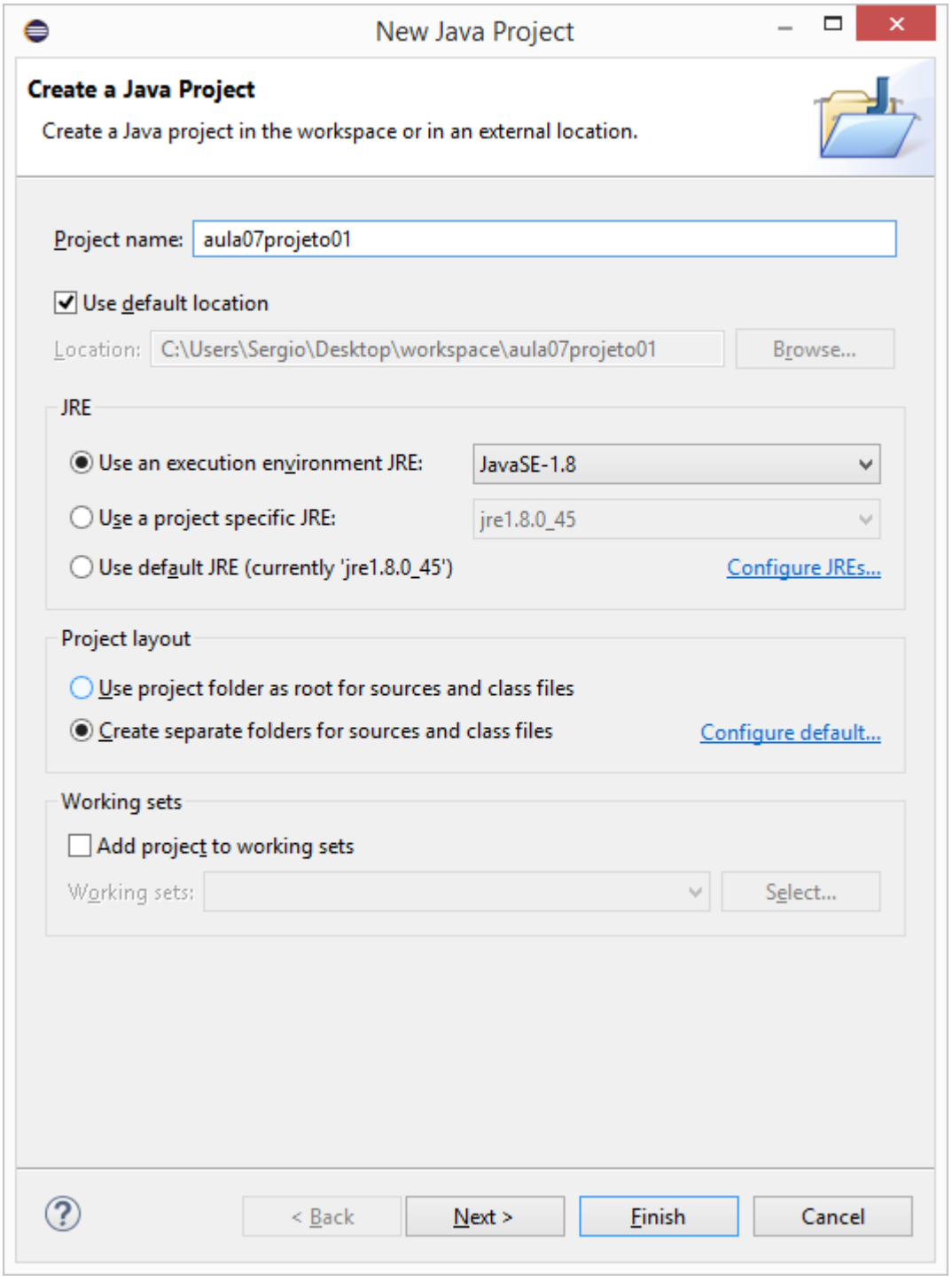
Novo projeto:

File > New > Java Project



Criando o Script do banco de dados
Relacionamento Muitos para Muitos

```
drop database if exists aula07;
create database aula07;
use aula07;

create table aluno(
	idaluno		integer		auto_increment,
	nome		varchar(50)	not null,
	email		varchar(50)	not null unique,
	primary key(idaluno));

create table turma(
	idturma		integer		auto_increment,
	nome		varchar(50)	not null,
	periodo		enum('Manha', 'Tarde', 'Noite') not null,
	primary key(idturma));

desc aluno;
desc turma;

create table turma_aluno(
	idturma		integer		not null,
	idaluno		integer		not null,
	primary key(idturma, idaluno),
	foreign key(idturma) references turma(idturma),
	foreign key(idaluno) references aluno(idaluno));

desc turma_aluno;

insert into aluno(nome, email) values('Diego', 'diego@gmail.com');
insert into aluno(nome, email) values('Ingrid', 'ingrid@gmail.com');
select * from aluno;

insert into turma(nome, periodo) values('Java', 'Manha');
insert into turma(nome, periodo) values('C#', 'Tarde');
insert into turma(nome, periodo) values('Oracle', 'Noite');
insert into turma(nome, periodo) values('PHP', 'Manha');
select * from turma;

insert into turma_aluno(idturma, idaluno) values(1, 1);
insert into turma_aluno(idturma, idaluno) values(1, 2);
insert into turma_aluno(idturma, idaluno) values(2, 2);
```

```
insert into turma_aluno(idturma, idaluno) values(2, 1);
insert into turma_aluno(idturma, idaluno) values(3, 1);
insert into turma_aluno(idturma, idaluno) values(4, 2);
select * from turma_aluno;

--consulta de turmas com alunos
--select de turma junção com turma_aluno e junção com aluno
--exibe os dados de turmas e alunos (muitos para muitos)
select
	t.idturma,
	t.nome as nometurma,
	t.periodo,
	a.idaluno,
	a.nome as nomealuno,
	a.email
from turma as t
inner join turma_aluno as ta
on t.idturma = ta.idturma
inner join aluno as a
on a.idaluno = ta.idaluno
order by idturma;

--view (visão)
create view vw_turmasalunos
as
select
	t.idturma,
	t.nome as nometurma,
	t.periodo,
	a.idaluno,
	a.nome as nomealuno,
	a.email
from turma as t
inner join turma_aluno as ta
on t.idturma = ta.idturma
inner join aluno as a
on a.idaluno = ta.idaluno
order by idturma;

--consultar os dados da view..
select * from vw_turmasalunos;

--consulta para trazer a quantidade de alunos por turma..
select
```

```
        t.idturma,
        t.nome as nometurma,
        t.periodo,
        count(ta.idaluno) as qtd_alunos
from turma as t
left join turma_aluno as ta
on t.idturma = ta.idturma
group by idturma, nometurma, periodo;


--criando uma view para a consulta acima..
create view vw_turmaqtdalunos
as
select
        t.idturma,
        t.nome as nometurma,
        t.periodo,
        count(ta.idaluno) as qtd_alunos
from turma as t
left join turma_aluno as ta
on t.idturma = ta.idturma
group by idturma, nometurma, periodo;

desc vw_turmaqtdalunos;

select * from vw_turmaqtdalunos;


--criando uma view para obter alunos e quantidade de turmas
create view vw_alunoqtdturmas
as
select
        a.idaluno,
        a.nome as nomealuno,
        a.email,
        count(ta.idturma) as qtd_turmas
from aluno as a
left join turma_aluno as ta
on a.idaluno = ta.idaluno
group by idaluno, nome, email;

desc vw_alunoqtdturmas;

select * from vw_alunoqtdturmas;
```

