

UNIVERSIDADE FEDERAL DE CAMPINA GRANDE
CENTRO DE ENGENHARIA ELÉTRICA E INFORMÁTICA
CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Descrição e armazenamento de cenários de
treinamento de operadores a partir de uma
ontologia de domínio

Gustavo Barreto Rangel

Campina Grande - PB
Abril de 2015

GUSTAVO BARRETO RANGEL

DESCRIÇÃO E ARMAZENAMENTO DE CENÁRIOS DE TREINAMENTO DE
OPERADORES A PARTIR DE UMA ONTOLOGIA DE DOMÍNIO

Trabalho de conclusão de curso submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Orientadora: Professora Maria de Fátima Queiroz Vieira, Ph.D

Campina Grande - PB

Abril de 2015

GUSTAVO BARRETO RANGEL

DESCRIÇÃO E ARMAZENAMENTO DE CENÁRIOS DE TREINAMENTO DE
OPERADORES A PARTIR DE UMA ONTOLOGIA DE DOMÍNIO

Trabalho de conclusão de curso submetido à Unidade Acadêmica de Engenharia Elétrica da Universidade Federal de Campina Grande como parte dos requisitos necessários para a obtenção do grau de Bacharel em Ciências no Domínio da Engenharia Elétrica.

Aprovada em Abril de 2015

BANCA EXAMINADORA

Professora Maria de Fátima Queiroz Vieira, Ph.D - Orientadora
UFCG

Professor Avaliador
UFCG

Campina Grande - PB
Abril de 2015

Dedicado à minha família e aos meus
amigos.

pessoa também.

são interpretados e os espaços são respeitados)

"It's not the load that breaks you down. It's the way you carry it"

Lena Horne

Resumo

Este trabalho foi realizado como parte de um projeto P&D contratado pela empresa CHESF - Companhia Hidroelétrica do São Francisco. O projeto, cujo título é Gerador de Cenários para o Treinamento de Operadores no Ambiente SAGE/SIMULOP, consiste no desenvolvimento de um Gerador de Cenários para o SIMULOP, ambiente de simulação de eventos do sistema supervisorio utilizado pela CHESF, o SAGE - Sistema Aberto para Gerenciamento de Energia, desenvolvido pela CEPEL - Centro de Pesquisas de Eneergia Elétrica.

Este trabalho consiste no desenvolvimento de uma API (*application programming interface*) para descrição, armazenamento e recuperação de cenários de treinamento localmente utilizando uma ontologia de domínio. Por meio da ferramenta em desenvolvimento, é possível armazenar cenários em um arquivo local e no banco de dados do ambiente. A recuperação de cenários locais para edição ou consulta também faz parte deste trabalho. Outro fator presente é o desenvolvimento da Base de Casos, isto é, um local dentro do banco de dados da ferramenta no qual cenários estarão disponíveis para os usuários através do servidor da empresa.

Palavras-chave: treinamento de operadores, simulador, ontologia, cenário de treinamento.

Abstract

This paper was done as part of a project R&D contracted by company CHESF - Companhia Hidroelétrica do São Francisco. The project consists of the development of a Scenario Generator for the SIMULOP, an event simulator environment of the supervisory system used by CHESF, named SAGE - Sistema Aberto para Gerenciamento de Energia, developed by CEPTEL - Centro de Pesquisas de Energia Elétrica.

This paper consists of the development of an API (*application programming interface*) for the description, storage and retrieval of local training scenario using a domain ontology. Through the developing software, it is possible to store scenarios in a local file and in the database of the environment. The local scenario retrieval for edition or consultation is also a part to this work. Another important factor is the development of a Case Base, which is a place in the database of the Generator where scenarios will be available for users through the company server.

Keywords: operator training, simulator, ontology, training scenario.

Sumário

1	Introdução	1
1.1	Motivação	2
1.2	Objetivos	3
1.3	Estrutura do documento	3
2	Revisão Bibliográfica	4
2.1	Ontologia	5
2.1.1	Linguagens	7
2.1.2	OWL	7
2.1.3	Jena	8
2.1.4	Ferramentas	9
2.1.5	Metodologias	9
2.2	Sistema Gerenciador de Banco de Dados (SGBD)	11
3	Projeto da API do Gerador de Cenários	13
3.1	Análise de Requisitos	14
3.2	Arquitetura	14
3.3	Ontologia de Cenários de Treinamento	16
3.4	Modelo Entidade Relacionamento	18
3.5	API para Mapeamento de Dados	23
4	Validação - Instanciação de Cenários	25
5	Conclusões	32
	Referências Bibliográficas	32

A Dicionário de Dados do BD

Lista de Tabelas

2.1	Linguagens para concepção de ontologias	7
A.1	Entidade acao	35
A.2	Entidade acao	35
A.3	Entidade atores_envolvidos	35
A.4	Entidade atores_envolvidos	36
A.5	Entidade cenario	36
A.6	Entidade cenario	36
A.7	Entidade comparacao	37
A.8	Entidade comparacao	37
A.9	Entidade constante	37
A.10	Entidade constante	37
A.11	Entidade dificuldade	37
A.12	Entidade dificuldade	37
A.13	Entidade disparo_condicional	37
A.14	Entidade disparo_condicional	38
A.15	Entidade disparo_temporal	38
A.16	Entidade disparo_temporal	38
A.17	Entidade entrada	38
A.18	Entidade entrada	38
A.19	Entidade equipamento	39
A.20	Entidade equipamento	39
A.21	Entidade evento	39
A.22	Entidade evento	39
A.23	Entidade evento	40
A.24	Entidade evento	40

A.25 Entidade frequencia	40
A.26 Entidade frequencia	40
A.27 Entidade grupo_eventos	40
A.28 Entidade grupo_eventos	40
A.29 Entidade instalação	41
A.30 Entidade instalação	41
A.31 Entidade item_roteiro_execucao_cenario	41
A.32 Entidade item_roteiro_execucao_cenario	41
A.33 Entidade item_roteiro_preparacao_cenario	41
A.34 Entidade item_roteiro_preparacao_cenario	42
A.35 Entidade nivel_tensao	42
A.36 Entidade nivel_tensao	42
A.37 Entidade operacao_aritmetica	43
A.38 Entidade operacao_aritmetica	43
A.39 Entidade operacao_binaria	43
A.40 Entidade operacao_binaria	43
A.41 Entidade operacao_logica	43
A.42 Entidade operacao_logica	44
A.43 Entidade prioridade	44
A.44 Entidade prioridade	44
A.45 Entidade sinalizacao	44
A.46 Entidade sinalizacao	44
A.47 Entidade sinalizacao_cenario	44
A.48 Entidade sinalizacao_cenario	45
A.49 Entidade tarefa	45
A.50 Entidade tarefa	45
A.51 Entidade tipo_acao	45
A.52 Entidade tipo_acao	45
A.53 Entidade tipo_comparacao	46
A.54 Entidade tipo_comparacao	46
A.55 Entidade tipo_da_tarefa	47
A.56 Entidade tipo_da_tarefa	47

A.57 Entidade tipo_disparo	47
A.58 Entidade tipo_disparo	47
A.59 Entidade tipo_disparo_temporal	47
A.60 Entidade tipo_disparo_temporal	47
A.61 Entidade tipo Equipamento	47
A.62 Entidade tipo Equipamento	48
A.63 Entidade tipo Medida	48
A.64 Entidade tipo Medida	48
A.65 Entidade tipo_Medida_tipo_Equipamento	48
A.66 Entidade tipo_Medida_tipo_Equipamento	48
A.67 Entidade tipo_oper_aritmetica	48
A.68 Entidade tipo_oper_aritmetica	49
A.69 Entidade tipo_oper_logica	49
A.70 Entidade tipo_oper_logica	49
A.71 Entidade tutor	49
A.72 Entidade tutor	49
A.73 Entidade variavel_Medida	50
A.74 Entidade variavel_Medida	50

Lista de Figuras

2.1	Sub-linguagens OWL	8
2.2	Interface gráfica do <i>Protégé</i>	10
3.1	Representação da Arquitetura do Gerador de Cenários	15
3.2	Diagrama UML de sequência para salvar e disponibilizar um cenário na base de casos	16
3.3	Hierarquia de Classes da Ontologia de Domínio	17
3.4	Lista de Métodos e Atributos da Classe Fachada da Ontologia	18
3.5	Parte 1 do Modelo do BD	20
3.6	Parte 2 do Modelo do BD	20
3.7	Parte 3 do Modelo do BD	21
3.8	Lista de Métodos e Atributos da Fachada do BD	22
3.9	Casos de Uso: Atualização do BD/Ontologia/Base de Casos	24
4.1	Descritivo do cenário de treinamento parte 1	26
4.2	Descritivo do cenário de treinamento parte 2	27
4.3	Diagrama Unifilar de CGD Retirado do SAGE	28
4.4	Parte do descritivo mostrado na ferramenta Gerador de Cenários	29
4.5	Medida criada para a geração do cenário	30
4.6	Condicional criada para a geração do cenário	30
4.7	Tabela de eventos criada para a geração do cenário	31
4.8	Arquivo que é carregado no SAGE	31

Capítulo 1

Introdução

apresentacao.

Com a crescente complexidade do sistema elétrico brasileiro, torna-se cada vez mais difícil capacitar funcionários aptos a enfrentar os diversos problemas que podem surgir no setor. A pressão por uma resposta rápida e correta após um fenômeno incomum exige um alto grau de conhecimento e experiência do operador. Com isso, o treinamento de operadores de sistemas elétricos é essencial em qualquer companhia de energia elétrica (1).

O treinamento de operadores para o sistema elétrico é feito com aulas teóricas e a prática é obtida observando operadores experientes e operando o sistema real. Este processo é incompleto visto que um operador pode ter anos de trabalho sem nunca antes ter se deparado com uma situação específica de emergência (2).

O Sistema Aberto para Gerenciamento de Energia – SAGE – foi desenvolvido pelo Centro de Pesquisa de Energia Elétrica – CEPEL – para atuar na área de supervisão e controle de sistemas elétricos e possui propriedades como portabilidade, interconectividade, expansibilidade e modularidade. Atualmente o SAGE é usado como sistema supervisorio por companhias de geração, transmissão e distribuição de energia elétrica nacionais. Por ser uma solução que abrange todos os níveis de supervisão e controle do sistema elétrico possibilita a redução de custos relacionados à implantação, operação e manutenção do serviço (3).

Diante da necessidade de treinamento de operadores no uso do sistema SAGE, desde 2001 a CHESF, em parceria com a CEPEL, desenvolveu o Projeto SIMULOP, um “Simulador para Treinamento de Operadores”. Este é um ambiente de simulação

de sistemas elétricos utilizado no treinamento de novos operadores e na reciclagem de operadores experientes.

O SIMULOP é o resultado da integração da ferramenta EMS (Energy Management System) de supervisão e controle de sistemas de potência - SAGE, com um simulador digital em tempo-real de sistemas elétricos de potência - OTS (Operator Training Simulator) produzido pela EPRI (Electric Power Research Institute) (4).

1.1 Motivação

A geração de cenários de treinamento para o SIMULOP é atualmente uma tarefa repetitiva e complexa, pois a ferramenta de geração de cenários de treinamento oferecida neste simulador apresenta problemas de usabilidade e oferece poucos recursos para edição. O esforço de criação de cenários de treinamento aumenta proporcionalmente ao número de eventos que precisam ser descritos para compor um cenário. Por outro lado, os tutores, responsáveis pela criação dos cenários de treinamento, dispõem de pouco tempo para realizarem esta tarefa.

A partir desta dificuldade, está em andamento no LIHM/DEE um projeto de P&D para construção de um ambiente de geração de cenários para o SIMULOP, para a empresa CHESF – Companhia Hidroelétrica do São Francisco.

Neste projeto, está prevista a construção de uma base de casos com cenários de treinamento, os quais serão armazenados em um banco de dados conectado à intranet da empresa. Neste repositório os instrutores responsáveis pela elaboração de treinamentos de operadores poderão salvar e terão acesso aos cenários de treinamento elaborados para a empresa. Tal banco deverá facilitar o acesso, o compartilhamento e a gestão de um acervo de cenários que reproduzem tanto situações onde o sistema elétrico opera em condições normais, quanto em situação de contingência. Os cenários serão elaborados para operadores de instalações (subestações) e de centros de operação (sistema), da empresa CHESF.

Por outro lado, os tutores precisam ser independentes de qualquer conexão com um servidor para a construção de cenários. Dessa forma, se faz necessário armazenar localmente os cenários em elaboração no ambiente do Gerador de Cenários. Além disso, todas as informações necessárias para a construção de um cenário também devem estar

presentes em um arquivo local.

Uma vez concluída a edição destes cenários, eles são armazenados na Base de Casos e disponibilizados para consulta por todos os instrutores da empresa. O armazenamento local se dá em uma base de conhecimento, estruturada na forma de uma ontologia que representa o domínio de cenários de treinamento realizados na empresa CHESF, no ambiente do SIMULOP (5).

1.2 Objetivos

Desenvolvimento de uma API (*Application Programming Interface*) para descrever cenários de treinamento, com base em uma ontologia de domínio, e armazená-los em uma Base de conhecimento e em um Banco de Dados.

Esta API também deverá oferecer recursos para recuperar os dados de cenários já armazenados na base de conhecimento ou no Banco de Dados para disponibilizá-los para edição.

1.3 Estrutura do documento

Além deste capítulo introdutório, o presente trabalho é constituído por mais 4 capítulos:

Capítulo 2 - Revisão Bibliográfica: apresenta os fundamentos deste trabalho, relacionados à ontologia, como seu conceito e a metodologia utilizada. Há também uma explicação sobre a linguagem da ontologia, as ferramentas e *framework* escolhidos.

Capítulo 3 - Projeto da API do Gerador de Cenários: trata do projeto do Gerador de Cenários e como a ontologia apoia sua construção. A arquitetura do *software* e como são tratados o mapeamento de dados do SAGE para a ontologia e para o BD.

Capítulo 4 - Validação: neste capítulo, é apresentado o método adotado para a validação da API, o qual consistiu na construção de uma fachada para testes na ferramenta *EasyAccept* (6) e na instaciação de um cenário de treinamento, o qual foi salvo e recuperado na base de conhecimento e no banco de dados.

Capítulo 4 - Conclusões: apresenta as conclusões sobre o presente trabalho e discute se os objetivos alcançados estão de acordo com os objetivos propostos.

Capítulo 2

Revisão Bibliográfica

apresentacao.

