	Carátula para entrega de prácticas	
Facultad de Ingeniería		Laboratorio de docencia

Laboratorios de computación Salas A y B

Profesor: Alejandro Pimentel

Asignatura: Laboratorio de protraction

Grupo: 135

No de Práctica(s): Práctica 8

Areli González Segura

Integrante(s):

*No. de Equipo de
cómputo empleado:* 23

No. de Lista o Brigada: 5319 no. Lista: 19

Semestre: Primer semestre

Fecha de entrega: 7/octubre/2019

Observaciones: Bastante bien, pero te agradecería si en futuras prácticas pones las capturas con mas zoom.
Me facilitaría revisarla mas rápido.

CALIFICACIÓN: 10

Objetivo:

Elaborar programas en lenguaje C que incluyan las estructuras de selección if, if-else, switch y ternaria para la resolución de problemas básicos.

Introducción:

- **If**

La sentencia if permite controlar qué sentencias de una política se ejecutan mediante la evaluación del valor de una expresión para comprobar si es verdadera. La sentencia if, en el lenguaje de políticas de Impact es la misma que la utilizada en los lenguajes de programación como C/C++ y Java.

La sintaxis de una sentencia if consta de la palabra clave if seguida de una expresión booleana encerrada entre paréntesis. Esta expresión está seguida por un bloque de sentencias delimitado por llaves de cierre. Opcionalmente, la sentencia if puede estar seguida por las palabras clave else o elseif, que también están seguidas por un bloque de sentencias.

- **Switch**

Switch case es una estructura de control empleada en programación. Se utiliza para agilizar la toma de decisiones múltiples; trabaja de la misma manera que lo harían sucesivos if, if else o until anidados, así como combinaciones propias de determinados lenguajes de programación.

El switch no es tan popular como el if, pero se utiliza con regularidad en la programación. En principio la funcionalidad de un switch también se puede implementar con múltiples if anidados. En el caso de que haya muchas acciones dependientes de muchos valores iniciales, es recomendable su uso.

El switch favorece la Facilidad y rapidez en la programación.

El switch no solo te ayuda en ciertos casos. Si no que te permite realizar " Opciones " Que en un if no lo pudieras hacer nada"

- **Condicional**

Un **condicional**, como su nombre lo indica, es una condición para discernir entre una opción u otra, y en el proceso mental normalmente se manifiesta con un "Si"; por ejemplo: Si (va a llover), coge el paraguas.

Operadores lógicos

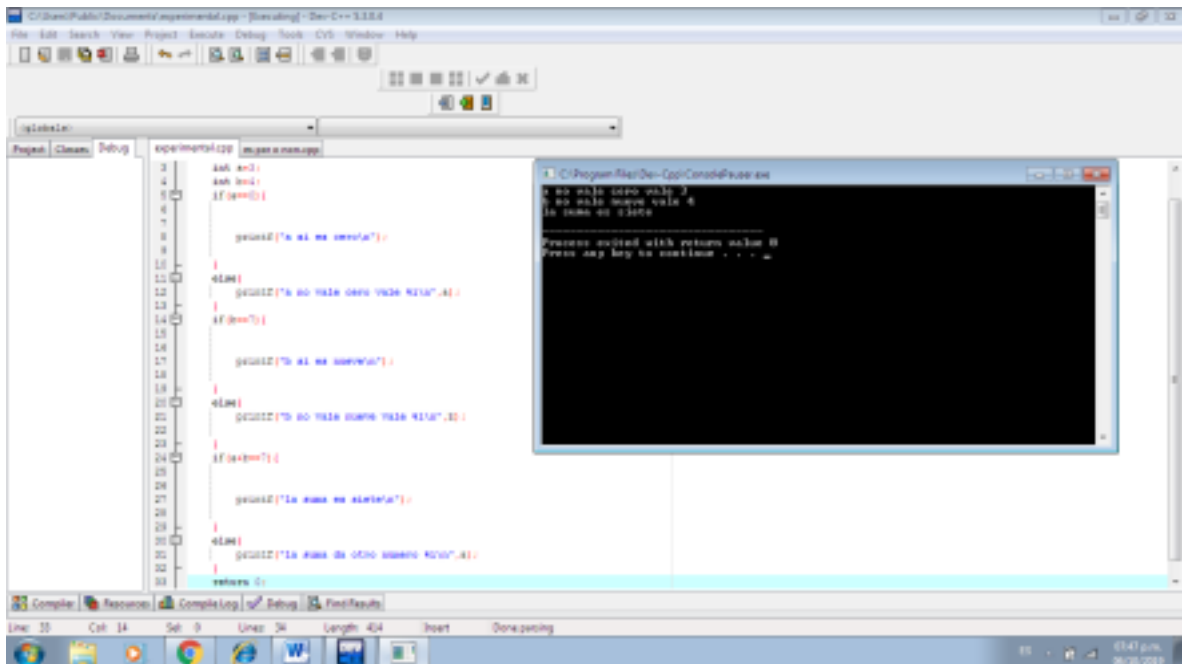
Para crear condiciones, por muy simples que sean, se necesitan los operadores lógicos. A continuación voy a explicarlos de modo que, después de leer este artículo, puedas escribir condiciones ajustadas a lo que necesites.

