

Decisiones

Decisiones

- Las decisiones son una medida que utilizamos para controlar el flujo de nuestro algoritmo, así pues por medio de una sentencia, podremos evaluar datos los cuales nos llevaran hacia uno de los varios posibles caminos que podríamos definir en nuestro programa.

Decisiones

- Podremos denotar la decisión por medio de la palabra “if” la cual le seguirá de unos paréntesis la cual tendrá la condición que se desea evaluar, finalmente se deberá abrir llaves (“{ }”) luego de los paréntesis para introducir las sentencias que ocurrirán si esta condición ocurre. Si la condición no ocurre entonces el programa no ejecutara esta sección de código. Para la condición es probable que necesitemos hacer comparaciones, para esto se definirá algunos símbolos que utilizaremos

Decisiones

- $>$: lo utilizaremos para cuando necesitemos decir que un numero cualquiera “X” es mayor a un numero cualquiera “Y” ($X > Y$)
- $<$: se utilizaría contrariamente al anterior, es decir un numero cualquiera “X” es menor que un numero cualquiera “Y” ($X < Y$)
- \geq : Este es el símbolo para “Mayor o igual” , lo utilizaremos para decir que un numero cualquiera “X” es mayor o igual un numero cualquiera “Y” ($X \geq Y$)
- \leq : opuesto al anterior, un numero cualquiera “X” es mayor o igual a un numero cualquiera “Y” ($X \leq Y$)
- $==$: finalmente esta operación la utilizaremos para decir que un numero es igual a otro, cabe resaltar que no se debe confundir con solo el “=” pues uno solo indica Asignación.
- $!=$: para cuando en la decisión se requiera que un dato A no sea igual a un dato B.

Decisiones

Es posible que en un mismo if puedan haber mas de una condición (2 o mas). Para poder evaluarlas en conjunto si se necesitan cumplir las dos o solo una de ellas deberemos dictarle a la maquina las instrucciones :

- `&&` : sirve para decir “Y” , por ejemplo si se debe cumplir la condición A “Y” la condición B, es decir, ambas.
- `||` : sirve para decir “O”, si tenemos que cumplir la condición A “O” La B, es decir, al menos una de ellas.
- `!` : este sirve para negar una condición, es decir si NO se cumple la decisión A entonces, puede ejecutarse la sección del código en el if.
- Es importante que las condiciones tengan su propio paréntesis cada una para utilizar estas operaciones, es decir, `(A)&&(B)` por poner un ejemplo.

Decisiones

- Si necesitamos denotar también una acción que se deba hacer para cuando no se cumpla el “if”, deberemos usar “else” que se traduciría como “de otra manera” , el cual solo necesitara declararse y luego abrir llaves (“{ }”).
- Si necesitamos denotar también una acción que se deba hacer para cuando no se cumpla el “if”, deberemos usar “else” que se traduciría como “de otra manera” , el cual solo necesitara declararse y luego abrir llaves (“{ }”).
- Por ejemplo realizaremos un programa que diga si un numero es par o no. Para esto deberemos entender primero que cualquier numero es par si y solo si cuando se divida entre 2 tenga un resto igual a 0.

Decisiones

```
C main.c x
1  #include<stdio.h>
2
3
4  int main () {
5
6      int a;
7
8      printf("Introduzca numero\n");
9      scanf("%i",&a);
10
11     if(!(a%2)){
12
13         printf("El numero es par \n");
14
15     }
16     else {
17
18         printf("El numero es impar \n");
19
20     }
21
22     return 0;
23 }
24
```

Decisiones

- Es probable que necesitemos evaluar mas de uno o dos casos de decisión para un problema en nuestro código. Para esto podríamos usar varios “if” y resolver el problema, sin embargo hay una solución mucho mas elegante y sofisticada empleando la instrucción “switch”,
- Para hacer un switch solo hace falta escribir “switch” seguido del dato que se desee evaluar entre paréntesis y seguidamente abriendo llaves. Este se encargara de anidar casos que se denotaran por la palabra “case” seguidos del resultado que debería tener la variable a evaluar para que entre en el caso, a esto se le seguirán unos dos puntos (“:”) seguidamente de las sentencias que se deseen declarar para el caso, el cual marcara fin al caso con la instrucción “break” (Interrumpir).