O projeto do Gerador de Cenários foi desenvolvido para o SIMULOP com a principal finalidade de facilitar a criação de cenários de treinamento de operadores.

O Simulop replica o ambiente real de uma sala de comando, interligando o Simulador Digital de Redes Elétricas: Eprl OTS, com o Sistema SAGE de Gerenciamento de Energia, possibilitando o treinamento dos operadores do sistema elétrico, com realismo e eliminando o risco aos operadores e ao sistema elétrico (7).

O OTS (Operator Training Simulator) é um *software* capaz de simular o comportamento de um sistema elétrico de potência, com resposta em tempo-real às condições operativas e eventos sistêmicos. Ou seja, na ocorrência de um evento, o OTS realiza cálculos de modo a simular a resposta do sistema real (8).

O SAGE é um *software* supervisor desenvolvido pelo CEPEL (Centro de Pesquisas de Energia Elétrica). Este supervisor realiza funções de gerenciamento de energia nos sistemas elétricos de potência, sendo configurável para diferentes plataformas instaladas em subestações e Centros de Operação de Sistemas (9).

Um cenário de treinamento consiste da descrição do estado inicial do ambiente simulado, da especificação da evolução das ações do treinamento (descrição da sequência de eventos que devem ocorrer no durante o treinamento) e, da descrição dos recursos que serão utilizados na evolução do cenário (10).

2.1 Ontologia

Esta revisão se fundamenta no texto da tese de doutorado de Flávio Torres Filho, apresentada ao programa de pós-graduação em engenharia elétrica da UFCG - Universidade Federal de Campina Grande (11).

Historicamente, o termo “Ontologia” tem origem no grego *ontos*, que significa "ser", e *logos*, que significa "palavra". Utilizado inicialmente no campo da Filosofia, o termo ontologia diz respeito à área do conhecimento que trata da natureza e da organização do ser. (11)

Na Ciência da Informação, dentre as definições de ontologia mais citadas na literatura, destaca-se que uma ontologia pode ser definida como “uma especificação explícita e formal de uma conceitualização compartilhada” (12), apud (11).

O termo conceitualização refere-se a um conjunto de definições de conceitos e de relacionamentos existentes entre estes conceitos. Uma conceitualização é uma visão abstrata do domínio que se deseja representar. Uma “especificação explícita” quer dizer que as limitações de uso de conceitos e relacionamentos estão explicitamente definidas. Estando também formalizada, a ontologia é passível de ser processada por máquinas (13), apud (11).

Sendo assim, uma ontologia é composta de um conjunto de conceitos organizados como uma taxonomia e usados para descrever uma determinada realidade.

Os componentes básicos de uma ontologia são classes (dispostas de forma hierárquica), relações (que representam as formas de interação entre conceitos do domínio ontológico), axiomas (regras que regulam as associações entre as classes e relações) e instâncias (elementos específicos do domínio).

Para a engenharia de *software*, uma ontologia representa um elemento importante no processo de desenvolvimento de aplicações (14). Por exemplo, ao disponibilizar um vocabulário especializado para descrição de um dado domínio, a ontologia permite reduzir erros conceituais e de terminologia durante a modelagem e o desenvolvimento do *software*. Ao mesmo tempo, possibilita tanto o compartilhamento do conhecimento entre os desenvolvedores do *software* quanto à transferência e a aquisição desse mesmo conhecimento, apoiados pela análise ontológica.

Além disso, uma ontologia, enquanto base conceitual, pode apoiar a definição, a classificação e a identificação de componentes de *software* voltados para uma aplicação

dentro do domínio ontológico.

Dessa forma, um catálogo de componentes de *software* pode ser constituído com base na ontologia do domínio. E, uma vez disponível, novos produtos podem ser mais rapidamente desenvolvidos e, ainda, agregar valores como padronização, confiança, baixo custo de desenvolvimento e fácil manutenção, obtidos a partir do reuso de componentes de *software* já utilizados, testados e validados.

Por essa razão, uma abordagem de desenvolvimento com base em ontologias é especialmente adequada para projetos de *software* que pertençam ao mesmo domínio ou domínios semelhantes, o que permite estender a noção de reuso para a fase de modelagem, e não apenas na implementação como de costume. Assim, ontologias permitem identificar modelos de componentes reutilizáveis no sistema.

Uma ontologia pode ser usada para o projeto e desenvolvimento de software com os seguintes objetivos (14):

Especificação

ontologias são usadas para especificar tanto os requisitos e definição de componentes como as funcionalidades do sistema;

Validação

ontologias são usadas para avaliar o projeto do sistema e verificar sua consistência através de processos semiautomáticos;

Reuso

ontologias podem ser organizadas em módulos para definir domínios, subdomínios e suas tarefas relacionadas, que podem ser depois reutilizadas ou adaptadas para outros problemas;

Busca

ontologias são usadas como repositório de informações;

Manutenção

ontologias melhoram o uso e o armazenamento de informações para a manutenção do sistema;

Aquisição do conhecimento

Linguagem de ontologias tradicionais
Cycl, Ontolígua, F-Logic, CML, OCML, Loom, KIF.
Linguagens padrão Web
XML, RDF.
Linguagens de ontologias Web-based
OIL, DAML+OIL, SHOE, XOL, OWL.

Tabela 2.1: Linguagens para concepção de ontologias

ontologias podem ser usadas como um guia para o processo que aquisição do conhecimento.

Supondo que diferentes projetos de *software* sejam desenvolvidos com base em ontologias próprias, a identificação de modelos de componentes originalmente concebidos para um deles, mas passíveis de serem reutilizados no outro, poderá ser facilitada a partir do estudo das ontologias adotadas e do mapeamento entre termos com conceitos correspondentes.

2.1.1 Linguagens

Existem diversas linguagens que apoiam a especificação de ontologias. Segundo , essas linguagens podem ser categorizadas como: linguagens de ontologias tradicionais, linguagens padrão Web e linguagens de ontologias Web-based. Exemplos são apresentados na tabela 2.1.1.

2.1.2 OWL

A linguagem OWL (Web Ontology Language), desenvolvida pela W3C (World Wide Web Consortium), é uma linguagem recente, mas que vem ganhando grande aceitação. Ela deriva de duas propostas diferentes, a Européia OIL (Linguagem de Intercâmbio de Ontologias) e DAML (Linguagem de Anotação de Agentes do Departamento de Defesa dos Estados Unidos). Para atender a diferentes comunidades de usuários, a OWL possui três diferentes sub-linguagens, que mudam quanto ao nível de expressividade: OWL-Lite; OWL-DL; OWL-Full (15).

A figura 2.1 ilustra os três tipos de linguagem owl.

Essas sub-linguagens são caracterizadas da seguinte forma (16):

OWL-Lite

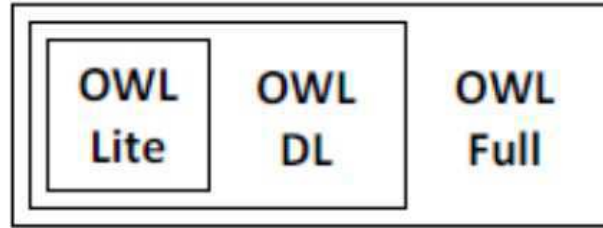


Figura 2.1: Sub-linguagens OWL

é a sub-linguagem sintaticamente mais simples. Destina-se a situações em que apenas são necessárias restrições e uma hierarquia de classe simples. Por exemplo, o OWL-Lite pode fornecer uma forma de migração para tesouros existentes, bem como de outras hierarquias simples;

OWL-DL

é mais expressiva que a OWL-Lite e baseia-se em lógica descritiva, um fragmento de Lógica de Primeira Ordem, portanto passível de raciocínio automático. É possível assim computar automaticamente a hierarquia de classes e verificar inconsistências na ontologia;

OWL-Full

é a sub-linguagem OWL mais expressiva. Destina-se a situações em que a alta expressividade é mais importante. Não é possível efetuar inferências em ontologias OWL-Full.

A W3C disponibiliza alguns guias para usuários interessados na linguagem, onde se pode aprender a sintaxe da linguagem e consultar vários exemplos de aplicação. Por outro lado, a maioria dos pesquisadores interessados no desenvolvimento de ontologias costuma utilizar ferramentas como o *Protégé*, que dispensam o contato direto do usuário com o código em OWL durante a modelagem, mas que exportam a ontologia para o formato .owl quando necessário.

2.1.3 Jena

O *Jena* (17) é um projeto que se originou dentro do núcleo de pesquisa em *Web Semântica* da HP. Seu objetivo é proporcionar um *framework* na linguagem *Java* que dê suporte à utilização da *Web Semântica* por qualquer aplicativo capaz de utilizá-lo. Este suporte inclui recursos para manipulação de RDF, RDFS, OWL e DAML+OIL.

O *Jena* trabalha de forma transparente em relação à linguagem adotada para representar a ontologia. De acordo com a linguagem, determinados recursos são habilitados ou não, porém, a forma de manipular e criar ontologias é a mesma para qualquer linguagem de representação (18).

No projeto do Gerador de Cenários, a ontologia é usada como repositório local de um cenário de treinamento, e o *Jena* foi utilizado para o desenvolvimento de código em *Java* a partir da ontologia representada na linguagem OWL-DL.

2.1.4 Ferramentas

Existem ferramentas de *software* desenvolvidas com o objetivo de apoiar o processo de desenvolvimento de ontologias, bem como o uso e manutenção subsequente. Neste trabalho, em particular, o processo de desenvolvimento foi apoiado pela ferramenta Protege-OWL, abordada a seguir.

Protégé

O Protege é um ambiente para desenvolvimento de ontologias, criado pelo departamento de informática da Universidade de Standford. Originalmente pretendia-se que o *software* servisse como ferramenta para aquisição do conhecimento para um sistema especialista, voltado à área medica. Entretanto, a ferramenta foi gradativamente modernizada e hoje permite a concepção de ontologias de domínio, a personalização de formulários de entrada de dados, e a inserção e edição de dados. Sua interface pode ser vista na figura 2.2.

O guia do usuário, disponível em <<http://protege.stanford.edu>>, descreve todas as funcionalidades da ferramenta, usando como exemplo o passo-a-passo do desenvolvimento de uma ontologia de Pizzas. O Protégé foi desenvolvido na linguagem *Java* e possui o código aberto. Dessa forma, permite-se que usuários possam desenvolver componentes integráveis ao *software* (plugins) utilizando as APIs do Protégé. Essa característica também facilitou o crescimento e a aceitação da ferramenta.

2.1.5 Metodologias

O desenvolvimento e a manipulação de ontologias têm sido sistematizados por metodologias. No entanto, nenhuma metodologia é amplamente utilizada, ou seja, con-

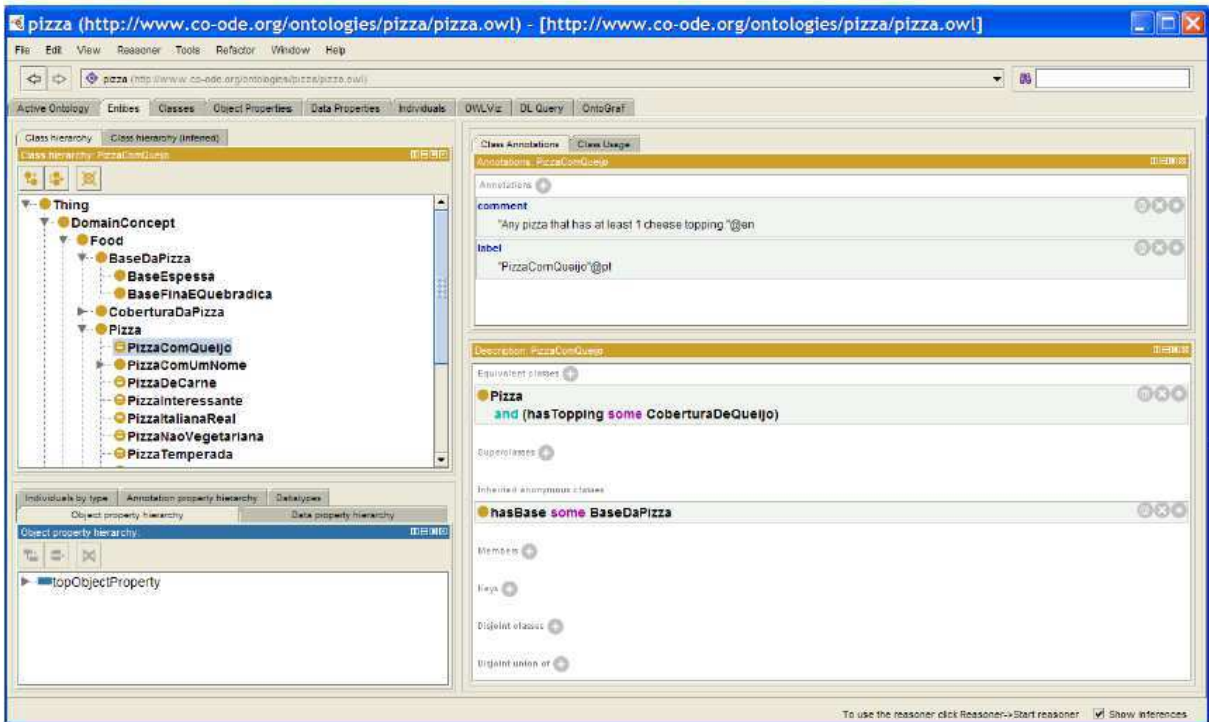


Figura 2.2: Interface gráfica do *Protégé*

sensual entre os desenvolvedores, que em sua maioria apoiam-se na própria experiência e utilizam-se de uma abordagem particular. Dentre as metodologias propostas na literatura estão On-To-Knowledge, método 101 e a Methontology. Para este projeto em particular, foi escolhido o método 101 (19).

A metodologia 101 é constituída por sete etapas iterativas que auxiliam os desenvolvedores na concepção de ontologias de domínio (19), como pode ser visto a seguir.

Etapa 1

Determinar o domínio e o escopo da ontologia;

Etapa 2

Considerar o reuso de ontologias existentes;

Etapa 3

Listar termos importantes para a ontologia;

Etapa 4

Definir as classes e a hierarquia das classes;

Etapa 5

Definir as propriedades de classes - fresta (slot);

Etapa 6

Definir as facetas das frestas (slots);

Etapa 7

Criar as instâncias.

2.2 Sistema Gerenciador de Banco de Dados (SGBD)

Um banco de dados é uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa. Por persistente, queremos sugerir intuitivamente que os dados desse banco de dados diferem em espécie de outros dados mais efêmeros, como dados de entrada, dados de saída, filas de trabalho, blocos de controle de *software*, instruções SQL, resultados intermediários e, de modo geral, quaisquer dados que tenham natureza transitória (20).

