

# IES Doctor Balmis

## Guía didáctica para el módulo de programación del curso 2024-25

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

[Descargar esta guía como documento pdf](#)

## Índice

- [Índice](#)
- [Presentación](#)
- [Características del módulo profesional](#)
- [Metodología](#)
- ▼ [Contenidos del módulo profesional y actividades a realizar](#)
  - [Contenidos](#)
  - [Actividades a realizar](#)
  - [Actividades a realizar](#)
- ▼ [Contenidos mínimos y criterios de evaluación](#)
  - ▼ [Contenidos mínimos](#)
    - [Incluidos en los Temas 1, 2, 3, 4, 5, 6, 7, 8, 10 y 10](#)
    - [Incluidos en los Temas 4, 5, 7, 9 y 11](#)
    - [Incluidos en los Temas 3, 5, 7, 8 y 10](#)
    - [Incluidos en los Temas 7, 9, 10 y 11](#)
    - [Incluidos en los Temas 2, 8 y 11](#)
    - [Incluidos en los Temas 5, 6, 9, 10, 11](#)
    - [Incluidos en el Tema 7 ,9 , 10 y 11](#)
    - [Incluidos en el Tema 12](#)
    - [Incluidos en el Tema 13](#)
  - ▼ [Criterios de calificación](#)
    - [Asistencia y trabajo diario](#)
    - [Prueba final ordinaria](#)

- Prueba final extraordinaria
- Control de copias
- Esquema Reasumen
- Ajuste evaluación por posible protocolo sanitario
- 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**:

- **RA1.** Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.
- **RA2.** Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.
- **RA3.** Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.
- **RA4.** Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.
- **RA5.** Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.
- **RA6.** Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.
- **RA7.** Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.
- **RA8.** Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.
- **RA9.** Gestiona información almacenada en bases de datos manteniendo la integridad y consistencia de los datos.

# 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.

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.




Todos los materiales necesarios se podrán obtener de forma directa en el aula, aunque también estarán expuestos en [la plataforma educativa](#) 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, que serán visibles según temporalización.
  - 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 cuando el profesor lo disponga. Además, de las propuestas de ejercicios de entrega obligatoria, habrá propuestas ejercicios de refuerzo de la que se proporcionará solución.
- **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 entrega una propuesta de solución**, para que los alumnos la puedan revisar y autoevaluarse 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, organizados por quincenas, siendo la siguiente planificación una previsión del **curso 2024/2025** 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

- **Tema 1:**
  - Introducción a Microsoft .Net
- **Tema 2:**
  - Primer programa, Variables y Entrada/Salida
  - Operadores y expresiones
- **Tema 3:**
  - Estructuras de control condicionales.
  - Estructuras de control bucles.
  - Algoritmia básica, contadores, acumuladores y flags.
- **Tema 4:**
  - Programación estructurada definiciones y conceptos.
  - Métodos: funciones y procedimientos.
  - Parametrización de métodos.
- **Tema 5:**
  - Introducción a la programación orientada a objetos.
  - Concepto Clases y diferencia entre métodos y campos de clase y de instancia.
  - Concepto de objeto e instanciación de objetos predefinidos en el lenguaje.
  - Mutabilidad e Inmutabilidad.
  - Cadenas de caracteres mutables e inmutables y sus operaciones.
  - Nulabilidad de objetos y operadores relacionados.
  - Colecciones de objetos de tamaño fijo: Arrays, Matrices y Tablas dentadas.
  - Enumeraciones excluyentes y no excluyentes (máscaras de bits).
- **Tema 6:**
  - Expresiones regulares.
  - Manejo de cadenas de caracteres avanzado.
- **Tema 7:**

- Definiendo de nuestras propias clases.
- Definiendo constructores.
- Concepto de destrucción de objetos.
- Encapsulación y definición de accesores y mutadores.
- Definiendo operaciones sobre objetos instanciados a través de métodos.
- Conceptos tipos valor y referencia y su relación con la instanciación de objetos.
- Concepto de value object y aproximación en C# a través de struct.
- Estructuras predefinidas en C# para el manejo de value objects de tiempo y fecha.
- Roles todo-parte entre clases. Composición y agregación.
- Aproximación desde C# a las relaciones todo-parte.
- Relación de herencia entre clases.
- Concepto de ocultación e invalidación de métodos.
- Polimorfismo de datos o inclusión.
- Downcasting y Upcasting (Principio de sustitución de Liskov).
- Ligadura o enlace dinámico de métodos.
- Clases y métodos abstractos.
- La clase Object en C#.
- Manejo de errores a través de excepciones.

