Politécnico Castelo Branco
Escola Superior de Tecnologia

EST – IPCB

Licenciatura em Engenharia Informática

Aplicação Distribuídas

Trabalho de Desenvolvimento Aplicacional

Plataforma de partilha de viagens (Car Pooling) e partilha de veículos (Car Sharing)

3º Ano / 1º Semestre – 2025/2026

1. OBJETIVOS

Pretende-se com este trabalho prático desafiar os estudantes utilizando as tecnologias estudadas na UC (Spring, Spring Boot e Spring Cloud, Docker e outras) a conceberem e desenvolverem uma aplicação segura com uma interface Web e baseada no padrão arquitetural micro serviços.

O trabalho pode ser realizado e apresentado em grupo entre 3 a 5 estudantes. Sugerindo-se que cada grupo funcione como uma pequena equipa com clara divisão de tarefas.

De seguida, apresenta-se o problema e requisitos chave da aplicação.

Prazo estimado de conclusão é de cerca de 1 mês e duas semanas (ver em baixo a seção prazo e forma de entrega).

2. Problema e Requisitos

2.1 Ideia do Projeto: Pretende-se que seja desenvolvida uma aplicação modular, baseada em **microserviços**, que **simule** o funcionamento de uma **plataforma de partilha de viagens (Car Pooling) e partilha de veículos (Car Sharing)**. O objetivo é promover a mobilidade sustentável e a redução da pegada ecológica, incentivando os utilizadores a **partilhar deslocações** ou a **utilizar veículos comuns**, reduzindo custos e emissões poluentes.

O grupo de estundantes deve optar por uma de entre as duas vertentes previstas:

- Na vertente de **Car Pooling**, os utilizadores podem oferecer ou procurar boleias, partilhando trajetos semelhantes e dividir as despesas da viagem.
- Na vertente de **Car Sharing**, a aplicação permitirá procurar veículos disponíveis nas redondezas para aluguer de curta duração (taxado por hora), sem necessidade de condutor, será próprio utilizador desde que habilitado, simulando o funcionamento de plataformas de partilha de veículos.

O sistema deve permitir o **registo de utilizadores**, a **criação e pesquisa de viagens**, a **reserva e gestão de boleias**, a **simulação de custos partilhados**, e a **monitorização da utilização de veículos** através de integração com **GPS**.

Como referência, poderão ser consultadas plataformas reais como **BlaBlaCar** ou **Bolt Drive**, que apresentam conceitos semelhantes de partilha de viagens e veículos, disponíveis em aplicações Android e iOS; embora no presente projeto não se preveja a implementação de um

aplicação móvel, espera-se que o *front-end* seja baseado numa vista Web responsiva (e.g., baseado no Bootstrap) para melhor demonstração.

2.2 Descrição do Sistema: A aplicação simula uma rede de mobilidade colaborativa, integrando os seguintes **componentes funcionais** (não limitados a):

1. **Gestão de Utilizadores e Perfis:**
    - Registo e autenticação de utilizadores.
    - Definição de perfis (passageiro, condutor, ou ambos).
    - Atualização de dados pessoais, preferências e avaliações (rating).
2. **Gestão de Viagens (Car Pooling):**
    - Criação de viagens por condutores, com origem, destino, hora e número de lugares disponíveis.
    - Pesquisa de viagens compatíveis por passageiros.
    - Reserva e confirmação de boleias.
    - Cálculo automático da divisão de custos de combustível ou de energia e portagens.
    - Sugere-se que haja um microserviço de registo dos veículos do condutor para efeitos de escolha.
3. **Gestão de Veículos e Car Sharing:**
    - Registo de veículos disponíveis para partilha (por particulares ou entidades).
    - Consulta e reserva de veículos numa área geográfica.
    - Simulação de alugueres temporários (por hora).
    - Cálculo do custo total de utilização (€/hora, km percorridos, taxas adicionais).
4. **Localização e Integração com GPS:**
    - Geolocalização de viagens e veículos disponíveis.
    - Sugestão de boleias próximas da posição atual do utilizador.
    - Cálculo de rotas otimizadas entre origem e destino.
5. **Gestão de Custos e Pagamentos:**
    - Simulação e divisão de despesas de combustível/energia e aluguer.
    - Possibilidade de cálculo automático do valor a pagar por passageiro.
    - Registo de histórico de pagamentos e custos por viagem.
6. **Administração e Moderação:**
    - Gestão global do sistema e das contas de utilizadores.
    - Visualização e bloqueio de utilizadores em caso de abuso.

