

# Recomendaciones para Introducción a la computación.

## (Errores comunes y sugerencias)

Autores:

[Jorge Chiquín](#)

[Jonathan Falcon](#)

---

En el int main() en el mismo orden que coloquen las interfaces, en ese MISMO orden saldrán en la consola.

---

No usen la "ñ" o acentos, por ejemplo si crearán una clase "compañia", pongale "compania". Es porque c++ es en inglés y la ñ o acentos no existen en inglés.

---

La interfaz que se coloca en el ciclo repetitivo es en la mayoría de las veces la que corresponde la clase "menor", recuerden que la mayor engloba a la menor o en palabras de estructuras la clase menor es subconjunto de la mayor, de otra forma la que va mayormente en el ciclo repetitivo es la interfaz de la clase que no tiene los contadores y acumuladores

Ejemplo:

Una tienda desea procesar las ventas que realizo en el día de cada una de las 25 ventas se tiene tipo "M" mayor –"D" detal y el monto además de la tienda se tiene el nombre, se pide mostrar por la tienda en monto total recaudado por las ventas

//para este ejemplo solo se tomara en cuenta el ciclo repetitivo

```
Int main()
```

```
{
```

```
Tienda tien;
```

```
InterfaceTienda intTie;
```

```
Venta ven;
```

```
InterfaceVenta intVen;
```

```
string nom;
```

```
int monto;
```

```
char tipo;
```

```
nom=intTie.leerNombre(); //la interfaz de la tienda solicita el nombre de la tienda una sola vez
```

```
tie.setNombre(nom);
```

```
for (int i=0; i<25; i++) //ciclo for que permite procesar 25 ventas
```

```
{
```

```
tipo=intVenta.LeerTipo();//se solicitan los datos de la venta como lo es el tipo y el monto dentro del ciclo repetitivo
```

```
monto=intVenta.Leermonto();
```

```
ven.setTipo(tipo);
```

```
ven.setMonto(monto);
```

```
tie.procesarVentas(ven); // aquí se acumulan los montos por cada una de las ventas
```

```
intVen.reporteDeVenta(ven.getTipo(), ven.GetMonto());//aquí se genera el reporte de las ventas por cada una
```

```
}
```

```
intTie.ReporteTienda(tie.getNombre, tien.getAcumMontos); // salida con frecuencia única de la tienda
return 0;
}
```

---

-Si les sale este error: "cannot open output file Ejercicio4.exe: Permission denied" eso pasa porque tienen la consola activa, es decir estuvieron probando el programa y le dieron a compilar con la consola aún pidiendo datos, para prevenir eso denle al cuadrado rojo de detener en la consola antes de hacer cambios. La solución a ese error es o cerrar eclipse y volverlo a abrir o en el project explorer seleccionan la carpeta del ejercicio > click derecho > Close Project luego Open Project.. Compilan y listo o también pueden eliminar la carpeta debug de su proyecto y vuelven a compilar.

---

-Siempre que tengan errores guarden todo (shift+ctrl+s) y compilen. Muchos errores son porque hacen un cambio en un archivo y como no guardan eclipse no reconoce ese cambio aún.

---

-Si en el project explorer al ejercicio no le sale: src, debug, binaries e includes. No correrá por más que el código esté bien. Sin include no reconoce las librerías, sin binaries el .exe (osea el ejecutable que usa la consola) no existe, por tanto les sale el típico "Binaries not found"... ¿la solución? creen el proyecto de nuevo y sólo copien y peguen las clases y el principal.

---

### **Para importar:**

-Para pasar un ejercicio de un pendrive al eclipse, copien la carpeta del ejercicio desde su pendrive y ponganlo en el workspace: "Dirección donde se guardan sus programas", ahora van a eclipse normal y denle Import y ahí busquen la carpeta General > Existing Projects into Workspace, saldrá una ventana denle en Browse y busquen en el workspace la carpeta de su ejercicio, seleccionenla y denle finish... Así cargará todo el proyecto (con binaries, includes y códigos) y podrán trabajar normal.

-También pueden copiar su proyecto en el workspace y en vez de import pueden darle a create new project "crear un proyecto nuevo" y le colocan exactamente el mismo nombre de la carpeta donde está contenido su proyecto así les aparecerá una advertencia diciendo que el proyecto ya existe le dan finish y les cargará su proyecto.

-Otra forma un tanto más trivial es crear un proyecto nuevo y pegarle los documentos de texto del proyecto que ya tienen y luego compilan.

---

-"undefined reference to `WinMain@16'" Ese error es simplemente porque no han definido el int main(), cuando lo definan se irá.

---

-LAS ÚNICAS COSAS que van en las llaves del ciclo repetitivo (sea for o while) son: las interfaces que se tienen que repetir, los contadores, acumuladores, menores y mayores (los void de esos, no sus get)... NO COLOQUEN cálculos, promedios, porcentajes, etc. esas cosas sólo necesitan hacerse una sola vez.

---

-Los contadores SIEMPRE son enteros (int) los acumuladores si pueden variar (dependen de lo que acumulen). Si hacen un acumulador float, recuerden ponerlo en la clase mayor en el constructor de esta forma: "acumulador=0.00;" ya que si es float (real) tiene decimales, por eso el .00, un mayor siempre se inicializa en 0 (si es int), 0.00 (si es float), un menor siempre se inicializa en un número grande (normalmente 9999).