Gerenciadores de banco de dados são sistemas projetados para gerir um grande volume de informações e, dentre seus vários objetivos, destacam-se a garantia da consistência, da segurança e o fácil acesso aos dados.

Há vários gerenciadores de bancos de dados grátis e de código aberto, como o SQL Server desenvolvido pela Microsoft, o SQLite e o MySQL desenvolvido pela Oracle. Para este trabalho, foi escolhida a utilização do MySQL (21), pois este é o banco de dados usado pela CHESF, empresa financiadora do projeto.

O projeto do banco de dados foi pensado a partir do levantamento das características do projeto. Com a análise de requisitos, as regras de negócios foram inicialmente identificadas e, então, construiu-se o modelo Entidade-Relacionamento do Banco.

Para realizarmos o mapeamento objeto-relacional, foi utilizado mais um *framework*, o *Hibernate* (22). *Hibernate* é um *software* livre, de código aberto e distribuído com a licença LGPL.

Apesar da característica principal do *Hibernate* ser mapear classes em *Java* para tabelas do banco de dados, ele também faz a engenharia reversa, isto é, é possível transformar entidades do banco em classes *Java*. Isso é feito através de *Anotações Java* ou de arquivos XML. No caso do Gerador de Cenários, foram utilizados arquivos XML para gerar os códigos fontes necessários para a persistência das classes.

O desenvolvimento do *software* e do banco de dados ocorrem em paralelo. En-

tretanto, com o uso do *Hibernate*, isso não é problema, pois esse *framework* facilita a atualização da geração de classes em *Java* quando o modelo de dados é atualizado.

O próximo capítulo apresentará mais detalhes do Gerador de Cenários bem como a API proposta para salvar e recuperar cenários da base de conhecimento e da base de casos.

Capítulo 3

Projeto da API do Gerador de Cenários

apresentacao.

O projeto do Gerador de Cenários foi motivado pela a dificuldade de criação e edição de cenários do SIMULOP. Devido à complexidade e ao alto esforço exigido para a criação de um cenário utilizando o SAGE/SIMULOP, consome-se muito tempo para a finalização de todo o cenário e, ainda, corre-se o risco de haver erros durante a realização de tal tarefa.

No projeto do Gerador de Cenários desenvolvido pelo LIHM/DEE e financiado pela CHESF, objetiva-se simplificar o processo de criação de cenários, bem como, evitar a ocorrência de erros durante esse processo. Além disso, existe uma Base de Casos em que cenários prontos são armazenados. Tal banco deverá facilitar o acesso, o compartilhamento e a gestão de um acervo de cenários que reproduzem tanto situações onde o sistema elétrico opera em condições normais, quanto em situação de contingência. Os cenários serão elaborados para operadores de instalações (subestações) e de centros de operação (sistema), da empresa CHESF.

Há também um armazenamento local dos cenários na ferramenta Gerador de Cenários. O tutor pode, a qualquer momento durante a construção do cenário, interromper seu trabalho e salvá-lo numa base de dados local. A recuperação de um cenário é permitida a partir da seleção de um arquivo no formato owl. O armazenamento local se dá em uma base de conhecimento, estruturada na forma de uma ontologia.

3.1 Análise de Requisitos

Em busca de um melhor desempenho da ferramenta, o projeto Gerador de Cenários possui uma base de conhecimento local onde estão presentes todos os dados necessários para a criação de um novo cenário. O fato do *software* fazer buscas locais, ao invés de a um servidor, torna as consultas mais rápidas e independente de qualquer conexão externa.

A qualquer momento, o tutor pode salvar e recuperar um cenário em desenvolvimento. Quando o cenário de treinamento estiver finalizado, o responsável por aquele cenário pode disponibilizá-lo na base de casos para que os demais usuários do Gerador tenham acesso a ele.

3.2 Arquitetura

Há dois modos possíveis de utilizar o Gerador de Cenários, um *online*, em que é necessária a conexão com o servidor da empresa, e outro *offline*, independente de qualquer comunicação externa.

No modo *online*, é necessário que o tutor "faça login" no Gerador de Cenários. Ao fazer o login, ocorre automaticamente um mapeamento dos dados contidos no banco de dados do projeto para o arquivo owl base presente localmente na máquina. Devido à quantidade de informação que deve ser passada do banco para o arquivo, esse mapeamento leva alguns minutos. Entretanto, esse é um tempo que é facilmente compensado quando as consultas passam a ser locais, ao invés de externas.

Desta forma, durante a criação ou edição de um cenário, a busca de dados precisar ser feita, ela será realizada a partir do arquivo owl local. Sem a necessidade de estar se comunicando frequentemente com o servidor, a ferramenta ganha em desempenho e independência.

Após a finalização do cenário, pode-se salvá-lo na base de casos do projeto para que outros tutores tenham acesso a ele, tanto para consulta, quanto para a criação de outros cenários a partir daquele.

No modo *offline*, não é necessário passar por nenhum tipo de controle de acesso. O arquivo owl base contém todas as informações necessárias para a criação de um novo cenário. Esse arquivo é atualizado todas as vezes que a ferramenta é conectada no modo *online*. Portanto, é de responsabilidade do tutor a atualização do arquivo para que ele

possa desenvolver o cenário da sua escolha. Caso este arquivo não esteja atualizado, o tutor deve fazer o login e esperar que o mapeamento do banco de dados para o arquivo seja realizado.

A geração de um cenário é, então, realizada de forma rápida, pois não é necessária a comunicação com qualquer servidor externo. E, além disso, o tutor fica livre para trabalhar em qualquer ambiente que possua o Gerador instalado, independente de haver comunicação com a intranet da empresa.

Durante o processo de edição de um cenário de treinamento, o usuário pode salvar o cenário em um arquivo local, ou na base de casos, caso esteja no modo *online*, e fechar a ferramenta de geração de cenários. Quando o tutor quiser retomar a construção desse cenário, basta selecionar o arquivo salvo localmente, ou o cenário salvo na base de casos, e todas as informações relativas a esse cenário serão recuperadas.

Na figura 3.1 apresenta-se essa relação entre o Gerador, o Servidor e a Ontologia. A API desenvolvida atua no momento em que o usuário conecta-se ao servidor da ferramenta, de forma a atualizar o arquivo owl base com os dados presentes no banco de dados. Além disso, o Gerador faz uso da API no momento em que ele faz as consultas às informações necessárias para a criação de um cenário de treinamento. Exemplos dessas informações, que estão presentes em um arquivo owl local, são equipamentos e sinalizações.

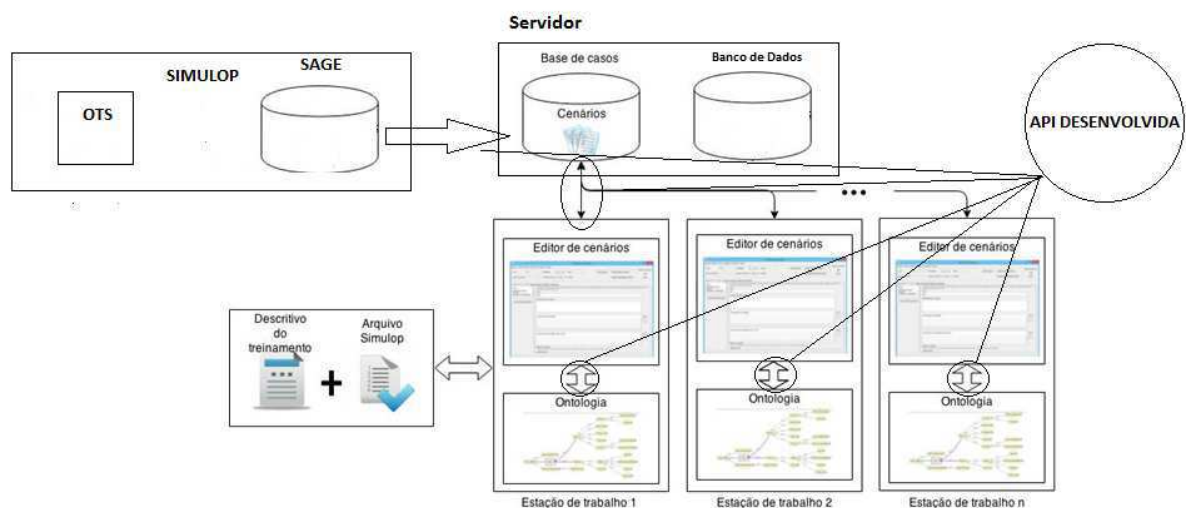


Figura 3.1: Representação da Arquitetura do Gerador de Cenários

Na figura 3.2 apresenta-se o diagrama de sequencia para salvar e disponibilizar um cenário na Base de Casos.

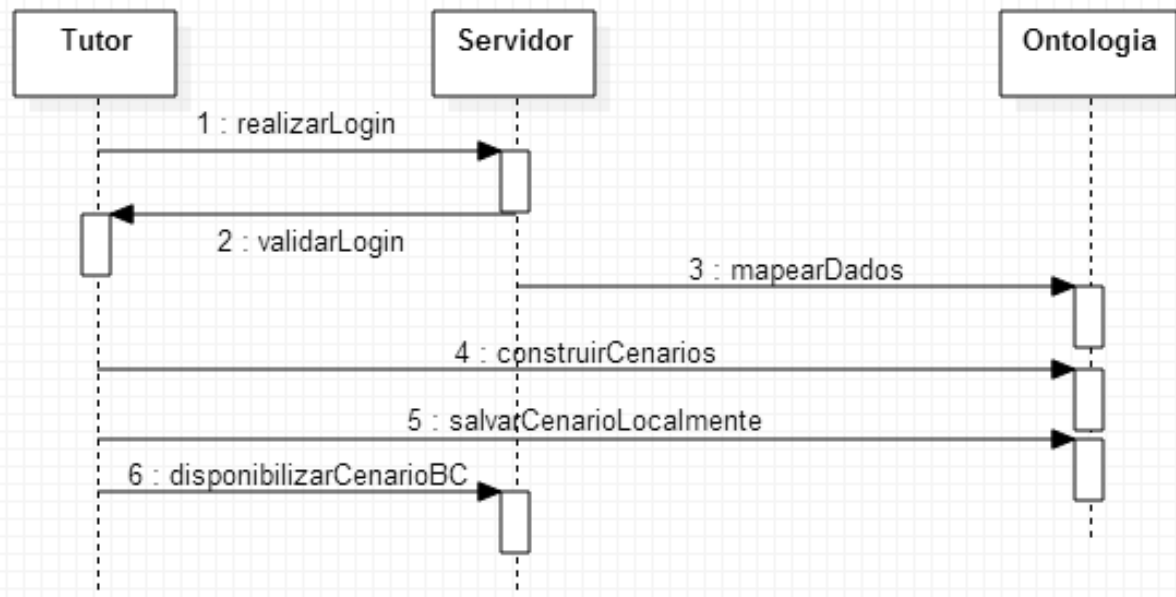


Figura 3.2: Diagrama UML de sequência para salvar e disponibilizar um cenário na base de casos

3.3 Ontologia de Cenários de Treinamento

Através da ferramenta *Protégé*, foi desenvolvida a ontologia do cenário de treinamento. Nela, estão contidas todas as classes e propriedades necessárias para descrever, gerar e salvar um cenário de treinamento para o ambiente SIMULOP, como pode-se observar na figura 3.3.

Foi criada uma classe, conhecida como classe fachada da ontologia, para abrigar todos os métodos que manipulem, de alguma forma, a ontologia. Por exemplo, métodos para o salvamento e a recuperação de cenários locais estão nessa classe. A lista de métodos e atributos dessa classe pode ser visto na figura 3.4.

No diagrama de classe da figura 3.4, foram destacados alguns métodos. Saber como eles funcionam é importante para compreender melhor como a API foi construída e como ela pode ser expandida.

FacadeOntologia

construtor da classe que recebe o diretório de um arquivo owl. Esse diretório é o cenário local selecionado pelo usuário para edição na ferramenta do Gerador.

salvarCenarioOntologia

salva os dados um cenário na ontologia.

