GUIA DOCENTE DE LA ASIGNATURA (∞)

INFORMÁTICA APLICADA A LA BIOQUÍMICA

Curso 2017-2018

(Fecha última actualización: 30/06/2017)

(Fecha de aprobación en Consejo de Departamento: 30/06/2017)

MÓDULO	MATERIA	CURSO	SEMESTRE	CRÉDITOS	TIPO
Física, Matemática e Informática para las Biociencias Moleculares	Informática aplicada a la Bioquímica	2º	3º	6	Obligatoria

PROFESORES[1]	DIRECCIÓN COMPLETA DE CONTACTO PARA TUTORÍAS (Dirección postal, teléfono, correo electrónico, etc.)
• Rafael Alcalá Fernández (Responsable de la asignatura)	La dirección completa del profesorado para tutorías puede consultarse en la web: http://decsai.ugr.es/index.php?p=profesores
	HORARIO DE TUTORÍAS Y/O ENLACE A LA PÁGINA WEB DONDE PUEDAN CONSULTARSE LOS HORARIOS DE TUTORÍAS[1]
	El horario de tutorías del profesorado puede consultarse en la web: http://decsai.ugr.es/index.php?p=profesores

GRADO EN EL QUE SE IMPARTE	OTROS GRADOS A LOS QUE SE PODRÍA OFERTAR
Grado en Bioquímica	

PRERREQUISITOS Y/O RECOMENDACIONES (si procede)

Se recomienda haber cursado Matemáticas en Bachillerato

BREVE DESCRIPCIÓN DE CONTENIDOS (SEGÚN MEMORIA DE VERIFICACIÓN DEL GRADO)

- Conceptos básicos. Sistemas numéricos y lógica booleana. Datos. Tipos de datos. Cálculo computacional. Errores. Sistemas operativos.
- Concepto de programa. Diagrama de flujo. Lenguajes de programación. Algoritmos y estructuras de datos.
- Elementos de programación en C/Python (o entorno equivalente).

[1] Consulte posible actualización en Acceso Identificado > Aplicaciones > Ordenación Docente

(∞) Esta guía docente debe ser cumplimentada siguiendo la "Normativa de Evaluación y de Calificación de los estudiantes de la Universidad de Granada" (http://secretariageneral.ugr.es/pages/normativa/fichasugr/ncg7121/!)

- Aplicación de la informática a la Bioquímica, la Biología Molecular.
- Algoritmos de diferenciación e integración numérica. Simulación virtual de procesos bioquímicos.
- Métodos de inteligencia artificial: redes neuronales y algoritmos genéticos. Aplicaciones.

COMPETENCIAS GENERALES Y ESPECÍFICAS

- BÁSICAS Y GENERALES

CG5.- Haber desarrollado las habilidades de aprendizaje necesarias para emprender estudios posteriores de especialización con un alto grado de autonomía, incluyendo la capacidad de asimilación de las distintas innovaciones científicas y tecnológicas que se vayan produciendo en el ámbito de las Biociencias Moleculares.
CB5.- Que los estudiantes hayan desarrollado aquellas habilidades de aprendizaje necesarias para emprender estudios posteriores con un alto grado de autonomía.

- TRANSVERSALES

CT1.- Adquirir la capacidad de razonamiento crítico y autocrítico.
CT4.- Tener capacidad de aprendizaje y trabajo autónomo.
CT6.- Saber reconocer y analizar un problema, identificando sus componentes esenciales, y planear una estrategia científica para resolverlo.
CT7.- Saber utilizar las herramientas informáticas básicas para la comunicación, la búsqueda de información, y el tratamiento de datos en su actividad profesional.

- ESPECÍFICAS

CE24.- Poseer las habilidades matemáticas, estadísticas e informáticas para obtener, analizar e interpretar datos, y para entender modelos sencillos de los sistemas y procesos biológicos a nivel celular y molecular.
CE25.- Saber buscar, obtener e interpretar la información de las principales bases de datos biológicos (genómicos, transcriptómicos, proteómicos, metabolómicos y similares derivados de otros análisis masivos) y de datos bibliográficos, y usar las herramientas bioinformáticas básicas.

OBJETIVOS (EXPRESADOS COMO RESULTADOS ESPERABLES DE LA ENSEÑANZA)

El estudiante debería:

- Conocer los componentes de un sistema informático.
- Conocer el funcionamiento de un ordenador.
- Mostrar algunas nociones sobre sistemas operativos y software de un ordenador.
- Comprender los conceptos de programación.
- Dominar los aspectos básicos de Python (o entorno equivalente).
- Ser capaz de diseñar y programar algoritmos para resolver problemas generales, como paso indispensable para abordar la resolución de problemas específicos de Bioquímica.
- Conocer y saber implementar algunos algoritmos de diferenciación e integración numérica.
- Conocer un panorama de herramientas de programación aplicadas al campo de la Bioquímica (librerías o cajas de herramientas para Bioinformática).
- Comprender la estructura y funcionamiento de una base de datos.
- Conocer los casos específicos de bases de datos en el ámbito de la Bioquímica.
- Conocer los aportes que el campo de la Inteligencia Artificial realiza en el ámbito de la Bioquímica.
- Conocer los conceptos básicos de los algoritmos bioinspirados.
- Conocer los principios de la simulación virtual aplicada al ámbito de la Bioquímica.