- *== significa "igual". If $x==y$, significa "si x es igual a y"*
- *> significa "mayor que". If $x>y$, significa "si x es mayor que y"*
- *< significa "menor que". If $x<y$, significa "si x es menor que y"*
- *!= significa "si es distinto". If $x!=y$, significa "si x es distinto de y"*

- `&&` significa “Y”, la conjunción copulativa; es decir, *If* $(x==y) \ \&\& \ (x==z)$, significa “si x es igual a y Y x igual a z ”
- `||` significa “O”, la conjunción adversativa; es decir, *If* $(x==y) \ || \ (x==z)$, significa “si x es igual a y O x igual a z ”

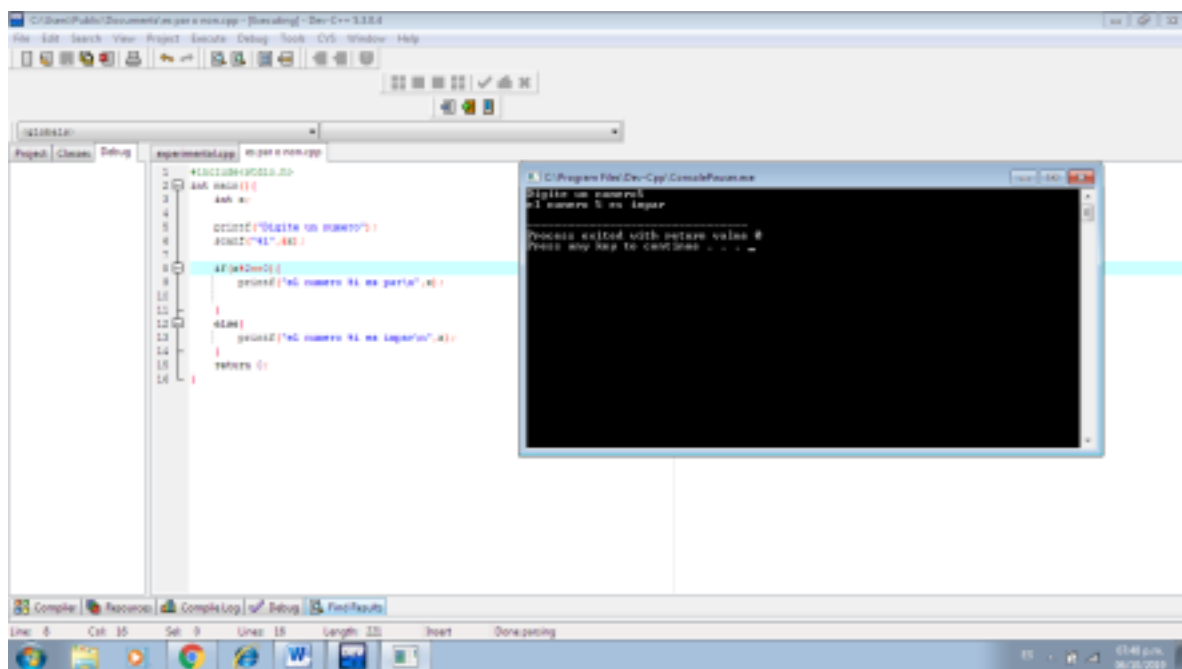
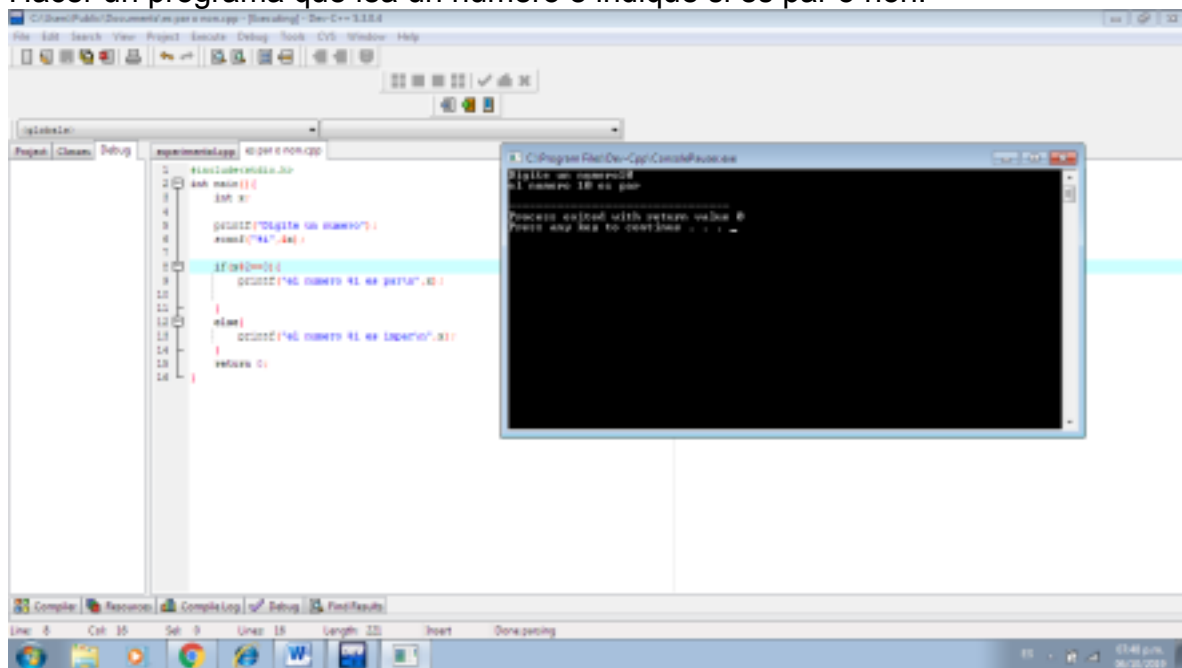
Actividad 1:

Experimentar al utilizar diferentes números dentro del condicional del "if" (asegúrense de usar el 0 entre sus elecciones).



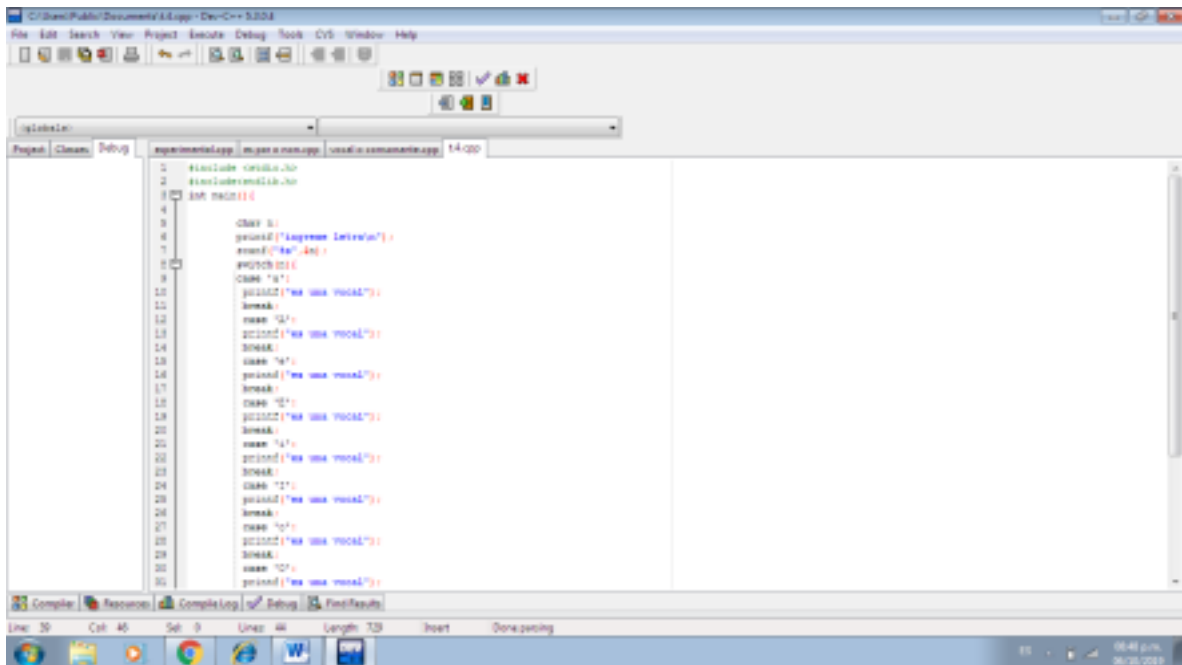
Actividad 2:

