

¿Cómo funcionan los constructores?

Un TDA simple con dos constructores

Vamos viendo con ejemplos simples.

Archivo fecha.h (Clase Fecha con métodos básicos)

```
#ifndef FECHA_H_INCLUDED
#define FECHA_H_INCLUDED

class Fecha {

private:
    // atributos
    int dia;
    int mes;
    int anio;

public:
    // metodos
    // Constructor sin parametros
    // PRE: -
    // POS: construye una Fecha con 01/01/2000
    Fecha();

    // Constructor con parametros
    /* PRE: los parametros son enteros validos
       d: entre 1 y 28, 29, 30 o 31 segun el mes y anio
       m: entre 1 y 12
       a: entero positivo
       POS: inicializa dia con d, mes con m y anio con a
    */
    Fecha(int d, int m, int a);

    // metodos asignar
    // PRE: d es un entero valido
    //      (1 a 28, 29, 30 o 31 dependiendo del mes y anio)
```

```

// POS: dia queda con el valor d
void asignar_dia(int d);

// PRE: m es un entero entre 1 y 12 inclusive
// POS: mes queda con el valor m
void asignar_mes(int m);

// PRE: a es un entero positivo
// POS: anio queda con el valor a
void asignar_anio(int a);

// Metodos para obtener los atributos
// PRE: -
// POS: retorna el valor de dia
int obtener_dia();

// PRE: -
// POS: retorna el valor de mes
int obtener_mes();

// PRE: -
// POS: retorna el valor de anio
int obtener_anio();
};

#endif // FECHA_H_INCLUDED

```

Archivo fecha.cpp (implementación de la clase Fecha)

```

#include "fecha.h"
#include <iostream>

// constructor por defecto
Fecha::Fecha() {
    dia = 1;
    mes = 1;
    anio = 2000;
    std::cout << "Constructor de fecha sin parametros" << std::endl;
}

```

```
}

// constructor con parametros
Fecha::Fecha(int d, int m, int a) {
    dia = d;
    mes = m;
    anio = a;
    std::cout << "Constructor de fecha con parametros" << std::endl;
}

// asigna dia
void Fecha::asignar_dia(int d) {
    dia = d;
}

// asigna mes
void Fecha::asignar_mes(int m) {
    mes = m;
}

// asigna anio
void Fecha::asignar_anio(int a) {
    anio = a;
}

// obtiene dia
int Fecha::obtener_dia() {
    return dia;
}

// obtiene mes
int Fecha::obtener_mes() {
    return mes;
}

// obtiene anio
int Fecha::obtener_anio() {
```

```
    return anio;
}
```

En los constructores imprimo un cartel solo para ver en qué momento se llaman, esto es válido para una implementación de prueba pero no tendría sentido para una en producción.

Archivo main.cpp (acá se usa la clase Fecha)

```
1. #include <iostream>
2. #include "fecha.h"

3. using namespace std;

4. int main()
5. {
6.     Fecha fecha;
7.     cout << "Esta fecha tiene anio: " << fecha.obtener_anio() << endl;
8.     Fecha hoy(22,9,2018);
9.     cout << "Hoy es: " << hoy.obtener_dia() << "/";
10.    cout << hoy.obtener_mes() << "/" << hoy.obtener_anio() << endl;
11.    return 0;
12. }
```

En la línea 6 se llama al constructor por defecto (sin parámetros). En la 8 al que lleva parámetros.

La salida es:

```
Constructor de fecha sin parametros
Esta fecha tiene anio: 2000
Constructor de fecha con parametros
Hoy es: 22/9/2018
```

Un vector de objetos

Supongamos que creamos vectores de objetos de la clase Fecha. Cambiamos el main por:

```
1. int main()
2. {
3.     Fecha fechas[5];
4.     cout << "Estas fechas son del anio: " << fechas[0].obtener_anio() << endl;
5.     Fecha varias[3] = { Fecha(2,1,2015), Fecha(1,5,2016), Fecha(3,3,2018) };
6.     for (int i = 0; i < 3; i++)
7.         cout << "Estas fechas son de los anios: " << varias[i].obtener_anio() << endl;
7.     return 0;
8. }
```

En la línea 3 creamos 5 objetos de tipo Fecha con el constructor por defecto, es decir que los 5 tienen la fecha 1/1/2000.

