

Estructuras repetitivas

Ciclos – Bucles

Los ciclos o bucles son sentencias de control, mediante las cuales se logra que una sentencia o más, se ejecuten varias veces **mientras se verifica determinada condición**. El acto de repetir un proceso se denomina **iteración**. Cuando la **condición** es falsa, el bucle se termina.



while	do-while	for
La condición del bucle se evalúa siempre al principio.	La condición se evalúa siempre al final del bucle.	La condición del bucle se evalúa siempre al principio.
Se puede ejecutar 0 (cero) - <i>si la condición es falsa</i> - ó más veces, si la condición evaluada es verdadera.	Aunque la condición sea falsa , se ejecuta al menos una vez.	Se ejecuta siempre y hasta el tope o límite dado.
Para finalizar un bucle, el valor de la variable de control debe ser modificado en las sentencias que componen el bucle.		Para finalizarlo, el valor de la variable de control debe ser modificado con el incremento/ decremento que es parte de la estructura que compone el bucle.

Características comunes a todos los ciclos

Todos los ciclos cuentan con una **expresión de control**, o más, -en este caso unidas por operadores lógicos-, que se evalúa o evalúan cada vez que se produce una iteración.

Las expresiones de control tienen una o más **variables de control** del ciclo, las cuáles deben ser inicializadas.

Los valores de las **variables de control** deben ser modificados en cada iteración.

Todas las **sentencias** o **bloque de instrucciones** encerradas entre las llaves de apertura y cierre del bloque correspondiente al ciclo, se ejecutarán cada vez que la expresión de control es verdadera.

Todos los ciclos deben finalizar, esto requiere que la **expresión de control** se convierta en falsa, y para que esto suceda la **variable de control** deberá alcanzar un valor no contemplado para el ciclo.

Con los ciclos aparecen dos elementos nuevos: **contadores y acumuladores**, necesarios para resolver muchos de los problemas.

El problema:

Supongamos un problema: nos piden que construyamos un programa que calcule el promedio de edad de todos los estudiantes asistentes a la clase de hoy, que se realizará por videoconferencia...



El Análisis:

Qué datos tengo?

- que para hacer el promedio necesito contar los asistentes, pedir y sumar las edades para obtener el promedio,
- que las edades son números enteros positivos > 0 ,
- que los invitados no se puedan conectar,
- que los invitados se puedan conectar a la hora pautada,
- que los invitados se puedan conectar más tarde de lo previsto,

Qué escenarios posibles tengo?

- 1) Ingresar las edades a medida que se conecten pero tengo que considerar la posibilidad de que ninguno lo haga,
- 2) Ingresar al menos una edad porque un asistente confirmó su asistencia previamente,
- 3) Ingresar las edades una vez que todos se conectaron,

Qué escenarios posibles tengo?

- 1) Ingresar las edades a medida que se conecten pero tengo que considerar la posibilidad de que ninguno lo haga,
- 2) Ingresar al menos una edad porque un asistente confirmó su asistencia previamente,
- 3) Ingresar las edades una vez que todos se conectaron,

