

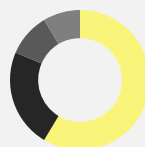
Actualízate

Potencia tu negocio con soluciones digitales para taxistas.



Ver Video Demostrativo

Impulsa tu negocio de taxis con TaxiApp:
¡Descubre cómo en nuestro video demo!.



Menos errores en los cobros



Información organizada



Mas precisión en los cálculos

TaxiApp

Tu solución de tarifas inteligentes para taxis

[Taxi App en GitHub](#)



¡Explora el corazón de nuestro proyecto! Te invitamos a revisar nuestro repositorio en GitHub.

100 % Open source

Únete a nuestra comunidad de desarrolladores y juntos impulsemos el futuro de la industria de los taxis.



Por qué nuestro código

En nuestro proyecto, aplicamos buenas prácticas de desarrollo para crear una aplicación sólida y escalable: planificación rigurosa, modularidad, código limpio, pruebas unitarias y control de versiones.



Sube al asiento del conductor y navega por nuestro tablero Kanban! Descubre cómo organizamos y gestionamos el desarrollo de nuestro proyecto

Buenas Practicas



Escalabilidad

Código Limpio

“ Control de versiones ”

Modularidad

```
1  from Controllers.Fees import Fees
2  from config_logger import setup_logger
3
4  logger = setup_logger()
5
6
7  # The ControlTravel class manages the fees for a travel and saves the travel history.
8  class ControlTravel:
9      def __init__(self):
10         self.send_fee = Fees()
11         self.is_move = False
12         self.is_new_travel = False
13         self.counter = 0
14
15     def toggle_fee(self):
16         """
17         The function toggles between stopping and starting a fee based on whether or not there is
18         movement.
19         """
20         if self.is_move:
21             self.send_fee.fee_stoped()
22             logger.info("Taxi stoped")
23         else:
24             self.send_fee.fee_movement()
25             logger.info("Taxi in movement")
26
27     def new_travel_fee(self):
28         """
29         The following function checks if there is a new trip and saves the history after the trip has finished.
30         """
31         if not self.is_new_travel:
32             self.send_fee.init_end_move()
33             self.is_new_travel = True
34             logger.info("Trip started")
35         else:
36             data = self.send_fee.end_travel()
37             json_data = data.data_to_json()
38             data.save_history_to_mongo(json_data)
39             logger.info("Trip ended")
40             return False
41
```