

**Projeto da disciplina de Algoritmos e Estrutura de Dados**  
**Departamento de Estatística e Informática**  
**Universidade Federal Rural de Pernambuco**  
**Prof. Tiago A. E. Ferreira**

## **Projeto 1: Aprovação de Transação de Cartão de Crédito**

### **1. Definição**

Ao utilizar um cartão de crédito em um estabelecimento comercial para a realização de uma compra, a maquineta onde se passa o cartão irá realizar uma consulta no sistema da administradora do cartão de crédito para verificar se aprova ou não a transação comercial. Tal consulta deve ser realizada on line, ou seja, a resposta deve ser em tempo real ou pelo menos gastar um tempo aceitável. Para toda compra realizada haverá uma comissão de 2,0 % para a administradora do cartão e o pagamento para o estabelecimento ocorrerá todo dia 10 de cada mês. Deve-se projetar um sistema para aprovação de crédito on line.

### **2. Estruturas de Dados e Detalhes de Implementação**

Tal sistema de aprovação de crédito deve ser feito da seguinte forma: armazenar em uma árvore vermelho-preto os números de todos os cartões de créditos válidos, chamada árvore de crédito, com todos os dados do respectivo cliente (inclusive seu limite de crédito) e armazenar em outra árvore vermelho-preto os estabelecimentos comerciais cadastrados para o uso dos cartões de crédito, chamada árvore de estabelecimentos. O sistema é composto de partes: Cadastramento de Cartões de Crédito e Estabelecimentos Comerciais, Consulta para autorização de compra, Relatório de pagamentos aos estabelecimentos comerciais.

#### **2.1. O cadastramento de Cartões de Crédito e Estabelecimentos:**

- Cadastrar Cartão:  
Campos: Número, Bandeira, Nome do Cliente, Limite Total, Limite disponível;
- Cadastrar Estabelecimento Comercial:  
Campos: Nome, Endereço, horário de funcionamento, montante comercializado no mês, Valor a ser pago no mês;

#### **2.2. Aprovação da Operação:**

- A operação de compra deverá ser aprovada se:
  - 1) O cliente tiver limite disponível;
  - 2) Se a compra estiver sendo realizada em um estabelecimento cadastrado e dentro do seu horário de funcionamento.

Observação: O sistema deve ser capaz de gerar relatórios sobre um cliente e/ou estabelecimento comercial, informando em ordem decrescente de montante negociado tanto por cliente como por estabelecimento.

### **3. Linguagem e Paradigma de Programação**

O sistema proposto neste projeto deve ser implementado em Python no paradigma de Orientação a Objetos.

# Projeto 2: Biblioteca: Verificando Livros

## 1. Definição

Em uma biblioteca há um conjunto de livros que podem ser emprestados para os usuários devidamente cadastrados. Para um livro ser emprestado ele deve estar disponível (presente na biblioteca e sem reserva) e o usuário deve estar em dia com a biblioteca e não ter excedido o número máximo de empréstimos.

## 2. Estruturas de Dados e Detalhes de Implementação

As verificações mencionadas na descrição acima devem ser feitas da seguinte forma: armazenar em uma árvore vermelho-preto todos os títulos válidos, chamada árvore de livros e armazenar em outra árvore vermelho-preto os números de todos os usuários, chamada árvore de usuários. O sistema é composto de duas partes: Cadastramento de livros e usuários.

### 2.1. O cadastramento de livros e usuários possuem as seguintes opções:

- Cadastrar livro ;  
A árvore deve conter um título por nodo.
- Descadastrar título ;
- Cadastrar usuário;  
A árvore deve conter um usuário por nodo.
- Descadastrar usuário;

### 2.2. O sistema possui as seguintes opções:

- O usuário pode realizar uma busca por livros;
- O usuário pode solicitar empréstimo do livro;
- A biblioteca deve poder dar baixa nos empréstimos contra devolução dos livros
- Apresentar o relatório por livro e por usuário.

## 3. Linguagem e Paradigma de Programação

O sistema proposta neste projeto deve ser implementado em Python no paradigma de Orientação a Objetos.

