



Universidade Federal do Maranhão
Engenharia da Computação
Curso de Inteligência Artificial
Prof. Dr. Thales Levi Azevedo Valente

Engenharia do Conhecimento: Ontologia

Integrantes:

Gabriel Felipe Carvalho Silva
Giordano Bruno De Araujo Mochel
Judson Rodrigues Ciribelli

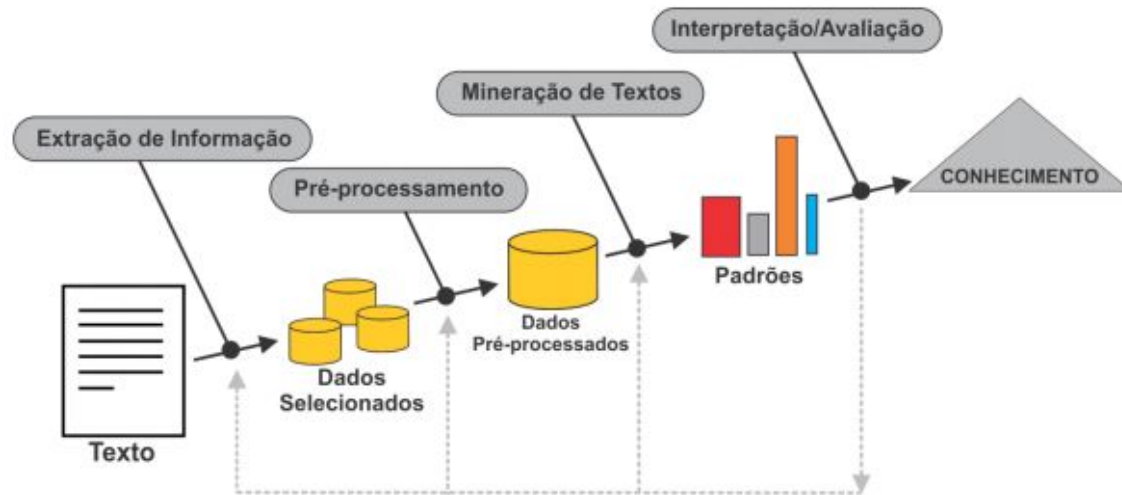
Sumário



1. Engenharia do Conhecimento
2. O que é Ontologia?
3. Estrutura de uma Ontologia
4. Áreas de Atuação
5. Casos de Uso da Ontologia
6. Ferramentas para Construção de Ontologias
7. Linguagens para Ontologias
8. Exemplos Práticos (Código)
9. Exemplos Práticos (Protegé)

1 Engenharia do Conhecimento

É o campo que trata da **aquisição, modelagem, organização e aplicação de conhecimento** para criar **sistemas inteligentes** que simulam o **raciocínio humano**.



Fonte: <https://ecufs-gisi.blogspot.com/2015/01/definicao-das-ferramentas-da-engenharia.html>

2 O que é Ontologia ?



Ontologia é uma **representação formal** de um conjunto de **conceitos** e das **relações** entre eles, utilizada para estruturar e organizar o **conhecimento** dentro de um **domínio** específico.

Origem do termo

Na computação:

- O conceito foi adaptado nos anos 1990 para a ciência da computação e a inteligência artificial.
- O termo foi adaptado para descrever uma forma de representar informações e conceitos que podem ser processados por **sistemas computacionais**.

3 Estrutura de uma Ontologia



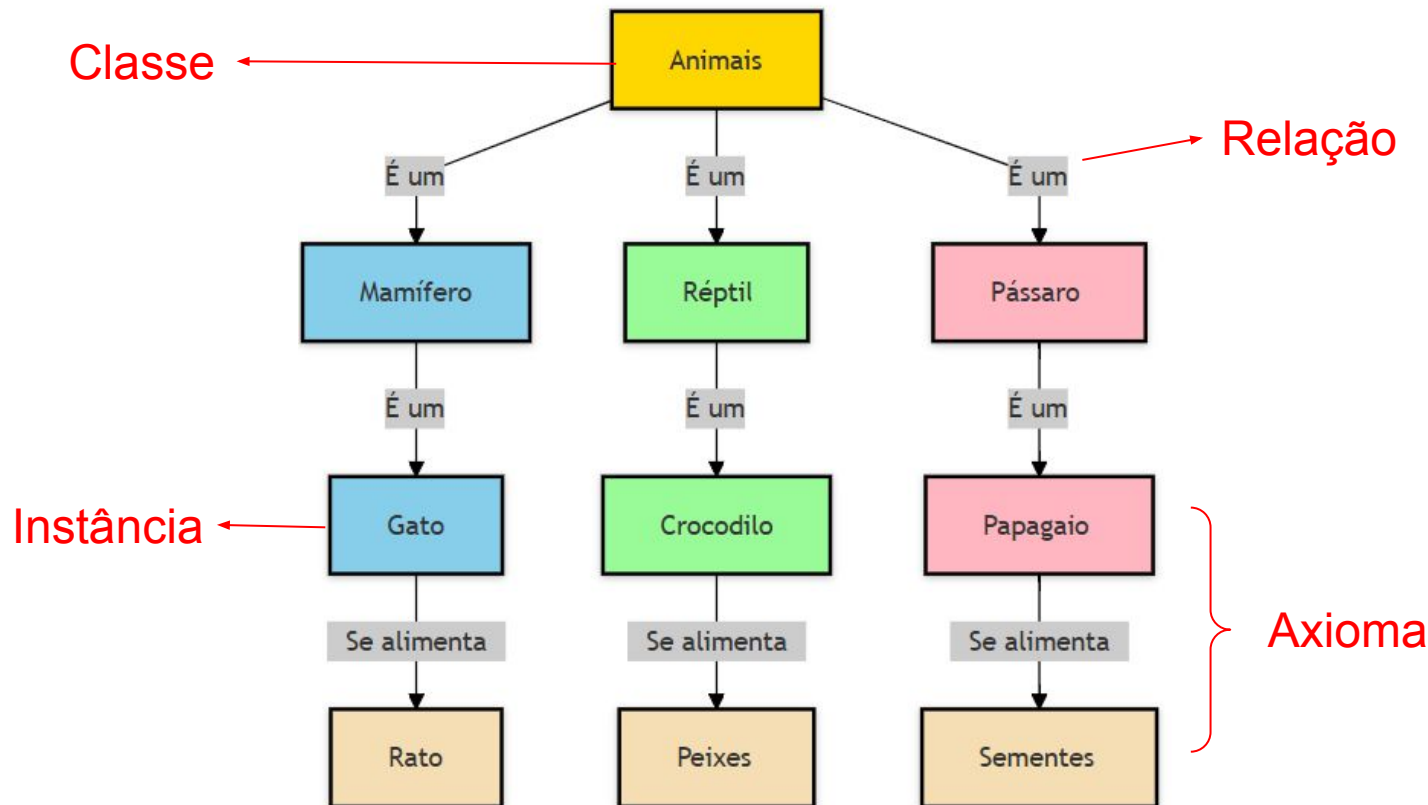
Classes ou Conceitos: Representam as entidades ou classes dentro de um domínio. Por exemplo, "Animal", "Mamífero", "Pássaro".

