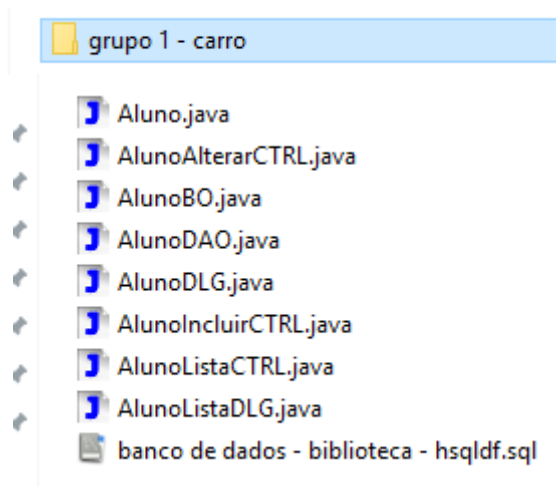



Projeto CRUD(Create, Read,update e delete)

O termo “CRUD” é utilizado no mundo do desenvolvimento de sistemas para designar as quatro operações básicas do gerenciamento de registros em um banco de dados. As operações Criar, Ler (listar), Atualizar e Excluir (Deletar).

Instruções gerais do trabalho

- Formação grupo para estudo dirigido para complementar carga horaria:
 - ◆ O grupo deverá obrigatoriamente ser composto no mínimo de 3 alunos e no máximo de 4 alunos. A formação de um grupo com um número diferente dependerá de aprovação Professor
 - ◆ Entrega da lista do grupo para professor
- Sorteio do grupo para as equipes será feito com um gerador aleatório feito em java
- A solução do projeto crud, deve ser enviada para o email wanderley.unip@gmail.com em formato zip. Nesse arquivo zip deverá conter os seguintes arquivos (exemplo abaixo): scripts do banco, classe view, controle, model até xx/xx/xxxx 18:00. **O projeto crud é do grupo e com prazo definido, de forma que cópias e soluções enviadas fora do prazo serão desconsiderados.**
 - **Requisito de envio: Campo assunto**
 - **Grupo numero – nome do tema**
Exemplo : grupo 1 – 10 carro



 UNIVERSIDADE PAULISTA Instituto de Ciências Exatas Campus Brasília	Curso: Ciência da Computação / Sistema de Informação Disciplina: ALPOO Prof(a): Msc Wanderley Gonçalves Freitas Projeto CRUD
---------------------------------------------------------------------------------------------------------------------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------

1. Objetivo

O objetivo é mostrar uma aplicação funcional completa que incorpore os seguintes:

- Parte I - JDBC
 - ◆ Formulário desktop – Swing e controlador
 - ◆ Criação do scripts do banco - HSQLDB
 - ◆ Visão (formulário); controle (controlador) e model (DAO, BO,VO).
- Parte II - JPA
 - ◆ Formulário web – JSF e manage bean
 - ◆ Visão (formulário); controle (controlador) e model (DAO, BO,VO).

Justificativa

Capacitar o aluno na elaboração e construção de sistemas orientados a objetos com os conceitos adquiridos nas disciplinas do curso.

Funcionalidade

A equipe deverá construir uma aplicação com arquitetura de software MVC que seja capaz de gerenciar um cadastro básico (crud).

