

**UNIVERSIDADE REGIONAL INTEGRADA DO ALTO URUGUAI E DAS MISSÕES
(URI)**

Vitor de Mattos, Fernando Lirio

OFICINA MECÂNICA

SANTIAGO

2022

SUMÁRIO

1. INTRODUÇÃO	3
2 ESCOPO	3
2.1 Cadastro do cliente	3
2.2 Atualização do cliente	3
2.2 Pesquisa do cliente	4
2.3 Agendamento do cliente	4
3 MODELAGEM CONCEITUAL	4
4. DESENVOLVIMENTO DO SISTEMA	9
5. CONCLUSÃO	12

1. INTRODUÇÃO

O projeto oficina mecânica tem como objetivo, ajudar os mecânicos em geral administrar seu trabalho, esse auxílio é para deixar sua organização de trabalho mais precisa, onde o mecânico vai conseguir ver os carros que estão cadastrado em seu sistema, os trabalhos que ele fez em carros anteriores, neste projeto usaremos as seguintes ferramenta e, Python, django, HTML, CSS e JavaScript,bootstrap essas ferramentas serão usada para modelar e esquematizar todo o Projeto Mecânico.

2 ESCOPO

O projeto tem como objetivo realizar quatro operações simples em um sistema de uma oficina mecânica como o cadastro do cliente, atualização do cliente, pesquisa do cliente e por fim agendamento do cliente.

2.1 Cadastro do cliente

O objetivo do cadastro é fazer o cadastro do cliente com seus dados mais comuns como nome,sobrenome,email,cpf e dados do carro como nome,placa e ano.

No caso dos atributos eles serão tratados de forma que não tenham erros na programação como o cpf que não deixará existir clientes com o mesmo cpf ou até mesmo contas de clientes

Sem dados o sistema terá campos obrigatórios para se preencher.

2.2 Atualização do cliente

Na atualização do cliente ele pode ser atualizado com qualquer um dos dados já estabelecidos por exemplo: nome do cliente, email do cliente, cpf do cliente e por fim exclusão do cliente, entretanto os dados já existentes não poderão ser cadastrados novamente.

2.2 Pesquisa do cliente

Na pesquisa do cliente vai existir um comando que irá percorrer uma lista dos clientes cadastrados no sistema.

2.3 Agendamento do cliente

No caso do agendamento do cliente, irá ter os agendamentos dos clientes que estão vinculados ao sistema.

3 MODELAGEM CONCEITUAL

Figura 1 - Diagrama da visão geral do sistema.