Relações: Definem como os conceitos estão conectados. Exemplos incluem "é_um" (relações hierárquicas) ou "se_alimenta_de" (relações funcionais).

Axiomas: Regras ou restrições que descrevem propriedades do domínio. Exemplo: "Todo mamífero é um animal".

Instâncias: São os exemplos concretos dos conceitos. Por exemplo, "Baleia" como instância do conceito "Mamífero".

3 Estrutura de uma Ontologia

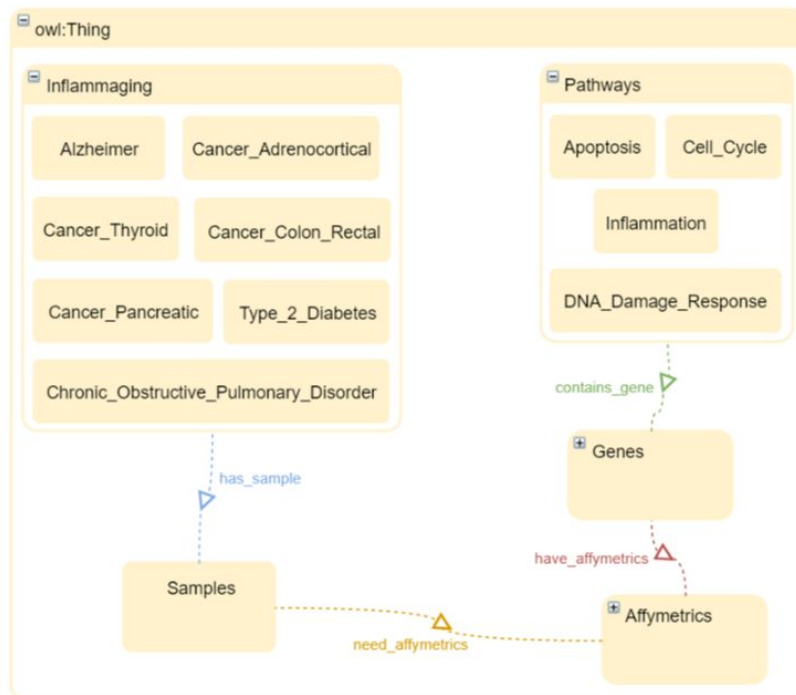


4 Áreas de Atuação

- I. Recuperação de informações na Internet.
- II. Processamento de Linguagem Natural.
- III. Gestão do Conhecimento.
- IV. Web-Semântica.
- V. Educação.

5 Casos de uso da Ontologia

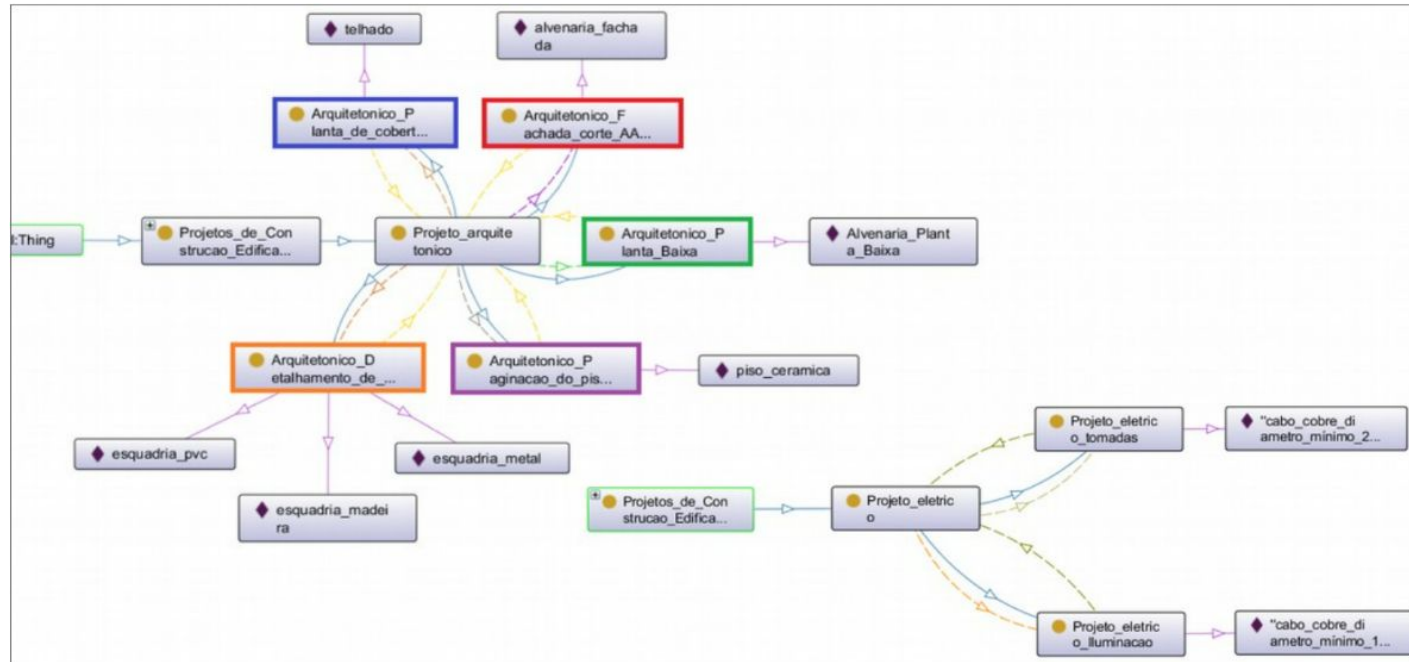
I. Engenharia Ontológica aplicada para o projeto e especificação da ontocancro (2019).



