**Universidade Estadual da Paraíba**

**Centro de Ciência e Tecnologias**

**Departamento de Computação**

**Professora:** Luciana Gomes

**Disciplina:** Linguagem de Programação 2

**Período:** 2018.1

Projeto: Pet Shop “Mundo Cão”

# Descrição do Problema

Deseja-se um sistema que informatize um Pet Shop. O sistema deve registrar os **serviços solicitados** pelos clientes (banho, tosa, escovação do pelo, escovação especial para os dentes, entre outros) e os **produtos que os clientes pretendem comprar** para seus pets (brinquedos, roupinhas, coleiras, rações, remédios, por exemplo). Os pedidos de serviços, produtos ou marcação de consultas devem ser realizados pelos funcionários do pet shop, que registra no sistema o que lhe foi solicitado. Serviços e produtos devem ser cadastrados e anotados pelo funcionário do pet shop com o código identificador do produto (100= Lacinho, 230=ração para gatos Whiscas, etc.) ou serviço solicitado (100= banho, 230=tosa, etc.). O programa deve ser capaz de informar o balanço do caixa. Na abertura do caixa, deve-se informar o montante em dinheiro disponível. As contas podem ser pagas em dinheiro, cheque ou cartão de crédito. O sistema também deve ter disponíveis pacotes que serão um combo de serviços oferecidos (por exemplo: banho, tosa e hotelzinho para um feriado prolongado) e esses pacotes devem ser cadastrados no sistema como tal.

# Requisitos Funcionais

O sistema de caixa automatizado deve possuir as seguintes funções:

* Abertura do caixa
* Movimentação do caixa (recebimento, troco, etc.)
* Cadastro de produtos (código, descrição, preço, etc)
* Cadastro de serviços (código, descrição, preço, etc)
* Cadastro de pacotes (código, descrição, preço, etc)
* Fornecimento de balanço
* Cadastro de clientes
* Cadastro de pets
* Cadastro de funcionários

Na movimentação do caixa, quando o cliente efetua o pagamento de sua conta, o sistema deve registrar o código do produto, serviço ou pacote e a quantidade solicitada. O sistema deve informar o preço a ser pago a cada item da conta ao lado da descrição do item e o total final. O sistema deve guardar todas as contas contabilizadas. Cada produto, serviço ou pacote registrado deve conter um código, uma descrição e o preço. O sistema deve ser capaz de informar o balaço atual do caixa: trata-se de uma listagem das datas/horas de cada conta paga com o seu respectivo valor total. A nota fiscal emitida para o cliente deve ter o código do funcionário que fez o atendimento para controle do dono do pet shop.

## Interface com o usuário

A interface com o usuário pode ser textual ou gráfica (estudar o pacote javax.swing). O sistema deve exibir um menu principal com as funções acima. Cada menu deve exibir um outro menu ou pedir dados para o usuário, dependendo do caso.

### **Menu cadastro de produtos**

Deve receber os dados de um produto, dentre eles:

* Código do produto
* Descrição
* Preço

### **Menu cadastro de serviços**

Deve receber os dados de um serviço, dentre eles:

* Código do serviço
* Descrição
* Preço

### **Menu cadastro de pacotes**

Deve receber os dados de um pacote, dentre eles:

* Código do pacote
* Descrição
* Itens do pacote
* Preço de cada item (aqui pode ser dado um desconto no cálculo final do pacote para que a sua aquisição se torne atraente para o cliente).

### **Menu cadastro de funcionários**

Deve receber os dados de um funcionário:

* Código
* Nome

### **Menu cadastro de clientes**

Deve receber os dados de um cliente:

* Código
* Nome
* Lista de pets do cliente (nome do animal, espécie, raça)

### **Cadastro de Pets**

Deve receber os dados de um pet:

* Código
* Nome
* Espécie
* Raça

### **Menu Abertura do Caixa**

Deve receber o montante de dinheiro inicial do caixa.

