

DATOS DE IDENTIFICACIÓN

Titulación:	Grado en Análisis de Negocios/Business Analytics		
Rama de Conocimiento:	Ciencias Sociales y Jurídicas		
Facultad/Escuela:	Derecho, Empresa y Gobierno		
Asignatura:	Programación II		
Tipo:	Optativa	Créditos ECTS:	6
Curso:	3	Código:	5352
Periodo docente:	Quinto semestre		
Materia:	Informática Aplicada al Análisis de Negocios / IT applied to Business Analytics		
Módulo:	Formación Disciplinar		
Tipo de enseñanza:	Presencial		
Idioma:	Castellano		
Total de horas de dedicación del alumno:	150		
Equipo Docente	Correo Electrónico		
Raúl Antonio del Águila Escobar	raul.delaguila@ufv.es		

DESCRIPCIÓN DE LA ASIGNATURA

Como continuación de la asignatura de Programación I, esta asignatura profundiza en el desarrollo de los conocimientos de programación y la resolución de problemas mediante el desarrollo de programas informáticos.

Se cubrirán aspectos como el uso avanzado de estructuras de datos objetos, funciones de orden superior, clases, errores y gestión de excepciones (continuación del Programación I), pruebas, concurrencia y paralelismo, conexiones a bases de datos, programación reactiva y de red.

Se trabajará en lenguaje de programación Python 3

Como continuación de la asignatura de Programación I, esta asignatura profundiza en el desarrollo de los conocimientos de programación, su aplicación en la analítica de datos y la puesta en producción de aplicaciones.

La asignatura cubrirá estas materias en 3 bloques diferenciados:

1. DevOps: se utilizarán, entre otras, tecnologías orientadas al versionado de código y despliegue de aplicaciones, tales como Git, GitActions, slack, contenedores y orquestación.
2. Programación avanzada en Python 3: este bloque profundizará en conceptos previamente analizados en la asignatura de Programación I, pero desde una perspectiva orientada al diseño, así como otros conceptos avanzados presentes en Python 3.
3. Programación orientada a la analítica de datos: se expondrán las principales librerías y frameworks de Python 3 para el manejo de datos en proyectos de analítica.

OBJETIVO

El principal objetivo de esta asignatura consiste en que el estudiante profundice en el análisis, diseño, implementación y prueba de software usando el paradigma de orientación a objetos. Al finalizar la asignatura, el alumno será capaz de emplear métodos, técnicas y herramientas para la construcción de aplicaciones eficientes y robustas, particularmente en el entorno de la ciencia de datos y como soporte a otras áreas como aprendizaje automático (machine learning) e inteligencia artificial.

CONOCIMIENTOS PREVIOS

Es altamente recomendable haber superado la asignatura de Programación I o tener conocimientos de Python análogos, así como soltura en el manejo de entornos basados en Unix.

CONTENIDOS

1. DevOps: Gestión de entornosMódulos y paquetesGit y Git ActionsPruebas unitariasContenedores: Docker, Docker-ComposeLogging
2. Programación avanzada en Python 3: Fundamentos teóricos del diseño orientado a objetos, herencia, interfaz, cohesión, acoplamiento, principios SOLID y patrones de diseñoElementos avanzados del lenguaje Python 3: Funciones de orden superior, decoradores, metaclasses, descriptores, sobrecarga de operadores.
3. Introducción a las metodologías y técnicas orientadas al desarrollo ágil: TDDExtreme Programming
4. Programación orientada a la analítica de datos: Iteradores, generadores, Pandas, NumPy, Matplotlib / Seaborn

ACTIVIDADES FORMATIVAS

Presenciales:

- Combinación de lecciones expositivas con clases prácticas, de manera que se favorezca la participación del alumno y la interacción alumno- profesor y alumno-alumno como vía para fomentar el aprendizaje colaborativo y la capacidad de autoaprendizaje, todo ello mediante estrategias de resolución de problemas y metodologías de intervención (aprendizaje basado en problemas, ABP)

No presenciales:

- Estudio individual, que permitirá trabajar en la fijación de los conceptos teóricos abordados en las clases expositivas.
- El profesor planteará problemas o tecnologías para que se piensen o investiguen fuera de clase y se aborden posteriormente en el aula (Flipped Learning).
- Entregas intermedias de la práctica final mediante los cuales el alumno podrá poner en práctica los conocimientos adquiridos y recibirá feedback continuo de su progreso. Estas tareas deberán abordarse mediante una metodología ágil, de forma que se facilite la revisión por pares y las tareas asignadas a los equipos, adaptando la metodología eXtremme Programming a la metodología didáctica ABProyectos.

