



# Voxus Challenge I - Lógica

## Instruções

1. A partir de agora você tem **24 horas** para resolver o teste abaixo. Você pode utilizar a linguagem de programação e framework que preferir, desde que siga **todas** as regras do enunciado;
2. Ao finalizar, ou no fim das 24 horas, crie um repositório **privado** em uma plataforma de código, sendo preferencial o **GitHub** ou **BitBucket**;
3. Conceda acesso de leitura ao repositório criado ao e-mail [gustavo.lei@voxus.tv](mailto:gustavo.lei@voxus.tv);
4. Certifique-se que seu repositório está privado e que o acesso foi concedido corretamente;
5. Não se esqueça de detalhar no **readme** o processo de configuração para rodar o seu teste;
6. Envie uma mensagem para os e-mails [gabriela.rodriques@voxus.tv](mailto:gabriela.rodriques@voxus.tv) e [recrutamento@voxus.tv](mailto:recrutamento@voxus.tv) contendo o link do seu repositório no corpo do e-mail.

## Enunciado

**1)** Consider that you're creating an auxiliary class for the new Voxus product, and this class is responsible for finding the day number of the year, based on a received date. The date will be a **string**, represented using [Gregorian Calendar](#) and formatted as **YYYY-MM-DD**.

### Example I:

Input: "2019-01-09"  
Output: 9  
Explanation: Given date is the 9th day of the year in 2019.

### Example II:

Input: "2019-02-10"  
Output: 41

### Constraints:

- `date.length == 10`;
- `date[4] == date[7] == "-"`, and all other `date[i]`'s are digits;
- `date` represents a calendar date between Jan 1st, 1900 and Dec 31th, 2022.
- Any received string not following the constraints **should not** be **accepted** by your code.
- You may **NOT** use libraries or language internal functions that convert the `string` to `date` or that calculates the day number of the year for the developer. You **must** create your own logic to solve the problem.
- Don't forget to follow [coding best practices](#) for naming your variables, functions and class.
  - Each language use *different naming conventions*, so try to follow **your** language conventions.
- You are **required** to containerize your solution using Docker. Don't forget to include in the `readme` the details to run your containerized application.

**2)** For the previously implemented code, create a suite of automated tests that validates that your solution works as expected for all possible cases that you can think of (both success and error). Don't forget to follow [testing best practices](#).