- **Tema 8:**

- Gestión de ficheros y directorios.
- Entrada/Salida de datos.
- Tipos de flujos de datos y patrones asociados.

- **Tema 9:**

- Conceptos de paquete y artefacto.
- Concepto de espacio de nombres y su relación con los paquetes.
- Concepto de librería y su relación con los paquetes.
- Gestión de dependencias y librerías.
- Gestor de paquetes y repositorios.
- Polimorfismo funcional.
- Propiedades y su relación con la encapsulación y los métodos accesores y mutadores.
- Redefinición de operadores. (Ampliación Opcional)
- Abstracción a través de interfaces.
- Clases y métodos parametrizados o genéricos y su uso en C#
- Principios SOLID y algunos patrones de diseño. (Ampliación Opcional)

- **Tema 10:**

- Concepto de lista enlazada y listas enlazadas en C#.
- Listas, Pilas, Colas y Diccionarios en C#.
- Patrón iterador, su relación con las colecciones y uso.



- **Tema 11:**

- Concepto y uso de Delegados y Eventos.
- Concepto de tipos anónimos y su uso.
- Concepto de métodos de extensión y su uso.
- Generación perezosa de secuencias.
- El tipo record y su uso como DTO.
- Concepto de HOF, expresiones lambda y su uso.
- Operaciones agregadas sobre secuencias usando programación funcional.
- Concepto de Recursividad (Ampliación Opcional).
- Serialización y deserialización de objetos con JSON.

- **Tema 12:**

- Bases de datos orientadas a objetos. Estándares. Características.
- Definiendo Operaciones CRUD (Create, Read, Update, Delete).


- **Tema 13:**


- Acceso a bases de datos relacionales. Estándares. Características.
- Establecimiento de conexiones.
- Definiendo Operaciones CRUD (Create, Read, Update, Delete).

## Actividades a realizar

## 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 clase y se proporcionará su solución al abrir el bloque.

 **Importante:** Todas las entregas realizará siempre a través de la plataforma educativa *Aules Presencial* 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, 2, 3, 4, 5, 6, 7, 8, 10 y 10**

### Contenidos mínimos RA1

- 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, 7, 9 y 11

### Contenidos mínimos RA2

- Características de los objetos.
- Instanciación de objetos.
- Utilización de métodos. Parámetros.
- 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, 5, 7, 8 y 10

### Contenidos mínimos RA3

- Estructuras de selección.
- Estructuras de repetición.
- Estructuras de salto.
- Control de excepciones.
- Aserciones.
- Prueba, depuración y documentación de la aplicación.

### 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.
- Se han creado excepciones.
- Se han utilizado aserciones para la detección y corrección de errores durante la fase de desarrollo.

## Incluidos en los Temas 7, 9, 10 y 11

### Contenidos mínimos RA4

- Concepto de clase.
- Estructura y miembros de una clase. Visibilidad.
- 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 11

### Contenidos mínimos RA5

- Flujos. Tipos: bytes y caracteres. Clases relacionadas.
- 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.
- Entrada desde teclado. Salida a pantalla. Formatos de visualización.
- Interfaces gráficas.
- 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 gráficas de usuario simples.
- Se han programado controladores de eventos.
- Se han escrito programas que utilicen interfaces gráficas para la entrada y salida de información.

## Incluidos en los Temas 5, 6, 9, 10, 11

### Contenidos mínimos RA6

- Estructuras estáticas y dinámicas.
- Creación de matrices (arrays).
- Matrices (arrays) multidimensionales.
- Genericidad.
- Cadenas de caracteres. Expresiones regulares.
- Colecciones: Listas, Conjuntos y Diccionarios.
- Operaciones agregadas: filtrado, reducción y recolección.

### Criterios de evaluación

- Se han escrito programas que utilicen matrices (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 las 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.
- Se han identificado las clases relacionadas con el tratamiento de documentos escritos en diferentes lenguajes de intercambio de datos.
- Se han realizado programas que realicen manipulaciones sobre documentos escritos en diferentes lenguajes de intercambio de datos.
- Se han utilizado operaciones agregadas para el manejo de información almacenada en colecciones.