El trabajo colaborativo se hará usando herramientas colaborativas tales como Office 365, Slack o en entorno Git (sistema de código abierto para el trabajo y control de versión para el desarrollo de software).

Todo el estudio y trabajo realizado por el alumno será supervisado y guiado por el profesor mediante tutorías, individuales o en grupo. El alumno se pondrá en contacto con el profesor para concertar citas (presenciales o virtuales).

Las conductas de plagio, así como el uso de medios ilegítimos en las pruebas de evaluación, serán sancionados conforme a lo establecido en la Normativa de Evaluación y Convivencia de la universidad. El control de plagio se extiende a todo trabajo entregado como propio, incluido el código.

DISTRIBUCIÓN DE LOS TIEMPOS DE TRABAJO

ACTIVIDAD PRESENCIAL	TRABAJO AUTÓNOMO/ACTIVIDAD NO PRESENCIAL
60 horas	90 horas

COMPETENCIAS

Competencias básicas

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

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

Que los estudiantes tengan la capacidad de reunir e interpretar datos relevantes (normalmente dentro de su área de estudio) para emitir juicios que incluyan una reflexión sobre temas relevantes de índole social, científica o ética

Que los estudiantes puedan transmitir información, ideas, problemas y soluciones a un público tanto especializado como no especializado

Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía



Competencias generales

Capacidad de organización, sistematización y planificación en la identificación de problemas, pautas y modelos en el contexto del big data

Capacidad para el cumplimiento de objetivos, resolución de problemas y toma de decisiones en un entorno de datos masivos tanto cuantitativos como cualitativos

Capacidad de liderazgo y de trabajar en equipo en la sociedad de la información

Competencias específicas

Conocer, comprender y aplicar los fundamentos y las herramientas de la programación para el uso y explotación de la información, garantizando su seguridad e integridad

Saber manejar herramientas cuantitativas e informáticas para la toma de decisiones

Ser capaz de conocer los fundamentos, paradigmas y técnicas propias de los sistemas inteligentes y analizar, diseñar y construir sistemas, servicios y aplicaciones informáticas que utilicen dichas técnicas en el ámbito del big data

Conocer y saber utilizar a nivel de analista herramientas de estadística avanzada, almacén de datos, bases de datos relacionales, bases de datos no relacionales y sistemas de gestión de big data.

RESULTADOS DE APRENDIZAJE

Desarrolla programas avanzados basados en el paradigma de programación orientada a objetos, desde el diseño y definición de las clases y objetos que los forman, hasta su implementación y posterior depuración con un lenguaje y entorno de desarrollo determinado.

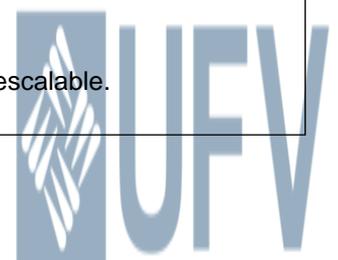
Desarrolla programas estables y fáciles de mantener: documenta, integra programas de pruebas, mantiene requerimientos de módulos, utiliza correctamente las excepciones y los registros (logging).

Conoce librerías de uso común para el manejo de tablas, arrays multidimensionales y de computación científica y técnica.

Es capaz de crear programas con que trabajan con múltiples procesos o múltiples hilos (threads) y coordinar su ejecución.

Es capaz de utilizar herramientas que permiten poner en producción aplicaciones de forma escalable.

SISTEMA DE EVALUACIÓN DEL APRENDIZAJE



Sistema de evaluación:

El sistema de evaluación contempla los siguientes elementos:

[1] Examen teórico-práctico: 50% en la nota final.

[2] Problemas o actividades propuestos: 10% en la nota final.

[3] Práctica final: 35% en la nota final.

[4] Participación en clase e implicación en la asignatura: 5% en la nota.

