PASSOS PARA O DESENVOLVIMENTO DO MEU CÓDIGO  
  
\*\*1. Configuração do Ambiente:\*\*

* Instalei um ambiente Python e a escolhi a framework Flask para construir a aplicação.

```bash

pip install Flask

```

\*\*2. Modelagem de Dados:\*\*

* Instalação de banco de dados para armazenar informações sobre veículos, usuários, reservas e histórico de empréstimos: com SQL Alchemy

```python

from flask\_sqlalchemy import SQLAlchemy

db = SQLAlchemy()

class Vehicle(db.Model):

id = db.Column(db.Integer, primary\_key=True)

name = db.Column(db.String(100), nullable=False)

# Poderiam ser adicionados mais campos para características do veículo

class User(db.Model):

id = db.Column(db.Integer, primary\_key=True)

name = db.Column(db.String(100), nullable=False)

contact = db.Column(db.String(20), nullable=False)

driver\_license = db.Column(db.String(20), nullable=False)

class Reservation(db.Model):

id = db.Column(db.Integer, primary\_key=True)

user\_id = db.Column(db.Integer, db.ForeignKey('user.id'), nullable=False)

vehicle\_id = db.Column(db.Integer, db.ForeignKey('vehicle.id'), nullable=False)

start\_date = db.Column(db.DateTime, nullable=False)

end\_date = db.Column(db.DateTime, nullable=False)

# Poderia adicionar mais campos conforme necessário

\*\*3. Criação das Rotas:\*\*

* Depois criei as rotas da aplicação, em Flask, para cumprir os requisitos mencionados (com o nome: RollingSoftware):

```python

from flask import Flask, request, jsonify

app = Flask(\_\_name\_\_)

@app.route('/vehicles', methods=['GET'])

def get\_available\_vehicles():

# Permite consultar os veículos disponíveis e mostrar como JSON

# Implementa a lógica para verificar disponibilidade de veículos

return jsonify({'vehicles': vehicles})

@app.route('/reserve', methods=['POST'])

def reserve\_vehicle():

# Recebe dados da reserva do usuário e armazena no banco de dados

# Implementa a lógica de validação e armazenamento

return jsonify({'message': 'Reservation successful'})

* Poderia adicionar mais rotas conforme necessário para a funcionalidade de gerenciamento da frota

if \_\_name\_\_ == '\_\_main\_\_':

app.run(debug=True)

```

\*\*4. Autenticação e Autorização:\*\*

* Implementei um sistema de autenticação e autorização para proteger as rotas registadas, com o Flask-Login.

\*\*5. Interface do Utilizador:\*\*

* Desenvolvi uma interface do utilizador simples e intuitiva, para que pudessem interagir facilmente com a App, através de uma framework front-end React para comunicar com o back-end Flask via API: não cheguei a conseguir concluir e fiz parte disso na MIT App Inventor.

\*\*6. Configuração do Banco de Dados:\*\*

* Tentei configurar o banco de dados com o MySQL e integrá-lo na aplicação Flask usando o SQLAlchemy. Deu erro. Tentei instalar o SQL Server como alternativa, mas também não consegui fazê-lo com sucesso e bloqueou-me o PC, obrigando-me a fechar a página do Python onde estava a fazer o trabalho.

\*\*7. Testes e Implantação:\*\*

* Não cheguei a fazer.