

# Relato de Experiência: Utilização de Ontologias Para Reestruturação de Serviços de Suporte e Desenvolvimento de Software

Junilson Pereira Souza ([junilson@gmail.com](mailto:junilson@gmail.com))

Programa de Pós-Graduação em Gestão e Organização do Conhecimento (PPGGOC)  
Escola de Ciência da Informação (ECI)  
Universidade Federal de Minas Gerais (UFMG)

## Resumo

**Objetos de estudo.** Processo de manutenção de software e respectivo sistema de gestão de demandas de uma empresa de desenvolvimento de software. **Problemas.** Dificuldade em organizar o fluxo de trabalho, falhas na comunicação, falta de padronização e dificuldade em gerar informações gerenciais. **Objetivo.** Criar uma ontologia que sirva de base para revisar a estrutura de uma ferramenta computacional de suporte ao processo de manutenção de software. **Justificativa.** A padronização dos tipos de demandas, relacionamentos e atributos permitirá uma melhor comunicação e gestão de demandas e melhoria nos serviços, permitindo trazer maior satisfação para as partes interessadas e maior sustentabilidade empresarial.

## 1. Introdução

A manutenção de produtos de software representa uma parcela significativa dos esforços e custos para empresas que provêm produtos de software como serviços. Nesse sentido, a utilização de ferramentas computacionais para gerenciamento de todo fluxo de trabalho torna-se praticamente uma obrigatoriedade, trazendo à reboque um desafio considerável que é a necessidade de adequar tais ferramentas ao contexto organizacional. Em alguns cenários, é comum a utilização de ferramentas chamadas *issue trackers* para o propósito pretendido. Uma característica comum dessas ferramentas é uma grande capacidade de personalização por meio do uso de recursos tais como campos e fluxos personalizados.

A personalização da ferramenta feita de maneira não estruturada pode acarretar vários problemas dentre os quais podem ser citados: a dificuldade de

comunicação em função de uso de terminologia inconsistente, a dificuldade em gerenciar e executar o trabalho de várias partes interessadas em função de um fluxo de trabalho que não reflita o entendimento comum. Tais problemas podem levar a um aumento do esforço, custo e prazo na entrega das demandas, além de impactar na qualidade dos produtos e serviços prestados, levando a insatisfação das partes interessadas.

O objetivo deste trabalho é criar um projeto de ontologia para mapear um processo de manutenção de software que possa ser implementado em uma ferramenta computacional, endereçando os problemas relatados pelas diversas partes interessadas.

Nas seções seguintes são apresentados um breve resumo dos conceitos associados ao processo de manutenção de software, além do conceito de ontologia e projetos dessa natureza. Em seguida, são detalhados os procedimentos metodológicos utilizados, seguidos dos resultados e respectivas discussões. Ao final, são feitas considerações sobre os resultados obtidos e sinalizações para a continuidade e implantação da ontologia.

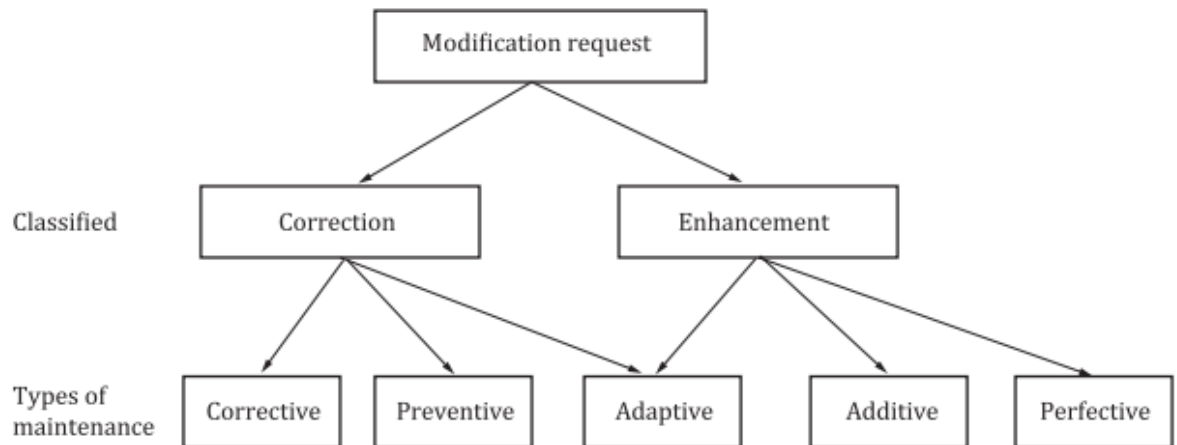
## **2. Referencial Bibliográfico**

Esta seção apresenta os dois principais pilares conceituais explorados no trabalho, que são os processos de manutenção de software e as ontologias.