Estes componentes funcionais podem ser adaptados e certos aspectos que exijam hw ou tempo real sejam simulados.

Exemplo de casos de usos (Caso de Sucesso):

1. **Criação e Procura de Viagens:**
    - O condutor "João" acede à aplicação e cria uma viagem de Castelo Branco → Lisboa para as 08:00h, com 3 lugares disponíveis.
    - A passageira "Maria" procura boleias para a mesma rota e encontra a viagem de João.
    - Maria envia um pedido de reserva, e João confirma a partilha.

2. **Gestão da Viagem e Custos:**
    - Após a viagem, o sistema calcula o custo total (por exemplo, 30 € em combustível e portagens) e divide automaticamente pelos ocupantes.
    - Cada passageiro paga 10 €, e o sistema regista o histórico da viagem e dos participantes.

3. **Car Sharing:**
    - O utilizador "Carlos" procura veículos disponíveis perto da sua localização.
    - O sistema lista um veículo elétrico disponível a 2 €/hora.
    - Carlos reserva o veículo por 3 horas, totalizando 6 €.
    - Após o uso, o veículo é devolvido e o sistema regista a sessão de aluguer.

4. **Administração:**
    - O administrador consulta relatórios com o número de viagens partilhadas, total de utilizadores ativos e idealmente o CO_2 poupado com base nas emissões do veiculo.
    - Pode ainda bloquear contas fraudulentas ou remover veículos inativos.

2.3. Principais Microserviços: O sistema deve ser implementado com base em microserviços independentes, cada um responsável por uma funcionalidade específica e comunicando entre si por APIs REST. Sugerem-se os seguintes micro serviços (não limitado, o grupo de estudantes pode decidir por outros):

- **Microserviço de Autenticação e Perfis**
    - Registo e autenticação de utilizadores (Spring Security).
    - Gestão de papéis (Admin, Condutor, Passageiro, Gestor de Frota).
    - Atualização de perfis e avaliação entre utilizadores.
- **Microserviço de Viagens (vertente Car Pooling)**
    - Criação, pesquisa e reserva de viagens.
    - Gestão de estados das viagens (aberta, em curso, concluída).
    - Cálculo e partilha dos custos entre os participantes.

- **Microserviço de Veículos (vertente Car Sharing)**
    - Registo e gestão de veículos disponíveis.
    - Cálculo de custos com base em tempo e distância.
    - Atualização do estado do veículo (disponível, em uso, manutenção).
- **Microserviço de Localização / GPS**
    - Gestão de coordenadas e rotas entre origem e destino.
    - Eventual Integração com APIs de mapas (Google Maps, OpenStreetMap, Mapbox).
    - Determinação de proximidade entre utilizadores e veículos.
- **Microserviço de Pagamentos e Custos**
    - Processamento de custos das viagens ou alugueres.
    - Registo e simulação de pagamentos e partilhas de despesa.
- **Microserviço de Notificações (Opcional)**
    - Envio de notificações por email ou push (ex: confirmação de reserva, cancelamento).
    - Alertas de início ou fim de viagem.
- **Microserviço de Administração**
    - Gestão global da aplicação.
    - Monitorização de serviços, utilizadores e métricas ambientais.
    - Bloqueio e ativação de contas.
- **Microserviço de Front-End**
    - Interface unificada para todos os tipos de utilizadores.
    - Acesso às funcionalidades de registo, procura e reserva de viagens/veículos.
    - Visualização de histórico, relatórios e estatísticas.
- **Microserviços de Descoberta de Microserviços e API Gateway.**
    - Sugere-se na implementação seguir uma arquitectura em que o API Gateway está depois do *front-end* para não ser sobrecarregado com o carregamento do html, css, thymleaf, imagens, etc.

4. Tipos de Utilizadores e Permissões

O sistema deve suportar **múltiplos perfis de utilizadores** com diferentes níveis de acesso:

1. **Utilizador Normal (Passageiro/Condutor):**
    - Pode criar, procurar e reservar viagens.
    - Pode consultar histórico e avaliações.
2. **Gestor de Frota (Car Sharing):**
    - Pode registar e gerir veículos partilhados.
    - Acompanhar o estado e uso dos veículos.