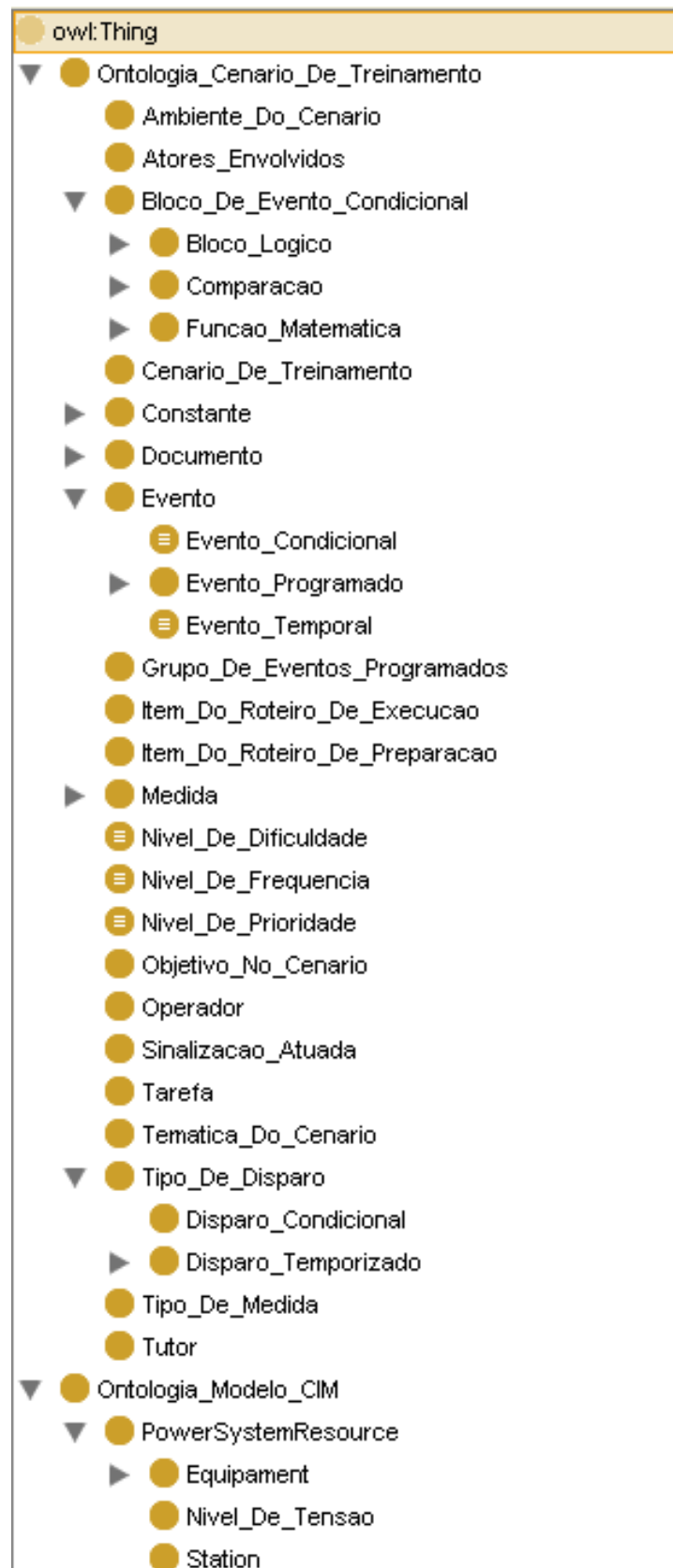


Figura 3.3: Hierarquia de Classes da Ontologia de Domínio

FacadeOntologia
<pre> + url : String + ns : String + ont : OntModel + FacadeOntologia () : void + FacadeOntologia (String urlSelecionada) : void - salvarCenarioOntologia (String nomeTutor, String siglaInstalacao, String titulo, String objGeral, String objEspecifico, Time duracao, String configAntesCenario, String configDepoisCenario, String descricaoCenario, String preRequisitos, String protecoesAtuadas, String protecoesAtuadasChassi, ArrayList<String> disjAbertosBloqueados, ArrayList<String> disjAbertosNaoBloqueados, Date dataCriacao, Date dataUltEdicao, ArrayList<String> listNomeAtores, ArrayList<String> listDescricaoAtores, ArrayList<String> listQuemAtua, ArrayList<String> listComoAtua, ArrayList<String> listOndeAtua, ArrayList<String> listRoteiroExecucao, ArrayList<String> listDescricaoTarefa, ArrayList<String> listTipoTarefa, ArrayList<String> listPrioridadeTarefa, ArrayList<String> listDificuldadeTarefa, ArrayList<String> listFrequenciaTarefa) : void + salvarAtoresOntologia (String nomeAtores, String descricaoAtores, int i) : Individual - salvarEquipamentoOntologia (String tipoEquipamento, String nivel_tensao, String subestacao, String codigo_equipamento) : void + salvarItemRoteiroPreparacaoOntologia (String quemAtua, String comoAtua, String ondeAtua, int i) : Individual + salvarItemRoteiroExecucaoOntologia (String acaoRoteiroExecucao, int i) : Individual + salvarTarefaOntologia (String descricaoTarefa, String tipoTarefa, String dificuldadeTarefa, String frequenciaTarefa, String prioridadeTarefa, int i) : Individual - salvarOntologia (String endereco) : void + verificaEquipamento (String equipamento) : boolean - salvarGrupoEventos (ArrayList<GrupoDeEventos> listGrupoEventos, ArrayList<Condicionalis> listCondicoes, ArrayList<Medidas> listMedidas) : void + getSubClass (OntClass tipoMedidaClass, String tipoMedida) : OntClass + salvarCondicao (Condicionalis condicao, ArrayList<Medidas> listMedidas, int i) : Individual + convertTempoIDCondToTime (String strID) : Time + convertStringToTime (String duracaoString) : Time + converteOntTempoEmString (RDFNode propertyValue) : String - recuperarCenario() : CenarioDescricao + getCondicaoDisparo (Individual tipoDisparoIndividual) : String - recuperarCondicionalis() : ArrayList<Condicionalis> + getTipoEntradaPelaClass (OntClass entradaClass) : String </pre>

Figura 3.4: Lista de Métodos e Atributos da Classe Fachada da Ontologia

salvarEquipamentoOntologia

mapeia os dados do banco de dados para a ontologia.

salvarOntologia

recebe um diretório, escolhido pelo usuário da ferramenta, para que o cenário seja salvo naquele local.

recuperarCenario

recupera para edição os dados de um cenário armazenado localmente.

3.4 Modelo Entidade Relacionamento

O Modelo Entidade Relacionamento (MER) do Gerador de Cenários, para efeitos didáticos, pode ser dividido em três partes: as entidades relativas a um equipamento, cenário e evento, como pode-se observar nas figuras 3.5, 3.6 e 3.7.

Na figura 3.5, a tabela "equipamento" possui atributos para salvar informações como o código e o tipo daquele equipamento, além de qual instalação e a qual nível

de tensão ele pertence. O atributo "ativo" define se aquele equipamento ainda existe no SAGE/SIMULOP. Caso ele tenha sido removido do supervisor, não deve-se simplesmente removê-lo do banco de dados, pois isso afetaria os cenários em que aquele equipamento fizesse parte.

Ainda na figura 3.5, existem duas tabelas para relacionar o tipo de equipamento e as ações que ele pode realizar ("tipo_acao_tipo_equipamento"), bem como as medidas que podem ser efetuadas nele ("tipo_medida_tipo_equipamento"). Essas tabelas são importantes, pois evitam que sejam criados cenários com ações ou medidas inconsistentes com um determinado equipamento.

A tabela "sinalizacao", presente na figura 3.5, guarda todas as mensagens de sinalização que podem aparecer, geralmente no painel do SAGE, durante a realização de um cenário. Essas mensagens foram retiradas de arquivos do SIMULOP e passadas para o BD e para a ontologia através do Gerador de Cenários.

A tabela "variavel_medida", como pode ser visto na figura 3.5, tem a função de salvar os dados de uma medida no momento que ela é criada. Nota-se a relação entre a esta entidade e a entidade "cenario", significando que uma medida existe, independente de ela estar relacionada a uma condição.

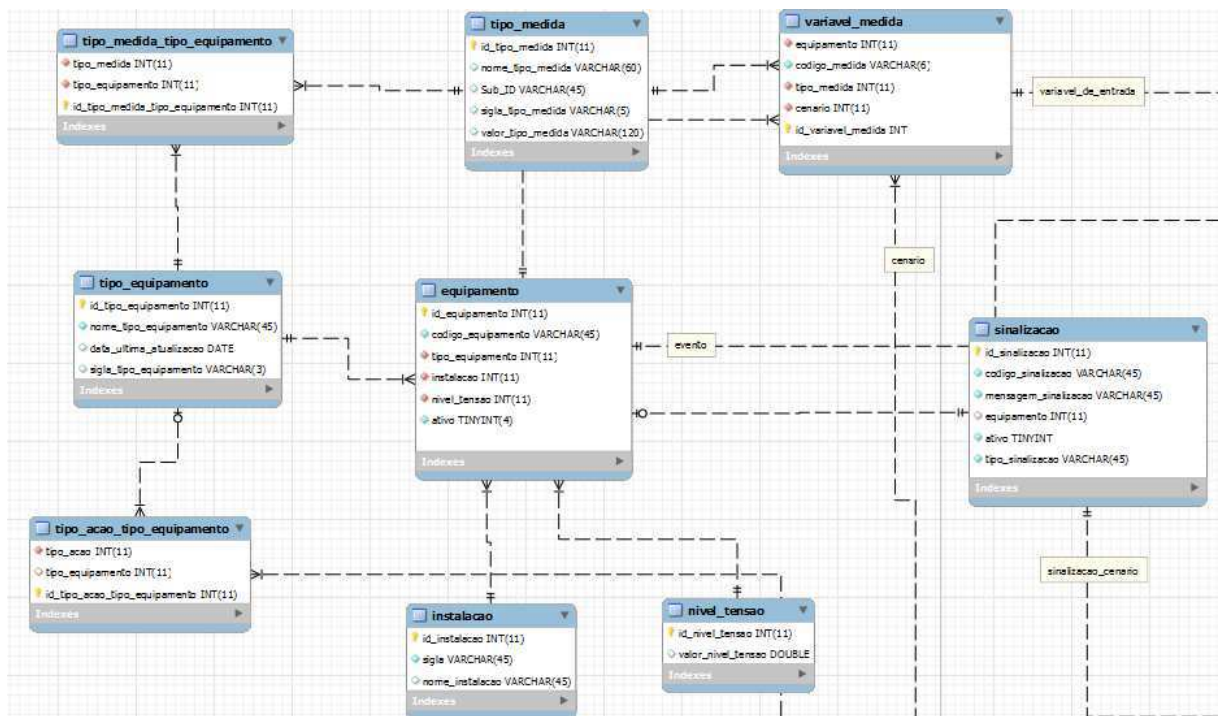


Figura 3.5: Parte 1 do Modelo do BD

Na figura 3.6, a tabela "cenario" possui todos os atributos para que se possa descre-

ver um cenário, como suas tarefas, as mensagens de sinalização e o tutor que o desenvolveu. A entidade "tutor" possui, além do nome e da matrícula de um tutor, atributos para controle de acesso. O usuário do Gerador que deseja ter acesso à base de casos, precisa "realizar um login". Os cenários ao qual ele tem acesso, bem como as ações que ele pode realizar na ferramenta são definidos pelo atributo "permissao".

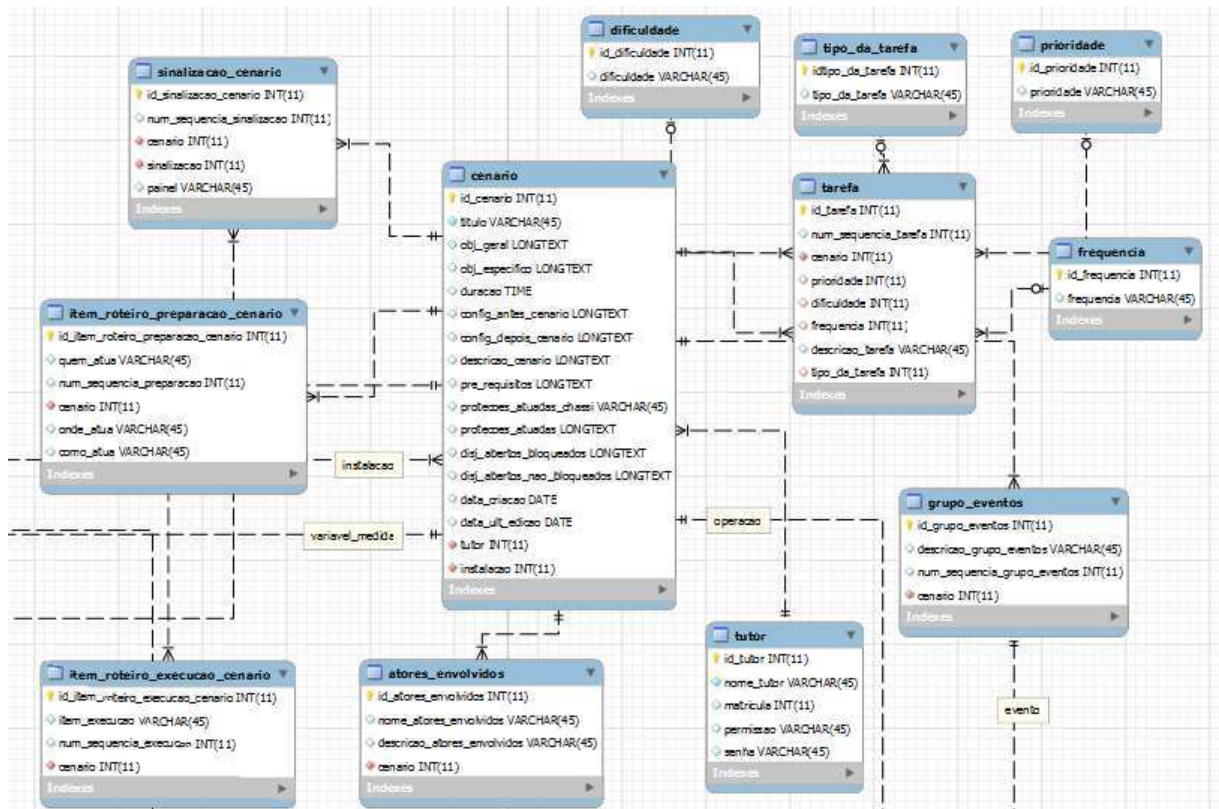


Figura 3.6: Parte 2 do Modelo do BD

Pode-se observar na figura 3.7 que a entidade "evento" tem informações sobre qual ação e em qual equipamento aquele evento vai acontecer. Também fazem parte da tabela, a mensagem que pode aparecer quando da ocorrência do evento e o tipo de disparo (temporal absoluto, temporal relativo ou condicional, repetitivo ou não) daquele evento.

Da mesma forma que há uma classe fachada para a manipulação da ontologia, também existe uma para o banco de dados. A classe fachada do banco de dados contém todos os métodos que façam consultas ou atualização na base de casos. A lista de métodos e atributos dessa classe está representada na figura 3.8.

