

SinglePath games design



Versión 3.1

Acerca del Autor

Introducción

1.- ¿Qué es la programación?

2.- Algoritmos

3.- Secuencias de Control

4.- Secuencia de Control Avanzada

5.- Métodos

6.- Secuencias repetitivas

7.- Arreglos o Matrices

8.- Lenguajes de Programación

9.- Ejemplos Algorítmicos

- 9.1- Qbasic
- 9.2- Pascal
- 9.3- C++
- 9.4- Euphoria
- 9.5- Java
- 9.6- JavaScript
- 9.7- Delphi
- 9.8- PHP
- 9.9- C#
- 9.10- VB.NET
- 9.11- Python

10.- ¿Cómo ser un buen programador?

11.- Comentarios Finales

12.- Enlaces

13.- Agradecimientos

14.- El Fin

Acerca del Autor



Mi Nombre es Alvaro Tejada, soy de Lima, Perú, tengo 26 años y junto con mi novia Milly, estudio Computación e Informática. Soy experto de la categoría de programación en Xpertia, bajo el pseudónimo de Blag. Y dueño de mi propia "compañía" (Ficticia por el momento) desarrolladora de software multimedia, SinglePath games design. (La cual espero poder fundar algún día no muy lejano).

Mis inicios en la programación datan aproximadamente del año 1991, cuando vivía en México, Distrito Federal. Allí lleve un curso de introducción al Pascal, aunque claro, en esos tiempos a pesar de que me llamaban la atención las computadoras, no tenía claro que quería hacer. Fue recién cuando vi por primera vez el juego Mortal Kombat de Midway que decidí que quería ser programador. En ese entonces tenía, si no me equivoco, una Macintosh Plus con Think Pascal. Para ser sinceros, no entendía nada y el único programa que logré hacer fue uno que venía como ejemplo en el manual de usuario y que nunca pude modificar. Luego el tiempo pasó y en el año 1997 ingresé a la Universidad de Lima en la cual llevé el curso de "Técnicas de Programación I", en el cual tuve mi primer contacto con C++, aunque aún no con los resultados que yo esperaba (Siempre la programación es difícil al principio). Fue a principios de 1998, en febrero más o menos, que mi papá me compró un libro de Visual Basic "Visual Basic 6 Bible" y uno de C++ "Teach Yourself C++ in 21 Days". Y así comencé mi carrera en el mundo de la programación. No voy a negar que al principio me costó bastante y que mis primeros programas eran bastantes simples, pero sentía la satisfacción de haberlos hecho yo mismo.

El tiempo pasaba y me retiré de la Universidad de Lima, para ingresar a Cibertec, en donde conocí por primera vez lo que era el lenguaje "Java". Ya una vez que mis conocimientos de programación habían aumentado, y mis programas y juegos mejorado, decidí explorar otros lenguajes de programación, como Visual C++, Euphoria, Visual FoxPro, QBasic, SQLServer, VBA, JavaScript, HTML, dBase III plus y Pascal. Actualmente, trabajo como Consultor en diversos lenguajes (SAP, VB, PHP). Hasta ahora he desarrollado cerca de 32 programas y 10 juegos en distintos lenguajes, los cuales pueden descargarse gratuitamente de SinglePath games design. - <http://www.iespana.es/singlepath>

Introducción

Este libro, no pretende enseñar a programar, sino más bien, dar algunas pautas necesarias para comenzar a aprender. Algunos conceptos básicos que se aplican a todos los lenguajes de programación, una breve historia de ellos, enlaces a páginas con códigos, tutoriales, preguntas a expertos, etc. Este libro pretende dar a conocer lo que es el mundo de la programación y que es lo que necesita saber o tener para comenzar a programar. Si quieres aprender a programar desde cero, pero no sabes con que lenguaje comenzar, entonces este libro puede ayudarte, si ya programas, talvés puedas encontrar consejos útiles y si eres un programador experto, puede interesarte lo que otro programador experto tiene que decir.

1.- ¿Qué es la programación?

Alguna persona que no este relacionada con el mundo de la computación, o que simplemente no haya escuchado nunca esto, podría preguntarse, ¿Qué es la programación?

Pues bien, la programación es el proceso por el cual, un problema o una situación, puede ser automatizada para ser usado con la computadora y así hacer las cosas más fáciles.

¿Pero como lograr que la computadora haga las cosas que yo necesito?, pues muy simple, escogiendo algunos de los lenguajes de programación existentes, que de contarlos, llegarían aproximadamente a unos 600. Estos lenguajes de programación son programas que permiten alterar o manejar actividades de la computadora. Estos lenguajes por supuesto han ido evolucionando con el tiempo, y cada vez son más sencillos de utilizar.

Por ejemplo, una persona puede necesitar un programa que le ayude a llevar un inventario de cuantos libros tiene en su biblioteca. Esto podría hacerse con cualquier lenguaje de programación, ya sea, utilizando el acceso a base de datos, con matrices o arreglos, con archivos de texto, etc.

La programación abarca muchos aspectos, que van desde el diseño del algoritmo o pseudo-código, diseño físico del programa, testeo y depuración y finalmente su distribución.

Cada uno de estos pasos son muy importantes y no deben obviarse para así poder lograr un programa que funcione y sea fácil de actualizar. En el capítulo siguiente, empieza de verdad el libro, con una introducción a lo que son los algoritmos de programación y sus sentencias más importantes.

2.- Algoritmos

Como una definición básica, podemos decir que los algoritmos de programación, son un conjunto ordenado y finito de pasos que conducen a la solución de un problema.

La mayor ventaja de utilizar estos algoritmos de programación, es que no están diseñados para un lenguaje de programación específico, sino que una vez desarrollado el algoritmo, puede convertirse en un programa sin importar el lenguaje que se quiera utilizar.

Un algoritmo de programación o pseudo-código, empieza y termina con las palabras Inicio y Fin respectivamente. Y utilizan palabras de lenguaje común como leer, imprimir, Hacer, Si...Sino, etc. Es por esto, que su comprensión es más sencilla. Además, puede contener símbolos matemáticos como +, -, *, /, >, <, ==, etc.

Como punto importante, debo mencionar que en el caso de los algoritmos de programación la igualdad se escribe == y = significa asignación, es decir si yo quiero guardar 5 en la variable día, escribiría día = 5. Y si yo quisiera comparar los valores de A y B, haría lo siguiente A == B. Además la diferencia se escribe como !=, por ejemplo, A != B.

En los algoritmos de programación, existen dos operadores para la división, uno de ellos, llamado divisor entero "/" y el otro módulo de la división o residuo "%".

La diferencia entre ellos, es que como su nombre lo indica, el divisor entero solo toma la parte entera de la división, es decir que 5/2, sería igual a 2.

Por otro lado, el módulo de la división o residuo de 5%2, sería igual a 1.

A continuación detallo los tipos de variables que se utilizan en un algoritmo de programación o pseudo-código.

- **Variable:** Una variable, es un espacio en memoria al que se le asigna un valor que va a ser utilizado a lo largo del programa y que puede ser modificado.

Tipo de dato	Descripción
Entero	Dato numérico sin punto decimal
Real	Dato numérico con punto decimal
Carácter	Un solo carácter alfanumérico
Cadena	Conjunto de caracteres
Lógico	Verdadero o falso

- Para declarar una variable, se comienza con el tipo de variable, y luego un nombre que la identifique, que puede ser cualquier nombre válido aunque se recomienda que represente al valor que va a almacenar.

Entero edad
Real sueldo
Cadena nombre

Ejemplo de un algoritmo de programación que suma dos numero ingresados:

En este ejemplo, el usuario ingresa 2 números Enteros, los cuales son leídos por el programa. Luego de lo cual son sumados, para poder obtener la respuesta y mostrarla en la pantalla.

```
Inicio //Comienzo del algoritmo
Entero numero1,numero2,suma //Declaración de variables, en este caso,
//son 3 variables de tipo entero
Leer numero1,numero2 //Asignamos los valores ingresados a las variables
suma = numero1 + numero2 //Guardamos la suma de los números
//en la variable suma
Imprimir suma //Imprimimos en pantalla el resultado
Fin //Término del algoritmo
```

Ejemplo de un algoritmo de programación que pide un nombre y edad:

En este ejemplo, declaramos dos variables, una de tipo Entero y la otra de tipo Cadena (Para textos), puesto que el usuario ingresará su nombre y su edad y el programa le devolverá un saludo.

```
Inicio //Comienzo del algoritmo
Entero edad //Declaración de variables
Cadena nombre //Declaración de variables
Leer edad,nombre //Asignamos los valores ingresados a las variables
Imprimir "Hola " + nombre + " tienes " + edad + " años"
```

```
//Imprimimos en pantalla el resultado, el símbolo "+" nos sirve para poder  
//imprimir texto y variables en una sola línea  
Fin //Término del algoritmo
```

Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

En este ejemplo, declaramos dos variables de tipo Real (Es decir, que acepta decimales) y variable de tipo Entero. Leemos los valores ingresados por el usuario, "Cantidad" y "Precio" y simplemente los multiplicamos, para poder obtener el precio a pagar.

```
Inicio //Comienzo del algoritmo  
Real totalpagar,precio //Declaración de variables  
Entero cantidad //Declaración de variables  
Leer cantidad,precio //Asignamos los valores ingresados a las variables  
totalpagar = cantidad * precio //Guardamos la multiplicación de la cantidad  
//por el precio en la variable totalpagar  
Imprimir "Tienes que pagar " + totalpagar  
//Imprimimos en pantalla el resultado  
Fin //Término del algoritmo
```

3.- Secuencias de Control

Las secuencias de control, son aquellas que controlan el flujo del algoritmo. Y pueden verificar si una condición se cumple, o el rango de valores que puede tomar.

Estas secuencias de control, son lo más usado a la hora de diseñar un programa, puesto que ofrecen gran control sobre el desempeño que va a tener nuestra aplicación final y permiten reducir los errores.

