

# Documento de Especificação de Ontologia de Referência

**Ontologia:** Ontologia de Mapeamento Objeto Relacional (Object/Relational Mapping Ontology) ORM-O

## 1. Introdução

Este documento apresenta a ontologia ORM-O (Object/Relational Mapping Ontology) e está organizado da seguinte forma: a Seção 2 contém uma descrição do propósito da ontologia e de seus usos pretendidos; a Seção 3 define o escopo (cobertura) da ontologia e apresenta uma breve descrição do domínio para o qual se está construindo a ontologia, considerando o escopo definido; a Seção 4 apresenta os requisitos funcionais (questões de competência) e não funcionais da ontologia; a Seção 5 apresenta a ontologia de referência propriamente dita, incluindo uma apresentação da arquitetura (forma de modularização) da ontologia e, para cada subontologia considerada na arquitetura, modelos conceituais OntoUML, axiomas (informais e formais); a Seção 6 apresenta o dicionário de termos da ontologia proposta; as duas últimas seções apresentam uma avaliação preliminar da ontologia: na Seção 7 é apresentada uma verificação através da identificação de respostas às questões de competência e na Seção 8 é apresentada uma validação através da instanciação dos conceitos da ontologia em uma linguagem de programação orientada a objetos com a utilização de um framework ORM.

## 2. Descrição do Propósito e dos Usos Pretendidos da Ontologia

A ontologia ORM-O tem o propósito de identificar e definir os conceitos aplicados pelos frameworks de linguagens de programação para mapeamento entre o código orientado a objetos e o banco de dados relacional e vice-versa. Além disso, a partir dessa definição de conceitos a ontologia pretende garantir a interoperabilidade semântica entre os diferentes frameworks, uma vez que cada linguagem de programação e framework elabora sua própria sintaxe e não necessariamente mantém uma mesma semântica.

Portanto, a ontologia pretende ser usada em contextos relacionados a interoperabilidade semântica entre frameworks ORM, tal como, migração de código de um framework de determinada linguagem para um outro framework da mesma ou de uma outra linguagem.

## 3. Definição do Escopo da Ontologia

A ontologia ORM-O tem como escopo a identificação do acréscimo de código necessário para uso de frameworks ORM em linguagens OO.

## Descrição do Domínio

O cenário no desenvolvimento de software muda de maneira frenética e isso pode talvez ser justificado pela necessidade crescente de infraestrutura e arquitetura que suporte novos aparatos tecnológicos que prometem facilitar a vida das pessoas. Dentre as tecnologias surgidas nessa evolução, o paradigma orientado a objetos (OO) e o relacional sustentam-se até hoje pela eficiência que cada um apresenta.

A diferença entre os dois paradigmas é notada quando se tenta trabalhar com ambas tecnologias e, o exemplo mais comum, é a maneira preferencial de acesso às informações. Enquanto que no paradigma OO o acesso aos objetos é realizado navegando-se entre seus relacionamentos, no paradigma relacional o acesso é realizado pela junção de registros de dados em tabelas. A essa diferença é dado o nome de Impedância Objeto-Relacional.

Uma maneira de contornar a incompatibilidade entre os paradigmas é utilizar serviços de persistência fornecido por frameworks que assumam a responsabilidade de traduzir objetos em registros para sua persistência e também traduzir registros em objetos ao realizar consultas. A essa tradução de um sistema para outro é dado o nome de mapeamento objeto-relacional (object-relational mapping - ORM).

O uso desses frameworks reduz consideravelmente o tempo de desenvolvimento de um projeto por reutilizar código já desenvolvido, testado e documentado por terceiros e se tornou estado da prática no desenvolvimento de sistemas que utilizam os dois paradigmas.

