

# IES Doctor Balmis

## Guía didáctica para el módulo de programación del curso 2022-23

### CFGS - Ciclo Formativo de Grado Superior DAW - Desarrollo de Aplicaciones Web

[Descargar esta guía como documento pdf](#)

## Índice

1. [Índice](#)
2. [Presentación](#)
3. [Características del módulo profesional](#)
4. [Tutorías](#)
5. [Metodología](#)
6. [Atención tutorial del grupo: información de contacto del profesor](#)
7. [Contenidos del módulo profesional y actividades a realizar](#)
  1. [Contenidos](#)
  2. [Actividades a realizar](#)
8. [Contenidos mínimos y criterios de evaluación](#)
  1. [Contenidos mínimos](#)
  2. [Criterios de calificación](#)
9. [Calendario](#)
10. [Instrumentos de evaluación](#)
11. [Solucionario de las actividades autoevaluables](#)
12. [Enlaces de internet y bibliografía recomendada](#)

# Presentación

La competencia general del ciclo se cita a continuación:

*desarrollar, implantar, y mantener aplicaciones web, con independencia del modelo empleado y utilizando tecnologías específicas, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de accesibilidad, usabilidad y calidad exigidas en los estándares establecidos*

La formación de este módulo en concreto contribuye a obtener los siguientes resultados de aprendizaje:

1. Evalúa sistemas informáticos identificando sus componentes y características.
2. Instala sistemas operativos planificando el proceso e interpretando documentación técnica.
3. Gestiona la información del sistema, identificando las estructuras de almacenamiento y aplicando medidas para asegurar la integridad de los datos.
4. Gestiona sistemas operativos utilizando comandos y herramientas gráficas y evaluando las necesidades del sistema.
5. Interconecta sistemas en red configurando dispositivos y protocolos.
6. Opera sistemas en red gestionando sus recursos e identificando las restricciones de seguridad existentes.
7. Elabora documentación valorando y utilizando aplicaciones informáticas de propósito general.

# Características del módulo profesional

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

- Seleccionar lenguajes, objetos y herramientas, interpretando las especificaciones para desarrollar aplicaciones web con acceso a bases de datos.
- Utilizar lenguajes, objetos y herramientas, interpretando las especificaciones para desarrollar aplicaciones web con acceso a bases de datos.
- Emplear herramientas y lenguajes específicos, siguiendo las especificaciones, para desarrollar componentes multimedia.
- Programar y realizar actividades para gestionar el mantenimiento de los recursos informáticos.

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

Las líneas de actuación en el proceso enseñanza-aprendizaje que permiten alcanzar los objetivos del módulo versarán sobre:

- La interpretación y aplicación de los principios de la programación orientada a objetos.
- La evaluación, selección y utilización de herramientas y lenguajes de programación orientados a objetos
- La utilización de las características específicas de lenguajes y entornos de programación en el desarrollo de aplicaciones informáticas.
- La identificación de las funcionalidades aportadas por los sistemas gestores de bases de datos y su incorporación a los programas desarrollados.
- La documentación de los programas desarrollados.

# Tutorías

Al ser un modelo de enseñanza *presencial* todas las preguntas y dudas sobre la materia que puedan surgir ser resolverán en clase.

# Metodología

El seguimiento del módulo se realizará a través de la plataforma educativa [Aules FP](#), 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 encontrarás 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.
  - 💡 **Idea/s**, '**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 ampliación auto-evaluables. 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, **la solución de los ejercicios no obligatorios** se podrán solicitar y 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.

# Atención tutorial del grupo: información de contacto del profesor

La atención tutorial se realizará siempre en clase y directamente con el profesor si se trata de algún asunto privado a tratar fuera de los contenidos del módulo. Para casos de *asuntos privados* de alumnos que no puedan asistir a clase se usará el correo electrónico.

- El email de contacto del profesor es: [s.linaresriestra@edu.gva.es](mailto:s.linaresriestra@edu.gva.es)


## Contenidos del módulo profesional y actividades a realizar

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 actual 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.

### Contenidos


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

### Actividades a realizar

1. En cada quincena al alumno/a se le propondrá la realización uno o varios bloques de actividades prácticas o ejercicios a realizar tanto
  - **Evaluables:** Aquellos ejercicios marcados con un 'check' verde  y **deberán entregarse de forma obligatoria** por la plataforma Aules para su comprobación y

calificación.

- **Autoevaluables:** Los ejercicios sin el '*check*', se podrán resolver en la tutorías durante la quincena correspondiente y se proporcionará su solución al abrir el bloque.

 **Importante:** Todas las entregas 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.**

2. Adicionalmente, algunas unidades más '*teóricas*' contendrán **test de auto-evaluación de conceptos y definiciones**. Estos test nos ayudarán a saber si hemos entendido las definiciones y conceptos teorico-abstractos tratados en la unidad correspondiente y no serán tenidos en cuenta en la calificación del módulo.

