



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Karina García Morales

Asignatura: Fundamentos de Programación

Grupo: 20

No. de práctica(s): 3

Integrante(s): Avilés Galán Iván Annael

No. de lista o brigada: 7

Semestre: 2023-1

Fecha de entrega: 21/09/2022

Observaciones:

CALIFICACIÓN: _____

OBJETIVO: El alumno elaborará algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas de Análisis y Diseño pertenecientes al Ciclo de vida del software

DESARROLLO:

Métodos para generar software.

En la práctica de laboratorio revisamos algunas de las tareas que nos permiten generar software de los cuales tenemos *planeación y estimación del proyecto* que resulta del ordenamiento de todas las ideas que surgen para elaborar un plan con un objetivo en concreto, así como de que ideas son viables y cuales no, también se tiene el *análisis de requerimientos del sistema y software* que trata de aquellas herramientas que son necesarias para el funcionamiento de los elementos del software, otra tarea es el *diseño de la estructura de datos, la arquitectura del programa y el procedimiento algorítmico* así como la *codificación* en el que se pone en proceso y se almacena toda la información e ideas con ayuda de un lenguaje ya formalizado, por ultimo están las *pruebas y mantenimiento* que es más para rectificar el funcionamiento del software y que cosas son posibles mejorar.

Ciclo de vida del software.

Como vimos durante las clases y en el laboratorio cualquier software contiene un ciclo de vida el cual se compone de ciertas actividades que requieren para su funcionamiento, desarrollo, mejoras hasta el termino de este, la etapa en la que nos estamos enfocando en esta práctica es en el análisis y resolución de problemas.

Solución de problemas.

Para resolver un problema se necesita en primer lugar un análisis o dicho de otra forma observar que elementos del problema se requiere saber para encontrar una posible solución, para esto es útil conocer dos conjuntos, el conjunto de entrada en el cual van a entrar todos los datos esenciales para alimentar al sistema y el conjunto de salida el cual va a arrojar aquellos resultados provenientes del proceso de datos del sistema.

Algoritmo.

A partir del análisis se necesita construir una serie de pasos o instrucciones detallados que me permitan resolver el problema, este debe ser lo más legible, eficiente y preciso posible para que el usuario pueda obtener una solución óptima.

Con todo lo anterior mencionado resolvimos a nuestra manera algunos ejemplos de solución de problemas en donde además se implemento las pruebas de escritorio en la cual verificamos si nuestro algoritmo es funcional.

14/09/2022

Ejemplo 1.

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

Restricciones: El número no puede ser cero

Datos de entrada: Número real (R)

Datos de salida: Si el número solicitado es positivo o negativo

Construcción.

Algoritmo.

- 1- Inicio
- 2- Solicitar un número (R) que no sea 0
- 3- Si el número ingresado es $\neq 0$ pasar al siguiente paso, si el número es igual a 0 entonces volver al paso 1
- 4- Verificar si el número dado es: $(R) > 0$ o $(R) < 0$ para saber si es positivo o negativo, continuar al siguiente paso
- 5- FIN.

Pruebas de Escritorio.

#Intentos	Número (R)	Salida
1	$10 > 0$	positivo
2	$-8 < 0$	negativo
3	$0 = 0$	- - -

EJEMPLO 2.

14/09/2022

Ejemplo 2.

Problema: Obtener el mayor de dos números dados.
Restricciones: Los números de entrada deben ser diferentes.

Datos de entrada: Número 1 (a), Número 2 (b)

Datos de salida: Obtención del número más grande

Construcción

Algoritmo.

1- INICIO

2- Solicitar dos números (a), (b) \neq entre ellos

3- Ingresar los números (a), (b)

4- Si $a = b$, entonces volver al paso 1 e ingresar nuevos valores, si $a \neq b$ pasar al paso siguiente.

5- Verificar si $a > b$ o $a < b$, entonces pasar al paso siguiente.

6- FIN

Pruebas de Escritorio

#Intentos	Número (a)	Número (b)	Salida
1	11	3	$11 > 3$
2	1	2	$1 < 2$
3	6	6	- - -

14/09/2022

Ejemplo 3

Problema: Obtener el factorial de un número dado.

Restricciones: El número dado debe entero (\mathbb{Z}) y no negativoDatos de entrada: Número Entero (\mathbb{Z})

Datos de salida: El factorial de un número

Construcción

Algoritmo.

- 1- INICIO
- 2- Solicitar un número entero (\mathbb{Z})
- 3- Si (\mathbb{Z}) ≤ 0 volver al paso 1 e ingresar otro valor, si (\mathbb{Z}) > 0 continuar al paso siguiente
- 4- Crear una (variablen contador) que inicie en 1 y una (variablen factorial)
- 5- Realizar una multiplicación (variablen contador)(variablen factorial) = F, donde F es donde se almacena el resultado de la multiplicación.
- 6- La (variablen contador) se incrementará en 1 cada vez que se realice la multiplicación.
- 7- FIN

Prueba de Escritorio.

Iteración	\mathbb{Z}	Variablen factorial = F	Variablen contador	Salida
1	9	1	2	—
2	9	2	3	—
3	9	6	4	—
4	9	24	5	—
5	9	120	6	—
6	9	720	7	—
7	9	5040	8	—
8	9	40320	9	—
9	9	362880	10	El Factorial es 362880

Ejemplo 4

Problema: Obtener la suma de un valor si este es menor a 10

Restricciones: Si el valor es mayor a 10 se resta 1

Datos de entrada: $\text{varValor} < 10 \rightarrow +10$, $\text{varValor} > 10 \rightarrow -1$

Datos de salida: La suma de un valor menor a 10

Construcción

Algoritmo.

1- INICIO

2- Pedir que se ingrese un número (R) $\leftarrow \text{varValor}$

3- Si el número ingresado ($\text{varValor} < 10$) se le sumará 10, si ($\text{varValor} > 10$) se le resta 1, seguir al paso 4

4- FIN

Pruebas de Escritorio.

#Intentos	$\text{VarValor} < 10$ o $\text{varValor} > +10$	Salida
1	$4 + 10$	14
2	$11 - 1$	10
3	$10 + 10$	--

Conclusiones.

El conocer el ciclo de vida del software es muy importante ya que nos va a permitir resolver problemas que en la carrera de Ingeniería Industrial son muy frecuentes por lo tanto saber hacer un análisis y un algoritmo es primordial ya que es la base de la construcción de un software además es necesario ser lo mas eficientes, claros y concisos con las instrucciones que proporcionemos para que el usuario al que esta dirigido le sea más fácil obtener una solución.

Bibliografía.

Laboratorio Facultad de ingeniería. Universidad Nacional Autónoma de México.

Recuperado de: <http://lcp02.fi-b.unam.mx/poll/login/>