Existem frameworks ORM para diversas linguagens de programação OO, como Java, Python e C++. Para a utilização dos frameworks, além de inclusão de pacotes e bibliotecas específicas, é necessário um acréscimo ou adaptação no código fonte da linguagem de programação OO a fim de indicar ao framework quais classes serão mapeadas para tabelas e quais variáveis dessas classes serão mapeadas para colunas do banco de dados.

Esses acréscimos e adaptações são diferentes em cada framework, a exemplo, o padrão JPA utiliza anotações, que são trechos de código precedidos do símbolo @. Em Python são utilizados classes e métodos de pacotes específicos. E em C++ são utilizadas diretivas pragma e templates de bibliotecas dos frameworks.

Apesar dessa diferença notável na sintaxe de utilização, é possível estabelecer padrões que se correlacionam nas funcionalidades básicas de frameworks: definição de classes e

seus relacionamentos, heranças e atributos e suas especificidades a serem mapeados quando instanciados.

#### 4. Requisitos da Ontologia

Tomando por base o propósito da ontologia e seus usos pretendidos, foram identificadas as questões de competência a serem respondidas por esta ontologia, as quais são listadas na Tabela 1.

Tabela 1 – Questões de Competência

Identificador	Questão
QC01	Que classes são mapeadas para o banco de dados?
QC02	Como os relacionamentos entre classes são mapeados para o banco de dados?
QC03	Que atributos de uma dada classe são mapeados para o banco de dados?
QC04	Que atributos de uma dada classe são mapeados para chave primária no banco de dados?
QC05	Que atributos de uma dada classe são mapeados para chave estrangeira no banco de dados?
QC06	Como os relacionamentos de herança entre classes são mapeados para o banco de dados?

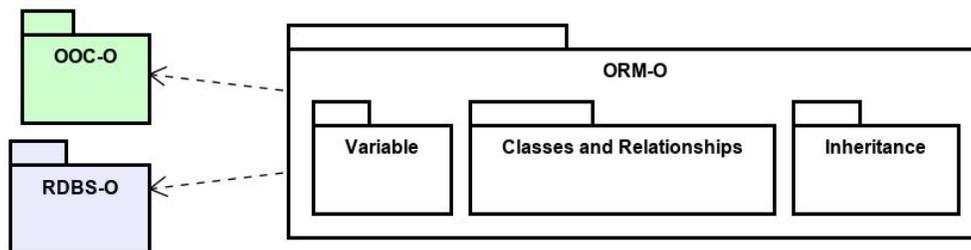
No desenvolvimento desta ontologia, devem ser considerados os requisitos não funcionais descritos na Tabela 2.

Tabela 2 – Requisitos Não Funcionais

Identificador	Sentença
RNF01	A ontologia deve considerar frameworks de diversas linguagens de programação
RNF02	A ontologia deve considerar diferentes frameworks para uma mesma linguagem de programação
RNF03	A ontologia deverá ser modularizada para permitir seu reuso

#### 5. Ontologia de Referência

Esta seção apresenta a Ontologia de Mapeamento Objeto Relacional (ORM-O), a qual está organizada nas subontologias mostradas na Figura 1 e descritas na Tabela 3.



**Figura 1 – Arquitetura da Ontologia Frameworks ORM.**

A ORM-O foi construída com base em duas ontologias preexistentes: a Ontologia de Sistema de Banco de dados Relacionais (RBDS-O), ontologia de referência que representa a estrutura do modelo de banco de dados relacional e a Ontologia de Código Orientado a Objetos (OOC-O), ontologia de referência que representa os conceitos fundamentais presentes no código-fonte orientado a objetos. Para maiores informações referentes a essas ontologias acesse: <https://nemo.inf.ufes.br/projects/sfwon/>.

A partir dessas ontologias reutilizadas, três subontologias de ORM-O foram desenvolvidas e são apresentadas na Tabela 3 juntamente com as questões de competência que respondem.