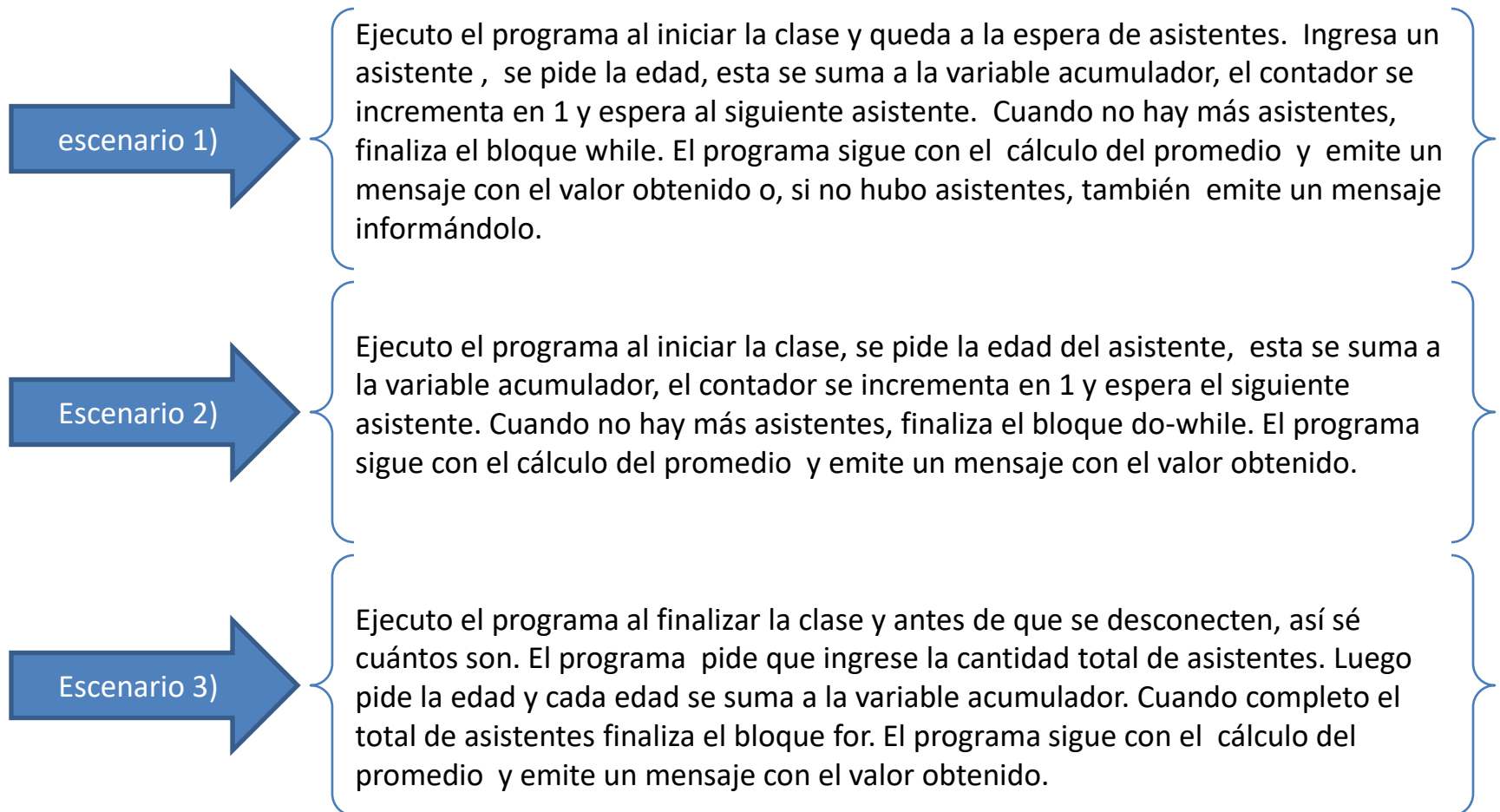
El diseño:

Qué ciclo elijo?

escenario 1) while, porque no sé cuántos asistirán, puede pasar que ninguno.

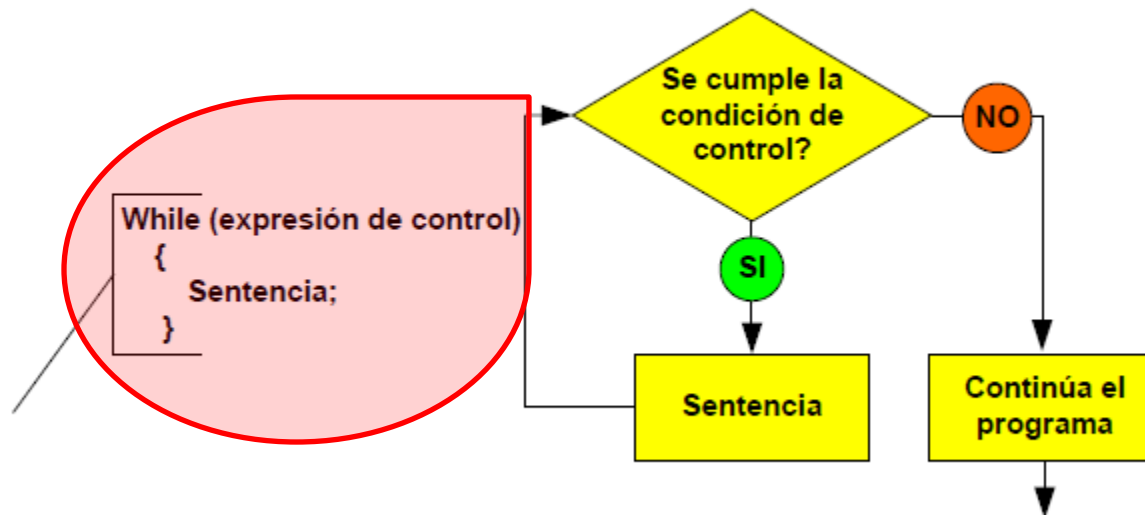
escenario 2) do-while, porque tengo una confirmación de asistencia.