- **Selección "Si"**: Esta secuencia de control es la más sencilla y verifica si una condición dada es verdadera o falsa, y en caso de que sea verdadera, realizará una acción. Cuando la acción es de más de una línea, debe estar entre {}.

```
Si(condición)
    acción
Si(condición)
{
    acción1
    acción2
    acción3
}
```

Ejemplo de un algoritmo de programación que pregunta si es fin de semana:

En este ejemplo, el usuario va a ingresar el nombre de un día de la semana, el cual se guardará en la variable día, que es de tipo Cadena, es decir, texto. En este caso, el día que nos gusta, es el "Domingo", por eso, deberemos saber si el usuario ha ingresado o no la palabra "Domingo". Utilizaremos el comando "Si" para poder hacer la validación.

```
Inicio //Comienzo del algoritmo
Cadena dia //Declaración de variables
Leer dia //Asignamos los valores ingresados a las variables
Si(dia == "Domingo") //Verifica si la condición se cumple
```

```

Imprimir "Me gusta el domingo" //Imprimimos en pantalla el resultado
Si(dia != "Domingo") //Si día no es Domingo
{
    imprimir "Avisame cuando sea domingo"
    imprimir "Me gustan mas los domingos"
}
Fin //Término del algoritmo

```

Ejemplo de un algoritmo de programación que determina el mayor y el menor de 3 números:

En este ejemplo, simplemente, leemos los 3 números ingresados por el usuario. Al primer número lo tomamos como si fuera el mayor, sin importarnos que número es. Una vez hecho esto, deberemos compararlo con el segundo para determinar cual es el mayor y una vez que lo hemos obtenido, hacemos lo mismo con el tercer número. Así obtendremos tanto el mayor como el menor.

```

Inicio //Comienzo del algoritmo
Entero n1,n2,n3,mayor,menor //Declaración de variables
Leer n1,n2,n3 //Asignamos los valores ingresados a las variables
mayor = n1
//Asignamos momentáneamente a la variable mayor el valor de n1
Si(n2 > mayor) //Si se cumple la condición
    mayor = n2 //El nuevo mayor es n2
Si(n3 > mayor) //Si se cumple la condición
{
    mayor = n3 //El nuevo mayor es n3
    menor = n1
//Asignamos momentáneamente a la variable menor el valor de n1
}
Si(n2 < menor) //Si se cumple la condición
    menor = n2 //El nuevo menor es n2
Si(n3 < menor) //Si se cumple la condición
    menor = n3 //El nuevo menor es n3

```

```

Imprimir mayor,menor //Imprimimos en pantalla el resultado
Fin //Término del algoritmo

```

Ejemplo de un algoritmo de programación que determina el día de la semana:

Este ejemplo, determina el día de la semana, de acuerdo al número que ingrese el usuario. Simplemente, deberemos preguntar que número es e imprimir el nombre del día adecuado.

```

Inicio //Comienzo del algoritmo
Entero dia //Declaración de variables
Leer dia //Asignamos los valores ingresados a las variables
Si(dia >= 0 && dia <= 7) //Si se cumple la condición, 0<=dia<=7
    Si(dia == 1) //Si dia es 1
        Imprimir "Es Lunes"
    Si(dia == 2) //Si dia es 2
        Imprimir "Es Martes"
    Si(dia == 3) //Si dia es 3
        Imprimir "Es Miércoles"
    Si(dia == 4) //Si dia es 4
        Imprimir "Es Jueves"
    Si(dia == 5) //Si dia es 5
        Imprimir "Es Viernes"
    Si(dia == 6) //Si dia es 6
        Imprimir "Es Sábado"
    Si(dia == 7) //Si dia es 7
        Imprimir "Es Domingo"
Si(dia < 0 || dia > 7) //Si el día esta fuera del rango
    Imprimir "Error, ingrese un número del 1 al 7"
Fin //Término del programa

```

- **Selección "Si-Sino":** Esta secuencia de control, verifica si una condición es verdadera o falsa, "si" es verdadera, ejecuta una acción, "sino", ejecutará otra acción. Cuando la acción es de más de una línea, debe estar entre {}.

```

Si(condición)
    acción1
Sino
    acción2
Si(condición)
{
    acción1
    acción2
    acción3
}
Sino
{
    acción1
    acción2
    acción3
}

```

Ejemplo de un algoritmo de programación que pregunta si es fin de semana:

En este ejemplo, preguntaremos al usuario por un día de la semana. Si el día elegido es "Domingo", imprimiremos un mensaje. En el caso de que no haya ingresado "Domingo", simplemente diremos...."Sino se cumple la condición....".

```

Inicio //Comienzo del algoritmo
Cadena dia //Declaración de variables
Leer dia //Asignamos los valores ingresados a las variables
Si(dia == "Domingo") //Verifica si la condición se cumple
Imprimir "Me gusta el domingo" //Imprimimos en pantalla el resultado
Sino //Si dia no es Domingo
Imprimir "Avisame cuando sea domingo"
Fin //Término del algoritmo

```

Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```

Inicio //Comienzo del algoritmo
Entero numero //Declaración de variables
Leer numero //Asignamos los valores ingresados a las variables
Si(numero % 2 == 0) //Verifica si la condición se cumple
Imprimir "Es par" //Imprimimos en pantalla el resultado
Sino //Si el residuo de numero % 2 no es 0
Imprimir "Es impar"
Fin //Término del algoritmo

```

- **Selección "Si-Sino-Si"**: En este caso, se evalúa una primera condición, si es verdadera, se ejecuta una acción, sino, se verificará una nueva condición. Cuando la acción es de más de una línea, debe estar entre {}.

```

Si(condición)
    acción1
Sino
    Si(condición2)
        acción2

```

Ejemplo de un algoritmo de programación que determina el día de la semana:

En este ejemplo, determinaremos de acuerdo al número que ingrese el usuario, el día de la semana correspondiente. Como podemos ver, Si se cumple una condición, ejecutamos una acción, pero si no se ejecuta, entonces crearemos otra condición, dependiente de la anterior. Es decir, que todas las condiciones están en cadena, es por eso que la última parte, no necesitamos especificar ninguna condición más, puesto que al estar todas en cadena, si no cumple ninguna de ellas, la última queda por defecto o por default.

```
Inicio //Comienzo del algoritmo
Entero dia //Declaración de variables
Leer dia //Asignamos los valores ingresados a las variables
    Si(dia == 1) //Si dia es 1
        Imprimir "Es Lunes"
    Sino
    Si(dia == 2) //Si dia es 2
        Imprimir "Es Martes"
    Sino
    Si(dia == 3) //Si dia es 3
        Imprimir "Es Miércoles"
    Sino
    Si(dia == 4) //Si dia es 4
        Imprimir "Es Jueves"
    Sino
    Si(dia == 5) //Si dia es 5
        Imprimir "Es Viernes"
    Sino
    Si(dia == 6) //Si dia es 6
        Imprimir "Es Sábado"
    Sino
    Si(dia == 7) //Si dia es 7
        Imprimir "Es Domingo"
    Sino //Si el dia esta fuera del rango
Imprimir "Error, ingrese un número del 1 al 7"
Fin //Término del programa
```

Ejemplo de un algoritmo de programación que determina si has pasado un examen (calificación de 0 a 20):

En este ejemplo, el usuario ingresará su nota y determinaremos ha pasado o no el curso. Para esto, deberemos preguntar con un Si, si la nota ingresada está entre 0 y 10, lo cual significaría que estaría reprobado. En caso contrario (Sino), preguntaremos si su nota se encuentra entre 10.5 y 20, lo cual significaría que está aprobado. En el caso de que ninguno de estos sea correcto (Sino), deberemos asumir, que no es una nota entre 0 y 20, y por lo tanto, no es válida.

```
Inicio //Comienzo del algoritmo
Entero nota //Declaración de variables
Leer nota //Asignamos los valores ingresados a las variables
    Si(nota >= 0 && nota <= 10) //Si la nota es 0 <= nota >= 10
        Imprimir "Estás desaprobado"
    Sino
        Si(nota >= 10.5 && nota <= 20) -->Si la nota es 10.5 <= nota >= 20
            Imprimir "Estás aprobado"
        Sino
            Imprimir "Error, ingrese una nota válida"
Fin //Término del programa
```

4.- Secuencia de Control Avanzada

Esta secuencia de control avanzada, se denomina "Switch".

- Selección "Switch": permite que el programa escoja entre un rango de valores y ejecute la acción que está asociada con el valor que ha elegido. En "Switch", es muy importante la sentencia break, porque le dice al programa que solo escoja uno de los casos y obvie los demás, si no se escribe el break luego de cada sentencia o grupo de sentencias, el programa las ejecutará todas.

```
Switch(numero)
{
    caso 0:
        accion
        break;
    case 1:
        accion
        break;
    ... default:
        accion
        break;
}
```

Esta sentencia es muy útil, cuando tenemos varios valores y debemos preguntar por todos ellos. En vez de utilizar infinitos Si - Sino, utilizamos Switch.

Ejemplo de un algoritmo de programación que pregunta si es fin de semana:

En este ejemplo, queremos saber si es Domingo, porque nos gustan los Domingos -;)

En este caso, en vez de utilizar Si - Sino, vamos a utilizar Switch, que nos permite, preguntar por todos los posibles valores de una variable. Si el caso es "Domingo", está bien, porque nos gustan los Domingos. Pero, si es cualquier otro día, esperaremos a que sea Domingo, para eso utilizamos Default, es decir, todos los demás valores, que no sean Domingo. Es importante notar que el Break, nos va a servir para que cuando se cumple una condición, las demás no se sigan ejecutando, es decir, si el usuario ingresa "Domingo", le diremos que nos gustan los Domingos, y gracias al Break, ahí termina el programa, pero si decidimos quitar ese Break, entonces, además de decirle que nos gusta el Domingo, le vamos a decir también que nos avise cuando sea Domingo....¿Un poco ilógico no? Por eso, siempre debemos utilizar Break, luego de cada Caso.

```

Inicio //Comienzo del algoritmo
Cadena dia //Declaración de variables
Leer dia //Asignamos los valores ingresados a las variables
Switch(dia) //Toma el valor de día
{
Caso "Domingo": //Si es domingo
Imprimir "Me gusta el domingo" //Imprimimos en pantalla el resultado
Break;
Default: //Si dia no es Domingo
Imprimir "Avisame cuando sea domingo"
Imprimir "Me gustan mas los domingos"
Break;
}
Fin //Término del algoritmo

```

Ejemplo de un algoritmo de programación que pide un número mayor que 0 y menor que 10:

En este ejemplo, el usuario debe ingresar un número que esté entre el 0 y el 10, en caso contrario (Default), recibirá un mensaje de error. Como estamos utilizando el Switch, entonces, podemos validar múltiples valores de la variable "numero", por eso que decimos: Caso1:Caso2:Caso3: Es decir, si el usuario ingresa los números 1, 2 ó 3. Los mismo haremos para los número 4, 5 y 6 y para los números 7, 8 y 9.

```
Inicio //Comienzo del algoritmo
Entero numero //Declaración de variables
Leer dia //Asignamos los valores ingresados a las variables
Switch(numero) //Toma el valor de numero
{
Caso 1: Caso 2: Caso 3:
    Imprimir "Muy chico"
Caso 4: Caso 5: Caso 6:
    Imprimir "Más o menos"
Caso 7: Caso 8: Caso 9:
    Imprimir "Así está bien"
Default:
    Imprimir "Te dije mayor a 0 y menor a 10"
Fin //Término del algoritmo
```

Ejemplo de un algoritmo de programación que determina el día de la semana:

En este ejemplo, asumimos que cada día de la semana es un número, por ejemplo Lunes → 1, Martes → 2, etc. Por lo tanto, cuando el usuario ingrese un número, por ejemplo 4, el programa deberá decirle que día de la semana es. Sería Jueves. Para eso, utilizamos un Switch, probando con todos los valores posibles, es decir, Casos del 1 al 7. Y un Default que indique el error, cuando el número ingresado no esté entre 1 y 7.

```
Inicio //Comienzo del algoritmo
Entero dia //Declaración de variables
Leer dia //Asignamos los valores ingresados a las variables
Switch(dia) //Toma el valor de día
{
Caso 1:
    Imprimir "Es Lunes"
    Break;
Caso 2:
    Imprimir "Es Martes"
    Break;
Caso 3:
    Imprimir "Es Miércoles"
    Break;
Caso 4:
    Imprimir "Es Jueves"
    Break;
Caso 5:
    Imprimir "Es Viernes"
    Break;
Caso 6:
    Imprimir "Es Sábado"
    Break;
Caso 7:
    Imprimir "Es Domingo"
```

```
        Break;  
Default:  
        Imprimir "Error, ingresa un número del 1 al 7"  
        Break;  
}  
Fin //Término del programa
```

5.- Métodos

Los métodos, son pequeños fragmentos de código que realizan una acción determinada y que pueden ser llamados cuantas veces sea necesario en el programa o el algoritmo de programación. Estos métodos pueden recibir o no parámetros, que son variables con las que puede interactuar y retornar o no un valor.

- **Métodos que no retornan un valor:** Este tipo de métodos, puede tener o no parámetros, y su principal función es la imprimir o asignar valores.

