



INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA

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

LEIC & LMATE

Introdução aos Sistemas de Informação

Semestre de Inverno 2023-2024

Reserva de bicicletas — Bicycle reservation

Trabalho prático
(Fase 3)

Ana Beire, Matilde Pato e Nuno Leite

Planeamento — Planning

As datas importantes a recordar são:

- Lançamento do enunciado: **6 de Outubro 2023**
- Entrega intermédia (Fase 1): **24 de Outubro de 2023**
- Entrega intermédia (Fase 2): **28 de Novembro de 2023**
- Entrega intermédia (Fase 3): **19 de Dezembro de 2023**

[PT]

Cada entrega intermédia deve apresentar o relatório e código (se houver) referentes **exclusivamente** a essa fase. 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 correcções 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: 'TP**N**ISI-2324-**MM**.ext' (N representa um dígito correspondente ao número da fase do trabalho, MM representa os dígitos do número do grupo, e 'ext' a extensão do ficheiro), e.g.: TP1ISI-2324-01.pdf.

[EN]

*Each intermediate delivery must present the report and code (if any) referring **exclusively** to that phase. 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: 'TP**N**ISI-2324-**MM**.ext' (N represents a digit corresponding to the work phase number, MM represents the digits of your group number, and 'ext' the file extension), e.g.: TP1ISI-2324-01.pdf.*

Objectivos de aprendizagem

No final da **terceira fase do trabalho**, os alunos devem ser capazes de:

- Estabelecer uma ligação ao SGBD pretendido, correctamente parametrizada, utilizando JDBC;
- Utilizar correctamente comandos parametrizados para executar operações em JDBC (*prepared statement*);
- Utilizar correctamente transações para garantir atomicidade nas operações, utilizando JDBC;
- Gerir correctamente o tempo de vida das ligações JDBC;
- Garantir a libertação de recursos, quando estes não estejam a ser utilizados;
- Utilizar correctamente o tipo `ResultSet`;
- Implementar **todas as restrições de integridade** aplicacionais que não foram possíveis em PostgreSQL.

Enunciado do trabalho (3ª fase)

Considerando o texto com os requisitos do sistema, apresentados na primeira fase do trabalho, e o modelo de dados implementado na segunda fase, deve garantir, no programa desenvolvido em Java que:

1. uma bicicleta que se encontra em manutenção, não poderá ser reservada;
2. o sistema de mudanças pode ser alterado dinamicamente.

Os alunos terão agora a oportunidade de utilizar uma API JDBC para, através de uma aplicação Java, acederem e manipularem os dados existentes no modelo físico criado na fase anterior. As alterações/actualizações/consultas irão ser feitas a nível da aplicação e não directamente na BD.

Sempre que se justificar devem ser usados os mecanismos transacionais necessários para garantir a atomicidade das operações. Também devem utilizar mecanismos que evitem ataques de “SQL injection”, bem como que evitem problemas de formatações, e.g. campos de datas.

Resultados pretendidos

Tendo em conta os objectivos de aprendizagem, deverão ser produzidos os seguintes resultados, como ilustra a Figura 2.

```
Bicycle reservation
1. Exit
2. Novel bikes
3. Update bike states
4. Average metrics for electric bikes
5. List of clients with reservations
6. List of devices
7. List of stores
8. List of managers and made reservations
>
```

Figura 1: Screenshot da aplicação Reserva de bicicletas

1. Uma aplicação Java (executado independente do ambiente de desenvolvimento) que permita realizar as seguintes operações (**não deve fazer qualquer alteração à BD já construída**):
 - (a) A empresa fez a aquisição de 2 novos modelos de bicicletas da marca Canyon¹ com as seguintes características:

peso	22,15 kg (eléctrica) e 18.26 kg (clássica)	raio	23 (ambas)
modelo	Precede:ON CF (eléctrica) e Commuter:ON 7	mudanca	12* (ambas)
autonomia	100 km	velocidade	25 km/h

*Sistema de mudanças da Shimano Deore M6100 12s. Torne a descrição do campo **mudanca** possível de ser alterado sempre que necessário. O utilizador pode fazer a inserção de novos modelos;

¹Fonte: <https://www.canyon.com/pt-pt/>.

- (b) Opção para alterar o estado de uma bicicleta de “livre” ou “ocupado” para “em manutenção”, garantindo que um cliente não poderá reservar uma bicicleta que se encontre “em manutenção”;
- (c) Opção para calcular a média em termos de autonomia, de velocidade e de bateria do dispositivo para cada bicicleta eléctrica dentro de uma marca. O utilizador deverá escolher por marca dentro da lista existente;
- (d) Opções que implementem as alíneas 2.d, 2.j, 3.f e 3.h (da Fase 2), garantindo que todos os parâmetros variáveis são alteráveis na interface com o utilizador. E.g. em 2.d “(...) estado encontra-se *em manutenção*” pode ser substituído por outro valor que não “em manutenção”; e, em 3.f “(...) efectuado mais de 5 reservas até à presente data (...)”, o utilizador poderá inserir um número de reservas diferente de 5 e, mesmo colocar um intervalo temporal.

