	Manual de prácticas del Laboratorio de Programación básica	Código:	MADO-18
		Versión:	01
		Página	29/184
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Guía práctica de estudio 02: Solución de problemas



Elaborado por:


M.C. Edgar E. García Cano
Ing. Jorge A. Solano Gálvez

Revisado por:

Ing. Laura Sandoval Montaña

Autorizado por:

M.C. Alejandro Velázquez Mena

	Manual de prácticas del Laboratorio de Programación básica	Código:	MADO-18
		Versión:	01
		Página	30/184
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Guía práctica de estudio 02: Solución de problemas

Objetivo:

Identificar el conjunto de entrada (datos de entrada) y el conjunto de salida (datos de salida), a partir del análisis de la definición de un problema.

Actividades:

- Identificar el conjunto de datos de entrada a partir de un problema dado.
- Identificar el conjunto de datos de salida a partir de un problema dado.

Introducción


Un problema se puede definir como el conjunto de instancias al cual corresponde un conjunto de soluciones, junto con una relación que asocia para cada instancia del problema un subconjunto de soluciones (posiblemente vacío).

Para poder solucionar un problema es necesario averiguar qué es lo que requiere el usuario (análisis de requisitos). Esta etapa permite definir los requisitos de forma clara y concisa (especificación de requisitos).

En la creación de software, los desarrolladores se enfrentan al principio de un proyecto con un documento escrito por el cliente, en el cual expresa, en términos de la aplicación, qué se requiere del sistema. Este documento se conoce como declaración de requisitos.

Ingeniería de Software

La Ingeniería de Software se define como el uso y establecimiento de principios de ingeniería sólidos, a fin de obtener un software que sea económicamente fiable y funcione eficientemente.

	Manual de prácticas del Laboratorio de Programación básica	Código:	MADO-18
		Versión:	01
		Página	31/184
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

La Ingeniería de Software provee métodos que indican cómo generar software. Estos métodos abarcan una amplia gama de tareas:


- Planeación y estimación del proyecto.
- Análisis de requerimientos del sistema y software.
- Diseño de la estructura de datos, la arquitectura del programa y el procedimiento algorítmico.
- Codificación.
- Pruebas y mantenimiento (validación y verificación).

Ciclo de vida del software

La ISO (International Organization for Standardization) en su norma 12207 define al ciclo de vida de un software como:

Un marco de referencia que contiene las actividades y las tareas involucradas en el desarrollo, la explotación y el mantenimiento de un producto de software, abarcando desde la definición hasta la finalización de su uso.

Dentro del ciclo de vida del software, la solución del problema se encuentra dentro de la etapa de análisis de software:

	Manual de prácticas del Laboratorio de Programación básica	Código:	MADO-18
		Versión:	01
		Página	32/184
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