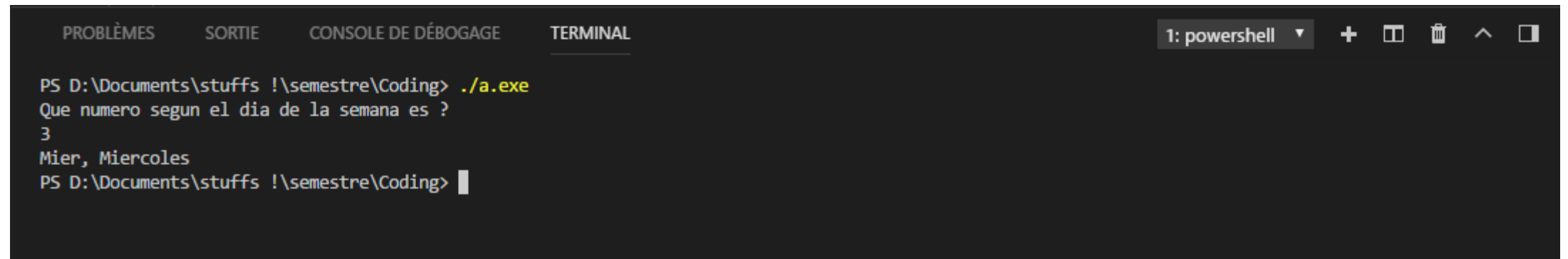
Decisiones

- Finalmente si se desea un caso por defecto el cual atrape algún otro caso no especificado se deberá escribir “default” como si fuera otro caso el cual seguirá la misma sintaxis de los otros casos.
- Por ejemplo, desarrollemos un programa que diga el día de la semana según el numero que se introduzca.
- El programa debe dar 1 para lunes , 2 para martes, y así sucesivamente hasta 7 para domingo. Si el usuario mete otro numero se debe mostrar en pantalla que ese día de la semana no existe.

Decisiones

```
C main.c x
1  #include<stdio.h>
2
3  int main() {
4
5      int dia;
6
7      printf("¿Qué número de día de la semana es?");
8      scanf("%i",&dia);
9
10     switch(dia) {
11     case 1 :
12         printf("Lun, Lunes");
13         break;
14     case 2 :
15         printf("Mar, Martes");
16         break;
17     case 3 :
18         printf("Mier, Miercoles");
19         break;
20     case 4 :
21         printf("Jue, Jueves");
22         break;
23     case 5 :
24         printf("Vie, Viernes");
25         break;
26     case 6 :
27         printf("Sab, Sabado");
28         break;
29     case 7 :
30         printf("Dom, Domingo");
31         break;
32     default :
33         printf("No existe");
34     }
35
36     return 0;
37 }
```

Decisiones



A screenshot of a Visual Studio Code terminal window. The terminal has a dark background with light gray text. At the top, there is a tab bar with four tabs: "PROBLÈMES", "SORTIE", "CONSOLE DE DÉBOGAGE", and "TERMINAL". The "TERMINAL" tab is selected and underlined. To the right of the tabs, there is a dropdown menu showing "1: powershell" with a downward arrow, followed by icons for adding a new terminal, toggling full screen, deleting the terminal, and a refresh icon. The terminal content shows a PowerShell prompt "PS D:\Documents\stuffs !\semestre\Coding>" followed by the command "./a.exe" in yellow. The output of the command is "Que numero segun el dia de la semana es ?" followed by the number "3" and the text "Mier, Miercoles". The prompt "PS D:\Documents\stuffs !\semestre\Coding>" is shown again at the end of the line.

```
PROBLÈMES  SORTIE  CONSOLE DE DÉBOGAGE  TERMINAL  1: powershell ▼ + [X] [X] ^ [X]

PS D:\Documents\stuffs !\semestre\Coding> ./a.exe
Que numero segun el dia de la semana es ?
3
Mier, Miercoles
PS D:\Documents\stuffs !\semestre\Coding> 
```