

[Descargar esta guía ecomo documento pdf](#)

## Índice

1. [Índice](#)
2. [Presentación del módulo: objetivos y resultados del aprendizaje según la normativa](#)
  1. [Temario resumido](#)
3. [Metodología](#)
  1. [Tutoría Colectiva](#)
  2. [Tutoría Individualizada](#)
4. [Contenidos y temporalización](#)
5. [Contenidos mínimos y criterios de evaluación](#)
  1. [Incluidos en los Temas 1](#)
  2. [Incluidos en los Temas 4, 5 y 6](#)
  3. [Incluidos en los Temas 3 y 7](#)
  4. [Incluidos en los Temas 5 y 7](#)
  5. [Incluidos en los Temas 2, 8 y 9](#)
  6. [Incluidos en los Temas 5, 6, 7 y 10](#)
  7. [Incluidos en el Tema 7 y 9](#)
6. [Procedimientos e instrumentos de evaluación](#)
  1. [Instrumentos de evaluación](#)
  2. [Criterios de calificación](#)
7. [Enlaces de Internet y bibliografía](#)

## Presentación del módulo: objetivos y resultados del aprendizaje según la normativa

La formación de este módulo en concreto contribuye a alcanzar algunos de los objetivos generales de este ciclo formativo, que son los siguientes:

- Seleccionar y emplear lenguajes, herramientas y librerías, interpretando las especificaciones para desarrollar aplicaciones multiplataforma con acceso a bases de datos.
- Seleccionar y emplear técnicas, lenguajes y entornos de desarrollo, evaluando sus posibilidades, para desarrollar aplicaciones en teléfonos y otros dispositivos móviles.
- Seleccionar y emplear lenguajes y herramientas, atendiendo a los requerimientos, para desarrollar componentes personalizados en sistemas ERP-CRM.
- Identificar los cambios tecnológicos, organizativos, económicos y laborales en su actividad, analizando sus implicaciones en el ámbito de trabajo, para mantener el espíritu de innovación.

**Podemos resumir diciendo que el módulo nos ayudará a comprender los paradigmas básicos de programación empleados en el desarrollo de aplicaciones multiplataforma.**








### Temario resumido

**Nota:** Debido al retraso por el comienzo del curso el temario se ha reducido a los siguientes contenidos.

- Introducción a Microsoft .Net
- Introducción y aspectos léxicos de C#.
- C# desde el punto de vista de la Programación Imperativa y Estructurada.
  - Sentencias de control.
  - Funciones, procedimientos y recursividad.
  - Tipos de datos estructurados, Expresiones Regulares, Ficheros...
- C# y la Programación Orientada a Objetos.
  - Herencia, Polimorfismo, Encapsulación, Indizadores, Redefinición de Operadores.
  - Colecciones de datos.

## Metodología


El seguimiento del módulo se realizará a través de la plataforma educativa *Aules Semipresencial*, en la que, para cada uno de los temas o unidades didácticas, el alumnado tendrá a su disposición:

- Uno o varios enlaces con los **temas teóricos** de los contenidos **por quincena**.
  - Estos contenidos dispondrán de **enlaces** que llevarán a contenidos para ampliar la información de un determinado concepto o de donde se ha obtenido la misma.
  - En los temas habrá **ejemplos prácticos** que podrás probar y modificar para asimilar los conceptos y que encontrará bajo el epígrafe.  
| **Ejemplo:**
  - En los temas también habrá **casos de estudio** más elaborados donde se te propondrá pensar una solución y a continuación dispondrás de una propuesta de solución comentada para *'autoevaluar'* tus solución. Los encontrarás bajo el epígrafe.  
|  **Caso de estudio:**
  - También puedes encontrar otros símbolos que te pueden resultar interesantes y pretenden expresar los siguientes conceptos:
    -  Simbolizará algo en lo que debemos **pararnos** a reflexionar o es importante que le leamos detenidamente.
    -  **Ideas, 'Tips'** o resumen de puntos que nos pueden ser de utilidad a la hora de usar algo o saber cuando aplicar algo.
    -  Indicará una **mala práctica** o algo que no deberíamos hacer porque no es adecuado o **produce errores**.
    -  Nota, excepción o algo con lo que deberemos llevar cuidado.
    -  Resumen de lo comentado hasta el momento.
-  Actividades de **'autoevaluación'** interactivas, para comprobar nuestra comprensión de las definiciones, conceptos o contenidos teóricos de los temas.
- **Uno o varios ejercicios prácticos** que deberán ser entregados **al final del periodo correspondiente a cada quincena**. Además, de las propuestas de ejercicios de entrega obligatoria, habrá propuestas ejercicios de introducción/refuerzo y de ampliación.  
De todos estos ejercicios **se proporcionará al final de cada quincena una propuesta de solución**, para que los alumnos la puedan revisar y autoevaluarse en el caso de no ser de entrega obligatoria. Además, **los ejercicios no obligatorios** se podrán resolver en alguna de las **tutorías colectivas** que se describen a continuación a propuesta de los alumnos y siempre dentro de su quincena de realización.
- En su caso, **uno o varios enlaces a diferentes recursos** que puedan ser interesantes para complementar los contenidos tratados en el tema.