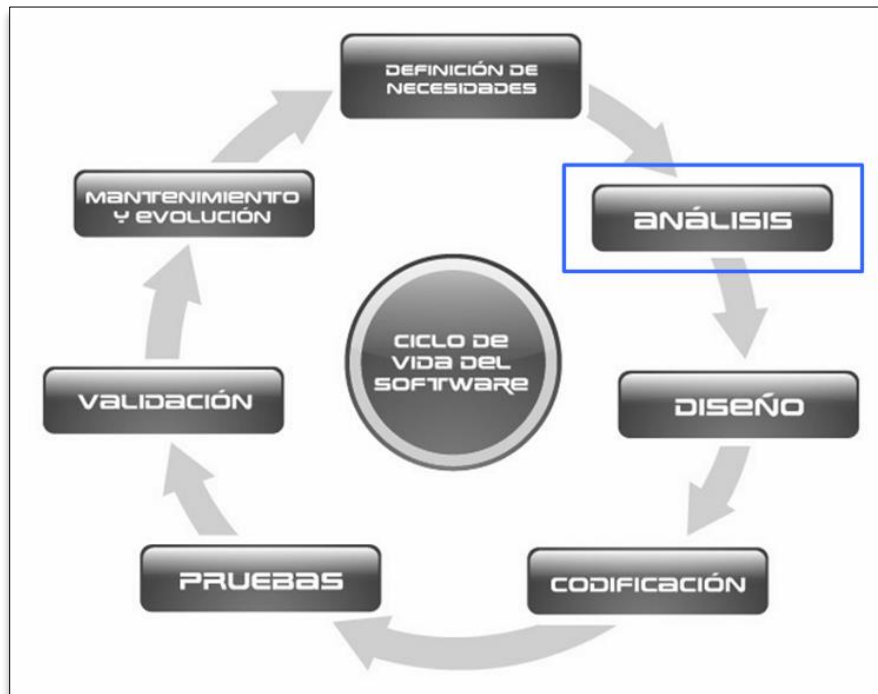


Figura 1: Ciclo de vida del software, resaltando la etapa de *análisis*, la cual corresponde a la solución de problemas.


El análisis es el proceso para averiguar qué es lo que requiere el usuario del sistema de software (análisis de requisitos). Esta etapa permite definir las necesidades de forma clara y concisa (especificación de requisitos).

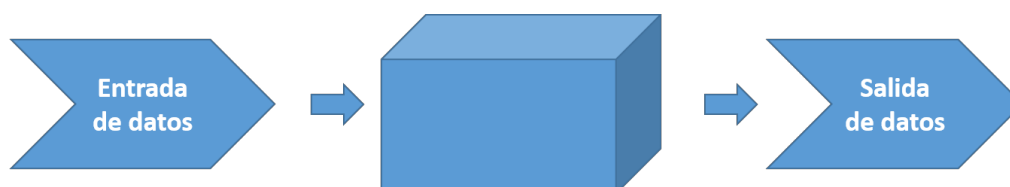
Por lo tanto, la etapa del análisis consiste en conocer qué es lo que está solicitando el usuario. Para ello es importante identificar dos grandes conjuntos dentro del sistema: el conjunto de entrada y el conjunto de salida.

El **conjunto de entrada** está compuesto por todos aquellos datos que pueden alimentar al sistema.

El **conjunto de salida** está compuesto por todos los datos que el sistema regresará como resultado del proceso. Estos datos se obtienen a partir de los datos de entrada.

La unión del conjunto de entrada y el conjunto de salida forman lo que se conoce como el dominio del problema, es decir, los valores que el problema puede manejar.

	Manual de prácticas del Laboratorio de Programación básica	Código:	MADO-18
		Versión:	01
		Página	33/184
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			



La etapa de análisis es crucial para la creación de un software de calidad, ya que si no se entiende qué es lo que se desea realizar, no se puede generar una solución. Sin embargo, es común caer en ambigüedades debido al mal entendimiento de los requerimientos iniciales.

A continuación, se presenta una imagen que representa los problemas más comunes en la administración de proyectos de software.