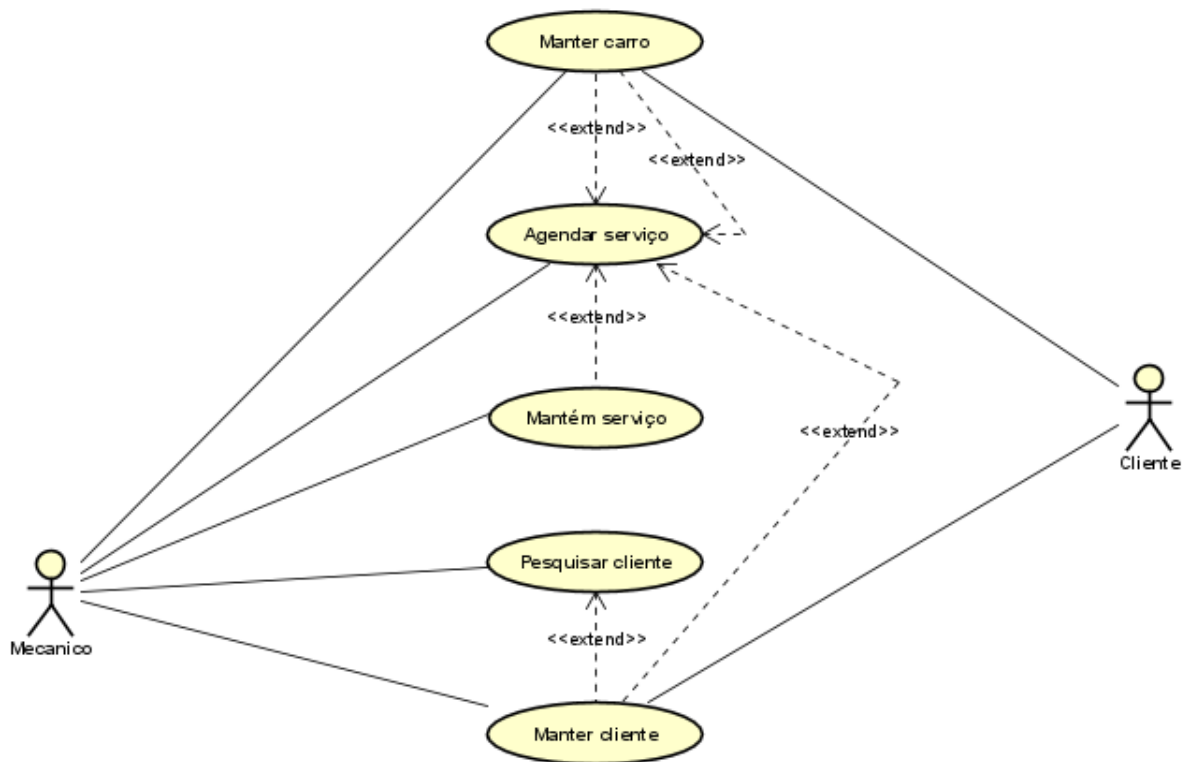


Figura 2 - Diagrama de classes.

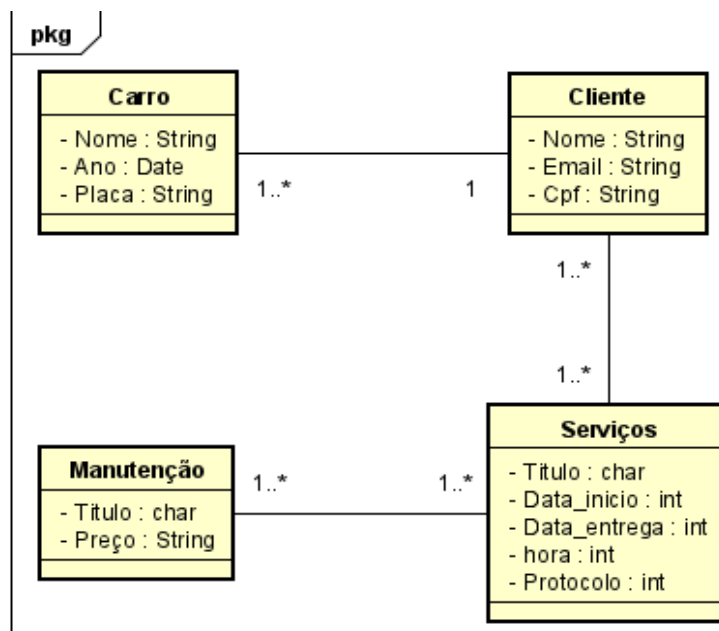


Figura 3 - Diagrama de atividade de adicionar cliente.

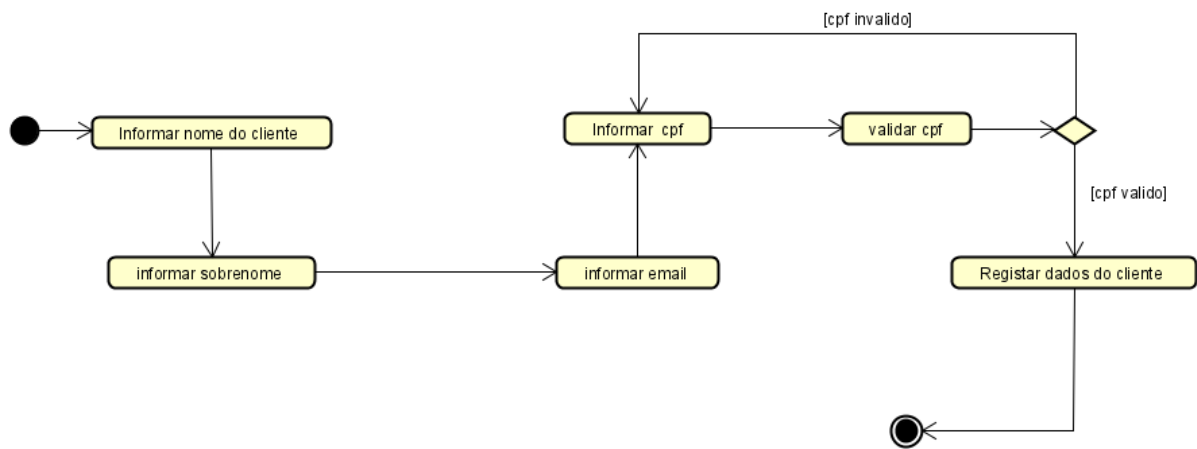


Figura 4 - Diagrama de atividade de adicionar carro.