Criando as classes de entidade:

```
package entities.types;

public enum Periodo {
	Manha,
	Tarde,
	Noite
}
```

```
package entities;

import java.util.Collection;

public class Aluno {

	private Integer idAluno;
	private String nome;
	private String email;

	private Collection<Turma> turmas;

	public Aluno() {
		// TODO Auto-generated constructor stub
	}

	public Aluno(Integer idAluno, String nome, String email) {
		this.idAluno = idAluno;
		this.nome = nome;
		this.email = email;
	}

	public Aluno(Integer idAluno, String nome, String email,
			Collection<Turma> turmas) {
		this(idAluno, nome, email);
		this.turmas = turmas;
	}

	public Integer getIdAluno() {
		return idAluno;
	}

	public void setIdAluno(Integer idAluno) {
		this.idAluno = idAluno;
	}

	public String getNome() {
		return nome;
	}
	public void setNome(String nome) {
```

```
		this.nome = nome;
	}

	public String getEmail() {
		return email;
	}

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

	public Collection<Turma> getTurmas() {
		return turmas;
	}

	public void setTurmas(Collection<Turma> turmas) {
		this.turmas = turmas;
	}

	@Override
	public String toString() {
		return "Aluno [idAluno=" + idAluno + ", nome="
				+ nome + ", email=" + email + "]";
	}
}

package entities;

import java.util.Collection;

import entities.types.Periodo;

public class Turma {

	private Integer idTurma;
	private String nome;
	private Periodo periodo;

	private Collection<Aluno> alunos;

	public Turma() {
		// TODO Auto-generated constructor stub
	}

	public Turma(Integer idTurma, String nome, Periodo periodo)
	{
		this.idTurma = idTurma;
		this.nome = nome;
		this.periodo = periodo;
	}
	public Turma(Integer idTurma, String nome, Periodo periodo,
```

```
            Collection<Aluno> alunos) {
            this(idTurma, nome, periodo);
            this.alunos = alunos;
      }

      public Integer getIdTurma() {
            return idTurma;
      }

      public void setIdTurma(Integer idTurma) {
            this.idTurma = idTurma;
      }

      public String getNome() {
            return nome;
      }

      public void setNome(String nome) {
            this.nome = nome;
      }

      public Periodo getPeriodo() {
            return periodo;
      }

      public void setPeriodo(Periodo periodo) {
            this.periodo = periodo;
      }

      public Collection<Aluno> getAlunos() {
            return alunos;
      }

      public void setAlunos(Collection<Aluno> alunos) {
            this.alunos = alunos;
      }

      @Override
      public String toString() {
            return "Turma [idTurma=" + idTurma + ", nome="
                  + nome + ", periodo=" + periodo + "]";
      }
}
```

Criando a Classe Dao
Data Access Object

```
package persistence;


import java.sql.CallableStatement;
```

```
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.PreparedStatement;
import java.sql.ResultSet;

public class Dao {

	private static final String DRIVER = "com.mysql.jdbc.Driver";
	private static final String URL = "jdbc:mysql://localhost:3306/aula07";
	private static final String USER = "root";
	private static final String PASSWORD = "brqbrq";

	protected Connection con;
	protected PreparedStatement stmt;
	protected CallableStatement call;
	protected ResultSet rs;

	protected void openConnection() throws Exception{
		Class.forName(DRIVER);
		con = DriverManager.getConnection(URL, USER, PASSWORD);
	}

	protected void closeConnection() throws Exception{
		if(con != null){
			con.close();
		}
	}
}
```

DTO – Data Transfer Object

Objetos para transferência de dados

Objeto de Transferência de Dados (do inglês, **Data transfer object**, ou simplesmente **DTO**),é um padrão de projeto de software usado para transferir dados entre subsistemas de um software. DTOs são frequentemente usados em conjunção com objetos de acesso a dados para obter dados de um banco de dados.

A diferença entre objetos de transferência de dados e objetos de negócio ou objetos de acesso a dados é que um DTO não possui comportamento algum, exceto o de armazenamento e obtenção de seus próprios dados. DTOs são objetos simples que não contêm qualquer lógica de negócio que requeira testes.