A antologia utilizada para anotar **características fenotípicas** e **genéticas de patologias** por meio da definição de **conceitos** e **relacionamentos semânticos** (v. 3).

5 Casos de uso da Ontologia

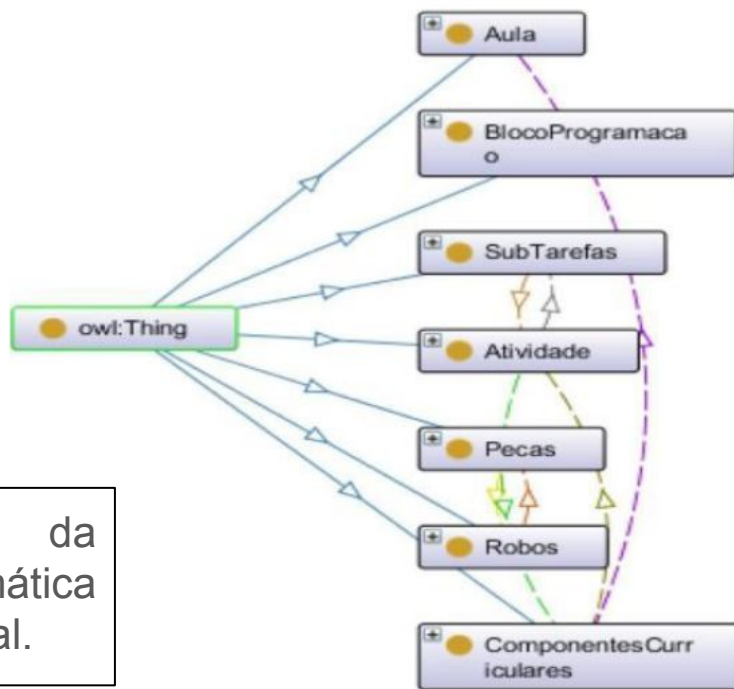
II. Modelo de Representação em Ontologia no Gerenciamento de Projetos de Edificações na Construção Civil pelos Sistemasontokem e Protégé (2021).



Proj Arquit
e eletrico.

5 Casos de uso da Ontologia

III. Uma Ontologia Para Apoiar O Ensino De Matemática Básica Com Uso De Robótica Educacional (2023).



Ensino e aprendizagem da disciplina de matemática utilizando robótica educacional.

6 Ferramentas para Construção de Ontologias

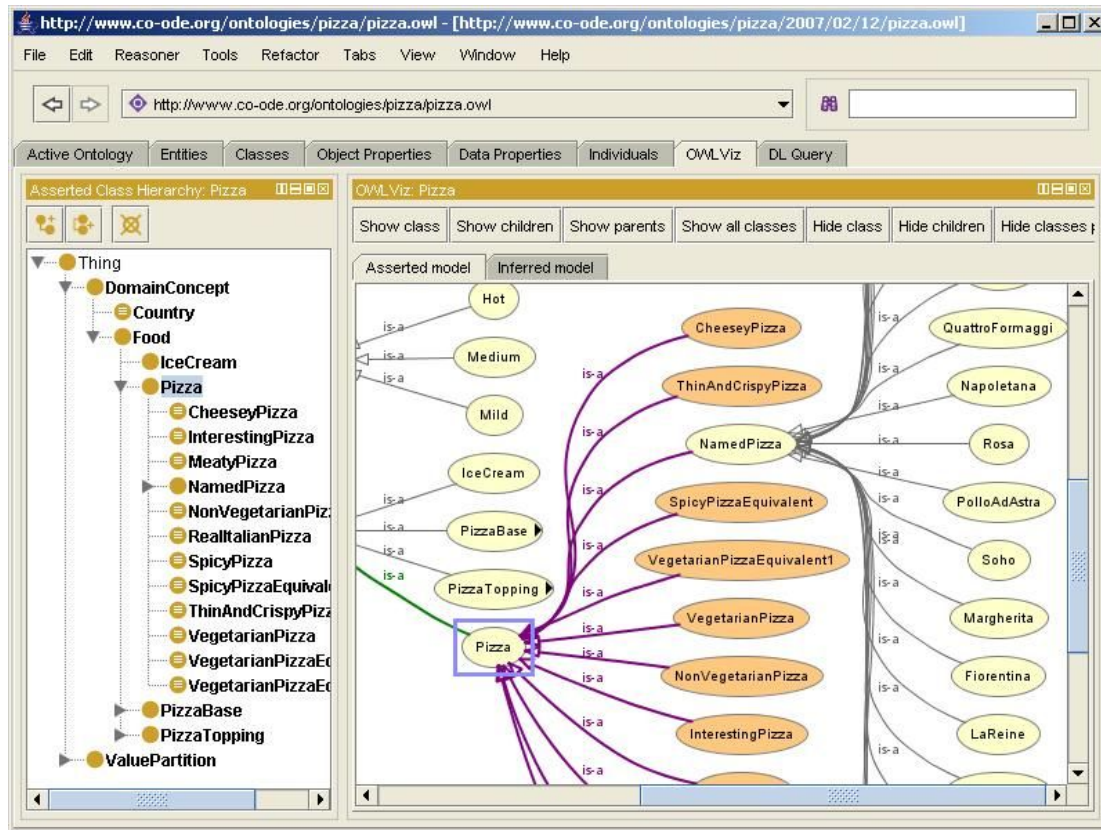
SEMPRE+ QUALIDADE INOVAÇÃO INCLUSÃO



I. Protégé



<https://protege.stanford.edu/>



6 Ferramentas para Construção de Ontologias

I. WebProtege



<https://webprotege.stanford.edu/>

WebProtégé

localhost:8080/webprotege_obs/WebProtege.html#Cerveau

You are signed out. | [Sign In](#)

My WebProtégé: Cerveau

MindMap | Classes | Properties | Individuals | Notes and Discussions

Add content to this tab | Add tab

Classes

Create Delete Watch Branch Search: Type

- owl:Thing
 - Acronym_Class
 - Artefact_Class
 - Brain_function_Class
 - Brain_Material_Class
 - Document_Class
 - Book_Class
 - Neural_Disease_Class
 - NeuralArea_Class
 - People_Class
 - Phenomen_Class
 - Symptom_Class
 - Term_Class

