Conceptos generales.

La relación entre el problema y la solución.

Así como un globo terráqueo es una representación del planeta tierra,

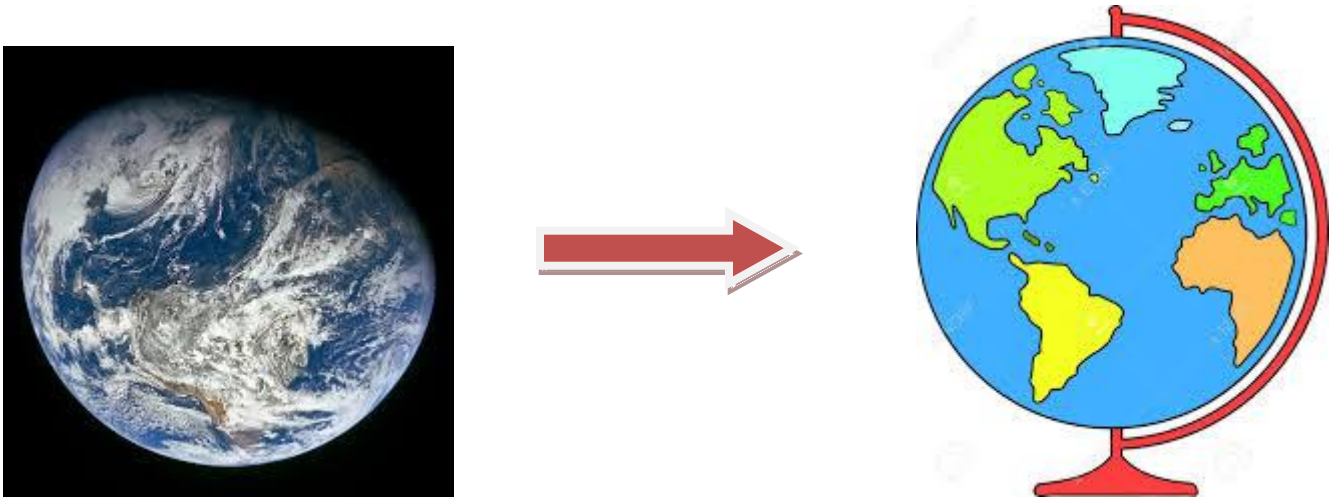

Gráfico 1

un sistema informático (conjunto de programas, o programa, o función) es una representación de la realidad, que trata de organizar los datos y sus transformaciones de una manera que represente lo más cercanamente posible la situación del mundo real, siendo dicha representación manejable por la computadora.

*Como en general, estos dos requerimientos resultan conflictivos, entonces centramos nuestra atención en **el significado de las cosas apropiado para su utilización**, dejando de lado aspectos que resulten irrelevantes o inadecuados para el propósito requerido.*

Una solución informática, generalmente, responde ante un problema del mundo real, sea algo concreto (como por ejemplo, determinar el monto de dinero que produce un plazo fijo por una dada cantidad de dinero, durante un tiempo determinado, a una determinada tasa de interés) o algo abstracto (como por ejemplo, el seguimiento de un proyecto para la construcción de un puente). Lo que sí debe quedar en claro, es que resulta imprescindible determinar para qué se necesita una solución, dicho en otras palabras, qué problema resuelve. Nótese, por ejemplo, que un sistema informático bancario utilizado para una cabina de simulador de vuelo para adiestramiento de pilotos, no sólo resultaría inútil, sino que sería tremendamente peligroso (ni que imaginar, si se usara para un sistema de gestión de vuelo en un avión).

Las entradas, la transformación y las salidas.

Una solución informática (sistema, o programa, o función), se caracteriza por TRANSFORMAR el INPUT (Entrada) en OUTPUT (Salida).

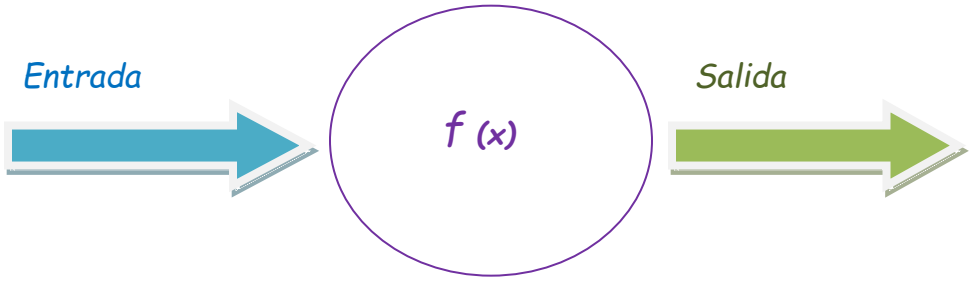

Gráfico 2

Comúnmente llamamos Aplicaciones (o Software de Aplicación) a los programas diseñados para facilitarle al usuario la realización de actividades específicas mediante la computadora (correo electrónico, navegador de internet, editor de texto, videojuego, etc.).

