

FICHA DE ASIGNATURA

Título: Programación Avanzada

Descripción: Durante el desarrollo de esta asignatura el estudiante progresará hacia campos más avanzados y específicos dentro de la programación, con aplicaciones prácticas directas y ejemplos de uso en la industria actual. Será capaz de crear sistemas de software complejos que puedan ser usados por terceros en situaciones reales. Se centra además en las buenas prácticas de programación (patrones de diseño, testing, clean code), paradigmas de desarrollo de software actuales (orientación a objetos, metodologías ágiles, programación funcional), estructuras de datos avanzadas y sus algoritmos relacionados para resolver la mayoría de los desafíos planteados para trabajar con ellas. Todo ello se muestra a través de un proyecto transversal, con la creación de un videojuego completo.

Carácter: Obligatoria

Créditos ECTS: 6

Contextualización: El desarrollo de piezas de software completas para resolver determinados problemas es el objetivo de cualquier programador. Estos programas o aplicaciones están presentes en todas partes, desde los móviles, hasta páginas webs, programas de escritorio,... Y constituyen la base de muchas industrias, además de herramientas indispensables para el funcionamiento de muchas otras. Ser capaz de enfrentarse a dichos problemas y crear aplicaciones capaces de resolverlos es imprescindible para triunfar en cualquier ámbito laboral.

Modalidad: Online

Temario:

1. Virtual Environments
2. Módulos, Paquetes, Namespaces, Uso del Sistema (Paths, Shell, Variables de Entorno, Sys, Os)
3. Control de Versiones y Git
4. Ingeniería de Software, Arquitectura, Clean Code, Craftmanship, Patrones de Diseño (Creacionales, Estructurales, de Comportamiento)
5. Testing, Test Driven Development, Testing Unitario, Testing Funcional, Fixtures, PyTest, Doubles
6. Gestión de Proyectos, Agile, Scrum, eXtreme Programming
7. Análisis de Rendimiento
8. Estructuras de Datos Avanzadas, Built-In, Mapas, HashTables, Arrays, Lists, Sets, Stacks, Queues, Heaps, Trees, Graphs
9. Algoritmos, Divide & Conquer, Recursion, Backtracking, Tree Traversal, Sorting, Searching, A*, Analysis, Big O
10. Desarrollo de Videojuegos, PyGame
11. Programación Funcional, Funciones Puras, Estructuras de Datos Inmutables, Lazy Evaluation, Recursion

Competencias:

Actividades Formativas:

Actividad Formativa	Horas	Presencialidad
Clases Expositivas	15	0%
Clases Prácticas sobre laboratorio informático	15	0%
Tutorías Online	10	30%
Trabajo Autónomo	110	0%
Total	150	

Metodologías docentes:

- Lección magistral
- Estudio de casos
- Resolución de problemas
- Simulaciones
- Laboratorio informático virtual
- Seguimiento

Sistema de Evaluación:

Sistemas de evaluación	Ponderación mínima	Ponderación máxima
Evaluación de portafolio. Informe sobre resolución de problemas o estudio de casos	20%	30%
Evaluación del portafolio. Informe sobre Diseño de Proyectos	10%	20%
Participación en grupos de debate	10%	10%
Evaluación de la prueba	40%	60%

Normativa específica: No Procede

Bibliografía:

- Tuckfield, B. (2021) *Dive Into Algorithms: A Pythonic Adventure for the Intrepid Beginner*. San Francisco: No Starch Press.
- Zlobin, G. (2013) *Learning Python Design Patterns*. Birmingham, Packt Publishing.
- Gamma, E. & Helm, R. & Johnson, R & Vlissides, J (1994) *Design Patterns: Elements of Reusable Object-Oriented Software*. Boston Massachusetts, Addison-Wesley.
- Martin, R. (2008) *Clean Code: A Handbook of Agile Software Craftsmanship*. London, Pearson Publishing.
- Oliveira, B. (2018) *pytest Quick Start Guide: Write better Python code with simple and maintainable tests*. Birmingham, Packt Publishing.

- *Meszaros, G. (2007) xUnit Test Patterns: Refactoring Test Code. Boston Massachusetts, Addison-Wesley.*
- *Appelo, J. (2010) Management 3.0: Leading Agile Developers, Developing Agile Leaders. Boston Massachusetts, Addison-Wesley.*
- *Cormen, T. & Leiserson, C. & Rivest, R. & Stein, C. (2010) Introduction to Algorithms. PHI Learning.*
- *Wilkes, M. (2020) Advanced Python Development: Using Powerful Language Features in Real-World Applications. New York, Apress.*
- *McGugan, W. (2015) Beginning Python Games Development, With PyGame(2nd Edition) New York, Apress.*
- *Sweigart, A. (2021) Beyond The Basic Stuff With Python: Best Practices for Writing Clean Code. Birmingham, Packt Publishing.*
- *Slatkin, B. (2019) Effective Python: 90 Specific Ways to Write Better Python. Boston Massachusetts, Addison-Wesley.*