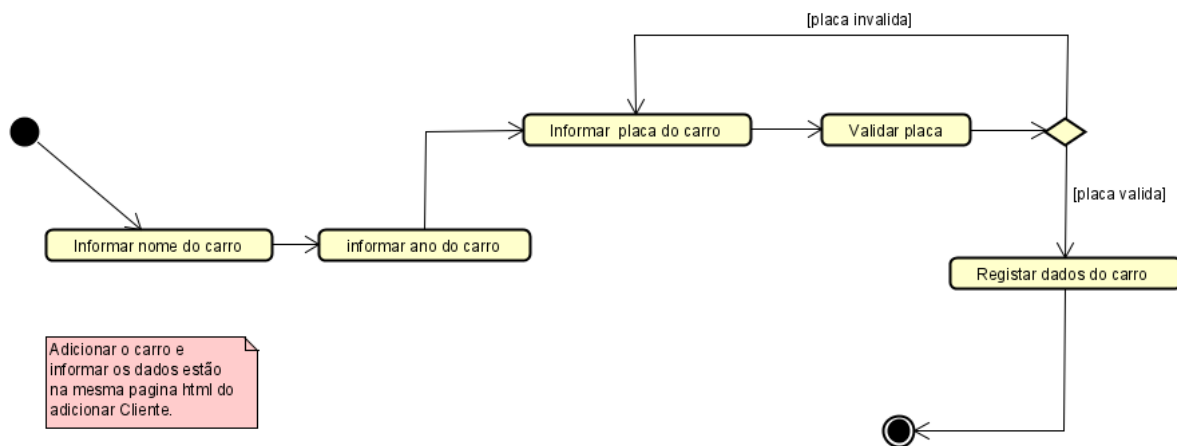


Figura 5 - Diagrama de atividade de adicionar serviço.

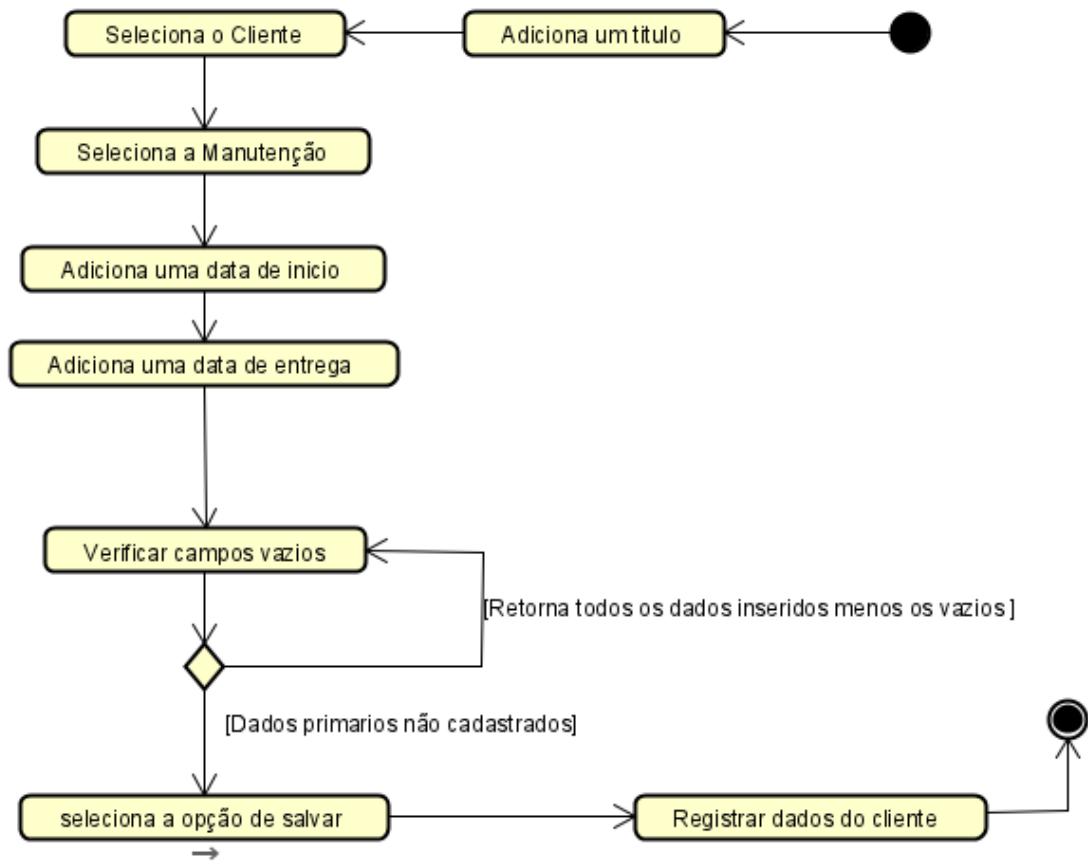


Figura 6 - Diagrama para selecionar histórico da agenda.