## Incluidos en el Tema 7 ,9 , 10 y 11

### Contenidos mínimos RA7

- Composición de clases.
- Herencia y polimorfismo.
- Jerarquía de clases: Superclases y subclases.
- Clases y métodos abstractos y finales.
- Interfaces.
- 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.
- Se han identificado y evaluado los escenarios de uso de interfaces.
- Se han identificado y evaluado los escenarios de utilización de la herencia y la composición.



## Incluidos en el Tema 12

### Contenidos mínimos RA8

- Bases de datos orientadas a objetos.
- Características de las bases de datos orientadas a objetos.
- Instalación del gestor de bases de datos.
- Creación de bases de datos.
- Mecanismos de consulta.
- El lenguaje de consultas: sintaxis, expresiones, operadores.
- Recuperación, modificación y borrado de información.
- Tipos de datos objeto; atributos y métodos.
- Tipos de datos colección.

### Criterios de evaluación

- Se han identificado las características de las bases de datos orientadas a objetos.
- Se ha analizado su aplicación en el desarrollo de aplicaciones mediante lenguajes orientados a objetos.
- Se han instalado sistemas gestores de bases de datos orientados a objetos.
- Se han clasificado y analizado los distintos métodos soportados por los sistemas gestores para la gestión de la información almacenada.
- Se han creado bases de datos y las estructuras necesarias para el almacenamiento de objetos.
- Se han programado aplicaciones que almacenen objetos en las bases de datos creadas.
- Se han realizado programas para recuperar, actualizar y eliminar objetos de las bases de datos.
- Se han realizado programas para almacenar y gestionar tipos de datos estructurados, compuestos y relacionados.

## Incluidos en el Tema 13

### Contenidos mínimos RA9

- Se han identificado las características y métodos de acceso a sistemas gestores de bases de datos.
- Se han programado conexiones con bases de datos.
- Se ha escrito un código para almacenar información en bases de datos.
- Se han creado programas para recuperar y mostrar información almacenada en bases de datos.
- Se han efectuado borrados y modificaciones sobre la información almacenada.
- Se han creado aplicaciones que muestren la información almacenada en bases de datos.
- Se han creado aplicaciones para gestionar la información presente en bases de datos.

### Criterios de evaluación

- Acceso a bases de datos. Estándares. Características.
- Establecimiento de conexiones.
- Almacenamiento, recuperación, actualización y eliminación de información en bases de datos.

# Crterios de calificacin

El sistema de evaluacin a emplear ser el de evaluacin continua e individualizada, dentro de un sistema didctico presencial. Dado que el Mdulo de Programacin tiene un carcter Teórico/Prctico, los alumnos irn resolviendo ejercicios diarios, realizando trabajos y controles peridicos. Sobre esta base los profesores tendrn una nota global del trabajo diario del alumno, que servir como Nota de Evaluacin.

## Asistencia y trabajo diario

La asistencia a clase y el trabajo diario, es requisito fundamental para llevar a cabo el proceso de evaluacin continua, por lo que, aplicando la legislacin vigente a tal efecto, la no asistencia al 85% o ms de las horas asignadas al mdulo o la no entrega del 85% o ms de las actividades propuestas por el profesor, conllevar la prdida del derecho a la evaluacin continua.

1. **70%** supondr la prueba terico/prcticas la cual calificar con un valor numrico real de 0 a 10.
2. **20%** ejercicios de control individuales que propondrn los profesores y que se realizarn durante las sesiones de clase de manera presencial. Se calificar con un valor real de 0 a 10.
3. **10%** cuestionarios y actividades prcticas de entrega obligatoria hasta el momento de la evaluacin con una calificacin de 0 a 10.

Su nota final, correspondiente a la evaluacin continua, vendr dada por la media ponderada de cada trimestre segn los siguientes porcentajes:

- **33% de la nota del primer trimestre.**
- **33% de la nota del segundo trimestre.**
- **34% de la nota del tercer trimestre.**



### Importante

Conforme a la nueva Ley de Formacin Profesional (LFP) los alumnos tendrn una nota para cada resultado de aprendizaje descritos al principio del documento. Esta nota se obtendr de la siguiente manera:

1. A cada tema de los contenidos se le asignar una ponderacin de aquellos RA que se traten y/o trabajen en l en la programacin didctica de mdulo.
2. Al final de la **evaluacin continua** se calcular la nota de cada RA como la suma del resultado de multiplicar la nota de cada evaluacin por la suma de los pesos asignados a

cada RA de los temas vistos en cada dicha evaluación. Para el calculo y según la temporalización los temas vistos en cada evaluación serán:

- 1ª Ev: temas 1, 2, 3, 4 y 5
- 2ª Ev: temas 5, 6, 7 y 8
- 3ª Ev: temas 8, 9, 10, 11, 12 y 13

3. Para las pruebas finales la notas de los RA coincidirá con la media ponderada de aquellos ejercicios prácticos del examen que se vean en los diferentes RA.

