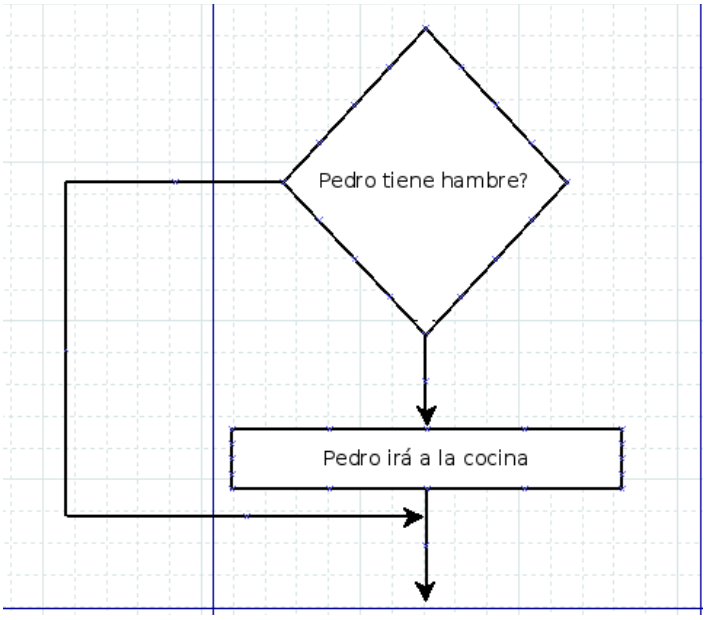
HDC

Bienvenidos nuevamente a una nueva clase de HDC. La semana anterior pudimos disfrutar de la conferencia Ekoparty, donde se dieron charlas muy interesantes (aunque debo admitir que estuvo un poco floja si miramos los años anteriores). Los que tengan la oportunidad de ir a alguna conferencia que esté cerca de su ciudad, les recomiendo que vayan:).

En la clase 36, habíamos visto comandos de control referenciados al pseudocódigo. Hoy lo pasaremos a código en C:

Uno de los comandos de control que vimos, fue el de "**if**". Este "**if**" se refiere al "**si" incondicional**. **Ejemplo**: "Si Pedro tiene hambre, irá a la cocina.". Digamos que se hará cierta acción **si es que cierta condición se cumple**.



En C, la **sintaxis** sería de esta manera:

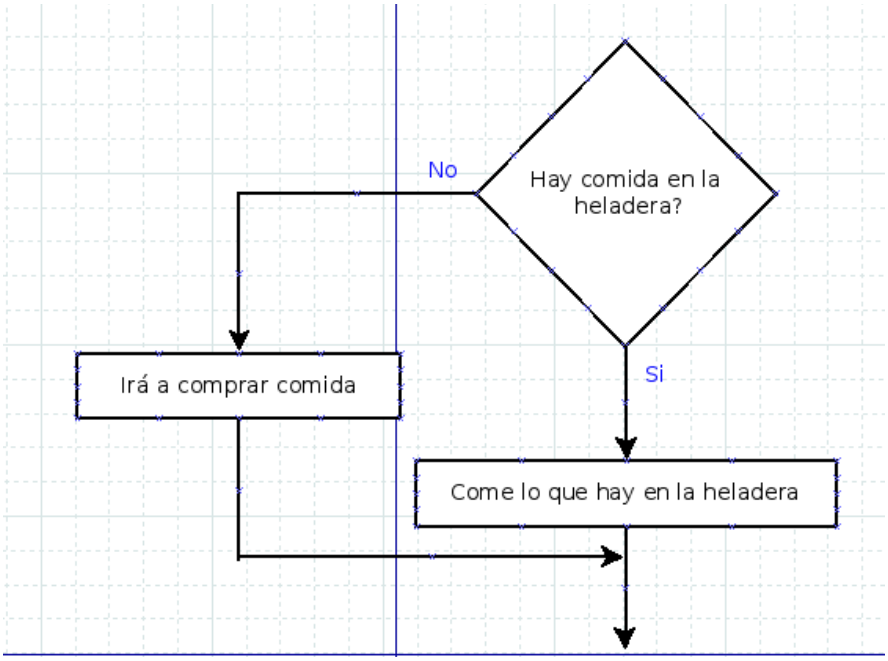
```
if (condicion)
{
        codigo
}
```