Requisitos Funcionais

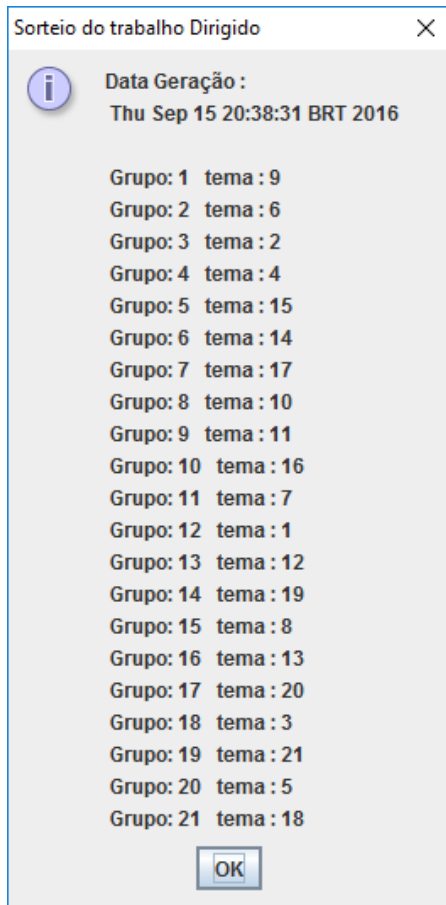
- ◆ Inclusão registro
- ◆ Listar todos os registros
- ◆ Consultar registro por campo específico
- ◆ Alterar um registro
- ◆ Remover um registro

Requisitos não-funcionais

- Parte I
 - ◆ Aplicação deve apresentar uma interface windows;
 - ◆ A aplicação deve ser flexível para utilizar bancos de dados relacionais - HSQLDB;

- Parte I
 - ♦ Aplicação deve apresentar uma interface web;
 - ♦ A aplicação deve ser flexível para utilizar bancos de dados relacionais - HSQLDB;