Hacer un programa que lea un número e indique si es par o non.

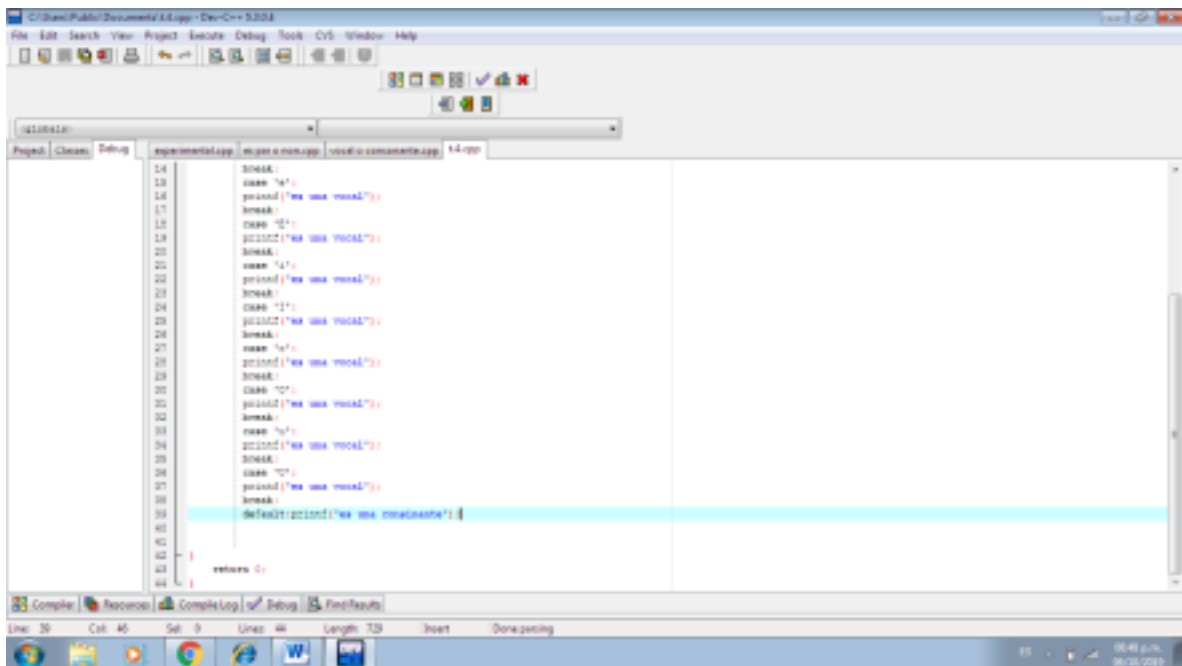


Actividad 3:

Hacer un programa que lea una letra e indique si es vocal o consonante.