escenario 3) for, porque sé la cantidad de asistentes.



escenario 1)

Ejecuto el programa al iniciar la clase y queda a la espera de asistentes. Ingresa un asistente, se pide la edad, esta se suma a la variable acumulador, el contador se incrementa en 1 y espera al siguiente asistente. Cuando no hay más asistentes, finaliza el bloque while. El programa sigue con el cálculo del promedio y emite un mensaje con el valor obtenido o, si no hubo asistentes, también emite un mensaje informándolo.



La codificación:

```
#include<stdio.h>

int main() {

    int edad, contador_asistentes=0, acumulador_edad=0;
    float promedio_edad;

    printf("Ingrese la edad del primer asistente,\n"
           "Para finalizar ingrese un valor igual %c menor a cero \n",162);
    scanf("%d", &edad);

    while (edad > 0) {

        contador_asistentes++;
        acumulador_edad = acumulador_edad + edad;

        printf("Ingrese la edad del siguiente asistente,\n"
               "Para finalizar ingrese un valor igual %c menor a cero\n",162);
        scanf("%d", &edad);
    }

    promedio_edad = (float) acumulador_edad / contador_asistentes;
    printf("El promedio de edades es %.2f\n", promedio_edad);

    return 0;
}
```


Revisión del código:

```
#include<stdio.h>

int main(){

    int edad, contador_asistentes=0, acumulador_edad=0;
    float promedio_edad;

    printf("Ingrese la edad del primer asistente,\n"
           "Para finalizar ingrese un valor igual %c menor a cero \n",162);
    scanf("%d", &edad);

    while (edad > 0) {

        contador_asistentes++;
        acumulador_edad = acumulador_edad + edad;

        printf("Ingrese la edad del siguiente asistente,\n"
               "Para finalizar ingrese un valor igual %c menor a cero\n",162);
        scanf("%d", &edad);
    }

    promedio_edad = (float) acumulador_edad / contador_asistentes;
    printf("El promedio de edades es %.2f\n", promedio_edad);

    return 0;
}
```

Inicializo la variable de control (edad) antes de comenzar el ciclo.

La variable de control es parte de la expresión de control del ciclo

La variable de control necesita tomar un nuevo valor para ser evaluada nuevamente y saber si sigue iterando o termina el ciclo.

Prueba: Debemos probar todos los casos posibles, extremos e intermedios.

1) Funcionamiento en el caso de un asistente.

```
Ingrese la edad del primer asistente,  
Para finalizar ingrese un valor igual ó menor a cero  
23  
Ingrese la edad del siguiente asistente,  
Para finalizar ingrese un valor igual ó menor a cero  
0  
El promedio de edades es 23.00
```

Ok

2) Funcionamiento en el caso de más de un asistente.

```
Ingrese la edad del primer asistente,  
Para finalizar ingrese un valor igual ó menor a cero  
22  
Ingrese la edad del siguiente asistente,  
Para finalizar ingrese un valor igual ó menor a cero  
23  
Ingrese la edad del siguiente asistente,  
Para finalizar ingrese un valor igual ó menor a cero  
25  
Ingrese la edad del siguiente asistente,  
Para finalizar ingrese un valor igual ó menor a cero  
0  
El promedio de edades es 23.33
```

Ok

3) Funcionamiento en el caso de ningún asistente.

```
Ingrese la edad del primer asistente,  
Para finalizar ingrese un valor igual ó menor a cero  
0  
El promedio de edades es -1.#J
```

Crashhh!!