# Projeto 3: Eleições – Contagem e Checagem de Votos

## 1. Definição

Nas nossas eleições é sabido que são utilizadas urnas eletrônicas. É desejado a construção de um sistema que consiga contar os votos e checar quais os títulos que realizaram o processo de votação e os que não votaram. Algumas medidas a serem tomadas são: verificar no momento que um indivíduo vota se o título de eleitor dele é válido e, verificar também, se este eleitor ainda não votou.

## 2. Estruturas de Dados e Detalhes de Implementação

As verificações mencionadas na descrição acima devem ser feitas da seguinte forma: armazenar em uma árvore vermelho-preta os números de todos os títulos válidos, chamada árvore de eleitores e armazenar em outra árvore binária de busca os números de todos os títulos que já votaram até o momento, chamada árvore de votação. O sistema é composto de duas partes: Cadastramento de Títulos de Eleitor e Votação.

### 2.1. O cadastramento de títulos de eleitor possuem as seguintes opções:

- Cadastrar título;
- Descadastrar título ;
- Carregar títulos;  
O grupo deverá gerar números automaticamente para preencher a árvore.
- Descadastrar usuário;

### 2.2. A votação possui as seguintes opções:

- Cadastrar candidatos (nome e número);
- Nova votação: dá um “reset” na árvore de votação que guarda os títulos que já votaram.  
Esse “reset” corresponde à liberação da memória dinamicamente alocada;
- Adicionar voto: ler número do título e o voto. Se o número do título é válido e se ainda não votou, contabiliza o voto e atualiza a árvore de votação que armazena os títulos de quem já votou;
- Gerar votos aleatórios (para encher a árvore de votação mais rapidamente): sempre acionando a operação de inserir mais um elemento na árvore de votação;
- Apresentar o resultado parcial da eleição;
- Sair: destruir todas as estruturas e encerrar o programa.

## 3. Linguagem e Paradigma de Programação

O sistema proposta neste projeto deve ser implementado em Python no paradigma de Orientação a Objetos.

# Projeto 3: Multas de Trânsito – Pontos na Habilitação de Motorista

## 1. Definição

O nosso código de trânsito prevê basicamente quatro tipos de multas: fraca, média grave e gravíssima, onde cada uma destas pontua com um determinado valor. Se um dado condutor obtiver em um prazo de 12 meses uma pontuação superior a um determinado limiar, este terá sua habilitação de motorista cassada. Esta pontuação é atrelada a placa do veículo e ao seu proprietário.

## 2. Estruturas de Dados e Detalhes de Implementação

As verificações mencionadas na descrição acima devem ser feitas da seguinte forma: armazenar em uma árvore vermelho-preta os números de todas as placas válidas de veículos, chamada árvore de veículos e armazenar em outra árvore binária de busca todos os proprietários de veículos chamada árvore de Proprietário. Cada multa recebida por um veículo deve ser armazenada na árvore veículo e os pontos computados ao seu respectivo proprietário na árvore de proprietários. O sistema é composto de duas partes: Veículos e proprietários.

### 2.1. O cadastramento de veículo possuem as seguintes opções:

- Cadastrar/ Descadastrar nome do veículo;  
O grupo deverá gerar números automaticamente para preencher a árvore.
- Identificador do proprietário;
- Local, data e hora das multas;
- Data de vencimento das multas;
- *Status* do veículo (em dia ou atrasado);

### 2.2. O cadastramento de proprietário possui as seguintes opções:

- Cadastrar Nome ;
- Cadastrar Endereço;
- Cadastrar habilitação;
- Cadastrar veículos que o mesmo possui;
- Cadastrar pontos na Habilitação;
- *Status*: o proprietário terá sua carteira cassada?

## 3. Linguagem e Paradigma de Programação

O sistema proposta neste projeto deve ser implementado em Python no paradigma de Orientação a Objetos.

## **Regras Gerais para todos os Projetos**

- O Projeto deverá ocorrer em grupos de no máximo 4 pessoas;
- Cada Grupo pode escolher um projeto dentre os apresentados;
- Os trabalhos deverão ser entregues em alguma mídia contendo os arquivos-fonte (classes em python) e um relatório descrevendo o projeto feito;
- Haverá defesa oral do projeto;
- Qualquer tipo de cópia atribuirá nota zero ao projeto (inclusive com os projetos já feitos por outras turmas!);
- O projeto terá peso de 40% para a nota da 2aV.A.;
- Data de Entrega e Apresentação: **16/06/2011**.