

Padrão de Codificação

Para observar a classe completa acesse o arquivo que esta nesta mesma pasta.

Pacote Model

Cliente.java

```
import exception.ClienteException;

/*Para fazer uma melhor validacao e captura do dados
 * se pega todos os dados como string.
 *
 */
public abstract class Cliente {
    private String nome;
    private String cpf;
    private String telefone;
    private String email;
    protected String matricula;

    //Mensagens de Erro e Alertas
    private final String NOME_INVALIDO = "Nome Invalido.";
    private final String NOME_BRANCO = "Nome em Branco.";
    private final String NOME_NULO = "Nome esta Nulo.";
    private final String CPF_INVALIDO = "CPF Invalido.";
    private final String CPF_BRANCO = "CPF em Branco.";
    private final String CPF_NULO = "CPF esta Nulo.";
    private final String TELEFONE_INVALIDO = "Telefone Invalido.";
    //private final String TELEFONE_BRANCO = "Telefone em Branco.";
    private final String TELEFONE_NULO = "Telefone esta Nulo.";
    //private final String EMAIL_INVALIDO = "E-mail Invalido.";
    //private final String EMAIL_BRANCO = "E-mail em Branco.";
    private final String EMAIL_NULO = "E-mail esta Nulo.";

    public Cliente(String nome, String cpf, String matricula,
        String telefone, String email) throws ClienteException{
        this.setNome(nome);
        this.setCpf(cpf);
        this.setMatricula(matricula);
        this.setTelefone(telefone);
        this.setEmail(email);
    }

    ...

    public void setNome(String nome) throws ClienteException{
        if(nome == null)
            throw new ClienteException(NOME_NULO);
        else if("").equals(nome.trim()))
            throw new ClienteException(NOME_BRANCO);
        else if(nome.trim().matches("[a-zA-Z][a-zA-Z\\s]+"))
            this.nome = nome.trim();
        else
            throw new ClienteException(NOME_INVALIDO);
    }
}
```

```
package model;

import exception.PatrimonioException;

public class Patrimonio {

    private String codigo;
    private String descricao;
    //Mensagens de Erro e Alertas
    //private final String CODIGO_INVALIDO = "Codigo Invalido.";
    private final String CODIGO_BRANCO = "Codigo em Branco.";
    private final String CODIGO_NULO = "Codigo esta Nulo.";
    //private final String DESCRICAO_INVALIDO = "Descricao Invalido.";
    private final String DESCRICAO_BRANCO = "Descricao em Branco.";
    private final String DESCRICAO_NULO = "Descricao esta Nula.";

    public Patrimonio(String codigo, String descricao) throws PatrimonioException
    {
        this.setCodigo(codigo);
        this.setDescricao(descricao);
    }

    public String getCodigo() {
        return codigo;
    }

    public String getDescricao() {
        return descricao;
    }

    public void setCodigo(String codigo) throws PatrimonioException {
        if(codigo == null)
            throw new PatrimonioException(CODIGO_NULO);
        else if ("".equals(codigo.trim()))
            throw new PatrimonioException(CODIGO_BRANCO);
        //else if(codigo.matches("PATTERN"))
        //this.codigo = codigo;
        //else
        //throw new PatrimonioException(CODIGO_INVALIDO);
        this.codigo = codigo;
    }

    public void setDescricao(String descricao) throws PatrimonioException {
        if(descricao == null)
            throw new PatrimonioException(DESCRICAO_NULO);
        else if ("".equals(descricao.trim()))
            throw new PatrimonioException(DESCRICAO_BRANCO);
        this.descricao = descricao;
    }
}
```

Reserva.java

```
package model;

import java.text.SimpleDateFormat;
import java.util.Date;

import exception.ReservaException;

public class Reserva {

    private String hora;
    private String data;

    //Mesagens, Alertas e Padroes
    private final String HORA_NULA = "A hora esta nula.";
    private final String HORA_INVALIDA = "A hora eh invalida.";
    private final String HORA_BRANCA = "A hora esta em branco.";
    private final String HORA_PATTERN = "^([012]?[\\d]:[0-5][\\d])$";
    private final String DATA_NULA = "A data esta nula.";
    private final String DATA_INVALIDA = "A data eh invalida.";
    private final String DATA_BRANCA = "A data esta em branco.";
    private final String DATA_PATTERN = "^([0123]?[\\d])([./-])[01]?[\\d]\\1[\\d]{2,4}$";

    public Reserva(String data, String hora) throws ReservaException {
        this.setData(data);
        this.setHora(hora);
    }

    public String getHora() {
        return this.hora;
    }

    public String getData() {
        return this.data;
    }

    public void setHora(String hora) throws ReservaException {
        if(hora == null)
            throw new ReservaException(HORA_NULA);

        hora = hora.trim();
        if(hora.equals(""))
            throw new ReservaException(HORA_BRANCA);
        else if(hora.matches(HORA_PATTERN)){
            if(hora.length() == 4)
                this.hora = "0" + hora;
            else
                this.hora = hora;
        }
        else
            throw new ReservaException(HORA_INVALIDA);
    }
}
```