Para los identificadores (nombres) de contadores, acumuladores, etc. Es recomendable usar "cont", "acum", "mayor", "menor", seguido de lo que cuentan/acumulan, etc. Por ejemplo:

contMujeres (contador de mujeres), contAlumnosAprob (contador de alumnos aprobados), acumSueldo, mayorNota, menorEdad.

Y para los métodos relacionados a estos sería:

void contarMujeres(), void contarAlumnosAprob(), void acumularSueldo(), void determinarMayorNota(), void determinarMenorEdad()

Y sus métodos gets:

getContMujeres(), getContAlumnosAprob(), getAcumSueldo(), getMayorNota(), getMenorEdad()

---

-PARA HACER UN PORCENTAJE: necesitan sólo contadores nada de acumuladores "exceptuando algunos casos especiales", por ejemplo:

El porcentaje de alumnos aprobados de 20 alumnos de una sección: necesito un contador de alumnos aprobados, lo multiplico por 100 y lo divido por la cantidad total de alumnos.

Porcentaje=contador\*100/20;

---

-PARA HACER UN PROMEDIO: necesitan contador y acumulador o en su defecto cantidad y acumulador

Por ejemplo el promedio de notas de alumnos aprobados: Necesitan un contador de alumnos aprobados y un acumulador de notas de esos alumnos aprobados y quedaría:

```
if(contador>0)
promedio=acumulador/contador;
else
promedio=0;
```

Se le coloca ese condicional, porque supongamos que nadie aprobó, es decir el contador y el acumulador estarán vacíos (en 0), entonces quedaría 0/0 y eso no existe, entonces el condicional dice que si en el contador hay 1 (1 es mayor a 0) entonces ahí si hará la división. Y si nadie aprueba simplemente el promedio es 0.

De otra forma:

El promedio de notas de 25 alumnos: No necesitan un contador de alumnos ya que la cantidad de alumnos ya la tenemos y un acumulador de notas de esos alumnos y quedaría:

```
if(contador>0)
promedio=acumulador/25;
else
promedio=0;
```

---

-Mayormente los enunciados tienen una conchita de mango y te dan las clases, por ejemplo en el ejercicio 13 dice al final:

"Mostrar por empleado: XXXXXX "Quiere decir que hay que hacer un reporte por empleado, por tanto Empleado es una clase."Mostrar por la universidad:XXXXXX" Lo mismo que el anterior.

---

NUNCA le cambien el nombre a un proyecto, aunque lo hagan el .exe (osea el ejecutable) quedará con el anterior nombre e inclusive a veces se crean dos .exe (uno por cada nombre) y uno funciona y el otro no. Mejor dejarlo con el nombre inicial.

---

NUNCA hagan un método con el nombre de un atributo, es decir si llaman a un contador: contMujeres, no vayan a poner "void contMujeres()", cámbienle algo (póngale cal o una letra como quieran), por ejemplo "void contarMujeres()"

---

A los condicionales, los for, los while NO LLEVAN ";" de resto todo lo que esté dentro de llaves pónganle ";" al final.

---

Si tienen que agregar un atributo sexo o tipoDeEmpleado, háganlo char (char es alfanum de un sólo carácter), así sólo usan la inicial (por ejemplo sexo: femenino (F) y masculino (M) o tipoDeEmpleado: Docente (D), Obrero (O), etc). Y por ejemplo si tienen que poner

if(sexo=='F') //Osea que si el sexo es F se cumplirá lo siguiente.

Se usan doble igual "==" porque se lee: "getSexo **es exactamente igual a F**", un igual (=) asigna y dos iguales (==) compara, de modo que dentro de un if o while siempre se usa "==". Además, cuando es un sólo carácter se coloca comilla simple ( ' ') y si es ya más de un carácter o un string comillas normales ( " ").

Ahora, si por ejemplo tienen un atributo diaSemana, no podrán usar la inicial porque Martes y Miércoles empiezan igual, podrían hacer ese atributo de tipo entero (int) y así codifican los días con números: (1)Lunes, (2)Martes...(7)Domingo. Obviamente deben especificarle al usuario en la interfaz de entrada como debe indicar el día de la semana.

---

Si quieren hacer un condicional para mujeres con edades entre 30 y 40 sería:

if( (sexo=='F') and ( (edad>=30) and (edad<=40) ) )

Recuerden los paréntesis, un paréntesis mayor para TODA la condición, y uno que agrupe a las condiciones.

---

Y (conjunción) es equivalente en c++ a: "and" también a "&&"

O (disyunción) es equivalente en c++ a: "or" también a "||" (esas barras es con Alt derecho + 1)

Por ejemplo en el while pueden poner: while(respuesta=='S' or respuesta=='s') **equivalente a** while(respuesta=='S' || respuesta=='s')

Así si el usuario se equivoca no pasará nada, acepta la minúscula y mayúscula.

---

Si una clase tiene atributos (exceptuando contadores, acumuladores, etc) DEBE tener en la interfaz métodos de lectura de esos datos

---

Bueno eso es todo, espero aprovechen estas recomendaciones.

---