# Mars Rover kata (TDD).

## Description

In this exercise we'll implement from scratch a subset of the Mars Rover kata using TDD.

This is a description of the behavior of the rover that we have to implement:

- It's located on a grid at some point with coordinates (x,y) and facing a direction encoded with a character.

- The meaning of each direction character is:

    - `N` -> North
    - `S` -> South
    - `E` -> East
    - `W` -> West

- The rover receives a sequence of commands (a string of characters) which are codified in the following way:

    - When it receives an `f`, it moves forward one position in the direction it is facing.
    - When it receives a `b`, it moves backward one position in the direction it is facing.
    - When it receives a `l`, it turns left changing its direction

(by 90º).
    - When it receives a `r`, it turns right changing its direction (by 90º).

## Goals

The main goal of this exercise is to practice test-driven development (TDD).

## How we'll do it?

1. Before the session read the description of what the code should do and then have a look at the formulation above.
2. We'll start by analyzing the problem a bit and creating a test list, then we'll start writing a solution.

## Things to keep in mind.

1. Writing a list of tests.
2. How the order of the tests affects the process.
3. How to choose the next test so that it makes the functionality grow in small increments.
4. The size of steps.
5. The different moments in the TDD cycle in which we are actually designing.
6. Refactoring as soon as we detect a code smell, but not before.
7. Using wishful programming.
8. Making test failure feedback as clear as possible.

9. Having focused tests with expressive names.
10. Avoiding bureaucratic tests
11. Deleting redundant tests.
12. Only adding production code after a failing test