No diagrama de classe da figura 3.8 foram destacados alguns métodos que são diferentes em relação aos já comentados da fachada da ontologia ilustrados no diagrama de classe da figura 3.4. Há vários métodos da fachada do BD com ideias similares aos

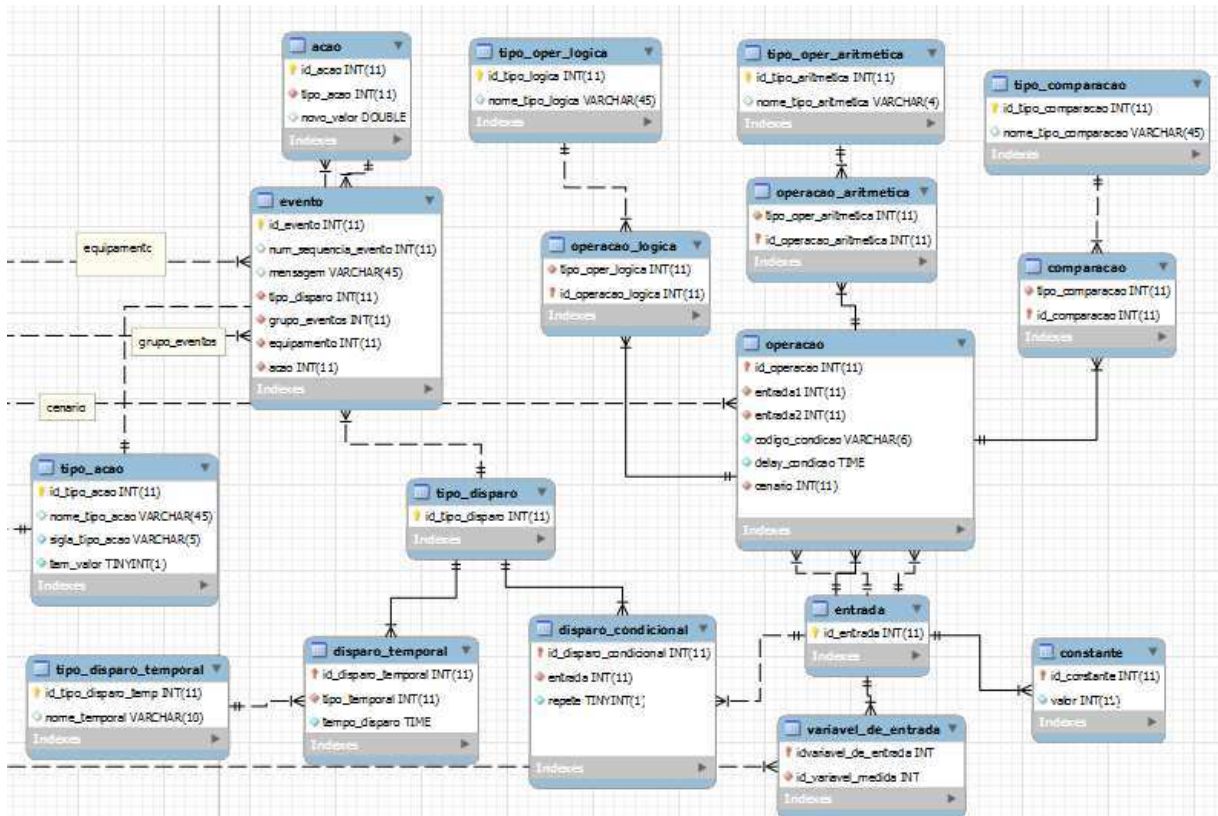


Figura 3.7: Parte 3 do Modelo do BD

da fachada da ontologia. Por isso, não houve muito esforço adicional para a construção desses métodos. Por exemplo, o algoritmo para salvar o cenário na base de casos e na base de conhecimento é o mesmo, havendo mudança apenas na sintaxe do código.

startOperation

inicializa as variáveis do *Hibernate* necessárias para fazer a persistência dos dados. Toda função que fizer alguma atualização no banco de dados, deve inicialmente chamar esse método.

salvarEquipamento

recebe um arquivo contendo uma lista de equipamentos de algum tipo (transformador, disjuntor etc) e atualiza a tabela do banco de dados que contém a lista de todos os equipamentos necessários para a criação de um cenário.

getCenarios

filtra os cenários de acordo com título, subestação e o tutor que o construiu.

atualizarCenario

atualiza os dados modificados de um cenário já existente no banco de dados.

facade
<p>+ session : Session + tx : Transaction</p>
<p>+ Facade () : void + startOperation () : void</p>
<p>+ salvarCenario (String nomeTutor, String siglaInstalacao, String titulo, String objGeral, String objEspecifico, Date duracao, String configAntesCenario, String configDepoisCenario, String descricaoCenario, String preRequisitos, String protecoesAtuadas, String protecoesAtuadasChassi, String disjAbertosBloqueados, String disjAbertosNaoBloqueados, Date dataCriacao, Date dataUltEdicao) : void + checarNivelTensao (ArrayList<Double> nivelTensao) : void + checarInstalacao (ArrayList<String> siglaInstalacao) : void</p>
<p>- salvarEquipamento (String tipoEquipamento, ArrayList<Double> nivelTensao, ArrayList<String> siglaInstalacao, ArrayList<String> codigoEquipamento) : void + novaDataAtualizacaoEqpm (TipoEquipamento tipoEquipamento) : void - salvarTipoTarefa (String opcao) : void - salvarCondicionalis (ArrayList<Condicionalis> condicao, ArrayList<Medidas> listMedidas) - salvarMedidas (ArrayList<Medidas> listaMedidasFromIHM) : void - carregarMedidas (String medidaID) : Medidas - salvarMedida (ArrayList<Medidas> arrayListMedidas) : void - cadastrarTutor (Tutor tutor) : void - getDadosEquipamento (String codigoEquipamento) : List<Equipamento> - salvarSinalizacaoCenario (ArrayList<String> listPainel, ArrayList<String> listID, ArrayList<String> listMensagem) : void - getSiglaTipoAcao (int tipoEquipamentold) : List<TipoAcaoTipoEquipamento> - salvarRoteiroPreparacao (ArrayList<String> quemAtua, ArrayList<String> comoAtua, ArrayList<String> ondeAtua) : void - salvarRoteiroExecucao (ArrayList<String> listRoteiroExecucao) : void - salvarGrupoDeEventos (ArrayList<GruposDeEventos> gruposDeEventos, ArrayList<Medidas> arrayListMedidas, ArrayList<Condicionalis> arrayListCondicionalis, int cenarioID) : void - carregarGruposDeEvento (int idCenario) : ArrayList<GruposDeEventos> - salvarTarefa (ArrayList<String> listDescricaoTarefa, ArrayList<String> listTipoTarefa, ArrayList<String> listPrioridadeTarefa, ArrayList<String> listDificuldadeTarefa, ArrayList<String> listFrequenciaTarefa) : void - getTipoMedida (int tipoEquipamentold) : List<TipoMedidaTipoEquipamento> : void - salvarTipoDaTarefa (String nomeTipoDaTarefa) : void - listAllInstalacoes () : ArrayList<String> - listAllTipoEquipamento () : ArrayList<String> - listAllTipoEquipamentoPossuiTipoAcao () : ArrayList<String></p>
<p>- getCenarios (String nomeCenario, String subestacao, String nomeTutor) : List<Cenario> - getCenarioPorID (int id) : Cenario</p>
<p>- getItemRoteiroExecucao (Cenario cenario) : String - getItemRoteiroPreparacao (Cenario cenario) : List<ItemRoteiroPreparacaoCenario> - getTarefaPorCenario (Cenario cenario) : List<Tarefa> - getSinalizacaoPorCenario (cenario cenario) : List<SinalizacaoCenario> - salvarAtores (ArrayList<String> listNomeAtores, ArrayList<String> listDescricaoAtores) : void - getAtoresPorCenario (Cenario cenario) : List<AtoresEnvolvidos></p>
<p>- atualizarCenario (String nomeTutor, String subestacao, String titulo, String objGeral, String listObjEspecificos, Date duracao, String listConfigInstalacao, String listConfigInstalacaoAposEvento, String descricao, String listRequisitos, String listProtecoes, String listProtecoesChassi, String listDisjBloqueados, String listDisjNaoBloqueados, Date dataAtualizacao, Cenario cenario) : void - atualizarTarefas (Cenario cenario, ArrayList<String> listDescricaoTarefa, ArrayList<String> listTipoTarefa, ArrayList<String> listPrioridadeTarefa, ArrayList<String> listDificuldadeTarefa, ArrayList<String> listFrequenciaTarefa) : void - atualizarRoteiroExecucao (Cenario cenario, ArrayList<String> listRoteiroExecucao) : void - atualizarRoteiroPreparacao (Cenario cenario, ArrayList<String> quemAtua, ArrayList<String> comoAtua, ArrayList<String> ondeAtua) : void - atualizarAtores (Cenario cenario, ArrayList<String> listNomeAtores, ArrayList<String> listDescricaoAtores) : void - atualizarSinalizacaoCenario (Cenario cenario, ArrayList<String> listPainelSinalizacao, ArrayList<String> listDescricaoSinalizacao, ArrayList<String> listQuadroRelesSinalizacao) : void - getUltimoCenario () : Cenario - getEquipamentosPorSubTensaoTipoCodigo (String subestacao, int rangeTensao, String tipoEqpm, String codigo) : List<Equipamento> - getDataUltAtualizacao (String tipoEqpm) : String - salvarSinalizacao (ArrayList<String> listCodigo, ArrayList<String> listMensagem, ArrayList<String> listEqp, String tipoSinalizacao) : void - buscarSinalizacoesAtivasPorCodigoEquipamento (String codigoEquipamento) : List<Sinalizacao> : void - coisdoEquipamentoExiste (String codigoEquipamento) : boolean - getAllCodigoEquipamentoAtivos() : ArrayList<String></p>
<p>- excluirCenarioPorID (int i) : void - excluirObjetosRelacionadosACenario (Cenario cenario) : void - excluirAtoresEnvolvidosPorCenario (Cenario cenario) : void - excluirItemRoteiroExecucaoPorCenario (Cenario cenario) : void - excluirItemRoteiroPreparacaoPorCenario (Cenario cenario) : void - excluirTarefaPorCenario (Cenario cenario) : void - excluirGrupoEventosPorCenario (Cenario cenario) : void - excluirEventosPorGrupoEventos (GrupoEventos grupoEventos) : void - getAllEquipamentosPorTipo (String nomeTipoEquipamento) : ArrayList<String> - padronizarGrupoDeEventos (ArrayList<GruposDeEventos> arrayList) : ArrayList<GruposDeEventos> - getAllSinalizacoes () : List<Sinalizacao> - limparTabelaSinalizacoes (String digital) : void - getTipoEquipamentoPorEquipamento (int tipoEquipamentold) : String - listAllTipoEquipamentoPossuiEquipamento() : List<String></p>

Figura 3.8: Lista de Métodos e Atributos da Fachada do BD

excluirCenarioPorID

exclui um cenário selecionado pelo usuário do banco de dados.

Para maiores detalhes, pode-se consultar o dicionário de dados do banco de dados que se encontra no Apêndice A.

3.5 API para Mapeamento de Dados

Todos os dados, sobre o sistema elétrico CHESF, necessários para a criação de um cenário de treinamento foram retirados do SAGE/SIMULOP. Foram criados *parsers*, isto é, "tradutores" dos arquivos retirados do SAGE para povoar o banco de dados. Desta forma, é de responsabilidade de um administrador do banco de dados da ferramenta, realizar *upload* desses arquivos para que o banco se mantenha atualizado em relação ao sistema supervisorio da CHESF. O arquivo owl local é atualizado todas as vezes que o tutor efetua o login e conecta-se à base de dados, por meio de um mapeamento do banco de dados para a ontologia.

Assim, informações como as de equipamentos, subestações e sinalizações que estão presentes no SIMULOP podem ser acessadas, através do BD ou da ontologia de domínio, para a geração de cenários.

Quando novos dados forem adicionados ou atualizados no SAGE/SIMULOP, o tutor deve exportar os arquivos contendo essas novas informações e importá-los para a ferramenta, de modo que o Gerador esteja sempre atualizado em relação ao sistema supervisorio da CHESF.

Na figura 3.9 apresenta-se o diagrama UML de casos de uso para o mapeamento de dados do SAGE para o banco de dados e do BD para a ontologia.

O próximo capítulo trata da validação da API proposta. Um cenário foi salvo e recuperado e os resultados foram comentados.

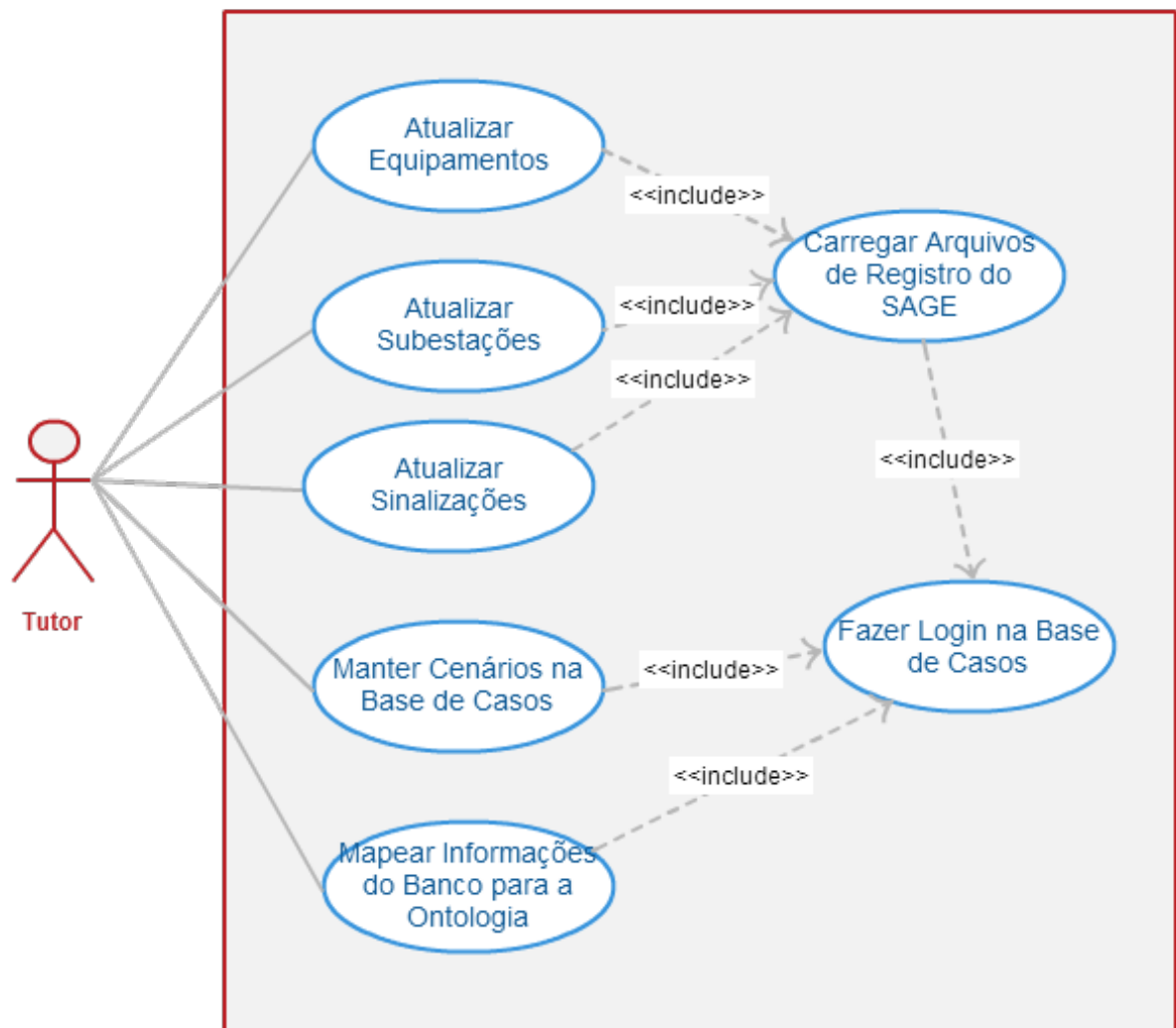


Figura 3.9: Casos de Uso: Atualização do BD/Ontologia/Base de Casos

Capítulo 4

Validação - Instanciação de Cenários

As duas classes fachadas que manipulam todos os dados da ontologia e do banco de dados foram testados através do *EasyAccept* (6). *EasyAccept* é uma ferramenta grátis e de código aberto que ajuda a criar e executar testes de aceitação de uma maneira fácil e rápida. Por testes de aceitação, entende-se um teste de caixa-preta em que vários tipos de entradas diferentes são inseridas por meio da classe fachada e as saídas são observadas.

