

# IES Doctor Balmis

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

### CFGSS - Ciclo Formativo de Grado Superior Presencial DAM - Desarrollo de Aplicaciones Multiplataforma

[Descargar esta guía como documento pdf](#)

## Índice

1. [Índice](#)
2. [Presentación](#)
3. [Características del módulo profesional](#)
4. [Metodología](#)
5. [Contenidos del módulo profesional y actividades a realizar](#)
  1. [Contenidos](#)
  2. [Actividades a realizar](#)
6. [Contenidos mínimos y criterios de evaluación](#)
  1. [Contenidos mínimos](#)
  2. [Criterios de calificación](#)
7. [Enlaces de internet y bibliografía recomendada](#)

# Presentación

La competencia general del ciclo se cita a continuación:

*Desarrollar, implantar, documentar y mantener aplicaciones informáticas multiplataforma, utilizando tecnologías y entornos de desarrollo específicos, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de "usabilidad" calidad exigidas en los estándares establecidos.*

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

1. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.
2. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.
3. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.
4. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.
5. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.
6. Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.

# 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 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.**

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.

# Metodología

A lo largo del todo el curso se seguirá una metodología compartida teórico/práctica. Al inicio de cada tema, el profesor explicará, por el método tradicional de exposición, los conceptos básicos de la unidad. Tan pronto como sea posible, la metodología pasará a ser fundamentalmente procedimental, con la resolución de problemas sencillos y la realización de prácticas, invitando al alumno, bien individualmente o bien en grupo, a que exponga su planteamiento ante sus compañeros, para efectuar los pertinentes comentarios, intercambio de pareceres y discusión de las soluciones propuestas.

En la medida de lo posible se tendrá una atención individualizada por parte de los profesores a cada alumno o grupo de alumnos. Se fomentará que cada uno plantee sus dudas o problemas sobre el ejercicio. Finalmente se comentarán las posibles soluciones, así como los fallos y errores que suelen cometerse de forma más habitual.

El seguimiento del módulo se realizará a través de la plataforma educativa [Aules Presencial](#), 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.
  - 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.
- **Uno o varios ejercicios prácticos** que deberán ser entregados **dentro del plazo que se establezca para cada ejercicio**. 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 una propuesta de solución después del plazo de entrega**, para que los alumnos la puedan revisar y autoevaluar en el caso de no ser de entrega obligatoria.

# 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, siendo la siguiente planificación una previsión del curso 2022/2023 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. A lo largo el curso 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 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, en algunas unidades o grupo de unidades, se propondrán controles cortos que constarán de un ejercicio que deberá ser resuelto de forma autónoma por el alumno, durante una sesión o sesión y media de clase. Estos controles formarán parte de la nota de

los ejercicios de ese trimestre. Con ello se pretende un seguimiento más efectivo de los conocimientos que va adquiriendo el alumno.

# 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

El sistema de evaluación a emplear será el de evaluación continua e individualizada, dentro de un sistema didáctico presencial. Dado que el Módulo de Programación tiene un carácter Teórico/Práctico, los alumnos irán resolviendo ejercicios diarios, realizando trabajos y controles periódicos. Sobre esta base los profesores tendrán una nota global del trabajo diario del alumno, que servirá como Nota de Evaluación.

## Asistencia y trabajo diario

La asistencia a clase y el trabajo diario, es requisito fundamental para llevar a cabo el proceso de evaluación continua, por lo que, aplicando la legislación vigente a tal efecto, la no asistencia al 85% o más de las horas asignadas al módulo o la no entrega del 85% o más de las actividades propuestas por el profesor, conllevará la pérdida del derecho a la evaluación continua.

## Criterios de calificación por trimestre

Será el resultado de la ponderación de los bloques del módulo tratados en dicho periodo según los siguientes porcentajes:

**20% La evaluación de ejercicios y trabajo realizado en clase.** Esta calificación vendrá dada por un valor numérico real de 0 a 10, el cual será el resultado de la media obtenida en los diferentes ejercicios propuestos en clase, así como cualquier trabajo de ampliación. Cada ejercicio será evaluado con la calificación APTO o NO APTO.

A la hora de calificar los ejercicios se tendrá en cuenta...

- Su corrección y adecuación a lo solicitado.
- La entrega en plazo de los mismos.
- 3/4 Presentación de la asignatura Xusa García y Juanjo Guarinos
- La puntualidad en la asistencia.
- La actitud positiva y exigente respecto a las actividades.
- El respeto y ayuda a sus compañeros y disposición al trabajo en grupo.
- La participación en la clase y en las actividades que en ella se realicen.

**80% supondrán las pruebas teórico/prácticas realizadas a lo largo del trimestre, ya sean resolución de problemas, preguntas cortas o tipo test las cuales calificarán con un valor numérico real de 0 a 10.**

## Criterios de calificación final global ordinaria

Si al final del último trimestre el alumno ha asistido con regularidad y ha realizado las prácticas cotidianamente, conservará el derecho a evaluación continua. Su nota final 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 por no tener derecho a evaluación continua o por no haber podido superar positivamente esta, 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. Esta prueba se realizará con posterioridad a la última realizada en la evaluación continua.

