**Plataforma de Cursos Online**

**App Web UserConnectSTS** — conectando pessoas e conhecimento

**Entidades**

1. **User** – representa os usuários (**alunos** ou **instrutores**).
2. **Course** – representa os cursos disponíveis.
3. **Enrollment(Matrícula)** – entidade associativa que liga o User ao Course.

**Modelo de Domínio (conceitual)**

**User (1) ⟷ (N) Enrollment (N) ⟷ (1) Course**

**Entidades (Java)**

**User.java**

package com.example.domain;

import jakarta.persistence.\*;

import java.util.HashSet;

import java.util.Set;

**@Entity**

**@Table(name = "tb\_user")**

public class User {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String name;

private String email;

@OneToMany(mappedBy = "user", cascade = CascadeType.ALL, orphanRemoval = true)

private Set<Enrollment> enrollments = new HashSet<>();

// Getters, setters e construtores

public User() {}

public User(String name, String email) {

this.name = name;

this.email = email;

}

// Getters e Setters

public Long getId() { return id; }

public String getName() { return name; }

public void setName(String name) { this.name = name; }

public String getEmail() { return email; }

public void setEmail(String email) { this.email = email; }

public Set<Enrollment> getEnrollments() { return enrollments; }

}

**Course.java**

package com.example.domain;

import jakarta.persistence.\*;

import java.util.HashSet;

import java.util.Set;

**@Entity**

**@Table(name = "tb\_course")**

public class Course {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

private String title;

private String category;

@OneToMany(mappedBy = "course", cascade = CascadeType.ALL, orphanRemoval = true)

private Set<Enrollment> enrollments = new HashSet<>();

public Course() {}

public Course(String title, String category) {

this.title = title;

this.category = category;

}

// Getters e Setters

public Long getId() { return id; }

public String getTitle() { return title; }

public void setTitle(String title) { this.title = title; }

public String getCategory() { return category; }

public void setCategory(String category) { this.category = category; }

public Set<Enrollment> getEnrollments() { return enrollments; }}

**Enrollment.java (entidade associativa)**

package com.example.domain;

import jakarta.persistence.\*;

import java.time.LocalDate;

**@Entity**

**@Table(name = "tb\_enrollment")**

public class Enrollment {

@EmbeddedId

private EnrollmentId id = new EnrollmentId();

@ManyToOne

@MapsId("userId")

@JoinColumn(name = "user\_id")

private User user;

@ManyToOne

@MapsId("courseId")

@JoinColumn(name = "course\_id")

private Course course;

private LocalDate enrollmentDate;

public Enrollment() {}

public Enrollment(User user, Course course) {

this.user = user;

this.course = course;

this.enrollmentDate = LocalDate.now();

this.id = new EnrollmentId(user.getId(), course.getId());

}

// Getters e Setters

public EnrollmentId getId() { return id; }

public User getUser() { return user; }

public Course getCourse() { return course; }

public LocalDate getEnrollmentDate() { return enrollmentDate; }

public void setEnrollmentDate(LocalDate enrollmentDate) { this.enrollmentDate = enrollmentDate; }

}

**EnrollmentId.java (chave composta)**

package com.example.domain;

import jakarta.persistence.Embeddable;

import java.io.Serializable;

import java.util.Objects;

@Embeddable

public class EnrollmentId **implements** Serializable {

private Long userId;

private Long courseId;

public EnrollmentId() {}

public EnrollmentId(Long userId, Long courseId) {

this.userId = userId;

this.courseId = courseId;

}

// Getters, Setters, equals e hashCode

public Long getUserId() { return userId; }

public void setUserId(Long userId) { this.userId = userId; }

public Long getCourseId() { return courseId; }

public void setCourseId(Long courseId) { this.courseId = courseId; }

@Override

public boolean equals(Object o) {

if (this == o) return true;

if (!(o instanceof EnrollmentId)) return false;

EnrollmentId that = (EnrollmentId) o;

return Objects.equals(userId, that.userId) &&

Objects.equals(courseId, that.courseId);

}

@Override

public int hashCode() {

return Objects.hash(userId, courseId);