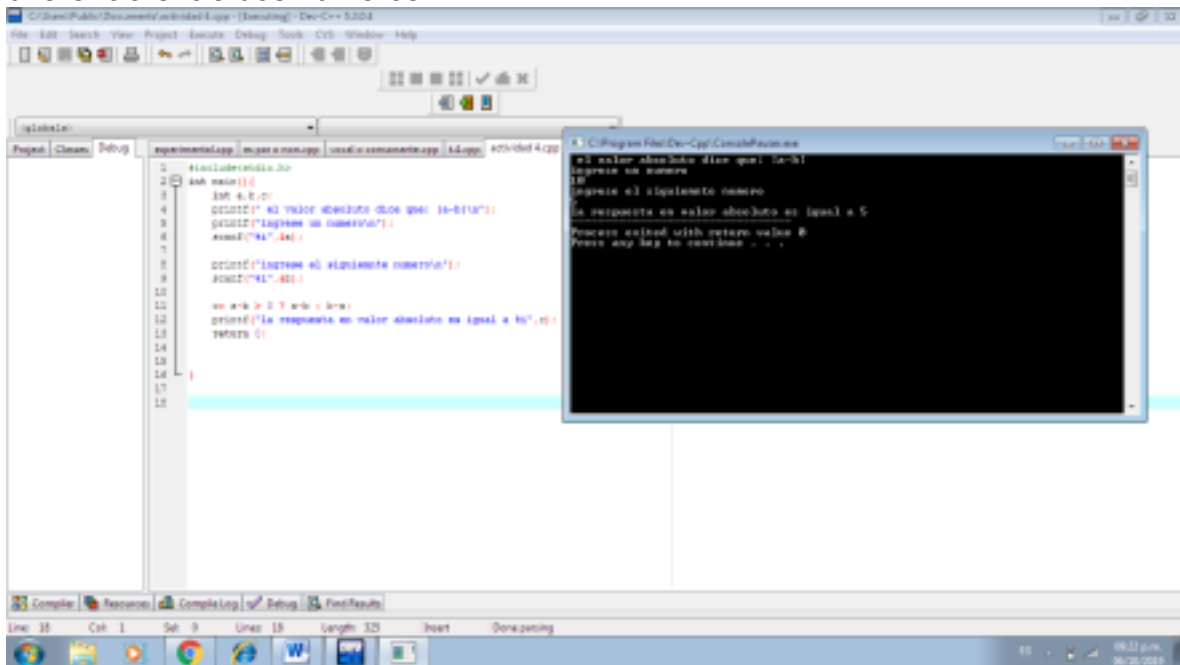
```
1 #include <iostream>
2 #include <conio.h>
3 using namespace std;
4
5 int main()
6 {
7     char c;
8     cout << "Ingrese letra: ";
9     cin >> c;
10    switch(c)
11    {
12        case 'a':
13            cout << "es una vocal";
14            break;
15        case 'e':
16            cout << "es una vocal";
17            break;
18        case 'i':
19            cout << "es una vocal";
20            break;
21        case 'o':
22            cout << "es una vocal";
23            break;
24        case 'u':
25            cout << "es una vocal";
26            break;
27        default:
28            cout << "es una consonante";
29    }
30    return 0;
31 }
```



```
14    cout << "es una vocal";
15    break;
16    case 'e':
17        cout << "es una vocal";
18        break;
19    case 'i':
20        cout << "es una vocal";
21        break;
22    case 'o':
23        cout << "es una vocal";
24        break;
25    case 'u':
26        cout << "es una vocal";
27        break;
28    default:
29        cout << "es una consonante";
30    }
31    return 0;
32 }
```


Actividad 4:

Usar condicional para hacer un programa que obtenga el valor absoluto de la diferencia entre dos números.



The screenshot shows a C++ IDE with a project named 'actividad4'. The main file, 'actividad4.cpp', contains the following code:

```
1 #include <iostream>
2 using namespace std;
3 int a, b;
4 int main()
5 {
6     cout << "El valor absoluto de los que: a-b es:" << endl;
7     cout << "Ingrese un numero a:" << endl;
8     cin >> a;
9     cout << "Ingrese el siguiente numero b:" << endl;
10    cin >> b;
11    int c = a - b;
12    cout << "La respuesta es: el valor absoluto es igual a " << abs(c) << endl;
13    return 0;
14 }
```

The console window shows the execution of the program. It prompts the user to enter two numbers. The first input is 10, and the second input is 5. The program outputs: "La respuesta es: el valor absoluto es igual a 5".

Conclusiones:

Programar en lenguaje C implica de mucho cuidado pues aunque tengas la idea de la estructura del programa para resolver un problema, pero se te llega a pasar algún error en cuestión de la escritura, tu programa no funcionará.

A mí, personalmente, lo que más me cuesta trabajo es recordar algunos comandos como por ejemplo: como se guarda una variable como entero y como se guarda un carácter ya que no es lo mismo. Cabe mencionar que la estructura del programa también es algo sumamente importante porque, por ejemplo: si declaras las variables después de poner las operaciones o lo que quieras que haga tu programa, no funcionará.

Por otra parte, estas prácticas me están dejando ver lo útil que es el lenguaje de programación, pues es algo que facilita muchas actividades.