Y el **código** en C:

```c
#include <stdio.h>

int main ()
{
      //creo variable entera
      int variable;

      //Le asigno el valor 0
      variable = 0;

      //si la variable es 0 entra al if
      if (variable == 0)
      {
                  printf("La variable es igual a 0");
      }

      //si la variable es distinta de 0 entra al if
      if(variable != 0)
      {
                 printf("La variable es distinta de 0");
      }
}
```

*Para la comparación, **"=="** significa "**es igual**" y "**!=**" significa "**es distinto**"*

Si la condición no se cumple, el programa simplemente ignorará el código interno. Pero si queremos que se ejecute algo si es que la condición **no se cumple**, podemos usar "**else**". **Ejemplo**: "Si hay comida en la heladera, se alimenta; si no, va a comprar comida."

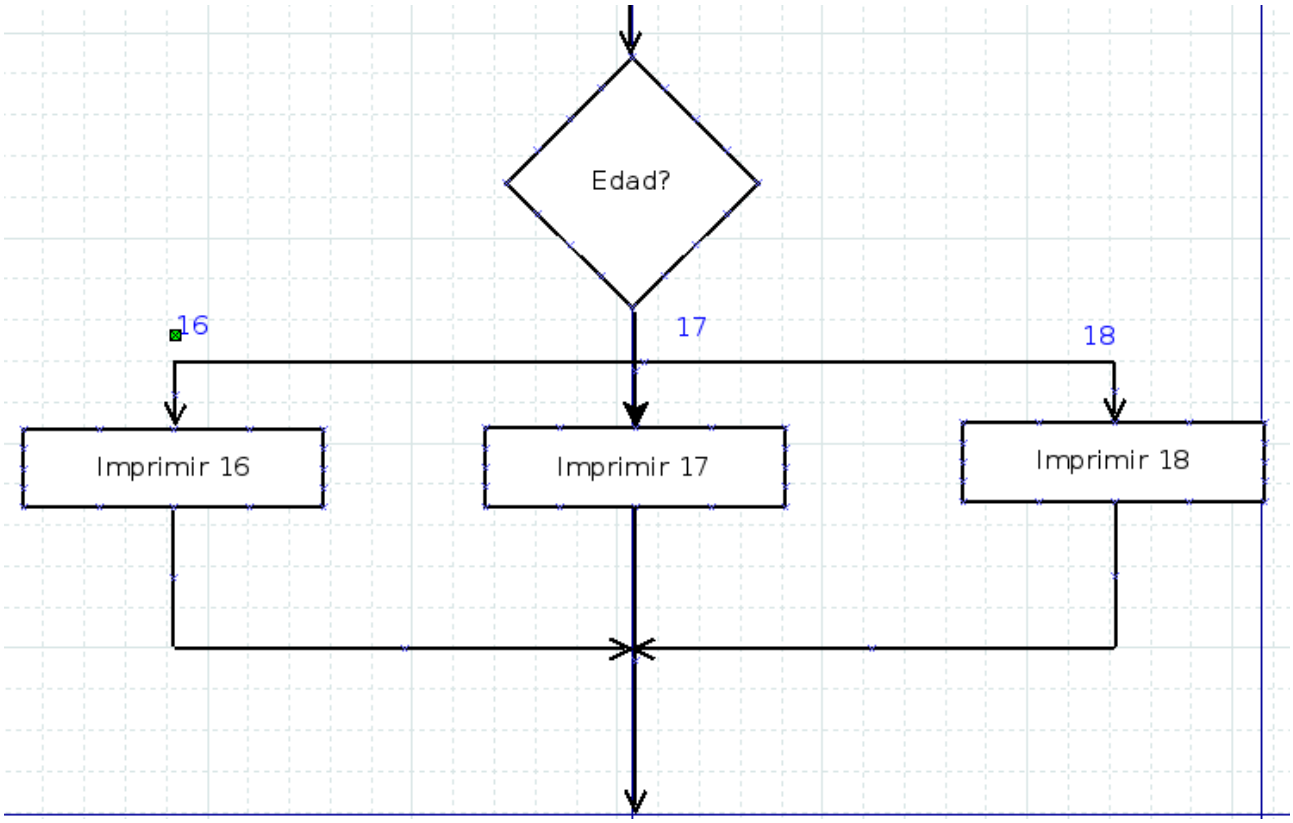


Sintaxis:

```
if(condicion)
{
        codigo
}

else