### **2.1. Manutenção de software**

A manutenção de produtos de software representa uma parcela significativa dos esforços e custos para empresas que provêm produtos de software como serviços. Dentre as várias referências para a realização do processo de manutenção, destaca-se a norma ISO/IEC/IEEE 14764:2022, *Software engineering - Software life cycle processes - Maintenance* (ISO, 2022). A norma propõe uma estruturação das requisições de modificação conforme mostrado na figura 1, com os respectivos tipos de manutenção detalhados na sequência.

Figura 1 - Tipos de requisições de mudança



Fonte: (ISO, 2022).

Solicitação de mudança: item de informação que identifica e descreve mudanças propostas em um produto ou serviço.

Manutenção corretiva: modificação de um produto de software realizada após a entrega para corrigir problemas detectados. A modificação visa reparar o produto de software para atender aos requisitos de sistema definidos.

Manutenção preventiva: modificação de um produto de software após a entrega para corrigir falhas latentes no software antes que elas ocorram no sistema em produção.

Manutenção adaptativa: modificação de um produto de software, realizada após a entrega, para mantê-lo utilizável em um ambiente alterado ou em constante mudança. A manutenção adaptativa fornece melhorias necessárias para acomodar as mudanças no ambiente em que o software opera. Essas mudanças ajudam a acompanhar a evolução do ambiente; por exemplo, uma atualização do sistema operacional resulta em alterações nos aplicativos.

Manutenção aditiva: modificação de um produto de software realizada após a entrega para adicionar funcionalidades ou recursos que aprimorem o uso do produto. O tipo de manutenção aditiva se distingue do tipo de manutenção

"perfectiva" e é reconhecido como um tipo de manutenção diferente por ser capaz de: — fornecer novas funções ou recursos adicionais para melhorar a usabilidade, o desempenho, a manutenibilidade ou outros atributos do software no futuro; — adicionar funcionalidades ou recursos com adições ou alterações relativamente grandes no software para melhorar os atributos do software após a entrega, com oportunidades identificadas para negociar quaisquer adições ou alterações na estratégia, métodos, recursos, acordos ou níveis de serviço de manutenção entre fornecedores e adquirentes.

Manutenção aperfeiçoativa (*perfective*): modificação de um produto de software para proporcionar melhorias para os usuários, aprimoramento das informações para os usuários e registro de dados para melhorar o desempenho, a capacidade de manutenção ou outros atributos do software.

Para a manutenção e atendimento das solicitações de mudança do software, são necessários diversos processos, que podem ser estruturados conforme indicado nos modelos de referência de software - MR-SW (Softex, 2024) e serviços - MR-SV (Softex, 2024b):

- Processos de serviço (MR-SV): Gerência de Operação de Serviços, Atendimento e Solução e Gerência de Nível de Serviços;
- Processos de projeto (MR-SW): Gerência de Projetos, Engenharia de Requisitos, Projeto e Construção do Produto, Integração do Produto e Verificação e Validação;
- Processos organizacionais (MR-SW): Gerência de Configuração, Medição, Gerência de Decisões, Capacitação, Gerência de Processos, Gerência Organizacional (Software e Serviço).

## 2.2. Ontologias

O avanço das tecnologias computacionais em geral e dos mecanismos de gerenciamento de dados em particular, têm sido insuficientes para garantir um uso pleno e efetivo por todas as partes interessadas, especialmente quando são consideradas as dimensões de acessibilidade, interoperabilidade, compartilhamento e reuso. Em determinadas situações, a diversidade de tecnologias adotadas pelos especialistas pode ser um fator complicador. Além disso, apesar de ter sido experimentada uma grande expansão na capacidade de memória e processamento, os computadores ainda demonstram não serem inteligentes; tanto por não entenderem a si mesmos, nem a forma como são programados ou a interpretação pretendida (Arp; Smith; Spear, 2015).

Há ainda outros obstáculos para acesso, seleção e uso dos dados, tais como: (a) idiosincrasia humana: soluções pensadas apenas para legitimar uma percepção e não com base em uma compreensão objetiva e isenta da realidade; (b) torre de Babel: multiplicidade de significados, vocabulários e interpretações; (c) entrada e saída em sistemas: a baixa qualidade dos dados de entrada em um sistema geram resultados geralmente insatisfatórios, impactando inclusive na capacidade computacional de análise; (d) solipsismo em sistemas: quando a análise é baseada apenas em intuição (Almeida, 2021).

Nesse contexto, um elemento que pode servir como parte da solução é a ontologia, que pode ser entendida em dois contextos: um filosófico e outro tecnológico. Historicamente, o conceito remonta a Grécia Antiga, tendo em Aristóteles seu precursor. O termo em si foi adotado em obras da teoria escolástica (Almeida, 2021). No contexto filosófico, conforme (Arp; Smith; Spear, 2015), “a ontologia tem sido tradicionalmente definida como a teoria do que existe (ou do ‘ser enquanto ser’): o estudo dos tipos de entidades na realidade e das relações que essas entidades mantêm entre si”.

