

# Introducción a la Programación con Objetos en C++

Aprendemos lo relacionado a objetos estudiando sus **atributos** y observando su **comportamiento**. Por ejemplo, tenemos un objeto al que llamamos **balón**, cuyos **atributos** son: su tamaño (longitud de la circunferencia), su color (rojo), su peso (liviano), etc. y observando su **comportamiento** (rebota, suena)<sup>1</sup>

Hablar de **Clases** (concepción, idea) y de **Objetos** (el estado materializado de la Clase) en programación es hablar de muchas de las cosas que usamos corrientemente, por ejemplo, nuestra fecha de nacimiento (que es ya la parte material de una idea general llamada **Fecha**). Si decimos: “*nací el 20 de setiembre del año 2005*”, en realidad estamos mencionando el día, el mes y el año como una sola unidad (la **Fecha**), por lo que podemos decir que **fecha\_nacimiento es un Objeto de una Clase llamada Fecha**. El objeto **fecha\_nacimiento** es una **instancia** (una materialización) de la clase **Fecha** y tiene datos miembro que son el *día* (valor entero), el *mes* (valor entero) y el *año* (valor entero), y podrá tener métodos o funciones para el tratamiento de los datos miembro. Por ejemplo, definimos una clase **Fecha** con tres datos miembro:

```
class Fecha
{
    public:
        int dia;
        int mes;
        int año;
};

void main()
{
    Fecha fecha_nacimiento;
    ...
}
```

A partir de este momento, **fecha\_nacimiento** será una **instancia** de la clase **Fecha**. Tendrá todos los atributos que tiene la clase **Fecha**, sus datos miembro (día que es entero, mes que es entero, y año que también será entero). No todos los datos de una Clase deben ser del mismo tipo, pueden ser variados e inclusive haber sido definidos en otras Clases.

Otro ejemplo de uso diario, la **Hora**, cuando preguntamos *¿qué hora es?*, generalmente nos responden: “*son las ocho de la mañana con quince minutos*”, y si tenemos a alguien que le gusta la precisión, nos respondería: “*son las ocho de la mañana, quince minutos y veintiún segundos*”. También podríamos tener una respuesta como “*son*

<sup>1</sup> Deitel P. & Deitel H, *How to program C. With an introduction to C++*

*las cinco de la tarde, con veinte minutos y doce segundos*". En este caso, la clausula **de la tarde** nos hace entender que ha pasado la mitad del día (doce horas del día, o sea medio día), y estamos en las cinco horas en la tarde, por lo que es equivalente a decir "*son las diecisiete horas, veinte minutos y doce segundos*". La concepción o idea (la **Clase**) sería **Hora** (con datos miembro cuyos valores que van desde 0 hasta 23 para la hora, 0 a 59 para los minutos y para los segundos), y la **instancia o las instancias** las especificaríamos de acuerdo a nuestro interés: **Hora\_Entrada** y **Hora\_Salida**. Su definición en el lenguaje de programación C++ sería:

```

Class Hora
{
    public:
        int hora;           // 0-23
        int minutos;       // 0-59
        int segundos;      // 0-59
};

void main()
{
    Hora Hora_Entrada, Hora_Salida;
    ...
}

```

En este caso, hemos hecho **dos** instancias de la **clase Hora: Hora\_Entrada** y **Hora\_Salida**. Cada una tendrá sus respectivos campos hora, minutos y segundos.

Si se dan cuenta, en la definición de las clase **Fecha** y **Hora**, los datos miembro están precedidos por la clausula **public**, lo que significa que estos datos estarán disponibles para cualquier función o sentencia en nuestro programa, para otros programas nuestros o para programas de cualquier otro usuario. Si no deseamos esta situación, tendríamos que declarar a los datos miembro de la clase como **private**. Podemos tener en una clase datos miembro que sean públicos o datos miembro privados, o datos miembro públicos y datos miembro privados en una sola definición.

Volviendo al ejemplo de la fecha, un usuario estaría en condiciones de responder: "*mi cumpleaños es el treinta y dos de marzo del año dos mil veintiocho*", aunque sabemos que nos está respondiendo de acuerdo a lo que le hemos solicitado (día, mes y año), también sabemos que el día y el año no son correctos. Una clase puede estar dotada de "**métodos**", que no son otra cosa más que funciones, para validar nuestros datos (por ejemplo, los meses sólo pueden ser: 1, 2, etc (para enero, febrero, etc), o el año no puede ser mayor al año actual, lo mismo que para el día (del 1 al 30, al 31, al 28 o al 29, dependiendo del mes o si es o no bisiesto el año). Idéntico caso con las horas (entre 00 y 12), los minutos y segundos (entre 00 y 59).

Nuestra definición de clase puede incluir métodos o funciones para validar los datos de entrada o para procesar nuestros datos. Por ejemplo: **valida\_hora** (un

método para validar que la hora esté entre 00 y 12), **valida\_minutos** (valor entre 00 y 59), etc. También podríamos incluir métodos para hacer cálculos a partir de los datos que posee nuestra instancia, por ejemplo, **dias\_faltantes**, nos devolverá el número de días que faltan, a partir de la fecha ingresada, para finalizar el año.