## Prueba final ordinaria

Justo después 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 a finales de junio o principios de julio.

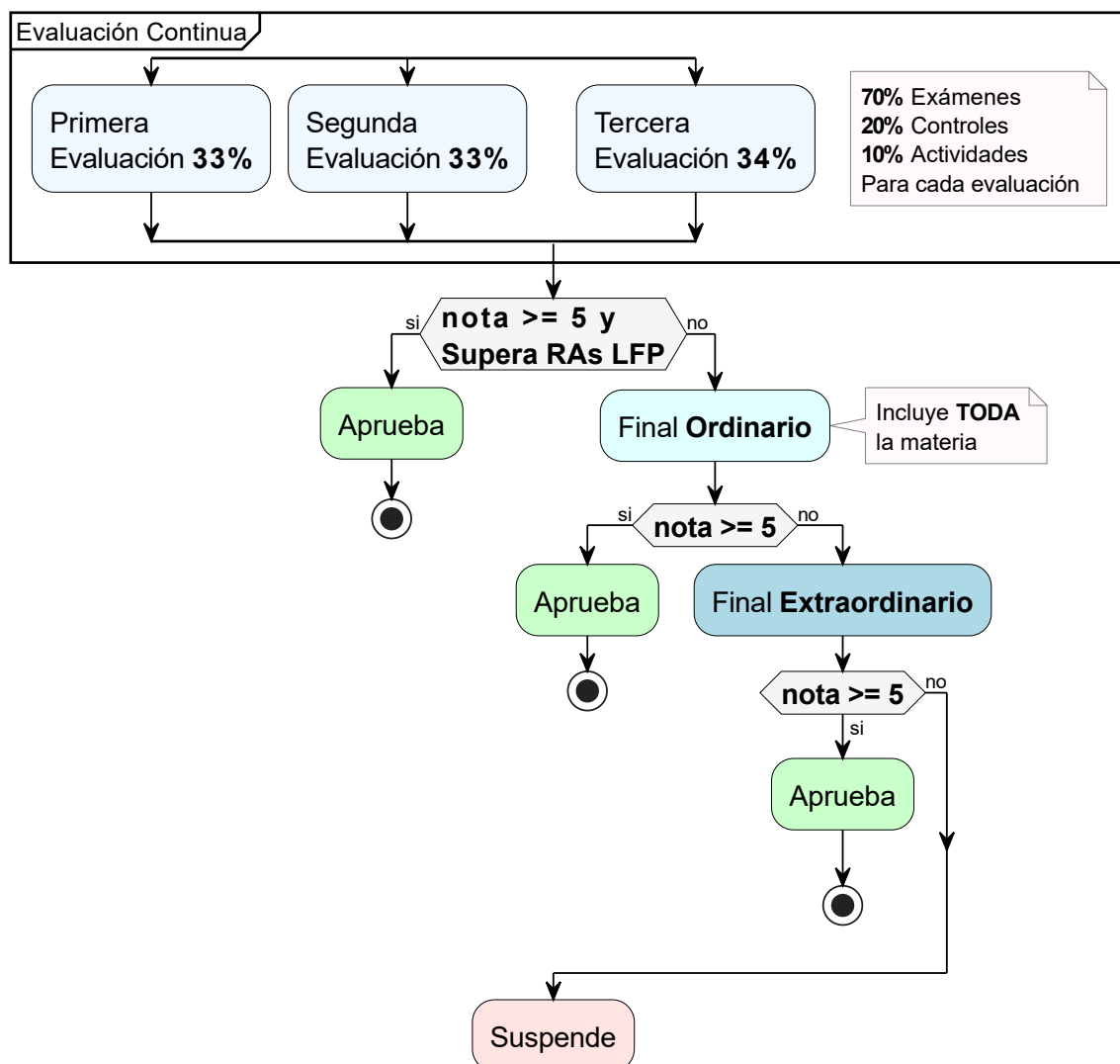
## Control de copias

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



## **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 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.