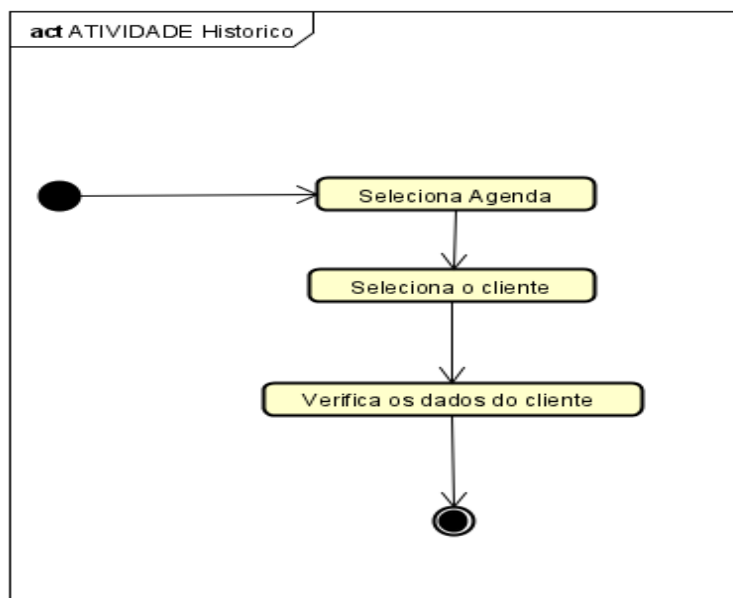


Figura 7 - Diagrama de atividade de atualizar cliente.