Corrección de errores: Si analizamos el código, el error de la tercera prueba se produce porque no consideramos que podría no haber asistentes. En una división, implica división por cero, uno de los tantos controles que debemos hacer en nuestros programas.

```
#include<stdio.h>

int main(){

    int edad, contador_asistentes=0, acumulador_edad=0;
    float promedio_edad;

    printf("Ingrese la edad del primer asistente,\n"
           "Para finalizar ingrese un valor igual %c menor a cero \n",162);
    scanf("%d", &edad);

    while (edad > 0) {

        contador_asistentes++;
        acumulador_edad = acumulador_edad + edad;

        printf("Ingrese la edad del siguiente asistente,\n"
               "Para finalizar ingrese un valor igual %c menor a cero\n",162);
        scanf("%d", &edad);
    }

    {
        if (contador_asistentes > 0) {
            promedio_edad = (float)acumulador_edad / contador_asistentes;
            printf("El promedio de edades es %.2f\n", promedio_edad);
        }
        else printf("No hubo asistentes a la clase hoy\n");
    }

    return 0;
}
```

Agrego un bloque if que evalúe si hubo asistentes en la clase. De acuerdo a su valor realizo los cálculos y emito el resultado.

Para tener en cuenta

El tipo de ciclo **while** utilizado en el ejemplo es el controlado por variable centinela. *Existe el while controlado por contador.*

La variable centinela que se use puede ser una de las variables que se utilice para el proceso, -como en el ejemplo dado-, u otra, que no tenga que ver con el proceso pero sí con el control de ciclo, por lo tanto, para el ejemplo dado, hay otras soluciones.

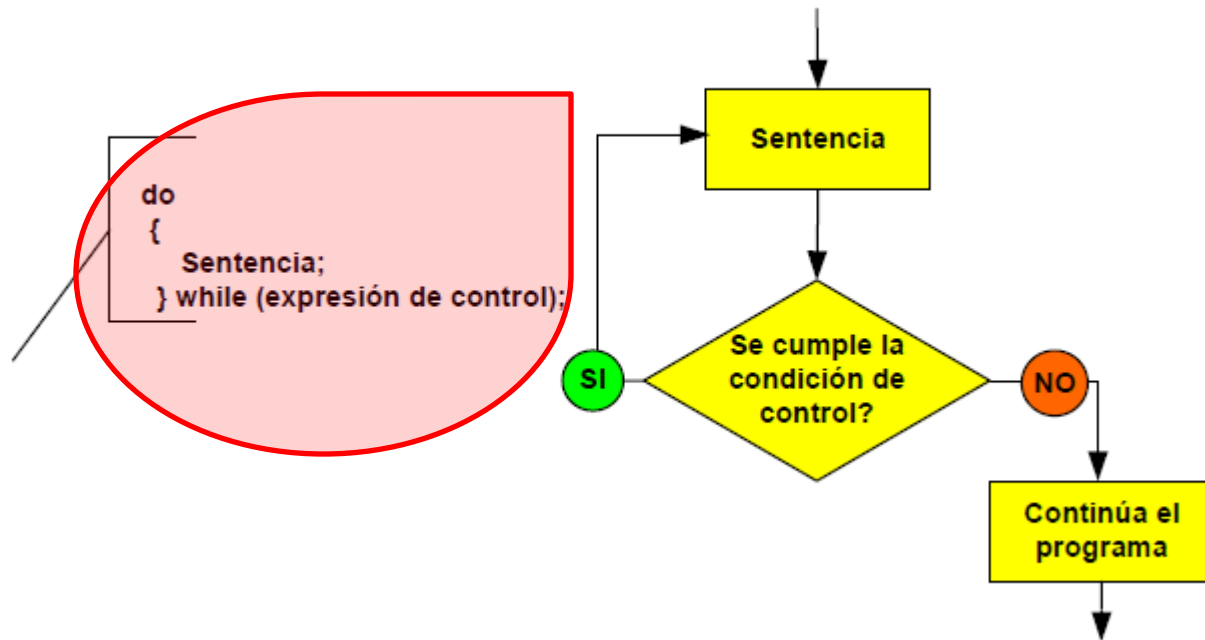
Los contadores se inicializan siempre, dependerá de algoritmo dónde, cuando y cuántas veces se hará. Hay que tener en cuenta el valor de un contador que podría haberse inicializado en 0 cuando se tenga que realizar una multiplicación.

