Testing

Introducción para desarrollo de aplicaciones backend en Java

Javier Del Águila Alba

17-05-2024

Definiendo el testing

¿Qué es?

- El testing es un conjunto de pruebas que tienen la finalidad de garantizar la calidad del software que desarrollamos.

Definiendo el testing

Conceptos relacionados

prueba SIN. / ANT.

1. f. Acción y efecto de probar.

SIN.: ▪ [para comprobar o conseguir algo] examen, experimento, comprobación, verificación, ensayo, intento, tentativa, tanteo, estudio, contraste.

▪ pena[1], trabajo, infortunio, amargura, trago[1], sufrimiento, dolor.

2. f. Razón, argumento, instrumento u otro medio con que se pretende mostrar y hacer patente la verdad o falsedad de algo.

SIN.: demostración, corroboración, testificación, justificación, evidencia, muestra, testimonio, argumento, justificante, manifestación.

3. f. Indicio, señal o muestra que se da de algo.

SIN.: indicio, señal, signo, evidencia, muestra.

4. f. Ensayo o experimento que se hace de algo, para saber cómo resultará en su forma definitiva.

SIN.: ensayo, experimento, muestra.

5. f. Operación matemática que se ejecuta para comprobar que otra ya hecha es correcta.

Definiendo el testing

Cualidades de un buen test

- Objetivo
- Mensurable
- Reproducible
- Sencillo

Definiendo el testing

¿Por qué realizamos test?

- Garantizar que el código entregado cumple los requisitos especificados
- El desarrollo guiado por pruebas (TDD) puede ayudarnos a hacer un código más robusto
- Como pruebas de regresión
- Documentación viva de un proyecto
- Otros…

Definiendo el testing

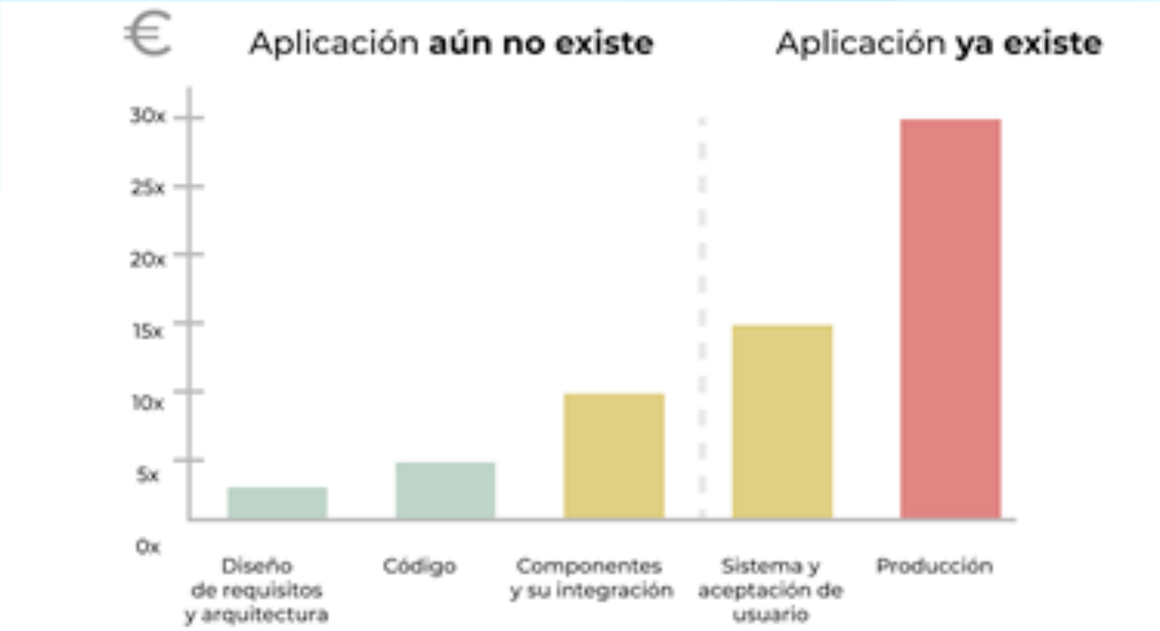
Los 7 principios del testing (ISTQB)

- El testing sirve para demostrar defectos: el objetivo de las pruebas es la identificación y resolución de errores
- No es posible realizar un testing exhaustivo

Definiendo el testing

Los 7 principios del testing (ISTQB)