Tabela 3 – Subontologias e Questões de Competência Relacionadas

Subontologia	Descrição	QCs
<b>ORM-O Class and Relationship</b>	Subontologia de conceitos relacionados ao mapeamento de classes e relacionamentos entre classes.	QC01, QC02,
<b>ORM-O Inheritance</b>	Subontologia de conceitos relacionados ao mapeamento de herança entre classes.	QC06
<b>ORM-O Variable</b>	Subontologia de conceitos relacionados ao mapeamento de variáveis.	QC03, QC04, QC05

As subseções seguintes apresentam, para cada subontologia, seu modelo conceitual e axiomas.

### 5.1 – Subontologia ORM-O Class and Relationship

O diagrama OntoUML da Figura 2 apresenta os conceitos da ORM-O relativos ao mapeamento objeto/relacional de classes e suas possíveis relações. Dado que classes são mapeadas para tabelas, os conceitos **Class** (OOC-O) e **Table** (RDBS-O) são aqui reutilizados.

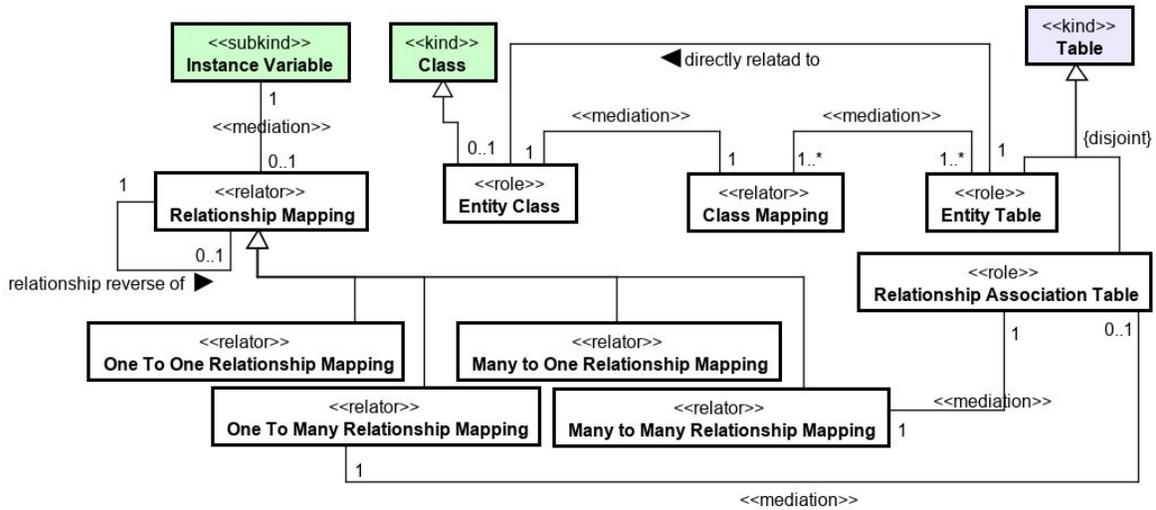


Figura 2 – Diagrama OntoUML da subontologia ORM-O Class and Relationship

### 5.2 – Subontologia ORM-O Inheritance

O diagrama OntoUML da Figura 3 apresenta os conceitos da ORM-O relativos ao mapeamento objeto/relacional da hierarquia de classes.