{
        codigo
}
```

```c
#include <stdio.h>

int main ()
{
	//creo variable entera
	int variable;

	//Le asigno el valor 0
	variable = 0;

	//si la variable es 0 entra al if
	if (variable == 0)
	{
			printf("La variable es igual a 0");
	}

	//si la variable no es igual a 0 entra al else
	else
	{
			printf("La variable es distinta de 0");
	}
}
```

A veces tenemos una variable que puede tener **muchos estados**, y según el valor que tome es la decisión que haga el programa. Para ello podemos usar un "**switch**".
Ejemplo: "Imprimir en pantalla el año de nacimiento según la edad de adolescentes entre los 16 y los 18 años."



Sintaxis:

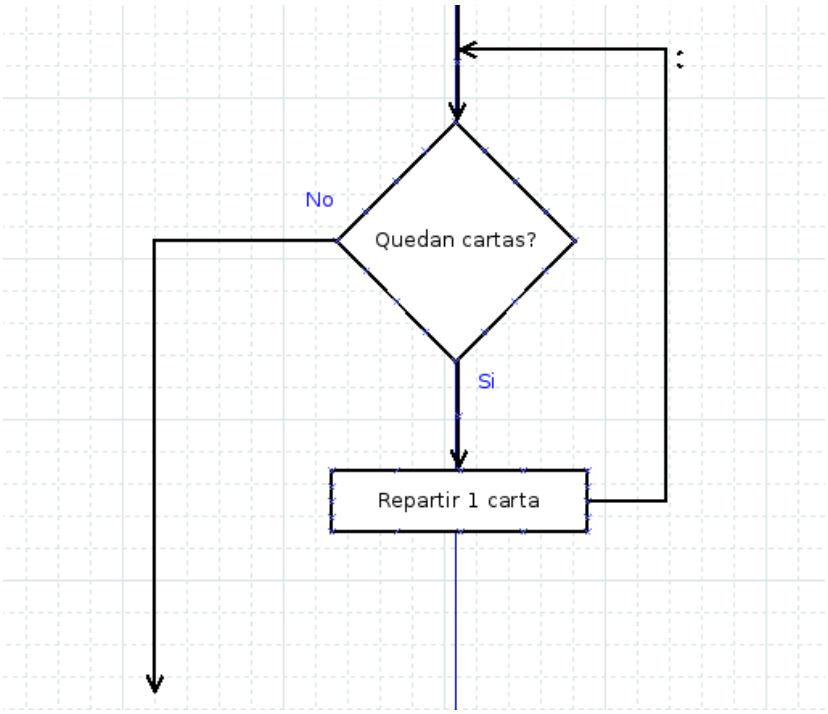
```
switch(variable)
{
        case condicion:
                codigo
                break;
        case condicion2:
                codigo
                break;
        default:
                codigo
                break;
}
```