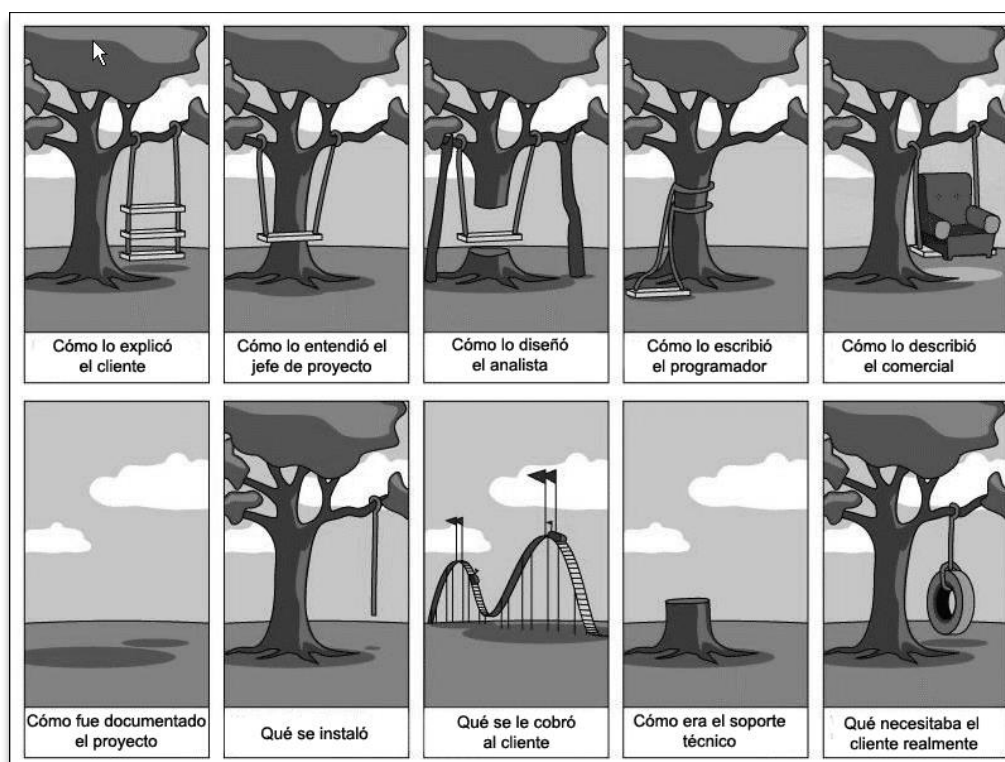




Figura 2: Fábula del columpio que describe las etapas de un proyecto contra el entendimiento en cada etapa.

	Manual de prácticas del Laboratorio de Programación básica	Código:	MADO-18
		Versión:	01
		Página	34/184
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

	Manual de prácticas del Laboratorio de Programación básica	Código:	MADO-18
		Versión:	01
		Página	35/184
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Ejemplo 1

PROBLEMA: Determinar si un número dado es positivo o negativo.

RESTRICCIONES: El número no puede ser cero.

DATOS DE ENTRADA: El conjunto de datos de entrada E está compuesto por el conjunto de los números reales, excepto el cero.

$$E \subset \mathbb{R}^1, \text{ donde} \\ \text{num} \in E \text{ de } (-\infty, \infty) - \{0\}$$

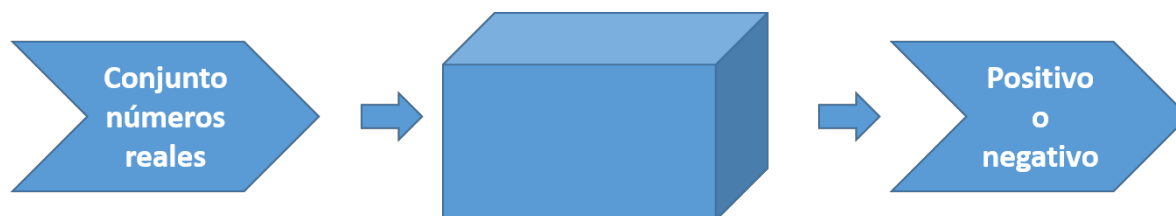
NOTA: \mathbb{R}^1 representa al conjunto de números reales de una dimensión.


DATOS DE SALIDA: El conjunto de salida S está compuesto por dos valores mutuamente excluyentes.

Un posible conjunto de salida son los valores enteros 0 o 1, donde 0 indica que el valor es positivo y 1 indica el valor es negativo.

$$\text{res} = 0, \text{ si } \text{num} \in (0, \infty), \text{ res} = 1, \text{ si } \text{num} \in (-\infty, 0)$$

Otro posible conjunto de datos de salida son los valores booleanos *Verdadero* o *Falso*, donde *Verdadero* indica que el valor es positivo y *Falso* indica que el valor es negativo; o viceversa, *Verdadero* indica que el valor es negativo y *Falso* indica que el valor es positivo.



	Manual de prácticas del Laboratorio de Programación básica	Código:	MADO-18
		Versión:	01
		Página	36/184
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Ejemplo 2

PROBLEMA: Obtener el mayor de dos números diferentes dados.

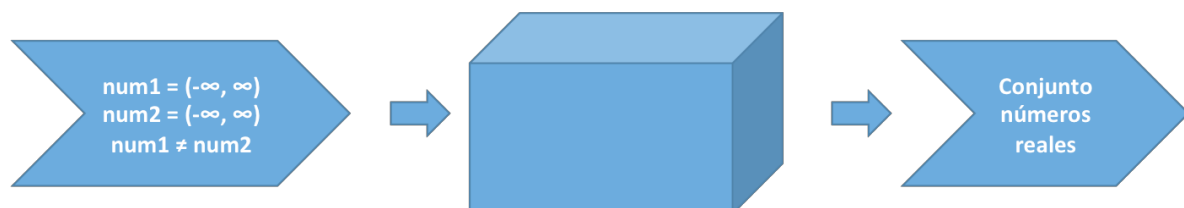
RESTRICCIONES: Los números de entrada deben ser diferentes.