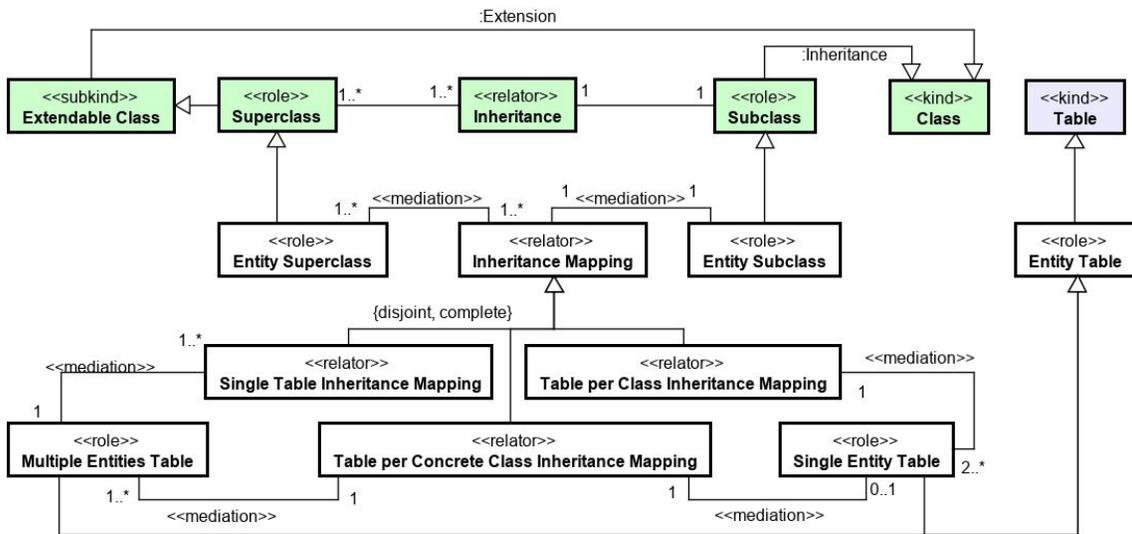


Figura 3 – Diagrama OntoUML da subontologia ORM-O Inheritance

### 5.3 – Subontologia ORM-O Variable

O diagrama OntoUML da Figura 4 apresenta os conceitos da ORM-O relativos ao mapeamento objeto/relacional de variáveis.

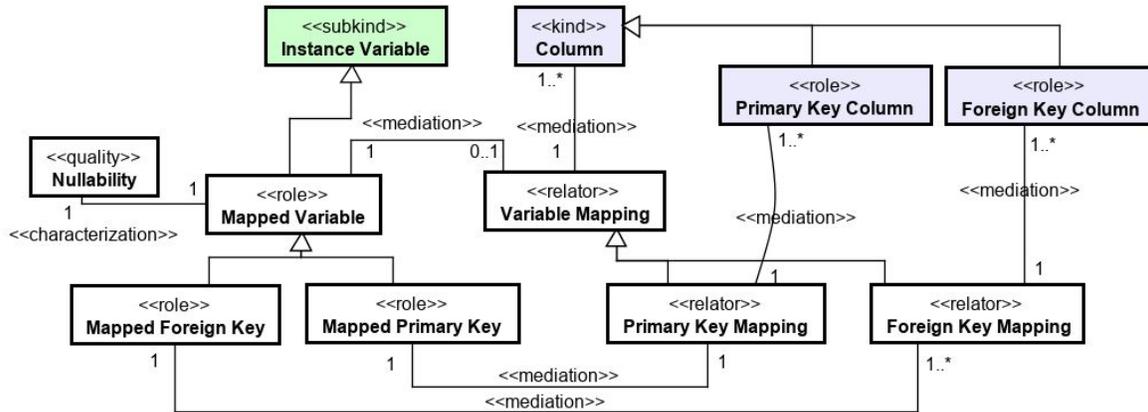


Figura 4 – Diagrama OntoUML da subontologia ORM-O Variable

## 6. Dicionário de Termos

Esta seção apresenta as definições em linguagem natural dos conceitos da ontologia ORM-O. A Tabela 4 apresenta, além das definições, as fontes a partir das quais as mesmas foram estabelecidas.