## Tutoría Colectiva

Cada semana habrá **cuatro horas** de tutoría colectiva en la que se podrán plantear y resolver dudas relacionadas con las actividades realizadas y aclarar y explicar aquellos contenidos teóricos que se considere necesario.

La **asistencia a la tutoría colectiva** tiene carácter **voluntario**.

- Por razones sanitarias, la tutoría en **horario de mañana** se realizará de forma **telemática** a través de la plataforma **WebEx** los **lunes, de 11:05 a 12:55** horas.  
| **Nota:** Encontrarás instrucciones para unirse a ambas plataformas en *Aules Semipresencial*.
- La tutoría en **horario de tarde**, tendrá carácter **presencial** y se realizará los **miércoles, de 15:30 a 17:20** horas.
- En el *Aules Semipresencial* dispondréis de un enlace al **servicio de reservas** de tutorías colectivas. El cual, en el cual os pedirá introducir el correo del centro en el momento de la reserva y os mandará un correo confirmación de la misma con una opción de cancelación.  
|  Por razones organizativas y dada la actual situación sanitaria, para asistir a la tutoría presencial, el alumnado deberá pedir cita **antes del jueves de la semana anterior**.

## Tutoría Individualizada

- Además, semanalmente habrá **dos horas de tutoría individual** en horario de mañana, en la que se pueden hacer consultas y/o plantear dudas sobre los temas tratados.
- Las tutorías individuales pueden ser por **correo electrónico** o preferiblemente vía el **foro de dudas** en plataforma *Aules Semipresencial*.
- El horario asignado de atención de estas tutorías individuales son el **martes de 11:05 a 12:55** y el **jueves de 9:50 a 12:00** por lo que posiblemente las respuestas se produzcan en ese momento.
- Además, en esa franja horaria el profesor del módulo intentará estar disponible en las plataformas **WebEx** antes mencionadas.
- En estas tutorías individuales se atenderán **preferentemente dudas correspondientes a la quincena actual**.

## Contenidos y temporalización

Los contenidos que se desarrollarán para la consecución de los objetivos antes expuestos son los que se presentan en la siguiente tabla, organizados por quincenas, siendo la siguiente planificación una previsión del curso 2020/2021 y pudiendo ser susceptible a modificaciones debido a causas de diversa índole: refuerzo educativo de temas o aspectos que se detecten complejos para el alumno, razones sanitarias, etc.

### Primera Evaluación del 10 septiembre al 26 noviembre

- **Quincena del 13 septiembre - 26 septiembre**
  - **Semana 1**
    - *Tema 1:* Introducción a .NET o dotnet
    - *Tema 2.1:* Primer programa, Variables y Entrada/Salida
  - **Semana 2** *Tema 2.2:* Operadores y expresiones
- **Quincena del 27 septiembre - 10 octubre**
  - **Semana 1** *Tema 3.1:* Estructuras de control I (Condicionales)
  - **Semana 2** *Tema 3.2:* Estructuras de control II (bucle while y do-while, contadores y acumuladores).
- **Quincena del 11 octubre - 24 octubre**
  - **Semana 1** *Tema 3.2:* Estructuras de control II (bucle para, rupturas, flags y bucles anidados).
  - **Semana 2** *Tema 4:* Programación estructurada - definiciones, conceptos, métodos en C#
- **Quincena del 25 octubre - 7 noviembre**
  - **Semana 1**
    - *Tema 4:* Divisiones simples de un programa módulos a través de métodos.
    - *Tema 4:* Repaso de tips para definir un módulo y caso práctico de modularización.
  - **Semana 2** *Tema 5.1:* Introducción in conceptos básicos de la POO.
- **Quincena del 8 enero - 21 noviembre**
  - **Semana 1** *Tema 5.1:* Manejo de cadenas (inmutables y mutables).
  - **Semana 2** *Tema 5.1:* Tablas homogéneas unidimensionales (Arrays).

