

PROGRAMMING II

2ºASIR

Departamento Informática

C r i t e r i o s d e E v a l u a c i ó n

Curso: 2024/2025

Miguel Manuel Juan Sanjuán

Contenido

Criterios de evaluación	2
Criterios de calificación	2
Criterios ortográficos.....	5
Criterios de recuperación	6

Criterios de evaluación

La consecución de los objetivos del módulo, mediante el estudio y práctica de sus contenidos, será determinada mediante la superación de los resultados de aprendizaje listados a continuación, en negrita, y desgranados en sus diferentes criterios de evaluación:

1. Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.

Criterios de evaluación:

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

2. Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.

Criterios de evaluación:

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

3. Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.

Criterios de evaluación:

- a) Se ha escrito y probado código que haga uso de estructuras de selección.
- b) Se han utilizado estructuras de repetición.
- c) Se han reconocido las posibilidades de las sentencias de salto.
- d) Se ha escrito código utilizando control de excepciones.
- e) Se han creado programas ejecutables utilizando diferentes estructuras de control.
- f) Se han probado y depurado los programas.
- g) Se ha comentado y documentado el código.
- h) Se han creado excepciones.
- i) Se han utilizado aserciones para la detección y corrección de errores durante la fase de desarrollo.

4. Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.

Criterios de evaluación:

- a) Se ha reconocido la sintaxis, estructura y componentes típicos de una clase.
- b) Se han definido clases.
- c) Se han definido propiedades y métodos.
- d) Se han creado constructores.
- e) Se han desarrollado programas que instancien y utilicen objetos de las clases creadas anteriormente.
- f) Se han utilizado mecanismos para controlar la visibilidad de las clases y de sus miembros.
- g) Se han definido y utilizado clases heredadas.
- h) Se han creado y utilizado métodos estáticos.
- i) Se han creado y utilizado conjuntos y librerías de clases.

5. Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.

Criterios de evaluación:

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

6. Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.

Criterios de evaluación:

- a) Se han escrito programas que utilicen matrices (arrays).
- b) Se han reconocido las librerías de clases relacionadas con tipos de datos avanzados.
- c) Se han utilizado listas para almacenar y procesar información.
- d) Se han utilizado iteradores para recorrer los elementos de las listas.
- e) Se han reconocido las características y ventajas de cada una de las colecciones de datos disponibles.
- f) Se han creado clases y métodos genéricos.
- g) Se han utilizado expresiones regulares en la búsqueda de patrones en cadenas de texto.
- h) Se han identificado las clases relacionadas con el tratamiento de documentos escritos en diferentes lenguajes de intercambio de datos.
- i) Se han realizado programas que realicen manipulaciones sobre documentos escritos en diferentes lenguajes de intercambio de datos.
- j) Se han utilizado operaciones agregadas para el manejo de información almacenada en colecciones.

7. Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.

Criterios de evaluación:

- a) Se han identificado los conceptos de herencia, superclase y subclase.
- b) Se han utilizado modificadores para bloquear y forzar la herencia de clases y métodos.
- c) Se ha reconocido la incidencia de los constructores en la herencia.
- d) Se han creado clases heredadas que sobrescriben la implementación de métodos de la superclase.
- e) Se han diseñado y aplicado jerarquías de clases.
- f) Se han probado y depurado las jerarquías de clases.
- g) Se han realizado programas que implementen y utilicen jerarquías de clases.
- h) Se ha comentado y documentado el código.
- i) Se han identificado y evaluado los escenarios de uso de interfaces.
- j) Se han identificado y evaluado los escenarios de utilización de la herencia y la composición.

8. Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.

Criterios de evaluación:

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

9. Gestiona información almacenada en bases de datos manteniendo la integridad y consistencia de los datos.

Criterios de evaluación:

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

Criterios de calificación

Se realizarán **ejercicios teórico-prácticos individuales en clase** a lo largo del curso, después de que el profesor explique cada unidad. Esta tarea, además del trabajo personal de repaso en casa de cada unidad, servirá para que el alumnado adquiera los conocimientos necesarios y alcance los objetivos del curso.

El profesorado supervisará y ayudará al alumnado en clase durante la realización de los citados ejercicios teórico-prácticos. Este proceso no obtendrá calificación alguna ya que persigue que el alumnado sea responsable de la formación que se le ofrece, y que persiga el aprendizaje más allá de una recompensa inmediata y cuantificable. Se le mentalizará con ese propósito, más próximo a la búsqueda de una buena carrera profesional dentro del mundo laboral que a la búsqueda de un impecable boletín de notas.

Asimismo, se realizará al alumnado **una prueba teórico-práctica escrita**, individual y con tiempo limitado. Será de carácter teórico-práctico y se realizará conforme determine el profesor en cada caso, pudiendo ser en papel en unos casos, o mediante ordenador en otros. Se realizará una prueba **antes de finalizar cada trimestre**, como norma general, pero podrá realizarse más de una por trimestre si el profesor lo considera necesario y siempre con la finalidad de favorecer el aprendizaje del alumnado.