- Necesidad de realizar pruebas tempranas



Fuente: hiberus.com

Definiendo el testing

Los 7 principios del testing (ISTQB)

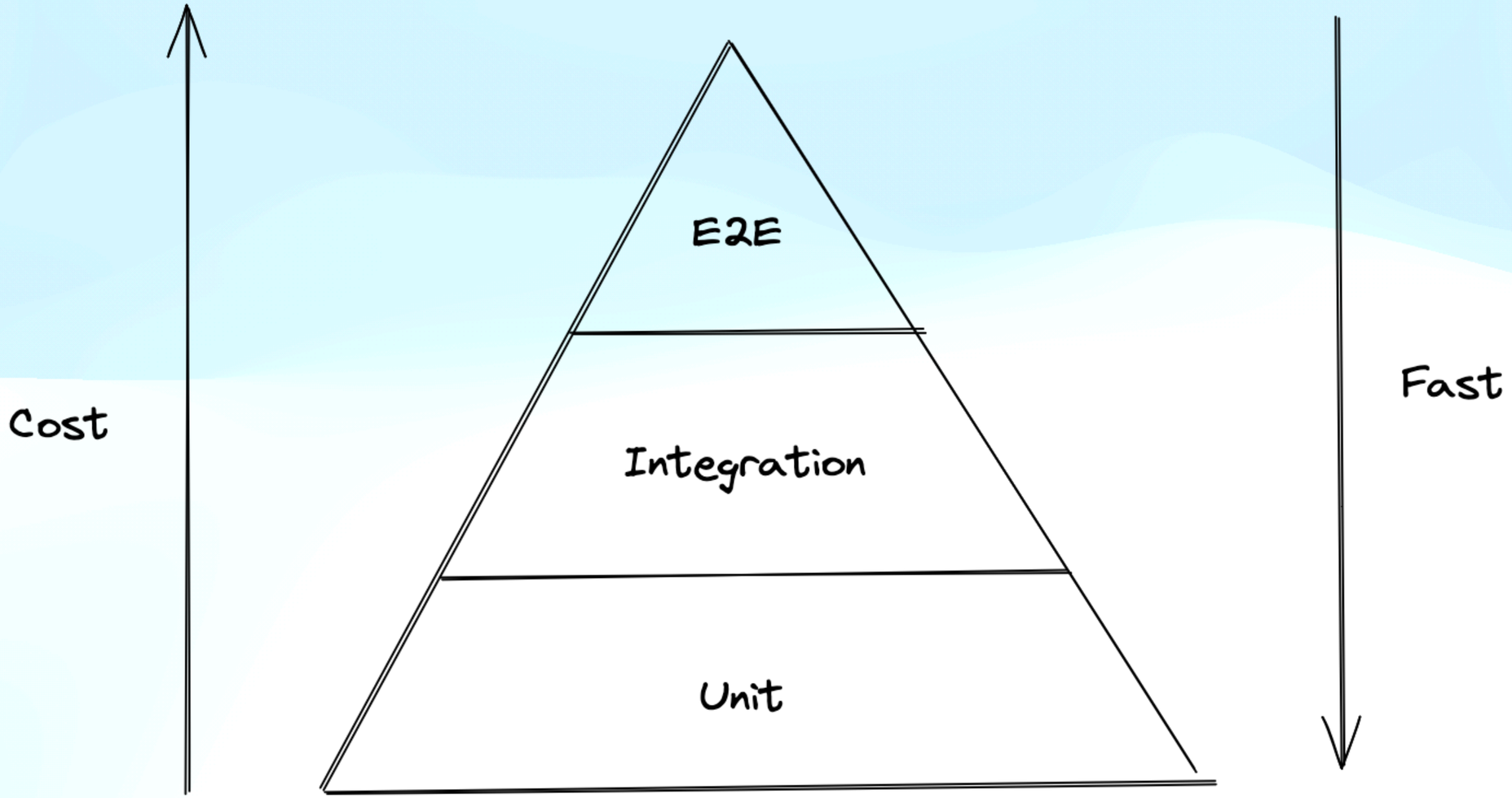
- Aglutinación de defectos: es frecuente que determinados grupos funcionales contengan muchos errores por su complejidad
- Paradoja del pesticida
- Los tests tienen un contexto
- La ausencia de errores es una falacia