Graph for Erreur_de_Descartes

Book_Class

is_published_by

is_about

Emotion

Decision

Image_URL

IMG/Erreur_de_I

Emotion

Decision

Damasio_Antonio

has_description_in

publish

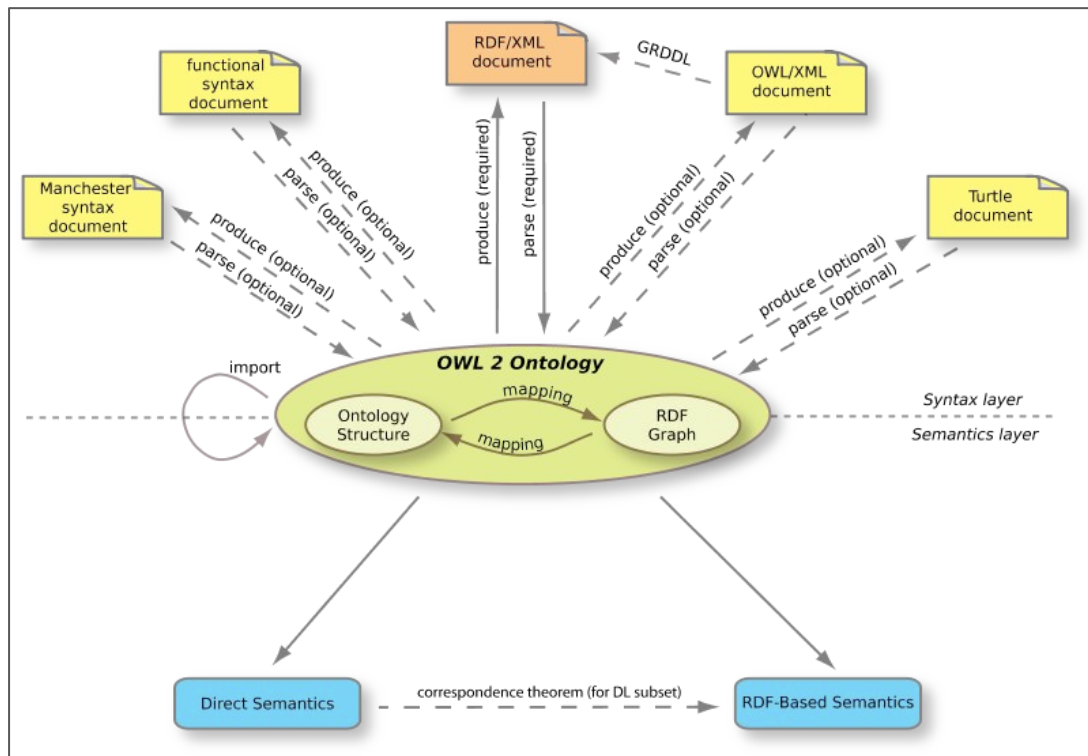
Erreur_de_Descartes

7 Linguagens para Ontologias

I. OWL (Web Ontology Language)

```
<owl:AllDifferent>
  <owl:distinctMembers rdf:parseType="Colecao">
    <vin:VinhoCor rdf:about="#Tinto" />
    <vin:VinhoCor rdf:about="#Branco" />
    <vin:VinhoCor rdf:about="#Rose" />
  </owl:distinctMembers>
```

<https://www.w3.org/TR/owl2-overview/>



7 Linguagens para Ontologias

II. Python

```
>>> from owlready2 import *  
>>> onto_path.append("/path/to/your/local/ontology/repository")  
>>> onto = get_ontology("http://www.lesfleursdunormal.fr/static/_downloads/pizza_onto.owl")  
>>> onto.load()
```

Lib:
<https://pypi.org/project/owlready2/>

```
>>> with onto:  
...     class NonVegetarianPizza(onto.Pizza):  
...         equivalent_to = [  
...             onto.Pizza  
...             & ( onto.has_topping.some(onto.MeatTopping)  
...                 | onto.has_topping.some(onto.FishTopping)  
...             ) ]  
...         def eat(self): print("Beurk! I'm vegetarian!")
```

Exemplos Práticos (Código)

8 Exemplos Práticos (Código)

- **owlready2:** Biblioteca Python usada para criar e manipular ontologias em formato OWL.
- A variável `onto` cria uma nova ontologia chamada **full_health.owl**.
- O endereço **http://example.org/full_health.owl** é usado como a identificação única da ontologia.

```
logia.py > ExameImagem
from owlready2 import *

# Criar um novo arquivo de ontologia
onto = get_ontology("http://example.org/full_health.owl")
```


8 Exemplos Práticos (Código)

- **with onto** Define um contexto no qual todas as classes, propriedades e axiomas criados serão automaticamente associados à ontologia **onto**.
- As classes principais **Paciente**, **Sintoma**, **Doenca**, **Exame** e **Tratamento** são criadas.
- Todas elas herdam de **Thing**, a classe base de todos os conceitos na ontologia.

```
with onto:  
    ### **1. Definir Classes**  
    class Paciente(Thing): pass  
    class Sintoma(Thing): pass  
    class Doenca(Thing): pass  
    class Exame(Thing): pass  
    class Tratamento(Thing): pass
```

- **pass** é usado para declarar as classes sem definir comportamento ou propriedades adicionais no momento.

8 Exemplos Práticos (Código)

- **SintomaRespiratorio**, **SintomaCardiaco** e **SintomaGastrointestinal** são subclasses da classe **Sintoma**.
- **DoencaInfecciosa**, **DoencaCronica**, **DoencaGastrointestinal** e **DoencaRespiratoria** são subclasses da classe **Doenca**.
- **ExameImagem**, **ExameLaboratorial** e **ExameFuncional** são subclasses da classe **Exame**.
- **TratamentoMedicamentoso** e **TratamentoNaoMedicamentoso** são subclasses da classe **Tratamento**.

```
### **2. Subclasses**  
# Sintomas  
class SintomaRespiratorio(Sintoma): pass  
class SintomaCardiaco(Sintoma): pass  
class SintomaGastrointestinal(Sintoma): pass  
  
# Doenças  
class DoencaInfecciosa(Doenca): pass  
class DoencaCronica(Doenca): pass  
class DoencaGastrointestinal(Doenca): pass  
class DoencaRespiratoria(Doenca): pass  
  
# Exames  
class ExameImagem(Exame): pass  
class ExameLaboratorial(Exame): pass  
class ExameFuncional(Exame): pass  
  
# Tratamentos  
class TratamentoMedicamentoso(Tratamento): pass  
class TratamentoNaoMedicamentoso(Tratamento): pass
```