Tabela 4 – Dicionário de Termos

Conceito	Definição
<b>Entity Class</b>	Classe cujo as instâncias são objetos de domínio a serem persistidos.
<b>Class Mapping</b>	Especificação de mapeamento de uma Entity Class para uma Entity Table
<b>Entity Table</b>	Tabela do banco de dados para o qual uma Entity Class é mapeada.
<b>Relationship Mapping</b>	Mapeamento que especifica uma associação entre Entity Classes.
<b>One To One Relationship Mapping</b>	Mapeamento que especifica uma associação de valor único para outra entidade que tenha multiplicidade um para um.
<b>One To Many Relationship Mapping</b>	Mapeamento que especifica uma associação de muitos valores com multiplicidade um-para-muitos.
<b>Many to One Relationship Mapping</b>	Mapeamento que especifica uma associação de valor único para outra classe de entidade que tenha multiplicidade de muitos para um..

<b>Many To Many Relationship Mapping</b>	Mapeamento que especifica uma associação de muitos valores com multiplicidade de muitos para muitos.
<b>Relationship Association Table</b>	Tabela do banco de dados definida no mapeamento de associação de Entity Classes.
<b>Entity Superclass</b>	Uma Entity Class que também é Superclass.
<b>Entity Subclass</b>	Uma Entity Class que também é Subclass
<b>Inheritance Mapping</b>	Mapeamento de herança entre Entity Classes.
<b>Single Table Inheritance Mapping</b>	Mapeamento de herança que utiliza a estratégia Single Table, isto é, uma única tabela para a hierarquia de Entity Classes.
<b>Table per Class Inheritance Mapping</b>	Mapeamento de herança que utiliza a estratégia Table Per Class, isto é, uma tabela para cada Entity Class da hierarquia.
<b>Table per Concrete Class Inheritance Mapping</b>	Mapeamento de herança que utiliza a estratégia Table Per Concrete Class, isto é, uma tabela para cada Entity Class da hierarquia que seja concreta.
<b>Multiple Entities Table</b>	Entity Table que contém informações de mais de uma Entity Class, utilizada no mapeamento de herança do tipo Single Table Inheritance Mapping.
<b>Single Entity Table</b>	Entity Table que contém informações de uma única Entity Class ou das Entity Classes de uma única ramificação da hierarquia.
<b>Mapped Variable</b>	Instance Variable (OOO-O) que é mapeada para o banco de dados.
<b>Variable Mapping</b>	Especificação do mapeamento de uma Mapped Variable para uma coluna do banco de dados.
<b>Mapped Primary Key</b>	Mapped Variable que é uma Primary Key da Entity Class.
<b>Mapped Foreign Key</b>	Mapped Variable que é uma Foreign Key da Entity Class, ou seja, faz referência a uma Column (RDBS-O) de outra Entity Table.
<b>Primary Key Mapping</b>	Especificação do mapeamento de uma ou mais Mapped Primary Key para uma ou mais Primary Key Column (RDBS-O).
<b>Foreign Key Mapping</b>	Especificação do mapeamento de uma Mapped Foreign Key para uma Foreign Key Column (RDBS-O).

## 7. Verificação das Questões de Competência

Aqui serão apresentadas as verificações das competências da ontologia ORM-O.

Tabela 5 – Verificação da Competência da Subontologia ORM-O Class and Relationship

Questão de Competência	Conceitos, Relações e Propriedades
QC01	<b>Entity Class</b> <i>subtype of Class</i> <b>Class Mapping</b> <i>mapped by Entity Class</i> <b>Class Mapping</b> <i>mapped to Entity Table</i> <b>Entity Table</b> <i>subtype of Table</i>
QC02	<b>Instance Variable</b> <i>mediation Relationship Mapping</i> <b>One To One Relationship Mapping, One To Many Relationship Mapping, Many to One Relationship Mapping and Many to Many Relationship Mapping</b> <i>subtype of Relationship Mapping</i> <b>Entity Table and Relationship Association Table</b> <i>subtype of Table</i> <b>Many to Many Relationship Mapping</b> <i>mediation Relationship Association Table</i> <b>One to Many Relationship Mapping</b> <i>mediation Relationship Association Table</i>

