

## 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.