de comportamento, à rastreabilidade de decisões de projeto e à mitigação de riscos associados ao uso indevido, excessivo ou inadequado do sistema.

Este Technical Report tem como objetivo **mapear de forma explícita a aplicação das práticas de Engenharia de Software ao SCMV Nemosine Nous ©**, documentando:

1. o entendimento da dor e do contexto do usuário (especificação);
2. as decisões de arquitetura e modelagem do sistema (projeto);
3. as estratégias de implementação e organização técnica (implementação);
4. os critérios de validação e verificação do comportamento do sistema (testes);
5. e os mecanismos de evolução e manutenção ao longo do tempo.

A motivação central deste documento não é apenas registrar o *como* o sistema é construído, mas **tornar auditável o porquê das decisões técnicas adotadas**, especialmente em um cenário no qual sistemas cognitivos assistidos por software podem produzir efeitos não triviais sobre o comportamento humano.

Nesse contexto, o SCMV Nemosine Nous © assume, desde sua concepção, o compromisso de **não operar como sistema de aconselhamento clínico, terapêutico ou médico**, tampouco como substituto de acompanhamento profissional. Ainda assim, reconhece-se que sistemas simbólico-cognitivos podem exercer **influência indireta** sobre estados emocionais e processos decisórios do usuário, o que exige uma abordagem de engenharia orientada à **prevenção de danos, à detecção de estados de risco e à definição clara de limites funcionais**.

Como diretriz estratégica de médio e longo prazo, o desenvolvimento e a evolução do SCMV Nemosine Nous © consideram o **alinhamento progressivo com normas internacionais de governança de Inteligência Artificial**, em especial a **ISO/IEC 42001:2023**, que estabelece requisitos para Sistemas de Gestão de Inteligência Artificial (AIMS).

Tal alinhamento visa orientar decisões de projeto, documentação, governança e mitigação de riscos associados ao uso de sistemas de IA, especialmente no que se refere a **impactos humanos, cognitivos e**

comportamentais, sem implicar, neste estágio, certificação formal ou conformidade plena, mas assegurando **consistência conceitual e rastreabilidade técnica desde as fases iniciais do ciclo de vida do software**.

Por fim, este documento é estruturado como um **artefato de engenharia vivo**, atualizado incrementalmente à medida que o sistema evolui. Seu papel é servir simultaneamente como:

1. registro técnico interno do projeto;
 2. instrumento de comunicação com colaboradores, avaliadores e interessados técnicos;
 3. base de rastreabilidade para análises futuras de conformidade, segurança e impacto.
-

2. Contexto de Uso e Demanda do Usuário (Etapa de Especificação e Levantamento de Necessidades)

O SCMV Nemosine Nous © é concebido para atender usuários que buscam **apoio cognitivo estruturado**, especialmente em contextos caracterizados por alta carga mental, complexidade decisória, acúmulo de informações e necessidade de organização simbólica do pensamento.

O sistema não se destina a um perfil genérico de usuário, mas a indivíduos que, por diferentes motivos, enfrentam **dificuldades em estruturar, priorizar ou navegar seus próprios processos mentais**, seja em atividades profissionais, acadêmicas, criativas ou pessoais. Essas dificuldades não são, necessariamente, patológicas, mas frequentemente decorrem de ambientes informacionais saturados, pressões externas contínuas e ausência de instrumentos adequados de organização cognitiva.

2.1 Caracterização do Contexto de Uso

O contexto de uso do SCMV pode ser descrito como **sociotécnico**, no qual fatores humanos, simbólicos e tecnológicos coexistem e se influenciam mutuamente. O sistema opera em um espaço intermediário entre:

1. ferramentas tradicionais de produtividade;
2. sistemas de apoio à tomada de decisão;
3. ambientes de reflexão assistida e modelagem conceitual.

Nesse contexto, o usuário interage com o sistema de forma **iterativa e prolongada**, utilizando-o não apenas para executar tarefas pontuais, mas como **ambiente recorrente de organização e reflexão**. Essa característica distingue o SCMV de aplicações utilitárias de uso episódico e impõe requisitos adicionais de previsibilidade, estabilidade e clareza de escopo funcional.

Além do contexto de uso imediato, reconhece-se que sistemas cognitivos assistidos por software podem assumir, ao longo do tempo, um papel **longitudinal** na organização do pensamento do usuário, acompanhando ciclos extensos de reflexão, decisão e reconstrução conceitual.

Embora o escopo do SCMV Nemosine Nous ©, nesta fase, esteja deliberadamente limitado a funções de apoio cognitivo estruturado, sua concepção modular e evolutiva admite, como hipótese futura, a investigação de **modelos de interação prolongada**, nos quais aspectos como continuidade contextual, memória estrutural e rastreabilidade histórica possam ser analisados sob uma perspectiva técnica, ética e de governança apropriada.

Tais investigações, quando e se conduzidas, constituirão objeto de documentação específica e independente, não fazendo parte do escopo funcional nem das responsabilidades operacionais do sistema descrito neste documento.

2.2 Dores e Problemas Identificados

A especificação do SCMV parte da identificação de dores recorrentes observadas no contexto de uso pretendido, dentre as quais destacam-se:

- dificuldade em organizar pensamentos complexos de forma estruturada;
- sobrecarga cognitiva decorrente do acúmulo de informações desconexas;

- perda de rastreabilidade de decisões, ideias e processos mentais ao longo do tempo;
- sensação de dispersão, fragmentação ou estagnação cognitiva;
- ausência de instrumentos que permitam externalizar, visualizar e reorganizar estruturas mentais de maneira sistemática.

Essas dores não se apresentam de forma uniforme em todos os usuários, mas compõem um **espectro de necessidades** que orienta a concepção modular do sistema.

2.3 Limites Deliberados de Atuação do Sistema

Como parte fundamental da especificação, o SCMV Nemosine Nous © define explicitamente **limites funcionais e conceituais** de atuação. O sistema:

- não realiza diagnóstico clínico;
- não fornece aconselhamento médico, psicológico ou terapêutico;
- não substitui acompanhamento profissional especializado;
- não opera como sistema de avaliação de risco individual.

Esses limites são tratados como **requisitos não funcionais críticos**, orientando decisões de projeto, linguagem de interação, escopo de respostas e mecanismos de contenção simbólica do sistema.

2.4 Considerações de Risco e Responsabilidade

Reconhece-se que sistemas cognitivos assistidos por software podem exercer **influência indireta** sobre processos emocionais e decisórios do usuário, especialmente quando utilizados de forma intensiva ou prolongada. Por esse motivo, a especificação do SCMV incorpora, desde as fases iniciais, preocupações relacionadas a:

- prevenção de dependência funcional excessiva;
- mitigação de interpretações indevidas do papel do sistema;
- clareza contínua sobre as capacidades e limitações da ferramenta;

- rastreabilidade de decisões de projeto relacionadas à segurança e governança.