Tabela 6 – Verificação da Competência da Subontologia ORM-O Inheritance

Questão de Competência	Conceitos, Relações e Propriedades
QC06	<b>Entity Subclass</b> <i>subtype of Subclass</i> <b>Entity Subclass</b> <i>mediation Inheritance Mapping</i> <b>Entity Superclass</b> <i>subtype of Superclass</i> <b>Entity Superclass</b> <i>mediation Inheritance Mapping</i> <b>Single Table Inheritance Mapping, Table per Class Inheritance Mapping and Table per Concrete Class Inheritance Mapping</b> <i>subtype of Inheritance Mapping</i> <b>Entity Table</b> <i>subtype of Table</i> <b>Multiple Entities Table and Single Entity Table</b> <i>subtype of Entity Table</i> <b>Single Table Inheritance Mapping</b> <i>mediation Multiple Entities Table</i> <b>Table per Concrete Class Inheritance Mapping</b> <i>mediation Single Entity Table</i> <b>Table per Concrete Class Inheritance Mapping</b> <i>mediation Multiple Entities Table</i> <b>Table per Class Inheritance Mapping</b> <i>mediation Single Entity Table</i>

--	--

Tabela 7 – Verificação da Competência da Subontologia ORM-O Variable

Questão de Competência	Conceitos, Relações e Propriedades
QC03	<b>Mapped Variable</b> <i>subtype of Instance Variable</i> <b>Mapped Variable</b> <i>mediation Variable Mapping</i> <b>Variable Mapping</b> <i>mediation Column</i>
QC04	<b>Mapped Variable</b> <i>subtype of Instance Variable</i> <b>Mapped Primary Key</b> <i>subtype of Mapped Variable</i> <b>Mapped Primary Key</b> <i>mediation Primary Key Mapping</i> <b>Primary Key Mapping</b> <i>mediation Primary Key Column</i>
QC05	<b>Mapped Variable</b> <i>subtype of Instance Variable</i> <b>Mapped Foreign Key</b> <i>subtype of Mapped Variable</i> <b>Mapped Foreign Key</b> <i>mediation Foreign Key Mapping</i> <b>Foreign Key Mapping</b> <i>mediation Foreign Key Column</i>

## 8. Instanciação da Ontologia

Aqui serão apresentadas instâncias dos conceitos da ontologia, os quais são usados para mostrar que a ontologia é capaz de apresentar situações de mundo real. A implementação foi realizada utilizando a linguagem Java e o framework Hibernate, que é uma implementação do JPA.

Os dados apresentados nas tabela 8, 9 e 10 são trechos do código desenvolvido para o modelo de classes da figura 5.