Criando a classe de persistência para Aluno

```
package persistence;

import entities.Aluno;

public class AlunoDao extends Dao{

	//método para gravar um aluno na base de dados..
	public void insert(Aluno a) throws Exception{

		String query = "insert into aluno
				(nome, email) values(?,?)";

		openConnection();
		stmt = con.prepareStatement(query);
		stmt.setString(1, a.getNome());
		stmt.setString(2, a.getEmail());
		stmt.execute();
		stmt.close();
		closeConnection();
	}

	public static void main(String[] args) {
		try{

			Aluno a = new Aluno
			(null, "Sergio Mendes", "sergio@gmail.com");

			AlunoDao d = new AlunoDao();
			d.insert(a); //gravando..

			System.out.println("Dados gravados.");
		}
		catch(Exception e){
			System.out.println("Erro: " + e.getMessage());
		}
	}
}
```

Criando a classe de persistência para a entidade Turma:

```
package persistence;

import entities.Turma;
import entities.types.Periodo;

public class TurmaDao extends Dao{

	//método para gravar um turma na base de dados..
	public void insert(Turma t) throws Exception{

		String query = "insert into turma(nome, periodo) values(?,?)";

		openConnection();
		stmt = con.prepareStatement(query);
		stmt.setString(1, t.getNome());
		stmt.setString(2, t.getPeriodo().name());
		stmt.execute();
		stmt.close();
		closeConnection();
	}

	public static void main(String[] args) {
		try{

			Turma t = new Turma(null, "PostGreSQL",
					Periodo.Manha);

			TurmaDao d = new TurmaDao();
			d.insert(t); //gravando..

			System.out.println("Dados gravados.");
		}
		catch(Exception e){
			System.out.println("Erro: " + e.getMessage());
		}
	}
}
```

Criando DTOs para consultas baseadas em views:

```java
package dto;

//DTO - Data Transfer Object
public class TurmaAlunoDto {

	// atributos da consulta..
	private Integer idTurma;
	private String nomeTurma;
	private String periodo;
	private Integer idAluno;
	private String nomeAluno;
	private String email;

	public Integer getIdTurma() {
		return idTurma;
	}

	public void setIdTurma(Integer idTurma) {
		this.idTurma = idTurma;
	}

	public String getNomeTurma() {
		return nomeTurma;
	}

	public void setNomeTurma(String nomeTurma) {
		this.nomeTurma = nomeTurma;
	}

	public String getPeriodo() {
		return periodo;
	}

	public void setPeriodo(String periodo) {
		this.periodo = periodo;
	}

	public Integer getIdAluno() {
		return idAluno;
	}

	public void setIdAluno(Integer idAluno) {
		this.idAluno = idAluno;
	}
```

```
	public String getNomeAluno() {
		return nomeAluno;
	}

	public void setNomeAluno(String nomeAluno) {
		this.nomeAluno = nomeAluno;
	}

	public String getEmail() {
		return email;
	}

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

	@Override
	public String toString() {
		return "TurmaAlunoDto [idTurma=" + idTurma + ",
		nomeTurma=" + nomeTurma + ", periodo="
		 + periodo + ", idAluno="
		 + idAluno + ", nomeAluno=" + nomeAluno
		 + ", email=" + email + "]";
	}
}


package dto;

public class AlunoQtdTurmasDto {

	private Integer idAluno;
	private String nomeAluno;
	private String email;
	private Integer qtdTurmas;

	public Integer getIdAluno() {
		return idAluno;
	}

	public void setIdAluno(Integer idAluno) {
		this.idAluno = idAluno;
	}


	public String getNomeAluno() {
```

```
		return nomeAluno;
	}

	public void setNomeAluno(String nomeAluno) {
		this.nomeAluno = nomeAluno;
	}

	public String getEmail() {
		return email;
	}

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

	public Integer getQtdTurmas() {
		return qtdTurmas;
	}

	public void setQtdTurmas(Integer qtdTurmas) {
		this.qtdTurmas = qtdTurmas;
	}

	@Override
	public String toString() {
		return "AlunoQtdTurmasDto [idAluno=" + idAluno + ",
			nomeAluno=" + nomeAluno + ", email=" + email
			+ ", qtdTurmas=" + qtdTurmas + "]";
	}
}


package dto;

public class AlunoQtdTurmasDto {

	private Integer idAluno;
	private String nomeAluno;
	private String email;
	private Integer qtdTurmas;

	public Integer getIdAluno() {
		return idAluno;
	}
```

```
	public void setIdAluno(Integer idAluno) {
		this.idAluno = idAluno;
	}

	public String getNomeAluno() {
		return nomeAluno;
	}

	public void setNomeAluno(String nomeAluno) {
		this.nomeAluno = nomeAluno;
	}

	public String getEmail() {
		return email;
	}

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

	public Integer getQtdTurmas() {
		return qtdTurmas;
	}

	public void setQtdTurmas(Integer qtdTurmas) {
		this.qtdTurmas = qtdTurmas;
	}

	@Override
	public String toString() {
		return "AlunoQtdTurmasDto [idAluno=" + idAluno
		 + ", nomeAluno=" + nomeAluno + ", email=" + email
		+ ", qtdTurmas=" + qtdTurmas + "]";
	}
}
```