DATOS DE ENTRADA: El conjunto de entrada E está dividido en dos subconjuntos E y E' . El primer número (num1) puede adquirir cualquier valor del conjunto de los números reales ($E = (-\infty, \infty)$), sin embargo, el conjunto de entrada del segundo número (num2) es un subconjunto de E , es decir, E' está compuesto por el conjunto de los números reales excepto num1 ($E' = (-\infty, \infty) \neq \text{num1}$).

$E, E' \subset \mathbb{R}^1$, donde
 $\text{num1} \in E$ de $(-\infty, \infty)$,
 $\text{num2} \in E'$ de $(-\infty, \infty) - \{\text{num1}\}$

DATOS DE SALIDA: El conjunto de datos de salida S que puede tomar el resultado r está compuesto por el conjunto de los números reales.

$S \subset \mathbb{R}^1$, donde $r \in S$ de $(-\infty, \infty)$



	Manual de prácticas del Laboratorio de Programación básica	Código:	MADO-18
		Versión:	01
		Página	37/184
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Ejemplo 3

PROBLEMA: Obtener el factorial de un número dado. El factorial de un número está dado por el producto de ese número por cada uno de los números anteriores hasta llegar a 1. El factorial de 0 (0!) es 1:

$$n! = n * (n-1)!$$

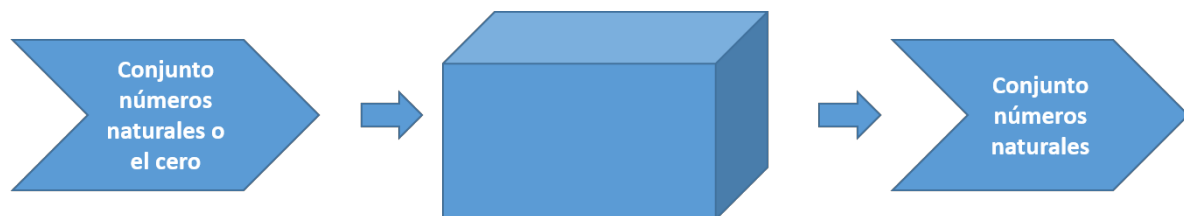
RESTRICCIONES: El número de entrada debe ser entero positivo o cero. No puede ser negativo.


DATOS DE ENTRADA: El conjunto de entrada E está dado por el conjunto de los números naturales o por el cero.

$$E \subset \mathbb{N}^1, \text{ donde} \\ \text{num} \in E \text{ de } [1, \infty) \cup \{0\}$$

DATOS DE SALIDA: El conjunto de salida S está conformado por el conjunto de los números naturales.

$$S \subset \mathbb{N}^1; \text{ donde} \\ \text{res} \in S \text{ de } [1, \infty)$$



	Manual de prácticas del Laboratorio de Programación básica	Código:	MADO-18
		Versión:	01
		Página	38/184
		Sección ISO	8.3
		Fecha de emisión	20 de enero de 2017
Facultad de Ingeniería		Área/Departamento: Laboratorio de computación salas A y B	
La impresión de este documento es una copia no controlada			

Referencias

- Raghu Singh (1995). International Standard ISO/IEC 12207 Software Life Cycle Processes. Agosto 23 de 1996, de ISO/IEC. Consulta: junio de 2015. Disponible en: <http://www.abelia.com/docs/12207cpt.pdf>
- Carlos Guadalupe (2013). Aseguramiento de la calidad del software (SQA). [Figura 1]. Consulta: junio de 2015. Disponible en: <https://www.mindmeister.com/es/273953719/aseguramiento-de-la-calidad-del-software-sqa>
- Andrea S. (2014). Ingeniería de Software. [Figura 2]. Consulta: Junio de 2015. Disponible en: <http://ing-software-verano2014.blogspot.mx>