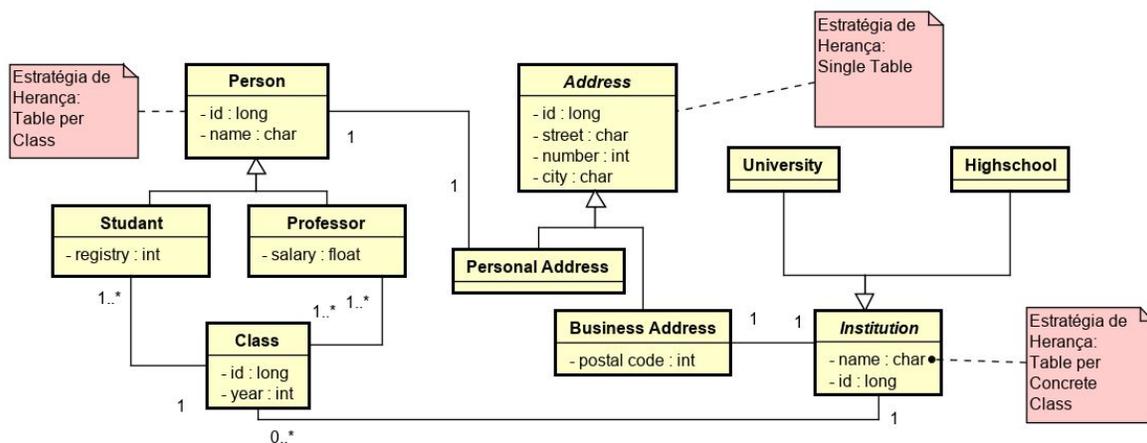
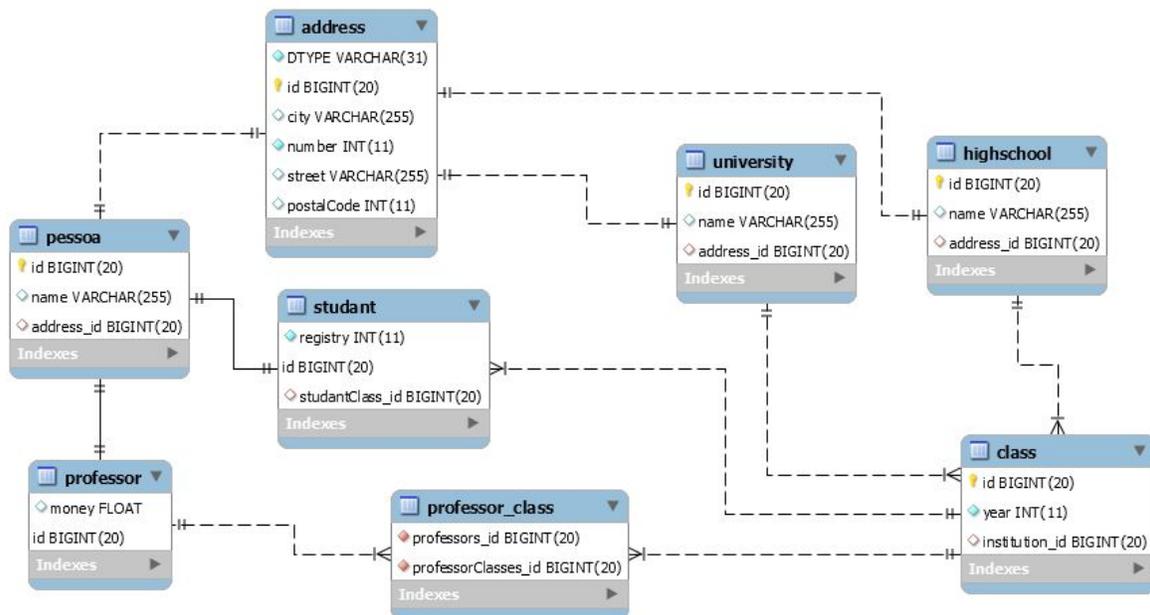


Figura 5 – Diagrama de classes do domínio instanciado

O código, após implementado, gerou uma estrutura de banco de dados equivalente à representada no modelo entidade relacionamento mostrado na figura 6.



**Figura 6 – Modelo Entidade Relacionamento do domínio instanciado**

O código completo implementado para o domínio apresentado está disponível em <https://github.com/feluzan/DomainModelExample>.

**Tabela 8 - Tabela de Instanciação da Subontologia ORM-O Class and Relationship**

Conceito	Instâncias
<b>Entity Class</b>	<code>@Entity</code> <code>public class Person{...};</code>
<b>Class Mapping</b>	<code>@Entity</code> <code>public class Person{...};</code>  A existência de uma Entity Class (Pessoa) faz com que exista implicitamente um Class Mapping.
<b>Entity Table</b>	<code>@Entity</code> <code>@Table(name="Pessoa")</code> <code>public class Person {...}</code>  Caso a anotação <code>@Table</code> seja omitida, a <b>Entity Table</b> será a tabela do banco de dados cujo o nome é idêntico ao nome da <b>Entity Class</b> .
<b>Relationship Mapping</b>	As instâncias de Relationship Mapping são representadas por suas especializações
<b>One To One Relationship Mapping</b>	<code>@Entity</code> <code>public class Person {</code> <code>...</code>