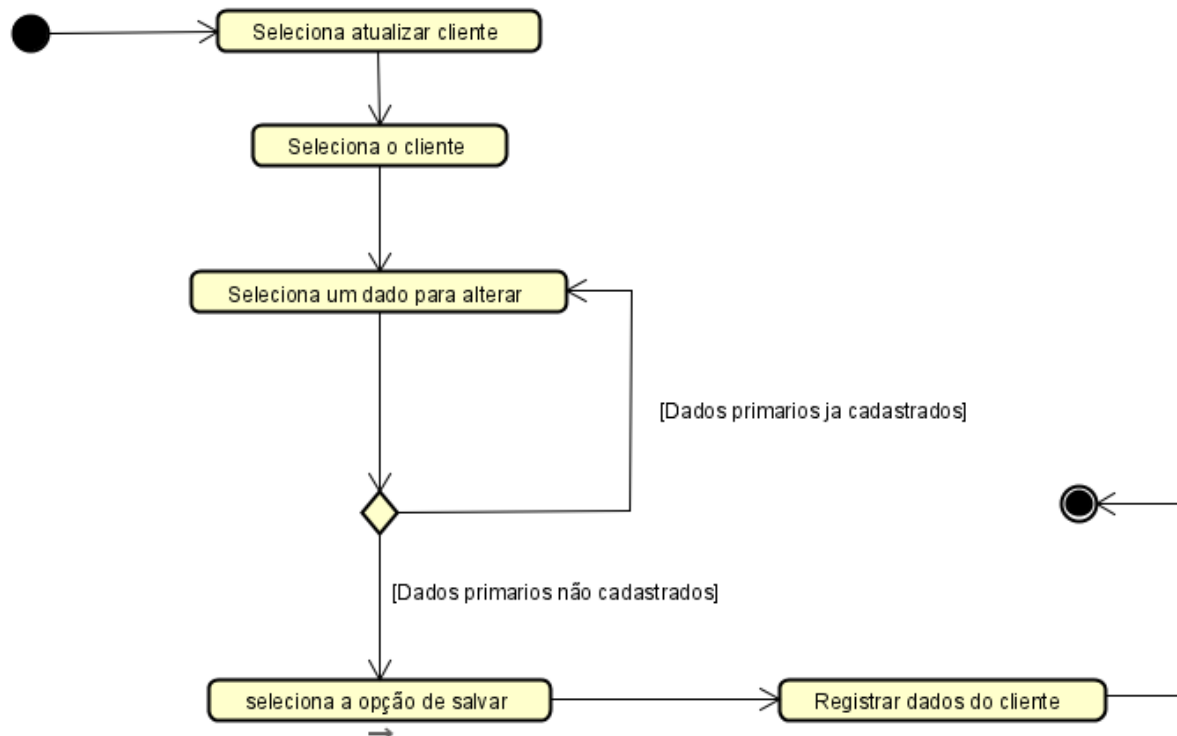
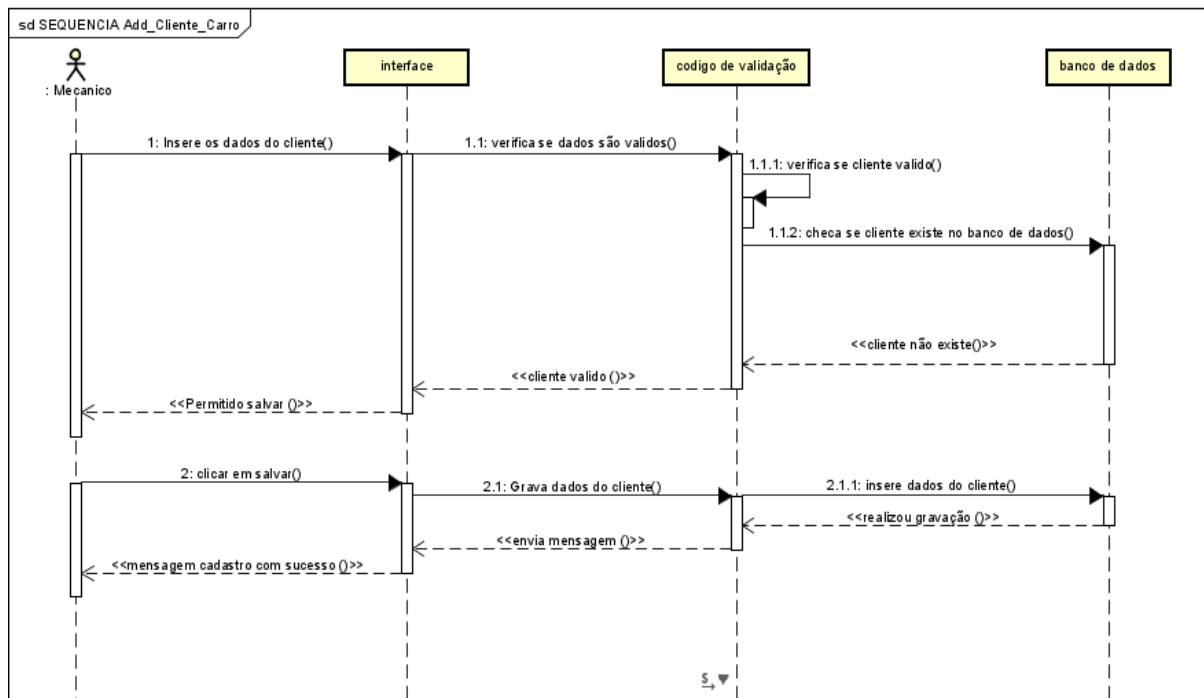
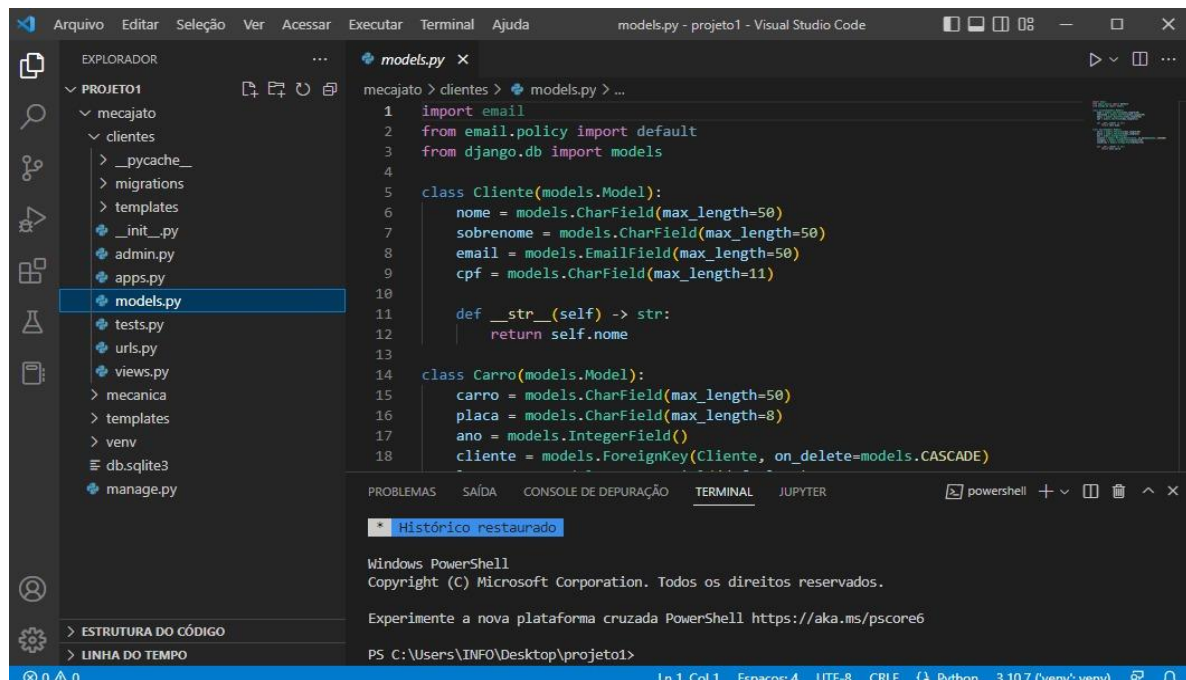


Figura 8 - Diagrama de sequência de adicionar cliente.