```
LeerDatos()
{
    Entero edad
    Cadena nombre
    leer edad,nombre
}
```

```
sumar(Entero n1,Entero n2)
{
    Entero suma
    suma = n1 + n2
    Imprimir suma
}
```

- **Métodos que retornan un valor:** Este tipo de métodos, puede tener o no parámetros, y deben devolver un valor que se asignará a una variable.

```
Entero leerDatos()
{
    Entero edad
    Leer edad
    Retornar edad
}
```

```

Real Area(Entero altura,Entero base)
{
    Real area
    area = altura * base
    Retornar base
}

```

Ejemplo de un algoritmo de programación que pregunta si es fin de semana:

En este ejemplo, vamos a utilizar un método que nos diga que número ha sido ingresado por el usuario. Como se podrán dar cuenta, lo declaramos al final del programa. Cadena Leer_Dia(). Esto significa, que es una función que nos va a devolver una valor de tipo Cadena, es decir, un día de la semana. Una vez que tenemos el día ingresado por el usuario, utilizamos nuestro Switch, para ver que mensaje mostramos.

```

Inicio //Comienzo del algoritmo
Cadena dia = leer_dia() //Declaración de variables y asignación
Switch(dia) //Toma el valor de dia
{
    caso "Domingo": //Si es domingo
    imprimir "Me gusta el domingo" //Imprimimos en pantalla el resultado
    break;
    default: //Si dia no es Domingo
    imprimir "Avisame cuando sea domingo"
    imprimir "Me gustan mas los domingos"
    break;
}
Fin //Término del algoritmo

```

```

Cadena leer_dia()
{
    Cadena dia
    Leer dia
    retornar dia //Devuelve el valor de la variable día
}

```

Ejemplo de un algoritmo de programación que suma dos número enteros:

En este ejemplo, vamos a sumar dos números enteros, y para eso utilizamos una función o método Sumar. Este método, va a recibir 2 parámetros, N1 y N2, es decir, los números ingresados por el usuario. Además, va a devolver el valor de la variable suma, que va a contener, el valor de la suma de los dos números ingresados.

```

Inicio //Comienzo del algoritmo
Entero n1,n2,suma //Declaración de variables
Leer n1,n2 //Asignamos los valores ingresados a las variables
suma = sumar(n1,n2) //Llamamos al método y le pasamos como parámetros
                    //los número ingresados
Imprimir suma //Mostramos el resultado
Fin //Término del programa

Entero Sumar(Entero n1,Entero n2)
{
    Entero suma
    suma = n1 + n2
    Retornar suma //Devuelve el valor de la variable suma
}

```

Ejemplo de un algoritmo de programación que suma, multiplica y resta dos números:

En este ejemplo, deberemos utilizar 3 métodos, uno para cada operación, Suma, Multiplicación y División. Como podemos ver, los 3 son bastante similares, puesto que reciben dos parámetros, N1 y N2, que son los valores ingresados por el usuario, y los 3 devuelven una variable de tipo Entero con el resultado de la operación realizada.

```

Inicio //Comienzo del algoritmo
Entero n1,n2,suma,multi,resta //Declaración de variables
Leer n1,n2 //Asignamos los valores ingresados a las variables
suma = Sumar(n1,n2) //Llamamos al método y le pasamos como parámetros
                    //los número ingresados
multi = Multip(n1,n2)
resta = Restar(n1,n2)
Imprimir suma,multi,resta //Mostramos el resultado
Fin //Término del programa

Entero Sumar(Entero n1,Entero n2)
{
Entero suma
suma = n1 + n2
Retornar suma //Devuelve el valor de la variable suma
}

Entero Multip(Entero n1,Entero n2)
{
Entero multi
multi = n1 * n2
Retornar multi //Devuelve el valor de la variable multi
}

```

```
Entero Resta(Entero n1,Entero n2)
{
Entero resta
resta = n1 - n2
Retornar resta //Devuelve el valor de la variable resta
}
```

Ejemplo de un algoritmo de programación que imprime el nombre del usuario:

En este ejemplo, simplemente leemos el nombre del usuario, y llamamos al método `Escribir_Nombre` para que lo imprima en pantalla.

```
Inicio //Comienzo del algoritmo
Cadena Nombre
Leer Nombre
Escribir_Nombre(Nombre)
Fin //Termino del programa

Escribir_Nombre(Cadena Nombre)
{
Imprimir "Hola " + Nombre //Imprime el nombre del usuario en pantalla
}
```

6.- Secuencias repetitivas

Las secuencias repetitivas, nos permiten realizar una acción o un grupo de acciones, un número determinado de veces, mientras una condición se cumpla o hasta que la condición se cumpla.

- **Mientras:** Esta secuencia repetitiva, ejecutará la acción o grupo de acción solamente mientras la condición sea verdadera.

```
Mientras(condición)
{
    acción
}
```

Ejemplo de un algoritmo de programación que imprime un nombre 5 veces:

Este ejemplo, imprime mi nombre 5 veces -;) Como? Muy fácil, primero creamos una variable, y le damos el valor de 0. Luego decimos, mientras X (Nuestra variable que vale 0), sea menor que 5, imprime este nombre. Claro, también, además de imprimir mi nombre, debería de hacer que la variable X, aumente en 1, por cada vez que imprime mi nombre, ¿No? Porque sino, nunca terminaría de imprimir. Entonces, como comenzó $X = 0$, cada vez que imprima, va a aumentar en 1. Y cuando $X = 5$, la condición $X < 5$ va a ser falsa y el programa va a terminar.

```
Inicio //Comienzo del algoritmo
Entero x //Declaración de variables
x = 0 //Inicializamos a x con 0
Mientras (x < 5)
{
    Imprimir "Alvaro Tejada Galindo"
    x++ //Lo mismo que x = x + 1
}
Fin //Término del algoritmo
```

- **Hacer...Mientras:** Esta secuencia repetitiva, ejecutará la acción o grupo de acción y luego verificará la condición, es decir, ejecutará por lo menos una vez.

```
Hacer
{
  acción
}Mientras(condición)
```

Ejemplo de un algoritmo de programación que imprime un nombre 5 veces:

Este ejemplo, imprime mi nombre 5 veces -;) Y es muy similar al anterior, con la diferencia de que este utiliza un Hacer.....Mientras, es decir, primero ejecuta y luego pregunta si puede o no ejecutar. Es decir, de todos modos va a imprimir mi nombre por lo menos una vez.

```
Inicio //Comienzo del algoritmo
Entero x //Declaración de variables
x = 0 //Inicializamos a X con 0
Hacer
{
  Imprimir "Alvaro Tejada Galindo"
  x++ //Lo mismo que x = x + 1
}Mientras (x < 5)
Fin //Término del algoritmo
```

- **Para:** Esta secuencia repetitiva, toma una variable, la inicializa, verifica la condición y la aumenta o disminuye.

```
Para(x=0;i<5;i++)
{
  acción
}
```

Ejemplo de un algoritmo de programación que imprime un nombre 5 veces:

En este ejemplo, volvemos otra vez a imprimir mi nombre -:D Aunque ahora, lo hacemos utilizando Para.....este comando lo que hace, para la variable X que vale 0, hasta que sea mayor o igual que 5, aumenten su valor en 1. Y todo el código que este dentro de los { } será ejecutado el número de veces que el Para decida, en este caso, 5 veces.

```
Inicio //Comienzo del algoritmo
Entero x //Declaración de variables
Para (x = 0; x < 5; x++ )
{
  Imprimir "Alvaro Tejada Galindo"
}
Fin //Término del algoritmo
```

7.- Arreglos o Matrices

Los arreglos o matrices, son una colección de elementos del mismo tipo almacenados uno tras del otro en una variable contenedora. Los arreglos o matrices son muy útiles cuando se quiere simular una base de datos sencilla. Los arreglos o matrices, comienzan de 0 hasta n-1.

Arreglo o matriz de 4 elementos:

Índice	Elemento	Contenido
0	edad[0]	20
1	edad[1]	24
2	edad[2]	18
3	edad[3]	15

Ejemplo de un algoritmo de programación que leer valores para una inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

En este ejemplo, debemos llenar una Matriz o Arreglo de 5 elementos. Por eso declaramos nuestra variable array[4], porque tiene las casillas del 0 al 4, entonces, son 5 elementos. Para poder acceder a cada una de estas casillas y guardar un valor en ellas, necesitamos un Para, que vaya desde 0 hasta el límite de la Matriz, es decir $i \leq 4$. Al escribir array[i], significa que por cada valor que tome i dentro del Para, accederemos a esa casilla de la matriz. Por lo tanto, para imprimir los valores, haremos lo mismo, un Para de 0 a ≤ 4 . Y listo.

```

Inicio //Comienzo del algoritmo
Entero array[4],i //Declaración de variables
Para (i = 0 ; i <= 4; i++) //De 0 a 4 = 5 elementos
{
Leer array[i] //Asigna los valores a cada elemento de la matriz
}
  
```

```

Para (i = 0 ; i <= 4; i++)
{
  Imprimir "El arreglo o matriz es: " + array[i]
}
//Imprime los valores de los elementos de la matriz
Fin //Término del programa

```

Ejemplo de un algoritmo de programación que imprime los valores pares de una matriz de 6 elementos:

En este ejemplo, el usuario debe ingresar 6 valores, y nosotros deberemos determinar si son pares o no. En el caso de que sean pares, los imprimiremos. Como antes, utilizamos un Para, para poder llenar la matriz o arreglo, con los datos ingresados por el usuario. Una vez llenados, utilizaremos otro Para, para poder imprimir los valores, pero antes de poder imprimirlos, debemos saber si son o no, números pares, para esto, deberemos dividir cada valor del arreglo o matriz, entre 2 y obtener su residuo (Con %), si el resultado es 0. Significa que es par y podremos imprimirlo. En caso contrario, si es un número impar, no haremos nada.

