

Introducción a la Programación

Prof. Agustín Gravano

Primer semestre de 2022

Clase teórica 5: Especificación de funciones

Especificación de funciones

Con frecuencia durante la resolución de un problema, identificamos una funcionalidad que sería conveniente abstraer: un problema particular que podría ser resuelto mediante una función.

Lo primero que debemos hacer es especificar esa función: describir con precisión qué se supone que debe hacer.

Una **especificación** tiene tres partes:

- ▶ **Encabezado**: Indica el nombre de la función, el nombre, tipo y número de los parámetros, y el tipo del valor de retorno.
- ▶ **Requiere**: Describe restricciones sobre los parámetros de la función.
- ▶ **Devuelve**: Describe el valor de retorno de la función.

Especificación de funciones

Relevamiento de requerimientos

Hay muy buenos motivos para especificar qué debe hacer una función.



1



2



3



4



5



6

1. Qué describió el cliente.
2. Qué entendió el líder del proyecto.
3. Qué se terminó programando.
4. Qué se documentó del proyecto.
5. Qué se le cobró al cliente.
6. Qué necesitaba el cliente.

Cliente, líder del proyecto, programador/a, etc. son solo **roles**, que pueden ser desempeñados por una misma persona, o bien por distintas personas.

Especificación de funciones

Relevamiento de requerimientos

Al relevar requerimientos de un sistema, es crucial evitar ambigüedades en operaciones críticas, p.ej. usando un lenguaje formal como **Z**:

| |
|----------------------------------|
| $NuevoClienteOk$ |
| $\Delta Banco$ |
| $d? : DNI$ |
| $rep! : MENSAJES$ |
| $d? \notin \text{dom } ca$ |
| $ca' = ca \cup \{d? \mapsto 0\}$ |
| $rep! = ok$ |

Fuente: <https://www.fceia.unr.edu.ar/asist/z-a.pdf>

La especificación funciona como un **contrato**. Clientes y desarrolladores la deben aprobar y firmar antes de empezar a programar. (Algo similar a los planos de una obra.)

Este tema se profundizará en la materia "Gestión de Proyectos Digitales".

Especificación de funciones

- ▶ Mucho código se escribe sin partir de una buena especificación. Es una **mala costumbre** y suele llevar a numerosos problemas.
- ▶ Claramente conviene, **antes** de sentarse a programar, invertir tiempo en (1) especificar **qué** queremos resolver, y (2) pensar algoritmos que digan **cómo** lograrlo.
- ▶ En esta materia, como aún no manejamos lenguajes formales suficientemente expresivos, vamos a **especificar en español**, procurando ser claros para minimizar las ambigüedades.
- ▶ Además, vamos a complementar esto con una segunda técnica muy útil: **escribir ejemplos** del comportamiento esperado.
- ▶ **Objetivo de una especificación:** Describir con la menor ambigüedad posible qué debe hacer una función. Debe poder entenderla (correctamente) una persona distinta a quien la escribió.

Especificación de funciones · Ejemplo

Supongamos que formamos parte de un equipo que está desarrollando un sistema. Nos piden que implementemos la función `volumen_cilindro`, en base a la especificación y los ejemplos de uso que preparó otra persona.

Calcular el volumen de un cilindro de radio `r` y altura `h`.

- ▶ **Encabezado:** `volumen_cilindro(r:float, h:float) → float`
- ▶ **Requiere:** `r > 0`; `h > 0`
- ▶ **Devuelve:** aproximadamente $\pi \cdot r^2 \cdot h$, donde $\pi \approx 3.1415927$

Conjunto de ejemplos:

| r | h | Resultado | Criterio/comentarios |
|----------|----------|------------------|-----------------------------|
| 1.0 | 1.0 | 3.14159 | valores enteros |
| 10.0 | 10.0 | 3141.5927 | valores enteros |
| 12.345 | 6.789 | 3250.4080 | valores con parte decimal |
| 0.1 | 0.1 | 0.0031416 | valores con parte decimal |

1. Pensar un algoritmo que resuelva este problema.
2. Programarlo en Python y probarlo para los ejemplos dados.

Especificación de funciones · Otro ejemplo

Ahora supongamos que estamos a cargo del relevamiento de requerimientos, y nos piden especificar una función que cuente la cantidad de vocales en un texto formado solo por letras, espacios y signos de puntuación.

- ▶ **Encabezado:** `cant_vocales(texto:str) → int`
- ▶ **Requiere:** texto está formado solo por los caracteres `,. ; : ¿ ? ¡ ! - ' " a á b c d e é f g h i í j k l m n ñ o ó p q r s t u ú v w x y z`, sus mayúsculas y espacios en blanco.
- ▶ **Devuelve:** la cantidad de apariciones en texto de los caracteres `a á e é i í o ó u ú ü` y sus mayúsculas.

Conjunto de ejemplos:

| texto | Resultado | Criterio/comentarios |
|-----------------------------|-----------|--------------------------------------|
| ' ' | 0 | caso vacío |
| 'aáÁÁ' | 4 | variantes de vocales |
| 'aeiouAEIOUáéíóúüÁÉÍÓÚÜ' | 22 | todos las vocales que deben contarse |
| 'bcdfghjklmnpqrstvwxyz' | 0 | consonantes minúsculas |
| 'BCDFGHJKLMNPQRSTUVWXYZ' | 0 | consonantes mayúsculas |
| ',. ; : ¿ ? ¡ ! - \ ' ' ' ' | 0 | signos de puntuación |
| 'eeee' | 5 | vocales repetidas |

¿Qué algoritmo resolvería este problema?

Docstrings en Python · Ejemplos

En el código en Python, volcamos la especificación en un **docstring**:

```
1  def volumen_cilindro(r:float, h:float) -> float:
2      ''' Requiere: r > 0, h > 0
3          Devuelve: el volumen aproximado de un cilindro de
4              radio r y altura h, calculado como pi*r*r*h,
5              donde pi ~ 3.1415927
6          '''
7      [...]
```

```
1  def cant_vocales(texto:str) -> int:
2      ''' Requiere: texto está formado solo por caracteres
3          ,.,:¿?¡!"aábcdeéfgghiíjklmññoópqrstuúvwxyz,
4          sus mayúsculas y espacios en blanco.
5          Devuelve: la cantidad de apariciones en texto de
6              los caracteres aáeeííoóuúü y sus mayúsculas.
7          '''
8      [...]
```

Además, los conjuntos de ejemplos se transformarán en **casos de test**, como veremos en la próxima clase.

Repaso de la clase de hoy

- ▶ La especificación de funciones es útil para el relevamiento de requerimientos.
- ▶ El objetivo es describir sin ambigüedad **qué** debe hacer una función.
- ▶ En esta materia vamos a trabajar con **especificaciones en español**, sumadas a **conjuntos de ejemplos** para describir el comportamiento esperado.

Bibliografía complementaria: Misma que la clase pasada.

Con lo visto, ya pueden resolver hasta el Ejercicio 3 de la Guía de Ejercicios 2.