Pacote Persistence

ResSalaAlunoDAO.java

```
package persistence;

import java.sql.ResultSet;
import java.sql.SQLException;
import java.text.SimpleDateFormat;
import java.util.Date;
import java.util.Iterator;
import java.util.Vector;

import exception.ClienteException;
import exception.PatrimonioException;
import exception.ReservaException;

import model.Aluno;
import model.ReservaSalaAluno;
import model.Sala;

@SuppressWarnings("unchecked")
public class ResSalaAlunoDAO extends DAO{

    //Mensagens e Alertas
    private final String NULA = "Termo nulo.";
    private final String ALUNO_INDISPONIVEL = "O aluno possui uma reserva no mesmo dia e horario.";
    private final String SALA_INDISPONIVEL = "A Sala esta reservada no mesmo dia e horario.";
    private final String ALUNO_INEXISTENTE = "Aluno inexistente.";
    private final String SALA_INEXISTENTE = "Sala inexistente";
    private final String RESERVA_INEXISTENTE = "Reserva inexistente";
    private final String RESERVA_EXISTENTE = "A reserva ja existe.";
    private final String CADEIRAS_INDISPONIVEIS = "O numero de cadeiras reservadas esta indisponivel para esta sala.";
    private final String DATA_JA_PASSOU = "A data escolhida ja passou.";
    private final String HORA_JA_PASSOU = "A hora escolhida ja passou.";

    //Singleton
    private static ResSalaAlunoDAO instance;
    private ResSalaAlunoDAO(){
    }
    public static ResSalaAlunoDAO getInstance(){
        if(instance == null)
            instance = new ResSalaAlunoDAO();
        return instance;
    }
}
```

ProfessorDAO.java

```
package persistence;

import model.Professor;

import java.sql.*;
import java.util.Vector;

import exception.ClienteException;

public class ProfessorDAO {

    //Mensagens
    private static final String PROFESSOR_JA_EXISTENTE = "O Professor ja esta cadastrado.";
    private static final String PROFESSOR_NAO_EXISTENTE = "O Professor nao esta cadastrado.";
    private static final String PROFESSOR_NULO = "O Professor esta nulo.";
    private static final String PROFESSOR_EM_USO = "Sala esta sendo utilizada em uma reserva.";
    private static final String CPF_JA_EXISTENTE = "Ja existe um professor cadastrado com esse CPF.";
    private static final String MATRICULA_JA_EXISTENTE = "Ja existe um professor cadastrado com essa matricula.";

    //Singleton
    private static ProfessorDAO instance;
    private ProfessorDAO(){
    }
    public static ProfessorDAO getInstance(){
        if(instance == null)
            instance = new ProfessorDAO();
        return instance;
    }
}
```

Pacote Control

ManterSala.java

```
package control;

import java.sql.SQLException;
import java.util.Vector;

import persistence.SalaDAO;
import exception.PatrimonioException;
import model.Sala;

public class ManterSala {

    private Vector<Sala> salas_vet = new Vector<Sala>();

    //Singleton
    private static ManterSala instance;
    private ManterSala() {
    }
    public static ManterSala getInstance() {
        if(instance == null)
            instance = new ManterSala();
        return instance;
    }
    //

    public Vector<Sala> getSalas_vet() throws SQLException, PatrimonioException{
        this.salas_vet = SalaDAO.getInstance().buscarTodos();
        return this.salas_vet;
    }

    public void inserir(String codigo, String descricao, String capacidade) throws
    PatrimonioException, SQLException {
        Sala sala = new Sala(codigo, descricao, capacidade);
        SalaDAO.getInstance().incluir(sala);
        this.salas_vet.add(sala);
    }
}
```

ManterResSalaAluno.java

```
package control;

import java.sql.SQLException;
import java.util.Vector;

import model.Aluno;
import model.ReservaSalaAluno;
import model.Sala;
import persistence.ResSalaAlunoDAO;
import exception.ClienteException;
import exception.PatrimonioException;
import exception.ReservaException;

public class ManterResSalaAluno {

    private Vector<ReservaSalaAluno> rev_sala_aluno_vet = new
Vector<ReservaSalaAluno>();
    //Singleton
    private static ManterResSalaAluno instance;

    private ManterResSalaAluno() {
    }

    public static ManterResSalaAluno getInstance() {
        if (instance == null) {
            instance = new ManterResSalaAluno();
        }
        return instance;
    }
    //

    public Vector<ReservaSalaAluno> getReservasHora(String hora) throws SQLException,
PatrimonioException, ClienteException, ReservaException{
        return ResSalaAlunoDAO.getInstance().buscarPorHora(hora);
    }

    public Vector<ReservaSalaAluno> getReservasMes(String data) throws SQLException,
PatrimonioException, ClienteException, ReservaException{
        return ResSalaAlunoDAO.getInstance().buscarPorDia(data);
    }

    public Vector<ReservaSalaAluno> getResAlunoSala_vet() throws SQLException,
PatrimonioException, ClienteException, ReservaException {
        this.rev_sala_aluno_vet = ResSalaAlunoDAO.getInstance().buscarTodos();
        return this.rev_sala_aluno_vet;
    }

    public int cadeirasDisponveis(Sala sala, String data, String hora) throws
SQLException, PatrimonioException, ClienteException, ReservaException {
        return ResSalaAlunoDAO.getInstance().cadeirasDisponiveis(sala, data, hora);
    }

    public void inserir(Sala sala, Aluno aluno,
        String data, String hora, String finalidade, String cadeiras_reservadas)
        throws SQLException, ReservaException, ClienteException,
        PatrimonioException {

        ReservaSalaAluno r = new ReservaSalaAluno(data, hora, sala, finalidade,
cadeiras_reservadas, aluno);
        ResSalaAlunoDAO.getInstance().incluir(r);
        this.rev_sala_aluno_vet.add(r);
    }
}
```

Pacote View

No Pacote View não se teve a preocupação por padrão já que a view é gerada por ferramentas.