Criando classe de persistência para operações com Turma e Aluno:

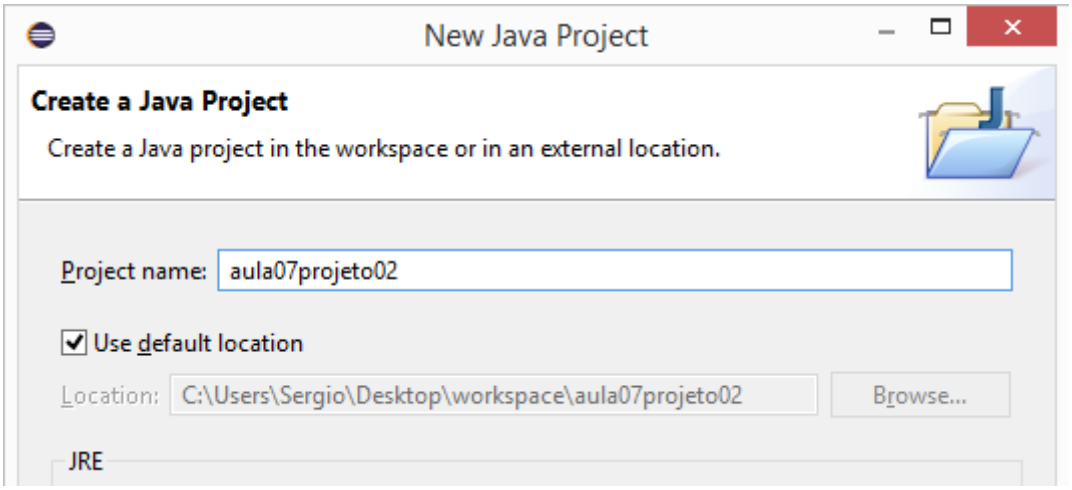
```
package persistence;

import java.util.ArrayList;
import java.util.List;

import dto.AlunoQtdTurmasDto;
import dto.TurmaAlunoDto;
```

```
import dto.TurmaQtdAlunosDto;

public class TurmaAlunoDao extends Dao {

	// gravar um registro na entidade associativa..
	public void insert(Integer idTurma, Integer idAluno) throws Exception {

		String query = "insert into turma_aluno(idturma, idaluno) values(?, ?)";

		openConnection();
		stmt = con.prepareStatement(query);
		stmt.setInt(1, idTurma);
		stmt.setInt(2, idAluno);
		stmt.execute();
		stmt.close();
		closeConnection();
	}

	// método para consultar e retornar os dados da view..
	public List<TurmaAlunoDto> findAll() throws Exception {

		String query = "select * from vw_turmasalunos";

		openConnection();
		stmt = con.prepareStatement(query);
		rs = stmt.executeQuery();

		List<TurmaAlunoDto> lista = new ArrayList<TurmaAlunoDto>();

		while (rs.next()) { // enquanto houver registros..
			TurmaAlunoDto dto = new TurmaAlunoDto();

			dto.setIdTurma(rs.getInt("idturma"));
			dto.setNomeTurma(rs.getString("nometurma"));
			dto.setPeriodo(rs.getString("periodo"));
			dto.setIdAluno(rs.getInt("idaluno"));
			dto.setNomeAluno(rs.getString("nomealuno"));
			dto.setEmail(rs.getString("email"));

			lista.add(dto); // adicionar na lista..
		}

		stmt.close();
		closeConnection();

		return lista; // retornando a lista..
	}
```

```
// método para obter a quantidade de alunos por turma..
public List<TurmaQtdAlunosDto> findAllQtdAlunos() throws Exception {

	String query = "select * from vw_turmaqtdalunos";

	openConnection();

	stmt = con.prepareStatement(query);
	rs = stmt.executeQuery();

	List<TurmaQtdAlunosDto> lista = new ArrayList<TurmaQtdAlunosDto>();

	while (rs.next()) { // enquanto houver registros na consulta..
		TurmaQtdAlunosDto dto = new TurmaQtdAlunosDto();