Essas considerações servem como base para decisões posteriores nas fases de **projeto, implementação e testes**, bem como para o alinhamento progressivo com práticas de governança de IA, conforme diretrizes internacionais relevantes.

3. Visão Geral do Ciclo de Vida do Software

O ciclo de vida do software do **SCMV Nemosine Nous** © é concebido de forma **evolutiva e incremental**, reconhecendo que sistemas cognitivos assistidos por IA não se beneficiam de abordagens rígidas ou estritamente sequenciais de desenvolvimento.

Diferentemente de aplicações com requisitos totalmente estáveis e bem delimitados desde a fase inicial, o SCMV opera em um domínio no qual **hipóteses de uso, limites funcionais e necessidades do usuário são progressivamente refinados** à medida que o sistema é utilizado, observado e analisado. Essa característica torna inadequada a adoção de modelos de ciclo de vida puramente lineares, como o modelo em cascata clássico.

3.1 Abordagem de Ciclo de Vida Adotada

O SCMV adota uma abordagem de ciclo de vida **incremental com elementos evolutivos**, combinando:

1. especificação progressiva de requisitos;
2. ciclos curtos de projeto e implementação;
3. validação contínua do comportamento do sistema;
4. incorporação controlada de ajustes e extensões funcionais.

Essa abordagem permite que o sistema evolua de forma **controlada e rastreável**, reduzindo riscos associados a mudanças abruptas de escopo e favorecendo a manutenção da estabilidade operacional.

3.2 Justificativa da Abordagem Incremental

A escolha por um ciclo de vida incremental é justificada por fatores técnicos e contextuais específicos do SCMV, dentre os quais destacam-se:

1. natureza exploratória de parte dos requisitos cognitivos e simbólicos;
2. necessidade de observar padrões reais de interação antes da consolidação de funcionalidades;
3. importância de avaliar impactos humanos e comportamentais antes da expansão do escopo;
4. possibilidade de isolamento, teste e reversão de módulos específicos.

Essa abordagem permite que cada incremento seja tratado como uma **unidade controlada de evolução**, com objetivos claros, critérios de aceitação definidos e mecanismos de verificação associados.

3.3 Relação entre Ciclo de Vida e Governança

No contexto do SCMV, o ciclo de vida do software não é tratado apenas como um fluxo técnico de desenvolvimento, mas como parte integrante da **governança do sistema**. Decisões relacionadas a quando implementar, expandir ou restringir funcionalidades são avaliadas considerando:

1. riscos técnicos;
2. riscos de interpretação indevida do sistema;
3. impactos potenciais sobre o comportamento do usuário;
4. aderência a diretrizes de uso responsável de IA.

Essa integração entre ciclo de vida e governança contribui para a rastreabilidade das decisões de engenharia e para a coerência do sistema ao longo do tempo.

3.4 Visão Geral das Fases do Ciclo de Vida

De forma sintética, o ciclo de vida do SCMV Nemosine Nous © pode ser organizado nas seguintes fases, que se repetem de maneira iterativa:

1. **Especificação**: identificação e refinamento das necessidades, dores e limites do sistema;
2. **Projeto**: definição de arquitetura, modularização e decisões estruturais;
3. **Implementação**: desenvolvimento incremental de módulos e funcionalidades;
4. **Testes e Validação**: verificação do comportamento técnico e funcional do sistema;
5. **Manutenção e Evolução**: ajustes, correções e extensões controladas ao longo do tempo.

As seções seguintes detalham cada uma dessas fases, explicitando práticas, ferramentas e critérios adotados no contexto específico do SCMV.

4. Especificação de Requisitos

Esta seção formaliza os requisitos do **SCMV Nemosine Nous** ©, estabelecendo critérios técnicos objetivos que orientam decisões de projeto, implementação, testes e governança do sistema. Os requisitos são organizados em **funcionais** e **não funcionais**, além de requisitos específicos relacionados à segurança, governança e limites operacionais.

A especificação aqui apresentada não busca exaustividade absoluta, mas clareza, **rastreabilidade** e **controle de escopo**, permitindo evolução incremental sem perda de coerência técnica.

4.1 Requisitos Funcionais (RF)

Os requisitos funcionais descrevem **o que o sistema deve fazer**, independentemente da tecnologia utilizada.

RF-01 — Estruturação Modular do Conteúdo: O sistema deve permitir a organização do conteúdo cognitivo do usuário em **módulos independentes**, passíveis de criação, edição, reorganização e exclusão controlada.

RF-02 — Navegação Estruturada entre Módulos: O sistema deve permitir a navegação explícita entre módulos, mantendo a rastreabilidade das relações conceituais estabelecidas pelo usuário.

RF-03 — Persistência de Estados Cognitivos: O sistema deve registrar e manter estados estruturais das interações (configuração de módulos, relações, versões), possibilitando retomada posterior sem perda de contexto estrutural.

RF-04 — Interação Iterativa Assistida: O sistema deve suportar ciclos iterativos de interação, permitindo refinamento progressivo de estruturas conceituais sem imposição de fluxos rígidos.

RF-05 — Visualização de Estruturas: O sistema deve fornecer meios de visualização das estruturas organizadas (hierarquias, relações ou agrupamentos), de forma compreensível e manipulável.

4.2 Requisitos Não Funcionais (RNF)

Os requisitos não funcionais estabelecem **como o sistema deve se comportar**, influenciando arquitetura, desempenho e qualidade.

RNF-01 — Estabilidade Operacional: O sistema deve manter comportamento previsível e consistente durante interações prolongadas.

RNF-02 — Clareza de Escopo Funcional: O sistema deve comunicar de forma explícita suas capacidades e limitações, evitando interpretações indevidas sobre seu papel.

RNF-03 — Escalabilidade Modular: A arquitetura deve permitir a adição ou remoção de módulos sem impacto sistêmico desproporcional.

RNF-04 — Rastreabilidade Técnica: Decisões de projeto e evolução funcional devem ser documentáveis e rastreáveis ao longo do ciclo de vida.

RNF-05 — Manutenibilidade: O sistema deve ser estruturado de modo a facilitar correções, ajustes e extensões incrementais.

4.3 Requisitos de Segurança e Governança

Considerando a natureza cognitiva do sistema, são definidos requisitos específicos de segurança e governança.

RSG-01 — Limitação Deliberada de Atuação: O sistema não deve realizar diagnósticos, avaliações clínicas ou aconselhamento terapêutico.

RSG-02 — Prevenção de Dependência Funcional: O sistema deve evitar mecanismos que incentivem uso compulsivo, dependência emocional ou delegação excessiva de decisões pessoais.

RSG-03 — Contenção de Escopo Interpretativo: As respostas e interações do sistema devem ser formuladas de modo a reduzir ambiguidades quanto à autoridade ou papel do sistema.