Cuando un desarrollador construye una aplicación, debe considerar que el usuario de la aplicación va a interactuar con el software, es decir que al momento de la ejecución (run) del programa, el usuario visualizará las salidas en la pantalla (y/o escuchará sonido) que propone el programa y, eventualmente, puede que tenga que ingresar datos a través del teclado (y/o sonido a través del micrófono, y/o elegir opciones mediante el mouse). En síntesis, programar significa ***escribir software imaginando cómo reaccionará el usuario frente al programa*** *(qué datos ingresará, qué es lo que verá, etc.).*

Gráfico 3

Normalmente, un programa está compuesto por ***funciones****, es decir, por pequeñas piezas de software, cada una de las cuales tiene un objetivo específico bien determinado. Esto facilita la solución, al descomponer el problema en sub-problemas menores, y a la vez, facilita las pruebas de buen funcionamiento, permitiendo probar cada una de las partes y luego el todo.*

En general, una función: puede recibir 0 o más datos de entrada, efectuar una o más operaciones y, finalmente, producir un resultado.

Veamos algunos ejemplos bien distintos:

- *A) una función podría **no recibir datos** y **mostrar un mensaje** de bienvenida genérico.*
- *B) una función podría pedirle al usuario, que **ingrese** su nombre **desde el teclado** y luego saludarlo **mostrando en pantalla** un mensaje de bienvenida que incluya su nombre.*
- *C) una función podría **no recibir datos** y **devolver como resultado** la fecha del día.*
- *D) una función podría **recibir por parámetro** un número entero y **mostrar en pantalla** la tabla de multiplicar de ese número desde el 1 hasta el 12.*
- *E) una función podría **recibir por parámetro** un número entero y **devolver como resultado** el factorial de ese número.*
- *F) una función podría pedirle al usuario, que **ingrese por teclado** su fecha de nacimiento, primero el año en cuatro dígitos, luego el mes en dos dígitos y finalmente el día en dos dígitos, luego validar que esos tres datos constituyan una fecha posible y si no fuera así pedirle al usuario que vuelva e ingresarlos, hasta que sean válidos. Finalmente, la función **devolvería como resultado** una fecha válida en formato (día, mes, año).*

En Python, las posibles formas en que una función recibe la Entrada de datos y las formas posibles en que una función produce su Salida o resultado, podría sintetizarse como sigue:

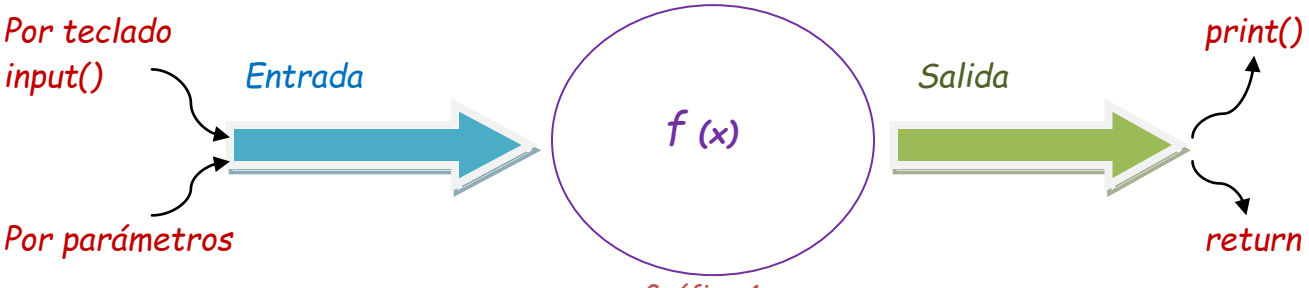

Gráfico 4

De acuerdo al contexto en que se usará una función y al uso que se dará a su salida, se decide en la etapa de diseño de qué manera recibirá los datos y en qué forma producirá su resultado.

Ahora ya tenemos algunas nociones básicas para comprender mejor los tres primeros pasos de la Metodología sencilla para la resolución de problemas con programación que propone la Unidad 2 del Apunte de la materia y que complementaremos a continuación.

1) *Análisis del Problema:*

Comencemos diciendo que el problema está en el mundo real.

Por ejemplo, supongamos que llamo a un pintor y le muestro una pared interior, descascarada, con manchas de humedad, y le digo "Mire cómo está esa pared, me gustaría que estuviera bien pintada, de color borgoña".