📌 **Exámenes 1ª evaluación:** del 22 de noviembre al 28 noviembre.

### Segunda Evaluación

- **Quincena del 29 noviembre - 12 diciembre**
  - **Semana 1**
    - *Tema 5.2:* Tablas homogéneas multidimensionales (Matrices y Tablas dentadas).
    - *Tema 5.2:* Enumeraciones.
  - **Semana 2**
    - *Tema 6:* Expresiones regulares.
- **Quincena del 13 diciembre - 22 diciembre**
  - **Semana 1** *Tema 7.1:* Definiendo nuestras propias clases (Referencia y valor).
  - **Semana 2** *Tema 7.1:* Rol "todo parte" composición y agregación.
- **Quincena del 10 enero - 23 enero**
  - **Semana 1** *Tema 7.2:* Rol "es un" herencia (conceptos, diseño y abstracción).
  - **Semana 2** *Tema 7.3:* Gestión de errores en POO.
- **Quincena del 24 enero - 6 febrero**
  - **Semana 1** *Tema 8.1:* Flujos de entrada y salida de datos.
  - **Semana 2** *Tema 8.2:* Persistencia de objetos y serialización.
- **Quincena del 7 febrero - 20 febrero**
  - **Semana 1** *Tema 9.1:* Gestión y organización de paquetes y librerías.
  - **Semana 2** *Tema 9.2:* Propiedades, Redefinición de operadores e Indicadores.

📌 **Exámenes 2ª evaluación:** del 21 de febrero al 27 febrero.

### Tercera Evaluación

- **Quincena del 28 febrero - 1 marzo**
  - **Semana 1** *Tema 9.3:* Interfaces, Genéricos.
  - **Semana 2** *Tema 9.4:* Principios SOLID.
- **Quincena del 14 marzo - 27 marzo**
  - **Semana 1** *Tema 10.1:* Colecciones I.
  - **Semana 2** *Tema 10.2:* Colecciones II.
- **Quincena del 28 marzo - 13 abril**
  - **Semana 1**
    - *Tema 11.1:* Delegados.
    - *Tema 11.2:* Eventos.
  - **Semana 2**
    - *Tema 11.3:* Otras funcionalidades del lenguaje.
    - *Tema 11.4:* Introducción a la Programación Funcional.
- **Quincena del 26 abril - 8 mayo**
  - **Semana 1** *Tema 11.5:* Programación Funcional Avanzada.
  - **Semana 2** *Tema 11.6:* Recursividad.
- **Quincena del 9 mayo - 22 mayo**
  - Esta quincena se deja para posibles retrasos acumulados en la programación, repaso o trabajo final si la programación ha ido correctamente.

📌 **Exámenes 3ª evaluación:** del 21 de mayo al 29 mayo.

📌 **Exámenes Finales Ordinarios:** del 30 de mayo al 5 junio.

## Contenidos mínimos y criterios de evaluación

### Incluidos en los Temas 1

#### Contenidos mínimos

- Estructura y bloques fundamentales.
- Variables.
- Tipos de datos.
- Literales.
- Constantes.
- Operadores y expresiones.
- Conversiones de tipo.
- Comentarios.

#### Criterios de evaluación

- Se han identificado los bloques que componen la estructura de un programa informático.
- Se han creado proyectos de desarrollo de aplicaciones
- Se han utilizado entornos integrados de desarrollo.
- Se han identificado los distintos tipos de variables y la utilidad específica de cada uno.
- Se ha modificado el código de un programa para crear y utilizar variables.
- Se han creado y utilizado constantes y literales.
- Se han clasificado, reconocido y utilizado en expresiones los operadores del lenguaje.
- Se ha comprobado el funcionamiento de las conversiones de tipo explícitas e implícitas.
- Se han introducido comentarios en el código

## Incluidos en los Temas 4, 5 y 6

### Contenidos mínimos

- Características de los objetos.
- Instanciación de objetos.
- Utilización de métodos.
- Utilización de propiedades.
- Utilización de métodos estáticos.
- Constructores.
- Destrucción de objetos y liberación de memoria.

### Criterios de evaluación

- Se han identificado los fundamentos de la programación orientada a objetos.
- Se han escrito programas simples.
- Se han instanciado objetos a partir de clases predefinidas.
- Se han utilizado métodos y propiedades de los objetos.
- Se han escrito llamadas a métodos estáticos.
- Se han utilizado parámetros en la llamada a métodos.
- Se han incorporado y utilizado librerías de objetos.
- Se han utilizado constructores.
- Se ha utilizado el entorno integrado de desarrollo en la creación y compilación de programas simples.