```

Inicio //Comienzo del algoritmo
Entero array[5],i //Declaración de variables
Para (i = 0; i < 6; i++) //De 0 a 6
{
  Leer array[i] //Asigna los valores a cada elemento de la matriz
}
Para (i = 0; i < 6; i++)
{
  Si (array[i] % 2 == 0)
  Imprimir array[i]
}
Fin //Término del programa

```

8.- Lenguajes de programación

Los lenguajes de programación, son programas que nos permiten desarrollar programas, esto es, herramientas que no permiten automatizar acciones que queremos que la computadora realice. Estos lenguajes de programación han sufrido muchos cambios a través de la historia, en un inicio, estos fueron pensados y realizados para un selecto grupo de científicos, quienes eran los únicos que podían escribir programas, dada la complejidad de los lenguajes existentes, entre ellos, el "Assembler", que es el lenguaje de programación más cercano al lenguaje máquina, que está compuesto por 1's y 0's.

- **Assembler:** Lenguaje de programación, que es el más cercano al lenguaje de máquina y trabaja con instrucciones tanto binarias como hexadecimales. Y como comandos mnemónicos como Add, Mov y Jmp.

Luego de este, que era muy complejo, Jim Backus desarrolló el Fortran (Formula Translator). Que fue el primer lenguaje de programación de alto nivel.

- **Fortran:** Lenguaje de programación, utilizado para realizar cálculos matemáticos, científicos o de ingeniería.

Posteriormente, se desarrolló el Cobol (Common Business Oriented Language). Que se extendió rápidamente, aunque aún seguía siendo bastante complicado.

- **Cobol:** Lenguaje de programación, que se utiliza para aplicaciones de negocios y es aún usado en grandes corporaciones como bancos y empresas.

Otro de los lenguajes más extendidos por su facilidad de aprendizaje, fué el Basic (Beginner's All Purpose Symbolic Instruction Code), desarrollado por John Kemeny y Thomas Kurtz y funcionó por primera vez en 1964.

- **Basic:** Lenguaje de programación, de sintaxis sencilla y de mucha potencia. De este, surgió la variante más conocida y extendida QBasic.

- **QBasic:** Lenguaje de programación, desarrollado por Microsoft, y que reconoce estructuras de control, permite omitir las líneas en los programas y es más veloz que los lenguajes interpretados.

Los lenguajes de Alto Nivel arriba mencionados (Fortran, Cobol, Basic y QBasic), eran lenguajes que seguían la metodología de la programación "Estructurada", lo que quiere decir, que los programas se escribían en bloques.

Para mejorar esto, en 1979, Bjarne Stroustrup comenzó a desarrollar el lenguaje C++, basándose en el popular lenguaje C. Pero añadiéndole mejoras, tales como la filosofía de programación "Orientada a Objetos", en la cual, cada elemento es un objeto que tiene características, propiedades y métodos. C++ vio la luz, oficialmente en 1983.

- **C++:** Lenguaje de programación, "Orientado a Objetos" y que es uno de los más utilizados, tanto para desarrollar aplicaciones como juegos. A pesar de que es una versión mejorada de C, es aún bastante difícil de aprender, porque incluye muchas funciones que lo hacen ser uno de los lenguajes más robustos. De este, se crearon las variaciones, TurboC, Visual C++ y C#.

Como era de suponerse, con todos estos lenguajes madres, por así decirlo, dieron lugar a nuevos lenguajes, más sencillos y poderosos, tales como Visual Basic, Java y Euphoria.

- **Visual Basic:** Lenguaje de programación desarrollado por Microsoft, y "Orientado a Eventos", en el cual, todos los elementos responden a eventos producidos por el usuario, como, presionar una tecla o hacer click en un botón, actualmente, es el lenguaje de programación más difundido y utilizado.
- **Java:** Lenguaje de programación, desarrollado por James Gosling de Sun Microsystems, en base a un lenguaje de programación para electrodomésticos, en el cual estaba trabajando llamado "Oak" y que se basaba en C++, aunque mejorando algunos errores que el encontraba. Luego, Bill Joy cofundador de Sun y uno de los principales desarrolladores del Unix de Berkeley, se dio cuenta de que Oak, podía ser utilizado en Internet y así hacer frente a Microsoft. En 1995, Java fue introducido al mercado.

- **Euphoria:** Lenguaje de programación, desarrollado por Rapid Deployment Software. Este nuevo lenguaje, permite desarrollar aplicaciones tanto en entorno Dos como Windows y posee una versión para Linux, de fácil entendimiento y gratuito hasta cierto punto, es uno de los lenguajes de programación por los cuales apuesto y estoy seguro de que con un poco de apoyo, puede llegar a convertirse en uno de los lenguajes más utilizados, ya sea por su potencia, su tamaño reducido o el precio de su licencia que no sube de 30 dólares.

9.- Ejemplos Algorítmicos

En este capítulo, vamos a presentar una serie de ejemplos algorítmicos que van a seguir de guía para desarrollar los programas, en variados lenguajes de programación (Qbasic, C++, Pascal, Euphoria, Java, JavaScript, Delphi, PHP), así podrán darse cuenta de como un algoritmo bien diseñado, puede ser transcrito en cualquier lenguaje de programación, para esto, todos los ejemplos van a ser iguales, aunque escritos en cada lenguaje de programación distinto, para ayudar a conocer la lógica. Con lo cual, ustedes podrán decidir con que lenguaje se identifican más.

1) Programa que escribe en la pantalla "Hola Mundo" (Primer ejemplo de programación, creado por Brian Kernighan y Dennis Ritchie en 1976).

```
Inicio //Comienzo del algoritmo
Imprimir "Hola Mundo" //Imprimimos en pantalla el mensaje
Fin //Término del algoritmo
```

2) Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

```
Inicio //Comienzo del algoritmo
Real totalpagar,precio //Declaración de variables
Entero cantidad //Declaración de variables
Leer cantidad,precio //Asignamos los valores ingresados a las variables
totalpagar = cantidad * precio //Guardamos la cantidad
//por el precio en la variable totalpagar
Imprimir "Tienes que pagar " + totalpagar
//Imprimimos en pantalla el resultado
Fin //Término del algoritmo
```

3) Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```
Inicio //Comienzo del algoritmo
Entero numero //Declaración de variables
Leer numero //Asignamos los valores ingresados a las variables
Si (numero % 2 == 0) //Verifica si la condición se cumple
Imprimir "Es par" //Imprimimos en pantalla el resultado
Sino //Si el residuo de numero % 2 no es 0
Imprimir "Es impar"
Fin //Término del algoritmo
```

4) Ejemplo de un algoritmo de programación que lee valores para inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

```
Inicio //Comienzo del algoritmo
Entero array[4],i //Declaración de variables
Para (i = 0;i < 5;i++) //De 0 a 4
Leer array[i] //Asigna los valores a cada elemento de la matriz
Para (i = 0;i < 5;i++)
Imprimir "El arreglo o matriz es: " + array[i]
//Imprime los valores de los elementos de la matriz
Fin //Término del programa
```

5) Ejemplo de un algoritmo de programación que determine las cifras de las centenas, decenas y unidades de un número de 3 cifras:

```

Inicio //Comienzo del algoritmo
Entero numero,cien,decena,unidad,aux //Declaración de variables
Leer numero //Asignamos los valores ingresados a las variables
Si (numero > 999 || numero < 100) //Si es de cuatro o dos cifras
Imprimir "Error. Ingrese un número de 3 cifras"
Sino
{
cien = numero / 100 //La parte entera
aux = numero % 100 //Guardamos el residuo en un auxiliar
decena = aux / 10 //Obtenemos la parte entera del auxiliar
unidad = numero % 10 //Obtenemos el residuo
}
Imprimir cien,decena,unidad
Fin //Término del algoritmo

```

6) Ejemplo de un algoritmo de programación que determina cuantas cifras tiene un número:

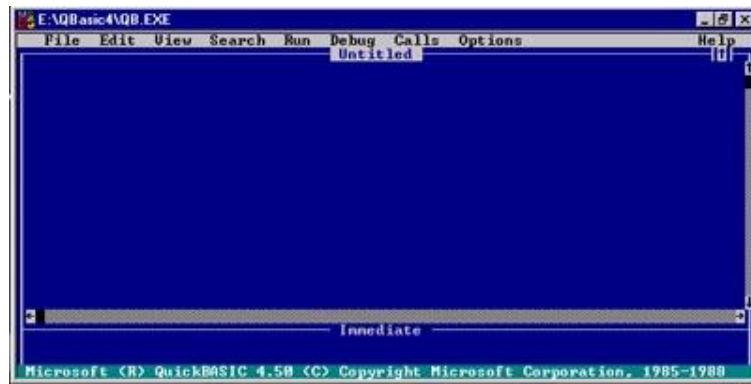
```

Inicio //Comienzo del algoritmo
Entero numero,cant = 0,aux //Declaración de variables
Leer numero //Asignamos los valores ingresados a las variables
Hacer //Ejecutar este bloque de código
{
cant++ //Lo mismo que cant = cant + 1
aux = numero / 10 //Obtenemos la parte entera de la división
numero = numero / 10 //Asignamos el valor de la división a la variable
}Mientras (aux > 0) //La condición se cumple mientras que el valor de aux se
//mayor que cero

Imprimir "Tiene " + cant + " digitos" //Imprimimos el resultado
Fin //Término del algoritmo

```

9.1.- QBasic



1) Programa que escribe en la pantalla "Hola Mundo" (Primer ejemplo de programación, creado por Brian Kernighan y Dennis Ritchie en 1976).

```
PRINT "HOLA MUNDO"
```

2) Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

```
INPUT "Ingrese la cantidad de productos ", cantidad#  
INPUT "Ingrese el precio por unidad ", precio#  
totalpagar# = cantidad# * precio#  
PRINT "Tienes que pagar ", totalpagar#
```

3) Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```
INPUT "Ingresa un número", numero  
IF (numero MOD 2 = 0) THEN  
PRINT "Es par"  
ELSE  
PRINT "Es impar"  
END IF
```

4) Ejemplo de un algoritmo de programación que leer valores para una inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

```
DIM array(4)
FOR i = 0 to 4
INPUT "Ingrese valores ",array(i)
NEXT i
FOR i = 0 to 4
PRINT "El arreglo o matriz es: ", array[i]
NEXT i
```

5) Ejemplo de un algoritmo de programación que determine las cifras de las centenas, decenas y unidades de un número de 3 cifras:

```
INPUT "Ingresa un número ", numero
IF (numero > 999 OR numero < 0) THEN
PRINT "Error. Ingrese un número de 3 cifras"
ELSE
cien = FIX (numero / 100)
aux = numero MOD 100
decena = FIX (aux / 10)
unidad = numero MOD 10
PRINT "Las centenas son ",cien
PRINT "Las decenas son ",decena
PRINT "Las unidades son ",unidad
END IF
```

6) Ejemplo de un algoritmo de programación que determina cuantas cifras tiene un número:

```
cant = 0
INPUT "Ingresa número ",numero
aux = 1
DO WHILE aux > 0
cant = cant + 1
aux = FIX (numero / 10)
numero = FIX (numero / 10)
LOOP
PRINT "El número tiene ",cant," dígitos"
```

9.2.- Pascal



1) Programa que escribe en la pantalla "Hola Mundo" (Primer ejemplo de programación, creado por Brian Kernighan y Dennis Ritchie en 1976).

```
PROGRAM Hola;
USES crt;
BEGIN
  BEGIN
    WRITELN ('Hola Mundo');
  END;
  REPEAT UNTIL KEYPRESSED;
END.
```

2) Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

```
PROGRAM PrePro;
USES crt;
VAR iCantidad:INTEGER;
VAR rTotalPagar,rPrecio:REAL;
BEGIN
  BEGIN
    WRITE ('Ingrese Cantidad: ');
    READLN(iCantidad);
```

```
WRITE ('Ingrese Precio: ');  
READLN(rPrecio);  
rTotalPagar := iCantidad * rPrecio;  
WRITELN('Tienes que pagar ',rTotalPagar:1:2);  
END;  
REPEAT UNTIL KEYPRESSED;  
END.
```

3) Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```
PROGRAM ParImpar;  
USES crt;  
VAR iNumero:INTEGER;  
BEGIN  
  BEGIN  
    WRITE ('Ingresa un número: ');  
    READLN(iNumero);  
    IF (iNumero MOD 2 = 0) THEN  
      BEGIN  
        WRITELN('Es par');  
      END  
    ELSE  
      WRITELN('Es impar');  
    END;  
    REPEAT UNTIL KEYPRESSED;  
  END.
```

4) Ejemplo de un algoritmo de programación que leer valores para una inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

```
PROGRAM Array;
USES crt;
VAR iMatriz:ARRAY[1..5] OF INTEGER;
VAR i:INTEGER;
BEGIN
  BEGIN
    FOR i := 1 TO 5 DO
      BEGIN
        WRITE('Ingresa valores: ');
        READ(iMatriz[i]);
        END;
      FOR i := 1 TO 5 DO
        WRITELN('El arreglo o matriz es: ',iMatriz[i]);
        END;
      REPEAT UNTIL KEYPRESSED;
    END.
```