# Contenidos mínimos y criterios de evaluación

## Contenidos mínimos

### 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.

# Criterios de calificación

1. **80%** supondrá la prueba teórico/prácticas la cual calificará con un valor numérico real de 0 a 10.
2. **20%** actividades prácticas de entrega obligatoria hasta el momento de la evaluación con una calificación de 0 a 10.

Cada ejercicio obligatorio (✓) dentro de un bloque de ejercicios se calificará como '**Apta**' o '**No Apta**' y la nota del bloque vendrá definida por el número de ejercicios '**Aptos**' sobre el total de ejercicios obligatorios del bloque.

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 en el mes de julio.

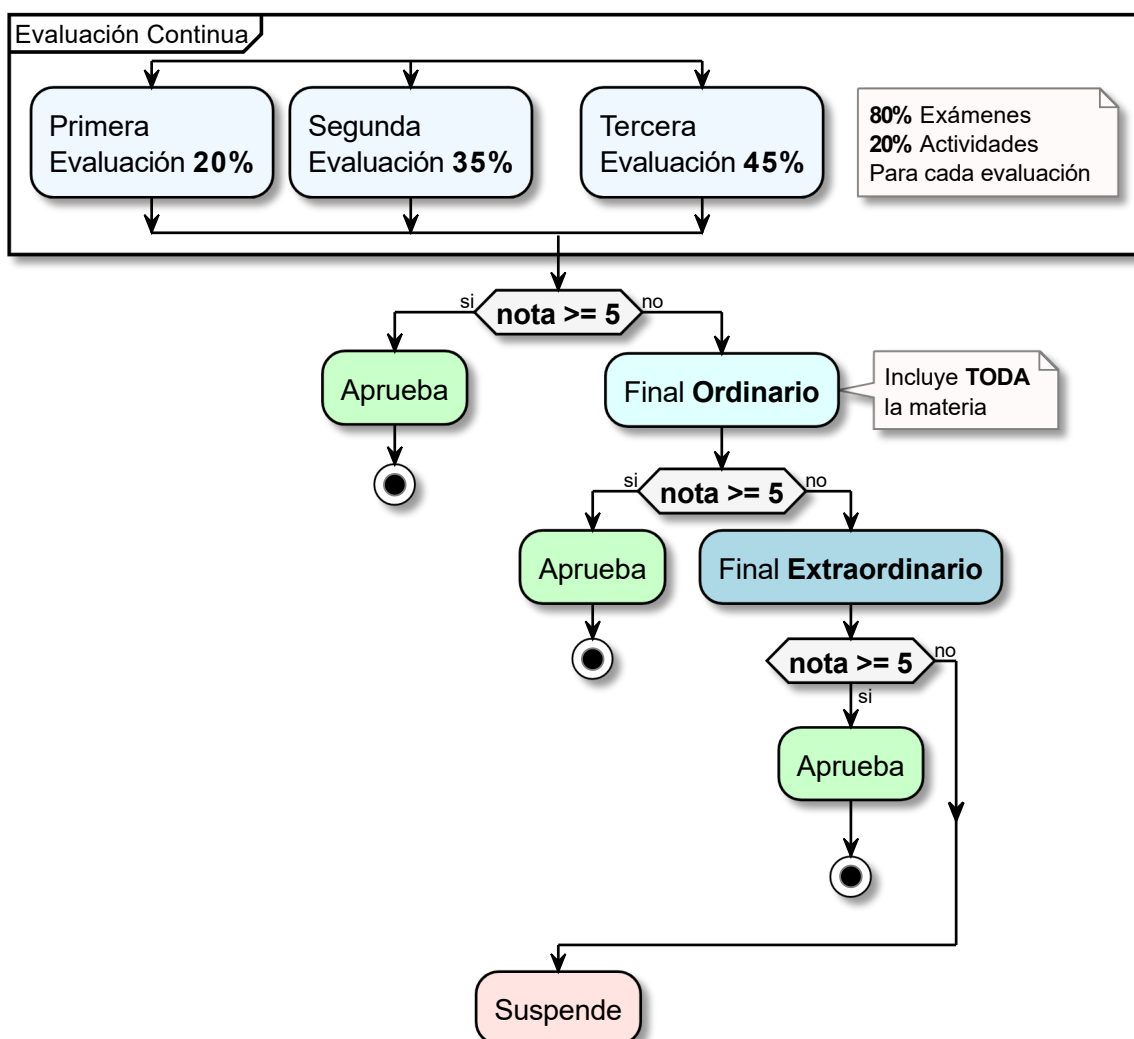
## Control de copias

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





# Calendario

## Primera Evaluación

- **Quincena del 12 septiembre - 25 septiembre**
  - **Semana 1**
    - *Tema 1:* Introducción a .NET
    - *Tema 2.1:* Primer programa, Variables y Entrada/Salida
  - **Semana 2** *Tema 2.2:* Operadores y expresiones
- **Quincena del 26 septiembre - 9 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 10 octubre - 23 octubre**
  - **Semana 1** *Tema 3.2:* Estructuras de control II (bucle para, rupturas, flags, bucles anidados, calcular máximos y mínimos).
  - **Semana 2** *Tema 4:* Programación estructurada - definiciones, conceptos, métodos en C#
- **Quincena del 24 octubre - 6 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.
    - *Tema 5.1:* Manejo de cadenas (inmutables y mutables).
- **Quincena del 7 noviembre - 20 noviembre**
  - **Semana 1** *Tema 5.1:* Tablas homogéneas unidimensionales (Arrays).
  - **Semana 2** Semana de repaso antes del examen o compensación de posibles retrasos.