RSG-04 — Governança de Evolução Funcional: Alterações significativas de escopo devem ser avaliadas sob critérios técnicos, éticos e de impacto humano antes de incorporação.

4.4 Critérios de Aceitação e Validação Inicial

Para cada incremento funcional, devem ser definidos critérios mínimos de aceitação, incluindo:

1. conformidade com os requisitos funcionais especificados;
2. ausência de comportamentos fora do escopo declarado;
3. previsibilidade das respostas do sistema;
4. manutenção da integridade estrutural dos módulos existentes.

Esses critérios servirão de base para as fases de **testes e validação**, descritas em seções posteriores.

4.5 Relação dos Requisitos com as Próximas Fases

A especificação apresentada nesta seção fundamenta diretamente:

1. as decisões de **arquitetura e design** (Seção 5);
2. a definição de **componentes de backend e frontend**;
3. a seleção de **ferramentas e tecnologias**;
4. a construção de **casos de teste derivados dos requisitos**.

A partir deste ponto, o documento transita explicitamente da **análise** para o **projeto técnico do sistema**.

5. Arquitetura e Projeto do Sistema

Esta seção descreve a arquitetura do **SCMV Nemosine Nous** ©, explicitando a separação de responsabilidades, os componentes principais do sistema e as decisões de projeto adotadas para atender aos requisitos funcionais, não funcionais e de governança definidos na Seção 4.

A arquitetura proposta prioriza **modularidade, rastreabilidade e controle de escopo**, evitando acoplamentos excessivos e permitindo evolução incremental segura.

5.1 Visão Arquitetural Geral

O SCMV adota uma arquitetura **em camadas**, com separação explícita entre:

1. **Camada de Apresentação (Frontend)**
2. **Camada de Aplicação / Orquestração (Backend)**
3. **Camada de Domínio Cognitivo-Modular**
4. **Camada de Persistência e Registro**

Essa separação permite isolar responsabilidades técnicas e conceituais, reduzindo riscos de efeitos colaterais indesejados e facilitando manutenção e auditoria.

5.2 Camada de Apresentação (Frontend)

A camada de frontend é responsável pela **interação direta com o usuário**, fornecendo meios para criação, navegação e visualização das estruturas modulares do sistema.

Responsabilidades principais:

1. apresentação clara das estruturas cognitivas organizadas;
2. suporte à navegação entre módulos;
3. visualização de relações, hierarquias e agrupamentos;
4. comunicação explícita de limites funcionais do sistema.

Considerações de projeto:

1. o frontend deve permanecer **tecnicamente leve**, evitando lógica cognitiva complexa;
2. validações críticas devem ocorrer no backend;
3. a interface deve reforçar o papel do sistema como **ferramenta de apoio**, e não como agente decisório.

5.3 Camada de Aplicação / Backend

A camada de backend concentra a **lógica de orquestração**, validação e governança do sistema.

Responsabilidades principais:

1. gerenciamento do ciclo de vida dos módulos;
2. validação das operações solicitadas pelo frontend;
3. controle de versões estruturais;
4. aplicação de regras de escopo e limites funcionais;
5. mediação entre o domínio cognitivo e a persistência.

Essa camada atua como **ponto de contenção**, impedindo que decisões críticas sejam delegadas exclusivamente à interface ou a componentes externos.

5.4 Camada de Domínio Cognitivo-Modular

A camada de domínio representa o **núcleo conceitual do SCMV**, onde são definidos:

1. o modelo de módulos;
2. as relações permitidas entre módulos;
3. os estados estruturais possíveis;
4. as regras de consistência interna.

Essa camada é **independente de tecnologia de interface** e deve ser projetada de modo a permitir:

1. evolução incremental do modelo;
2. isolamento de experimentações;
3. rastreabilidade de mudanças conceituais ao longo do tempo.

A modelagem do domínio cognitivo-modular do SCMV é suportada por esquemas taxonômicos e diagramas estruturais que descrevem entidades, eixos e relações funcionais do sistema.

*Esse esquemas utilizam **notações estruturais inspiradas em práticas de engenharia de software** (ex.: UML) com finalidade descritiva, exploratória e comunicacional, não constituindo especificação técnica executável nem impondo restrições diretas à implementação.*

*A representação de domínio mencionada (Esquema de Taxonomias Cognitivas do SMC Nemosine Nous) encontra-se disponível em:
<https://zenodo.org/records/17562035>*

5.5 Camada de Persistência e Registro

A camada de persistência é responsável pelo armazenamento seguro e consistente dos dados estruturais do sistema.

Responsabilidades principais:

1. persistência de módulos e relações;
2. versionamento estrutural;
3. registro de alterações relevantes;
4. suporte à recuperação de estados anteriores.

Decisões relativas à persistência devem priorizar **integridade** e **rastreabilidade**, em detrimento de otimizações prematuras de desempenho.

5.6 Integração entre Camadas

A comunicação entre as camadas deve ocorrer por **interfaces bem definidas**, minimizando acoplamentos implícitos. Em particular:

1. o frontend não deve acessar diretamente a camada de persistência;
2. a lógica de domínio não deve depender de detalhes de apresentação;
3. decisões de governança devem ser centralizadas no backend.

Essa organização reduz a probabilidade de comportamentos emergentes não controlados e facilita testes isolados por camada.

5.7 Considerações de Segurança Arquitetural

A arquitetura do SCMV incorpora medidas de segurança estrutural, tais como:

1. validação centralizada de operações;
2. limitação explícita de capacidades sensíveis;
3. isolamento de componentes experimentais;
4. possibilidade de desativação ou reversão de módulos.

Essas medidas são alinhadas às diretrizes de governança de IA e preparam o sistema para avaliações futuras de conformidade.

5.8 Relação da Arquitetura com as Próximas Fases

A arquitetura descrita nesta seção fundamenta diretamente:

1. a implementação técnica (Seção 6);
2. a definição de APIs e contratos internos;
3. a construção de casos de teste por camada;
4. a estratégia de manutenção e evolução do sistema.

5.9 Gestão de Identidade, Acesso e Perfis (IAM)

O SCMV Nemosine Nous © prevê a existência de um mecanismo de **Gestão de Identidade e Acesso (Identity and Access Management - IAM)** como componente transversal da arquitetura, responsável por controlar autenticação, autorização e perfis de uso do sistema.

A arquitetura deve contemplar, em nível de projeto:

1. identificação única de usuários;
2. controle de sessões e estados de autenticação;
3. definição de **perfis e papéis** (por exemplo: usuário final, desenvolvedor, administrador);
4. associação de permissões a módulos e operações específicas.

O mecanismo de IAM é tratado como **infraestrutura de governança**, não como funcionalidade central do domínio cognitivo, devendo ser projetado de forma desacoplada da lógica principal do sistema.

5.10 APIs e Integrações Externas

A arquitetura do SCMV considera a exposição controlada de funcionalidades por meio de **interfaces de programação de aplicações (APIs)**, permitindo integração com sistemas externos, módulos adicionais ou serviços complementares.