8 Exemplos Práticos (Código)

- Conecta: **Paciente** → **Sintoma**.
- Conecta: **Paciente** → **Doença**.
- Conecta: **Sintoma** → **Exame**.
- Conecta: **Doença** → **Exame**.

```
### **3. Propriedades**  
class apresenta(ObjectProperty):  
    domain = [Paciente]  
    range = [Sintoma]  
  
class diagnosticado_com(ObjectProperty):  
    domain = [Paciente]  
    range = [Doenca]  
    is_functional = True # Um paciente só pode ser diagnosticado com uma doença por vez  
  
class requer_exame(ObjectProperty):  
    domain = [Sintoma]  
    range = [Exame]  
  
class confirmada_por(ObjectProperty):  
    domain = [Doenca]  
    range = [Exame]  
    is_transitive = True # Propriedade transitiva  
  
class tratada_com(ObjectProperty):  
    domain = [Doenca]  
    range = [Tratamento]
```

8 Exemplos Práticos (Código)

- Todo paciente deve apresentar pelo menos **um sintoma**.
- Todo sintoma deve estar relacionado a pelo menos **um exame**.
- Sintomas respiratórios requerem **exames de imagem**.
- Sintomas cardíacos requerem **exames funcionais ou laboratoriais**.
- Sintomas gastrointestinais requerem **exames laboratoriais ou de imagem**.
- Toda doença deve ser confirmada por pelo menos **um exame**.

```
### **4. Adicionar Axiomas**
Paciente.equivalent_to = [
    Thing & apresenta.some(Sintoma) # Todo paciente deve apresentar pelo menos um sintoma
]
Sintoma.equivalent_to = [
    Thing & requer_exame.some(Exame) # Todo sintoma deve estar relacionado a pelo menos um exame
]
SintomaRespiratorio.equivalent_to = [
    Sintoma & requer_exame.some(ExameImagem) # Sintomas respiratórios requerem exames de imagem
]
SintomaCardiaco.equivalent_to = [
    Sintoma & requer_exame.some(ExameFuncional | ExameLaboratorial) # Sintomas cardíacos requerem exames funcionais ou laboratoriais
]
SintomaGastrointestinal.equivalent_to = [
    Sintoma & requer_exame.some(ExameImagem | ExameLaboratorial) # Sintomas gastrointestinais requerem exames laboratoriais ou de imagem
]
Doenca.equivalent_to = [
    Thing & confirmada_por.some(Exame) # Toda doença deve ser confirmada por pelo menos um exame
]
```

8 Exemplos Práticos (Código)

- Doenças infecciosas requerem **exames laboratoriais**.
- Doenças crônicas requerem **exames laboratoriais ou funcionais**.
- Doenças gastrointestinais requerem **exames laboratoriais ou de imagem**.
- Doenças respiratórias requerem **exames de imagem**.
- Todo tratamento deve estar relacionado a pelo menos **uma doença**.
- Tratamentos medicamentosos estão relacionados a **doenças confirmadas**.
- Tratamentos não medicamentosos tratam **doenças crônicas ou respiratórias**.

```
DoencaInfecciosa.equivalent_to = [  
  Doenca & confirmada_por.some(ExameLaboratorial) # Doenças infecciosas requerem exames laboratoriais  
]  
  
DoencaCronica.equivalent_to = [  
  Doenca & confirmada_por.some(ExameLaboratorial | ExameFuncional) # Doenças crônicas requerem exames laboratoriais ou funcionais  
]  
  
DoencaGastrointestinal.equivalent_to = [  
  Doenca & confirmada_por.some(ExameImagem | ExameLaboratorial) # Doenças gastrointestinais requerem exames laboratoriais ou de imagem  
]  
  
DoencaRespiratoria.equivalent_to = [  
  Doenca & confirmada_por.some(ExameImagem) # Doenças respiratórias requerem exames de imagem  
]  
  
Tratamento.equivalent_to = [  
  Thing & tratada_com.some(Doenca) # Todo tratamento deve estar relacionado a pelo menos uma doença  
]
```

8 Exemplos Práticos (Código)

- Febre, Tosse, Falta de ar pertencem à classe SintomaRespiratorio.
- Dor no peito pertence à classe SintomaCardiaco.
- Náuseas e Diarreia pertencem à classe SintomaGastrointestinal.
- Raio-X, Tomografia, Endoscopia pertencem à classe ExameImagem.
- Hemograma, Gasometria pertencem à classe ExameLaboratorial.
- Teste de Esforço pertence à classe ExameFuncional.

```
### **5. Instâncias**  
  
# Sintomas  
febre = SintomaRespiratorio("Febre")  
tosse = SintomaRespiratorio("Tosse")  
dor_no_peito = SintomaCardiaco("Dor no peito")  
nauseas = SintomaGastrointestinal("Náuseas")  
diarreia = SintomaGastrointestinal("Diarreia")  
falta_ar = SintomaRespiratorio("Falta de ar")  
  
# Exames  
raio_x = ExameImagem("Raio-X")  
tomografia = ExameImagem("Tomografia Computadorizada")  
hemograma = ExameLaboratorial("Hemograma Completo")  
gasometria = ExameLaboratorial("Gasometria")  
teste_esforco = ExameFuncional("Teste de Esforço")  
endoscopia = ExameImagem("Endoscopia Digestiva")  
  
# Doenças  
covid19 = DoencaInfecciosa("COVID-19")  
pneumonia = DoencaRespiratoria("Pneumonia")  
hipertensao = DoencaCronica("Hipertensão")  
gastrite = DoencaGastrointestinal("Gastrite")  
insuficiencia_respiratoria = DoencaRespiratoria("Insuficiência Respiratória")  
  
# Tratamentos  
isolamento = TratamentoNaoMedicamentoso("Isolamento")  
antiviral = TratamentoMedicamentoso("Antiviral")  
dieta_baixa_acidez = TratamentoNaoMedicamentoso("Dieta Baixa em Acidez")  
antibiotico = TratamentoMedicamentoso("Antibiótico")  
suplemento_oxigenio = TratamentoNaoMedicamentoso("Suplemento de Oxigênio")  
anti_hipertensivo = TratamentoMedicamentoso("Medicamento Anti-hipertensivo")  
  
# Pacientes  
paciente1 = Paciente("Joao")  
paciente2 = Paciente("Maria")
```


