

1. DATOS BÁSICOS

Asignatura	Fundamentos de Programación II
Titulación	Grado en Business Analytics
Escuela/ Facultad	Facultad de Ciencias Sociales y de la Comunicación
Curso	Segundo
ECTS	6
Carácter	Obligatoria
Idioma/s	Español
Modalidad	Presencial
Semestre	3
Curso académico	2024-2025
Docente coordinador	Fernando Aparicio Galisteo
Docente	Fernando Aparicio Galisteo

2. PRESENTACIÓN

La asignatura pretende continuar las bases para que los y las estudiantes puedan afrontar el aprendizaje de entornos que conlleven interfaces programables. Esta asignatura es la segunda de la materia de Fundamentos de Programación. Las competencias y resultados de aprendizaje que se describen en la memoria del título están asociados a la materia, por lo que se conseguirán al finalizar esta segunda asignatura.

Se espera acentuar los conocimientos adquiridos previamente para centrarnos en profundidad en los siguientes elementos:

- Diseño e Implementación de Clases.
- Atributos, constructores, métodos.
- Herencia.
- Diseño avanzado de clases. Sobrecarga y reescritura de métodos.
- Clases abstractas. Polimorfismo, Interfaces.
- Interpretación de diseños UML.
- Aseguramiento de la calidad y Patrones de diseño.
- Interfaz de usuario.

También se hará hincapié en la aplicación de buenas prácticas de programación, con el objetivo de que los estudiantes puedan generar código de manera limpia y estructurada.

3. COMPETENCIAS Y RESULTADOS DE APRENDIZAJE

Competencias básicas:

- CB1: Que los estudiantes hayan demostrado poseer y comprender conocimientos en un área de estudio que parte de la base de la educación secundaria general, y se suele encontrar a un nivel que, si bien se apoya en libros de texto avanzados, incluye también algunos aspectos que implican conocimientos procedentes de la vanguardia de su campo de estudio.

- CB2 - Que los estudiantes sepan aplicar sus conocimientos a su trabajo o vocación de una forma profesional y posean las competencias que suelen demostrarse por medio de la elaboración y defensa de argumentos y la resolución de problemas dentro de su área de estudio.

Competencias específicas:

- CE22 - Capacidad para seleccionar y aplicar las herramientas analíticas más adecuadas a cada situación de la compañía.
- CE31: Capacidad para gestionar la incertidumbre debida al constante cambio de las fuentes de información.

Resultados de aprendizaje:

- RA1. Desarrollar el **pensamiento algorítmico**. Traducir un problema en una secuencia de acciones a realizar que lo resuelva.
- RA2: Diseñar e implementar **soluciones para problemas** de dificultad media incluyendo el uso de bases de datos, estructuras de datos y orientación a objetos.
- RA3: Utilizar entornos de programación para realizar la **compilación**, el linkado y la ejecución de programas, así como la identificación y subsanación de errores en cada etapa.
- RA4. Valorar una buena **documentación** de los diseños realizados, así como la introducción de comentarios en el código para facilitar la interpretación y reutilización del software realizado.
- RA5: Participar en **trabajos grupales**, argumentando su posición, escuchando de forma activa y positiva y empatizando con las posiciones de sus compañeros.

En la tabla inferior se muestra la relación entre las competencias que se desarrollan en la asignatura y los resultados de aprendizaje que se persiguen:

Competencias	Resultados de aprendizaje
CB1, CE22, CE31	RA1, RA2, RA3
CB2, CE22, CE31	RA4, RA5

4. CONTENIDOS

La materia está organizada en los siguientes temas:

Tema 1. Repaso de Python

Tema 2: Introducción a la programación orientada a objetos (POO)

Tema 3. Introducción a UML (Lenguaje Unificado de Modelado)

Tema 4. Modelado básico de objetos y sus relaciones

Tema 5. Implementación básica (clases, herencia, asociaciones, etc.)

Tema 6. Implementación avanzada (sobrecarga, sobre escritura, modificadores de acceso, polimorfismo, interfaces, etc.)