Nos últimos anos, o termo “ontologia” tem se tornado recorrente nas ciências da computação e informação. Nessa perspectiva, uma conceituação importante foi

feita por Tom Gruber em 1993, que descreve “ontologia como uma descrição, como uma especificação formal de um programa, dos conceitos e relações que podem existir para um agente ou uma comunidade de agentes” (Gruber, 1992). Uma associação importante de uma ontologia com sua realização prática é dada por Almeida (2021), destacando que ontologia é um “artefato formal, para fins de representação da informação e do conhecimento, que têm como vantagens a formulação rigorosa - livre de ambiguidades - de definições, assim como a possibilidade de implementação computacional”.

## **2.3. Projetos de Ontologias**

Para que a ontologia possa adquirir um caráter prático e aplicado, é importante que sejam estruturados projetos que deem suporte a sua implementação. O principal objetivo de um projeto é orquestrar boas práticas tanto conceituais quanto terminológicas, de forma a conseguir empreendimentos de sucesso. Soma-se aos princípios, o uso de metodologias adequadas para desenvolvimento, tais como a OntoForInfoScience que, por sua vez, reutiliza etapas oriundas de outras já existentes como a Methontology, o 101 Method e a NeOn (Almeida, 2021). Neste trabalho, foi utilizada a metodologia Pronto (Almeida, 2024).

### **2.3.1. Princípios e boas práticas de projeto**

Projetos de ontologia devem ser construídos a partir de referências conceituais sólidas, inerentes à origem e propósito da área. Nesse sentido, torna-se necessário fazer uso de uma série de princípios que possam nortear os trabalhos desenvolvidos. A tabela 1 apresenta princípios teórico-complementares que devem ser seguidos em projetos de ontologia.

Tabela 1 - Princípios teórico-complementares para projetos de ontologia

<b>Princípio</b>	<b>Descrição</b>
Realismo	O objetivo de uma ontologia é descrever a realidade. O realismo é a posição filosófica segundo a qual a realidade e seus constituintes existem independentemente de nossas visões (linguísticas, conceituais, teóricas ou culturais).
Perspectivismo	Este princípio reconhece que a realidade é complexa e variada. O Perspectivismo afirma que existem várias descrições igualmente precisas da realidade.
Falibilismo	O Falibilismo sustenta que as ontologias, assim como as teorias científicas que representam, são falíveis.
Adequatismo	Defende que as entidades em um domínio específico devem ser levadas a sério nos seus próprios termos e não devem ser vistas como redutíveis a outros tipos de entidades mais simples.
Reutilização	Ontologias já existentes devem ser tratadas como referências (benchmarks) e reutilizadas sempre que possível ao construir ontologias para novos domínios.
Utilidade	Não é razoável sacrificar princípios realistas por considerações de utilidade no curto prazo.
Atualização	As ontologias científicas estão sempre sujeitas à atualização a partir de avanços no conhecimento.
Facilidade	Ao projetar uma ontologia de domínio, o desenvolvedor deve começar identificando as características do assunto que são mais fáceis e claras de entender e definir.

Fonte: (Almeida, 2021).

Além dos princípios teórico-complementares faz-se necessário atentar para questões que impactarão cada etapa do projeto de ontologias, incluindo aspectos como seleção, definição e formatação de termos, além da criação de taxonomias. De forma a garantir a consistência e qualidade do projeto de ontologia, é proposto um conjunto de princípios, como os apresentados na tabela 2.

Tabela 2 - Princípios conceituais e terminológicos

<b>Categoria</b>	<b>Princípio</b>
Seleção de termos	1. Incluir termos usados por grupos de cientistas ou profissionais influentes para representar universais.
Seleção de termos	2. Buscar consenso de cientistas ou profissionais para o uso de termos, o que envolve a discussão entre grupos de especialistas.
Seleção de termos	3. Identificar regiões de sobreposição interdisciplinar e verificar onde o uso dos termos não é consistente.
Seleção de termos	4. Manter tanto quanto possível os termos usados por especialistas, fazendo uso de recursos existentes.
Formatação de termos	5. Usar substantivos singulares.
Formatação de termos	6. Usar minúsculas para substantivos comuns.
Formatação de termos	7. Evitar acrônimos.
Formatação de termos	8. Associar cada termo a um identificador alfanumérico único.
Formatação de termos	9. Garantir a univocidade dos termos.
Formatação de termos	10. Garantir a univocidade das expressões de relacionamentos.
Formatação de termos	11. Evitar substantivos incontáveis.
Definições de termos	12. Fornecer todos os termos com definições, exceto o raiz.
Definições de termos	13. Usar definições no formato aristotélico (S é um G que D).
Definições de termos	14. Usar características essenciais para definir termos.
Definições de termos	15. Iniciar pelos termos mais gerais do domínio.
Definições de termos	16. Evitar circularidade ao definir termos.