8 Exemplos Práticos (Código)

- COVID-19 pertence à classe DoencaInfecciosa.
- Pneumonia e Insuficiência Respiratória pertencem à classe DoencaRespiratoria.
- Hipertensão pertence à classe DoencaCronica.
- Gastrite pertence à classe DoencaGastrointestinal.
- Isolamento, Dieta Baixa em Acidez, Suplemento de Oxigênio pertencem à classe TratamentoNaoMedicamentoso.
- Antiviral, Antibiótico, Medicamento Anti-hipertensivo pertencem à classe TratamentoMedicamentoso.
- João e Maria pertencem à classe Paciente.

```
### **5. Instâncias**  
  
# Sintomas  
febre = SintomaRespiratorio("Febre")  
tosse = SintomaRespiratorio("Tosse")  
dor_no_peito = SintomaCardiaco("Dor no peito")  
nauseas = SintomaGastrointestinal("Náuseas")  
diarreia = SintomaGastrointestinal("Diarreia")  
falta_ar = SintomaRespiratorio("Falta de ar")  
  
# Exames  
raio_x = ExameImagem("Raio-X")  
tomografia = ExameImagem("Tomografia Computadorizada")  
hemograma = ExameLaboratorial("Hemograma Completo")  
gasometria = ExameLaboratorial("Gasometria")  
teste_esforco = ExameFuncional("Teste de Esforço")  
endoscopia = ExameImagem("Endoscopia Digestiva")  
  
# Doenças  
covid19 = DoencaInfecciosa("COVID-19")  
pneumonia = DoencaRespiratoria("Pneumonia")  
hipertensao = DoencaCronica("Hipertensão")  
gastrite = DoencaGastrointestinal("Gastrite")  
insuficiencia_respiratoria = DoencaRespiratoria("Insuficiência Respiratória")  
  
# Tratamentos  
isolamento = TratamentoNaoMedicamentoso("Isolamento")  
antiviral = TratamentoMedicamentoso("Antiviral")  
dieta_baixa_acidez = TratamentoNaoMedicamentoso("Dieta Baixa em Acidez")  
antibiotico = TratamentoMedicamentoso("Antibiótico")  
suplemento_oxigenio = TratamentoNaoMedicamentoso("Suplemento de Oxigênio")  
anti_hipertensivo = TratamentoMedicamentoso("Medicamento Anti-hipertensivo")  
  
# Pacientes  
paciente1 = Paciente("Joao")  
paciente2 = Paciente("Maria")
```

8 Exemplos Práticos (Código)

- Relacionamento das instâncias
- Febre requer hemograma, tomografia.
- Tosse requer raio-X.
- Dor no peito requer teste de esforço.
- Náuseas requer endoscopia.
- Diarreia requer endoscopia, hemograma.
- Falta de ar requer gasometria, tomografia.
- COVID-19 confirmada por raio-X, hemograma.
- Pneumonia confirmada por raio-X, tomografia.
- Hipertensão confirmada por teste de esforço, hemograma.
- Gastrite confirmada por endoscopia, hemograma.
- Insuficiência Respiratória confirmada por gasometria, tomografia.

```
### **6. Relacionar Instâncias**  
febre.requer_exame = [hemograma, tomografia]  
tosse.requer_exame = [raio_x]  
dor_no_peito.requer_exame = [teste_esforco]  
nauseas.requer_exame = [endoscopia]  
diarreia.requer_exame = [endoscopia, hemograma]  
falta_ar.requer_exame = [gasometria, tomografia]  
  
covid19.confirmada_por = [raio_x, hemograma]  
pneumonia.confirmada_por = [raio_x, tomografia]  
hipertensao.confirmada_por = [teste_esforco, hemograma]  
gastrite.confirmada_por = [endoscopia, hemograma]  
insuficiencia_respiratoria.confirmada_por = [gasometria, tomografia]  
  
covid19.tratada_com = [isolamento, antiviral]  
pneumonia.tratada_com = [antibiotico, suplemento_oxygenio]  
hipertensao.tratada_com = [anti_hipertensivo]  
gastrite.tratada_com = [dieta_baixa_acidez]  
insuficiencia_respiratoria.tratada_com = [suplemento_oxygenio]  
  
paciente1.apresenta = [febre, tosse, falta_ar]  
paciente2.apresenta = [nauseas, diarreia, dor_no_peito]  
  
paciente1.diagnosticado_com = [covid19]  
paciente2.diagnosticado_com = [gastrite, hipertensao]
```


8 Exemplos Práticos (Código)

- COVID-19 tratado com isolamento, antiviral.
- Pneumonia tratado com antibiótico, suplemento de oxigênio.
- Hipertensão tratado com medicamento anti-hipertensivo.
- Gastrite tratado com dieta baixa em acidez.
- Insuficiência Respiratória tratado com suplemento de oxigênio.
- João apresenta febre, tosse, falta de ar.
- Maria apresenta náuseas, diarreia, dor no peito.
- João diagnosticado com COVID-19.
- Maria diagnosticada com gastrite, hipertensão.

```
### **6. Relacionar Instâncias**  
febre.requer_exame = [hemograma, tomografia]  
tosse.requer_exame = [raio_x]  
dor_no_peito.requer_exame = [teste_esforco]  
nauseas.requer_exame = [endoscopia]  
diarreia.requer_exame = [endoscopia, hemograma]  
falta_ar.requer_exame = [gasometria, tomografia]  
  
covid19.confirmada_por = [raio_x, hemograma]  
pneumonia.confirmada_por = [raio_x, tomografia]  
hipertensao.confirmada_por = [teste_esforco, hemograma]  
gastrite.confirmada_por = [endoscopia, hemograma]  
insuficiencia_respiratoria.confirmada_por = [gasometria, tomografia]  
  
covid19.tratada_com = [isolamento, antiviral]  
pneumonia.tratada_com = [antibiotico, suplemento_oxigenio]  
hipertensao.tratada_com = [anti_hipertensivo]  
gastrite.tratada_com = [dieta_baixa_acidez]  
insuficiencia_respiratoria.tratada_com = [suplemento_oxigenio]  
  
paciente1.apresenta = [febre, tosse, falta_ar]  
paciente2.apresenta = [nauseas, diarreia, dor_no_peito]  
  
paciente1.diagnosticado_com = [covid19]  
paciente2.diagnosticado_com = [gastrite, hipertensao]
```