4. DESENVOLVIMENTO DO SISTEMA

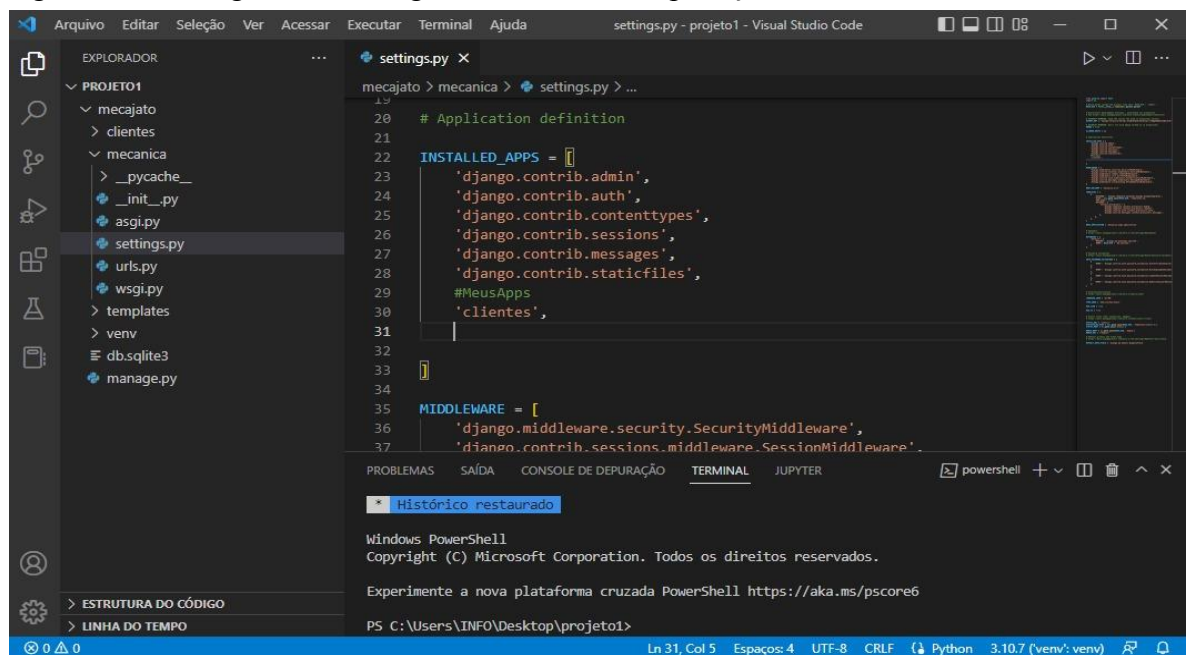
Figura 10 - visão geral dos códigos do sistema do cliente e do carro.



```
mecajato > clientes > models.py > ...
1 import email
2 from email.policy import default
3 from django.db import models
4
5 class Cliente(models.Model):
6     nome = models.CharField(max_length=50)
7     sobrenome = models.CharField(max_length=50)
8     email = models.EmailField(max_length=50)
9     cpf = models.CharField(max_length=11)
10
11     def __str__(self) -> str:
12         return self.nome
13
14 class Carro(models.Model):
15     carro = models.CharField(max_length=50)
16     placa = models.CharField(max_length=8)
17     ano = models.IntegerField()
18     cliente = models.ForeignKey(Cliente, on_delete=models.CASCADE)
```