3. **Administrador:**
    - Pode gerir todas as contas e atividades.
    - Pode suspender utilizadores ou remover entidades abusivas.

O grupo de trabalho pode definir outros tipos de utilizadores e decidir sobre os privilégios a atribuir a cada utilizador.

2. Outros Requisitos não funcionais

A aplicação deve ser escalável e ser passível de ser implementada num ambiente Cloud, devendo para o efeito ser adotada uma arquitetura baseada em microserviços. Vários microserviços com diferentes responsabilidades devem compor a aplicação (conforme indicado antes).

Qualquer instância criada deve ser registada no micro serviço de registo e localização de micro serviços Spring Eureka.

O ponto de entrada da aplicação é através de um API Cloud Gateway, o qual deve permitir o encaminhamento dinâmico apenas para uma das instâncias do front-end. Refira-se da possibilidade de este poder interagir negativamente com o Spring Cloud Security aquando da *dockerização* (nesse caso adotar uma solução em que o front-end tem/usa o seu próprio gateway web). As comunicações entre microserviços, são baseadas no Spring Cloud OpenFeign.

Também poderá ser necessário existirem várias instâncias do mesmo microserviço para suportar o aumento de carga. Deve prever a possibilidade de replicação de microserviços. Uma forma simples de o fazer é prever a containerização dos microserviços e cenários de replicação a partir do *yaml* de ferramenta Docker compose.

Valoriza-se a implementação de circuit breakers.

Valoriza-se prever o build e *deployment* da aplicação de forma automatizada usando um ficheiro yaml **Docker compose**.

2. PRAZO E FORMA DE ENTREGA

Todos os documentos e ficheiros devem ser entregues em formato eletrónico.
Será disponibilizado na plataforma Moodle uma ligação para submissão do trabalho.
As aplicações deverão ser acompanhadas de uma pequena apresentação *powerpoint* ou outra (e.g., keynote), que descreva e apresente a aplicação.

Durante a apresentação deve ser realizada uma demonstração breve.

As datas importantes a observar são:

Data de entrega: 0h10m de 14 de janeiro de 2025 / data exame da respetiva época.
Apresentação: 14 de janeiro de 2025 (horário conforme a turma do grupo)

3. AVALIAÇÃO DO TRABALHO

3.1. Critérios de Avaliação

O trabalho será avaliado de acordo com as seguintes ponderações (total de 20 valores):

- Apresentação Oral e Breve Apresentação (máximo 2v) com uma estrutura semelhante à de um *pitch – a apresentação e demo terá uma duração máxima de 12 minutos*, sendo colocadas no final algumas questões pelo docente:
  - Problema
  - Solução
  - Principais Funcionalidades
  - Características Técnicas
  - Demonstração
  - Vantagens
  - Sumário
- Protótipo Funcional (total 18v):
  - Conceção e Desenvolvimento Base (máximo 14v):
    - Controlador(es) MVC e templates Thymeleaf (3v);
    - Arquitetura baseada microserviços com responsabilidades bem definidas para suporte à aplicação e acesso aos dados incluindo os controladores Rest (4v).
    - Solução adotada, Lógica de Negócio, Simulações (3.5v);
    - Base de Dados ou *Datastore*, e dos repositórios de Dados de suporte à aplicação (2.5v) (valoriza-se se adotar base da dados externa);
    - Incorporação Swagger UI, Actuator e Circuit-breakers (1v)
  - Considerar outros aspetos/funcionalidades relevantes à implementação e *deployment* incluindo (máximo 4v):
    - Mecanismos de Segurança MVC e REST (1.5v);
    - *Containerização* dos microserviços (build e run) (1v).
    - Outros aspetos valorizáveis apresentados pelo grupo (1.5v). Este campo pode valer até 2.5 se acumular a pontuação do ponto da *containerização* (caso não apresente/implemente esta solução). Exemplo: apresentação de coleção de testes postman.

Os elementos para avaliação serão entregues através de link disponibilizado no Moodle para o efeito, e devem incluir:

- Apresentação powerpoint ou outro formato (em PDF).
- Ficheiro .zip com código elaborado.
- Outros elementos relevantes a critério do estudante.

O docente, Prof. Doutor Alexandre Fonte
10 de novembro de 2025