Em nível de projeto, são estabelecidas as seguintes diretrizes:

1. definição explícita de contratos de API;
2. versionamento de interfaces para evitar rupturas não controladas;

3. autenticação e autorização para acesso a APIs;
4. isolamento entre APIs internas e externas.

A existência de APIs não implica, neste estágio, abertura irrestrita do sistema, mas garante **preparação arquitetural para extensibilidade futura**.

5.11 Extensibilidade e Arquitetura de Plugins

O SCMV é concebido com potencial de **extensibilidade modular**, permitindo que funcionalidades adicionais sejam incorporadas sem comprometer a estabilidade do núcleo do sistema.

Essa extensibilidade é tratada, em nível de projeto, como:

1. definição de interfaces claras para módulos adicionais;
2. isolamento de módulos experimentais;
3. controle explícito de permissões e escopo de atuação de plugins;
4. possibilidade de desativação ou remoção sem impacto sistêmico.

A arquitetura de plugins não é obrigatoriamente ativada nas fases iniciais, mas sua previsão evita decisões irreversíveis de acoplamento estrutural.

5.12 Privacidade, Proteção de Dados e LGPD

A arquitetura do SCMV incorpora princípios de **privacidade por design e por padrão**, alinhados à legislação aplicável de proteção de dados pessoais, incluindo a Lei Geral de Proteção de Dados (LGPD).

Em nível de projeto, são consideradas:

1. minimização da coleta de dados;
2. separação entre dados identificáveis e dados estruturais;
3. definição de políticas de retenção e descarte;
4. possibilidade de anonimização ou pseudonimização;
5. rastreabilidade de decisões relacionadas a dados pessoais.

Essas diretrizes orientam decisões arquiteturais, independentemente das tecnologias específicas adotadas na implementação.

5.13 Camada de Mediação de Privacidade e Direcionamento de Prompts

Considerando o uso de serviços de IA em nuvem, a arquitetura do SCMV prevê a existência de uma **camada intermediária de mediação**, responsável por avaliar e tratar conteúdos sensíveis antes de qualquer encaminhamento a serviços externos.

Essa camada, concebida em nível de projeto, tem como responsabilidades:

1. classificação do conteúdo quanto à sensibilidade;
2. aplicação de regras de contenção, abstração ou sombreamento;
3. definição de políticas de direcionamento para serviços externos;
4. preservação do sigilo cognitivo do usuário.

Essa abordagem permite conciliar o uso de serviços de IA de alto desempenho com requisitos de privacidade, segurança e governança, sem dependência de processamento local integral.

5.14 Pagamentos, Comercialização e Não-Escopo Atual

A arquitetura do SCMV reconhece a possibilidade futura de **modelos de comercialização**, incluindo planos, serviços ou módulos pagos. No entanto, tais aspectos **não fazem parte do escopo funcional atual** do sistema.

Em nível de projeto, limita-se a:

1. evitar decisões arquiteturais que inviabilizem modelos futuros;
2. manter separação clara entre lógica cognitiva e lógica comercial;
3. tratar mecanismos de pagamento como componentes externos e desacoplados.

A implementação de funcionalidades comerciais será objeto de documentação específica, caso venha a ser considerada.

5.15 Camada de Representação Narrativa e Playbooks Cognitivos

A arquitetura do SCMV Nemosine Nous © admite a existência de uma **camada de representação narrativa**, destinada a mediar a interação do usuário com as estruturas cognitivas do sistema por meio de **modelos metafóricos configuráveis**, também denominados *playbooks cognitivos*.

Esses playbooks não alteram o núcleo funcional, os dados ou as regras do domínio, atuando exclusivamente como **camadas de organização, visualização e interpretação**, alinhadas a diferentes perfis de usuário e contextos de uso. Exemplos de representações incluem, mas não se limitam a, metáforas organizacionais, territoriais ou simbólicas.

Em nível de projeto, essa camada deve respeitar os seguintes princípios:

1. separação estrita entre lógica funcional e representação narrativa;
2. impossibilidade de indução de decisões ou interpretações normativas;
3. clareza quanto ao caráter instrumental da metáfora adotada;
4. possibilidade de substituição ou desativação sem impacto estrutural.

A inclusão dessa camada reconhece que diferentes usuários operam com **modelos mentais distintos**, e que a adaptação da representação pode facilitar a organização cognitiva sem interferir na integridade técnica ou na governança do sistema.

5.16 Interoperabilidade e Importação de Artefatos Estruturais

A arquitetura do SCMV Nemosine Nous © prevê a possibilidade de **importação controlada de artefatos estruturais externos**, tais como dashboards, painéis, mapas conceituais ou outros formatos de representação organizacional.

Essa capacidade visa permitir que usuários incorporem estruturas previamente existentes, respeitando limites de compatibilidade, validação e adaptação ao modelo interno do sistema. Em nível de projeto, considera-se:

1. definição de formatos suportados ou adaptáveis;
2. validação estrutural dos artefatos importados;
3. isolamento de inconsistências ou incompatibilidades;
4. preservação da integridade do domínio cognitivo interno.

A importação de artefatos não implica equivalência automática de semântica, sendo tratada como **processo assistido de incorporação**, e não como simples cópia de estruturas externas.

5.17 Modelos de Distribuição, Instalação e Acesso ao Sistema

A arquitetura do SCMV Nemosine Nous © considera diferentes **modelos de distribuição e acesso**, reconhecendo que a forma de disponibilização do sistema influencia diretamente requisitos de segurança, privacidade, desempenho e experiência do usuário.

Em nível de projeto, são considerados os seguintes modelos não excludentes:

1. **Aplicação Web**: acesso por navegador, sem instalação local obrigatória, permitindo rápida adoção e atualização centralizada;
2. **Aplicação Instalável**: empacotamento do sistema como software instalável em ambiente local ou corporativo, com maior controle sobre dados e execução;
3. **Aplicação Móvel**: disponibilização parcial ou adaptada das funcionalidades em dispositivos móveis, respeitando limitações de interface e contexto de uso;
4. **Ambiente Híbrido**: combinação de componentes locais e serviços em nuvem, mediada por camadas de segurança e privacidade.

A arquitetura deve evitar dependência rígida de um único modelo de instalação, preservando flexibilidade para evolução futura. A escolha ou coexistência de modelos será orientada por critérios técnicos, contextuais e de

governança, não fazendo parte do escopo de implementação inicial descrito neste documento.

Detalhes técnicos relativos a empacotamento, instalação, atualização e distribuição serão tratados em seções posteriores ou documentação específica.

5.18 Geração e Exportação de Artefatos

A arquitetura do SCMV Nemosine Nous © prevê a capacidade de **geração e exportação de artefatos externos**, permitindo que conteúdos estruturados no ambiente cognitivo sejam convertidos em formatos utilizáveis fora do sistema.