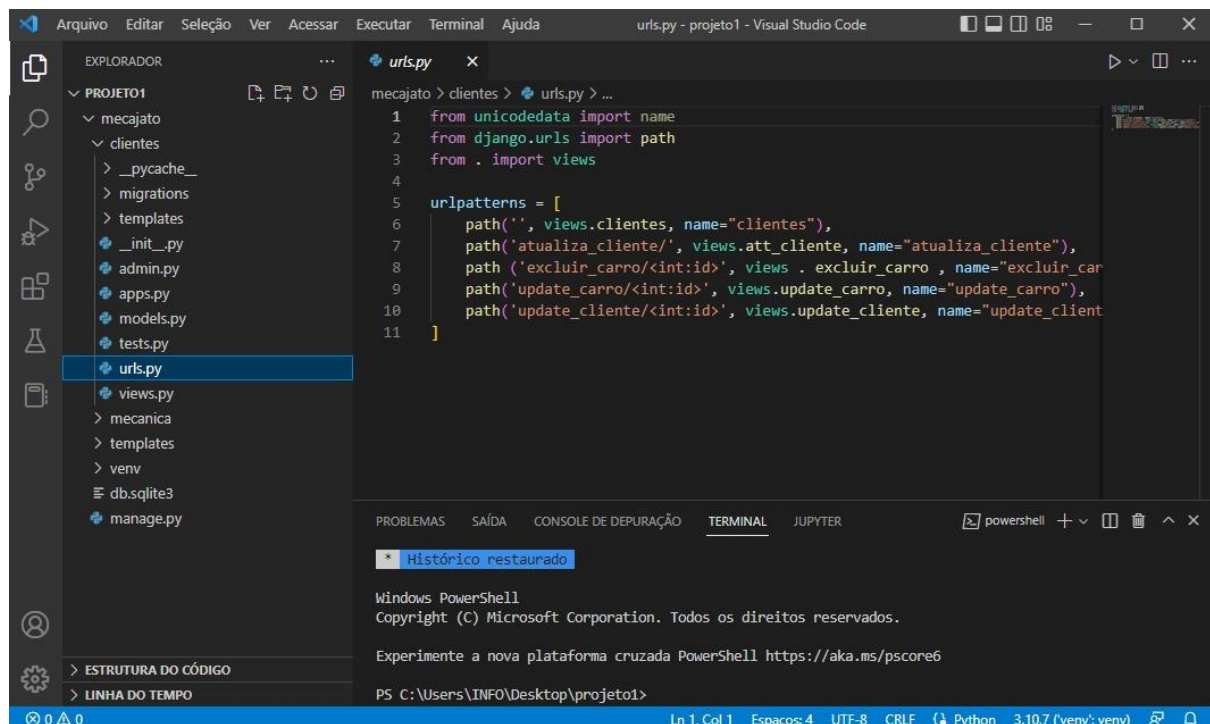
Na figura 10 mostra-se a criação das classes do cliente e do carro onde onde pode ser ver que existem limites dos campos como nome,sobrenome,email e cpf do cliente do carro onde seus campos são carro placa ano e cliente pode se notar que o cliente está em cascade que o sistema de cascata que quando um cliente for excluído todos os carros daquele certo cliente também vão ser excluídos.

Figura 11 - visão geral dos códigos do sistema das aplicações usadas no sistema.



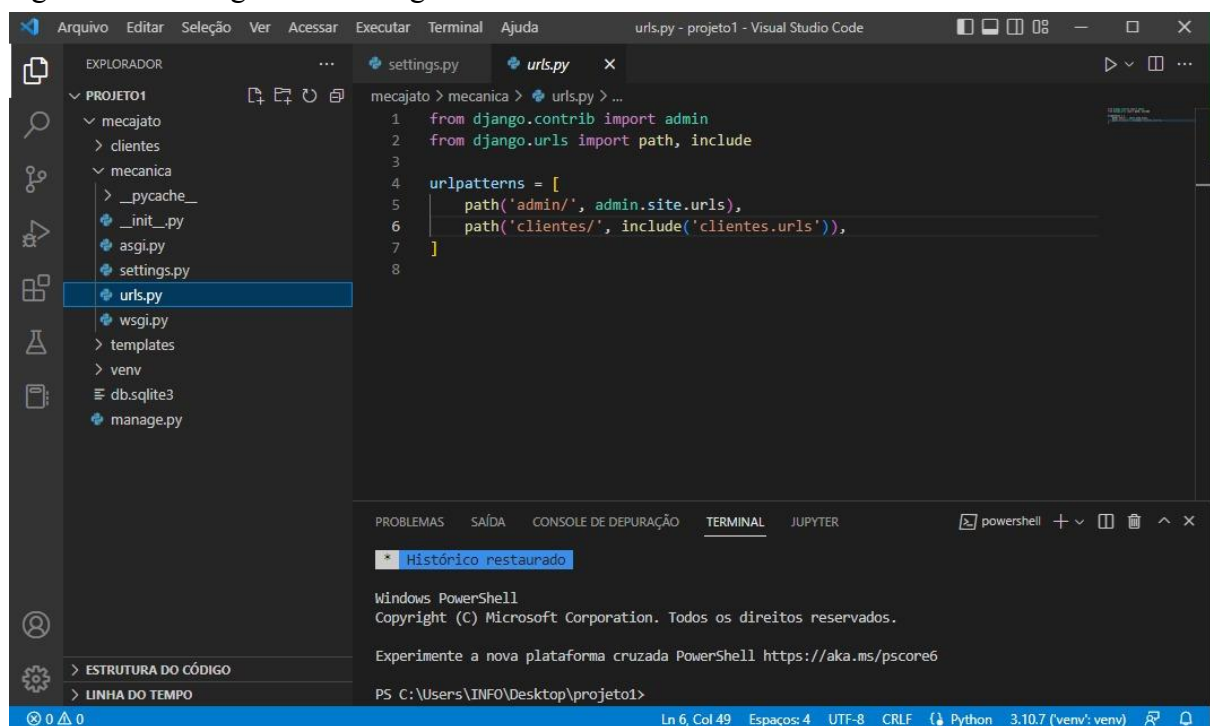
```
mecajato > mecanica > settings.py > ...
19
20 # Application definition
21
22 INSTALLED_APPS = [
23     'django.contrib.admin',
24     'django.contrib.auth',
25     'django.contrib.contenttypes',
26     'django.contrib.sessions',
27     'django.contrib.messages',
28     'django.contrib.staticfiles',
29     #MeusApps
30     'clientes',
31 ]
32
33
34
35 MIDDLEWARE = [
36     'django.middleware.security.SecurityMiddleware',
37     'django.contrib.sessions.middleware.SessionMiddleware',
38 ]
```