Código en C:

```c
int main ()
{
     //creo variable entera
     int variable;

     //Le asigno el valor 0
     variable = 0;

     switch(variable)
     {
             case 0:
                  printf("La variable es igual a 0");
                  break;
             case 1:
                  printf("La variable es igual a 1");
                  break;
             default:
                  printf("La variable tiene un valor incorrecto");
                  break;
     }
}
```

El comando “**case**” viene antes de la condición (que debe ser un valor absoluto, como vimos en la última imagen) y luego, para **cerrar el switch**, lo hacemos mediante el comando “**break**” luego de cada case. También puede agregarse el comando “**default**” que dirá qué pasa en caso de que la variable no cumpla con ninguna de las condiciones de los case. Aunque aquí lo usamos de esta manera, es bueno usarlo como estado de control de errores.

Otra de las cosas que podemos crear son **bucles**. Es decir, código que se ejecuta una y otra vez cumpliéndose cierta condición. Una de nuestras herramientas será el "**while**", correspondiente a "**mientras** se cumpla esta condición, se ejecuta este código.". **Ejemplo**: "Repartir cartas de una baraja, hasta que no queden más cartas".



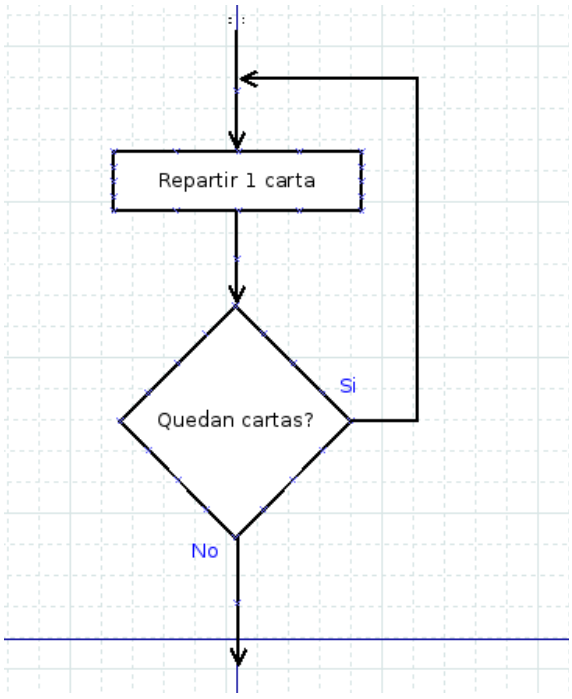
Sintaxis:

```
while (condicion)
{
        codigo
}
```

En el **código** de C:

```
int main()
{
    //creo variables
    int variable;
    int contador;

    //asigno valores de inicializacion
    variable = 0;
    contador = 0;

    while(variable == 0)
    {
                    //en cada bucle, contador aumenta en 1
                    contador = contador + 1;

                    //cuando el contador llegue a 5, entra al if
                    if (contador == 5)
                    {
                                //Valor para poder salir del while
                                variable = 1;
                    }
    }
}
```

En caso de que la condición no se cumpla la primera vez, nunca ejecuta el código interno. En caso de que queramos hacerlo, existe otra manera que se denomina "**do while**". La única diferencia con respecto al anterior, es que **primero se ejecuta el código interno y luego se fija si la condición se cumple**.



Sintaxis:

```
do
{
        codigo
}while(condicion);
```

El **código** en C:

```c
int main()
{
    //creo variables
    int variable;
    int contador;

    //asigno valores de inicializacion
    variable = 0;
    contador = 0;

    do
    {
                //en cada bucle, contador aumenta en 1
                contador = contador + 1;

                //cuando el contador llegue a 5, entra al if
                if (contador == 5)
                {
                            //Valor para poder salir del while
                            variable = 1;
                }
    }while(variable == 0);
}
```

Para terminar los comandos de control veremos el “**for**”. Este se usa cuando queremos hacer un bucle una **cantidad determinada de veces** (y ya sabemos cuantas). La **sintaxis** es asi:

```
for( variableInt = 0; condicion de bucle; variableInt++)
{
        codigo
}
```