	<pre> @OneToOne private PersonalAddress address;  }  @Entity public abstract class Address{     ...     @Id     private Long id;     ... } </pre>
<b>One To Many Relationship Mapping</b>	<pre> @Entity public class Class {     ...     @OneToMany(mappedBy="studentClass")     private List&lt;Student&gt; students;     ... } </pre>
<b>Many To One Relationship Mapping</b>	<pre> @Entity public class Student extends Person {     ...     @ManyToOne     private Class studentClass;     ... } </pre>
<b>Many To Many Relationship Mapping</b>	<pre> @Entity public class Class {     ...     @ManyToMany(mappedBy="professorClasses")     private List&lt;Professor&gt; professors;     ... }  @Entity public class Professor extends Person {     ...     @ManyToMany     private List&lt;Class&gt; professorClasses;     ... } </pre>
<b>Relationship Association Table</b>	<pre> ... @ManyToMany @JoinTable(name="tabela_professor_classes") private List&lt;Class&gt; professorClasses; </pre>

	...
--	-----

Tabela 9 – Tabela de Instanciação da Subontologia ORM-O Inheritance

Conceito	Instâncias
<b>Entity Subclass</b>	<code>@Entity public class Student extends Person{...};</code>
<b>Entity Superclass</b>	<code>@Entity public class Student extends Person{...};</code>  A existência de uma <b>Entity Subclass</b> (Estudante) faz com que a classe no qual ela estende seja um <b>Entity Superclass</b> (Pessoa)
<b>Inheritance Mapping</b>	As instâncias de <b>Inheritance Mapping</b> são representadas pelas instâncias de suas especializações.
<b>Single Table Inheritance Mapping</b>	<code>@Entity @Inheritance(strategy = InheritanceType.SINGLE_TABLE) public abstract class Address {...}</code>
<b>Table per Concrete Class Inheritance Mapping</b>	<code>@@Entity @Inheritance(strategy = InheritanceType.TABLE_PER_CLASS) public abstract class Institution {...}</code>
<b>Table per Class Inheritance Mapping</b>	<code>@Entity @Inheritance(strategy = InheritanceType.JOINED) public class Person {...}</code>

Tabela 10 – Tabela de Instanciação da Subontologia ORM-O Variable

Conceito	Instâncias
<b>Mapped Variable</b>	<code>@Entity public class Professor extends Person {     ...     @Column(name="money")     private float salary;     ... }</code>  Qualquer atributo de uma <b>Entity Class</b> que não seja procedido da anotação <code>@Transient</code>
<b>Variable Mapping</b>	<code>@Entity public class Professor extends Person {     ...     @Column(name="money")     private float salary;</code>

	<pre>... }</pre> <p>A existência de uma <b>Mapped Variable</b> implica na existência de uma <b>Variable Mapping</b>. Caso a anotação <code>@Column</code> seja omitida, a <b>Variable Mapping</b> será associada à <b>Column</b> com nome idêntico ao nome da <b>Mapped Variable</b>.</p>
<b>Mapped Primary Key</b>	<pre>... @Id private Long id; ...</pre>
<b>Mapped Foreign Key</b>	<pre>... @OneToOne private PersonalAddress address; ...</pre>
<b>Primary Key Mapping</b>	<pre>... @Id private Long id; ...</pre> <p>A existência de uma <b>Mapped Primary Key</b> gera implicitamente na existência de uma <b>Primary Key Mapping</b>.</p>
<b>Foreign Key Mapping</b>	<pre>... @OneToOne private PersonalAddress address; ...</pre> <p>A existência de uma <b>Mapped ForeignKey</b> gera implicitamente na existência de uma <b>Foreign Key Mapping</b></p>