Todas as instruções devem vir indicadas (e, explicadas) no “relatório” que dá suporte a este trabalho. Se considerar necessário, pode a título ilustrativo, mostrar tabela(s) de resultados.

Data limite para entrega: 19 de Dezembro de 2023 até às 23:59.

A entrega deve incluir um “relatório” e o código, enviados de forma electrónica 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 correcta implementação de todas as funcionalidades, incluindo o acesso a dados;
2. Deve criar em Java um modelo de dados que mapeie as relações utilizadas para objectos em memória;
3. A lógica de interface com o utilizador deve estar em classes separadas da lógica de acesso a dados;
4. Deve ser possível aferir cada um dos objectivos de aprendizagem no material que entregar.

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, assumindo como único pré-requisito a existência da **máquina virtual Java 1.8²**.

²Versões superiores serão penalizadas.

Learning Objectives

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

- Establish a connection to the desired DBMS, correctly parameterised, using JDBC;
- Correctly use parameterised commands to perform operations in JDBC (*prepared statement*);
- Use transactions correctly to guarantee atomicity in operations using JDBC;
- Manage the lifetime of JDBC connections correctly;
- Ensure that resources are freed up when they are not being used;
- Use the `ResultSet` type correctly;
- Implement application **all integrity restrictions** that were not possible in PostgreSQL.

Work statement (3rd part)

Considering the system requirements presented in the first phase of the project and the data model implemented in the second phase, the Java programme must ensure:

1. a bicycle that is currently being serviced cannot be reserved;
2. the modification system can be changed dynamically.

Students now have the chance to utilise a JDBC API to access and control the information in the physical layout developed in the previous stage using a Java application. Changes, updates or queries will be conducted at the application level rather than directly with the database.

Whenever you need to, the transactional mechanisms should be applied to ensure the atomicity of operations. Mechanisms should be employed to prevent *SQL injection attacks* and format issues related to date fields. Using objective and clear language without biased evaluations or figurative language is essential.

Expected results

Taking into account the learning objectives, the following results should be produced, as illustrated in Figura 2.

```
Bicycle reservation

1. Exit
2. Novel bikes
3. Update bike states
4. Average metrics for electric bikes
5. List of clients with reservations
6. List of devices
7. List of stores
8. List of managers and made reservations
>
```

Figura 2: Screenshot of the app Cycle reservation

1. A Java application (which runs independently of the development environment) enables users to carry out the following operations **(it must not make any changes to the already established database)**:

- (a) The business has acquired two fresh Canyon³ bicycle models with the below features:

peso	22,15 kg (eléctrica) e 18.26 kg (clássica)	raio	23 (ambas)
modelo	Precede:ON CF (eléctrica) e Commuter:ON 7	mudanca	12* (ambas)
autonomia	100 km	velocidade	25 km/h

*Shimano's Deore M6100 12-speed shifting system allows for flexible modification of the gear field description, with the added capability of new model insertion;

- (b) Option to calculate the average range, speed and battery of the device for each electric bike within a brand. The user must choose by brand from the existing list;

³Source: <https://www.canyon.com/pt-pt/>.

- (c) Option to calculate the average range, speed, and battery of each electric bicycle within a specific brand from a pre-existing list. The user is required to select the preferred brand from the list;
- (d) Provide options that incorporate points 2.d, 2.j, 3.f and 3.h (of Phase 2), making sure that all variables can be changed via the user interface. For instance, in 2.d, the term "in maintenance" can be substituted by an alternate value; and in 3.f, "made more than five bookings to date" can be altered to represent any number of bookings, or even be modified to include a time frame.

All necessary guidelines should be detailed in the report supporting the study. If deemed relevant, tables may be included to support the findings.

Delivery deadline: 19 de Dezembro de 2023 by 23:59.

The delivery must include a 'report' and the code, sent electronically via Moodle. The report is delivered in PDF format. Finally, you should put the code and the pdf in a single ZIP file.

Notas:

1. You must guarantee the correct implementation of all functionalities, including data access;
2. You must create a data model in Java that maps the relationships used to objects in memory;
3. The user interface logic should be in separate classes from the data access logic;
4. It should be possible to assess each learning objective in the material delivered.

All the code delivered must be executed independently of the development environment, on the command line. Students must provide the execution instructions, with the only prerequisite being the existence of the **Java 1.8 virtual machine**⁴.

⁴Higher versions will be penalised