

Programação Orientada aos Objetos**2025/2026****Projeto****Mobilidade UC****Prazo de entrega:** 05/12/2025**Plataforma:** Inforestudante**Cotação:** 5 valores

Nota: A fraude denota uma grave falta de ética e constitui um comportamento não admissível. Situações de fraude serão severamente penalizadas, de acordo com os regulamentos e legislação em vigor.

A Universidade de Coimbra pretende disponibilizar um serviço de Mobilidade para estudantes e funcionários poderem alugar bicicletas, trotinetes elétricas e e-bikes. Os alugueres são faturados com base no tempo de utilização e podem incluir serviços adicionais, como um capacete ou uma luz. Diferentes tipos de utilizadores (estudante, funcionário) beneficiam de regras de preço distintas.

Os veículos de aluguer (bicicleta, trotinete e e-bike) são caracterizados pelo seu identificador (número de 4 dígitos) e a sua localização atual (GPS). Os veículos elétricos (trotinetes e e-bikes), são ainda caracterizados pelo respetivo nível da bateria. As bicicletas podem ser individuais ou para duas pessoas. As e-bikes podem ter bateria fixa ou removível. As trotinetes podem dispor de ecrã LCD com mapa ou não. Os utilizadores (estudante e funcionário) são caracterizados pelo seu número mecanográfico e pelo método de pagamento (cartão de crédito ou referência multibanco). O estudante deve indicar o curso em que está matriculado e o Pólo Universitário que mais frequenta (Pólo I, II ou III). O funcionário deve indicar o ano do contrato, e pode ser Docente ou Não Docente. O funcionário Docente deve indicar uma lista das Faculdades em que leciona. O funcionário Não Docente deve indicar o serviço onde trabalha. Um aluguer associa um utilizador a um veículo, incluindo a data e hora de início e a data e hora de fim do aluguer. Um aluguer pode ainda incluir serviços adicionais.

A tabela seguinte apresenta os custos associados aos alugueres de cada tipo de veículo. Os valores apresentados correspondem ao custo em Euros por hora. Os alugueres diários (superiores a 24 horas) têm um custo diário equivalente a um aluguer de 8 horas. O aluguer de um capacete tem um custo de 5 Euros por dia, e o de uma luz, 2,5 Euros por dia. Os funcionários Não Docentes beneficiam de um desconto de 50% no valor final do aluguer.

Veículo	Estudante	Funcionário
Bicicleta (uma pessoa)	1,00	2,00
Bicicleta (duas pessoas)	2,00	3,00
Trotinete (sem LCD)	1,00	2,50
Trotinete (com LCD)	1,10	2,60
E-Bike (bateria fixa)	1,25	2,75
E-Bike (bateria removível)	1,50	3,00

Funcionalidades da aplicação a desenvolver:

- **Carregar utilizadores:** ao iniciar, a aplicação deve carregar os utilizadores a partir de um ficheiro de texto (utilizadores.txt).
- **Carregar veículos:** ao iniciar, a aplicação deve carregar os veículos a partir de um ficheiro de texto (veículos.txt).
- **Carregar alugueres:** ao iniciar, a aplicação deve carregar os alugueres a partir de um ficheiro de objetos (alugueres.obj), caso exista.
- **Criar aluguer:** a aplicação deve solicitar o número mecanográfico do utilizador, o ID do veículo e os restantes dados do aluguer. No final, deve apresentar um resumo do aluguer, incluindo o custo.
- **Listar alugueres:** apresentar a lista de todos os alugueres com as principais características de cada um e o respetivo custo. No final, apresentar o valor total dos alugueres.
- **Terminar:** antes de terminar, os alugueres devem ser exportados para um ficheiro de objetos (alugueres.obj), criando-o caso não exista.

A aplicação a desenvolver deve ser disponibilizada com ficheiros de texto pré-preenchidos, contendo dados de pelo menos: seis utilizadores, dez veículos e seis alugueres (para demonstração de todas as funcionalidades). A estrutura dos ficheiros de texto e do ficheiro de objetos é definida pelos estudantes. Na definição dos ficheiros de texto, deve ter-se em conta a facilidade de edição e a simplificação do *parsing* pela aplicação.

Implementação

A aplicação deverá ser implementada na linguagem Java e ter em conta os seguintes aspetos:

1. Elaboração de um diagrama de classes (UML) antes de iniciar a implementação, para prever a estrutura do projeto.
2. Serão penalizadas soluções que não sigam os princípios da programação orientada aos objetos e os conceitos de herança e polimorfismo.
3. A utilização de *instanceof*, *class.getName()* ou de métodos semelhantes será penalizada.
4. As soluções devem ter em consideração boas práticas de programação.
5. Cada classe deverá gerir internamente os seus dados, pelo que deverá cuidar da proteção das suas variáveis e métodos.
6. Cada classe deverá ser responsável por uma tarefa ou objetivo específico, não lhe devendo ser atribuídas funções indevidas.
7. Utilize a *keyword static* apenas quando tal se justifique e não para contornar erros do compilador.
8. Comente as classes e os métodos públicos segundo o formato Javadoc. Isto permitirá gerar automaticamente uma estrutura de ficheiros HTML. Comente o restante código sempre que a leitura dos algoritmos não seja óbvia.
9. Será penalizado o uso abusivo de variáveis e métodos *public*.
10. Na escolha de nomes para variáveis, classes e métodos devem ser seguidas as convenções adotadas na linguagem Java.
11. Na organização das classes deverá ser evitada a redundância do código.

Entrega

O trabalho deve ser entregue no Infoestudante até ao dia **5 de dezembro de 2025**. O projeto deve ser realizado em grupos de dois estudantes da mesma turma prática.

Devem ser entregues os seguintes ficheiros (**sem compressão**):

- Diagrama de classes em UML (pdf).
- Todas as classes .java e ficheiros de texto para teste.
- Ficheiro ZIP com o Javadoc (ZIP).
- Relatório (pdf).

Avaliação do trabalho

Para a avaliação do trabalho são considerados fatores de dois tipos:

- Caixa preta (tal como é percecionado pelo utilizador):
 - Conjunto de funcionalidades implementadas.
 - Robustez do programa.
 - Qualidade da interface.
- Caixa branca (a forma como está construído):
 - Qualidade das soluções técnicas encontradas para os problemas.
 - Estruturação do código.
 - Qualidade dos comentários.

Nota: Não se aceitam trabalhos que apresentem erros de compilação no momento da defesa. Todos os erros de execução deverão ser detetados e tratados.

Defesa final do trabalho

O trabalho será defendido através de uma discussão presencial. Os alunos deverão inscrever-se num horário de defesa no Infoestudante.

MUITO IMPORTANTE

Os trabalhos serão comparados (tanto entre os trabalhos da disciplina como com código disponível na Internet), no sentido de detetar eventuais fraudes por cópia. Nos casos em que se verifique que houve cópia de trabalho total ou parcial, os grupos envolvidos terão os projetos anulados, reprovando à disciplina. Serão aplicadas as regras da Universidade de Coimbra relativamente ao plágio.