Definiendo el testing

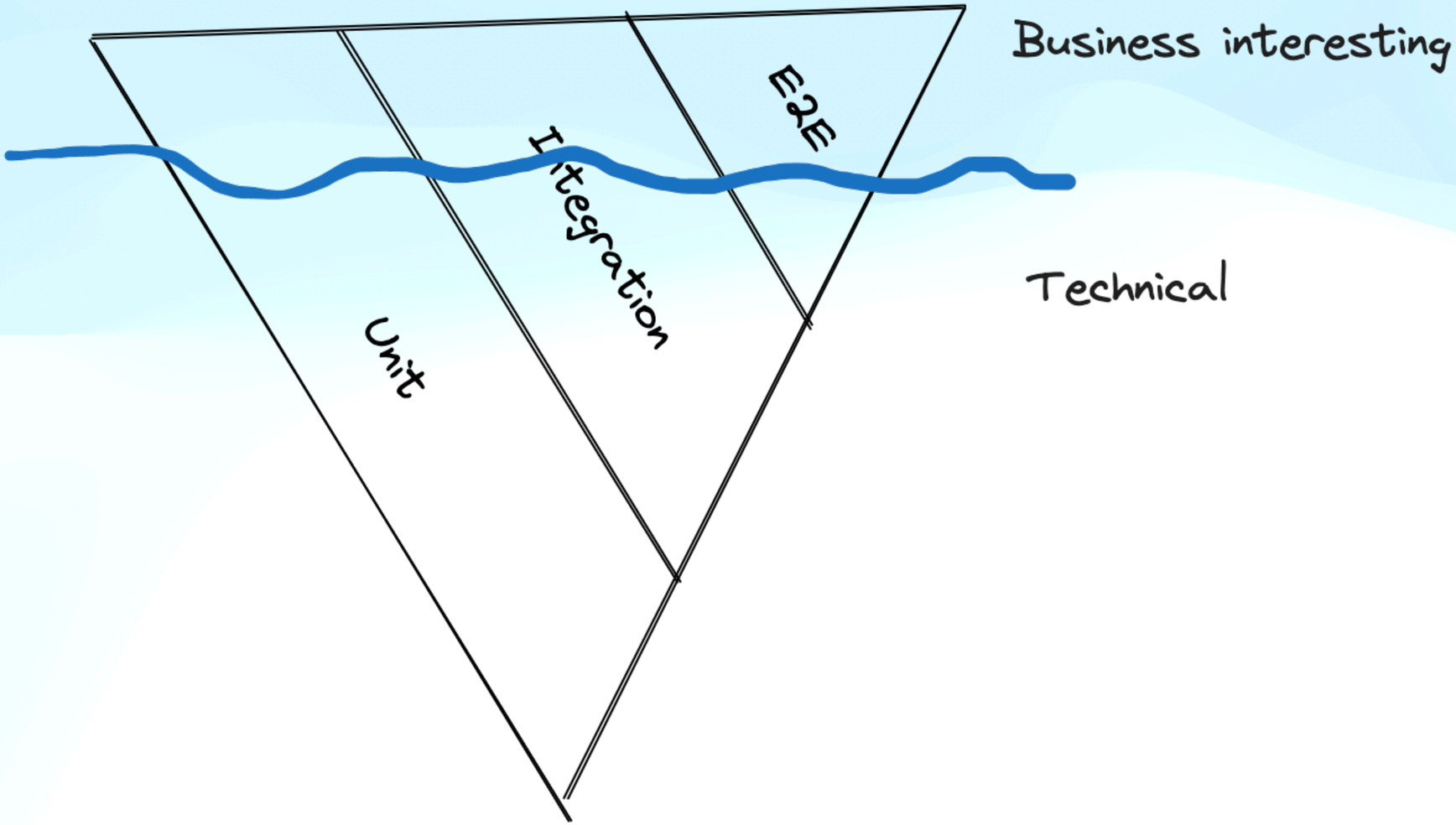
Patrón AAA

- **Arrange**: fase de preparación del test
- **Act**: ejecución de la funcionalidad que se quiere probar
- **Assert**: comprobación de resultados

