



UNIVERSIDADE ESTADUAL DE FEIRA DE SANTANA
DISCIPLINA: TEC 502 MI CONCORRÊNCIA E CONECTIVIDADE
DISCENTES: JEFERSON ALMEIDA E ROGÉRIO CERQUEIRA.

RELATÓRIO TÉCNICO: SISTEMA DISTRIBUÍDO DE RECARGA DE VEÍCULOS ELÉTRICOS

Feira de Santana, BA

2025

1. INTRODUÇÃO

Este relatório apresenta a análise técnica do sistema distribuído de recarga de veículos elétricos, composto por três entidades principais: clientes (veículos), servidor central e pontos de recarga (charge-points). O sistema foi desenvolvido com Python, utilizando sockets TCP para comunicação entre as entidades, e containerizado com Docker para facilitar a distribuição e execução.

2. ESTRUTURA DO PROJETO

- **Diretórios principais:**




- `client/` - Contém o código do cliente (veículo).
- `server/` - Contém o código do servidor.
- `charge-point/` - Contém o código dos pontos de recarga.
- `services/` - Serviços auxiliares (monitoramento, cadastro, reservas).
- `Dockerfile-*` - Dockerfiles separados para cada componente.
- `docker-compose.yml` - Orquestra os containers.

- **Tecnologias utilizadas:**

- Python 3.12.3
- Sockets TCP/IP
- Docker + Docker Compose

- Estrutura modular por serviço

3. FUNCIONALIDADES IMPLEMENTADAS

- Envio automático de posição do cliente a cada 60 segundos
- Cálculo do posto de recarga mais próximo baseado na posição do cliente
- Sistema de reserva de posto com:
 - Cliente, posto, data e horário
 - Verificação de conflitos de agendamento
 - Status dinâmico:  Disponível,  Ocupado,  Livre em X min
- Registro de logs em `log.txt`
- Simulação do carregamento no charge-point com tempo fixo (3s)
- Scripts automatizados de apresentação com `bash`

4. SUCESSOS

- Sistema completamente funcional em containers
- Testes com múltiplos clientes e pontos de recarga simultâneos
- Boa separação de responsabilidades por componente

- Interface de comunicação simples e robusta
- Flexibilidade para expansão futura (API REST, banco de dados, dashboard)

5. FALHAS E PROBLEMAS ENCONTRADOS

- Uso misto de `reservas.txt` e `reservas.json`, gerando redundância
- O servidor inicialmente não tratava corretamente o comando `COMPLETE:` enviado pelo charge-point
- Faltava validação de conflito no agendamento de reservas (corrigido posteriormente)
- Mensagem do servidor para o cliente nem sempre padronizada, dificultando a interpretação do status
- Necessidade de corrigir divisor incorreto ao calcular minutos restantes no cliente
- Cliente sem opção de interação manual inicial (corrigido via scripts de simulação)

6. SUGESTÕES DE MELHORIA

- Consolidar o armazenamento de reservas em um único formato (JSON)
- Implementar dashboard web com Flask para monitoramento em tempo real

- Adicionar sistema de autenticação para clientes
- Melhorar logs com identificação por ID de cliente
- Criar testes automatizados (pytest)

7. CONCLUSÃO

O sistema apresenta um bom grau de maturidade e está apto para demonstrações funcionais e testes em simulações reais. As funcionalidades principais estão implementadas de forma incompleta, e a arquitetura atual permite expansões futuras com facilidade. No entanto, algumas exigências e funcionalidades não puderam ser completamente atendidas devido a fatores pessoais, limitações técnicas e de tempo. Ainda assim, o projeto se mostra uma base sólida e promissora para evolução e aplicação prática em contextos reais de recarga inteligente de veículos elétricos.

8. GITHUB

<https://github.com/rogeriocerqueira/mi-concorrenca-e-conectividade.git>