|  |
| --- |
| Laboratorios de computación salas A y B |

|  |  |  |
| --- | --- | --- |
| Profesor: | Rodriguez Espino Claudia Ing. | |
| Asignatura: | Fundamentos de Programación | |
| Grupo: | 03 | |
| No de Práctica(s): | 3  na Dulce Monica | |
| Integrante(s): | Perea Sánchez Olivia | |
|  |  | |
|  |  | |
| Semestre: | 2017-2 | |
| Fecha de entrega: | 21 – Febrero -- 2017 | |
| Observaciones: |  | |
|  |  | |
| CALIFICACIÓN: | |  |
|  |

Guía práctica de estudio 03: Solución de problemas y Algoritmos.

**Objetivos:**

Elaborar algoritmos correctos y eficientes en la solución de problemas siguiendo las etapas de Análisis y Diseño pertenecientes al Ciclo de vida del software.

**Actividades:**

* Se realizó la lectura de los siguientes conceptos:

Ciclo de vida del Software

Solución de Problemas

* Posteriormente se analizaron los ejemplos mostrados de distintos problemas, para identificar tanto los datos de entrada como de salida.

1. Si resolver , Si resolver

Si , resolver

Si , resolver

Y R

Datos de entrada:

X R

1. Determinar si un número es par o impar

Y= X

Si Y= 1

Si Y= 0

Y= impar

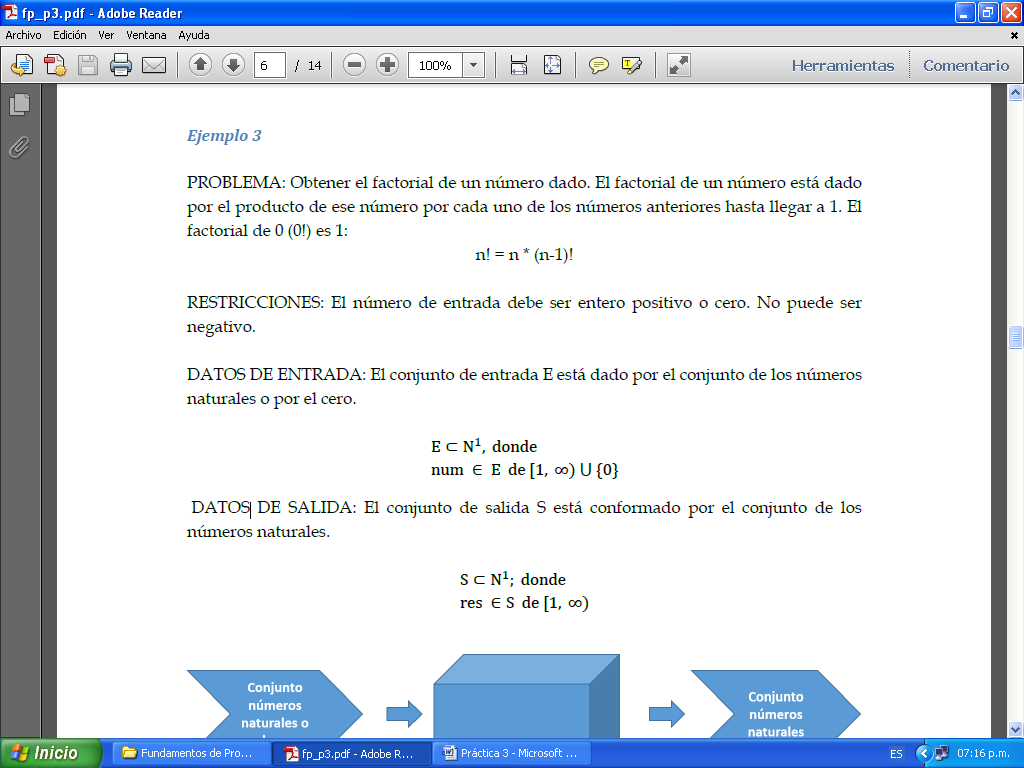
Datos de entrada:

X R

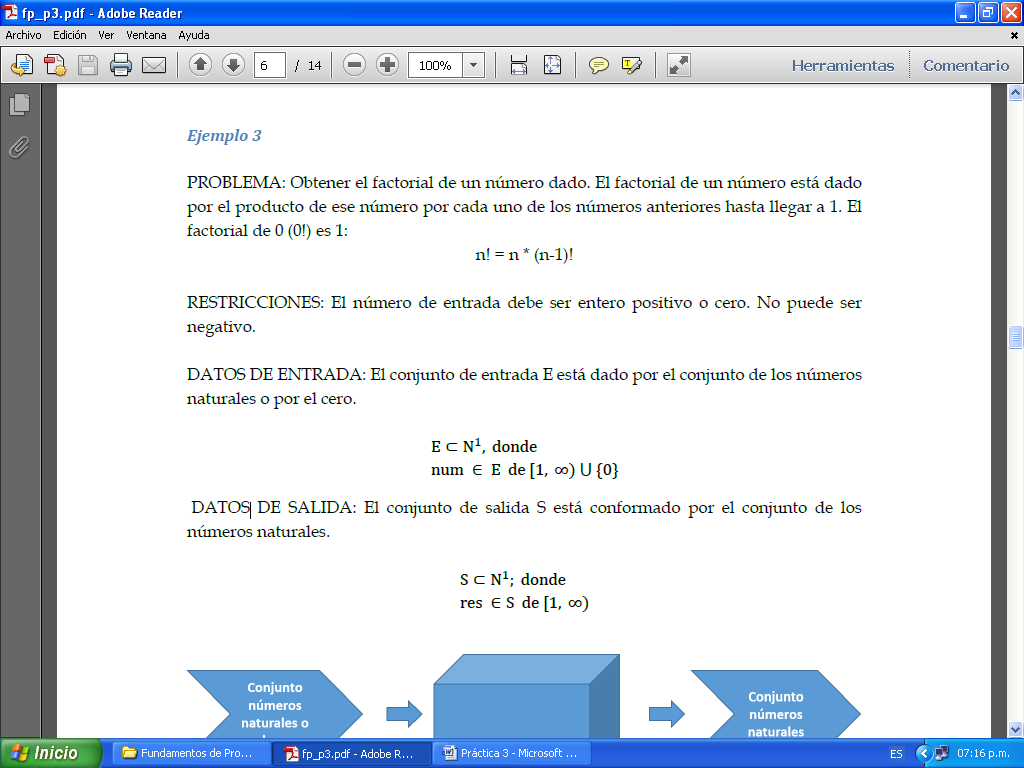
Y= par

1. 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.

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



Impresión del factorial del número



1. Fórmula General

Datos de entrada:

a , b , c

X1

X2

* Una vez realizada la actividad anterior, se procedió a diseñar o a crear un algoritmo para dar solución a cada problema, tomando en cuenta las características que debe cumplir un algoritmo.

1. Si resolver , Si resolver
2. Inicio
3. Pedir X
4. Si X es 2 ir al paso 4

3.1) Si X es 2 ir al paso 3.2

3.2) Resolver

3.3) Si imprime “no hay solución” e ir al paso 6

1. Resolver
2. Imprimir Y
3. Fin
4. Determinar si un número es par o impar
5. Inicio
6. Pedir valor de X
7. Realizar X/2

3.1) Si el residuo es 0, será un número par, pasar al paso 4.

3.2) De lo contrario imprime impar. Pasar al paso 5.

1. Imprimir “Es un número par”
2. Fin
3. Obtener el factorial de un número dado.
4. Inicio
5. Solicitar un número entero
6. Si el número entero es menor a cero regresar al punto 1.
7. Si el número entero es mayor a cero se crea una variable entera “contador” que inicie en 2 y una variable entera “factorial” que inicie en uno.
8. Si la variable contador es menor o igual al número entero de entrada se realiza lo siguiente:

5.1) Se multiplica el valor de la variable contador con el valor de la variable factorial. El resultado se almacena en la variable factorial.

5.2) Se incrementa en uno el valor de la variable contador.

5.3) Regresar al punto 4.

1. Si la variable contador no es menor o igual al número entero se muestra el resultado almacenado en la variable factorial.
2. Fin
3. Formula general
4. Inicio
5. Pedir los valores de a, b, c
6. Si 0. Si no ir al paso 4

3.1) Regresar al paso 2

1. Resolver
2. Resolver 1 2
3. Si 1 pasar a 6.1, si no a 7

6.1) Realizar 1 (-1)

6.2) Realizar

6.3) Realizar

1. Si W0 realizar 1
2. Realizar
3. Realizar X1
4. Realiza X2 - O s
5. Fin

* Al finalizar cada problema se tomó en cuenta, realizar la prueba de escritorio, para verificar que nuestro algoritmo sea el correcto.

1. Si resolver , Si resolver

|  |  |  |
| --- | --- | --- |
| Iteración | X | Salida |
| 1 | 2 | No hay solución |
|  | 3 | (3)2+3 (3)+20= 38 |
|  | 1 | 4 (1)2+2 (1)-50= - 44 |

1. Determinar si un número es par o impar

4 3

2 8 2 7

Impar

Par

1. 1
2. Obtener el factorial de un número dado.

4! = 4x3x2x1= 24

Número Entero: 4

Contador: 2+1=3+1=4+1=5

Factorial=1

Contador número entero 2 4?

Factorial= contador factorial 2 1=2

Factorial= 3 2= 6

Factorial=4 6= 24

1. Fórmula general.

1 4 (1) (1)

(-1)

X1

X2

**Conclusiones:**

Se aprendió a solucionar diferentes tipos de problemas, siguiendo los pasos indicados, analizarlos para identificar los datos de entrada y de salida, a base de lo que analizamos, crear un algoritmo para la solución de dicho problema, por último, para verificar que el algoritmo creado sea correcto, procedimos a realizar la prueba de escritorio, dando como resultado las soluciones a los problemas mostrados.