Pirámide del testing



Iceberg del testing



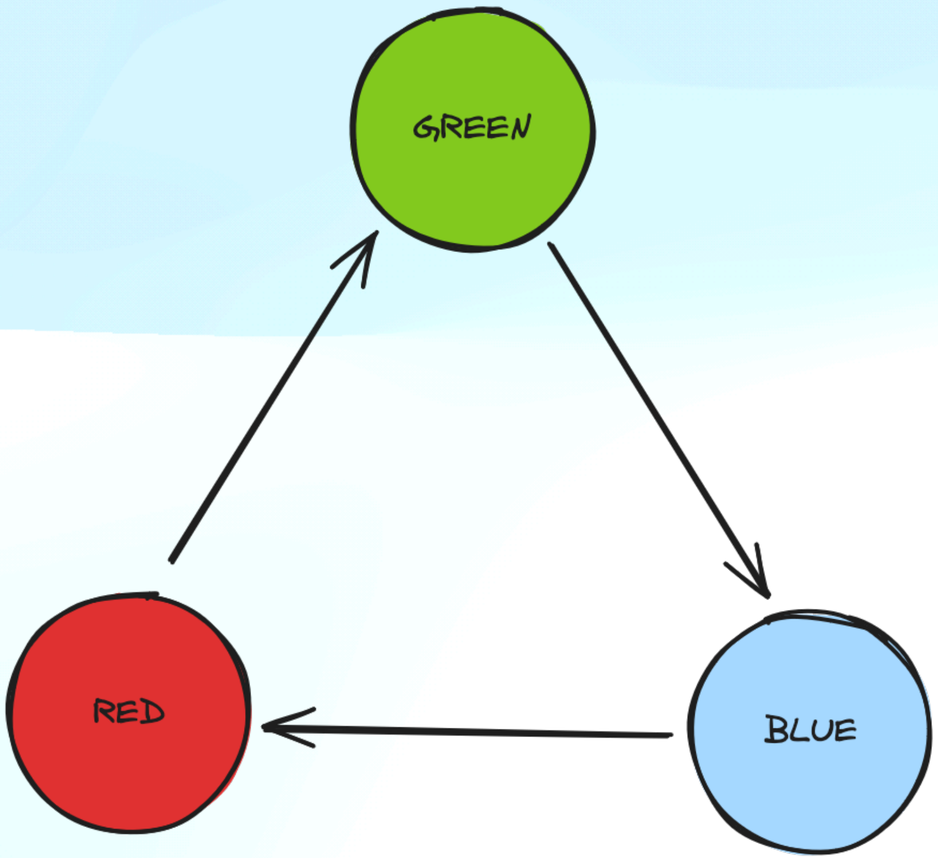
Testing unitario

Definiciones

- Tiene el menor alcance:
  - Solitario: únicamente testea el comportamiento de una clase, utilizando dobles de test si es necesario emular otras clases
  - Social: prueba el comportamiento de varias clases