Definições de termos	17. Ao definir, usar termos mais simples que o termo sob definição.
Definições de termos	18. Evitar criar termos para universais por combinação lógica.
Criação de taxonomias	19. Fundamentação ontológica e classificação por características.
Criação de taxonomias	20. Estrutura: níveis, é-um consistente e herança única.
Criação de taxonomias	21. Disjunção: entidades no mesmo nível são disjuntas.
Criação de taxonomias	22. Exaustividade: classificar tanto quanto possível do domínio.
Criação de taxonomias	23. Rigor conceitual: não adotar termos ambíguos.
Criação de taxonomias	24. Uniformidade: critérios de classificação consistentes.
Criação de taxonomias	25. Clareza e precisão: não adotar termos ou categorias vagas.
Criação de taxonomias	26. Meta-categorias: não adotar as que dependam de si mesmas.

Fonte: (Almeida, 2021).

### 2.3.2. Método para desenvolvimento

Além dos princípios elencados, um projeto de ontologia efetivo e de qualidade, deve seguir uma estrutura metodológica em que as etapas e respectivos passos sirvam de guia para os trabalhos rumo ao sucesso do empreendimento. A tabela 3 apresenta a metodologia PRONTO (Almeida, 2024).

Tabela 3 - Etapas e passos da metodologia PRONTO

<b>Etapas</b>	<b>Passos</b>
1. Especificação	1.1. Estabelecer questões de competência. 1.2. Estabelecer um conjunto de metadados. 1.3. Produzir o documento de especificação.
2. Aquisição de conhecimento	2.1. Extrair termos de outros artefatos. 2.2. Extrair termos de outros documentos. 2.3. Extrair termos elicitados de especialistas. 2.4. Registrar termos obtidos. 2.5. Criar modelo preliminar.
3. Organização de termos representativos de entidades	3.1. Importar termos genéricos. 3.2. Inserir termos de domínio. 3.3. Organizar taxonomia preliminar.
4. Organização de termos representativos de relacionamentos	4.1. Selecionar e inserir relacionamentos padrão. 4.2. Criar e inserir relacionamentos adicionais.
5. Definição de entidades	5.1. Aplicar método de definições. 5.2. Registrar definições em linguagem natural. 5.3. Registrar conjunto de metadados.
6. Formalização	6.1. Produzir axiomas. 6.2. Classificar a ontologia.
7. Avaliação	7.1. Avaliar para ajustes. 7.2. Avaliar para resultados.
8. Documentação	8.1. Reunir material usado. 8.2. Criar um documento resumo da ontologia.

Fonte: (Almeida, 2024).

A seção a seguir apresenta os procedimentos metodológicos adotados com base no uso da metodologia selecionada.

### **3. Procedimentos metodológicos**

O projeto de criação da ontologia ocorreu no contexto de uma empresa de médio porte e com histórico de atuação de mais de 30 anos. A organização está ligada ao ramo de desenvolvimento de soluções de software cujo principal serviço é a oferta de um produto para um domínio específico. A sede da empresa fica na cidade de Belo Horizonte, Minas Gerais. A partir deste ponto, a empresa será referenciada como Empresa XPTO.

O projeto de ontologia nasceu dentro de outra iniciativa que é o projeto de melhoria dos processos de software e serviços de tecnologia da informação da Empresa XPTO, baseados nos modelos de referência de software (MR-SW) e serviços (MR-SV), constituintes do programa de Melhoria do Processo de Software Brasileiro (MPS-BR) e promovidos pela Softex (<https://softex.br>).

Os trabalhos específicos ligados à ontologia tiveram uma duração de 30 dias, considerando o prazo decorrido entre a identificação do problema e a execução de todos os passos metodológicos propostos para a obtenção da ontologia. Para coleta de informações, utilizou-se como base o conhecimento previamente adquirido durante o projeto de melhoria, além de acesso a documentação técnica de processos e procedimentos e da ferramenta Redmine (<https://redmine.org>), em uma versão específica instalada nos domínios da Empresa XPTO. Participaram diretamente da iniciativa, o autor deste trabalho, no papel de consultor em melhoria de processos e a equipe responsável pelo grupo de processos da Empresa XPTO, além de outros profissionais entrevistados sob demanda.

As seções a seguir apresentam os principais produtos resultantes da execução das etapas da metodologia proposta.

#### **3.1. Especificação**

As principais informações geradas nesta etapa estão associadas ao propósito do projeto, o domínio da ontologia e os respectivos requisitos, conforme apresentado na figura 2.

