



**INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA**

**Departamento de Engenharia de Electrónica e Telecomunicações e de  
Computadores**

**LEIC**

**Sistemas de Informação**

**Semestre de Verão 2023-2024**

**Sistema de reserva bicicletas — Bicycle reservation system**

Trabalho prático  
( Fase 1 )

Ana Beire, João Vitorino, Matilde Pato e Nuno Datia

**Maio, 2024**



# Planeamento — Planning

As datas importantes a recordar são:

- Lançamento do enunciado: **10 maio de 2024**
- Entrega: **6 de junho de 2024**

[PT] O relatório deve seguir um dos *templates* fornecidos, obrigatoriamente, sob pena de penalização. **Este deve ser conciso e apresentar a justificação de todas as decisões tomadas** (ver Critérios de Avaliação). A capa do relatório deve indicar a composição do grupo, a unidade curricular e a fase do trabalho que relata. Caso tenha adendas e/ou correções a fazer a modelos já entregues, deve indicá-las de forma explícita no relatório seguinte.

O pdf (e, o zip) gerado deve seguir o nome da seguinte forma: 'TPSisInf-2324-**MM**.ext' (MM representa os dígitos do número do grupo, e 'ext' a extensão do ficheiro), e.g.: TPSisInf-2324-01.pdf.

[EN] *The report must obligatorily follow one of the templates provided, or penalties may ensue. **This must be concise and present the justification for all decisions made** (see Assessment Criteria). The report's cover must indicate the group composition, the curricular unit and the phase of the work being reported. If you have additions and corrections to deliver to models previously submitted, tell them explicitly in the following report.*

*The generated pdf (and zip) must follow the name as follows: 'TPSisInf-2324-**MM**.ext' (MM represents the digits of your group number, and 'ext' the file extension), e.g.: TPSisInf-2324-01.pdf.*

10 de Maio de 2024, Ana Beire, João Vitorino, Matilde Pato e Nuno Datia



## Objetivos de aprendizagem

No final do **projeto**, os alunos devem ser capazes de:

- Desenvolver uma camada de acesso a dados, que use uma implementação de JPA e um conjunto dos padrões de desenho, incluindo *DataMapper*, *Repository* e *UnitOfWork*;
- Enquadrar e justificar o uso dos padrões de desenho, descritos na alínea anterior, no desenvolvimento da solução;
- Desenvolver uma aplicação em Java, que use adequadamente a camada de acesso a dados;
- Utilizar corretamente processamento transacional, através de mecanismos disponíveis no JPA (*Jakarta Persistence*);
- Garantir a correta libertação de ligações e recursos, quando estes não estejam a ser utilizados;
- Garantir a correta implementação das restrições de integridade e/ou lógica de negócio.

## Enunciado do trabalho (Documento de requisitos do sistema)

Pretende-se desenvolver um sistema de reservas de bicicletas para uma empresa de turismo “Go-Cycle”. A empresa possui várias lojas, que são caracterizados por um código único, um gestor, uma morada (inclui localidade), um número de telefone, e um endereço eletrónico.

A empresa oferece dois tipos de bicicletas: clássica e elétrica, sendo cada bicicleta caracterizada por um identificador (único), o peso (em gramas), um modelo, uma marca, o sistema de mudanças, um estado, e está associada a um dispositivo GPS. O sistema de mudanças descreve o número de velocidades e tem os seguintes valores {1, 6, 18, 24}. A descrição das mudanças pode ser alterada/adicionada dinamicamente. O estado pode tomar os valores *livre*, *ocupado*, *em reserva* e, *em manutenção*. As bicicletas elétricas têm, adicionalmente, uma autonomia (em km) e uma velocidade máxima (km/h).

O dispositivo GPS tem um número de série único, uma localização<sup>1</sup> constituída por latitude e

<sup>1</sup><https://www.gps-coordinates.net/>

longitude em graus decimais (DD, na sigla em inglês), e uma percentagem de bateria, reportadas periodicamente para o sistema. O dispositivo GPS está associado a uma única bicicleta.

Os clientes são identificados por um identificador (único), e caracterizados por um nome, uma morada, um endereço eletrónico, um no. de telefone, o no. de CC ou passaporte (estrangeiros), e uma nacionalidade. Um cliente pode fazer várias reservas ao longo do tempo. Cada reserva é caracterizada por um número de reserva, único para cada loja, uma data de início e de fim da reserva, a bicicleta a reservar, e o valor a pagar (em euros).

Uma bicicleta pode ser reservada várias vezes, mas uma reserva está associada a apenas uma bicicleta. Não podem existir duas reservas para o mesmo cliente na mesma data de início. Para garantir as reservas, só podem ser feitas novas reservas se o número de bicicletas em reserva for igual ou superior em 10% das bicicletas disponíveis para reservas.

Note-se que um gestor também pode ser um cliente. Para garantir a consistência dos dados, a remoção de clientes, gestores e bicicletas é feita de forma lógica, i.e., deixam de estar ativas no sistema mas mantêm-se na base de dados.