Es importante aclarar que el usuario (sea un funcionario o empleado de una empresa, o un particular que tiene un pequeño emprendimiento), en general, no tiene una idea clara de la solución, pero sí, en cambio, puede describir cuál es el problema (esto lo podrán comprobar en este mismo ejemplo, dado que yo no sé pintar).

*El **Análisis del problema**, consiste en producir un documento escrito que relate con precisión y de manera completa, el problema que describió el usuario.*

En el ejemplo, tendría que incluir las medidas de la pared que hay que pintar y la clase de pintura, de modo que el pintor posiblemente me preguntará sobre el tipo de pintura, tomará una cinta de medir y anotará las dimensiones. Luego, quizás, escribiría como documento de Análisis:

"El usuario me pide que prepare una pared interior de 4 x 5, muy deteriorada, y que la pinte dándole dos manos de látex color borgoña".

Un buen truco para producir un buen análisis, consiste en imaginar que una vez que el "analista" redacta el informe, allí termina su tarea, de modo que ese informe será enviado a otras personas que continuarán con los pasos siguientes.

2) *Especificación de la Solución:*

Ahora sí, entramos en el mundo de la solución, ya no estamos en presencia del usuario. Nuestro único elemento para trabajar es el informe de Análisis generado en el paso anterior, con un agravante: el "analista" (el pintor), no puede aclarar nuestras dudas, porque está dedicado a analizar otros problemas.

*La **Especificación de la solución**, consiste en describir **QUÉ** debe hacer la solución, Cuáles serán las **Entradas** y Cuáles serán las **Salidas** (Gráfico 2)*

Para nuestro ejemplo (disculpen quienes sepan pintar), la especificación podría ser:

"Hay que rasquetear una pared interior de 4 x 5, enduir, lijar y darle dos manos de pintura látex color borgoña".

***Entrada**:*
Pared de 4 x 5 descascarada
Látex interior borgoña (4 lt)
Enduído interior (1 lt)

***Salida**:* *Pared de 4 x 5 borgoña*

Algunos preguntarán si lijas, pinceles o rodillos, espátulas, escalera, forman parte de la Entrada. Aunque el ejemplo difiere mucho de lo que es producir software, podríamos pensar que los útiles que usa el pintor para trabajar no constituyen elementos de Entrada (no serían como los datos que ingresa el usuario al sistema, sino como las herramientas que usa el programador para programar). Véase que, del mismo modo, no constituye una Salida "un pintor cansado, pero contento", aunque posiblemente sea una consecuencia.

Es importante destacar que en ningún momento se menciona CÓMO realizar la tarea.

Otra vez, es conveniente imaginar que una vez terminado este paso, concluye nuestra tarea, y enviaremos la Especificación a quienes continuarán los pasos siguientes.

3) *Diseño de la Solución:*

Ahora centramos la atención en considerar posibles maneras de realizar lo que describe la especificación. Puede haber distintos CÓMO.

Por ejemplo, podría diseñar que dos personas especializadas en preparación, dividan la pared en dos sectores (uno para cada uno), rasqueteen, luego apliquen enduído, dejen secar y finalmente lijen. Al día siguiente una persona especializada en pintar, pinte con brocha la pared completa, de izquierda a derecha y de arriba hacia abajo, deje secar y aplique la segunda mano con rodillo, de arriba hacia abajo y de izquierda a derecha.

Otra alternativa, sería que haga toda la actividad un solo pintor: divide la pared en tres sectores (A, B, C), entonces rasquetea con espátula el sector A, luego aplica enduído en ese sector; luego rasquetea con cepillo de acero el sector B y aplica enduído en ese sector; luego rasquetea con espátula el sector C y aplica enduído en ese sector. Al día siguiente, lija los tres sectores, luego da la primera mano de pintura con pincel en el sector A, continúa pintando el sector B y finalmente el sector C, siempre dando pinceladas verticales, desde la parte más alta del sector y concluyendo con la parte más baja. Al día siguiente da la segunda mano en el mismo orden, pero usando un rodillo, en lugar de pincel.

Otras alternativas, serían: que el pintor cambie el orden en que prepara los sectores, o que utilice cepillo de acero para rasquetear los tres sectores o que pinte primero la parte más baja de cada sector y vaya subiendo hacia la parte más alta (no recomendable).

Es probable que exista más de una manera de hacer lo que indica la Especificación. El Diseño resuelve cuál es la manera más conveniente, según el caso. Define cuáles son los algoritmos y las estructuras de datos que se usarán. Por ejemplo, se decide si la solución será un solo programa o si utilizará diferentes funciones y, si es así, qué debe hacer cada función y qué clase de entradas y de salidas tendrá respectivamente (Gráfico 4).

Estas ideas quizás puedan ayudar a comprender el ítem 2.1 del Apunte de la materia.