5) Ejemplo de un algoritmo de programación que determine las cifras de las centenas, decenas y unidades de un número de 3 cifras:

```
PROGRAM CienDec;
USES crt;
VAR iNumero,iCien,iDecena,iUnidad,iAux:INTEGER;
BEGIN
  WRITE('Ingresa un número: ');
  READ(iNumero);
  IF (iNumero > 999) OR (iNumero < 100) THEN
    WRITELN('Error. Ingrese un número de 3 cifras')
  ELSE
    BEGIN
```

```
iCien := iNumero DIV 100;  
iAux := iNumero MOD 100;  
iDecena := iAux DIV 10;  
iUnidad := iNumero MOD 10;  
END;  
WRITELN('Centenas ',iCien,' Decenas ',iDecena,' Unidades ',iUnidad);  
BEGIN  
END;  
REPEAT UNTIL KEYPRESSED;  
END.
```

6) Ejemplo de un algoritmo de programación que determina cuantas cifras tiene un número:

```
PROGRAM CienDec;  
USES crt;  
VAR iNumero,iCant,iAux:INTEGER;  
BEGIN  
  BEGIN  
    iCant := 0;  
    WRITE('Ingresa un número: ');  
    READ(iNumero);  
    REPEAT  
      iCant := iCant + 1;  
      iAux := iNumero DIV 10;  
      iNumero := iNumero DIV 10;  
    UNTIL iAux = 0;  
    WRITELN('Tiene ',iCant,' dígitos');  
  END;  
  REPEAT UNTIL KEYPRESSED;  
END.
```

9.3.- C++



1) Programa que escribe en la pantalla "Hola Mundo" (Primer ejemplo de programación, creado por Brian Kernighan y Dennis Ritchie en 1976).

```
#include <conio.h>
#include <iostream.h>
int main()
{
    cout<<"Hola Mundo";
    getch();
}
```

2) Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

```
#include <conio.h>
#include <iostream.h>
int main()
{
    int iCantidad;
    float fPrecio,fTotalpagar;
    cout<<"Ingrese cantidad a comprar: ";
    cin>>iCantidad;
    cout<<"Ingrese precio: ";
```

```

cin>>rPrecio;
fTotalpagar = iCantidad * rPrecio;
cout<<"Tienes que pagar: "<<totalpagar;
getch();
}

```

3) Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```

#include <conio.h>
#include <iostream.h>
int main()
{
int iNumero;
cout<<"Ingresa un número: ";
cin>>iNumero;
if(iNumero % 2 == 0)
    cout<<"Es par";
else
    cout<<"No es par";
getch();
}

```

4) Ejemplo de un algoritmo de programación que leer valores para una inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

```

#include <conio.h>
#include <iostream.h>
int main()
{
int iArray[4],i;

```

```

for(i = 0; i < 5; i++)
{
    cout<<"Ingresa valores: ";
    cin>>iArray[i];
}
for(i = 0; i < 5; i++)
    cout<<"\nEl arreglo o matriz: "<<iArray[i];
    getch();
}

```

5) Ejemplo de un algoritmo de programación que determine las cifras de las centenas, decenas y unidades de un número de 3 cifras:

```

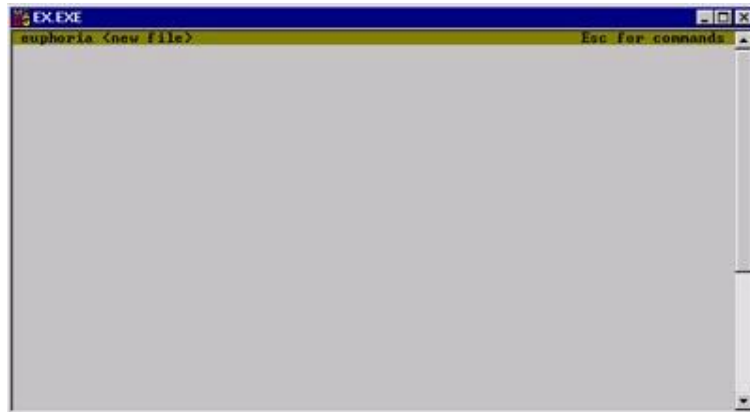
#include <conio.h>
#include <iostream.h>
int main()
{
    int iNumero,iCien,iDecena,iUnidad,iAux;
    cout<<"Ingresa un numero de 3 cifras: ";
    cin>>iNumero;
    if(iNumero > 999 || iNumero < 100)
        cout<<"Error. Ingresa un numero de 3 cifras";
    else
    {
        iCien = iNumero / 100;
        iAux = iNumero % 100;
        iDecena = iAux / 10;
        iUnidad = iNumero % 10;
    }
    cout<<"Centenas: "<<iCien<<"  Decenas:  "<<iDecena<<"  Unidades:
    "<<iUnidad;
    getch();
}

```

6) Ejemplo de un algoritmo de programación que determina cuantas cifras tiene un número:

```
#include <conio.h>
#include <iostream.h>
int main()
{
    int iNumero,iCant = 0,iAux;
    cout<<"Ingresa un número: ";
    cin>>iNumero;
    do
    {
        iCant++;
        iAux = iNumero / 10;
        iNumero = iNumero / 10;
    }while(iAux > 0);
    cout<<"Tiene "<<iCant<<" dígitos";
    getch();
}
```

9.4. - Euphoria



1) Programa que escribe en la pantalla "Hola Mundo" (Primer ejemplo de programación, creado por Brian Kernighan y Dennis Ritchie en 1976).

```
puts(1,"Hola Mundo")
```

2) Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

```
include get.e

atom aCantidad,aPrecio,aTotalpagar
aCantidad = prompt_number("Ingrese cantidad a comprar: ",{})
aPrecio = prompt_number("Ingrese precio a pagar: ",{})
pTotalpagar = aPrecio * aCantidad
puts(1,"Tienes que pagar: ")
print(1,pTotalpagar)
```

3) Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```
include get.e

atom aNumero,aAux
aNumero = prompt_number("Ingresa un número,{})
aAux = remainder(aNumero,2)
if(aAux = 0) then
puts(1,"Es par")
else
puts(1,"No es par")
end if
```

4) Ejemplo de un algoritmo de programación que leer valores para una inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

```
include get.e

sequence sNumero
numero = {{},{},{},{},{}}
for i = 1 to 5 do
    sNumero[i] = prompt_number("Ingresa valores: ",{})
end for
for i = 1 to 5 do
    puts(1,"\nLa matriz o arreglo es: ")
    print(1,sNumero[i])
end for
```

5) Ejemplo de un algoritmo de programación que determine las cifras de las centenas, decenas y unidades de un número de 3 cifras:

```
include get.e

atom aNumero,aCien,aDecena,aUnidad,aAux
aNumero = prompt_number("Ingresa un número de 3 cifras")
if(aNumero > 999 or aNumero < 100) then
    puts(1,"Error. Ingresa un número de 3 cifras")
else
    aCien = floor(aNumero / 100)
    aAux = remainder(aNumero,100)
    aDecena = floor(aAux / 10)
    aUnidad = remainder(aNumero,10)
    puts(1,"Centenas: ")
    print(1,aCien)
    puts(1," Decenas: ")
    print(1,aDecena)
    puts(1," Unidades: ")
    print(1,aUnidad)
end if
```

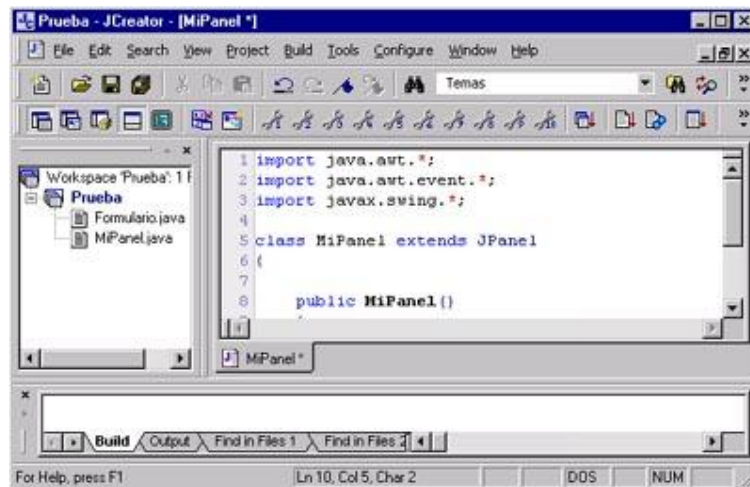
6) Ejemplo de un algoritmo de programación que determina cuantas cifras tiene un número:

```
include get.e

atom aNumero,aCant,aAux
aCant = 0
aAux = 1
aNumero = prompt_number("Ingresa un número: ",{})
while aAux > 0 do
    aCant = aCant + 1
    aAux = floor(aNumero / 10)
```

```
        aNumero = floor(aNumero / 10)
    end while
    puts(1,"Tiene ")
    print(1,ACant)
    puts(1," aCifras")
```

9.5.- Java



1) Programa que escribe en la pantalla "Hola Mundo" (Primer ejemplo de programación, creado por Brian Kernighan y Dennis Ritchie en 1976).

```
import java.awt.*;
import java.applet.*;

public class Prueba extends Applet
{
    public void paint( Graphics g )
    {
        g.drawString("Hola Mundo",20,20);
    }
}
```

2) Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

```
import java.awt.event.*;
import java.awt.*;
import java.applet.*;
```

```
public class Prueba extends Applet implements ActionListener
{
    int iCantidad;
    double dTotalPagar,dPrecio;
    TextField txtCant,txtPre;
    TextArea txtSalida;
    Button btnProcesar;
    Label lblMensaje,lblOtroMsg;

    public void init()
    {
        lblMensaje = new Label("Ingresa la cantidad a comprar");
        add(lblMensaje);
        txtCant = new TextField("");
        add(txtCant);
        lblOtroMsg = new Label("Ingresa el precio");
        add(lblOtroMsg);
        txtPre = new TextField("");
        add(txtPre);
        btnProcesar = new Button("Procesar");
        btnProcesar.addActionListener(this);
        add(btnProcesar);
        txtSalida = new TextArea("");
        add(txtSalida);
    }

    public void actionPerformed( ActionEvent e )
    {
        if(e.getSource().equals(btnProcesar))
        {
            iCantidad = Integer.parseInt(txtCant.getText());
            dPrecio = Double.parseDouble(txtPre.getText());
            dTotalPagar = iCantidad * dPrecio;
```

```

        txtSalida.setText("Tienes que pagar " + dTotalPagar);
    }
}
}

```

3) Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```

import java.awt.event.*;
import java.awt.*;
import java.applet.*;

public class Prueba extends Applet implements ActionListener
{
    int iNumero;
    TextField txtNum;
    TextArea txtSalida;
    Button btnProcesar;
    Label lblMensaje;

    public void init()
    {
        lblMensaje = new Label("Ingresa un número");
        add(lblMensaje);
        txtNum = new TextField("");
        add(txtNum);
        btnProcesar = new Button("Procesar");
        btnProcesar.addActionListener(this);
        add(btnProcesar);
        txtSalida = new TextArea("");
        add(txtSalida);
    }
}

```

```

        public void actionPerformed((ActionEvent e)
        {
            if(e.getSource().equals(btnProcesar))
            {
                iNumero = Integer.parseInt(txtNum.getText());
                if(iNumero % 2 == 0)
                    txtSalida.setText("Es Par");
                else
                    txtSalida.setText("Es Impar");
            }
        }
    }
}

```

4) Ejemplo de un algoritmo de programación que leer valores para una inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

