Guidelines for the development of the theoretical work
Datas Structures: *TRAIN*

Objetive

In this theoretical and practical work, made in groups, the aim is to **specify** and **implement**, with static and dynamic memory, the Abstract Data Type TRAIN. The ADT will have the following requirements:

- Initially, a single wagon is included in a train, and wagons may be added as needed up to a maximum number specified for each train.
- All wagons have a wagon number and n rows and 4 columns of seats (being, respectively, left window, left aisle, right aisle and right window). n will be defined at the time of the train creation.
- Seats can be free or occupied. An occupied seat must contain an identification of the occupant.
- `seatReservation` operation: When choosing the next free seat on a train, the following criteria shall be followed:
    - The available wagons will be occupied gradually, so that the seat assignment will be made in the wagon with more capacity (more free seats), and within this wagon the search will be done in an increasing way by rows and columns.
    - If there is no free seat in any of the wagons of the train, a null value will be returned. However, in case the train has capacity for more wagons, a new wagon must be added in order to make it possible to reserve a seat.
- The `numPassengers` operation will indicate the total number of passengers in the train.
- The `trainFull` operation will return a true value if all seats are occupied and if no new wagons can be added.
- Other operations should be available in order to have a complete and useful ADT.

Once the ADT has been implemented, a Java program must be created for testing. In this case, a menu will be created with at least the following options:

- **Create a new train**. The necessary parameters will be indicated.
- **Buy n tickets**. In this case the system will allow to make a sequential purchase of n tickets. The name of the passengers will be the name indicated followed by the ticket number (e.g. P1, P2... Pn).
- **Show train**. It will show on the screen the train composition and the occupation of the seats.

This activity is focused on demostrating **team work** and **oral presentation** skills of the students, as well as using and implementing data structures to solve programming problems.

Presentation and delivery requirements

All Java files from the implementation and the internal documentation files of the programs generated will be submitted in a compressed file.

The work will be presented by means of a **short video** (with a duration between 6 and 9 minutes) through which students will do the following:

- Present the algebraic specification (syntax) of the proposed ADT.
- Explain the organization and memory approach they define for this abstract data type.
- Explain the Java classes and methods that compose their TAD.
- Show the documentation files in the video (API of the ADT).
- Show the execution of the main menu options illustrating how seats are assigned to different passengers, how the creation of new wagons is performed, etc. A set of cases will be shown to check that the program works correctly.

In “Campus Virtual” you can find a file with a guide and examples of *screen-casting* (see Theoretical Work section). Besides, the students can use other applications such as “Fotos” (included in Windows 10 OS), or a different one.

This work will be carried out in **groups of 3 students**. In the first 5 seconds of the video the title of the problem, names, photos of all the students, and participation percentage (using the participation template criteria, the

same as the one presented in the laboratory section) will be shown. The video should have a good quality, in which the voice of all students will be clearly heard.

The work will be submitted through the corresponding task in "Campus Virtual" by a group member (attaching a link to the video (avoid uploading the whole video), and must be named with the surnames of the students. The deadline for submission is **January 8, 2023 at 23:00h**.

Assessment

Remember that this work will take a score of 15% on the total grade of the course. This activity is *not mandatory*.