Los acumuladores se inicializan siempre, dependerá de algoritmo dónde, cuando y cuántas veces se hará.

Se pueden resolver la mayoría de los problemas eligiendo cualquiera de los tipos de ciclos.

Escenario 2)

Ejecuto el programa al iniciar la clase, se pide la edad del asistente, esta se suma a la variable acumulador, el contador se incrementa en 1 y espera el siguiente asistente. Cuando no hay más asistentes, finaliza el bloque do-while. El programa sigue con el cálculo del promedio y emite un mensaje con el valor obtenido.



La codificación:

```
#include<stdio.h>

int main(){

    int edad, contador_asistentes=0, acumulador_edad=0;
    float promedio_edad;

    do{

        printf("Ingrese la edad del primer asistente,\n"
               "Para finalizar ingrese un valor igual %c menor a cero \n",162);
        scanf("%d", &edad);

        if (edad > 0 ){
            acumulador_edad = acumulador_edad + edad;
            contador_asistentes++;
        }

    } while (edad > 0);

    if (contador_asistentes > 0) {
        promedio_edad = (float)acumulador_edad / contador_asistentes;
        printf("El promedio de edades es %.2f\n", promedio_edad);
    }
    else printf("No hubo asistentes a la clase hoy\n");

    return 0;
}
```

La codificación:

```
#include<stdio.h>

int main(){

    int edad, contador_asistentes=0, acumulador_edad=0;
    float promedio_edad;

    do{

        printf("Ingrese la edad del primer asistente,\n"
               "Para finalizar ingrese un valor igual %c menor a cero \n",162);
        scanf("%d", &edad);

        if (edad > 0 ){
            acumulador_edad = acumulador_edad + edad;
            contador_asistentes++;
        }

    } while (edad > 0);

    if (contador_asistentes > 0) {
        promedio_edad = (float)acumulador_edad / contador_asistentes;
        printf("El promedio de edades es %.2f\n", promedio_edad);
    }
    else printf("No hubo asistentes a la clase hoy\n");

    return 0;
}
```

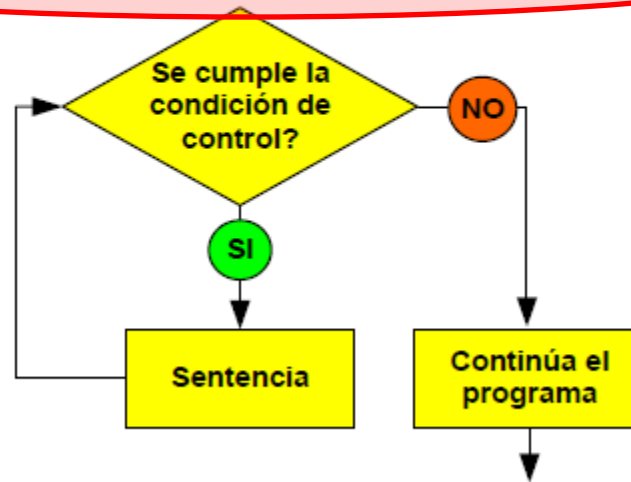
La variable de control necesita tomar un valor para el proceso, bien para iterar o para finalizar el ciclo.

La variable de control es parte de la expresión de control del ciclo .

Escenario 3)

Ejecuto el programa al finalizar la clase y antes de que se desconecten, así sé cuántos son. El programa pide que ingrese la cantidad total de asistentes. Luego pide la edad y cada edad se suma a la variable acumulador. Cuando completo el total de asistentes finaliza el bloque for. El programa sigue con el cálculo del promedio y emite un mensaje con el valor obtenido.

```
for (variable de control; (expresión de control); incremento o decremento)  
Sentencia;
```