 **Exámenes 1ª evaluación:** del 21 de noviembre al 27 noviembre.

## Segunda Evaluación

- **Quincena del 28 noviembre - 11 diciembre**
  - **Semana 1** Tema 5.2: Tablas homogéneas multidimensionales (Matrices y Tablas dentadas).
  - **Semana 2** Tema 5.2: Enumeraciones.
- **Quincena del 12 diciembre - 22 diciembre**
  - **Semana 1** Tema 6: Expresiones regulares.
  - **Semana 2** Tema 7.1: Definiendo nuestras propias clases (Referencia y valor).
- **Quincena del 9 enero - 22 enero**
  - **Semana 2** Tema 7.1: Rol "todo parte" composición y agregación.
  - **Semana 1** Tema 7.2: Rol "es un" herencia (conceptos, diseño y conceptos de polimorfismo abstracción).
- **Quincena del 23 enero - 5 febrero**
  - **Semana 1**
    - Tema 7.3: Gestión de errores en POO.
    - Tema 8.1: Gestión del sistema de ficheros y rutas.
  - **Semana 1** Tema 8.1: Flujos de entrada y salida de datos.
- **Quincena del 6 febrero - 19 febrero**
  - **Semana 1** Tema 9.1: Gestión y organización de paquetes y librerías.
  - **Semana 2** Semana de repaso antes del examen o compensación de posibles retrasos.

 **Exámenes 2ª evaluación:** del 20 de febrero al 26 febrero.

## Tercera Evaluación

- **Quincena del 27 febrero - 12 marzo**

- **Semana 1**

- *Tema 9.1:* Polimorfismo Funcional y Propiedades
    - *Tema 9.2:* Redefinición de operadores
    - *Tema 9.3:* Indizadores (opcional).

- **Semana 2** *Tema 9.4:* Interfaces.

- **Quincena del 13 marzo - 26 marzo**

- **Semana 1** *Tema 9.5:* Clases parametrizadas (Genéricos).

- **Semana 2** *Tema 9.6:* Patrones de diseño y principios SOLID.

- **Quincena del 27 marzo - 5 abril**

- **Semana 2** *Tema 10.1:* Colecciones I.

- **Semana 1** *Tema 10.2:* Colecciones II.

- **Quincena del 17 abril - 30 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 1 mayo - 14 mayo**

- **Semana 1**

- *Tema 11.5:* Programación Funcional Avanzada.
    - *Tema 11.6:* Recursividad (opcional).

- **Semana 2** Semana de repaso antes del examen o compensación de posibles retrasos.

- **Del 15 mayo - 21 mayo**

- Repaso o trabajo final si la programación ha ido correctamente.

 **Exámenes 3ª evaluación:** del 22 de mayo al 28 mayo.

- **Del 5 junio - 11 junio**

- Resolución dudas de cara al examen final ordinario.

 **Exámenes Finales Ordinarios:** del 5 de junio al 11 junio.

 **Exámenes Finales Extraordinarios:** el 27 y 28 de junio.

# 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:** en cada quincena al alumno/a se le propondrá la realización uno o varios bloques de actividades prácticas o ejercicios a realizar tanto autoevaluables como evaluables descritas anteriormente.
3. **Seguimiento** de actividades, asistencia a tutorías colectivas, participación en los foros y observación diaria. Se tendrá en cuenta el trabajo diario que realice el alumno y su actitud, lo que engloba la participación en tutorías colectivas, la realización de las diferentes actividades propuestas y la aplicación de las distintas directrices definidas para cada una de las unidades didácticas.

## Solucionario de las actividades autoevaluables

Los apuntes entregados disponen de test de autoevaluación y prácticas cuyas soluciones se proporcionan en AULES.

# Enlaces de internet y bibliografía recomendada

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 (Canal de Píldoras Informáticas).
- [Curso en YouTube C#](#) Castellano (Canal de Nacho Cabanes) + [libro gratuito del autor en OpenLibra](#).
- [Curso Web de C#](#) Castellano.
- [Libro de C# gratuito en OpenLibra](#) Inglés.
- [hackr.io](#) [Buscador de cursos gratuitos](#). Inglés
- [Libros y cursos de C#](#). Principalmente en inglés y de pago.
- [Curso On-Line CodeAcademy](#) Inglés, precisa de registro y contenido adicional de pago.
- [Aprenda a programar como si estuviera en primero](#). Castellano. Nivel principiante.

Otros recursos para profundizar una vez finalizado el curso:

- 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.