```

import java.awt.event.*;
import java.awt.*;
import java.applet.*;

public class Prueba extends Applet implements ActionListener
{
    int iArray[] = new int[5], counter = 0, iNumero;
    TextField txtNum;
    TextArea txtSalida;
    Button btnProcesar;
    Label lblMensaje;

    public void init()
    {
        lblMensaje = new Label("Ingresa valores");
        add(lblMensaje);
    }
}

```

```
txtNum = new TextField("");
add(txtNum);
btnProcesar = new Button("Procesar");
btnProcesar.addActionListener(this);
add(btnProcesar);
txtSalida = new TextArea("");
add(txtSalida);
}

public void actionPerformed( ActionEvent e )
{
    if(e.getSource().equals(btnProcesar))
    {
        if(counter < 5)
        {
            iNumero = Integer.parseInt(txtNum.getText());
            iArray[counter] = iNumero;
            counter++;
            txtNum.setText("");
            txtNum.requestFocus();
        }
        else
        {
            txtNum.setText("");
            for(int i = 0; i < counter; i++)
            {
                txtSalida.append("\nLa matriz o arreglo es: " + iArray[i]);
            }
        }
    }
}
```

5) Ejemplo de un algoritmo de programación que determine las cifras de las centenas, decenas y unidades de un número de 3 cifras:

```
import java.awt.event.*;
import java.awt.*;
import java.applet.*;

public class Prueba extends Applet implements ActionListener
{
    int iNumero,iCien,iDecena,iUnidad,iAux;
    TextField txtNum;
    TextArea txtSalida;
    Button btnProcesar;
    Label lblMensaje;

    public void init()
    {
        lblMensaje = new Label("Ingresa un número");
        add(lblMensaje);
        txtNum = new TextField("");
        add(txtNum);
        btnProcesar = new Button("Procesar");
        btnProcesar.addActionListener(this);
        add(btnProcesar);
        txtSalida = new TextArea("");
        add(txtSalida);
    }

    public void actionPerformed( ActionEvent e )
    {
        if(e.getSource().equals(btnProcesar))
        {
            iNumero = Integer.parseInt(txtNum.getText());
```

```

        if(iNumero > 999 || iNumero < 100)
            txtSalida.setText("Error. Ingresas un número de 3 cifras");
        else
        {
            iCien = iNumero / 100;
            iAux = iNumero % 100;
            iDecena = iAux / 10;
            iUnidad = iNumero % 10;
            txtSalida.setText("Centenas: " + iCien +
                " Decenas: " + iDecena + " Unidades: " + iUnidad);
        }
    }
}

```

6) Ejemplo de un algoritmo de programación que determina cuantas cifras tiene un número:

```

import java.awt.event.*;
import java.awt.*;
import java.applet.*;

public class Prueba extends Applet implements ActionListener
{
    int iNumero,iCant = 0,iAux;
    TextField txtNum;
    TextArea txtSalida;
    Button btnProcesar;
    Label lblMensaje;

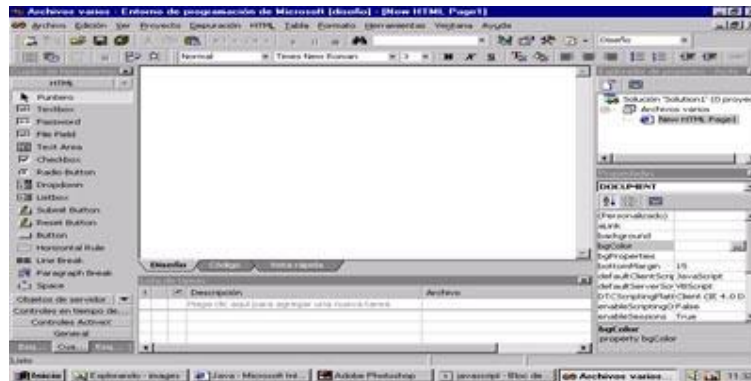
    public void init()
    {
        lblMensaje = new Label("Ingresa un número");
    }
}

```

```
add(lblMensaje);
txtNum = new TextField("");
add(txtNum);
btnProcesar = new Button("Procesar");
btnProcesar.addActionListener(this);
add(btnProcesar);
txtSalida = new TextArea("");
add(txtSalida);
}

public void actionPerformed( ActionEvent e )
{
    if(e.getSource().equals(btnProcesar))
    {
        iNumero = Integer.parseInt(txtNum.getText());
        do
        {
            iCant++;
            iAux = iNumero / 10;
            iNumero = iNumero / 10;
        }while(iAux > 0);
        txtSalida.setText("Tiene " + iCant + " dígitos");
    }
}
}
```

9.6. - JavaScript



1) Programa que escribe en la pantalla "Hola Mundo" (Primer ejemplo de programación, creado por Brian Kernighan y Dennis Ritchie en 1976).

```
<html>
<head>
<script language="JavaScript">
    document.write("Hola Mundo");
</script>
</head>
</html>
```

2) Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

```
<html>
<head>
<script language="JavaScript">
function Precios()
{
var cant = Compras.cant.value;
var pre = Compras.pre.value;
var cantpre = (cant * pre);
```

```

alert("El precio es: " + cantpre);
}
</script>
</head>
<body>
<form name="Compras">
Ingresa la cantidad a comprar <input type="text" name="cant">
Ingresa el precio <input type="text" name="pre">
<input type="submit" value="Procesar" onClick="Precios()">
</form>
</body>
</html>

```

3) Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```

<html>
<head>
<script language="JavaScript">
function ParImpar()
{
var num = Numeros.num.value;
if(num % 2 == 0)
    alert(num + " Es par");
else
    alert(num + " No es par");
}
</script>
</head>
<body>
<form name="Numeros">
Ingresa un número <input type="text" name="num">
<input type="submit" value="Procesar" onClick="ParImpar()">

```

```

</form>
</body>
</html>

```

4) Ejemplo de un algoritmo de programación que leer valores para una inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

```

<html>
<head>
<script language="JavaScript">
var iArray = new Array(),counter = 0,iNumero;

function Llenar_Array()
{
    iArray[0] = prompt('Ingresa el primer número');
    iArray[1] = prompt('Ingresa el segundo número');
    iArray[2] = prompt('Ingresa el tercero número');
    iArray[3] = prompt('Ingresa el cuarto número');
    iArray[4] = prompt('Ingresa el quinto número');
    Llenar_Caja();
}

function Llenar_Caja()
{
    for (i = 0; i < 5; i++)
        Arreglo.Box.value += iArray[i] + "\n" ;
}

</script>
</head>
<body onLoad="Llenar_Array()">
<form name="Arreglo">

```

```
<textarea name="Box" rows="10" cols="10">
</textarea>
</form>
</body>
</html>
```

5) Ejemplo de un algoritmo de programación que determine las cifras de las centenas, decenas y unidades de un número de 3 cifras:

```
<html>
<head>
<script language="JavaScript">
var iNumero,iCien,iDecena,iUnidad,iAux;
function Convertir()
{
iNumero = prompt('Ingresa un número');
if(iNumero > 999 || iNumero < 100)
document.write("Error. Ingresa un número de 3 cifras");
else
{
iCien = Math.floor(iNumero / 100);
iAux = iNumero % 100;
iDecena = Math.floor(iAux / 10);
iUnidad = iNumero % 10;
document.write("Centenas: " + iCien + " Decenas: " + iDecena + " Unidades: " +
iUnidad);
}
}
</script>
```

```

</head>
<body>
<input type="Button" value="Valores" onClick="Convertir()">
</body>
</html>

```

6) Ejemplo de un algoritmo de programación que determina cuantas cifras tiene un número:

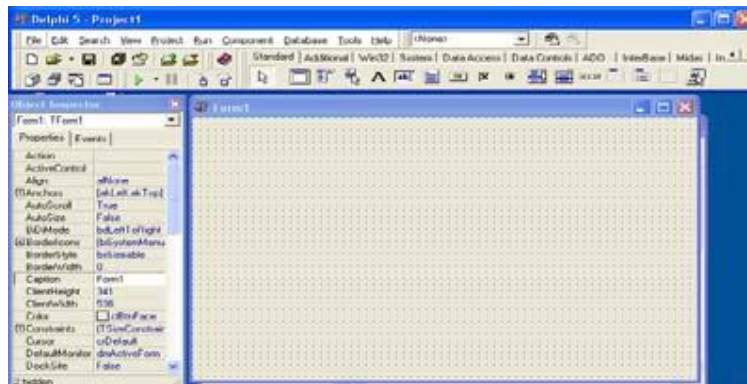
```

<html>
<head>
<script language="JavaScript">
var iNumero,iCant = 0,iAux;
function Cantidad()
{
iNumero = prompt('Ingresa un número');
do
{
iCant++;
iAux = Math.floor(iNumero / 10);
iNumero = Math.floor(iNumero / 10);
}while(iAux > 0);
document.write("Tiene " + iCant + " dígitos");
}
</script>

</head>
<body>
<input type="Button" value="Cantidad" onClick="Cantidad()">
</body>
</html>

```

9.7.- Delphi



- 1) Programa que escribe en la pantalla "Hola Mundo" (Primer ejemplo de programación, creado por Brian Kernighan y Dennis Ritchie en 1976).

```
program HolaMundo;
{$APPTYPE CONSOLE}

uses
  SysUtils;

begin
  writeln ('Hola Mundo');
  readln;
end.
```

- 2) Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

```
program PrePro;
{$APPTYPE CONSOLE}

uses
  SysUtils;
```

```

var
iCantidad:integer;
rTotalPagar,rPrecio:real;

begin
write ('Ingrese Cantidad: ');
readln(iCantidad);
write ('Ingrese Precio: ');
readln(rPrecio);
rTotalPagar := iCantidad * rPrecio;
writeln('Tienes que pagar ',rTotalPagar:1:2);
readln;
end.

```

3) Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```

program ParImpar;
{$APPTYPE CONSOLE}
uses
  SysUtils;

var
iNumero:integer;

begin
write('Ingresa un número: ');
readln(iNumero);
if (iNumero mod 2 = 0) then
begin
writeln('Es par');
end
else

```

```
writeln('Es impar');  
readln;  
end.
```

4) Ejemplo de un algoritmo de programación que leer valores para una inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

```
program Arreglos;  
{$APPTYPE CONSOLE}  
uses  
    SysUtils;  
  
var  
    iMatriz:array[1..5] of integer;  
    i:integer;  
  
begin  
    for i := 1 to 5 do  
        begin  
            write('Ingresa valores: ');  
            read(iMatriz[i]);  
            end;  
        for i := 1 to 5 do  
            begin  
                writeln('El arreglo o matriz es: ',iMatriz[i]);  
                readln;  
            end;  
        end.
```

5) Ejemplo de un algoritmo de programación que determine las cifras de las centenas, decenas y unidades de un número de 3 cifras:

```
program CienDec;
{$APPTYPE CONSOLE}
uses
  SysUtils;

var
  iNumero,iCien,iDecena,iUnidad,iAux:integer;

begin
  write('Ingresa un número: ');
  read(iNumero);
  if (iNumero > 999) or (iNumero < 100) then
    begin
      writeln('Error. Ingrese un número de 3 cifras');
      iCien := 0;
      readln;
    end
  else
    begin
      iCien := iNumero div 100;
      iAux := iNumero mod 100;
      iDecena := iAux div 10;
      iUnidad := iNumero mod 10;
    end;
    writeln('Centenas ',iCien,' Decenas ',iDecena,' Unidades ',iUnidad);
    readln;
    readln;
  end.
```

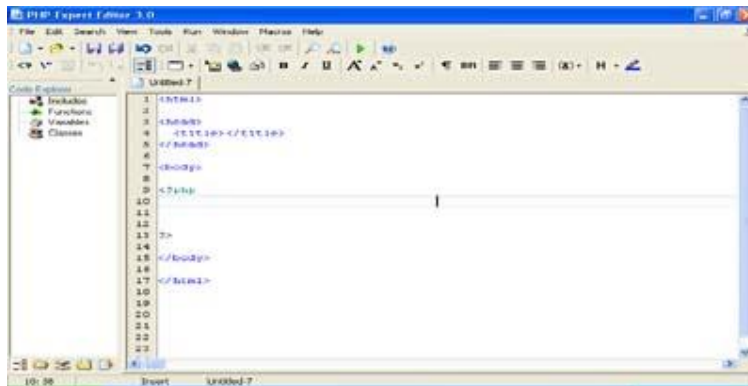
6) Ejemplo de un algoritmo de programación que determina cuantas cifras tiene un número:

```
program CienDec;
{$APPTYPE CONSOLE}
uses
  SysUtils;

var iNumero,iCant,iAux:integer;

begin
  iCant := 0;
  write('Ingresa un número: ');
  read(iNumero);
  repeat
    iCant := iCant + 1;
    iAux := iNumero div 10;
    iNumero := iNumero div 10;
  until iAux = 0;
  writeln('Tiene ',iCant,' dígitos');
  readln;
  readln;
end.
```

9.8. - PHP



- 1) Programa que escribe en la pantalla "Hola Mundo" (Primer ejemplo de programación, creado por Brian Kernighan y Dennis Ritchie en 1976).