Em nível de projeto, essa capacidade compreende:

1. produção de documentos textuais estruturados;
2. geração de imagens, esquemas ou representações visuais;
3. exportação de conteúdos organizados para formatos amplamente utilizados.

A geração de artefatos é tratada como **função de saída do sistema**, não como mecanismo de autoria autônoma, devendo preservar a rastreabilidade do conteúdo e a autoria do usuário. Detalhes técnicos sobre formatos, bibliotecas e mecanismos de renderização são tratados em nível de implementação.

5.19 Integração com Plataformas Externas de Comunicação

O SCMV considera a integração controlada com **plataformas externas de comunicação**, tais como redes sociais, mensageiros ou serviços de compartilhamento de conteúdo, com o objetivo de facilitar a circulação contextualizada de informações produzidas no sistema.

Em nível de projeto, essa integração é caracterizada por:

1. utilização de links, referências ou mecanismos de acesso contextual;
2. ausência de substituição direta das plataformas externas;
3. preservação da separação entre o ambiente cognitivo e os canais sociais.

O sistema não se propõe a atuar como rede social própria, limitando-se a **orquestrar pontos de conexão** entre o usuário e plataformas externas quando relevante.

*As integrações descritas nesta seção não configuram, em hipótese alguma, um ambiente de interação social interno ao SCMV Nemosine Nous ©. O sistema não hospeda, media ou promove comunicação direta entre usuários, limitando-se a fornecer **pontos de acesso contextual** a plataformas externas, sob controle explícito do próprio usuário.*

Essa diretriz arquitetural visa preservar o caráter individual, instrumental e introspectivo do sistema, evitando dinâmicas de engajamento social, comparação ou validação externa que não fazem parte de seu escopo funcional.

5.20 Coleta de Feedback, Pesquisa e Comunicação com Desenvolvedores

A arquitetura do SCMV inclui mecanismos para **coleta estruturada de feedback**, visando aprimoramento contínuo, validação empírica e comunicação entre usuários e desenvolvedores.

Essa capacidade contempla, em nível de projeto:

1. formulários de feedback funcional;
2. instrumentos de pesquisa e validação acadêmica;
3. canais formais para registro de solicitações ou reclamações.

Esses mecanismos são concebidos como **instrumentos de governança e melhoria contínua**, não como substitutos de suporte profissional ou mediação humana direta.

5.21 Espaço Pessoal de Armazenamento e Registros do Usuário

O SCMV prevê a existência de um **espaço pessoal de armazenamento**, destinado ao registro e organização de materiais produzidos ou utilizados pelo próprio usuário no contexto do sistema.

Esse espaço pode incluir:

1. anotações pessoais;
2. arquivos auxiliares;
3. tabelas, planilhas ou documentos de apoio.

O objetivo desse componente é oferecer um **ambiente de trabalho cognitivo integrado**, sem a pretensão de substituir serviços genéricos de armazenamento ou gestão de arquivos.

5.22 Persistência, Logs de Interação e Base para Personalização

A arquitetura do SCMV inclui mecanismos de **persistência e registro de interações**, destinados a sustentar a evolução controlada do sistema e a personalização progressiva da experiência do usuário.

Em nível de projeto, considera-se:

1. registro estrutural de interações relevantes;
2. separação entre dados identificáveis e dados de uso;
3. definição de políticas de retenção e descarte;
4. possibilidade de utilização desses registros como substrato para adaptação do sistema.

Esses mecanismos são projetados com atenção explícita a privacidade, governança e conformidade regulatória, evitando coleta indiscriminada ou retenção excessiva de dados.

5.23 Constituição Funcional do SCMV Nemosine Nous ©

A Constituição Funcional do SCMV Nemosine Nous © **declara e operacionaliza**, no âmbito do software, os **princípios invariantes, limites**

deliberados e diretrizes de governança já previstos na Constituição Nemosínica vigente, orientando a concepção, evolução e uso legítimo do sistema. Seu objetivo é garantir coerência interna, previsibilidade de comportamento e alinhamento ético-funcional ao longo do ciclo de vida do software, independentemente de implementações específicas.

Esta constituição não se confunde com documentação técnica de implementação, nem com políticas operacionais externas, atuando como **camada fundacional de projeto** que precede e orienta decisões arquiteturais e técnicas.

A Constituição Nemosínica encontra-se formalmente registrada e versionada em repositório público, com cópia arquivada em ambiente de preservação científica, disponível em: <https://zenodo.org/records/17478379>

5.23.1 Diretrizes Constitucionais Aplicadas ao SCMV

A Constituição Funcional do SCMV Nemosine Nous © não estabelece princípios novos, mas **declara e operacionaliza, no âmbito do software**, as diretrizes já previstas na Constituição Nemosínica vigente.

Esta seção tem por finalidade explicitar como tais diretrizes constitucionais são refletidas no escopo, nos limites e na governança do sistema, servindo como ponte entre o corpo constitucional superior e as decisões de projeto descritas neste documento.

1. **Centralidade do Usuário Individual:** manifesta-se na concepção do sistema como ambiente de uso individual, voltado à organização, reflexão e modelagem cognitiva pessoal.

2. **Instrumentalidade Cognitiva:** traduz-se no papel do SCMV como ferramenta de apoio e estruturação do pensamento, sem substituição de julgamento humano, decisão profissional ou mediação terapêutica.

3. **Modularidade e Evolução Controlada:** reflete-se no projeto de componentes modulares, permitindo evolução incremental sem ruptura das diretrizes constitucionais.

4. Transparência Funcional: orienta a adoção de estruturas comprehensíveis em nível conceitual e técnico, evitando mecanismos opacos que impeçam auditoria.

5. Governança e Responsabilidade: materializa-se na observância de princípios de responsabilidade, privacidade e conformidade regulatória nas decisões críticas de projeto e uso.

5.23.2 Limites Deliberados de Escopo

A Constituição Funcional define explicitamente limites de atuação do SCMV, entre os quais destacam-se:

1. o sistema **não se caracteriza como rede social**, nem promove dinâmicas de engajamento social, comparação ou validação externa;
2. o SCMV **não substitui** acompanhamento profissional nas áreas de saúde, psicologia, psiquiatria ou afins;
3. o sistema **não assume responsabilidade por decisões finais do usuário**, atuando exclusivamente como ambiente de apoio cognitivo;
4. funcionalidades que impliquem interação social direta, mediação de conflitos ou aconselhamento clínico encontram-se fora do escopo do sistema.

Esses limites são considerados **elementos estruturais do projeto**, não restrições temporárias.

5.23.3 Diretrizes de Privacidade e Uso Ético

O SCMV deve observar, como diretriz constitucional:

1. minimização da coleta de dados;
2. separação clara entre dados do usuário e dados operacionais;
3. controle explícito do usuário sobre registros e persistência;
4. prevenção de usos que induzam dependência, sofrimento psicológico ou exposição indevida.