En cambio en la línea 5 creamos 3 objetos utilizando el constructor con parámetros, por lo tanto tenemos que crear esos objetos y decir que corresponden al vector que llamamos *varias*.

Como vemos, esto es bastante molesto, y si tuviéramos un vector de 100 posiciones sería muy incómodo inicializar todas las fechas, por eso es conveniente contar con un constructor sin parámetros.

La salida es:

```
Constructor de fecha sin parametros
Constructor de fecha sin parametros
Constructor de fecha sin parametros
Constructor de fecha sin parametros
Constructor de fecha sin parametros
Estas fechas son del anio: 2000
Constructor de fecha con parametros
Constructor de fecha con parametros
Constructor de fecha con parametros
Estas fechas son de los anios: 2015
Estas fechas son de los anios: 2016
Estas fechas son de los anios: 2018
```

Objetos que contienen otros objetos

Vamos a implementar una clase Viaje que tiene un número de línea del colectivo y una Fecha. Lo hacemos en dos archivos aparte (un .h y el .cpp para implementación)

Archivo viaje.h

```
#ifndef VIAJE_H_INCLUDED
#define VIAJE_H_INCLUDED

#include "fecha.h"

class Viaje {
```

```

private:
    // atributos
    int linea;
    Fecha fecha;

public:
    // metodos
    // constructor sin parametros
    /*
    PRE: -
    POS: en linea deja 100 y fecha 1/1/2000
    */
    Viaje();

    /*
    Constructor con 4 parametros: linea, dia, mes y anio
    PRE: l es una linea de colectivo valida
           d es un entero entre 1 y 28, 29, 30 o 31 segun m y a
           m es un entero entre 1 y 12
           a es un entero positivo
    POS: linea queda con el valor l
           La fecha se crea con d, m y a como dia, mes y anio
    */
    Viaje(int l, int d, int m, int a);

    /*
    Constructor con 2 parametros: linea y Fecha
    PRE: l es una linea de colectivo valida
           f es una Fecha valida
    POS: linea queda con el valor l
           fecha queda con los valores de f
    */
    Viaje(int l, Fecha f);

    // PRE: l es una linea de colectivo valida

```

```

// POS: linea queda con el valor 1
void asignar_linea(int l);

// metodos asignar
// PRE: d es un entero valido
//      (1 a 28, 29, 30 o 31 dependiendo del mes y anio)
// POS: dia del objeto fecha queda con el valor d
void asignar_dia(int d);

// PRE: m es un entero entre 1 y 12 inclusive
// POS: mes del objeto fecha queda con el valor m
void asignar_mes(int m);

// PRE: a es un entero positivo
// POS: anio del objeto fecha queda con el valor a
void asignar_anio(int a);

// Metodos para obtener los atributos
// PRE: -
// POS: retorna el valor de dia del objeto fecha
int obtener_dia();

// PRE: -
// POS: retorna el valor de mes del objeto fecha
int obtener_mes();

// PRE: -
// POS: retorna el valor de anio del objeto fecha
int obtener_anio();

// PRE: -
// POS: retorna el numero de linea
int obtener_linea();
};

#endif // VIAJE_H_INCLUDED

```

Como ven hay tres constructores distintos:

- Uno es sin parámetros, pone la línea en 100 (colectivo 100) y la fecha la crea con el constructor por defecto que pone 1/1/2000.
- Hay otro constructor que recibe 4 datos: el número de línea y los tres datos de la fecha: día, mes y año.
- Por último, hay un constructor que recibe el número de línea y la fecha la recibe como un objeto ya creado.