## Prueba final extraordinaria

Quienes no superen el curso en la evaluación ordinaria, quedarán convocados a una prueba extraordinaria en el mes de julio. La junta evaluadora del grupo al que pertenezca cada alumno, tiene potestad para determinar, una vez oído al alumno, si se le mantiene o no dicha convocatoria.

## Control de copias

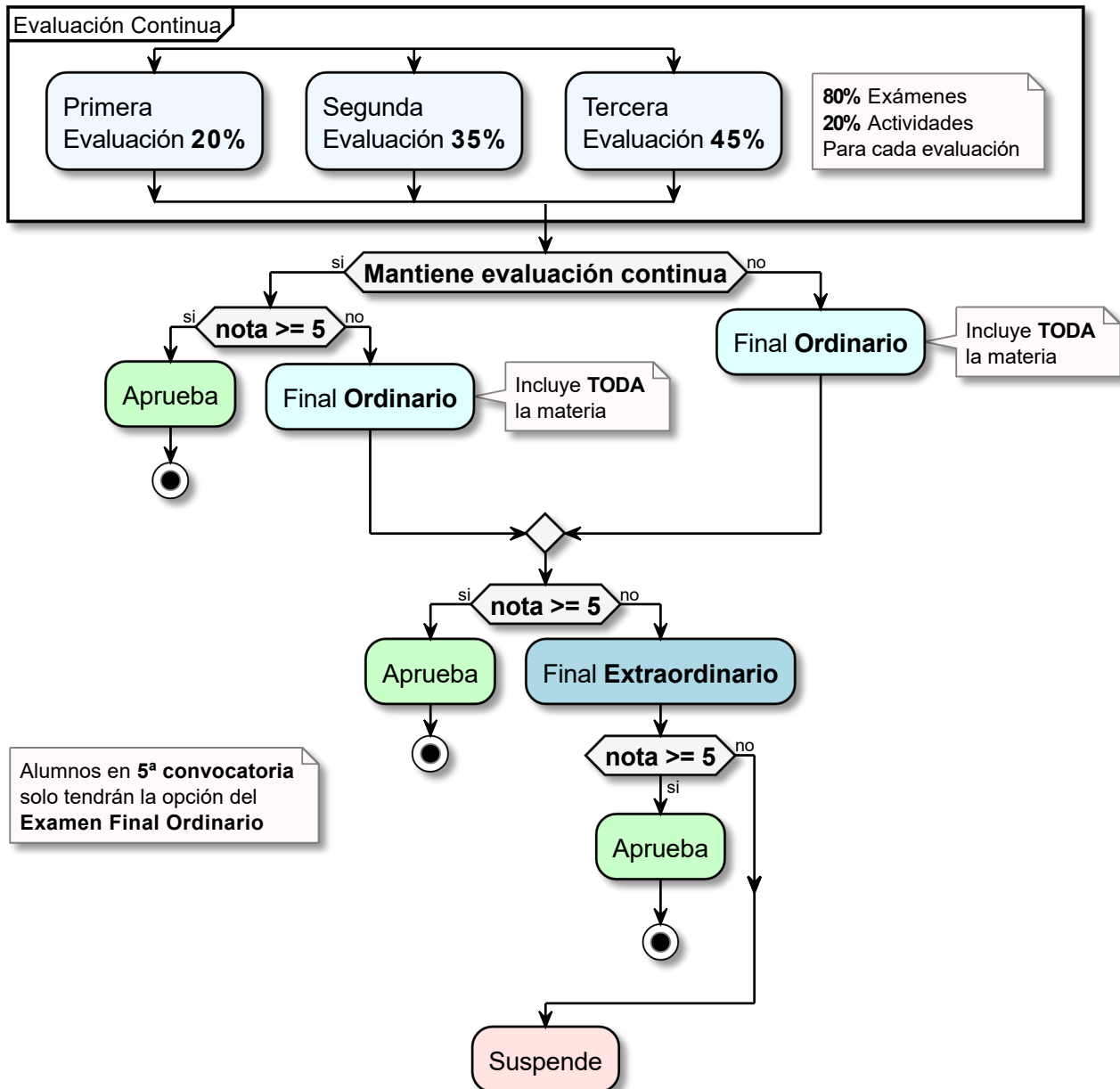
El profesor podrá decidir que, los alumnos implicados en copias de ejercicios, prácticas o exámenes sean calificados con un cero, contando así en las medias pertinentes.

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



## Ajuste evaluación por posible protocolo sanitario

En caso de no poderse impartir todo el contenido de las 3 evaluaciones. Se reasignará proporcionalmente el peso del contenido no impartido.

# Enlaces de internet y bibliografía recomendada

A través de la plataforma educativa *Aules Presencial*, 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.