La codificación:

```
#include<stdio.h>

int main(){

    int edad, contador_asistentes, acumulador_edad=0, cantidad_alumnos;
    float promedio_edad;

    printf("Ingrese la cantidad de alumnos que sistieron a la clase de hoy\n");
    scanf("%d", &cantidad_alumnos);

    for (contador_asistentes=0; contador_asistentes < cantidad_alumnos; contador_asistentes++) {

        printf("Ingrese la edad del siguiente asistente\n");
        scanf("%d", &edad);

        acumulador_edad = acumulador_edad + edad;

    }

    if (contador_asistentes > 0) {
        promedio_edad = (float)acumulador_edad / contador_asistentes;
        printf("El promedio de edades es %.2f\n", promedio_edad);
    }
    else printf("No hubo asistentes a la clase hoy\n");

    return 0;
}
```

La codificación:

```
#include<stdio.h>
```

```
int main(){
```

```
    int edad, contador_asistentes, acumulador_edad=0, cantidad_alumnos;  
    float promedio_edad;
```

```
    printf("Ingrese la cantidad de alumnos que sistieron a la clase de hoy\n");  
    scanf("%d", &cantidad_alumnos);
```

```
    for (contador_asistentes=0; contador_asistentes < cantidad_alumnos; contador_asistentes++) {
```

```
        printf("Ingrese la edad del siguiente asistente\n");  
        scanf("%d", &edad);
```

```
        acumulador_edad = acumulador_edad + edad;
```

```
    }
```

```
    if (contador_asistentes > 0) {  
        promedio_edad = (float)acumulador_edad / contador_asistentes;  
        printf("El promedio de edades es %.2f\n", promedio_edad);  
    }
```

```
    else printf("No hubo asistentes a la clase hoy\n");
```

```
    return 0;
```

```
}
```

Inicializo la variable de control (control _asistentes)

La variable de control se incrementa porque debe modificarse para ser comparada en la siguiente iteración y así saber si continúa o termina el ciclo.

La variable de control es parte de la expresión de control del ciclo y será comparada con el tope (cantidad_alumnos)

Sabiendo la cantidad de asistentes anticipadamente, elegimos un ciclo for, pero también podríamos haber elegido un ciclo while controlado por contador. Y cómo sería?...

```
#include<stdio.h>

int main() {

    int edad, contador_asistentes=0, acumulador_edad=0, cantidad_alumnos;
    float promedio_edad;

    printf("Ingrese la cantidad de alumnos que sistieron a la clase de hoy\n");
    scanf("%d", &cantidad_alumnos);

    while (contador_asistentes < cantidad_alumnos) {

        printf("Ingrese la edad del asistente,\n");
        scanf("%d", &edad);

        acumulador_edad = acumulador_edad + edad;
        contador_asistentes++;

    }

    if (contador_asistentes > 0) {
        promedio_edad = (float)acumulador_edad / contador_asistentes;
        printf("El promedio de edades es %.2f\n", promedio_edad);
    }
    else printf("No hubo asistentes a la clase hoy\n");

    return 0;
}
```

```
#include<stdio.h>
```

```
int main(){
```

```
    int edad, contador_asistentes=0, acumulador_edad=0, cantidad_alumnos;  
    float promedio_edad;
```

```
    printf("Ingrese la cantidad de alumnos que sistieron a la clase de hoy\n");  
    scanf("%d", &cantidad_alumnos);
```

```
    while (contador_asistentes < cantidad_alumnos)
```

```
    {  
        printf("Ingrese la edad del asistente,\n");  
        scanf("%d", &edad);
```

```
        acumulador_edad = acumulador_edad + edad;  
        contador_asistentes++;  
    }
```

```
    if (contador_asistentes > 0) {  
        promedio_edad = (float)acumulador_edad / contador_asistentes;  
        printf("El promedio de edades es %.2f\n", promedio_edad);  
    }
```

```
    else printf("No hubo asistentes a la clase hoy\n");
```

```
    return 0;  
}
```

Inicializo la variable de control (contador_asistentes) antes de comenzar el ciclo.

Igual que en el ciclo for, le doy un tope

La variable de control es parte de la expresión de control del ciclo

La variable de control necesita ser modificada para ser evaluada nuevamente y saber si sigue iterando o termina el ciclo.

Para empezar:

Construir y probar el caso de do-while

Construir y probar el caso del for

Construir y probar el caso del while controlado por contador.

FIN