Archivo viaje.cpp (implementación)

```
#include "viaje.h"
#include <iostream>

// Constructor sin parametros
Viaje::Viaje() {
    linea = 100;
    std::cout << "Constructor de Viaje sin parametros" << std::endl;
}

// Constructor con 4 parametros: linea, dia, mes y anio
Viaje::Viaje(int l, int d, int m, int a) : fecha(d, m, a) {
    linea = l;
    std::cout << "Constructor de Viaje con 4 parametros" << std::endl;
}

// Constructor con 2 parametros: linea y Fecha
Viaje::Viaje(int l, Fecha f) : fecha(f) {
    // fecha = f;
    linea = l;
    std::cout << "Constructor de Viaje con 2 parametros" << std::endl;
}

// asignar linea
void Viaje::asignar_linea(int l) {
    linea = l;
}

// asignar dia
```



```

void Viaje::asignar_dia(int d) {
    fecha.asignar_dia(d);
}

// asignar mes
void Viaje::asignar_mes(int m) {
    fecha.asignar_mes(m);
}

// asignar anio
void Viaje::asignar_anio(int a) {
    fecha.asignar_anio(a);
}

// obtener linea
int Viaje::obtener_linea() {
    return linea;
}

// obtener dia
int Viaje::obtener_dia() {
    return fecha.obtener_dia();
}

// obtener mes
int Viaje::obtener_mes() {
    return fecha.obtener_mes();
}

// obtener anio
int Viaje::obtener_anio() {
    return fecha.obtener_anio();
}

```

Los dos constructores resaltados en celeste utilizan lo que se llama inicializadores. Estas instrucciones van antes del cuerpo del método (antes de la llave de apertura) y después de los parámetros colocando un dos puntos “:”.

En el constructor con dos parámetros, si sacamos el inicializador y descomentamos la línea resaltada en amarillo, lo que haría es primero construir el objeto *fecha* sin parámetros y luego asignarle los valores que recibe de *f*.

Hay que notar que para acceder a *dia*, *mes* y *anio* del objeto *fecha*, debemos hacerlo por medio de su interfaz, ya que son atributos privados, por más que el objeto *fecha* esté dentro de un objeto *viaje*.

En el main ahora debemos incluir *viaje.h*.

```
1. Archivo main.cpp
2. #include <iostream>
3. #include "viaje.h"

4. using namespace std;

5. void imprimir_viaje(Viaje v);

6. int main()
7. {
8.     Viaje viaje_hace_mucho;
9.     imprimir_viaje(viaje_hace_mucho);
10.    Viaje viaje_de_recien(109, 22, 9, 2018);
11.    imprimir_viaje(viaje_de_recien);
12.    Fecha ayer(21, 9, 2018);
13.    Viaje viaje_de_ayer(109, ayer);
14.    imprimir_viaje(viaje_de_ayer);
15.    return 0;
16.}

17. void imprimir_viaje(Viaje v) {
18.     cout << "Este viaje fue: " << endl;
19.     cout << "---- En la linea: " << v.obtener_linea() << endl;
20.     cout << "---- El dia: " << v.obtener_dia() << "/";
21.     cout << v.obtener_mes() << "/" << v.obtener_anio() << endl;
22.     cout << "-----" << "\n\n";
23. }
```

En la línea 8 creamos un objeto de clase *Viaje* con los constructores por defecto (sin parámetros). En la 9 imprimos los datos.

En la línea 10 creamos otro objeto de clase *Viaje* con el constructor que lleva 4 parámetros. Y en la 11 también lo imprimimos.

En la línea 12 primero creamos un objeto de tipo *Fecha*, el cual va por parámetro en la línea 13 al constructor de 2 parámetros de *Viaje*.

La salida queda:

```
Constructor de fecha sin parametros
Constructor de Viaje sin parametros
Este viaje fue:
---- En la linea: 100
---- El dia: 1/1/2000
-----

Constructor de fecha con parametros
Constructor de Viaje con 4 parametros
Este viaje fue:
---- En la linea: 109
---- El dia: 22/9/2018
-----

Constructor de fecha con parametros
Constructor de Viaje con 2 parametros
Este viaje fue:
---- En la linea: 109
---- El dia: 21/9/2018
-----
```

Hay que notar que para cada Viaje, no importa de la forma en que se cree, también se crea un objeto Fecha.