Figura 2 - Documento de Especificação de Ontologia

## Propósito

---

- Criar e manter a ontologia relativa aos processos de suporte e manutenção de software, de forma a apoiar a evolução das ferramentas de gestão.

## Escopo e domínios

---

- Processos de suporte e manutenção de software da empresa XPTO ligados ao produto XYZ.

## Linguagem de implementação e grau de formalidade

---

- Linguagem: OWL 2.
- Grau de formalidade: baixo.

## Público-alvo e cenários de usos pretendidos

---

- Gestores de processo: manter a ontologia e garantir implementação nas ferramentas de gestão.
- Usuários dos processos: validar e aplicar a implementação da ontologia refletida nas ferramentas.

## Requisitos da ontologia

---

### › Requisitos não funcionais

- A ontologia deve ser mapeada e implementada na ferramenta de gestão de demandas Redmine (<https://redmine.org>).
- A ontologia deve ser documentada por meio do Protégé (<https://protege.stanford.edu>).
- Os diagramas complementares devem ser documentados por meio do Draw.io (<https://app.diagrams.net/>).
- Os produtos de trabalho devem ser mantidas sob controle de versão por meio do GitHub (<https://github.com>).
- As informações complementares devem ser mantidas no Google Workspace (<https://workspace.google.com>).

### Requisitos funcionais (questões de competência)

- Quais as entidades e respectivos tipos de relacionamentos?
- Quais são os tipos de uma entidade?
- Que informações permitem acompanhar o ciclo de vida de uma entidade?
- Quais as ações constituem o ciclo de vida de uma entidade?
- Quais os estados e transições permitidas para uma entidade?

## Metadados

---

- rdfs:comments (definição das classes), dc:creator (quem criou a definição), dc:source (referência para a definição).

Fonte: autor - com base no exemplo de (Almeida, 2024).

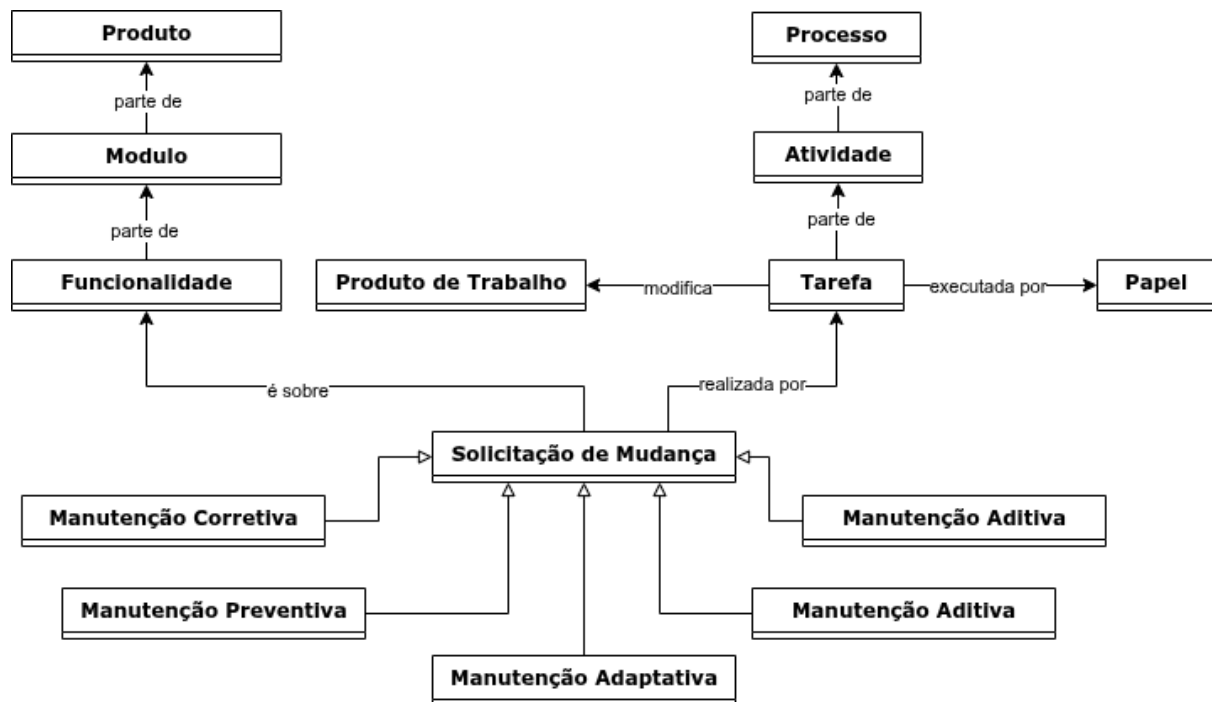
### 3.2. Aquisição de conhecimento

Os artefatos de representação foram obtidos por meio das definições de solicitações de mudança e atividades técnicas providas pela norma ISO/IEC/IEEE 14764:2022 (ISO, 2022), além de informações contidas nos guias de software (Softex, 2024) e serviços (Softex, 2024b) do MPS.BR.

