



**UNIVERSIDADE DE FORTALEZA**

**CURSO DE ANÁLISE E DESENVOLVIMENTO DE SISTEMAS**

**DISCIPLINA: N704-PROGRAMAÇÃO FUNCIONAL**

## **“ESTACIONAMENTO SHEI”**

### **INTEGRANTES DA EQUIPE:**

Francisca Erivânia Teixeira de Sousa - 2222915

Hemerson Cordeiro da Silva - 2222801

Humberto Barroso Neto - 2226019

Isabel Cristina Monteiro de Carvalho - 2222821

**FORTALEZA-CE**

**SETEMBRO/2024**

## Introdução:

O propósito deste trabalho é a criação de uma aplicação denominada “ESTACIONAMENTO SHEI”, que possibilita o cadastramento de clientes de um estacionamento.

Para a criação da aplicação foi utilizada a linguagem Python. Linguagem esta que teve de ser apreendida pelo grupo durante a execução deste trabalho, pois nenhum dos membros possuía prévia familiaridade com a referida linguagem.

A aplicação foi planejada da seguinte forma:

1. Login: usuário Master que terá acesso a modificar opções importantes, como cadastrar e listar clientes. A senha deverá seguir padrões de segurança e criptografia;
2. Cadastrar Cliente: os clientes serão cadastrados com nome, placa e hora de entrada;
3. Listar Cliente: a listagem mostrará os clientes ativos, há quanto tempo estão e o valor que está sendo cobrado até o momento. Neste campo também mostrará os clientes que são mensalistas e diaristas;
4. Sair: sai da aplicação total e será necessário novo login para voltar.

Conforme exigido no escopo da Proposta do Trabalho, dividimos os elementos obrigatórios do código da seguinte maneira:

- 1) Função lambda foi usada para calcular o preço a ser pago
  - a cada hora, dez reais
  - a cada 15min, 3 reais e a
  - diária é 25 reais;

- 2) List Comprehension foi usada para listar os clientes ativos, separando os diaristas dos mensalistas;

- 3) A closure foi utilizada dentro da Função de Alta Ordem para adicionar clientes, encapsulando a lógica de registro dos mesmos.

Endereço do GitHub onde o trabalho poderá ser acessado:

<https://github.com/EriTXS/ProgramacaoFuncional.git>

## Definição de Papéis

TAREFA	RESPONSÁVEL
Elaborar a documentação de requisitos	Humberto Barroso Neto
Implementar os códigos atendendo os tópicos exigidos na proposta da avaliação, utilizando o editor VS Code.	Francisca Erivânia Teixeira de Sousa e Isabel Cristina Monteiro de Carvalho
Elaborar teste de clientes (se eles foram inseridos corretamente) e o teste de senha.	Hemerson Cordeiro da Silva

## Requisitos de Sistema

### Requisitos Funcionais:

#### 1. Autenticação de Usuário:

- O sistema deverá permitir que usuários façam login.
- Implementação: Na função login(), onde verifica credenciais.

#### 2. Cadastro de Clientes:

- O sistema deverá permitir o cadastro de novos clientes com nome, placa e tipo (se diarista ou mensalista).
- Implementação: Na função cadastrar\_cliente(), que usa registrar\_cliente.

#### 3. Listagem de Clientes:

- O sistema deverá listar clientes ativos, mostrando tempo e valor acumulado, separados entre diaristas e mensalistas.
- Implementação: Na função listar\_clientes(), que usa list comprehension para separação entre diaristas e mensalistas.

#### 4. Cálculo de Preços:

- O sistema deverá calcular o preço a ser pago com base no tempo de permanência.
- Implementação: Função lambda calcular\_preco.

#### 5. Encerramento de Sessão:

- O sistema deverá permitir que o usuário saia, exigindo novo login para retorno.
- Implementação: No menu(), opção "3. Sair".

### Requisitos Não Funcionais:

#### 1. Segurança das Credenciais:

- O sistema deverá garantir segurança para as credenciais armazenadas.
- Implementação: Nas funções criptografar\_senha() e verificar\_senha(), que usam a biblioteca bcrypt.

#### 2. Eficiência na Consulta de Dados:

- O sistema deverá listar rapidamente os clientes organizados por tipo.
- Implementação: Uso de list comprehension em listar\_clientes().

#### 3. Manutenção Simples:

- O sistema deverá ser modular e fácil de manter.
- Implementação: Uso de funções separadas e closures (criar\_cliente\_fun).

### Observações:

#### • Função de Alta Ordem (Closure):

- Utilizada para encapsular o registro de clientes.
- Implementação: criar\_cliente\_fun() cria a função registrar\_cliente.