### **Menu Movimentar Conta**

Deve receber o número sequencial da conta ou qualquer outra informação que identifique a conta, cada item solicitado e o código do funcionário (após a digitação deste último deve aparecer o nome do funcionário na tela). Esta tela vai permanecer aberta a maior parte do tempo, pois é a função principal do sistema. Após digitar todos os itens, o sistema deve informar o subtotal. O valor pago deve ser recebido e o troco informado, se for o caso. Deve ser possível abrir outra conta sem ter que recorrer ao menu principal novamente. A tela poderia oferecer opções fechar, cancelar e nova conta, por exemplo.

### **Menu Fornecer Balanço**

O balanço é um relatório sobre a movimentação do caixa até o momento. A listagem deve exibir o dia e hora em que cada conta foi fechada e o valor total pago. Ao final, o total arrecadado até o momento pelo caixa.

# Documentos Produzidos

O grupo deve entregar os seguintes documentos:

* **Protótipo funcional** do software especificado neste documento implementado na linguagem de programação Java e seguindo uma metodologia orientada a objetos.
* **Documentação** de todas as classes no formato produzido pelo aplicativo **javadoc**.
* **Diagrama de classes** na notação UML. Em cada classe devem constar o seu nome, atributos e operações com sua assinatura completa. As relações de herança e agregação devem estar presentes também.
* **Manual do usuário**, descrevendo como realizar as principais operações do sistema. Este manual deve levar em contra que o sistema será usado por alguém que não leu o código fonte.

# Requisitos Não Funcionais

Os seguintes requisitos não funcionais deverão pautar todo o projeto e desenvolvimento do software:

**Código fonte**

* Código fonte documentado. Endentação e organização do código;
* Regras de nomenclatura para os identificadores segundo o padrão;
* Modularização do código fonte por uso de pacotes Java;
* Uso de herança, interfaces, classes abstratas;
* Tratamento e definição de novas exceções;
* Aplicação correta dos modificadores aos membros da classe, quando necessário (final, abstract, private, protected, etc.);
* Armazenamento em arquivo (streams);
* O código deve utilizar-se dos conceitos da orientação a objetos e das facilidades oferecidas pela linguagem Java, porém sem exageros. Utilize o bom senso e explore os recursos da linguagem fornecidos pela API atual.

**Interface**

* Interface intuitiva, de aparência agradável e fácil de usar.
* Utilização dos recursos da API Swing na confecção dos elementos gráficos da interface.
* Interface robusta, capaz de resistir a erros cometidos pelo usuário.

# Prazos e outras informações

Data da entrega e apresentação:

O trabalho deverá ser apresentado pela equipe. Cada equipe terá 15 minutos para explicar, de forma geral, detalhes e estratégias de sua implementação (classes e sua hierarquia, detalhes de implementação de uma classe, interfaces e suas vantagens no projeto, são alguns exemplos). Esta apresentação integrará a nota, e a participação, condução da apresentação, pontualidade e domínio demonstrados pelos alunos serão pontuados. Esta apresentação não deverá ser uma demonstração completa do software, devem ser apresentadas as principais funcionalidades e também poderá ser utilizada na demonstração alguma parte do sistema para ilustrar um aspecto inovador, por exemplo. A equipe deve instalar o sistema antes da data para não comprometer o tempo da apresentação.

* A nota será dada ao grupo. Todos devem demonstrar participação, integração e domínio completo sobre a implementação utilizada.
* Não haverá reposição para o projeto.
* Serão aceitos projetos fora do prazo acordado, com a penalização de 1,0 ponto por dia.
* Os projetos deverão ser enviados por e-mail.
* Qualquer cópia, mesmo parcial, de alguma funcionalidade do sistema, seja de outra equipe ou de terceiros será desconsiderada e a equipe receberá automaticamente ZERO. Os alunos de outras equipes podem ser consultados, desde que nenhum código seja fornecido. Não caia na tentação! Afaste o demônio da preguiça e apegue-se apenas ao Deus Java para solução dos seus problemas...