

UNIVERSIDAD COPPEL



actividad | 1 |

Calculadoras

Diseño de Interfaces 1

Ingeniería en Desarrollo de
Software

TUTOR: Aarón Iván Salazar Macías

ALUMNO: Gerardo Acosta Bustamante

FECHA:24/11/2024

Introducción.....	4
Descripción.....	5
Justificación.....	6
Desarrollo.....	7
Conclusión.....	10
Referencias.....	11

Introducción.

Las diferencias entre protocolo y algoritmo son bastante engañosas, a simple vista se puede creer que son lo mismo, pero su naturaleza es distinta, es decir; el protocolo dicta las reglas a seguir y cómo debe de realizarse algo, mientras que el algoritmo, determina el modo como eso se hace.

Por ejemplo: Me pueden decir ‘para caso de incendios deben seguir el protocolo’, entonces, el protocolo serían las reglas y cómo se debe evacuar el edificio: no tocar puertas calientes, seguir las flechas verdes, no empujarse. Ahora, el algoritmo define cómo cada humano cumplirá con las reglas establecidas por el protocolo, ejemplo: me levanto, giro a la derecha, avanzo hacia adelante, me formo en la fila...

Como nos podemos dar cuenta, el protocolo y el algoritmo, aunque similares, son dos cosas completamente distintas, a lo mejor, en esencia, puedan parecer iguales, pero no lo son. En resumen:

Protocolo: define las reglas generales.

Algoritmo: define las acciones a realizar paso a paso para poder cumplir con el protocolo.

Descripción

Esto es importante porque el algoritmo debe ser capaz de definir paso a paso cómo se harán las cosas, de ahí las diferencias entre los tipos de lenguajes de programación; por ejemplo, React, esta es una librería que permite desarrollar de manera más rápida y optima.

Si optamos por irnos a javascript puro y duro, nos encontraremos con que debo definir paso a paso cómo se debe hacer aquello que deseo, estaríamos tratando con el algoritmo en su máxima expresión, pero en React, no debo decir cómo hacerlo, sólo aquello que quiero que el programa haga, y es así que por un lado, debemos decir qué hacer y cómo hacerlo, por el otro, sólo el qué quiero, internamente React ya tiene predefinido los algoritmos que cumplen con lo que se quiere.

Por otra parte, a nuestro alrededor, podemos observar que los algoritmos también existen en la naturaleza, y esto es, cuando encontramos patrones en un proceso natural, podemos definir el algoritmo que define a los procesos mediante sucesión de acciones que conlleven a un lugar en concreto.

Justificación

En esta ocasión, se abordará el tema de las calculadoras, que lo que hacen es precisamente eso, calcular; como en todo programa existen tres partes fundamentales, entrada – proceso – salida-Nosotros introducimos un dato, y esperamos una salida/respuesta. Ahora:

Contextualización: La empresa MathTech requiere a un ingeniero en desarrollo de software que sea capaz de realizar la tarea de programar tres tipos de calculadoras diferentes para implementar en los colegios y escuelas públicas:

- La primera calculadora deberá de llevar por nombre Primos, y su objetivo será identificar los números primos que se ingresen, por ejemplo si el usuario ingresa el número 83, deberá imprimir el siguiente mensaje: “El número (número ingresado) si es primo”, en caso de que no sea primo se imprimirá el siguiente mensaje “El número (número ingresado) no es primo”. Básicamente se encargará de identificar si un número es divisible entre 1 y el mismo.
- La segunda calculadora se llamará Par/Impar, su objetivo es que se ingresen 10 números, ya sean pares o impares, por ejemplo si se ingresa el número 9, el programa deberá de indicar que es un número impar, pero si se trata del número 2, el programa deberá indicar que se trata de un número par. De 10 números enteros, se debe determinar cuáles son pares y cuáles son impares.
- El último programa se llamará Al Revés, su objetivo es que el usuario ingrese un número de 4 dígitos y que sea un número entero, y este programa se encargará de regresar los número al revés o invertidos. Por ejemplo, si se ingresa el número 7631, el programa matemático deberá regresar 1367.