El alumnado deberá demostrar mediante de estas pruebas teórico-prácticas, evaluadas por el profesor, **que ha asimilado al menos un 50% de conocimiento en el conjunto de los criterios de evaluación para cada resultado de aprendizaje**, total o parcialmente propuestos para cada curso académico, teniendo cada criterio de evaluación la misma valoración dentro del resultado de aprendizaje que describe.

Los resultados de aprendizaje serán trabajados en dos años académicos, según la siguiente tabla:

1 ^{er} CURSO	RA1 (100%), RA2 (100%), RA3 (100%), RA4 (50%), RA5 (50%), RA6 (50%), RA8 (50%)
2 ^o CURSO	RA4 (50%), RA5 (50%), RA6 (50%), RA7 (100%), RA8 (50%), RA9 (100%)

Resumen de los resultados de aprendizaje y su valoración:

RA1	Reconoce la estructura de un programa informático, identificando y relacionando los elementos propios del lenguaje de programación utilizado.	11,11%
RA2	Escribe y prueba programas sencillos, reconociendo y aplicando los fundamentos de la programación orientada a objetos.	11,11%
RA3	Escribe y depura código, analizando y utilizando las estructuras de control del lenguaje.	11,11%
RA4	Desarrolla programas organizados en clases analizando y aplicando los principios de la programación orientada a objetos.	11,11%
RA5	Realiza operaciones de entrada y salida de información, utilizando procedimientos específicos del lenguaje y librerías de clases.	11,11%
RA6	Escribe programas que manipulen información seleccionando y utilizando tipos avanzados de datos.	11,11%
RA7	Desarrolla programas aplicando características avanzadas de los lenguajes orientados a objetos y del entorno de programación.	11,11%
RA8	Utiliza bases de datos orientadas a objetos, analizando sus características y aplicando técnicas para mantener la persistencia de la información.	11,11%
RA9	Gestiona información almacenada en bases de datos manteniendo la integridad y consistencia de los datos.	11,11%

Correspondencia de las unidades didácticas por trimestre, **resultados de aprendizaje trabajados y ponderación de los criterios de evaluación:**

Unidad Didácticas por trimestre Extraordinariamente el curso 2024-25	Resultados Aprendizaje	Ponderación de los CE del RA por trimestre	CALIFICACIÓN: % de RA / total
1st trimestre: <i>Unit 1: Introduction Unit 2: Variables</i> <i>Unit 3: Conditionals Unit 4: Functions</i> <i>Unit 5: Iterations Unit 6: Strings</i> <i>Unit 7: Files Unit 8: Lists</i> <i>Unit 9: Dictionaries Unit 10: Tuples</i> <i>Unit 11: Regex Unit 12: Networked Programs</i> <i>Unit 13: WS Unit 14: OO Programming</i> <i>Unit 15: Databases Unit 16: Visualizing data</i>	RA1	100%	500/900 = 55,55%
	RA2	100%	
	RA3	100%	
	RA4	50%	
	RA5	50%	
	RA6	50%	
	RA8	50%	
	2nd trimestre: Unit 17: Organizing Files Unit 18: Working with Excel Spreadsheets Unit 19: Working with PDF and Word Documents Unit 20: Working with CSV Files and JSON Data Unit 21: Keeping Time, Scheduling Tasks, and Launching Programs Unit 22: Sending Email and Text Messages	RA4	
RA5		50%	
RA6		50%	
RA7		100%	
RA8		50%	
RA9		100%	

Criterios ortográficos

Dentro del Plan de Mejora de Escritura del Alumnado, el Dpto. de Informática ha acordado los siguientes criterios de calificación ortográficos para CFGS:

Penalización de 0,25 puntos por error gráfico (0,15 por tilde), hasta un máximo de 2,5 puntos.

Criterios de recuperación

El alumnado que no supere todos los Resultados de Aprendizaje en cada curso, tendrá la oportunidad de hacer un examen de recuperación días antes de que se lleve a cabo la evaluación ordinaria. Se le evaluará únicamente de los RA no superados con el fin de superarlos y con ello superar el curso.

De no superar todos los RA, en convocatoria ordinaria, deberá realizar una prueba teórico-práctica, en convocatoria extraordinaria, en donde se le evaluará de todos los RA que hasta la fecha no haya superado. Una vez superados estos RA, habrá superado el curso.

Recuperación para el alumnado que pasa a 2º curso sin haber superado Programación de 1º curso:

Durante el primer trimestre de segundo curso, se hace un repaso de lo que se imparte en primer curso. Por tanto, el alumnado que supere la evaluación de este primer trimestre se entenderá que ha adquirido satisfactoriamente los resultados de aprendizaje del primer curso.

Si aún así este alumnado no ha sido capaz de superar esta prueba, tendrá una nueva oportunidad días antes a la Evaluación Ordinaria de primer curso (mes de mayo).