

Programación I - Segundo Cuatrimestre

Unidad 2: Diseño de Programas

"La Receta Infalible para Programar"

Septiembre 2021

Natalia Colussi

Licenciatura en Ciencias de la Computación
Facultad de Ciencias Exactas, Ingeniería y Agrimensura
Universidad Nacional de Rosario

1. Diseño de Programas

Diseño de Programas

Cómo vemos nuestros programas ...

- Comunicación.

¿Quien lee nuestro código?

- Trabajo Colaborativo.

¿Cómo trabajamos en programación?

- Complejidad de los Problemas. ¿Cómo son los programas que desarrollamos?

- Requerimientos. ¿Sabe el cliente lo que quiere?

- Tiempo. ¿Cuánto tiempo nos lleva desarrollar un programa?

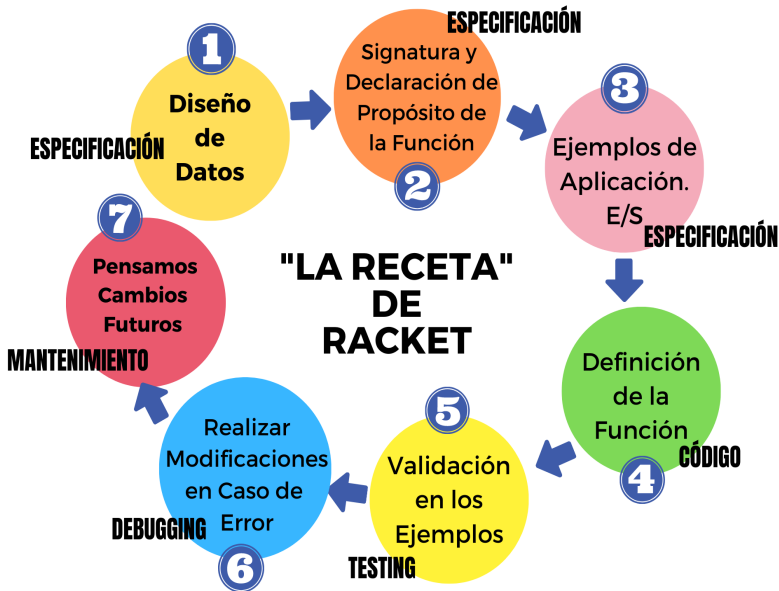
- Errores. ¿Quien los produce? ¿Cómo los detectamos? ¿Cómo los arreglamos?

Definición

Un programa bien diseñado es aquel que viene acompañado de una explicación de lo que hace, qué tipo de entradas espera y qué resultados produce. También debería, idealmente, demostrar que realmente hace lo que se afirma, pero nos conformamos con una verificación parcial automática contra un subconjunto de datos entrada y la salida esperada.

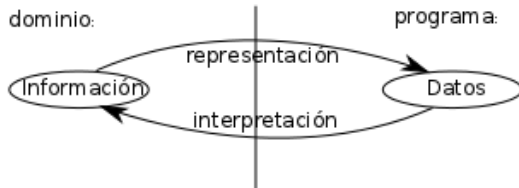
- Algunos de los objetivos en el diseño de un programa son: *correcto, compatible, componible, durable, eficiente, extensible, flexible, mantenible, eficiente, portable, legible, confiable, reusable, escalable, testeable, usable, útil, etc.*
- Utilizaremos una RECETA para producir un buen diseño del programa.

- La RECETA que seguiremos para construir un programa, nos dirá ¿Qué pasos debemos seguir? y ¿En qué orden hay que hacerlo?
 1. Diseño de datos
 2. Signatura y declaración de propósito
 3. Ejemplos
 4. Definición de la función (código)
 5. Evaluar el código en los ejemplos (testing)
 6. Realizar modificaciones en caso que el paso anterior genere errores.(debugging)



- Usaremos La Receta anterior cada vez que escribamos una función en Racket.
- No es la única metodología, pero es la que usaremos en este curso.
- Diseño vs Definición
- Al diseñar nuestras funciones bajo la metodología explicada logramos que éstas sean confiables al estar testeadas, legibles, extensibles, reusables, mantenibles, usables, durables, componible, entre otras.

El Diseño de Datos



- Los problemas existen en el mundo real. (*Información*)
- Los programas existen dentro de un modelo computacional. (*Datos*)
- Ejemplos:
 - ✓ dinero : Number
 - ✓ temperatura: Number
 - ✓ natural: Number
 - ✓ velocidad: Number
 - ✓ palabra: String
 - ✓ emoji: Image

- Establecemos la cantidad de argumentos y el tipo asociado a cada uno.
- Establecemos el tipo que produce la función.
- Separamos por espacio los tipos de los argumentos.
- Escribimos el tipo del resultado de la función después de la flecha.
- Se escribe como un comentario en el código, primero escribimos el nombre de la función, luego dos puntos, y luego la signatura.
- Ejemplos:
 - ✓ ; Number -> Number
 - ✓ ; Number String String -> Image
 - ✓ ; String -> String
 - ✓ ; Number Number Image -> Boolean
 - ✓ ; String Image -> Number | String
 - ✓ ; Any -> Number
 - ✓ ; Any -> Any

- Ejemplo 1:
Diseñe una función que calcula la suma de los cuadrados de dos números dados.
- Vamos por algunos ejemplos más...



¿Preguntas?

- Pueden dejar las consultas en el foro del capítulo, aclaren qué son sobre cuestiones teóricas.
- Pueden consultar también sobre sus dudas en los horarios de clases teóricos síncronos.