Desarrollo

Primos:

```

1  Algoritmo Primos
2      Escribir "hola soy Primos"
3      Escribir "Ingeresa un número"
4      Leer N;
5      vueltas ← 1;
6      primo ← 0;
7      Mientras vueltas ≤ N Hacer
8          Si N % vueltas == 0 Entonces
9              Escribir "dividido por " vueltas;
10             primo ← primo + 1;
11         Fin Si
12         vueltas ← vueltas +1;
13     Fin Mientras
14     si (primo == 2) Entonces
15         Escribir "es primo"
16     SiNo
17         Escribir "no es primo "
18     FinSi
19
20 FinAlgoritmo
21
22

```

```

PSeInt - Ejecutando proceso PRIMOS
*** Ejecución Iniciada. ***
hola soy Primos
Ingeresa un número
> 12
dividido por 1
dividido por 2
dividido por 3
dividido por 4
dividido por 6
dividido por 12
no es primo
*** Ejecución Finalizada. ***
☐ No cerrar esta ventana ☒ Siempre visible Reiniciar

```

Aquí podemos apreciar el algoritmo, primero saluda, luego solicita información, la procesa y finalmente ocurre una salida con un resultado. En este caso evalúa si la cantidad de vueltas es igual a N (número que nos dio el usuario), si dice que no, entonces da una vuelta más, hace una serie de evaluaciones y de paso aumenta en uno el valor de vueltas, y así en cada iteración.

The image shows a pseudocode editor on the left and a terminal window titled 'PSeInt - Ejecutando proceso CALCULAR_PAR_IMPAR' on the right. The pseudocode is for a program to check if a number is even or odd and to ask if the user wants to exit.

```

1  Algoritmo Calcular_Par_impar
2  salir ← 0;
3  contador ← 0;
4
5  Repetir
6  |   Escribir "Hola, introduce un número y te diré si es un número par o es impar"
7  |
8  |   leer N
9  |   Si (N mod 2) = 0 Entonces
10 |       Escribir "Es par"
11 |   SiNo
12 |       Escribir "Es impar"
13 |   Fin Si
14 |   Escribir "¿Desea salir? 1 / 0 "
15 |   Leer respuesta
16 |   si respuesta ≥ 1 Entonces
17 |       salir ← 1
18 |   FinSi
19 |
20 |   contador ← contador +1
21 |   Hasta Que salir = 1 o contador = 10
22
23 FinAlgoritmo
24

```

The terminal window shows the execution of the program. It prompts the user to enter a number and checks if it is even or odd. It also prompts the user to enter a response to whether they want to exit (1 for yes, 0 for no). The terminal shows three iterations of the loop.

```

PSeInt - Ejecutando proceso CALCULAR_PAR_IMPAR
Hola, introduce un número y te diré si es un número par o es impar
> 12
Es par
¿Desea salir? 1 / 0
> 0
Hola, introduce un número y te diré si es un número par o es impar
> 23
Es impar
¿Desea salir? 1 / 0
> 0
Hola, introduce un número y te diré si es un número par o es impar
> 84
Es par
¿Desea salir? 1 / 0
> |

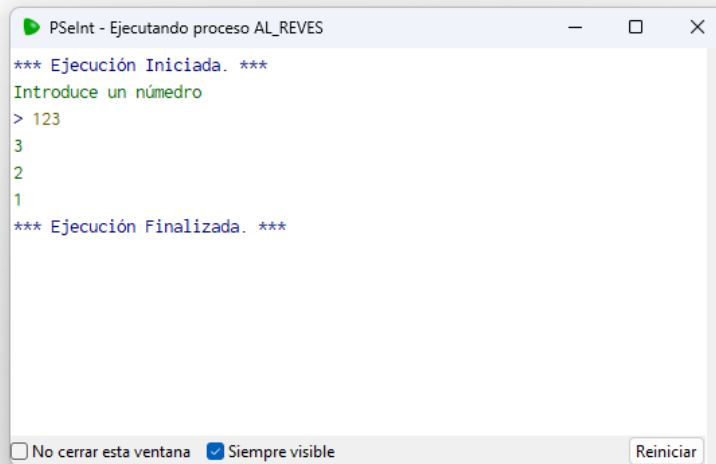
```