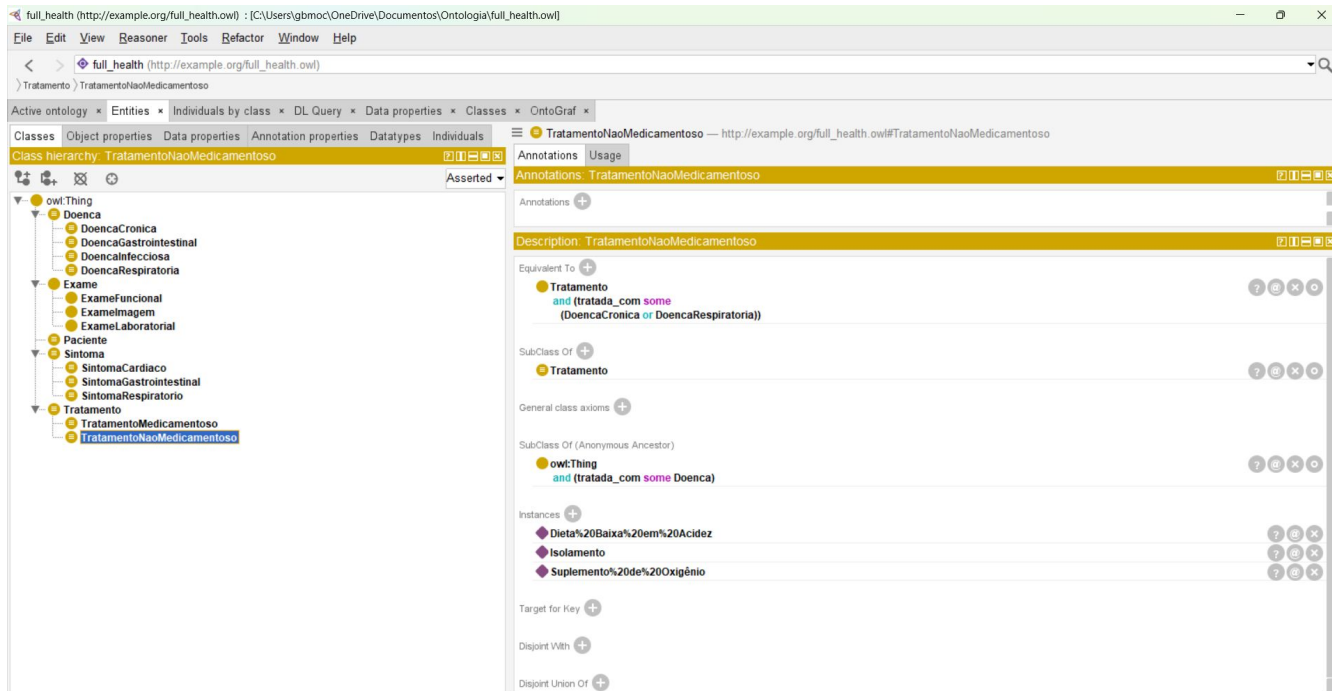
8 Exemplos Práticos (Código)

- **Função:** Salvar a ontologia criada no formato OWL.
- **Arquivo gerado:** `full_health.owl`.
- **Mensagem exibida:** Confirmação de que a ontologia foi salva corretamente

```
### **7. Salvar a Ontologia**  
onto.save("full_health.owl")  
print("Ontologia criada e salva no arquivo 'full_health_with_instances.owl'")
```

9 Exemplos Práticos (Protegé)

Tela de configuração de classes:



9 Exemplos Práticos (Protegé)



Annotations: Adicione descrições ou comentários para documentar a classe.

SubClass Of: Organize a classe dentro de uma hierarquia maior.

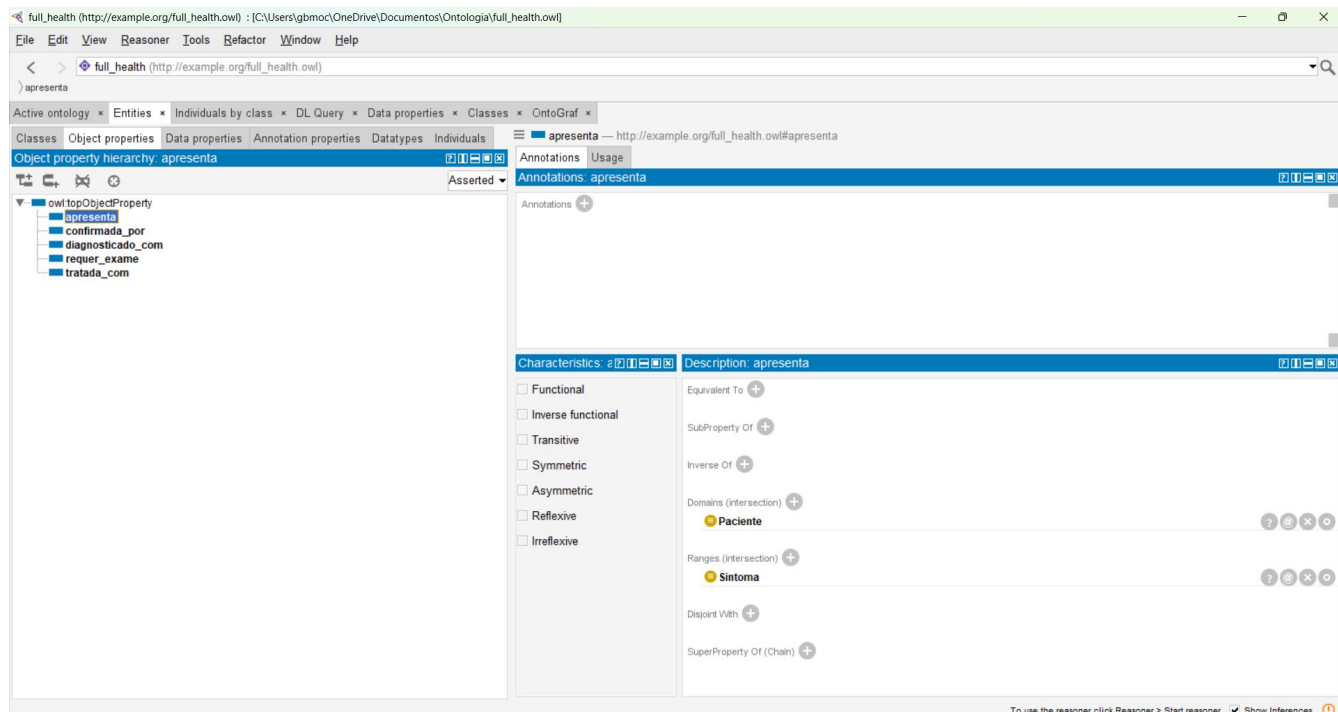
Equivalent To: Defina condições lógicas para determinar a pertença à classe.