5. METODOLOGÍAS DE ENSEÑANZA-APRENDIZAJE

A continuación, se indican los tipos de metodologías de enseñanza-aprendizaje que se aplicarán:

- Clase magistral / Web Conference
- Método del caso
- Aprendizaje basado en enseñanzas de taller y en proyectos

6. ACTIVIDADES FORMATIVAS

A continuación, se identifican los tipos de actividades formativas que se realizarán y la dedicación en horas del estudiante a cada una de ellas:

Actividad formativa	Número de horas
Tutorías	10
Lecciones magistrales	18
Lecciones magistrales asíncronas	12
Análisis de casos	30
Elaboración de informes y escritos	9,5
Trabajo autónomo	37,5
Actividades en talleres y/o laboratorios	33
TOTAL	150

7. EVALUACIÓN

A continuación, se relacionan los sistemas de evaluación, así como su peso sobre la calificación total de la asignatura:

Sistema de evaluación	Peso
Pruebas presenciales de conocimiento	40
Presentación del caso/problema (Exposiciones orales)	5
Actividades Resolución de problemas (Informes y escritos)	15
Caso / Problema (proyecto final)	25
Prácticas de laboratorio. Prototipos	15

En el Campus Virtual, cuando accedas a la asignatura, podrás consultar en detalle las actividades de evaluación que debes realizar, así como las fechas de entrega y los procedimientos de evaluación de cada una de ellas.

7.1. Convocatoria ordinaria

Para superar la asignatura en convocatoria ordinaria deberán satisfacerse todas las condiciones siguientes:

- Obtener una calificación superior igual a 5 en todas las pruebas de conocimiento presenciales.
- Obtener una calificación media igual o superior a 5 en las prácticas de laboratorio.
- Obtener una calificación superior igual a 5 en el proyecto final (caso/problema).
- Obtener una nota media final igual o superior a 5, a partir de las ponderaciones descritas en la tabla 4.

En caso de no cumplir alguno de los puntos anteriores la calificación será de suspenso. Se calculará la nota final a partir de la tabla 4, pudiendo obtener como máximo una calificación de 4.

7.2. Convocatoria extraordinaria

La convocatoria extraordinaria se realizará de manera individual. Para superar la asignatura en convocatoria extraordinaria es necesario obtener una calificación mayor o igual que 5 sobre 10 en la calificación final (media ponderada) de la asignatura.

Se deben superar las actividades no superadas en convocatoria ordinaria, tras haber recibido las correcciones correspondientes a las mismas por parte del profesor, o bien aquellas que no fueron entregadas.

Deben considerarse las siguientes situaciones:

- En caso de que la actividad 7 sea menor de 5, se deberá acordar con el profesor e implementar una nueva idea original y documentarla. Así se sustituirá el 20% correspondiente.

En caso de no cumplirse alguno de los anteriores aspectos, la calificación máxima en convocatoria extraordinaria, tras aplicar los porcentajes, sería de 4 sobre 10.

8. CRONOGRAMA

En este apartado se indica el cronograma con fechas de entrega de actividades evaluables de la asignatura:

Actividades evaluables	Fecha
<i>Actividad 1</i> Prueba de conocimiento	Tercer mes
<i>Actividad 2</i> Prueba de conocimiento	Quinto mes
<i>Actividad 3</i> Entrega de aplicación	Primer mes
<i>Actividad 4</i> Entrega de aplicación	Segundo mes
<i>Actividad 5</i> Entrega de aplicación	Tercer mes
<i>Actividad 6</i> Entrega de aplicación	Cuarto mes
<i>Actividad 7</i> Entrega de aplicación grupal	Quinto mes
<i>Actividad 8</i> Participación en foro	Todo el curso

Este cronograma podrá sufrir modificaciones por razones logísticas de las actividades. Cualquier modificación será notificada al estudiante en tiempo y forma.

9. BIBLIOGRAFÍA

A continuación, se indica la bibliografía recomendada:

- Rosalía Peña, “Resolución de problemas para ingenieros con Python estructurado”. Madrid (2016). Ibergarceta Publicaciones, S.L.

- Yue Zhang, "An introduction to Python and Computer Programming" 1st ed. 2015. Springer
- Martelli, Alex, "Python: guía de referencia", (2008), Anaya Multimedia.
- Shaw, Zed A., "Aprenda a programar con PYTHON 3", (2017), Anaya Multimedia
- WikiBooks: Object Oriented Programming
<http://en.wikibooks.org/wiki/Object_Oriented_Programming> [Consulta: 01/2022]
- "El lenguaje unificado de modelado" / Grady Booch, James Rumbaugh, Ivar Jacobson; traducción José Sáez Martínez ; supervisión de la traducción y revisión técnica Jesús J. García Molina. Disponible en Colección General (Campus Villaviciosa) (QA76.64.B66818 2004)
- "UML y patrones : introducción al análisis y diseño orientado a objetos y al proceso unificado" / Craig Larman ; traducción, Begoña Moros Valle ; supervisión de la traducción y revisión técnica, Jesús García Molina. Disponible en Colección General (Campus Villaviciosa) (QA76.64.L37318 2008)

10. UNIDAD DE ATENCIÓN A LA DIVERSIDAD

Desde la Unidad de Orientación Educativa y Diversidad (ODI) ofrecemos acompañamiento a nuestros estudiantes a lo largo de su vida universitaria para ayudarles a alcanzar sus logros académicos. Otros de los pilares de nuestra actuación son la inclusión del estudiante con necesidades específicas de apoyo educativo, la accesibilidad universal en los distintos campus de la universidad y la equiparación de oportunidades.