Então, ao longo do desenvolvimento das fachadas, por meio do *EasyAccept*, os métodos dessas classes foram testados para que erros e saídas indevidas fossem evitados.

Um cenário já existente foi escolhido e, por meio do Gerador de Cenários, o recriou-se com a finalidade da validação da ferramenta. Todo o descritivo, as medidas, condições e eventos foram criados e salvos, utilizando a API desenvolvida, com sucesso, tanto na base de casos quanto em um arquivo owl local.

O cenário escolhido tinha como título "Evento 1" e seu objetivo era o desligamento do transformador 04T5 da subestação CGD devido a um curto circuito no TP 82T5 fase B (quebra de isoladores). Os descritivo completo do cenário pode ser visto nas figuras 4.1 e 4.2.

O diagrama unifilar da subestação em que ocorre o evento, a CGD, está ilustrado na figura 4.3.


	Roteiro do Evento Simulado	Área: Operação de Instalação
	EVENTO-1	Processo: Gestão Administrativa
	□	Código: F.O.GA.OPI.01.00
Objetivo Geral		
□		
Objetivo Específico		
Treinar/capacitar a equipe de Operadores de Tempo Real objetivando atingir a eficiência e eficácia na recomposição de Equipamentos/Subestação, através dos aspectos: Flash, preenchimento dos anexos I/II/III, identificação da sinalização atuada, inspeção nos equipamentos, caracterização do impedimento e verificação da comunicação verbal.		
Verificar conhecimento da área de atuação da proteção diferencial do trafo-04T5		
Configuração da Instalação:		
Configuração normal da instalação		
□		
□		
□		
□		
Descrição do Evento:		
Desligamento do trafo-04T5 da SE CGD devido curto-circuito no TP-82T5 fase B (quebra de isoladores).		
□		
Tempo de Duração Esperado:		
00:30:00		
Configuração após o Evento:		
Trafo-04T5 desenergizados com disjuntores-14T5 e-12T5 abertos.		
□		
□		
Proteções Atuadas:		
87 fase "B"		
86-04T5		
□		
Disjuntores Abertos e Bloqueados:		
14T5, 12T5		
□		
□		
Disjuntores Abertos e Não Bloqueados:		
Nenhum		
□		

Figura 4.1: Descritivo do cenário de treinamento parte 1


	Roteiro do Evento Simulado	Área: Operação de Instalação	
	EVENTO-1	Processo: Gestão Administrativa	
	□	Código: F.O.GA.OPI.01.00	
Principais Sinalizações			
Painel	Descrição do Anunciador	Quadro de Relés Auxiliares	
SAGE	CGD:04T5:87;-PDIF:-Tripe-Relé Diferencial-87	Quadro Anunciador	
□	CGD:04T5:86:ATRD:-Atuação-relé-de-bloqueio-lado-230kV	□	
□	CGD:04T5:86:ATRB:-Atuação-relé-de-bloqueio-lado-	□	
□	CGD:04T5:86:ATRS:-Atuação-relé-de-bloqueio-lado-69kV	□	
□	CGD:04T5:86:TRF0:-Trafo-fora-de-operação	□	
Principais Proteções Atuadas no Chassi de Proteção:			
04T5-87-B			
Roteiro de Preparação do Cenário			
□	□	□	
Roteiro de Execução do Evento			
□	Ação	Comentários	Horário
□	Efetuar inspeção local entre o-94T5-e-o-92T5-para-identificar-o-defeito-e-isolar-trafo-04T5;	□	□
□	OPI-informar-ao-SLOG-Ocorrência;	□	□
□	Comunicar-ao-OPS,-em-até-01-minuto,-as-informações-preliminares-(flash)-04T5-indisponível;	□	□
□	Analisar-a-ocorrência-com-base-nos-formulários-preenchidos;	□	□
□	Silenciar-os-alarmes-sonoros-e-registrar-o-horário-da-ocorrência;	□	□
□	Comunicar-ao-OPS-as-informações-adicionais-e-as-ações-adoptadas-(trafo-04T5-isolado);	□	□
□	Preencher-e/ou-imprimir-o-formulário-dos-Anexos-I,-II-e-III;	□	□
□	Comunicar-ao-OPS,-em-até-dez-minutos,-as-informações-complementares;	□	□
□	Identificar-se-o-desligamento-foi-"geral"-ou-"parcial";	□	□

Figura 4.2: Descritivo do cenário de treinamento parte 2

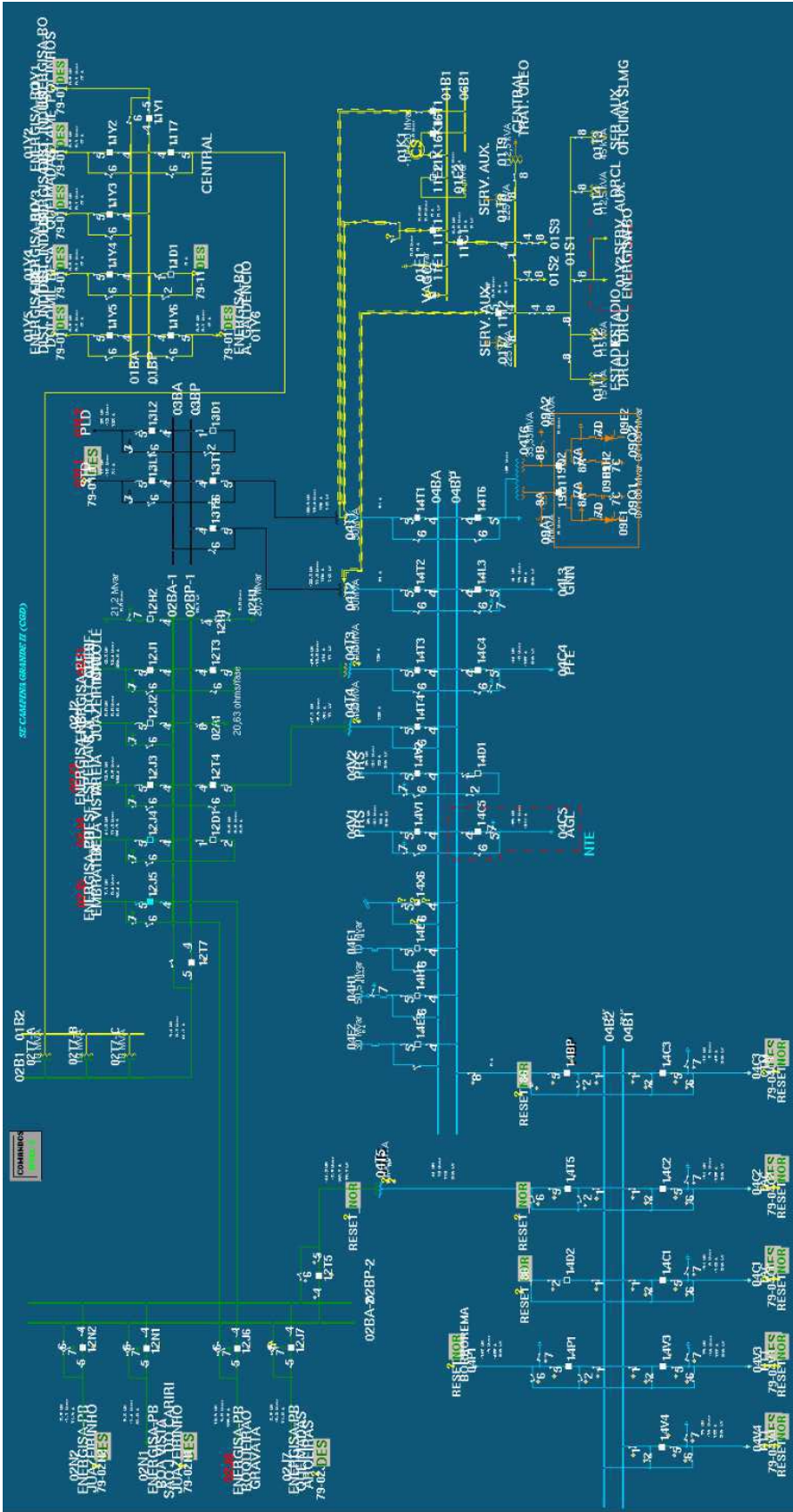


Figura 4.3: Diagrama Unifilar de CGD Retirado do SAGE

A construção do cenário se iniciou no momento em que "fizemos login"na ferramenta. Graças à API desenvolvida neste trabalho, houve o mapeamento de dados do banco de dados para a ontologia e, a partir de então, todas as nossas consultas eram realizadas de forma local.

A partir de então, foi realizada a descrição de cenário, listando seus objetivos, proteções atuadas, sinalizações, etc.

Em seguida, foi criada uma medida de id "med01"para detectar a tensão no transformador 04T5. Feita essa medida, criou-se uma condição de id "cond01"e do tipo comparação para ser *true* quando "med01"fosse maior que zero.

Então, um grupo de evento foi construído, e, em seguida, fizemos os eventos relacionados a esse grupo. Esses eventos constituíam em abrir e bloquear os disjuntores 14T5 e 12T5 assim que a "cond01"torna-se *true*, desligando assim o transformador 04T5.

As figuras 4.4, 4.5, 4.6, 4.7 apresentam o processo de criação do cenário através do Gerador.

Temática do Cenário

A temática do cenário de treinamento é definida a partir de um título; e da descrição dos estados inicial e final do ambiente de simulação, respectivamente antes e depois da execução dos eventos.

Título:
EVENTO 1

Descrição geral do cenário
Desligamento do trafo 04T5 da SE CGD devido curto-circuito no TP 82T5 fase B (quebra de isoladores).

Configuração da Instalação +
Configuração normal da instalação -

Configuração da Instalação após o Evento +
Trafo 04T5 desenergizados com disjuntores 14T5 e 12T5 abertos. -

Figura 4.4: Parte do descritivo mostrado na ferramenta Gerador de Cenários

O arquivo do cenário que "sobe"para o SAGE, isto é, aquele que o SIMULOP lê para executar o cenário de treinamento pode ser observado na figura 4.8. Esse arquivo foi carregado no SAGE e o cenário foi executado como planejado através do Gerador de Cenários..

A recuperação para possíveis edições ou consultas de cenário também foi realizada

Detalhes

ID da Medida

med01

ID do Equipamento

CGD:04T501

Tipo do Equipamento

Sub-ID

HIGH

Tipo de Medida

KV

Figura 4.5: Medida criada para a geração do cenário

Entrada 1

med01

Entrada 2

0

Condicao

maior

Condicao ID

cond01

Tempo

1

Figura 4.6: Condicional criada para a geração do cenário

Eventos			
Equipamento	Ação	Disparo	Tempo / Condição
CGD:12T5:52	BKD	C	cond01
CGD:14T4:52	BKD	C	cond01
CGD:14T4:52	BKT	C	cond01
CGD:12T5:52	BKT	C	cond01

Figura 4.7: Tabela de eventos criada para a geração do cenário

```

* Export Event Library 00/00/00 00:00:00
* OIS Model EXP Database eprirts/db/DBBA
* OIS Date 00/00/0000 TIME 00:00:00
* Maximum Number of Event Groups 20
* Maximum Number of Events Per Group 99
*
* ---- EVENT GROUP ---- 001
*
G:Abertura e bloqueio dos disjuntores relacionados ao trafo 04T5 quando da energização deste.
E:C:cond01:BKT :CGD :CGD:12T5:52 : : !
E:C:cond01:BKT :CGD :CGD:14T4:52 : : !
E:C:cond01:BKD :CGD :CGD:12T5:52 : : !
E:C:cond01:BKD :CGD :CGD:14T4:52 : : !
*
* ---- CONDITIONAL EVENT BLOCKS ----
*
M:med01 :XF:CGD :CGD:04T501 :HIGH:KV !
C:cond01: 1:M:med01 :GT:C:0 : :

```

Figura 4.8: Arquivo que é carregado no SAGE

com sucesso. O descritivo, a medida, a condição e os eventos foram exibidos da mesma maneira que foram salvos.

Capítulo 5

Conclusões

A ontologia de domínio, além de ser útil para a especificação de conceitos, mostrou ser também uma poderosa ferramenta para armazenamento de dados locais.

Com as consultas às informações necessárias a criação de um cenário sendo realizadas localmente, por meio da ontologia de domínio, observa-se um claro ganho de desempenho da ferramenta. A comunicação a todo instante com o servidor não é mais necessária.

Outra vantagem do uso da ontologia para descrição e armazenamento de cenários locais é uma maior independência do Gerador de Cenários, visto que, não é obrigatório haver uma conexão externa, como para a intranet da empresa, para um devido funcionamento da ferramenta. Pode-se trabalhar normalmente sem essa conexão e, quando a comunicação com o base de casos for realizada, a disponibilização do cenário que estava sendo criado pode ser realizada. A base de casos também demonstra ser uma ferramenta de importante utilidade, pois cenários podem ser acessados remotamente desde que tenham sido previamente disponibilizados nela.

Como trabalho futuro, tem-se a ampliação do Gerador para ser aplicado a cenários de treinamento de operação de sistemas. Atualmente, é possível a criação de cenários apenas no nível de subestações. Quando a ferramenta se expandir a ponto de ser utilizada pelos operadores de sistemas, a API, bem como o banco de dados da ferramenta, deverão ser modificados. Deverão ser adicionadas novas classes e atributos à ontologia de domínio, bem como novas tabelas ao banco de dados. Entretanto, o processo não difere do que já foi realizado até aqui. Podem ser utilizadas as mesmas ferramentas e *framework*, como o *Protégé* e o *Hibernate* para facilitar a expansão.

