0 -Lista de funcionários da portaria:

João Carlos Ferreira Nascimento

Matricula: 0001

-----------------------------------------------------------------------------------------------------

Laura Conceição Cardoso

Matricula: 0002

------------------------------------------------------------------------------------------------------

Anderson Pestana

Matricula: 0003

1. Cadastrar Veículo: Cadastrado

Placa : BEE4R22

CPF: 52998224725

Modelo : Palio

Nome: Falcão Cardoso Falópio

Bloco: 08 A

Apartamento: 805

-----------------------------------------------------------------------------------------------2- Cadastro de Veículo:

Placa: BCE 4F33

CPF: 390.533.447-05

Modelo: Honda Civic

Nome: Romano Pereira Carvalho

Bloco: 15 B

Apartamento: 501

-----------------------------------------------------------------------------------------------

3- Cadastro de Veículo:

Placa:ADE7F32

CPF: 123.456.789-09

Modelo:Punto 1.6

Nome: Vagener Moreira costa

Bloco: 10

Apartamento: 101

1- Cadastro de Veículo:

Placa:BDA5H58

CPF: 123.456.789-09

Modelo: Corolla 2.0

Nome: Jorge Mateus

Bloco:15 A

Apartamento: 503

1- Cadastro de Veículo:

Placa:AVX9F34

CPF: 987.654.321-00

Modelo:Gol gti 2.0

Nome:Roberto carlos Oliveira

Bloco:08

Apartamento:801

-----------------------------------------------------------------------------------------------

1- Cadastro de Veículo:

Placa:AVX9F35

CPF: 111.444.777-35

Modelo: Onix 2.0

Nome:Silvia Maria da Silva Leão

Bloco:15 A

Apartamento:401

Vamos fazer a perte visual com HTML,CSS, JAVASCRIPT

**✅ 1. Objetivo: melhorar a interface do usuário**

Atualmente, o sistema roda no terminal (modo texto). Você pode criar um **painel visual web** com:

* 📋 Formulários para cadastro de veículos e funcionários
* 🟢 Visualização das vagas ocupadas/livres
* 🕐 Relatórios de entrada/saída com datas
* 🚨 Alertas de tempo excedido
* 🔒 Login para supervisores

**🔁 2. Como conectar HTML/CSS/JS com seu código Python?**

**📌 Opção 1 – Flask (Recomendada e simples)**

Você transforma seu sistema em um mini servidor web:

* HTML e CSS criam a interface visual.
* JavaScript pode fazer requisições AJAX.
* Flask roda em Python e responde os dados (POST/GET).
* Tudo roda localmente (ou pode ser publicado em Render, PythonAnywhere, etc).

**Exemplo de fluxo:**

1. HTML exibe o botão “Estacionar veículo”
2. JavaScript envia os dados via fetch() para o Flask (POST)
3. Flask chama estacionar\_veiculo(...)
4. Retorna uma resposta (ex: “Veículo estacionado com sucesso”)
5. A tela atualiza com o status da vaga

**📌 Opção 2 – Interface Gráfica com Electron.js (avançado)**

Se quiser um aplicativo de desktop com interface web, pode usar:

* **Electron.js** com HTML/CSS/JS para a interface
* Python como backend usando sockets ou chamadas subprocess

**ETAPA 1 – Preparar o ambiente com Flask (backend em Python)**

**Objetivo:**

Criar um servidor web simples que possa receber requisições da interface (HTML/JS) e interagir com o seu código atual.

Contruido até agora

| **Item** | **Status** |
| --- | --- |
| Servidor Flask ativo (app.py) | ✅ OK |
| Rota /estacionar funcional | ✅ OK |
| HTML com formulário de cadastro | ✅ OK |
| Comunicação via fetch() (JavaScript) | ✅ OK |
| Estilo visual básico | ⚠️ Precisa melhorar |

Você está no caminho certo! Se quiser, posso gerar:

* ✅ A versão atualizada com <meta>, <label> e melhorias visuais
* ✅ Um style.css otimizado para esse HTML
* ✅ Componentização em páginas separadas

Atualização 22/06/2025.  
Corrigir o bug da hora no aquivo estacionamento.py

Data 04/07/2025

Melhora do código:

A parte “estacionar veículos”

Corrigir. pedir o ‘A PLACA’. E “MATRICULA DO FUNCIONARIO”

E retornar as informações do proprietário. COM “ CPF, NOME,BLOCO,APARTAMENTO e HORA DE ENTRADA’

A parte” liberar o veiculo:”

Corrigir pedir “Placa do veiculo.”  
“Matricula do funcionário.”

E retornar as informações do proprietário. COM “ CPF, NOME,BLOCO,APARTAMENTO e HORA DE SAIDA’

04/07/2025

No momento em que apertar o botão limpar os campos de escrita e ficar só as informações do proprietário.

Horário de entrada e saída: Corrija para o horário de Brasília.

04/07/2025

Vamos criar históricos para área supervisor e funcionário.

No momento em que apertar os botões limpar a área de escrita.

Area do Supervisor:

Login confirmado com sucesso!

Nome do Supervisor.

Abrir outro paguina html com as funcionalidades:

Cadastro de funcionário:

Funcionários cadastrados:

Funcionários Logados

Históricos dos serviços por matriculas.

e informar que o login foi bem sucedido e o nome do colaborador

No campo “funcionários logados” mostrar quem está logado, “nome Completo”.

No “histórico dos Serviços por Matricula.” Mostrar todas as atividades de entrada, saída, verificação do status das vaga.

Quando a área ‘área do Supervisor ’ login confirmado com sucesso!

Limpar a área de escrita e passar para o próxima pagina.

E retirar ‘Cadastrar Funcionário’ e ‘Funcionários Cadastrados’ desta pagina.

COM 8 MINUTOS NOTIFIQUE  
ATENÇÃO QUE O TEMPO NA VAGA “X”

ESTA ACABANDO!!!

E COM 10.  
POR FAVOR RETIRE O SEU VEICULO, OBRIGADO PELA COMPREESÃO!  
EXIBA -TODAS AS INFORMAÇOES DO CADASTRO.  
E NO FINAL “O CONDOMINIO AGRADEÇE!!!”

*caixa do "TEMPO EXCEDIDO!" (#notificacoesVagas)*

a caixa de base (#notificacaoBase)

Projeto estacionamento rotativo...  
16/07/2025..

Atualização do projeto estacionamento rotativo....

Linguagem “Backand python”, “front and Java script , css, html, e linguagens sugeridas”

Criar uma área de gestão.

Função do supervisor!

Cadastrar funcionários, com

Histórico de ação funcionários