Essas diretrizes orientam decisões técnicas posteriores, incluindo persistência, logs, integração com serviços externos e uso de inteligência artificial.

5.23.4 Relação com Implementação e Evolução do Sistema

A Constituição Funcional orienta, mas **não determina de forma rígida**, as escolhas de implementação descritas na Seção 6. Alterações técnicas, refatorações ou substituições de componentes são permitidas, desde que preservem os princípios e limites aqui definidos.

A evolução do SCMV deve respeitar a distinção entre:

1. **princípios constitucionais**, considerados invariantes;
 2. **decisões técnicas**, passíveis de revisão ao longo do desenvolvimento.
-

6. Implementação e Tecnologias Utilizadas

Esta seção descreve as **diretrizes de implementação** do SCMV Nemosine ©, incluindo opções tecnológicas consideradas, critérios de escolha e organização dos componentes técnicos, sem assumir compromisso definitivo com stacks específicas além do necessário para viabilizar uma primeira versão funcional.

O objetivo desta seção não é detalhar código ou infraestrutura final, mas **documentar decisões técnicas conscientes**, alinhadas aos requisitos e à arquitetura definidos nas seções anteriores.

6.1 Princípios Gerais de Implementação

A implementação do SCMV deve respeitar os seguintes princípios:

1. **simplicidade inicial**: evitar complexidade prematura;

2. **evolução incremental:** permitir refatoração e substituição de componentes;
3. **desacoplamento:** minimizar dependências rígidas entre camadas;
4. **observabilidade:** permitir inspeção e auditoria do comportamento do sistema;
5. **governança técnica:** garantir aderência às diretrizes de privacidade e escopo.

Esses princípios orientam a seleção de tecnologias e padrões, independentemente da stack específica adotada.

6.2 Organização Geral da Implementação

Em nível de implementação, o SCMV é organizado conforme a arquitetura em camadas descrita anteriormente:

1. **Frontend:** responsável pela interface e interação do usuário;
2. **Backend:** responsável pela orquestração, validação e governança;
3. **Domínio Cognitivo-Modular:** núcleo lógico e conceitual do sistema;
4. **Persistência e Logs:** armazenamento e registro de estados e interações;
5. **Camadas Transversais:** segurança, privacidade, IAM e mediação de prompts.

Essa organização deve ser refletida na estrutura do código e nos repositórios, evitando misturas de responsabilidade.

6.3 Frontend – Diretrizes Tecnológicas

A camada de frontend deve priorizar:

1. clareza visual;
2. navegação estruturada;
3. separação entre apresentação e lógica de domínio.

Em nível de projeto, são consideradas adequadas tecnologias baseadas em:

1. aplicações web modernas (SPA ou MPA);
2. componentes reutilizáveis;
3. comunicação com backend via APIs bem definidas.

A escolha específica de frameworks ou bibliotecas deve considerar maturidade, comunidade e facilidade de manutenção, não tendências momentâneas.

6.4 Backend – Diretrizes Tecnológicas

O backend concentra a lógica central do sistema, incluindo:

1. gerenciamento de módulos;
2. validação de operações;
3. aplicação de regras de governança;
4. mediação de acesso a serviços externos.

A implementação deve favorecer:

1. APIs explícitas e versionadas;
2. validação centralizada;
3. isolamento de componentes sensíveis;
4. facilidade de testes automatizados.

O backend deve atuar como **ponto único de controle**, evitando que decisões críticas sejam delegadas a camadas externas.

6.5 Persistência de Dados e Logs

A implementação da persistência deve atender a:

1. armazenamento consistente de estruturas cognitivas;
2. versionamento de estados relevantes;
3. separação entre dados do usuário e dados operacionais;
4. suporte à política de retenção e descarte.

Logs de interação devem ser tratados como **instrumentos técnicos**, não como captura indiscriminada de conteúdo, respeitando princípios de minimização e privacidade.

A concepção geral de estruturas cognitivas, persistência e organização de estados encontra-se descrita, em nível conceitual, em pedido de patente previamente depositado, que apresenta uma possível realização do modelo cognitivo subjacente ao sistema.

*O referido pedido tem caráter **descritivo e exemplificativo**, não constituindo especificação técnica vinculante para a implementação descrita nesta seção, a qual permanece orientada por princípios de modularidade, privacidade e evolução incremental.*

*Referência: Pedido de Patente – Método Implementado por Computador para Processamento Cognitivo Modular (INPI), disponível em:
<https://zenodo.org/records/17874787>*

6.6 Mediação de Prompts e Uso de Serviços de IA

A interação com serviços de IA em nuvem deve ocorrer por meio da **camada de mediação de privacidade**, conforme definido na arquitetura.

Em nível de implementação, essa camada deve permitir:

1. tratamento prévio de conteúdos sensíveis;
2. abstração de identificadores diretos;
3. controle explícito do que é enviado a serviços externos;
4. possibilidade de auditoria técnica do fluxo de dados.

A implementação dessa mediação é considerada **componente crítico**, devendo ser isolada e testável.

A lógica de mediação de prompts e classificação de sensibilidade é detalhada em blueprint técnico específico, versionado e publicado separadamente, servindo como referência conceitual para implementação desta camada.

A estratégia de mediação de prompts, classificação de sensibilidade e direcionamento de conteúdos para serviços de IA externos encontra-se esboçada em blueprint técnico específico, publicado separadamente como material de planejamento conceitual.

*Esse blueprint tem caráter **exploratório e não normativo**, servindo como referência de alto nível para discussão, avaliação e evolução da camada de mediação, sem impor aderência literal à implementação descrita nesta seção.*

Referência: Blueprint de Mediação de Prompts e Sigilo Cognitivo (Zenodo), disponível em: <https://zenodo.org/records/17081662>

6.7 Controle de Versão, Evolução e Implantação

A implementação do SCMV deve adotar práticas consolidadas de:

1. controle de versão de código;
2. versionamento de APIs;
3. evolução incremental de funcionalidades;
4. separação entre ambientes (desenvolvimento, teste, produção).

Detalhes de infraestrutura, automação e implantação são tratados como **decisões técnicas progressivas**, não exigidas na fase inicial do sistema.

6.8 Escopo da Primeira Implementação

Para evitar sobrecarga e dispersão, a primeira implementação do SCMV deve ser **deliberadamente limitada**, contemplando apenas:

1. núcleo cognitivo modular básico;
2. interface mínima funcional;
3. persistência essencial;
4. mediação básica de prompts;
5. ausência de funcionalidades sociais internas.

Funcionalidades avançadas, integrações extensas e modelos comerciais são considerados **fora do escopo inicial**, devendo ser introduzidos apenas após validação técnica e conceitual.