Referências Bibliográficas

- 1 M. F. Q. VIEIRA, J. A. N. NETO, A. SCAICO, C. SANTONI, and J. M. MERCANTINI. A model based operator training simulator to support human behavior studies. *transactions of the society for computer simulation*. 86:41–51, 2010.
- 2 C. R. R. Leite, J.J.R. de Oliveira, and J. G. de Oliveira. O uso de simuladores no treinamento de operadores da chesf como ferramenta para disseminação de conhecimentos na operação do sistema elétrico. *II Seminário Internacional–Reestruturação e regulação do setor de energia elétrica e gás natural*, 2007.
- 3 A. V. S. NETTO, VIEIRA M. F. Q., F. T. FILHO, F. F. V. M. FERREIRA, and J. SORAGHAN. Ferramenta para avaliação do treinamento de operadores de subestação elétrica apoiada por simulador. *Simpósio Brasileiro de Sistemas Elétricos, SBSE 2014, 2014, Foz do Iguaçu. Anais do Simposio Brasileiro de Sistemas Elétricos, 2014*, 2014.
- 4 J. J. R. OLIVEIRA, L. C. LIMA, L. A. C. PEREIRA, R. B. SOLLERO, C. R. R. LEITE, R. B. MUNIZ, C.A.B. COSTA, M. S. CAVALCANTE, U. A. CARMO, and A. S. ARAÚJO. Treinamento e certificação de operadores no sistema sage empregando o simulador epri/ots. *Grupo de Estudos de Operação de Sistemas Elétricos, XVIII SNPTEE, Curitiba-Paraná*, 2005.
- 5 F. T. FILHO and M. F. Q. VIEIRA. Processo para o desenvolvimento de cenários de treinamento para ambientes virtuais 3d. *Simpósio Brasileiro de Automação Inteligente, SBAI 2013, 2013, Fortaleza-CE. Anais do Simposio Brasileiro de Automação Inteligente, 2013*, 2013.
- 6 EasyAccept. Python easyaccept. Disponível em: <http://easyaccept.sourceforge.net/>. Acessado em 06 de Abril de 2015.
- 7 E. P. X. P. FILHO. Programação das proteções do sistema elétrico no gerador de cenários de treinamento para o ambiente sage/simulop. 2014.
- 8 P. Hirsch. User guide for powersimulator with epri ots, 2004.
- 9 CEPEL. Guia de configuração, 2014.
- 10 A. V. S. NETTO. Avaliação do desempenho de operadores no treinamento da operação de sistemas elétricos em ambientes simulados. *Tese (Doutorado em Engenharia Elétrica) - Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Paraíba.*, 2014.
- 11 F. T. FILHO. Abordagem ontológica para modelagem do treinamento de operadores de sistemas elétricos. *Tese (Doutorado em Engenharia Elétrica) -- Centro de Engenharia Elétrica e Informática, Universidade Federal de Campina Grande, Paraíba.*, 2015.

- 12 T. R. GRUBER. A translation approach to portable ontology specifications. *In: Knowledge Acquisition*, pages 199–200, 1993.
- 13 M ALMEIDA and M. BAX. Uma visão geral sobre ontologias: pesquisa sobre definições, tipos, aplicações, métodos de avaliação e de construção. *Revista Ciência da Informação. Brasília, ano 3, n. 32*, pages 7–20, 2013. Disponível em: <http://www.scielo.br/pdf/ci/v32n3/19019>. Acesso em: 10 Dez 2013.
- 14 Julita Bermejo-Alonso, Ricardo Sanz, Manuel Rodríguez, and Carlos Hernández. An ontological framework for autonomous systems modelling. *International Journal on Advances in Intelligent Systems*, 3(3 and 4):211–225, 2011.
- 15 Y. SURE, S. STAAB, and STUDER R. *STAAB, S. and STUDER, R. (Ed.). International Handbooks on Information Systems. Berlin, Heidelberg: Springer-Verlag, 2009.*
- 16 M. HORRIDGE. A practical guide to building owl ontologies using the protege-owl plugin and co-ode tools, edition 1.3, 2011. Disponível em: http://130.88.198.11/tutorials/protegeowltutorial/resources/ProtegeOWLTutorialP4_v1_3.pdf. Acesso em: 30 Junho 2013.
- 17 JENA API. Jena2 ontology api. Disponível em: <http://jena.sourceforge.net/ontology/index.html>. Acessado em 01 de Abril de 2015.
- 18 A. A. FROZZA and R. GONÇALVES. Jena – um framework web semântico em java. Disponível em: <http://jena.sourceforge.net/ontology/index.html>. Acessado em 10 de Abril de 2015.
- 19 F. N. NOY and D. L. MCGUINNESS. *Ontology development 101: A guide to creating your first ontology, technical report*. 2001.
- 20 C. J. DATE. *Introdução a Sistemas de Banco de Dados*. Campus, 8th edition, 2004.
- 21 MySQL. Mysql database. Disponível em: <https://www.mysql.com/>. Acessado em 05 de Abril de 2015.
- 22 HIBERNATE. Hibernate framework. Disponível em: <http://hibernate.org/>. Acessado em 05 de Abril de 2015.

Apêndice A

Dicionário de Dados do BD

Segue o dicionário de dados do Banco De Dados. As entidades tem uma descrição do conceito e, caso exista, o conceito associado à ontologia. São listados os atributos por nome, tipo, conceito e pela possibilidade de conterem valores nulos.

Entidade	Conceito na Ontologia	Descrição do Conceito
acao		Ação programada para ocorrer com o disparo do evento.

Tabela A.1: Entidade acao

ATRIBUTO	TIPO	NULO	CONCEITO
id_acao	INT	NAO	PK de acao
tipo_acao	INT	NÃO	Tipo da ação (FK de tipo_acao)
novo_valor	DOUBLE	SIM	Caso a ação altere o valor de alguma variável do sistema, novo_valor guardará essa informação

Tabela A.2: Entidade acao

Entidade	Conceito na Ontologia	Descrição do Conceito
atores_envolvidos		Atores envolvidos em um cenário.

Tabela A.3: Entidade atores_envolvidos

ATRIBUTO	TIPO	NULO	CONCEITO
id_atores_envolvidos	INT	NÃO	PK de atores_envolvidos
cenario	INT	NÃO	Cenário associado ao ator.
descricao_atores_envolvidos	VARCHAR(45)	SIM	Descrição dos atores envolvidos.
nome_atores_envolvidos	VARCHAR(45)	SIM	Nome dos atores envolvidos.

Tabela A.4: Entidade atores_envolvidos

Entidade	Conceito na Ontologia	Descrição do Conceito
cenario	Cenário de treinamento	Situação de treinamento criada por um tutor.

Tabela A.5: Entidade cenario

ATRIBUTO	TIPO	NULO	CONCEITO
id_cenario	INT	NÃO	Chave primária (PK) para a entidade Cenário.
instalacao	INT	NÃO	Instalação para a qual o cenário foi construído. Chave estrangeira (FK) da entidade instalação.
tutor	INT	NÃO	Autor do cenário. Chave estrangeira (FK) da entidade tutor.
titulo	VARCHAR(45)	NÃO	Título do cenário.
config_antes_cenario	LONGTEXT	SIM	Descrição da configuração da instalação antes da execução do cenário (caso base).
config_depois_cenario	LONGTEXT	SIM	Descrição da configuração da instalação depois da execução do cenário.
data_criacao	DATE	SIM	Data da criação do cenário.
data_ult_edicao	DATE	SIM	Data da última edição do cenário.
descricao_cenario	LONGTEXT	SIM	Descrição do cenário como um todo.
duracao	TIME	SIM	Tempo de duração estimado para execução do cenário pelo treinando.
disj_abertos_bloqueados	LONGTEXT	SIM	Disjuntores abertos e bloqueados.
disj_abertos_nao_bloqueados	LONGTEXT	SIM	Disjuntores abertos e não bloqueados.
obj_especifico	LONGTEXT	SIM	Objetivos específicos a serem alcançados com o cenário.
obj_geral	LONGTEXT	SIM	Objetivo geral a ser alcançado com o cenário.
pre_requisitos	LONGTEXT	SIM	Pré-requisitos do treinando para executar o cenário.
protecoes_atuadas	LONGTEXT	SIM	Proteções atuadas no início da execução do cenário.

Tabela A.6: Entidade cenario

Entidade	Conceito na Ontologia	Descrição do Conceito
comparacao	Comparação entre duas variáveis, que podem ser uma medida, uma constante ou uma função	Comparações já relacionadas a um disparo condicional.

Tabela A.7: Entidade comparacao

ATRIBUTO	TIPO	NULO	CONCEITO
id_comparacao	INT	NÃO	PK para comparação e FK para operacao.
tipo_comparacao	INT	NÃO	Tipo da comparação (igual, maior etc). FK para tipo_comparacao

Tabela A.8: Entidade comparacao

Entidade	Conceito na Ontologia	Descrição do Conceito
comparacao		Constantes que podem ser usadas em comparações ou operações aritméticas e lógicas em disparos condicionais.

Tabela A.9: Entidade constante

ATRIBUTO	TIPO	NULO	CONCEITO
id_constante	INT	NÃO	PK de constante e FK de expressao
Valor	INT	NÃO	Valor que a constante assume

Tabela A.10: Entidade constante

Entidade	Conceito na Ontologia	Descrição do Conceito
comparacao		Dificuldade de uma determinada tarefa.

Tabela A.11: Entidade dificuldade

ATRIBUTO	TIPO	NULO	CONCEITO
id_dificuldade	INT	NÃO	PK de dificuldade.
dificuldade	VARCHAR(45)	NÃO	Valores de dificuldade (“Simples” ou “Complexa”).

Tabela A.12: Entidade dificuldade

Entidade	Conceito na Ontologia	Descrição do Conceito
disparo_condicional		Disparo condicional para ativar eventos.

Tabela A.13: Entidade disparo_condicional

ATRIBUTO	TIPO	NULO	CONCEITO
id_disparo_condicional	INT	NÃO	PK de disparo_condicional.
entrada	INT	NÃO	FK de entrada.
repete	BOOLEAN	NÃO	Define se o evento será ativado toda vez que a condição for satisfeita (TRUE) ou não (FALSE).

Tabela A.14: Entidade disparo_condicional

Entidade	Conceito na Ontologia	Descrição do Conceito
disparo_temporal		Disparo temporal para ativar eventos.

Tabela A.15: Entidade disparo_temporal

ATRIBUTO	TIPO	NULO	CONCEITO
id_disparo_temporal	INT	NÃO	PK de disparo_temporal.
tempo_disparo	TIME	NÃO	Tempo até que o evento seja disparado.
tipo_temporal	INT	NÃO	Tipo do disparo temporal ('Absoluto' ou 'Relativo') (FK para tipo_disparo_temporal)

Tabela A.16: Entidade disparo_temporal

Entidade	Conceito na Ontologia	Descrição do Conceito
entrada		Integra constantes, variáveis medidas e resultados de operação para que eles nunca possuam o mesmo id.

Tabela A.17: Entidade entrada

ATRIBUTO	TIPO	NULO	CONCEITO
id_entrada	INT	NÃO	PK de entrada

Tabela A.18: Entidade entrada

Entidade	Conceito na Ontologia	Descrição do Conceito
equipamento	Equipamento	Elemento constituinte do sistema elétrico.

Tabela A.19: Entidade equipamento

ATRIBUTO	TIPO	NULO	CONCEITO
id_equipamento	INT	NÃO	PK para equipamento.
ativo	BOOLEAN	NÃO	Define se o equipamento ainda existe (TRUE) ou não (FALSE). O equipamento não pode ser deletado pois poderá afetar cenários já existentes.
codigo_equipamento	VARCHAR(45)	NÃO	Código do equipamento.
instalacao	INT	NÃO	Instalação onde o equipamento está instalado (FK de instalação).
nivel_tensao	INT	NÃO	Nível de tensão do equipamento (FK de nivel_tensao).
tipo_equipamento	INT	NÃO	Tipo do equipamento (FK de tipo_equipamento).

Tabela A.20: Entidade equipamento

Entidade	Conceito na Ontologia	Descrição do Conceito
evento	Evento	Ocorrência sobre o sistema elétrico que modifica o seu estado.

Tabela A.21: Entidade evento

ATRIBUTO	TIPO	NULO	CONCEITO
id_evento	INT	NÃO	PK para evento
acao	INT	NÃO	Ação tomada quando o evento for disparado (FK de ação).
equipamento	INT	NÃO	Equipamento relacionado ao evento (FK de equipamento).
grupo_eventos	INT	NÃO	Grupo ao qual o evento pertence (FK de grupoDeEventos).
tipo_disparo	INT	NÃO	Tipo de disparo do evento (FK de tipo_disparo).
mensagem	VARCHAR(45)	SIM	Comentário que será colocado no POWERSIMULATOR quando da ocorrência da ação.
num_sequencia_evento	INT	SIM	Número de sequência.

Tabela A.22: Entidade evento

Entidade	Conceito na Ontologia	Descrição do Conceito
evento	Evento	Ocorrência sobre o sistema elétrico que modifica o seu estado.

Tabela A.23: Entidade evento

ATRIBUTO	TIPO	NULO	CONCEITO
id_evento	INT	NÃO	PK para evento
acao	INT	NÃO	Ação tomada quando o evento for disparado (FK de ação).
equipamento	INT	NÃO	Equipamento relacionado ao evento (FK de equipamento).
grupo_eventos	INT	NÃO	Grupo ao qual o evento pertence (FK de grupoDeEventos).
tipo_disparo	INT	NÃO	Tipo de disparo do evento (FK de tipo_disparo).
mensagem	VARCHAR(45)	SIM	Comentário que será colocado no POWERSIMULATOR quando da ocorrência da ação.
num_sequencia_evento	INT	SIM	Número de sequência.

Tabela A.24: Entidade evento

Entidade	Conceito na Ontologia	Descrição do Conceito
frequencia	Evento	Frequência de uma determinada tarefa

Tabela A.25: Entidade frequencia

ATRIBUTO	TIPO	NULO	CONCEITO
id_frequencia	INT	NÃO	PK de frequencia
frequencia	VARCHAR(45)	NÃO	Valores de frequencia (“Frequente” ou “Rara”)

Tabela A.26: Entidade frequencia

Entidade	Conceito na Ontologia	Descrição do Conceito
grupo_eventos	Grupo de eventos	Conjunto de eventos interrelacionados

Tabela A.27: Entidade grupo_eventos

ATRIBUTO	TIPO	NULO	CONCEITO
id_grupo_eventos	INT	NÃO	PK de grupoDeEventos.
cenario	INT	NÃO	Cenário relacionado ao grupo de eventos (FK de cenario).
descricao_grupo_eventos	VARCHAR(45)	SIM	Descrição do grupo de eventos.
num_sequencia_grupo_ever	INT	SIM	Número de sequência do grupo de eventos, considerando todos os grupos de eventos do cenário.

