PetShopMoreira.exe

UC - Programação De Soluções Computacionais
Universidade Anhembi Morumbi
Unidade - Mooca

ArrayList<String> Integrantes

1° - Jordan Estevan Rodrigues Dos Santos - 125221103657

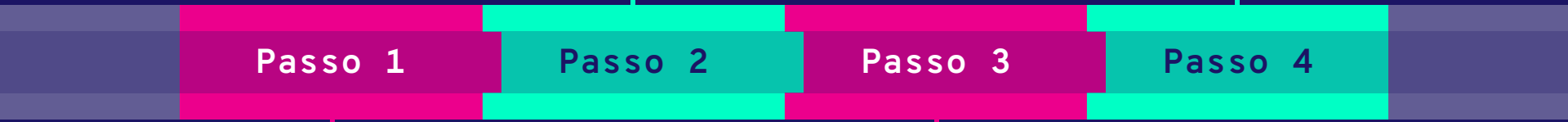
2° - Gustavo Lopes Silva - 12522212116

3° - Vilson da Silva Juvencio Junior - 12522218378

INTRODUÇÃO

Levando em consideração todas as dificuldades e estresses que enfrentamos na correria do dia, desenvolvemos um App, que efetua a marcação de consultas e simplifica processos do dia a dia de um **PetShop**.

Separando os Passos



Identificar os problemas

Analisar o ambiente a qual o programa será destinado e identificar seus empecilhos.

Levantar os requisitos

A partir da identificação dos problemas e dificuldades, Levantar os requisitos destinados a resolução do problema.

Codificação

Iniciar a codificação do programa, se baseando nos requisitos levantandos pela análise.

Testes

Após o término da codificação, testar a aplicação e capturar todas as exceções que serão encontradas.

IDENTIFICANDO O PROBLEMA

“Tecnologia é a habilidade de organizar o mundo de forma que não tenhamos que senti-lo.”

—Max Frisch

E QUAIS SÃO ESSES PROBLEMAS?

- FALTA DE ORGANIZAÇÃO – É muito comum que hoje em dia com a quantidade de informações, um comércio acabe cometendo erros por falta de organização, no caso do pet shop, é muito recorrente em um ambiente sem organização perder a data de algum cliente e seus dados.

- CONSUMO DE TEMPO – Tempo é dinheiro, isso é um fato, principalmente em um século onde tudo se acelera cada vez mais, dito isto, é notório que os comércios precisam otimizar de alguma forma suas tarefas, e a falta de organização que já foi citada, é uma grande inimiga disso somado a realização de algumas tarefas que poderiam ser automatizadas/otimizadas por um software, um cadastro de horário pode levar de 8 a 13 minutos de maneira manuscrita, enquanto digitalmente este tempo pode cair pela metade.

- SEGURANÇA – Com o avanço da tecnologia também aumenta-se o número de pessoas mal intencionadas, e ninguém quer os dados de seus clientes se perdendo ou sendo alterados, então faz-se explicitamente necessário que acompanhemos esse avanço com 1 passo a frente, mantendo a Segurança dos dados de um cliente nossa grande prioridade, fazendo com que eles não se percam nem sejam alterado.

- FALTA DE MODERNIDADE – Convenhamos que no ápice do século 21 e sua moderna tecnologia, algo que certamente não queremos é nos apegarmos a métodos antigos e datados, sendo que podemos por exemplo, automatizar algumas funções com um alguns cliques.

LEVANTANDO SEUS REQUISITOS

Os problemas apresentados anteriormente são muito comuns hoje em dia, e acabam ocorrendo em diversos tipo de instituições e comércios, não ficando de fora o pet shop escolhido para realizar a aplicação

Dito isto, dá se inicio ao levantamento de requisitos visando a otimização e solução dos problemas apresentados anteriormente:

- RF01 – O Sistema deve ser capaz de cadastrar um horário
- RF02 – O sistema deve ser capaz de armazenar os horário cadastrados em um vetor
- RF03 – O Sistema deve ser capaz de exibir os horários cadastrados
- RF04 – O sistema deve ser capaz de cadastrar novos funcionários
- RF05 - O sistema deve ser capaz de armazenar os funcionários em um vetor
- RF06 – O Sistema deve ser capaz de validar os parâmetros passados pelo usuário na tela de login, percorrendo o vetor de funcionários
- RF07 - O Sistema deve ser capaz de trocar de user durante sua execução.