Para registro de termos de artefatos de representação foi feita uma busca inicial superficial sobre ontologias específicas relacionadas ao tema, tendo sido identificada a *Representation Formalism for Software Engineering Ontologies* (REFSENO), que serviu de referência para o entendimento de "subontologias" (RuizZ et al., 2011). Como resultado, foi possível criar uma organização inicial dos grupos conceituais distintos de solicitação de mudança, produtos de trabalho, processo e papéis.

Para aquisição do conhecimento de especialistas, foram feitas entrevistas com os profissionais responsáveis pelo processo de software da empresa para validar a implementação original, disponível no Redmine. Os termos compilados foram registrados em uma planilha, disponível no repositório do projeto. Para fins de melhor representação das informações inicialmente coletadas, foi criada a figura 3, contendo um modelo preliminar das entidades, usando a simbologia de um diagrama de classes da *Unified Modeling Language*, UML.

Figura 3 - Modelo preliminar de entidades



Fonte: autor.

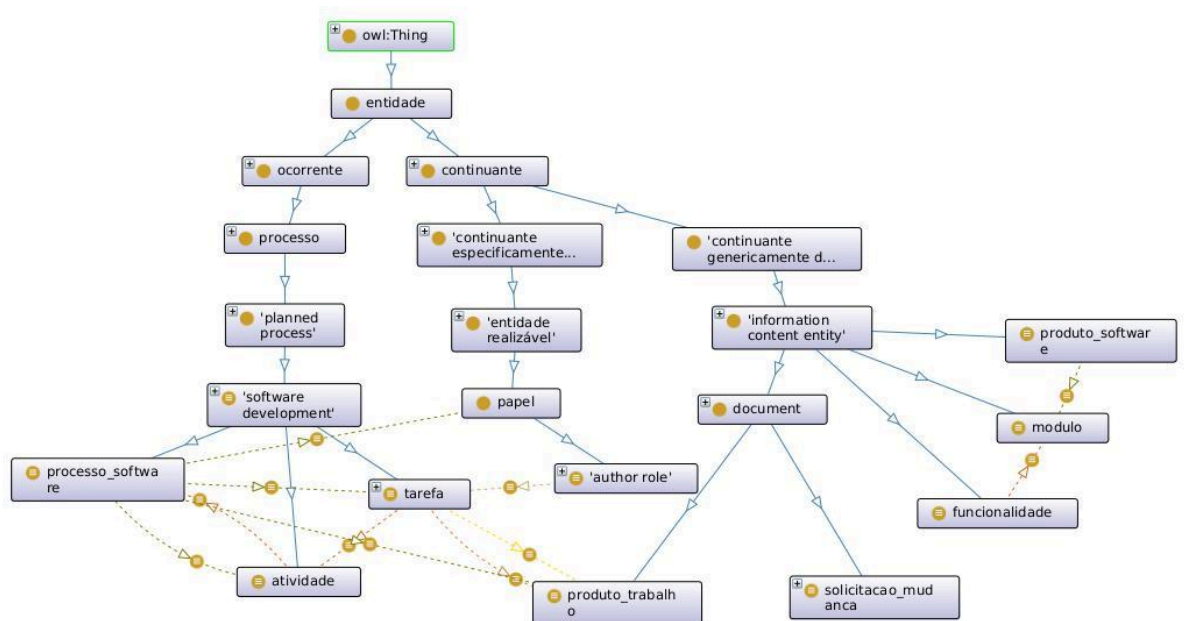
### 3.3. Organização de termos representativos de entidades

Os termos genéricos foram incorporados ao projeto por meio da importação da Basic Formal Ontology (BFO) para o Protégé, disponível em: <https://github.com/NCOR-BR/Framework-BFO>. Também foi utilizada a Informational Artifact Ontology (IAO), criada com o objetivo de servir como um recurso livre de domínios para a representação de entidades de conteúdo de informação, information content entities - ICE, como documentos, bancos de dados e imagens digitais (Smith; Ceusters, 2015), disponível em <https://github.com/information-artifact-ontology/IAO>.

### 3.4. Organização de termos representativos de relacionamentos

Após a importação tanto da BFO, IAO e da planilha com as classes identificadas na etapa anterior, foi feita uma estruturação no Protégé, conforme mostrado na figura 4. Foram utilizados os relacionamentos padrão providos pela BFO. Para fins de simplificação, não foram incluídos relacionamentos adicionais.