TDD

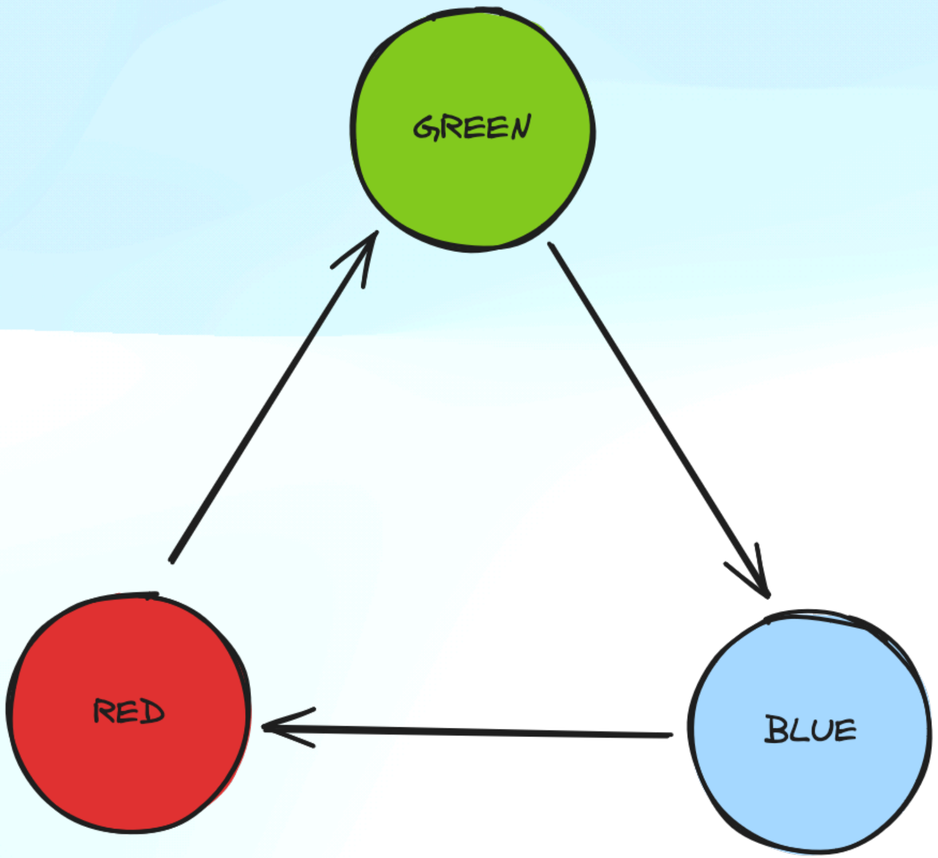
Fases



Red: ejecución de la prueba sin el código necesario para que pase

TDD

Fases

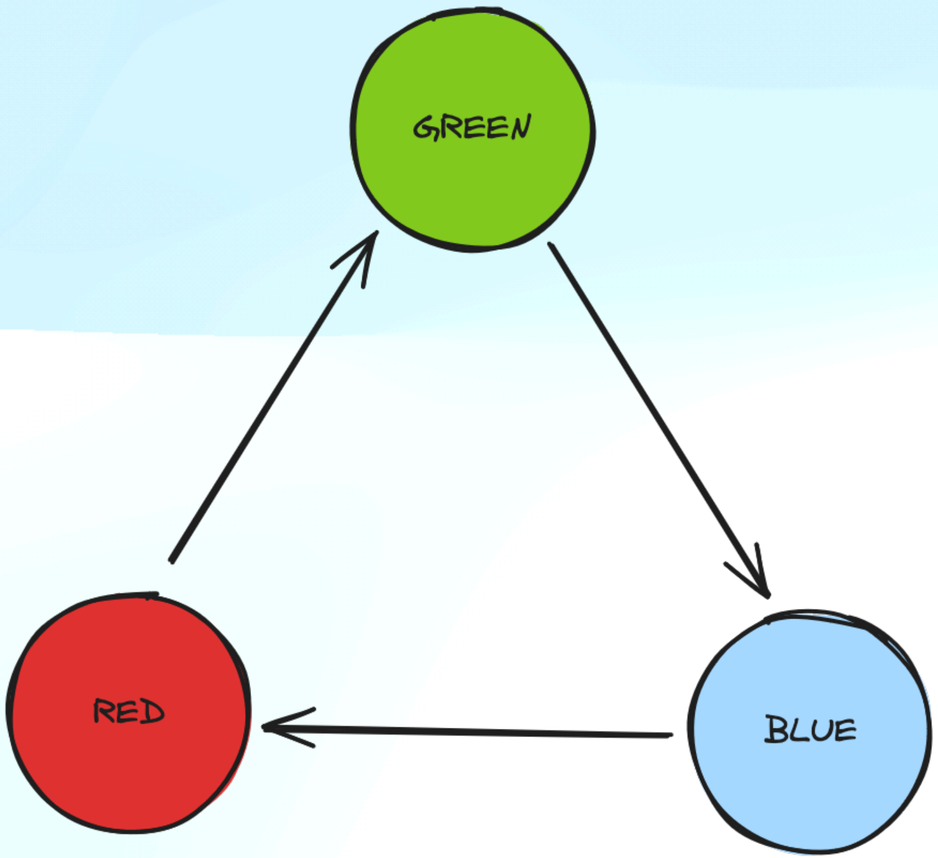


Green: implementación del código mínimo

para superar la prueba

TDD

Fases



Blue: refactor

TDD

Kata 1: FizzBuzz

- Escribir una función que reciba un número entero **n**.
  - Si **n** es 3, 6, … la función devuelve **“Fizz!”**
  - Si **n** es 5, 10, …. La función devuelve **“Buzz!”**
  - Si **n** es 15, 30, … la función devuelve **“FizzBuzz!”**
  - En cualquier otro caso, la función devuelve el número de entrada

TDD

Kata 1: FizzFuzz

TDD

Kata 2: Retrier

- Implementar un componente para reintentar la ejecución de un método un número dado de veces en caso de ciertas excepciones
- Si la excepción lanzada es técnica, se debe reintentar
- Si la excepción es de negocio, se debe elevar
- Si no se especifica un número de reintentos, éstos serán 3

TDD

Kata 2: Retrier

TDD

Kata 3: Roman

- Vamos a implementar dos funciones para convertir números arábigos en romanos y viceversa
- Se trata de números naturales positivos, y nunca superan el número 3.000
- Para más información, este enlace de wikipedia

¡Gracias!