Desde esta Unidad se ofrece a los estudiantes:

1. Acompañamiento y seguimiento mediante la realización de asesorías y planes personalizados a estudiantes que necesitan mejorar su rendimiento académico.
2. En materia de atención a la diversidad, se realizan ajustes curriculares no significativos, es decir, a nivel de metodología y evaluación, en aquellos alumnos con necesidades específicas de apoyo educativo persiguiendo con ello una equidad de oportunidades para todos los estudiantes.
3. Ofrecemos a los estudiantes diferentes recursos formativos extracurriculares para desarrollar diversas competencias que les enriquecerán en su desarrollo personal y profesional.
4. Orientación vocacional mediante la dotación de herramientas y asesorías a estudiantes con dudas vocacionales o que creen que se han equivocado en la elección de la titulación.

Los estudiantes que necesiten apoyo educativo pueden escribirnos a:

orientacioneducativa@universidadeuropea.es

11. ENCUESTAS DE SATISFACCIÓN

¡Tú opinión importa!

La Universidad Europea te anima a participar en las encuestas de satisfacción para detectar puntos fuertes y áreas de mejora sobre el profesorado, la titulación y el proceso de enseñanza-aprendizaje.

Las encuestas estarán disponibles en el espacio de encuestas de tu campus virtual o a través de tu correo electrónico.

Tu valoración es necesaria para mejorar la calidad de la titulación.

Muchas gracias por tu participación.

PLAN DE TRABAJO DE LA ASIGNATURA

CÓMO COMUNICARTE CON TU DOCENTE

Cuando tengas una duda sobre los contenidos o actividades, no olvides escribirla en los foros de tu asignatura para que todos tus compañeros y compañeras puedan leerla.

¡Es posible que alguien tenga tu misma duda!

Si tienes alguna consulta exclusivamente dirigida al docente puedes enviarle un mensaje privado desde el Campus Virtual. Además, en caso de que necesites profundizar en algún tema, puedes acordar una tutoría.

Es conveniente que leas con regularidad los mensajes enviados por estudiantes y docentes, pues constituyen una vía más de aprendizaje.

CRONOGRAMA DE ACTIVIDADES

En este apartado se indica el cronograma de actividades formativas, así como las fechas de entrega de las actividades evaluables de la asignatura:

Semana	Contenidos	Actividades formativas/evaluables	Peso en la evaluación de la actividad evaluable
12	Temas 1 al 5	Actividad 1 Prueba de conocimiento	20%
20	Temas 1 al 5	Actividad 2 Prueba de conocimiento	20%
4	Tema 1	Actividad 3 Actividad individual 1	7,5%
8	Tema 2	Actividad 4 Actividad individual 2	7,5%
12	Tema 3	Actividad 5 Actividad individual 3	7,5%
16	Tema 4	Actividad 6 Actividad individual 4	7,5%
20	Tema 3 al 5	Actividad 7 Actividad grupal	20%
20	Temas 1 al 5	Actividad 8 Participación en foro	10%

Este cronograma podrá sufrir modificaciones que serán notificadas al estudiante en tiempo y forma.

DESCRIPCIÓN DE LAS ACTIVIDADES DE EVALUACIÓN

ACTIVIDAD INDIVIDUAL 1

Parte 1: Seleccionar alguno de los programas monolíticos disponibles en lenguaje Java, o bien desarrollar un programa sencillo que realice operaciones básicas a través de la consola de comandos (I/O) para realizar algún tipo almacenamiento en memoria.

Comentar profusamente el código (preferiblemente con notación tipo Javadoc). Debe incluirse el nombre del estudiante como primer comentario del código.

Compilar, depurar y guardar este programa.

Parte 2: A continuación, realiza un diseño inicial identificando los objetos detectados, o que podrían ser de utilidad para el futuro crecimiento de funcionalidad de esta aplicación. Lleva a cabo un diagrama simple de las dependencias localizadas. Guarda este diagrama como parte del entregable.

Una vez hecha la identificación, genera las clases necesarias e incorpora los datos (propiedades) y las funciones (métodos) correspondientes a cada clase, intentando minimizar el código incorporado en el programa principal.

ACTIVIDAD INDIVIDUAL 2

Parte 1: Sobre el ejemplo de reserva de vuelos trabajado en clase, realiza un diseño de clases (notación UML completa) que aumente la funcionalidad de este incluyendo, al menos, una herencia, una dependencia y dos tipos de asociaciones (agregación, composición o simple).