Figura 4 - Organização das entidades no Protégé



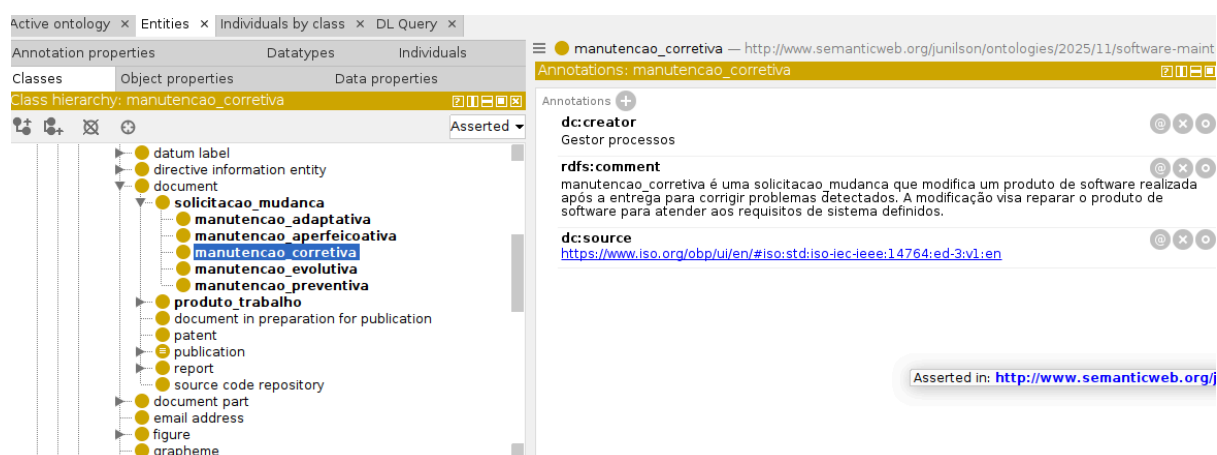
Fonte: autor.

### 3.5. Definição de entidades

Foram criadas definições para todas entidades, conforme o método aristotélico, que segue a seguinte estrutura: "S é um G que D", onde S - representa a espécie, G - representa o gênero e D - representa o diferencial da espécie.

Em seguida, as definições em linguagem natural foram registradas na view Annotations, metadados rdfs:comments, fonteDefinicao e responsavelDefinicao, conforme exemplo ilustrativo apresentado na figura 5. Para fins de simplificação, nesta primeira versão do projeto, não foram adicionados outros metadados.

Figura 5 - Exemplo de definição de entidade no Protégé



Fonte: autor

### 3.6. Formalização

Foram produzidos axiomas para explicitar as dependências entre as entidades com a descrição seguindo a sintaxe manchester, conforme apresentado na tabela 4.

Tabela 4 - Principais entidades e relacionamentos.

Entidade	Principais Relacionamentos
solicitacao_mudanca	('realized in' some tarefa) and ('is about' some funcionalidade)
processo_software	('has part' some atividade) and ('has part' some tarefa) and ('has part' some produto_trabalho) and ('has part' some papel)
atividade	part of some processo_software
desenvolver_solucão	('part of' some desenvolver_solucão) and ('preceded by' some planejar_entrega)
tarefa	('part of' some atividade) and (has_specified_input some produto_trabalho) and (has_specified_output some produto_trabalho) and ('has participant' some papel)
produto_trabalho	participates in' some tarefa
papel	participates in' some tarefa
produto_software	has part' some modulo
modulo	part of' some produto_software
funcionalidade	part of' some modulo

Fonte: autor.



No cadastro do Protegé foram incluídos todos relacionamentos, exceto nos casos onde havia possibilidade de anotações bidirecionais. Por exemplo, entre a entidade 'processo\_software' e a entidade 'atividade', há os relacionamentos:

- a) processo\_software 'has part' some atividade e
- b) atividade 'participates in' some processo\_software.

Nesses casos, optou-se por anotar apenas os relacionamentos das entidades que possuem o relacionamento 'participates in'). De forma análoga, em se tratando dos relacionamentos da entidade 'tarefa' com 'papel' e 'produto\_trabalho', optou-se por registrar os relacionamentos na entidade 'tarefa'.

### **3.7. Avaliação**

Foi feita a avaliação a partir da execução do reasoner. Para tanto, foi configurado o mecanismo HermiT.

### **3.8. Documentação**

Os artefatos produzidos neste projeto estão no repositório do GitHub (<https://github.com/valuedriven/software-main-ontology>). Para fins de simplificação, não foi criado nenhum mecanismo para publicação e acesso ao conteúdo da ontologia, além dos artefatos disponíveis no repositório e que podem ser acessados por meio do Protégé.

## **4. Resultados**

Um importante resultado obtido no desenvolvimento do projeto foi a possibilidade de se organizar de maneira sistemática as entidades que, inicialmente, estavam alocadas de maneira ad-hoc, dificultando o entendimento dos relacionamentos e hierarquias de classes.

