



Actividad | 1 | Algoritmos

Introducción al Desarrollo de Software

Ingeniería en Desarrollo de Software



TUTOR: Dra. Sandra Luz Lara Dévora

ALUMNO: Bruno Alberto German Meza

FECHA: 03 de diciembre de 2024

ÍNDICE

Introducción.....	3
Descripción.....	3
Justificación.....	3
Desarrollo	4
Calculadora “Números Primos”	4
Calculadora “Números par o impar”	5
Calculadora “Al revés”	6
Conclusión.....	7
Referencias	7

Introducción

En esta actividad lo que se va a realizar es enfocado en los algoritmos, esta primera semana nos enfocaremos en hacer 3 pseudocódigos con la aplicación que deseemos, pero preferentemente con pseint que nos da todas las herramientas para armar un pseudocódigo de forma sólida y clara. Los tres ejercicios que desarrollaremos son;

- 1.- Hacer una calculadora que nos ayude a identificar números primos
- 2.-Hacer una segunda calculado donde nos mostrara si un número ingresado es par o impar.
- 3.- Por último, haremos una calculadora donde la finalidad es revertir 4 números ingresados por el usuario, por ejemplo, el usuario da 1234 nosotros tenemos que devolver 4321.

Son ejercicios simples, pero clave a la hora de aprender lógica de programación, estas calculadoras nos serán de aprendizaje para en un futuro entender códigos mucho más complejos. También se explicará en cada una de las calculadoras como se obtuvo el resultado y detallar los pasos para que el maestro pueda entender con mayor exactitud los pasos que realizaste y el por qué.

Descripción

Se nos plantea que cierta empresa necesita de un ingeniero en desarrollo de software que tenga las capacidades para realizar tres calculadoras diferentes para implementar en colegios y escuelas públicas.

Es super importante que un desarrollador de software tenga una buena lógica a la hora de desarrollar. Siento que esas 3 calculadoras son ejercicios muy clave que nos enseñan lo básico y a lo ves lo mas importante que vamos a necesitar en el futuro.

Referente a las calculadoras, sí son importantes porque muchas veces nos sacan de un apuro, y existen infinidad de calculadoras sobre cualquier cosa en el mundo del internet, así como para recetas de cocina, escuela, deportes, videojuegos, etc.

Por ende, es importante saber crear calculadoras de todo tipo. Yo personalmente he usado calculadoras referentes a videojuegos, para medir mi índice corporal, sobre calorías que debo ingerir, sobre a que ritmo cardiaco debo correr para quemar más grasa, infinidad de calculadoras se pueden crear y se crean para facilitar las tareas de las personas.

Justificación

Como lo comenté anteriormente siento que estos ejercicios que son básicos nos ayudaran a tener una lógica de programación mas amplia, porque aprendimos a usar variables, bucles, a usar tipo de dato Float que nos permite poner decimales, funciones, etc.

Todo esto que se podría decir que son ejercicios básicos, son de un aprendizaje muy duro, ya que te da las bases para poder continuar en la programación y a la vez poder resolver problemas sencillos. Ahorita resolvimos todos estos ejercicios con pseudocódigo mediante pseint pero mas adelante tendremos que realizar los códigos con lenguaje de programación C.

Debemos dominar los ciclos while, for, if else, etc. Juntos con las variables y las constantes para tener un abanico mas amplio de herramientas a la hora de hacer código, así que estas 3 calculadoras nos sirvieron mucho de practica para poner a prueba nuestra lógica y razonamiento, lo importante es entender el código y saber explicarlo no solo copiarlo de algún lugar y ya, y yo siento que me quedo muy marcado cada uno de los pasos para hacer las calculadoras y también me tome el tiempo de comentar lo que iba haciendo mientras se ponía el código.