		dto.setIdTurma(rs.getInt("idturma"));
		dto.setNomeTurma(rs.getString("nometurma"));
		dto.setPeriodo(rs.getString("periodo"));
		dto.setQtdAlunos(rs.getInt("qtd_alunos"));

		lista.add(dto); // adicionar na lista..
	}

	stmt.close();
	closeConnection();

	return lista; // retornando a lista..
}

// método para obter a quantidade de turmas por aluno..
public List<AlunoQtdTurmasDto> findAllQtdTurmas() throws Exception {

	String query = "select * from vw_alunoqtdturmas";

	openConnection();

	stmt = con.prepareStatement(query);
	rs = stmt.executeQuery();

	List<AlunoQtdTurmasDto> lista = new ArrayList<AlunoQtdTurmasDto>();

	while (rs.next()) { // enquanto houver registros na consulta..
		AlunoQtdTurmasDto dto = new AlunoQtdTurmasDto();
		dto.setIdAluno(rs.getInt("idaluno"));
		dto.setNomeAluno(rs.getString("nomealuno"));
		dto.setEmail(rs.getString("email"));
		dto.setQtdTurmas(rs.getInt("qtd_turmas"));

		lista.add(dto); // adicionar na lista..
```

```
                }

                stmt.close();
                closeConnection();

                return lista; // retornando a lista..
        }

        public static void main(String[] args) {

                try {
                        TurmaAlunoDao d = new TurmaAlunoDao();
                        // d.insert(5, 1); //gravando..
                        // System.out.println("Dados gravados");

                        // exibir a consulta...
                        // for(TurmaAlunoDto dto : d.findAll()){
                        // System.out.println(dto); //toString()
                        // }

                        for (AlunoQtdTurmasDto dto : d.findAllQtdTurmas()) {
                                System.out.println(dto); // toString
                        }

                } catch (Exception e) {
                        System.out.println("Erro:" + e.getMessage());
                }
        }
}
```

Novo projeto:



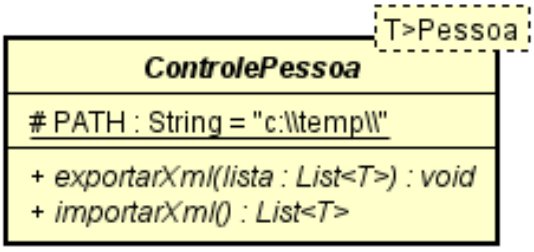
Criando uma modelagem de entidades baseado em Herança:

```
package entities;

public abstract class Pessoa {

	private Integer idPessoa;
	private String nome;

	public Pessoa() {
		// TODO Auto-generated constructor stub
	}

	public Pessoa(Integer idPessoa, String nome) {
		this.idPessoa = idPessoa;
		this.nome = nome;
	}

	public Integer getIdPessoa() {
		return idPessoa;
	}

	public void setIdPessoa(Integer idPessoa) {
		this.idPessoa = idPessoa;
	}

	public String getNome() {
		return nome;
	}

	public void setNome(String nome) {
		this.nome = nome;
	}

	@Override
	public String toString() {
		return idPessoa + ", " + nome;
	}
}


package entities;

public class Cliente extends Pessoa{
```

```
//atributos..
private String cpf;
private String email;
private String telefone;

public Cliente() {
	// TODO Auto-generated constructor stub
}

public Cliente(Integer idPessoa, String nome, String cpf,
	String email, String telefone) {
	super(idPessoa, nome);
	this.cpf = cpf;
	this.email = email;
	this.telefone = telefone;
}

public String getCpf() {
	return cpf;
}

public void setCpf(String cpf) {
	this.cpf = cpf;
}

public String getEmail() {
	return email;
}

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

public String getTelefone() {
	return telefone;
}

public void setTelefone(String telefone) {
	this.telefone = telefone;
}

@Override
public String toString() {
	return super.toString() + ", " + cpf + ", "
		+ email + ", " + telefone;
}
```

```
}


package entities;

public class Funcionario extends Pessoa{

	private Double salario;
	private String funcao;

	public Funcionario() {
		// TODO Auto-generated constructor stub
	}

	public Funcionario(Integer idPessoa, String nome, Double
		salario, String funcao) {
		super(idPessoa, nome);
		this.salario = salario;
		this.funcao = funcao;
	}

	public Double getSalario() {
		return salario;
	}

	public void setSalario(Double salario) {
		this.salario = salario;
	}

	public String getFuncao() {
		return funcao;
	}

	public void setFuncao(String funcao) {
		this.funcao = funcao;
	}

	@Override
	public String toString() {
		return super.toString() + ", " + salario + ", "
			+ funcao;
	}
}
```

Criando classes de controle:




```
package control;

import java.util.List;

import entities.Pessoa;

//<T> tipo generico que devera ser uma herança da classe Pessoa..
public abstract class ControlePessoa<T extends Pessoa> {

	/*
	 * Classes abstratas podem ter atributos, construtores e metodos
	 * tal qual uma classe concreta, mas tambem podem ter metodos abstratos
	 * similar aos de uma interface..
	 */

	//constante..
	protected static final String PATH = "c:\\temp\\";

	//método para receber uma lista e gravar os dados em XML..
	public abstract void exportarXml(List<T> lista) throws Exception;

	//método para retornar uma lista de objetos de um arquivo XML..
	public abstract List<T> importarXml() throws Exception;

}
```