Veamos, la primer parte de variableInt = 0 se refiere al **valor inicial** de la variable que es la que se va a tomar como referencia para saber cuánto repetimos este bucle. El tercer campo “variableInt++” se refiere a que **aumentamos en 1** el valor de la variable (siempre hablamos de enteros) cada vez que repetimos el bucle. Y por último, el campo del medio será la **condición** que **mientras se cumpla, repetirá el bucle**. Aquí también nombraremos a la variableInt. Por ejemplo: variableInt < 3 hará que se ejecute 3 veces el bucle.

```c
#include <stdio.h>

int main ()
{
     //creo variable entera
     int contador;

     //Le asigno el valor 0

     contador = 0;

    for (contador = 0; contador < 5; contador++)
    {
        printf("Repeticion de bucle");
    }
}
```

Bueno, aunque vimos todo rápido y como un manual de un software haré un ejemplo para que luego ustedes puedan practicar y hacer cosas propias.

Antes fijémosnos cómo es la tablita de los **operadores de comparación**, que me acabo de dar cuenta que no lo puse en toda la clase.

Gracias al Dr Febles :D. Sé que dice Java, pero para ambos lenguajes es igual.

Ejemplo: Sumar 3 números enteros en un resultado. El bucle se repite hasta que el número total sea mayor a 10

Primero debemos saber que la cuenta en el loop para el resultado es:

resultado = resultado + var1 + var2 + var3;

Esto se repite 5 veces al menos. Esto lo tenemos que contar. Yo voy a usar un “for” pero no quiere decir que ustedes no puedan usar otra técnica.

for(i = 0; i <5 ; i++)

El bucle es volver a sumar los 3 números si es que aún la cuenta no es mayor a 100. Y en cada suma debemos imprimir en pantalla el resultado. El bucle lo haré con un “while”.

while(resultado < 100)

Veamos cómo me quedaría el código:

```c
#include <stdio.h>

int main()
{
    //creo variables
    int numeroEntero1;
    int numeroEntero2;
    int numeroEntero3;
    int resultado;
    int i;

    numeroEntero1 = 1;
    numeroEntero2 = 3;
    numeroEntero3 = 5;

    for(i = 0; i < 5; i++)
    {
          resultado = resultado + numeroEntero1 + numeroEntero2 +
          printf("El resultado es: %i/n", resultado);
    }

    while(resultado < 100)
    {
          resultado = resultado + numeroEntero1 + numeroEntero2 +
          printf("El resultado es: %i/n", resultado);
    }
}
```

En donde saco la cuenta de resultado no se nota el final pero ya dije arriba la fórmula:D.

Hay un pequeño error en el código que me di cuenta luego de ejecutarlo. En el printf uso “/n” pero la barra debería ser invertida “\n”. Esto se usa para que el programa deje el cursor en el renglón de abajo y pueda escribir desde allí.

El resultado sería:

```
C:\Users\w7\Desktop>Untitled1.exe
El resultado es: 24
El resultado es: 33
El resultado es: 42
El resultado es: 51
El resultado es: 60
El resultado es: 69
El resultado es: 78
El resultado es: 87
El resultado es: 96
El resultado es: 105
```

Ejercicios para hacer:

1. **De tres variables, mostrar el valor mas alto.**
2. **Generar y mostrar la tabla de 10 múltiplos de 5 y especificar cuáles son pares**

Esto fue todo por esta clase. Nos vemos en la próxima :D

Pueden seguirme en Twitter: @RoaddHDC

Cualquier cosa pueden mandarme mail a: r0add@hotmail.com

Para donaciones, pueden hacerlo en bitcoin en la dirección siguiente: 1HqpPJbbWJ9H2hAZTmpXnVuoLKkP7RFSvw

Roadd.