6.9 Provas de Conceito e Implementações Preliminares

Como parte do processo de validação técnica inicial, foram desenvolvidas **provas de conceito (PoCs)** exploratórias, com o objetivo de avaliar a viabilidade de determinadas decisões arquiteturais e fluxos de interação.

Entre essas iniciativas, destaca-se uma implementação preliminar baseada em **Flask**, utilizada exclusivamente como experimento inicial de backend e orquestração de fluxos, sem caráter vinculante em relação à stack definitiva do sistema.

Essas provas de conceito devem ser compreendidas como **artefatos transitórios**, sujeitos a refatoração, substituição ou descarte, conforme a evolução do projeto.

*Como parte do processo de validação técnica inicial do SCMV Nemosine Nous ©, foram desenvolvidas **provas de conceito (PoCs)** com caráter exploratório, destinadas a testar viabilidade técnica, fluxos básicos de interação e integração entre componentes.*

*A primeira prova de conceito concentrou-se na **estruturação inicial do sistema em ambiente Python**, incluindo menu interativo, organização simbólica básica e experimentação de execução em ambiente Linux/Android (Termux), bem como procedimentos iniciais de versionamento e importação de código para repositório.*

*Uma segunda prova de conceito explorou um **fluxo full-stack básico**, envolvendo backend em Python (APIs) e frontend web simples, com integração a serviços externos de inteligência artificial, permitindo avaliar comunicação entre camadas, separação de responsabilidades e viabilidade de uma arquitetura baseada em APIs.*

*Essas provas de conceito possuem **caráter não vinculante**, não representando decisões definitivas de stack ou arquitetura, e devem ser compreendidas como artefatos transitórios de aprendizado, experimentação e validação preliminar.*

Referências às provas de conceito:

1. Nemosine PoC - Primeira Prova de Conceito (Zenodo):
<https://zenodo.org/records/17538243>
2. Nemosine Nous © - PoC Oficial (API) (Zenodo):
<https://zenodo.org/records/17685495>

Além das provas de conceito descritas, o processo de desenvolvimento do SCMV incluiu **experimentos técnicos complementares**, voltados ao aprendizado progressivo de práticas de desenvolvimento e organização do sistema. Entre esses registros destacam-se:

1. commits iniciais relacionados à estruturação do núcleo do sistema (*core*), incluindo o primeiro versionamento funcional do projeto;
2. experimentos de representação e organização de personas em formato HTML, utilizados como exploração preliminar de visualização e estruturação de conteúdo;
3. protótipos conceituais desenvolvidos em ambientes de modelagem e documentação, como Notion e NotebookLM, empregados como ferramentas auxiliares de reflexão, organização e validação conceitual.

Esses artefatos possuem caráter **exploratório e transitório**, não compondo a implementação técnica do sistema, mas registrando etapas relevantes do processo de aprendizado, concepção e consolidação do SCMV.

Referências complementares aos experimentos exploratórios mencionados nesta seção encontram-se disponíveis em repositórios públicos e ambientes de documentação, mantidos como registros históricos do processo de concepção e aprendizado do sistema:

1. Repositório de desenvolvimento e commits iniciais do SCMV (GitHub):
<https://zenodo.org/records/17477279>
2. Experimentos de representação de personas e estruturação inicial em HTML:
<https://zenodo.org/records/17478513>

3. Protótipos conceituais desenvolvidos em ambientes de modelagem e documentação:
 - a. Notion: <https://zenodo.org/records/17507986>
 - b. NotebookLM: <https://zenodo.org/records/17582836>
-

7. Testes, Validação e Qualidade

Este capítulo descreve a abordagem adotada para **testes, validação e controle de qualidade** do SCMV Nemosine Nous ©, considerando seu estágio atual de desenvolvimento, sua natureza exploratória e seu caráter de sistema cognitivo modular em evolução.

A estratégia aqui apresentada **não pressupõe maturidade industrial completa**, mas estabelece **critérios mínimos de confiabilidade, rastreabilidade e segurança conceitual**, compatíveis com o propósito do projeto nesta fase.

7.1 Estratégia Geral de Testes

A estratégia de testes do SCMV é orientada pelos seguintes princípios:

1. **testabilidade incremental**, acompanhando a evolução modular do sistema;
2. **separação entre validação conceitual e validação técnica**;
3. **priorização de estabilidade estrutural sobre performance otimizada**;
4. **evitação de testes artificiais que induzam falsa sensação de robustez**.

Os testes são concebidos como instrumentos de **confirmação de coerência e previsibilidade**, e não como garantia absoluta de correção em sistemas cognitivos complexos.

7.2 Tipos de Testes Considerados

No estágio atual, são considerados apropriados:

1. Testes unitários

Aplicados a funções isoladas do backend e a componentes lógicos bem delimitados, com foco em:

- consistência de entradas e saídas;
- validação de estados esperados;
- tratamento de exceções previsíveis.

2. Testes de integração

Voltados à verificação da comunicação entre:

- frontend e backend;
- backend e serviços externos (ex.: APIs de IA);
- camadas de persistência e domínio cognitivo.

3. Validação funcional exploratória

Realizada por meio de uso real do sistema, observando:

- previsibilidade do comportamento;
- ausência de estados incoerentes;
- clareza de respostas e transições.

Testes automatizados extensivos, testes de carga e testes de estresse **não são prioridade nesta fase**, sendo reservados para etapas futuras de maturação.

7.3 Critérios de Aceitação

Um componente, módulo ou funcionalidade é considerado aceitável quando:

1. executa sua função sem comprometer a coerência global do sistema;
2. não introduz efeitos colaterais não rastreáveis;
3. respeita os limites funcionais e éticos definidos na Constituição do sistema;
4. permite reversibilidade ou isolamento em caso de falha.

A aceitação é, portanto, **funcional e conceitual**, e não meramente baseada em métricas quantitativas.

7.4 Validação Ética e Cognitiva

Considerando a natureza do SCMV como ambiente de reflexão e organização cognitiva, a validação do sistema inclui:

1. observação de **potenciais efeitos adversos cognitivos**;
2. prevenção de indução de dependência, confusão ou distorção decisória;
3. respeito aos princípios de responsabilidade e mediação consciente definidos na Constituição.

Essa validação não substitui acompanhamento profissional externo, nem se propõe a operar como instrumento terapêutico.

7.5 Limites Deliberados da Fase Atual

De forma explícita, **não fazem parte do escopo atual de testes**:

1. certificações formais de qualidade;
2. auditorias externas independentes;
3. conformidade completa com normas internacionais de gestão;
4. validação estatística em larga escala com múltiplos usuários.

Esses elementos são reconhecidos como relevantes, mas **adiados conscientemente**, sendo tratados no roadmap evolutivo do sistema.

8. Riscos, Limitações e Decisões Deliberadas

Este capítulo explicita os **riscos identificados**, as **limitações assumidas** e as **decisões deliberadas de escopo** adotadas no desenvolvimento do SCMV Nemosine Nous ©. A inclusão deste capítulo visa **aumentar a transparência técnica**, reduzir ambiguidades interpretativas e evitar expectativas inadequadas quanto às capacidades e objetivos do sistema.