## Incluidos en los Temas 3 y 7

### Contenidos mínimos

- Estructuras de selección.
- Estructuras de repetición.
- Estructuras de salto.
- Control de excepciones.

### Criterios de evaluación

- Se ha escrito y probado código que haga uso de estructuras de selección.
- Se han utilizado estructuras de repetición.
- Se han reconocido las posibilidades de las sentencias de salto.
- Se ha escrito código utilizando control de excepciones.
- Se han creado programas ejecutables utilizando diferentes estructuras de control.
- Se han probado y depurado los programas.
- Se ha comentado y documentado el código.

## Incluidos en los Temas 5 y 7

### Contenidos mínimos

- Concepto de clase.
- Estructura y miembros de una clase.
- Creación de atributos.
- Creación de métodos.
- Creación de constructores.
- Utilización de clases y objetos.
- Utilización de clases heredadas.

### Criterios de evaluación

- Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.
- Se han definido clases.
- Se han definido propiedades y métodos.
- Se han creado constructores.
- Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.
- Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.
- Se han definido y utilizado clases heredadas.
- Se han creado y utilizado métodos estáticos.
- Se han definido y utilizado interfaces.
- Se han creado y utilizado conjuntos y librerías de clases.



## Incluidos en los Temas 2, 8 y 9

### Contenidos mínimos

- Tipos de flujos. Flujos de bytes y de caracteres.
- Clases relativas a flujos.
- Utilización de flujos.
- Entrada desde teclado.
- Salida a pantalla.
- Ficheros de datos. Registros.
- Apertura y cierre de ficheros. Modos de acceso.
- Escritura y lectura de información en ficheros.
- Utilización de los sistemas de ficheros.
- Creación y eliminación de ficheros y directorios.
- Interfaces.
- Concepto de evento.
- Creación de controladores de eventos.

### Criterios de evaluación

- Se ha utilizado la consola para realizar operaciones de entrada y salida de información.
- Se han aplicado formatos en la visualización de la información.
- Se han reconocido las posibilidades de entrada / salida del lenguaje y las librerías asociadas.
- Se han utilizado ficheros para almacenar y recuperar información.
- Se han creado programas que utilicen diversos métodos de acceso al contenido de los ficheros.
- Se han utilizado las herramientas del entorno de desarrollo para crear interfaces gráficos de usuario simples.
- Se han programado controladores de eventos.
- Se han escrito programas que utilicen interfaces gráficos para la entrada y salida de información.

## Incluidos en los Temas 5, 6, 7 y 10

### Contenidos mínimos

- Estructuras.
- Creación de arrays.
- Arrays multidimensionales.
- Cadenas de caracteres.
- Listas.

### Criterios de evaluación

- Se han escrito programas que utilizan arrays.
- Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados.
- Se han utilizado listas para almacenar y procesar información.
- Se han utilizado iteradores para recorrer los elementos de las listas.
- Se han reconocido las características y ventajas de cada una de la colecciones de datos disponibles.
- Se han creado clases y métodos genéricos.
- Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.

## Incluidos en el Tema 7 y 9

### Contenidos mínimos

- Composición de clases.
- Herencia.
- Superclases y subclases.
- Clases y métodos abstractos y finales.
- Sobreescritura de métodos.
- Constructores y herencia.

### Criterios de evaluación

- Se han identificado los conceptos de herencia, superclase y subclase.
- Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.
- Se ha reconocido la incidencia de los constructores en la herencia.
- Se han creado clases heredadas que sobrescriben la implementación de métodos de la superclase.
- Se han diseñado y aplicado jerarquías de clases.
- Se han probado y depurado las jerarquías de clases.
- Se han realizado programas que implementen y utilicen jerarquías de clases.
- Se ha comentado y documentado el código.

## Procedimientos e instrumentos de evaluación

### Instrumentos de evaluación

Los instrumentos de evaluación que se utilizarán son los siguientes:

1. **Pruebas teórico/prácticas:** se realizará una prueba práctica en cada una de las evaluaciones, que se llevará a cabo de forma presencial.
2. **Actividades prácticas:** al final de cada uno de los bloques quincenales, el alumnado deberá realizar los test de autoevaluación de contenidos y presentar aquellas actividades prácticas marcadas como obligatorias de forma telemática. Todo esto se realizará siempre a través de la plataforma educativa *Aules Semipresencial* y **en ningún caso se aceptarán envíos fuera de plazo por otros medios telemáticos como pudiera ser el correo electrónico.**

### Criterios de calificación

1. **70%** supondrá la prueba teórico/prácticas la cual calificará con un valor numérico real de 0 a 10.
2. **30%** actividades prácticas de entrega obligatoria de 0 a 10.