En este segundo escenario hacemos uso de ‘mod’ que se usa para representar el resto de una división, es un operador universal el cual es mejor usarlo en pseudocódigo ya que así se conceptualiza más y se evita llevar malas prácticas, esto porque dependiendo del lenguaje de programación tal vez el ‘%’ se comporte de forma diferente; en resumen, así se diferencia exactamente lo que se espera ‘el resto de una división’


```

1  Proceso al_revés
2
3  Escribir "Introduce un número"
4  leer NumeroUsuario
5  iterador ← 0;
6  Dimension cantidadNumeros[10];
7  Mientras NumeroUsuario > 0 Hacer
8      iterador ← iterador + 1
9      digito ← NumeroUsuario mod 10
10     Escribir digito
11     cantidadNumeros[iterador] ← digito;
12     NumeroUsuario ← Trunc(NumeroUsuario / 10)
13  Fin Mientras
14 FinProceso
15

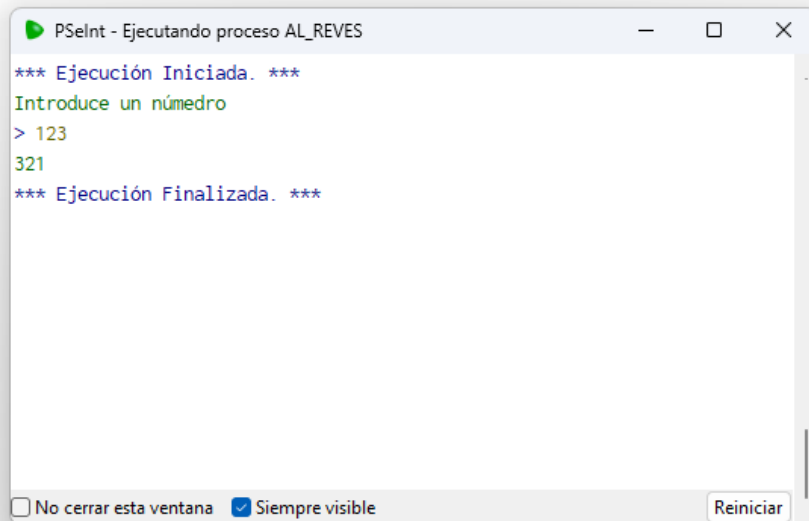
```



```

1  Proceso al_revés
2      inverso ← 0;
3  Escribir "Introduce un número"
4  leer NumeroUsuario
5  residuo ← NumeroUsuario;
6  Mientras residuo > 0 Hacer
7      inverso ← inverso * 10 + residuo mod 10;
8      residuo ← (residuo - residuo mod 10) / 10;
9  Fin Mientras
10
11  escribir inverso
12
13
14 FinProceso
15
16
17
18
19

```



Para este último hice dos programas, uno siguiendo el tutorial de la profesora, y el otro un intento de encontrar una solución alternativa.

Conclusión

A forma de conclusión; podemos observar que con esta herramienta es posible plasmar en un lienzo, nuestra idea de algoritmo necesario para determinadas operaciones según convenga.

Fue una actividad que me gustó, y que sobre todo es útil en la vida diaria incluso o más en el ámbito laboral, porque en todo encontramos un algoritmo o un patrón de comportamiento.

Hay un tema que en lo personal me llamó la atención, fue la recursividad en los algoritmos, un algoritmo recursivo me gustaría saber cómo se plantea en PSE INT, imagino debe ser sencillo.

Por otro lado, se trata de un buen inicio como introducción al Desarrollo de software, pues permite desde un principio ir creando algoritmos y programas que es la parte más satisfactoria del desarrollador.

Referencias

Video conferencing, web conferencing, webinars, screen sharing. (s. f.-e). Zoom.

<https://academiaglobal->

mx.zoom.us/rec/play/Mys68EE3_atOF69vGTh6W1tDK5O9X4EA6ovWSKA2FxOBKs4

0HUcxlKB38R8u4SzWC79V-

RR3xufe5KhL.vTWWOqe8H0YFa_8v?canPlayFromShare=true&from=share_recordin

g_detail&continueMode=true&componentName=rec-

play&originRequestUrl=https%3A%2F%2Facademiaglobal-

mx.zoom.us%2Frec%2Fshare%2FOzshr-

wKHvdk9Zunfj35sIXc59IvLTtk6G9B2a98IHe35g92wPysCHeOK1Uo0bY.KX_4DWdDS

iW9Gk_U