Figura 12 - visão geral das urls que estão funcionando no sistema.



```
1 from unicodedata import name
2 from django.urls import path
3 from . import views
4
5 urlpatterns = [
6     path('', views.clientes, name="clientes"),
7     path('atualiza_cliente/', views.att_cliente, name="atualiza_cliente"),
8     path('excluir_carro/<int:id>', views . excluir_carro , name="excluir_carro"),
9     path('update_carro/<int:id>', views.update_carro, name="update_carro"),
10    path('update_cliente/<int:id>', views.update_cliente, name="update_cliente")
11 ]
```

Figura 13 - visão geral dos códigos do sistema do cliente e do carro.



```
1 from django.contrib import admin
2 from django.urls import path, include
3
4 urlpatterns = [
5     path('admin/', admin.site.urls),
6     path('clientes/', include('clientes.urls')),
7 ]
8
```

Na figura 12 e 13 encontram-se as urls das páginas html que vão ser usadas no sistema web usando as funcionalidades.

Figura 14 - visão geral do front end.

A imagem mostra a interface de usuário de um sistema web. À esquerda, há um menu lateral azul escuro com o título "OFICINA" e ícones para "Página inicial", "Clientes", "Serviços" e "Agenda". O conteúdo principal tem um fundo cinza escuro. No topo, há dois botões verdes: "Adicionar clientes" e "Atualizar clientes". Abaixo, há campos de formulário brancos para "Nome:" (com subcampos "Primeiro nome" e "Sobrenome"), "E-mail:" (contendo "nome@email.com") e "CPF:". Abaixo dos campos, há uma seção "Carros" com um botão amarelo "Adicionar um carro" e um botão azul "Cadastrar".

O front end foi desenvolvido para ser simples e como atualmente em 2022 estamos vivendo a copa do mundo resolvemos colocar as cores da bandeira nacional para uma melhor representação no sistema.

5. CONCLUSÃO

O objetivo do projeto era realizar um sistema não tão simples porém não tão complexo, visando nisso o grupo decidiu realizar esse projeto com o intuito de auxiliar profissionais de mecânica.

Em todo o projeto foi pretendido que o sistema teria êxito em todas as suas funções em um determinado período de tempo então assim foi realizado os casos de uso que possuíram êxito em sua função de auxiliar a estrutura do sistema, e assim se deu a início a programação do mesmo, porém ao decorrer do projeto foram encontrados alguns problemas inesperados como bugs no código fazendo assim que o sistema não funcionasse como deveria, entre outros como erros no compilador do sistema.

entretanto mesmo com o sistema apresentando problemas foi realizado com êxito o cadastro do cliente com seu carro e a atualização do mesmo, onde os dados em questão são armazenado de maneira correta e respeitando os casos de uso e diagramas que originaram o sistema, em suma a aba de serviço e agenda não foram terminadas devido a esses contratempos inesperados no desenvolvimento do sistema.

Em virtude de tudo que foi mencionado no sistema e mesmo que ao decorrer do projeto não tenha sido completamente concluído por essas razões, é notório que a experiência adquirida com a construção dos diagramas uml e a programação usando as tecnologias apresentadas no projeto durante o período de tempo predeterminado irá auxiliar os membros do projeto em seus futuros desenvolvimentos.