Tabela A.28: Entidade grupo_eventos

Entidade	Conceito na Ontologia	Descrição do Conceito
instalação	Instalação	Uma instalação de uma empresa do setor elétrico (subestação ou usina).

Tabela A.29: Entidade instalação

ATRIBUTO	TIPO	NULO	CONCEITO
id_instalacao	INT	NÃO	PK para instalação.
sigla	VARCHAR(45)	NÃO	Sigla que identifica a instalação.
nome_instalacao	VARCHAR(45)	SIM	Nome da instalação.

Tabela A.30: Entidade instalação

Entidade	Conceito na Ontologia	Descrição do Conceito
item_roteiro_execucao_cenario	Item do roteiro de execução	Ação que precisa ser executada pelo operador em treinamento durante a execução do cenário de treinamento

Tabela A.31: Entidade item_roteiro_execucao_cenario

ATRIBUTO	TIPO	NULO	CONCEITO
id_item_roteiro_execucao_cenario	INT	NÃO	PK para item_roteiro_execucao_cenario.
cenario	INT	NÃO	Cenário associado a esse item de roteiro de execução (FK para cenario).
item_execucao	VARCHAR(45)	SIM	Descrição do item do roteiro de execução do cenário.
num_sequencia_execucao	INT	SIM	Número de seqüência do item do roteiro de execução.

Tabela A.32: Entidade item_roteiro_execucao_cenario

Entidade	Conceito na Ontologia	Descrição do Conceito
item_roteiro_preparacao_cenario	Item do roteiro de preparação de um cenário	Ação que precisa ser realizada pelos responsáveis antes de iniciar um cenário de treinamento

Tabela A.33: Entidade item_roteiro_preparacao_cenario

ATRIBUTO	TIPO	NULO	CONCEITO
id_item_roteiro_preparacao	INT	NÃO	PK para itemRoteiroDePreparacao.
cenario	INT	NÃO	Cenário associado ao item de roteiro de preparação (FK para cenário).
como_atua	VARCHAR(45)	SIM	Como o Item atua.
onde_atua	VARCHAR(45)	SIM	Onde o Item atua.
quem_atua	VARCHAR(45)	SIM	Quem atua no Item.
num_sequencia_preparacao	INT	SIM	Número de sequência do item do roteiro de preparação.

Tabela A.34: Entidade item_roteiro_preparacao_cenario

Entidade	Conceito na Ontologia	Descrição do Conceito
nivel_tensao	Nível de tensão	Nível de tensão atribuído a um equipamento.

Tabela A.35: Entidade nivel_tensao

ATRIBUTO	TIPO	NULO	CONCEITO
id_nivel_tensao	INT	NAO	PK para nivelDeTensao.
nivel_tensao	VARCHAR(45)	SIM	O nível de tensão.

Tabela A.36: Entidade nivel_tensao

Entidade	Conceito na Ontologia	Descrição do Conceito
operacao_aritmetica		Operações aritméticas já relacionadas a m disparo condicional.

Tabela A.37: Entidade operacao_aritmetica

ATRIBUTO	TIPO	NULO	CONCEITO
id_oper_aritmetica	INT	NÃO	PK para operacao_aritmetica e FK para operacao
tipo_oper_aritmetica	INT	NÃO	Tipo da operação aritmética (adição, divisão etc) FK para tipo_oper_aritmetica.

Tabela A.38: Entidade operacao_aritmetica

Entidade	Conceito na Ontologia	Descrição do Conceito
operacao_binaria		Operação entre duas entradas.

Tabela A.39: Entidade operacao_binaria

ATRIBUTO	TIPO	NULO	CONCEITO
id_operacao_bin	INT	NÃO	PK para operacao e FK para entrada.
cenario	INT	NÃO	Cenario ao qual a operação está associada. FK para cenario.
código_condicao	VARCHAR(6)	NÃO	ID da condição de disparo.
delay_condicao	TIME	NÃO	Tempo após a condição ser true para disparar o evento.
entrada1	INT	NÃO	Entrada 1 da operação. FK para entrada.
entrada2	INT	NÃO	Entrada 2 da operação. FK para entrada.

Tabela A.40: Entidade operacao_binaria

Entidade	Conceito na Ontologia	Descrição do Conceito
operacao_logica		Operações lógicas já relacionadas a um disparo condicional.

Tabela A.41: Entidade operacao_logica

ATRIBUTO	TIPO	NULO	CONCEITO
id_operacao_logica	INT	NÃO	PK para operacao_logica e FK para operação.
tipo_oper_logica	INT	NÃO	Tipo da operação lógica (and, or etc). FK para tipo_oper_logica.

Tabela A.42: Entidade operacao_logica

Entidade	Conceito na Ontologia	Descrição do Conceito
prioridade		Prioridade de uma determinada tarefa

Tabela A.43: Entidade prioridade

ATRIBUTO	TIPO	NULO	CONCEITO
id_prioridade	INT	NÃO	PK de prioridade.
prioridade	VARCHAR(45)	NÃO	Valores de prioridade (“Programada” ou “Emergencial”).

Tabela A.44: Entidade prioridade

Entidade	Conceito na Ontologia	Descrição do Conceito
sinalizacao	Descrição de sinalizações	Descrição das sinalizações que se encontram atuadas no início de uma simulação.

Tabela A.45: Entidade sinalizacao

ATRIBUTO	TIPO	NULO	CONCEITO
id_sinalizacao	INT	NÃO	PK para sinalizacao.
codigo_sinalizacao	VARCHAR(45)	NÃO	ID da sinalização.
equipamento	INT	NÃO	Equipamento ao qual a sinalização está relacionada a sinalização (FK para equipamento).
mensagem_sinalizacao	VARCHAR(45)	NÃO	Mensagem passada pela sinalização.

Tabela A.46: Entidade sinalizacao

Entidade	Conceito na Ontologia	Descrição do Conceito
sinalizacao_cenario		Sinalizações que estão ligadas a um cenário.

Tabela A.47: Entidade sinalizacao_cenario

ATRIBUTO	TIPO	NULO	CONCEITO
id_sinalizacao_cenario	INT	NÃO	PK para sinalizacao.
cenario	INT	NÃO	ID do cenário (FK para cenario).
sinalizacao	INT	NÃO	ID da sinalizacao (FK para sinalizacao).
num_sequencia_sinalizacao	INT	SIM	Ordem da sequência em que a sinalização deve aparecer durante o cenário.
painel	VARCHAR(45)	SIM	Painel em que a sinalização será mostrada.

Tabela A.48: Entidade sinalizacao_cenario

Entidade	Conceito na Ontologia	Descrição do Conceito
tarefa	Tarefa	Atribuição dada ao operador em treinamento durante a execução de um cenário de treinamento

Tabela A.49: Entidade tarefa

ATRIBUTO	TIPO	NULO	CONCEITO
id_tarefa	INT	NÃO	PK para tarefa.
cenario	INT	NÃO	Cenário relacionado à tarefa (FK para cenário).
descricao_tarefa	VARCHAR(45)	SIM	Descrição da tarefa.
dificuldade	INT	SIM	Nível de dificuldade da tarefa (PK para dificuldade).
frequencia	INT	SIM	Nível de frequência da tarefa (PK para frequencia).
prioridade	INT	SIM	Nível de prioridade da tarefa (PK para prioridade).
tipo_da_tarefa	INT	SIM	Tipo da tarefa (PK para tipoDa-Tarefa).

Tabela A.50: Entidade tarefa

Entidade	Conceito na Ontologia	Descrição do Conceito
tipo_acao	Tipo de ação	Tipo de ações possíveis para ocorrer com o disparo de eventos.

Tabela A.51: Entidade tipo_acao

ATRIBUTO	TIPO	NULO	CONCEITO
id_tipo_acao	INT	NÃO	PK de tipo_acao
nome_tipo_acao	VARCHAR(45)	NÃO	Tipo de ação tomada quando o evento for disparado.
sigla_tipo_acao	VARCHAR(5)	NÃO	Sigla do tipo de ação.
tem_valor	BOOLEAN	NÃO	Se o tipo de ação tem um valor associado (True), ou não (False).

Tabela A.52: Entidade tipo_acao

Entidade	Conceito na Ontologia	Descrição do Conceito
tipo_comparacao		Tipos de comparação que podem haver para ativar um evento condicional.

Tabela A.53: Entidade tipo_comparacao

ATRIBUTO	TIPO	NULO	CONCEITO
id_tipo_comparacao	INT	NÃO	PK de tipo_comparacao
nome_tipo_comparacao	VARCHAR(45)	NÃO	Valores que os tipos de comparações podem assumir.

Tabela A.54: Entidade tipo_comparacao

Entidade	Conceito na Ontologia	Descrição do Conceito
tipo_da_tarefa		Tipo de uma determinada tarefa.

Tabela A.55: Entidade tipo_da_tarefa

ATRIBUTO	TIPO	NULO	CONCEITO
id_tipo_da_tarefa	INT	NÃO	PK de tipo_da_tarefa.
tipo_da_tarefa	VARCHAR(45)	NÃO	Valores de tipo de tarefa .

Tabela A.56: Entidade tipo_da_tarefa

Entidade	Conceito na Ontologia	Descrição do Conceito
tipo_disparo	Tipo de disparo	Tipo de disparo atribuído a um evento programado.

Tabela A.57: Entidade tipo_disparo

ATRIBUTO	TIPO	NULO	CONCEITO
id_tipo_disparo	INT	NÃO	PK para tipo_disparo

Tabela A.58: Entidade tipo_disparo

Entidade	Conceito na Ontologia	Descrição do Conceito
tipo_disparo_temporal		Tipos que os disparos temporais podem assumir.

Tabela A.59: Entidade tipo_disparo_temporal

ATRIBUTO	TIPO	NULO	CONCEITO
id_tipo_disparo_temporal	INT	NÃO	PK de id_tipo_disparo_temporal.
nome_temporal	VARCHAR(10)	NÃO	Nome dos tipos de disparo temporal ("A" ou "R").

Tabela A.60: Entidade tipo_disparo_temporal

Entidade	Conceito na Ontologia	Descrição do Conceito
tipo Equipamento	Tipo de equipamento	Categoria de equipamentos do sistema elétrico. Pode assumir os valores: chaves, reatores, transformadores, etc.

Tabela A.61: Entidade tipo Equipamento

ATRIBUTO	TIPO	NULO	CONCEITO
id_tipo Equipamento	INT	NÃO	PK para tipo Equipamento.
tipo Equipamento	VARCHAR(45)	NÃO	O nome do tipo de equipamento.
data_ultima_atualizacao	DATE	SIM	A data quando aquele tipo de equipamento foi atualizado pela última vez no BD.
sigla Equipamento	VARCHAR(3)	SIM	A sigla do tipo de equipamento.

Tabela A.62: Entidade tipo Equipamento

Entidade	Conceito na Ontologia	Descrição do Conceito
tipo Medida	Tipo de Medida	Tipo de medida referente a um equipamento do sistema elétrico > pode assumir um dos valores: tensão, corrente, potência, etc.

Tabela A.63: Entidade tipo Medida

ATRIBUTO	TIPO	NULO	CONCEITO
id_tipo Medida	INT	NÃO	PK para tipo Medida.
sub_id	VARCHAR(45)	SIM	Sub ID (se existir) da medida.
tipo Medida	VARCHAR(45)	SIM	Nome para um tipo de medida.

Tabela A.64: Entidade tipo Medida

Entidade	Conceito na Ontologia	Descrição do Conceito
tipo Medida_tipo Equipamento		Tipos de medida que um determinado tipo de equipamento pode possuir.

Tabela A.65: Entidade tipo Medida_tipo Equipamento

ATRIBUTO	TIPO	NULO	CONCEITO
id_tipo Medida_tipo Equipamento	INT	NÃO	PK para tipo Medida_tipo Equipamento.
tipo Equipamento	INT	NÃO	FK de tipo Equipamento.
tipo Medida	INT	NÃO	FK de tipo Medida.

Tabela A.66: Entidade tipo Medida_tipo Equipamento

Entidade	Conceito na Ontologia	Descrição do Conceito
tipo Oper_Aritmetica		Tipos de operações aritméticas que podem haver para ativar um evento condicional.

Tabela A.67: Entidade tipo Oper_Aritmetica

ATRIBUTO	TIPO	NULO	CONCEITO
id_tipo_aritmetica	INT	NÃO	PK de tipo_oper_aritmetica.
nome_tipo_aritmetica	VARCHAR(45)	NÃO	Valores que as operações aritméticas podem assumir.

Tabela A.68: Entidade tipo_oper_aritmetica

Entidade	Conceito na Ontologia	Descrição do Conceito
tipo_oper_logica		Tipos de operações lógicas que podem haver para ativar um evento condicional.um evento condicional.

Tabela A.69: Entidade tipo_oper_logica

ATRIBUTO	TIPO	NULO	CONCEITO
id_tipo_logica	INT	NÃO	PK de tipo_oper_logica.
nome_tipo_logica	VARCHAR(45)	NÃO	Valores que as operações lógicas podem assumir.

Tabela A.70: Entidade tipo_oper_logica

Entidade	Conceito na Ontologia	Descrição do Conceito
tutor	Tutor	Funcionário (técnico ou engenheiro) responsável pela elaboração de cenários de treinamento.

Tabela A.71: Entidade tutor

ATRIBUTO	TIPO	NULO	CONCEITO
id_tutor	INT	NÃO	Chave primária (PK) para a entidade tutor.
nome_tutor	VARCHAR(45)	NÃO	Nome do tutor.
login	VARCHAR(45)	SIM	Login para controle de acesso do tutor.
matricula	INT	SIM	Matrícula do tutor.
permissao	VARCHAR(45)	SIM	Tipo de permissão do tutor.
senha	VARCHAR(45)	SIM	Senha para controle de acesso do tutor.

Tabela A.72: Entidade tutor

Entidade	Conceito na Ontologia	Descrição do Conceito
variavel_medida		Medidas que estão associadas a um cenário.

Tabela A.73: Entidade variavel_medida

ATRIBUTO	TIPO	NULO	CONCEITO
id_variavel_medida	INT	NÃO	PK para variavel_medida e FK para expressao.
cenário	INT	NÃO	Cenário ao qual a medida está associado. FK para cenário.
codigo_medida	VARCHAR(6)	NÃO	Rótulo da medida.
equipamento	INT	NÃO	FK para equipamento.
tipo_medida	INT	NÃO	FK para tipo_medida_tipo_equipamento

Tabela A.74: Entidade variavel_medida