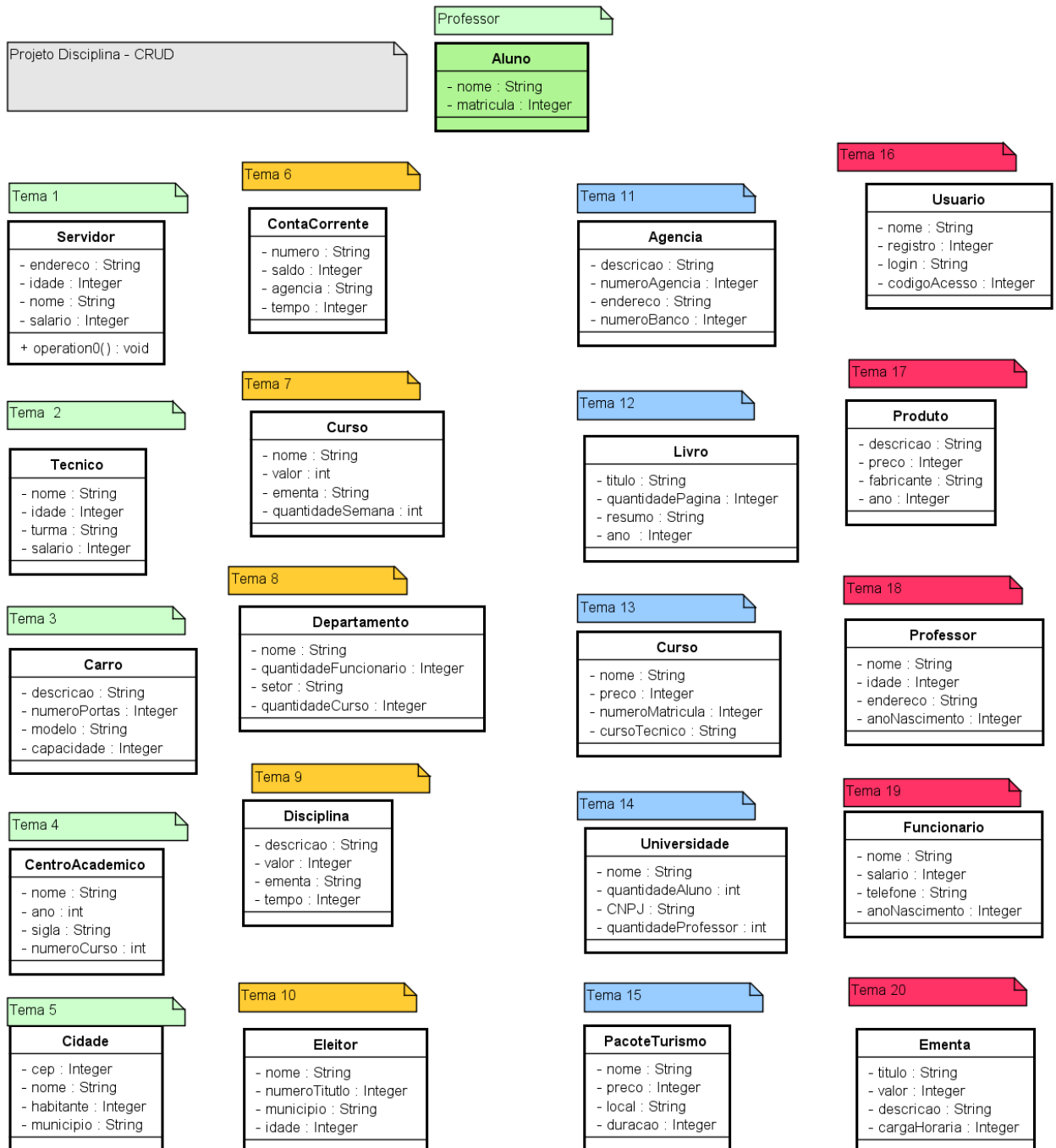
2. Sorteio dos grupos




3. Grupos

matricula	Nome	Turma	Grupo
C69BIG5	ANDRE LUIS MOURA SILVA	CC4P30	1
C57BI16	CHRISTIAN DINIZ S TAVARES	CC4P30	1
C696FI8	JEAN OLIVEIRA GOMES	CC4P30	1
C5563G7	LUIZ FILIPE VIEIRA	CC4P30	1

4. Temas



 <p>UNIVERSIDADE PAULISTA Instituto de Ciências Exatas Campus Brasília</p>	<p>Curso: Ciência da Computação / Sistema de Informação Disciplina: ALPOO Prof(a): Msc Wanderley Gonçalves Freitas Projeto CRUD</p>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------

5. Atividades

1. Construir o **script da tabela** e massa de dados para funcionalidade (**SGBDR - HSQLDB**). Tendo como regra de formação do nome: **nome do tema**
Exemplo : Aluno.sql
2. Construir classe VO (value object) entidade encapsulada da funcionalidade do grupo que herda classe GenericoVO (br.unip.alpoo.biblioteca.parte1.model.entidade). Tendo como regra de formação do nome: **nome da funcionalidade**
Exemplo : Aluno
3. Construir classe DAO (Data Access Object) da funcionalidade do grupo que herda classe GenericoDAO. (br.unip.alpoo.biblioteca.parte1.model.persistencia). Tendo como regra de formação do nome: **nome da funcionalidadeDAO**
Exemplo : AlunoDAO
4. Construir classe BO (Business Object) da funcionalidade do grupo que herda classe GenericoBO. (br.unip.alpoo.biblioteca.parte1.model.negocio). Tendo como regra de formação do nome: **nome da funcionalidadeBO**
Exemplo : AlunoBO
5. Construir classe DLG (JDialog) da funcionalidade do grupo que herda classe JDialog. (br.unip.alpoo.biblioteca.parte1.swing). Tendo como regra de formação do nome: nome da funcionalidadeDLG;
 - ✓ funcionalidadeDLG
 - ✓ funcionalidadeListarDLG
 Exemplo : AlunoDLG ; AlunoListaDLG
6. Construir classe CTRL da funcionalidade do grupo (br.unip.alpoo.biblioteca.parte1.swing). Tendo como regra de formação do nome: nome da funcionalidadeCTRL;
 - ✓ funcionalidadeIncluirCTRL;FuncionalidadeAlterarCTRL
 - funcionalidadeListaCTRL
 Exemplo : AlunoIncluirCTRL; AlunoAlterarCTRL; AlunoListaCTRL