Se exige una nota mínima de 5 en el examen teórico-práctico [1]. Se exige una nota mínima de 5 en la media de actividades de clase y prácticas ([2] y [3]).

La práctica final es obligatoria y consistirá en un proyecto cuyos requisitos iniciales establecerá el profesor el primer día de clase y que los alumnos deberán defender frente a sus compañeros. Conforme avance la asignatura, los alumnos deberán ir entregando al profesor una memoria de avance del proyecto y prototipos iniciales en el que conste cómo va avanzando de acuerdo con las materias introducidas durante la asignatura. Asimismo, la práctica conllevará la investigación por parte del alumno de tecnologías sin las cuales la práctica no puede desarrollarse.

Los problemas propuestos tendrán una fecha de entrega. Los estudiantes tendrán la opción de presentar el trabajo con posterioridad a la fecha de entrega, hasta un máximo de 7 días, pero se aplicará una penalización de un 5% diario, hasta un máximo del 35%.

No se admitirán entregas con más de 7 días de retraso.

No es obligatorio presentar todas las tareas y actividades de clase, pero la nota de una actividad o tarea no presentada se evalúa con un cero y hará media con el resto de las entregas.

Sistema de evaluación alternativo:

Para alumnos con dispensa académica aprobada y alumnos UFV en estancia de intercambio: un alumno podrá obtener dispensa académica por motivos de trabajo, incompatibilidad de horarios, enfermedad u otros que estime la Dirección de la Carrera, previa petición de la misma en Coordinación Académica aportando la documentación que se exija a tal efecto. Una vez concedida, se notificará oficialmente tanto al profesor afectado como al alumno que lo haya solicitado desde Coordinación Académica.

En cualquiera de los casos, es responsabilidad del alumno el seguimiento de la asignatura, así como de los aspectos que componen su evaluación.

Para alumnos en segunda matrícula o sucesivas: estos alumnos podrán acogerse al sistema de evaluación continua, siempre y cuando cumpla con todos los requisitos incluida la asistencia a clase. En caso contrario deberán seguir el sistema alternativo de evaluación, no siendo necesario solicitar la dispensa académica, pero debe avisar mediante correo electrónico al profesor correspondiente para que éste lo tenga en cuenta. Estos alumnos, que estén exentos de la obligación de asistir a clase, bien por segunda matrícula en la asignatura o sucesivas, bien por contar con autorización expresa de la Dirección del Grado, serán evaluados por el mismo tipo de pruebas. El 5% de la participación en clase podrán obtenerlo asistiendo al menos a tres tutorías con el profesor responsable de la asignatura.

Evaluación convocatoria extraordinaria:

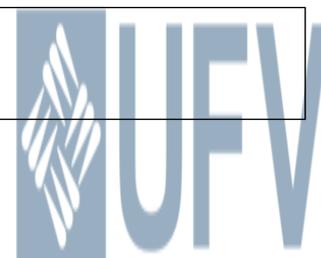
Los alumnos que no hayan alcanzado la nota mínima en el examen [1], habiendo suspendido por tanto en la convocatoria ordinaria, podrán optar a una recuperación del examen en la convocatoria extraordinaria. Los alumnos que no hayan alcanzado la nota mínima en la media de actividades de clase y prácticas ([2] y [3]), habiendo suspendido por tanto en la convocatoria ordinaria, podrán optar por la recuperación de prácticas en la convocatoria extraordinaria. Los alumnos que suspendan una o las dos partes en convocatoria extraordinaria tendrán que repetir íntegramente la asignatura (todas las partes) en la nueva convocatoria ordinaria.

Plagio:

Las conductas de plagio, así como el uso de medios ilegítimos en las pruebas de evaluación, serán sancionados conforme a lo establecido en la Normativa de Evaluación y Convivencia de la universidad. El control de plagio se extiende a todo trabajo entregado como propio, incluido el código.

BIBLIOGRAFÍA Y OTROS RECURSOS**Básica**

John Hunt Advanced Guide to Python 3 Programming Springer, 2019

Complementaria

Martin Fowler Refactoring, Improving the design of existing code Addison-Wesley

Sherwin John C. Tragura Building Python Microservices with FastAPI Packt, 1st. Ed.

Ebenezer Don Git Prodigy Leanpub