Desarrollo

Se desarrollaron mediante pseudocódigo las 3 calculadoras siguientes:

Calculadora “Números Primos”

The screenshot displays the PSeInt software interface. The main window shows the pseudocode for the 'Números Primos' algorithm. The code is as follows:

```
1 Algoritmo Primos
2 //Pedir el numero
3 Escribir "Dígame un número entero: ";
4 //Ahora lee la variable que ingresara el usuario
5 Leer digito;
6 //marcar variable para hacer un ciclo
7 repeticion = 1;
8 residuo = 0;
9 //Se utiliza el While para repetir los mismos pasos hasta que se cumpla cierta función.
10 //siempre sumando 1 en repetición hasta llegar el numero dado.
11 //Si el numero divisible de 0 en mas de 3 ocasiones significa que No es primo
12 //Pero si solo fue divisible 2 siempre dejando un residuo de 0 significa que es primo
13 Mientras repeticion ≤ digito Hacer
14     Si (digito % repeticion == 0) Entonces
15         residuo = residuo + 1;
16     Fin Si
17     repeticion = repeticion + 1;
18 Fin Mientras
19 Si (residuo == 2) Entonces
20     Escribir "El número SÍ es primo";
21 SiNo
22     Escribir "El número NO es primo";
23 Fin Si
24 Escribir " ";
25 Escribir "Elaborado por Bruno German Meza";
26 FinAlgoritmo
```

Overlaid on the main window is a smaller window titled 'PSeInt - Ejecutando proceso PRIMOS'. It shows the execution output:

```
*** Ejecución Iniciada. ***
Dígame un número entero:
> 3
El número SÍ es primo

Elaborado por Bruno German Meza
*** Ejecución Finalizada. ***
```

At the bottom of this execution window are checkboxes for 'No cerrar esta ventana' and 'Siempre visible', and a 'Reiniciar' button.

On the right side of the PSeInt interface is a 'Comandos' (Commands) panel. It contains icons for various operations: 'Escribir' (Write), 'Leer' (Read), 'Asignar' (Assign), 'Si-Entonces' (If-Then), 'Según' (Select), 'Mientras' (While), 'Repetir' (Repeat), 'Para' (For), and 'Función' (Function).

At the very bottom of the screenshot, a status bar message reads: 'La ejecución ha finalizado sin errores.'

Se pide el número, se lee la variable que ingresa el usuario, se utiliza la estructura de while “mientras” para repetir los pasos hasta que se cumpla la condición sumando 1 en la repetición para llegar al número ingresado, si el número de 0 en mas de 2 ocasiones significa que no es primo, pero si solo de 0 en 2 ocasiones significa que es primo, porque solo se dividió entre 1 y entre su propio número.

Calculadora "Números par o impar"

The screenshot displays the PSeInt software interface. The main window shows a Pascal program titled "Algoritmo ParoEmpar". The program uses a loop to iterate 10 times, prompting the user to enter a number. It checks if the number is even (divisible by 2) and prints the corresponding message. The program concludes with a thank you message and a final execution status.

```
1 Algoritmo ParoEmpar
2 //Primero marcamos la variable que vamos a reiterar hasta las 10 ocaciones.
3 //Utilizamos un comando Repetir "For si no me equivoco" para poder pautar las reiteraciones que ha
4 //Vamos aumentando en 1 la variable num por cada vuelta.
5 //Escribimos al usuario que ingrese un dígito el cual te dira si es par o impar
6 //gracias a una expresión lógica que dividira cada numero introducido entre 2, y si da 0 significa
7 num = 0;
8 Repetir
9     num = num + 1;
10    Escribir "Digite un número: "
11    Leer numero;
12    Si (numero % 2) == 0 Entonces
13        Escribir "Este número es par";
14    SiNo
15        Escribir "Este número es impar";
16    Fin Si
17 Hasta Que num = 10
18 Escribir "Hecho por Bruno Alberto German Meza,Gracias Maestra :D"
19
20 FinAlgoritmo
```

The console window titled "PSeInt - Ejecutando proceso PAROEMPAR" shows the following output:

```
Digite un número:
> 9
Este número es impar
Digite un número:
> 10
Este número es par
Hecho por Bruno Alberto German Meza,Gracias Maestra :D
*** Ejecución Finalizada. ***
```

At the bottom left, a status bar indicates: "La ejecución ha finalizado sin errores."

Primero tenemos que marcar una variable que se va a repetir hasta 10 veces, usamos la estructura de repetir y hacemos que la variable se vaya modificando con cada reiteración con de esta forma "num = num + 1" para ir sumando 1 a la variable num cada que inicie el ciclo hasta llegar a 10. Lo demás es simple si el numero deja un residuo de 0 es par, si no es impar. El ciclo se detendrá hasta que la variable num llegue a 10 porque pusimos que for termine cuando num = 10 "lo que significa que 10 veces paso por el ciclo".