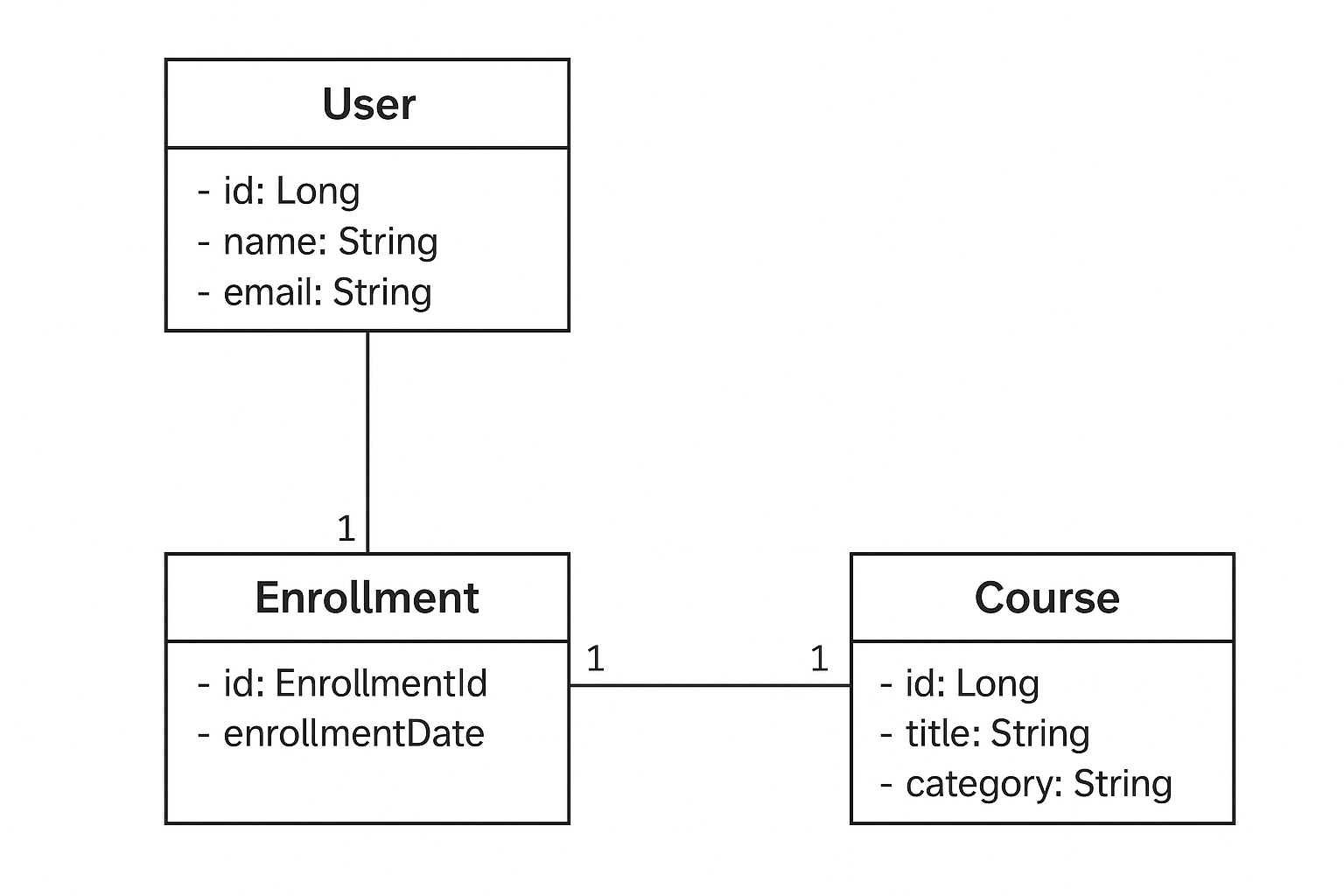
}

}

**Resultado**

* **User** → representa o aluno/instrutor.
* **Course** → representa um curso.
* **Enrollment** → representa a **relação associativa** entre User e Course, armazenando também dados adicionais como a data de matrícula.

**Diagrama UML — Domínio de Cursos**



**Relações**:

* **User** (1) ↔ (N) **Enrollment**
* **Course** (1) ↔ (N) **Enrollment**
* **Enrollment** tem PK **composta** (**user\_id, course\_id**)

**SQL DDL Gerado**

**Tabela principal: USER**

CREATE TABLE **tb\_user** (

id BIGINT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

nome VARCHAR(100) NOT NULL,

idade INT NOT NULL

);

**Tabela principal: COURSE**

CREATE TABLE **tb\_course** (

id BIGINT GENERATED BY DEFAULT AS IDENTITY PRIMARY KEY,

title VARCHAR(150) NOT NULL,

category VARCHAR(100)

);

**Tabela associativa: ENROLLMENT**

CREATE TABLE **tb\_enrollment** (

user\_id BIGINT NOT NULL,

course\_id BIGINT NOT NULL,

enrollment\_date DATE,

PRIMARY KEY (user\_id, course\_id),

CONSTRAINT fk\_enrollment\_user

FOREIGN KEY (user\_id)

REFERENCES users (id)

ON DELETE CASCADE,

CONSTRAINT fk\_enrollment\_course

FOREIGN KEY (course\_id)

REFERENCES courses (id)

ON DELETE CASCADE

);