```
<html>
<head><title></title></head>
<body>
<?php echo "Hola Mundo"; ?>
</body>
</html>
```

2) Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

```
<html>
<head><title>Prueba</title></head>
<body>
<form action="prueba.php" method="post">
  Cantidad: <input type="text" name="Cantidad"><br>
  Precio:  <input type="text" name="Precio"><br>
  <input type="submit" value="Calcular">
</form>
</body>
</html>

*****

<html>
<head><title>Resultado</title></head>
<body>
<?php
$TotalPagar = $Cantidad * $Precio;
echo "Tienes que pagar $TotalPagar";
?>
</body>
</html>
```

3) Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```
<?
if (isset($Numero))
{
if ($Numero % 2 == 0)
echo "Es Par.";
else
echo "Es Impar.";
}
?>

<html>
<head><title>Prueba</title></head>
<body>
<form action="<?php print $PHP_SELF?>" method="post">
    Número: <input type="text" name="Numero"><br>
    <input type="submit" value="Evaluar">
</form>
</form>
</body>
</html>
```

- 4) Ejemplo de un algoritmo de programación que leer valores para una inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

```

<?
if (isset($Num1) and isset($Num2) and
    isset($Num3) and isset($Num4) and
    isset($Num5))
{
$Matriz[] = $Num1;
$Matriz[] = $Num2;
$Matriz[] = $Num3;
$Matriz[] = $Num4;
$Matriz[] = $Num5;

$Counter = count($Matriz);
for($i = 0; $i < $Counter; $i++)
    echo "El arreglo o matriz es: $Matriz[$i]<br>";
}
?>

<html>
<head><title>Prueba</title></head>
<body>
<form action="<?php print $PHP_SELF?>" method="post">
    Número 1: <input type="text" name="Num1"><br>
    Número 2: <input type="text" name="Num2"><br>
    Número 3: <input type="text" name="Num3"><br>
    Número 4: <input type="text" name="Num4"><br>
    Número 5: <input type="text" name="Num5"><br>
    <input type="submit" value="Evaluar">
</form>
</form>
</body>
</html>

```

5) Ejemplo de un algoritmo de programación que determine las cifras de las centenas, decenas y unidades de un número de 3 cifras:

```
<?
$cien = 0;$decena = 0;$unidad = 0;$aux = 0;
if(isset($Numero))
{
if($Numero > 999 || $Numero < 100)
echo "Error. Ingrese un número de 3 cifras<br>";
else
{
$cien = Floor($Numero / 100);
$aux = $Numero % 100;
$decena = Floor($aux / 10);
$unidad = $Numero % 10;
}
echo "Las centenas son $cien<br>";
echo "Las decenas son $decena<br>";
echo "Las unidades son $unidad";
}
?>

<html>
<head><title>Prueba</title></head>
<body>
<form action="<?php print $PHP_SELF?>" method="post">
    Número: <input type="text" name="Numero"><br>
    <input type="submit" value="Evaluar">
</form>
</form>
</body>
</html>
```

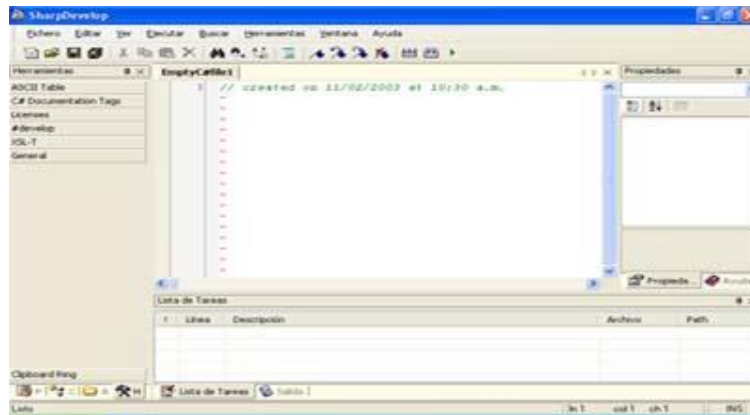
6) Ejemplo de un algoritmo de programación que determina cuantas cifras tiene un número:

```
<?
$scant = 0;
if(isset($Numero))
{
do
{
$scant++;
$aux = Floor($Numero / 10);
$Numero = Floor($Numero / 10);
}while($aux > 0);

echo "Tiene $scant dígitos";
}
?>

<html>
<head><title>Prueba</title></head>
<body>
<form action="<?php print $PHP_SELF?>" method="post">
    Número: <input type="text" name="Numero"><br>
    <input type="submit" value="Evaluar">
</form>
</form>
</body>
</html>
```

9.9.- C#



- 1) Programa que escribe en la pantalla "Hola Mundo" (Primer ejemplo de programación, creado por Brian Kernighan y Dennis Ritchie en 1976).

```
using System;

class MainClass
{
    public static void Main(string[] args)
    {
        Console.Write("Hola Mundo");
        Console.WriteLine();
    }
}
```

2) Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

```
using System;

class MainClass
{
    public static void Main(string[] args)
    {
        int iCantidad;
        float rTotalPagar,rPrecio;

        Console.Write("Ingrese Cantidad: ");
        iCantidad = Int32.Parse(Console.ReadLine());
        Console.Write("Ingrese Precio: ");
        rPrecio = float.Parse(Console.ReadLine());
        rTotalPagar = iCantidad * rPrecio;
        Console.WriteLine("Tienes que pagar: " + rTotalPagar);
    }
}
```

3) Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```
using System;

class MainClass
{
    public static void Main(string[] args)
    {
        int iNumero;

        Console.Write("Ingresa un número: ");
        iNumero = Int32.Parse(Console.ReadLine());
        if(iNumero % 2 == 0)
            Console.WriteLine("Es Par.");
        else
            Console.WriteLine("Es Impar.");
    }
}
```

4) Ejemplo de un algoritmo de programación que leer valores para una inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

```
using System;

class MainClass
{
    public static void Main(string[] args)
    {
        int[ ] iArray = {0,0,0,0,0};
        int i;
```

```

for(i = 0; i < 5; i++)
{
    Console.Write("Ingresa Valores: ");
    iArray[i] = Int32.Parse(Console.ReadLine());
}
for(i = 0; i < 5; i++)
{
    Console.Write("El Arreglo o Matriz es: ");
    Console.Write(iArray[i]);
    Console.WriteLine();
}
}
}

```

5) Ejemplo de un algoritmo de programación que determine las cifras de las centenas, decenas y unidades de un número de 3 cifras:

```

using System;

class MainClass
{
    public static void Main(string[] args)
    {
        int iNumero,iCien,iDecena,iUnidad,iAux;

        Console.Write("Ingresa un número: ");
        iNumero = Int32.Parse(Console.ReadLine());
        if (iNumero > 999 || iNumero < 100)
        {
            Console.WriteLine("Error. Ingrese un número de 3 cifras");
            iCien = 0;
            iDecena = 0;
            iUnidad = 0;

```

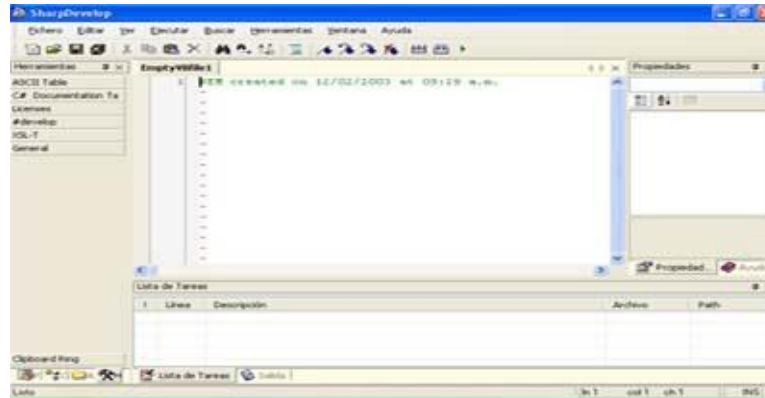
```
}  
else  
{  
    iCien = iNumero / 100;  
    iAux = iNumero % 100;  
    iDecena = iAux / 10;  
    iUnidad = iNumero % 10;  
}  
Console.WriteLine("Centenas " + iCien + " Decenas " + iDecena +  
    " Unidades " + iUnidad);  
}  
}
```

6) Ejemplo de un algoritmo de programación que determina cuantas cifras tiene un número:

```
using System;  
  
class MainClass  
{  
    public static void Main(string[] args)  
    {  
        int iNumero,iCant,iAux;  
  
        iCant = 0;  
        Console.Write("Ingresa un número: ");  
        iNumero = Int32.Parse(Console.ReadLine());
```

```
do
{
    iCant++;
    iAux = iNumero / 10;
    iNumero = iNumero / 10;
}while(iAux > 0);
Console.WriteLine("Tiene " + iCant + " dígitos");
}
}
```

9.9.- VB.NET



- 1) Programa que escribe en la pantalla "Hola Mundo" (Primer ejemplo de programación, creado por Brian Kernighan y Dennis Ritchie en 1976).