Gerando e lendo arquivos XML:

```
package control;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.ArrayList;
```

```
import java.util.List;

import com.thoughtworks.xstream.XStream;
import com.thoughtworks.xstream.io.xml.Dom4JDriver;
import com.thoughtworks.xstream.io.xml.StaxDriver;

import entities.Cliente;

public class ControleCliente extends ControlePessoa<Cliente> {

	@Override
	public void exportarXml(List<Cliente> lista) throws Exception {

		//instanciando a biblioteca para geração XML..
		XStream xstream = new XStream(new StaxDriver());

		//definir o nomeda <tag> raiz do xml
		xstream.alias("dados", List.class); //conter uma listagem..
		xstream.alias("cliente", Cliente.class); //conter objetos de Cliente..

		//gravar o arquivo xml..
		FileWriter arquivo = new FileWriter(new File(PATH + "clientes.xml"));
		arquivo.write(xstream.toXML(lista));
		//gerando um XML da lista de clientes..
		arquivo.close(); //fechar o arquivo..
	}

	@Override
	public List<Cliente> importarXml() throws Exception {

		//declarando uma lista de clientes...
		List<Cliente> lista = new ArrayList<Cliente>();

		XStream xstream = new XStream(new Dom4JDriver());
		xstream.alias("dados", List.class);
		xstream.alias("cliente", Cliente.class);

		BufferedReader arquivo = new BufferedReader
			(new FileReader(PATH + "clientes.xml"));
		lista = (List<Cliente>) xstream.fromXML(arquivo);
		arquivo.close();


		return lista; //retornando a lista..
```

```
	}
}
```

```
package control;

import java.io.BufferedReader;
import java.io.File;
import java.io.FileReader;
import java.io.FileWriter;
import java.util.ArrayList;
import java.util.List;

import com.thoughtworks.xstream.XStream;
import com.thoughtworks.xstream.io.xml.Dom4JDriver;
import com.thoughtworks.xstream.io.xml.StaxDriver;

import entities.Funcionario;

public class ControleFuncionario extends ControlePessoa<Funcionario> {

	@Override
	public void exportarXml(List<Funcionario> lista) throws Exception {

		XStream xstream = new XStream(new StaxDriver());

		xstream.alias("dados", List.class);
		xstream.alias("funcionario", Funcionario.class);

		FileWriter arquivo = new FileWriter
			(new File(PATH + "funcionarios.xml"));
		arquivo.write(xstream.toXML(lista));
		arquivo.close();
	}

	@Override
	public List<Funcionario> importarXml() throws Exception {

		List<Funcionario> lista = new ArrayList<Funcionario>();
		XStream xstream = new XStream(new Dom4JDriver());
		xstream.alias("dados", List.class);
		xstream.alias("funcionario", Funcionario.class);

		BufferedReader arquivo = new BufferedReader(new FileReader
			(PATH + "funcionarios.xml"));
		lista = (List<Funcionario>) xstream.fromXML(arquivo);
		arquivo.close();

		return lista;
	}
}
```

Executando:

```
package principal;

import java.util.ArrayList;
import java.util.List;

import control.ControleFuncionario;
import entities.Funcionario;

public class Main {

	public static void main(String[] args) {

		/*
		List<Cliente> lista = new ArrayList<Cliente>();
		lista.add(new Cliente(1, "Ana", "1234567890", "ana@mail.com", "21 5555-5555"));
		lista.add(new Cliente(2, "Leo", "0987654321", "leo@mail.com", "21 2222-2222"));
		lista.add(new Cliente(3, "Bia", "9988776655", "bia@mail.com", "21 4444-4444"));
		lista.add(new Cliente(4, "Rui", "2233445566", "rui@mail.com", "21 9999-9999"));

		try{

			ControleCliente c = new ControleCliente();
			c.exportarXml(lista);

			System.out.println("XML gerado com sucesso.\n");
			for(Cliente cli : c.importarXml()){
				System.out.println(cli);
			}
		}
		catch(Exception e){
			System.out.println("Erro: " + e.getMessage());
		}
		*/

		List<Funcionario> lista = new ArrayList<Funcionario>();
		lista.add(new Funcionario(1, "Joao", 2000.0, "Programador"));
		lista.add(new Funcionario(2, "Pedro", 2000.0, "Programador"));
		lista.add(new Funcionario(3, "Maria", 4000.0, "Analista"));
		lista.add(new Funcionario(4, "Carlos", 5000.0, "DBA"));

		try{

			ControleFuncionario c = new ControleFuncionario();
			c.exportarXml(lista);

			System.out.println("XML gerado com sucesso.\n");

			for(Funcionario f : c.importarXml()){
				System.out.println(f);
			}
		}
		catch(Exception e){
```

```
                    System.out.println(e.getMessage());
                }
            }
}
```

XStream

Biblioteca para geração de arquivos XML
http://mvnrepository.com/search?q=xstream