E DEPOIS DO PROBLEMA

VEM A SOLUÇÃO

ENTENDENDO O PROGRAMA

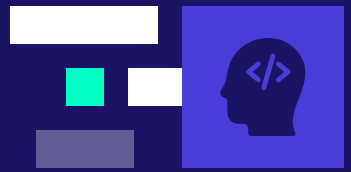
DEFINIÇÃO DAS CLASSES SIMPLES

Pessoa

Classe abstrata que servirá de herança para Funcionário, possui os atributos: String nome, String numtelef.

Funcionario

Filha de Pessoa. Responsável por determinar o que um funcionário é. Armazena os seguintes atributos: String user, senha; Int totalId (static), idpessoal; Boolean admin.



Horario

Responsável por determinar o que um horário tem. Armazena os seguintes atributos: String dia, hora, dono, raça, acao; Int qtdHorario (static), idPessoal; Double: valor

CLASSE LOGIN

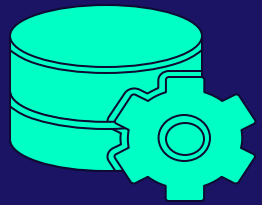
Classe Responsável pelo Login, busca no vetor que usamos como banco de dados, os dados passados pelo usuário e valida se os mesmos existem e coincidem, caso sim, permite o login.

Atributos:

- `private static boolean loginAprovado;`
- `private static Funcionario funcionarioLogado;`
- `private static boolean bg;`
- `private static int x;`
- `private static int contador;`
- `private static int num = 0;`

Método:

`Public Static Void realizarLogin();`



`BancoBDDS.funcionários[];`

Percorre o banco de dados e verifica se os dados inseridos pelo usuário existem e coincidem

BANCO DE DADOS

Classe BandoDDS com os 2 vetores que foram usados como banco de dados, sendo eles, o vetor Funcionarios e o Vetor Horários, além disso também está presente na classe, seus métodos de acesso, herdados da interface AcoesBBDS

Atributos:

- private boolean admcheck;
- private boolean idExist;
- private boolean firstHorario;
- private boolean firstCadastro;
- Instancia do Jpanel
- Instancia do JTextField

Vetores:

- static Horario[] horarios = new Horario[Horario.getQtdHorario() + 3];
- static Funcionario[] funcionarios = new Funcionario[Funcionario.getTotalid() + 3];

Métodos

- Void cadastrarHorário();
- Void CadastrarUsuário();
- Void cadastroSys (String nm, String usr, String snh, boolean adm);

INTERFACE:

AcoesBDDS

- Void cadastrarHorário();
- Void cadastrarUsuário();
- Void cadastroSys(String nm, String usr, String snh, boolean adm);

@OVERRIDE

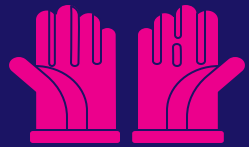
ENUMS

Foi feito o uso de enumeradores nesta aplicação para servir como opções no JOptionPane, fazendo o úsuário escolher uma opção predefinida.

HORARIOS

```
DEZMANHA("10:00"),
QUATORZETARDE("14:00"),
DEZESSEISTARDE("16:00"),
DEZENOVENOITE("19:00");
```

Descrição retornável que possui o horário em uma maneira mais legível.



SERVIÇOS

```
TOSASIMPLES("Tosa Simples", 50),
TOSACOMPLETA("Tosa Higiênica", 85),
BANHO("Banho", 45),
BANHOETOSA("Banho e Tosa", 110);
```

Descrição retornável que possui o nome da operação e o valor da respectiva.

Obrigado pela sua atenção !

Obrigado pelo semestre !

BOAS FESTAS E UM ÓTIMO FIM DE ANO !