A continuación, adapta el código para que aparezcan todas estas relaciones acordes al diseño.

Parte 2: Lleva a cabo esta misma tarea a partir del programa seleccionado en la primera actividad o uno similar.

ACTIVIDAD INDIVIDUAL 3

Parte 1 (50%): Haciendo uso de los conceptos de diseño más detallados, es decir, distinguiendo entre diferentes tipos de dependencias, de herencia (entre clases, con clases abstractas y con interfaces), asociaciones (simple, agregación, composición) y las diferentes cardinalidades y navegabilidades, lleva a cabo una solución conteniendo:

1.1 Requisitos que incorporen al menos uno de los diferentes elementos mencionados (se puede adaptar de ejercicios anteriores para que contenga estos nuevos elementos)

1.2 Diagrama de clases con notación completa (incluyendo cardinalidades, navegabilidad y los atributos que constituyen las diferentes dependencias)

Parte 2 (50%): Implementa la solución acorde a estos requisitos.

ACTIVIDAD INDIVIDUAL 4

Parte 1 (80%): Aplica a la actividad 3 los conocimientos de arquitectura y diseño (SOLID por ejemplo) incorporando de este modo el conocimiento de los digital blocks 3 y 4, es decir, alguna interfaz gráfica y algún punto del código en el que creas conveniente el control de excepciones.

Parte 2 (20%): Entrega correcta y sin retrasos.

RÚBRICAS DE LAS ACTIVIDADES EVALUABLES

ACTIVIDADES INDIVIDUALES

Las actividades individuales deberán ser originales, es decir, en ningún caso serán evaluadas actividades que estén plagiadas.

Cuando se entreguen en plazo, se regirán por una rúbrica de 1 a 3 en función del porcentaje de cumplimiento de los objetivos planteados:

1. Cumple entre el 30% y el 50% de los objetivos.
2. Cumple entre el 50% y el 80% de los objetivos.
3. Cumple más del 80% de los objetivos.

ACTIVIDAD GRUPAL

	3	2	1	0
Memoria y trabajo en equipo	Los contenidos de toda la estructura son adecuados, comenzando por las especificaciones, y es original	Contiene cronograma, porcentaje de la carga de trabajo e indicadores de haber seguido la planificación	Está bien estructurado y no contiene errores ni faltas	El documento tiene faltas o está mal estructurado
Especificación de Requisitos Funcionales	Los requisitos están detallados y ayudan a realizar un diseño acorde con todo detalle	Los requisitos están detallados, pero dan lugar a una amplia ambigüedad al realizar el diseño	Se acompaña de un apartado de requisitos, estando poco detallados	No se acompaña de requisitos
Diseño	El diseño coincide rigurosamente con el sistema prototipado, conteniendo elementos de todos los tipos de relaciones	El diseño es correcto y coincide con el prototipo desarrollado, aunque podría ser más detallado o bien no contiene algunos de las relaciones estudiadas	Se ha incluido un diseño que coincide con el prototipo resultante, aunque carece de detalle en las relaciones planteadas	No contiene diseño o contiene un diseño con errores en la notación
Implementación	La implementación cumple con todos los requisitos. Los archivos fuente desarrollados están correctamente documentados	La implementación es correcta, sin haber documentado el código. Se incluyen pruebas unitarias y de integración	Se presenta una implementación que funcionalmente no cumple con ciertos requisitos	La implementación tiene errores o no se llega a presentar

REGLAMENTO PLAGIO

Atendiendo al Reglamento disciplinario de los estudiantes de la Universidad Europea:

- El plagio, en todo o en parte, de obras intelectuales de cualquier tipo se considera falta muy grave.
- Las faltas muy graves relativas a plagios y al uso de medios fraudulentos para superar las pruebas de evaluación, tendrán como consecuencia la pérdida de la convocatoria correspondiente, así como el reflejo de la falta y su motivo, en el expediente académico.

REGLAMENTO USO DE IA

El estudiante debe ser el autor o autora de sus trabajos/actividades.

El uso de herramientas de Inteligencia Artificial (IA) debe ser autorizado por el docente en cada trabajo/actividad, indicando de qué manera está permitido su uso. El docente informará previamente en qué situaciones se podrá usar herramientas de IA para mejorar la ortografía, gramática y edición en general. El estudiante es responsable de precisar la información dada por la herramienta y declarar debidamente el uso de cualquier herramienta de IA, en función de las directrices que marque el docente. La decisión final sobre la autoría del trabajo y la idoneidad del uso reportado de una herramienta de IA recae en el docente y en los responsables de la titulación.