8.1 Riscos Técnicos

Os principais riscos técnicos identificados no estágio atual incluem:

1. **complexidade crescente do domínio cognitivo**, podendo dificultar manutenção e evolução se não houver controle modular rigoroso;
2. **dependência de serviços externos de IA**, sujeita a mudanças de disponibilidade, custo ou comportamento;
3. **acoplamento excessivo entre camadas**, caso decisões arquiteturais não sejam continuamente revisadas;
4. **fragilidade de provas de conceito**, inerente a artefatos exploratórios e não industrializados.

Esses riscos são mitigados prioritariamente por meio de modularidade, documentação explícita e isolamento de responsabilidades.

8.2 Riscos Cognitivos e de Uso Indevido

Considerando a natureza do SCMV como ambiente de reflexão e organização mental, são reconhecidos os seguintes riscos:

1. uso do sistema como **substituto indevido de julgamento humano ou profissional**;
2. interpretação equivocada das respostas do sistema como orientação normativa ou prescritiva;
3. dependência excessiva de mediação algorítmica para decisões pessoais ou profissionais;
4. extração do sistema para finalidades terapêuticas ou clínicas não previstas.

Esses riscos são tratados por meio de **delimitação explícita de escopo**, linguagem não prescritiva e princípios constitucionais de responsabilidade.

8.3 Limitações Atuais do Sistema

De forma objetiva, o SCMV apresenta, nesta fase:

1. ausência de validação empírica em larga escala;

2. inexistência de métricas quantitativas consolidadas de desempenho cognitivo;
3. limitação de recursos de automação avançada;
4. dependência de interação consciente e ativa do usuário;
5. inexistência de garantias formais de completude ou correção cognitiva.

Essas limitações são reconhecidas como inerentes a sistemas cognitivos experimentais em estágio inicial.

8.4 Decisões Deliberadas de Exclusão

Algumas decisões foram tomadas de forma consciente para **não integrar determinadas funcionalidades** nesta fase, incluindo:

1. não caracterização do sistema como rede social;
2. ausência de mecanismos de gamificação ou competição;
3. não adoção de modelos de recomendação automática invasivos;
4. exclusão de mecanismos de monetização agressiva;
5. não enquadramento do sistema como ferramenta clínica, terapêutica ou diagnóstica.

Essas exclusões visam preservar a **centralidade do usuário individual**, a previsibilidade do sistema e a clareza ética de uso.

8.5 Considerações Regulatórias e de Governança

Embora ainda não submetido a certificações formais, o desenvolvimento do SCMV considera princípios alinhados a boas práticas de governança de sistemas de IA, incluindo:

1. responsabilidade no uso;
2. transparência funcional;
3. minimização de riscos;
4. respeito à privacidade e aos limites regulatórios aplicáveis.

A adoção de padrões normativos mais formais é tratada como **etapa futura**, conforme descrito no roadmap do projeto.

9. Roadmap Técnico e Evolução Prevista

Este capítulo apresenta uma visão **prospectiva e não vinculante** da evolução do SCMV Nemosine Nous ©, organizada em horizontes temporais, com o objetivo de orientar expectativas e registrar intenções de desenvolvimento sem comprometer a flexibilidade do projeto.

O roadmap descrito a seguir **não constitui compromisso de entrega**, devendo ser entendido como diretriz adaptável conforme maturação técnica, validações empíricas e contexto regulatório.

9.1 Curto Prazo

No curto prazo, a evolução do sistema prioriza:

1. consolidação da arquitetura modular definida;
2. refino das interfaces de interação;
3. fortalecimento da separação entre domínio cognitivo, camadas técnicas e mediação de IA;
4. estabilização de fluxos básicos de uso e persistência;
5. documentação incremental das decisões de projeto.

9.2 Médio Prazo

Em horizonte de médio prazo, são considerados:

1. ampliação controlada de mecanismos de personalização;
2. melhoria de observabilidade, logs e rastreabilidade de estados;
3. integração mais estruturada com serviços externos de IA;
4. experimentos supervisionados de validação empírica;

5. evolução das práticas de governança e segurança.

9.3 Longo Prazo

No longo prazo, são reconhecidas como possíveis linhas de investigação e desenvolvimento:

1. aprofundamento em modelos de meta-inteligência pessoal;
2. estudos longitudinais de uso continuado;
3. avaliação de aderência a padrões internacionais de governança de IA (ex.: ISO/IEC 42001);
4. exploração de novos formatos de interação e representação cognitiva;
5. produção de documentação técnica e científica complementar.

Essas iniciativas permanecem **abertas e exploratórias**, sujeitas a revisão.

10. Conclusão

Este Technical Report apresentou o mapeamento estrutural, conceitual e técnico do **SCMV Nemosine Nous** ©, descrevendo seu escopo, arquitetura, princípios de governança, decisões deliberadas e limitações reconhecidas.

O documento não se propõe a definir uma implementação fechada, tampouco a oferecer um framework universal de engenharia de software. Seu objetivo é **estabelecer um artefato técnico de referência**, capaz de orientar a evolução consciente do sistema, preservar coerência interna e registrar decisões fundamentais ao longo de seu ciclo de vida.

Ao explicitar tanto o que o sistema é quanto o que deliberadamente não pretende ser, este relatório busca contribuir para uma compreensão clara, responsável e tecnicamente honesta do projeto, mantendo espaço para aprendizado contínuo, adaptação e aprofundamento futuro.

Referências

1. **Melo, E. J. S.** *SCMV Nemosine Nous © – Constituição Funcional do Sistema*. Zenodo.
2. **Melo, E. J. S.** *Esquema de Taxonomias Cognitivas do SCM Nemosine Nous ©*. Zenodo.
3. **Melo, E. J. S.** *Blueprint de Mediação Cognitiva e Sigilo de Prompts*. Zenodo.
4. **Melo, E. J. S.** *Provas de Conceito (PoCs) do SCMV Nemosine Nous ©*. Repositório GitHub / Zenodo.
5. **Instituto Nacional da Propriedade Industrial (INPI)**. Registro de Programa de Computador BR512025003335-4.
6. **Instituto Nacional da Propriedade Industrial (INPI)**. Pedido de Patente – Método Implementado por Computador para Processamento Cognitivo Modular.
7. **International Organization for Standardization (ISO)**. ISO/IEC 42001: Artificial Intelligence – Management System.
8. **Brasil**. Lei nº 13.709, de 14 de agosto de 2018 (Lei Geral de Proteção de Dados – LGPD).
9. **Object Management Group (OMG)**. *Unified Modeling Language (UML)* – documentação oficial.
10. **Zenodo Community**. *Sistema Nemosine – Série Técnica*.