Calculadora "Al revés"

The screenshot displays the PSeInt IDE with a Pascal program titled "Algoritmo AlRevés". The program uses a While loop to reverse a 4-digit number by repeatedly dividing by 10 and storing the remainder in an 'inverso' variable. The execution window shows the program running with the input 1234, resulting in the reversed number 4321.

```
1 Algoritmo AlRevés
2 //La importancia esta en saber encontrar el residuo.
3 //Al hacer la división por 10, deja un residuo.
4 //ese dígito final se va quedando en la variable residuo hasta
5 //que la iteración llega a 0 y no se repite mas.
6 //Dejando la variable inverso con los digitos residuos de cada iteración.
7 inverso = 0;
8 Escribir "Ingresa un número de 4 dígitos";
9 leer numero;
10 residuo = numero
11 Mientras residuo > 0 Hacer
12     inverso = inverso * 10 + residuo mod 10;
13     residuo = (residuo - residuo mod 10) / 10;
14 Fin Mientras
15 escribir "Valor del numero inverso es: ", inverso;
16 Escribir " El numero antes era: ", numero;
17
18 Escribir "Realizada por Bruno Alberto German Meza";
19 FinAlgoritmo
```

Comandos:

- Escribir
- Leer
- Asignar
- Si-Entonces
- Según

PSeInt - Ejecutando proceso ALREVES

```
*** Ejecución Iniciada. ***
Ingresa un número de 4 dígitos
> 1234
Valor del numero inverso es: 4321
El numero antes era: 1234
Realizada por Bruno Alberto German Meza
*** Ejecución Finalizada. ***
```

☐ No cerrar esta ventana ☐ Siempre visible

La ejecución ha finalizado sin errores.

Para mí fue la más difícil, pero es mas de lo mismo, ingresas una cantidad que será el numero ingresado = residuo y se va guardando el residuo verdadero de esta en una variable llamada inverso, por ejemplo 1234 dividido entre 10 es 4, se guarda en la variable inverso. Después al mismo número residuo se le resta el residuo "4" y se divide en 10 para terminar con una variable residuo de 123, de esta manera se volverá a repetir el ciclo multiplicando por 10 el inverso para pasarlo a decenas y este ciclo se repetirá hasta que la variable residuo sea 0 (Recordar que el ciclo While es mientras residuo sea mayor a 0 continuar) para ese entonces la variable inverso ya tendrá los 4 dígitos tomados de los residuos. Y cuando se termine en 0 la variable residuo escribirá la variable inverso junto con la variable **número** (variable que se ingresó inicialmente que después se convirtió en la variable residuo pero al estar afuera del ciclo While no pierde su valor original).

Conclusión

Personalmente yo antes había tomado un pequeño curso de lógica de programación, porque no soy muy bueno en esto, pero es algo que me llama la atención demasiado desde que era un jovencito. Estos algoritmos son básicos, pero nos sirven para ir identificando y saber leer códigos, si nunca haz hecho algoritmo incluso el pseudocódigo puede ser difícil de entender, gracias a los ejercicios explicados por la maestra logre entender la lógica detrás de ellos que es lo mas importante, no solo copiarlo, sino entenderlo y tratar de darle tu personalidad, añadiéndole o modificándole.

Sí bien esto es un simple pseudocódigo, es el primer paso para programar en un lenguaje de programación real y estos ejercicios nos darán las bases para poder adentrarnos más en el amplio mundo de la programación y la lógica de esta.

Esta actividad sirve mucho en la vida cotidiana, aunque no parezca, al fin y al cabo, nuestras actividades que realizamos en el día son como si fuera un código de programación.

Ejemplo, 1.- levantarse 2.- lavarse dientes, 3.- ducharse, etc. Se puede alterar el código cotidiano dependiendo tus necesidades.

En el campo laboral es super importante esta actividad realizada, nos da las bases para entender la lógica de los algoritmos, lo cual más adelante nos adentraremos en lenguajes de programación donde todo tendrá una sintaxis más complicada o específica. A futuro con mucho esfuerzo y dedicación puedes lograr ser un programador en alguna empresa reconocida o por tu propia cuenta.

Referencias

PSEInt. (s. f.). <https://pseint.sourceforge.net/>

<https://github.com/GermanMezaBruno/UMI-repositorio>