Disjoint With: Declare incompatibilidades entre classes.

Interatividade: Adicione, edite ou exclua anotações e configurações para personalizar a classe.

9 Exemplos Práticos (Protegé)

Tela de configuração de propriedades:

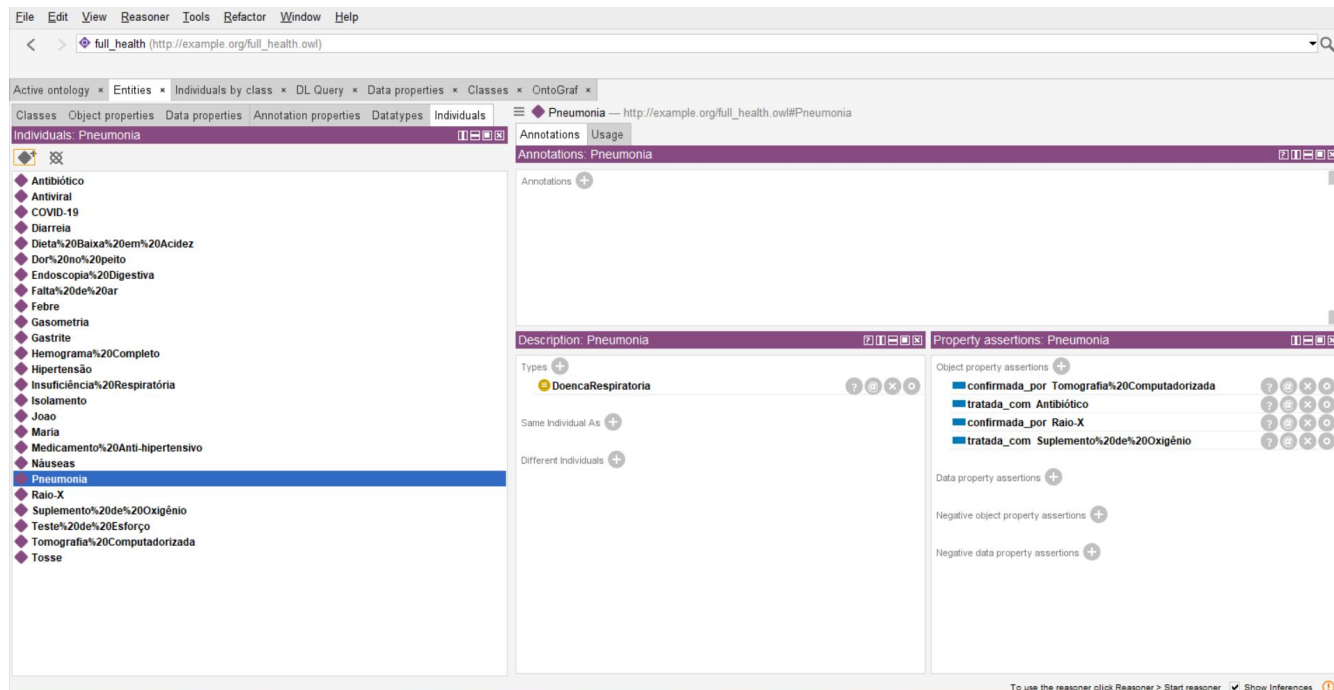


9 Exemplos Práticos (Protegé)

- **Annotations:** Adicione descrições ou comentários sobre a propriedade.
- **Characteristics:** Configure comportamentos como funcionalidade, simetria ou transitividade.
- **Domain e Range:** Defina a origem e o destino da relação da propriedade.
- **Inverse Of/SubProperty Of:** Relacione com propriedades inversas ou superiores.
- **Interatividade:** Adicionar, editar ou excluir definições para personalizar a propriedade.

9 Exemplos Práticos (Protegé)

Tela de configuração de indivíduos:



9 Exemplos Práticos (Protegé)



- **Annotations:** Adicione descrições ou comentários ao indivíduo.
- **Types:** Identifique as classes às quais o indivíduo pertence.
- **Same/Different Individuals:** Declare igualdade ou diferença com outros indivíduos.
- **Property Assertions:** Configure relações (objetos ou dados) e restrições negativas para o indivíduo.
- **Opções:** Adicionar, editar ou excluir atributos, relações e metadados do indivíduo.

Referências

Ontologias: conceitos, usos, tipos, metodologias, ferramentas e linguagens. 2007. Disponível em: https://ww2.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_001-07.pdf

Engenharia Ontológica aplicada para o projeto e especificação da ontocancro. 2019. Disponível em: https://repositorio.ufsm.br/bitstream/handle/1/19434/DIS_PPGCC_2019_BONINI_JESSICA.pdf?sequence=1&isAllowed=y

MODELO DE REPRESENTAÇÃO EM ONTOLOGIA NO GERENCIAMENTO DE PROJETOS DE EDIFICAÇÕES NA CONSTRUÇÃO CIVIL PELOS SISTEMAS ONTOKEM E PROTÉGÉ. 2021. Disponível em: <https://proceeding.ciki.ufsc.br/index.php/ciki/article/view/1080/611>

UMA ONTOLOGIA PARA APOIAR O ENSINO DE MATEMÁTICA BÁSICA COM USO DE ROBÓTICA EDUCACIONAL. 2023. Disponível em: <https://www2.ifrn.edu.br/ojs/index.php/HOLOS/article/view/10328/3975>

OBAMA, an Ontology-Based Software Tool for Agile Method Adoption. Disponível em: <https://arxiv.org/pdf/2206.02207>