Su nota final, correspondiente a la evaluación continua, vendrá dada por la media ponderada de cada trimestre según los siguientes porcentajes:

- **20% de la nota del primer trimestre.**
- **35% de la nota del segundo trimestre.**
- **45% de la nota del tercer trimestre.**

**Nota:** Esta ponderación es creciente debido a que los contenidos no son 'estancos' a lo largo del módulo sino más bien de carácter acumulativo.

### Prueba final ordinaria

Justo después de la de la prueba **teórico/práctica** del tercer trimestre. Aquellos alumnos que no hayan superado con una nota de 5 o superior la **evaluación continua** mediante la ponderación anteriormente indicada, dispondrán de una prueba única **teórico/práctica**, que **deberán realizar en su totalidad** y que se calificará con un valor entero entre 0 y 10 y cuyo resultado será la nota final de la convocatoria ordinaria del curso.

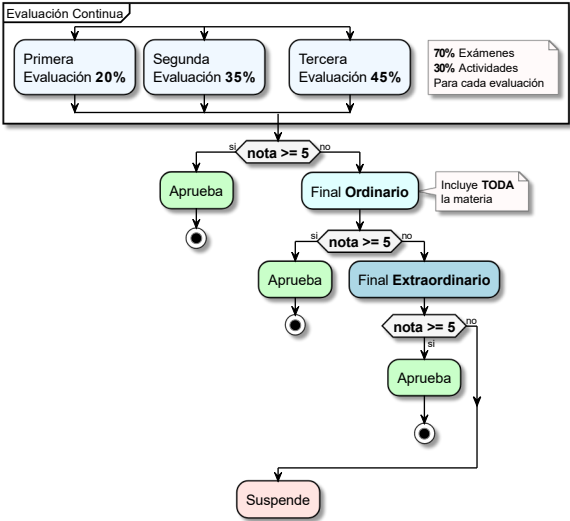
Prueba final extraordinaria

Quienes no superen el curso en la evaluación ordinaria, quedarán convocados a una prueba extraordinaria del 1 al 3 julio.

Control de copias

El profesor podrá decidir que, los alumnos implicados en copias de ejercicios, prácticas o exámenes sean calificados con un cero, contando así en las medias pertinentes. Si este proceder persiste, **el alumno perderá el derecho a evaluación continua** debiendo realizar la prueba final descrita para aquellos alumnos sin derecho a esta. Con este proceder se pretende evitar que los alumnos entreguen prácticas copiadas y que esto ocurra reiteradamente, ya que la nota de los ejercicios tiene un peso importante en la nota final del curso y debe ser representativa de los conocimientos del alumno.

Esquema Reasumen



## Enlaces de Internet y bibliografía

A través de la plataforma educativa *Aules Semipresencial*, se pondrá a disposición del alumnado los enlaces a los recursos que resulten interesantes para el seguimiento del módulo en cada momento. No obstante aquí hay una serie de enlaces interesantes:

- [Documentación oficial C#](#) Castellano.
- [Curso en YouTube C#](#) Castellano.
- [Libro gratuito creado por la comunidad de StackOverflow](#). Castellano. Todos los niveles.
- [Aprenda a programar como si estuviera en primero](#). Castellano. Nivel principiante.
- [Curso web de C#](#) Castellano.
- [Curso On-Line CodeAcademy](#) Inglés, precisa de registro y contenido adicional de pago.
- [hackr.io Buscador de cursos gratuitos](#). Inglés
- Code Complete 2nd Edition Steve MacConell. Inglés. Nivel alto.
- Código Limpio: Manual de estilo para el desarrollo ágil de software. Robert C. Martin. Castellano. Nivel alto.
- El limpiador de código: Código de conducta para programadores profesionales. Robert C. Martin. Castellano. Nivel alto.
- Refactoring: Improving the Design of Existing Code (2nd Edition). Martin Fowler. Inglés. Nivel alto.
- [Patrones de diseño](#). Erich Gamma. Castellano. Nivel alto.
- [Código Sólido](#). Gonzalo Ayuso. Castellano. Nivel alto.