**PS.1** Todos os atributos são obrigatórios, exceto se indicado em contrário;

**PS.2** Os valores monetários devem ser apresentados com 2 casas decimais;

**PS.3** Considere que todos os atributos com informação temporal têm data (dd/mm/aa) e tempo (hh:mm:ss);

**PS.4** Não existe nenhuma intenção de, neste exemplo, replicar com exatidão quaisquer situações reais.

## Resultados pretendidos

Tendo em conta os objetivos de aprendizagem, deverão ser produzidos os seguintes resultados:

1. Criação de uma aplicação Java que permita:

- (a) Criar um Cliente;
- (b) Listar as bicicletas existentes;
- (c) Verificar a disponibilidade de uma bicicleta num momento no tempo (têm de usar uma função);
- (d) Obter as reservas existentes;
- (e) Realizar uma reserva (têm de usar um procedimento armazenado);

- (f) Cancelar uma reserva.
2. Reimplemente a funcionalidade 1f usando *optimistic locking*.
  3. Teste o *optimistic locking*, apresentando uma mensagem de erro adequada em caso de alteração concorrente conflitante que inviabilize a operação. No relatório deve estar descrita a forma como as situações de erro foram criadas para teste desta alínea.

**Data limite para entrega: 6 de junho de 2024 até às 23:59.**

A entrega deve incluir um “relatório” e o código, enviados de forma eletrónica através do Moodle. O relatório **é entregue** em formato PDF. No final deverá colocar o código e o pdf num único um arquivo ZIP.

**Notas:**

1. Deve garantir a correta implementação de todas as funcionalidades, incluindo o acesso a dados;
2. A lógica de interface com o utilizador deve estar em classes separadas da lógica de acesso a dados;
3. O projeto deve ser possível de compilar e executar usando a ferramenta MAVEN;
4. A aplicação deve estar configurada por omissão para para aceder à base de dados do grupo disponível no servidor PostgreSQL indicado no Moodle;
5. Deve ser possível aferir cada um dos objetivos de aprendizagem no material que entregar;
6. O código deve ser compilado para a **máquina virtual Java 17<sup>2</sup>**.

Todo o código entregue tem de ser executado independente do ambiente de desenvolvimento, em linha de comandos. Os alunos têm de fornecer as instruções de execução.

---

<sup>2</sup>Versões diferentes implica penalização da nota.

## Learning Objectives

At the end of **the first step** of the work, students should be able to:

- Develop a data access layer that uses a JPA implementation and a set of design patterns, including `DataMapper`, `Repository` and `UnitOfWork`;
- Framing and justifying the use of the design patterns described in the previous paragraph in the development of the solution;
- Develop a Java application that uses the data access layer appropriately;
- Correctly use transactional processing, using mechanisms available in JPA (*Jakarta Persistence*);
- Ensure the correct release of connections and resources when they are not in use. used;
- Ensure the correct implementation of integrity and/or business logic restrictions.

## Project Statement (System Requirements Document)

The aim is to develop a bicycle booking system for a 'GoCycle' tourism company. The company has several shops, which are characterised by a unique code, a manager, an address (including locality), a telephone number, and an e-mail address.

The company offers two types of bicycles: classic and electric. Each bicycle is characterised by a identifier (unique), weight (in grams), a model, a brand, the gear system, a status, and it is associated with a GPS device. The gear system describes the number of gears and has the following values {1, 6, 18, 24}. The description of the gears can be changed/added dynamically. The state can take the values *free*, *occupy*, *in reserve* and *under maintenance*. Electric bicycles have a range (in km) and a maximum speed (km/h). Classic bikes have several gears, ranging from 0 to 5.

The GPS device is described by a unique serial number, a location<sup>3</sup> consisting of latitude and

---

<sup>3</sup><https://www.gps-coordinates.net/>



longitude in decimal degrees (DD), and a battery percentage , reported periodically to the system. The GPS device is associated with a single bicycle.

Customers are identified by an identifier, and characterised by a name, an address, email address, a phone no., a no. CC or passport (for foreigners), a nationality. A manager can be a customer. A customer can make multiple reservations over time. Each reservation is identified by a reservation number, unique for each store, contains the start and end date of the reservation, the bike to be booked, and the value (in euros).

A bike can be booked several times, but a booking is associated with only one bike. There cannot be two bookings for the same customer on the same start date. To guarantee bookings, new bookings can only be made if the number of bikes in reserve is equal to or greater than 10% of the bikes available for booking.

**PS.1** All attributes are required unless otherwise noted;

**PS.2** Monetary values must be presented to 2 decimal places;

**PS.3** Consider that all attributes with temporal information have date (dd/mm/yy) and time (hh:mm:ss);

**PS.4** There is no intention in this example to accurately replicate any real-life situations

## Expected results

Taking into account the learning objectives, the following results should be produced:

1. Creation of a Java application that allows:
  - (a) Create a Customer;
  - (b) List existing bikes;
  - (c) Check the availability of a bicycle at a moment in time (must use a function);
  - (d) Obtain existing bookings;
  - (e) Make a booking (must use a stored procedure);
  - (f) Cancel a booking.
2. Reimplement the 1f functionality using *optimistic locking*.
3. Test *optimistic locking*, displaying an appropriate error message in the event of a conflicting concurrent change that makes the operation unfeasible. The report must describe how the error situations were created to test this item.

**Deadline for submission: 6 de junho de 2024 until 23:59.**

The delivery must include a report and the EER model, sent electronically through the Moodle. The report is delivered in PDF format. At the end you must put the code and the pdf in a single ZIP file.

**Remarks:** It should be possible to measure each of the learning objectives in the material you deliver.