Outro resultado importante foi a definição rigorosa de cada entidade, o que forçou a busca de literatura especializada, permitindo o alinhamento e consenso sobre a diversidade de assuntos existentes em iniciativas dessa natureza.

Foi possível ainda, por meio da execução do projeto, corrigir inconsistências significativas na implementação original no Redmine, como a existência de entidade “Outros”, a eliminação de algumas ocorrências de herança múltipla entre entidades, além da eliminação de entidades consideradas “fantasmas”, que não traziam nenhum ganho para a implementação original.

## **5. Discussão**

O desenvolvimento do projeto de ontologia reforça um aspecto central prometido em iniciativas dessa natureza e já assinalados nos resultados que foi a padronização da terminologia empregada. Os ganhos obtidos com esse resultado vão além de apenas “usar termos comuns aos envolvidos” mas possibilita um melhor entendimento do processo de manutenção de software.

Um achado importante diz respeito a utilização do projeto de ontologia para obter um esclarecimento conceitual relativo às entidades. Como exemplo, a partir do uso da ontologia que foi possível separar clara e conceitualmente as classes solicitação de serviço e processo obtendo, com isso, maior consistência, modularização e organização da informação.

Um fator que potencializou o desenvolvimento do projeto foi a experiência e conhecimento do autor em projetos de melhoria de processos de software, além do conhecimento do contexto empresarial.

Dentre os aspectos que dificultaram o desenvolvimento do projeto, encontram-se a inexperiência do autor em iniciativas dessa natureza. Dentre as etapas desenvolvidas, a que mais foi afetada por essa limitação foi a avaliação. Apesar dessa limitação, os resultados ora obtidos são potencialmente aplicáveis na implementação das mudanças na ferramenta.

Outra limitação deveu-se ao escopo inicial proposto. Como foi feito um foco apenas no processo de manutenção de software, algumas atividades relevantes para a sustentação do produto de software ficaram de fora, tais como a gestão de serviços de tecnologia da informação, o que pode ser complementado nas ações futuras por meio da inclusão de referências como as normas ISO relativas ao assunto e o framework ITIL.

## **6. Considerações finais**

Com base nos resultados e discussões apresentados, pode ser considerado que este trabalho atingiu o objetivo inicialmente traçado que foi criar um projeto de ontologia para mapear um processo de manutenção de software que pudesse ser implementado em uma ferramenta computacional, endereçando os problemas relatados pelas diversas partes interessadas.

Como proposta de ações que podem ser desenvolvidas a partir deste trabalho, destacam-se: a) realizar a avaliação para resultados, especialmente a partir da revisão e implementação de consultas relativas às questões de competência; b) usar o projeto de ontologia como base para realizar análise de impacto da migração de versão do Redmine em função da elevada quantidade de campos personalizados; c) efetivar a implementação da ontologia na ferramenta Redmine; d) criar um documento resumo da ontologia provendo uma forma de consulta rápida a ontologia, utilizando, por exemplo, a ferramenta Widoco.

## Referências

ALMEIDA, Maurício Barcellos. **Ontologia em ciência da informação: teoria e método : coleção representação do conhecimento em ciência da informação**. 1ª ed. [S.l.]: CRV, 2021. v. 1

ALMEIDA, Maurício Barcellos. **Ontologia na prática – projeto, metodologia e construção**. Curitiba, PR: Editora CRV, 2024. v. 01

ARP, Robert; SMITH, Barry; SPEAR, Andrew D. **Building Ontologies with Basic Formal Ontology**. Cambridge, Massachusetts: The MIT Press, 2015.

GRUBER, Tom. **What is an Ontology?** Disponível em: <<http://www-ksl.stanford.edu/kst/what-is-an-ontology.html>>. Acesso em: 24 nov. 2025.

ISO. **ISO/IEC/IEEE 14764:2022(en), Software engineering - Software life cycle processes - Maintenance**. Disponível em: <<https://www.iso.org/obp/ui/en/#iso:std:iso-iec-ieee:14764:ed-3:v1:en>>. Acesso em: 27 nov. 2025.

RUIZZ, Francisco *et al.* An Ontology For The Management Of Software Maintenance Projects. **International Journal of Software Engineering and Knowledge Engineering**, 21 nov. 2011.

SMITH, Barry; CEUSTERS, Werner. Aboutness: Towards Foundations for the Information Artifact Ontology. 2015.

SOFTEX. **Guia Geral MPS de Software**. , 2024. Disponível em: <<https://softex.br/download/guia-geral-mps-de-software2024/>>. Acesso em: 3 dez. 2025

SOFTEX. **Guia Geral MPS de Serviços**. , 2024b. Disponível em: <<https://softex.br/download/guia-geral-mps-de-servicos2024/>>. Acesso em: 3 dez. 2025