```
Imports System
Module Main
    Sub Main()
        Console.WriteLine("Hola Mundo")
    End Sub
End Module
```

- 2) Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

```
Imports System
Module Main
    Sub Main()
        dim iCantidad as integer
        dim rTotalPagar,rPrecio as double

        Console.Write("Ingrese Cantidad: ")
        iCantidad = Console.ReadLine()
        Console.Write("Ingrese Precio: ")
```

```

rPrecio = Console.ReadLine()
rTotalPagar = iCantidad * rPrecio
Console.Write("Tienes que pagar: ")
Console.WriteLine(rTotalPagar)
End Sub
End Module

```

3) Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```

Imports System
Module Main
    Sub Main()
        dim iNumero as integer
        Console.Write("Ingresa un número: ")
        iNumero = Console.ReadLine()
        if(iNumero MOD 2 = 0) then
            Console.WriteLine("Es Par.")
        else
            Console.WriteLine("Es Impar.")
        end if
    End Sub
End Module

```

4) Ejemplo de un algoritmo de programación que leer valores para una inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

```

Imports System
Module Main
    Sub Main()
        dim iArray(5) as integer
        dim i as integer

```

```

for i = 1 to 5
    Console.Write("Ingresa Valores: ")
    iArray(i) = Console.ReadLine()
next i
for i = 1 to 5
    Console.Write("El Arreglo o Matriz es: ")
    Console.WriteLine(iArray(i))
next i
End Sub
End Module

```

5) Ejemplo de un algoritmo de programación que determine las cifras de las centenas, decenas y unidades de un número de 3 cifras:

```

Imports System
Module Main
Sub Main()
    dim iNumero,iCien,iDecena,iUnidad,iAux as integer

    Console.Write("Ingresa un número: ")
    iNumero = Console.ReadLine()
    if (iNumero > 999 or iNumero < 100) then
        Console.WriteLine("Error. Ingresa un número de 3 cifras")
        iCien = 0
        iDecena = 0
        iUnidad = 0
    else
        iCien = Math.Floor(iNumero / 100)
        iAux = iNumero MOD 100
        iDecena = Math.Floor(iAux / 10)
        iUnidad = iNumero MOD 10
        Console.Write("Centenas: ")
        Console.Write(iCien)
    end if
End Sub
End Module

```

```

        Console.Write(" Decenas: ")
        Console.Write(iDecena)
        Console.Write(" Unidades: ")
        Console.WriteLine(iUnidad)
    end if
End Sub
End Module

```

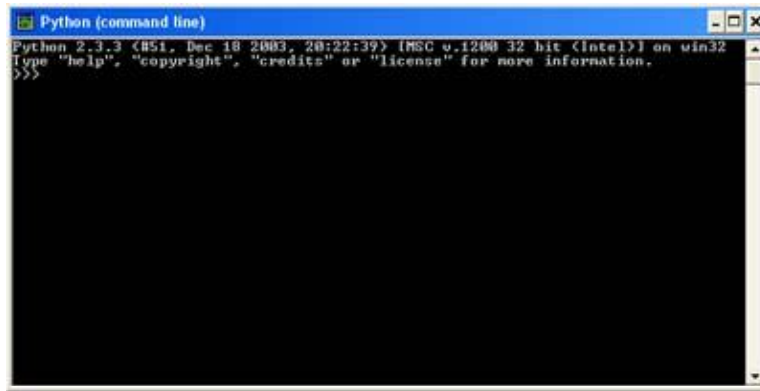
6) Ejemplo de un algoritmo de programación que determina cuantas cifras tiene un número:

```

Imports System
Module Main
Sub Main()
    dim iNumero,iCant,iAux as integer

    iCant = 0
    iAux = 1
    Console.Write("Ingresa un número: ")
    iNumero = Console.ReadLine()
    do while iAux > 0
        iCant = iCant + 1
        iAux = Math.Floor(iNumero / 10)
        iNumero = Math.Floor(iNumero / 10)
    loop
    Console.Write("Tiene ")
    Console.Write(iCant)
    Console.WriteLine(" dígitos")
End Sub
End Module

```



- 1) Programa que escribe en la pantalla "Hola Mundo" (Primer ejemplo de programación, creado por Brian Kernighan y Dennis Ritchie en 1976).

```
print "Hola Mundo"
```

- 2) Ejemplo de un algoritmo de programación que calcula el precio a pagar por cierta cantidad de productos:

```
iCantidad = int(raw_input("Ingresa cantidad: "))
rPrecio = float(raw_input("Ingresa precio: "))

rTotalPagar = iCantidad * rPrecio

print "Tienes que pagar: "
print rTotalPagar
```

- 3) Ejemplo de un algoritmo de programación que determina si un número es par o impar:

```
iNumero = int(raw_input("Ingresa un numero: "))
if iNumero % 2 == 0:
    print "Es Par"
else:
    print "Es Impar"
```

- 4) Ejemplo de un algoritmo de programación que leer valores para una inicializar una matriz o arreglo de 5 elementos y luego imprime los valores ingresados:

```
iCounter = 0
iArray = [ ]

while iCounter < 5:
    numero = int(raw_input("Ingrese valores: "))
    iArray.append(numero)
    iCounter = iCounter + 1

print iArray
```

- 5) Ejemplo de un algoritmo de programación que determine las cifras de las centenas, decenas y unidades de un número de 3 cifras:

```
iNumero = int(raw_input("Ingresa un numero: "))
if iNumero > 999 or iNumero < 100:
    print "Error. Ingrese un numero de 3 cifras"
else:
    iCien = iNumero / 100
    iAux = iNumero % 100
    iDecena = iAux / 10
    iUnidad = iNumero % 10

    print "Centenas: ", iCien
    print "Decenas: ", iDecena
    print "Unidades: ", iUnidad
```

6) Ejemplo de un algoritmo de programación que determina cuantas cifras tiene un número:

```
iCant = 0
iAux = 1

iNumero = int(raw_input("Ingresa un numero: "))
while iAux > 0:
    iCant = iCant + 1
    iAux = iNumero / 10
    iNumero = iNumero / 10

print "Tiene ", iCant, " digitos"
```

10.- ¿Cómo ser un buen programador?

No creo que esté escrito, ni que hayan reglas formales para saber quien es y quien no es un buen programador. Lo que si hay, es mi experiencia personal como programador autodidacta, después de más de cinco años de estudiar programación, he podido llegar a pensar unas cuantas cosas que pueden ayudar a los principiantes a convertirse en buenos programadores, y también "expertos" (Leer mi artículo "Que significa ser "Programador"). Para mi, un buen programador es aquel que disfruta programando, que es parte de su vida, algo que le nace y lo hace espontáneamente. Una vez, me preguntaron si programar era difícil, pues no, no es difícil, lo único que se necesita es tener lógica, imaginación y muchos deseos de aprender. Y claro, nunca sobran libros, tutoriales, artículos o códigos de los cuales aprender. Yo en lo personal le debo mucho a Internet y a los libros. Aunque como dije, no se pueden enumerar ni enmarcar los requerimientos para ser un buen programador, podemos mencionar aquí unos cuantos.

- Mantén un estándar de programación. Por ejemplo, usa un tipo de notación como la Camel Notation "iCantidad", donde i equivale a Integer,int,etc. O alguna otra con la que te sientas cómodo, pero tienes que mantenerla en todos tus programas.
- Comenta tus códigos para que te sea más fácil entenderlos, pero no comentes lo que hace sino como lo hace. Eso puede serte muy útil si revisas tus códigos cada mes o cada año.
- No copies códigos sin permiso, para usarlos en tus programas, y si lo haces da el crédito correspondiente al autor. Un buen programador debe ser sincero con su trabajo.
- Ayuda a los programador principiantes cuando estos te pidan ayuda. (Ya parece algo religioso esto, ¿No?).
- Trata de crear nuevas cosas y si te basas en cosas que ya existen, trata de mejorarlas o hacerles algo que las diferencien de las demás y se note tu esfuerzo.

Supongo que con estas pequeñas pautas, podrás darte cuenta de como tienes que ser o lo que tienes que hacer para ser un buen programador. Y algo más que es muy importante, tal vez lo más importante que he aprendido en estos 5 años y contando.

- No te preocupes sino puedes hacer ese súper programa que tanto querías, si ya pasaron 3 días y nada, una semana y no te sale, un mes y no entiendes nada, tranquilo, respira y tomate un descanso, espera unos días o una semanita y luego regresa a tu programa, vas a ver como las cosas van a comenzar a salirte.

Eso lo digo por experiencia propia, a veces el descanso es lo mejor para que la mente pueda descubrir el problema que nos mantiene estancados y cuando volvemos al código, nos damos cuenta de lo sencillo que había sido, y no nos habíamos dado cuenta porque estábamos demasiado molestos, porque creíamos que era imposible de hacer, y ya estábamos a punto de borrar el programa. -:)

La verdad es que el mundo de la programación es inmenso, existen una cantidad impresionante de lenguajes de programación, algunos fáciles, otros casi imposibles, otro que ya no existen, y otros que se están desarrollando, está en nosotros elegir el que más nos guste o convenga y aprenderlo a fondo.

11.- Comentarios Finales

Bueno, hasta aquí llega el libro. Espero que les haya gustado y les sirva para aprender aunque sea un poco, lo que es la programación y lo que es ser un programador. Confió, en que en un futuro, pueda sacar una versión mejorada de este libro, con más ejemplos, tips, etc. Por supuesto, su apoyo es vital en esto, por lo que les pido que me escriban si es que tienen algún comentario, queja, felicitación, o si hay algo que les gustaría ver en una futura publicación de este libro. Como el mundo de la programación es tan complejo, y en constante cambio, yo también me estaré actualizando constantemente, para así, poder ofrecer un mejor servicio. Porque yo como algunos de ustedes, también fui principiante alguna vez, y muchos programadores, me ayudaron indirectamente, esta es pues, una forma de agradecerles y contribuir con la creciente ola de programadores que recién se inician. Les deseo mucha suerte a todos, sigan programando y como alguna vez dije:

*No existen problemas difíciles...
Lo que faltan son soluciones sencillas.*

Adiós... -:)

*Alvaro Tejada
SinglePath games design*

<http://www.iespana.es/singlepath>

12.- Enlaces

[SinglePath games design \(http://www.iespana.es/singlepath\)](http://www.iespana.es/singlepath) Mi página web, en donde encontrarán, juegos y programas en versión Freeware (VB, C++, QBasic, Euphoria, Java), además de tutoriales, artículos, etc. (En español e inglés).

[Planet Source Code \(http://www.planet-source-code.com\)](http://www.planet-source-code.com) La mejor página de códigos del mundo (VB, C++, Pearl, ColdFusion, Java, Asp, Delphi, etc) (En inglés).

[La web del Programador \(http://www.lawebdelprogramador.com\)](http://www.lawebdelprogramador.com) La mejor página en español con todo tipo de recursos para programadores. (En español).

[Xpertia \(http://www.xpertia.com\)](http://www.xpertia.com) La mejor página de expertos, todas las categorías que puedas imaginar, tu preguntas, ellos responden. Búsquenme en la categoría de Programación, usuario "Blag". (En español).

[Euphoria HomePage \(http://www.rapideuphoria.com\)](http://www.rapideuphoria.com) Página oficial del lenguaje de programación "Euphoria", encontrarás el instalador, códigos, tutoriales, etc. (En inglés).

[Java Boutique \(http://www.javaboutique.com\)](http://www.javaboutique.com) La mejor página de Java, con todo lo que necesitas para aprender este lenguaje de internet. (En inglés).

[TelePort Media \(http://www.teleportmedia.com/foro/index\)](http://www.teleportmedia.com/foro/index) El mejor foro en español, para aprender a programar, crear juegos, diseñar en 3D, etc. (En español).

13.- Agradecimientos

- A mi novia Milly, porque siempre me apoya, y siempre es una inspiración para mí.
- A mi Pentium IV de 1.8 Ghz, con 256 de RAM y Windows XP, por servirme de tanto.
- A los grupos PUNX (Misfits, Nofx, Kortatu, Rancid, Sex Pistols) que escuche mientras escribía este libro.
- A todos los programadores de Planet Source Code.
- A "The Croft Times" por dejarme utilizar las imágenes de Lara Croft en mis programas.
- A los lenguajes de programación, por existir.
- A mi gato Cheko, por acompañarme.
- A todos los que tenían que estar aquí y no están.
- A mi amigo Tommy, por ayudarme a aprender más programación. -:D
- A la comunidad del Flat Assembler.
- A mis amigos del foro de TelePort Media (Cronodragón, Geo, JuanK, George, SkullFire, Rilart, FAQ3D).

14.- El Fin