La **Programación Orientada a Objetos (POO)** hace modelos del mundo real mediante sus contrapartes en software. Aprovecha las relaciones de clase, donde objetos de una cierta clase, como la clase **Vehículos**, tienen las mismas características. Aprovecha las relaciones de herencia, e inclusive las relaciones de herencia múltiple, donde clases recién creadas de objetos se derivan heredando características de clases existentes, pero poseyendo características únicas, propias de ellos mismos. Los bebés tienen muchas características de sus padres, pero ocasionalmente padres de baja estatura tienen hijos altos.<sup>2</sup>

Tenemos una clase **Habitación** que define del modo siguiente sus datos públicos y sus métodos **calcula\_area** y **calcula\_volumen**:

```
Class Habitacion
{
    public:
        double largo;
        double ancho;
        double altura;

        double calcula_area()
        {
            return largo * ancho;
        }

        double calcula_volumen()
        {
            return largo * ancho * altura;
        }
};
```

**Sala** y **Comedor** son objetos de la clase **Habitación**, y se declaran dentro de **main()**, para referirnos a un dato puntual (ejemplo, longitud de la Sala o ancho del Comedor), debemos mencionar al objeto y al dato miembro que nos interesa:

```
int main()
{
    Habitacion Sala, Comedor;

    Sala.largo = 3.40;
    Comedor.ancho = 4.00;
    ...
}
```

---

<sup>2</sup>Deitel P. & Deitel H, **How to program C. With an introduction to C++**

# Programas en C++ con Clases, Objetos y Métodos

## Clase Fecha

```
#include <iostream>

class Fecha
{
    private:
        int dia;
        int mes;
        int año;

    public:
        // Método Constructor: Inicializa datos miembro del objeto
        Fecha(int d, int m, int a) : dia(d), mes(m), año(a)
        {
        }

        // Métodos de la clase Fecha

        // Método para verificar si el año es bisiesto
        bool esBisiesto()
        {
            return (año % 4 == 0 && año % 100 != 0) || (año % 400 == 0);
        }

        // Validación principal
        bool esValida()
        {
            // Método para validar el año (no mayor a 2025)
            if (año > 2025 || año < 1)
            {
                std::cout << "Error: El año debe ser entre 1 y 2025." << std::endl;
                return false;
            }

            // Método para validar el mes (entre 1 y 12)
            if (mes < 1 || mes > 12)
            {
                std::cout << "Error: El mes debe estar entre 1 y 12." << std::endl;
                return false;
            }

            // Método para validar el día según el mes
            // y si es o no bisiesto
            int diasMaximos;

            switch (mes)
            {
                case 1:
                case 3:
                case 5:
                case 7:
                case 8:
                case 10:
                case 12:
                    diasMaximos = 31;
                    break;
                case 4:
                case 6:
                case 9:
                case 11:
                    diasMaximos = 30;
                    break;
            }
        }
    };
};
```

```

        case 2:
            diasMaximos = esBisiesto() ? 29 : 28;
            break;
        default:
            return false;
    }

    if (dia < 1 || dia > diasMaximos)
    {
        std::cout << std::endl << "ERROR: El día " << dia
            << " no es valido para el mes " << mes
            << "." << std::endl;
        return false;
    }
    return true;
}

// Método para mostrar la fecha
void mostrarFecha()
{
    std::cout << dia << "/" << mes << "/" << año;
}

};

// Función Principal

int main()
{
    int d, m, a;

    std::cout << "Ingrese fecha de nacimiento" << std::endl;
    std::cout << "    Ingrese día: ";
    std::cin >> d;
    std::cout << "    Ingrese mes: ";
    std::cin >> m;
    std::cout << "    Ingrese año: ";
    std::cin >> a;

    Fecha miFecha(d, m, a);

    if (miFecha.esValida())
    {
        std::cout << std::endl << "La fecha ";
        miFecha.mostrarFecha();
        std::cout << "es VALIDA." << std::endl << std::endl;
    }
    else
    {
        std::cout << std::endl
            << "La fecha ingresada NO ES VALIDA."
            << std::endl << std::endl;
    }

    return 0;
}

```

## Clase Hora

```
#include <iostream>

class Hora
{
public:
    int hora;
    int minutos;
    int segundos;

public:
    Hora(int hh, int mm, int ss) : hora(hh), minutos(mm), segundos(ss)
    {
    }

    bool esHora()
    {
        return (hora >= 0 && hora <= 23);
    }

    bool esMinuto()
    {
        return (minutos >= 0 && minutos <= 59);
    }

    bool esSegundo()
    {
        return (segundos >= 0 && segundos <= 59);
    }

    // Muestra hora militar

    void horaMilitar()
    {
        std::cout << "Formato Militar: ";

        std::cout << hora << ":" << minutos << ":"
            << segundos << std::endl;
    }

    // Muestra hora estandar

    void horaEstandar()
    {
        std::cout << "Formato Estándar: ";

        if (hora < 12)
            std::cout << hora << ":" << minutos << ":"
                << segundos << " AM" << std::endl;
        else
            if (hora == 12 && minutos == 0 & segundos == 0)
                std::cout << hora << ":" << minutos << ":"
                    << segundos << " M" << std::endl;
            else
                if (hora > 12)
                    std::cout << hora - 12 << ":" << minutos << ":"
                        << segundos << " PM" << std::endl;
                else
                    std::cout << hora << ":" << minutos << ":"
                        << segundos << " PM" << std::endl;
    }
};
```

```
// Función Principal

int main()
{
    int hh, mm, ss;

    std::cout << "Ingrese hora en formato de 24 horas: " << std::endl;
    std::cout << "Hora: ";
    std::cin >> hh;
    std::cout << "Minutos: ";
    std::cin >> mm;
    std::cout << "Segundos: ";
    std::cin >> ss;

    std::cout << std::endl;

    Hora miHora(hh,mm,ss);

    if (miHora.esHora() && miHora.esMinuto() && miHora.esSegundo())
    {
        miHora.horaEstandar();
        miHora.horaMilitar();
    }

    std::cout << std::endl;
}
```



Esta obra está bajo una Licencia [Creative Commons Atribución-NoComercial-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/).