TEMARIO DETALLADO DE LA ASIGNATURA

TEMARIO TEÓRICO (CLASES MAGISTRALES – 0,9 ECTS/ 23 h):

- Tema 1. Introducción a la Informática (1h)
- Tema 2. Fundamentos de programación (2h)
- Tema 3. Introducción al Lenguaje de Programación Python (2h)
- Tema 4. Programación en Python (10h)
- Tema 5. Bases de Datos (2h)
- Tema 6. Inteligencia Artificial y Bioquímica (3h)
- Tema 7. Algoritmos Bioinspirados (3h)

TEMARIO PRÁCTICO (TALLER DE PROGRAMACIÓN – 1,4 ECTS/ 35 h):

Seminarios/Talleres (0,2 ECTS/ 5hs)

- Seminario 1. Introducción a los Sistemas Operativos (Windows, Unix/Linux) (1h)
- Seminario 2. Introducción al entorno de programación de Python (1h)
- Seminario 3. Herramientas para la Diferenciación e Integración Numérica (1h)
- Seminario 4. Búsqueda de información en Internet y en Bibliotecas Digitales (1h)
- Seminario 5. Introducción al módulo Biopython para Bioinformática (1h)

Prácticas de Laboratorio (1,2 ECTS/ 30 h):

- Práctica 1. Sistemas Operativos (1h)
- Práctica 2. Primeros Programas en Python (2h)
- Práctica 3. Estructuras de Control (7h)
- Práctica 4. Tipos de Datos Complejos (8h)
- Práctica 5. Creación de Módulos (4h)
- Práctica 6. Algoritmos de Diferenciación e Integración Numérica (3h)
- Práctica 7. Interacción con Bases de Datos en Bioquímica: PDB, SwissProt, GenBank, PIR, EMBL (2h)
- Práctica 8. Herramientas de Programación para Bioquímica: Uso del módulo Biopython (3h)

BIBLIOGRAFÍA

BIBLIOGRAFÍA FUNDAMENTAL:

- J.G. Molina, F.M. Dato, et al., Una introducción a la Programación. Un enfoque algorítmico, Thompson, 2005.
- How to Think Like a Computer Scientist: Interactive Edition, Runestone Interactive Project at Luther College, led by Brad Miller and David Ranum. Based on the original work by J. Elkner, A.B. Downey and C. Meyers, How to Think Like a Computer Scientist: Learning with Python, 2nd Edition (Python 2.x), 2012: http://interactivepython.org/runestone/static/thinkcspy/index.html
- J. Elkner, A.B. Downey and C. Meyers, Aprenda a Pensar como un Programador: Con Python, Green Tea Press: Wellesley, Massachusetts, 2002 (Python 2.x), http://www.cmat.edu.uy/cmat/cursos/licenciatura/cursos/computacion/documentos/Think Python-esp.pdf
- Lahoz-Beltra, Rafael, Bioinformática: simulación, vida artificial e Inteligencia Artificial, López de Santos, 2004.

BIBLIOGRAFÍA COMPLEMENTARIA:

- E. Rich, K. Knight, Inteligencia Artificial, segunda edición, Mc Graw Hill Co. 1992.
- S. Russell, P. Norvig, Inteligencia Artificial: un Enfoque Moderno, 2ª Edición, Ed. Pearson-Prentice Hall, 2004.
- Pons O., Marín N., Medina J.M., Acid S., Vila M.A., Introducción a las Bases de Datos: El modelo Relacional, Thomson Paraninfo, 2005.
- James A. Goodrich; Jennifer F. Kugel, Binding and Kinetics for Molecular Biologists, Spiral bound, Cold Spring

Harbor Laboratory Press
- Uri Alon , An Introduction to Systems Biology : Design Principles of Biological Circuits, CRC Press, 2006

ENLACES RECOMENDADOS

http://biopython.org//wiki/Biopython
http://www.open-bio.org/wiki/Main_Page
http://www.mathworks.es/help/toolbox/bioinfo/
http://bioperl.org/wiki/Main_Page
http://biojava.org/wiki/Main_Page

METODOLOGÍA DOCENTE

	ACTIVIDAD FORMATIVA	COMPETENCIAS	ECTS	%	
Actividades Presenciales	Clases de teoría, problemas y casos prácticos	CG5, CB5, CE24, CE25, CT6	0,9	15 %	40 %
	Clases prácticas de laboratorio y/o de informática	CG5, CB5, CE24, CE25, CT1, CT5, CT6, CT7	1,4	23,3 %	
	Evaluación	CG5, CB5, CT1, CT7	0,1	1,7 %	

	ACTIVIDAD FORMATIVA	COMPETENCIAS	ECTS	%	
Actividades No Presenciales	Estudio de la Teoría	CG5, CB5, CE24, CE25, CT1, CT6, CT7	0,8	13,3 %	60 %
	Programación	CG5, CB5, CE25, CT6, CT7	1,8	30 %	
	Resolución de Ejercicios	CG5, CB5, CE25, CT6, CT7	1	16,7 %	

1. ACTIVIDAD FORMATIVA: Clases de teoría, problemas y casos prácticos (Lección magistral/expositiva)
- Descripción: Presentación en el aula de los conceptos fundamentales y desarrollo de los contenidos propuestos.
- Propósito: Transmitir los contenidos de las materias del módulo motivando al alumnado a la reflexión, facilitándole el descubrimiento de las relaciones entre diversos conceptos matemáticos, estadísticos e informáticos y formarle una mentalidad analítica para la resolución de problemas y el tratamiento de datos biológicos.
- Contenido en ECTS: 0,9
- Competencias: CG5, CB5, CE24, CE25, CT6

2. ACTIVIDAD FORMATIVA: Clases prácticas de laboratorio y/o de informática (Taller de programación - Seminarios)
- Descripción: Actividades a través de las cuales se pretende mostrar al alumnado cómo aplicar los conocimientos teóricos adquiridos en la resolución de ejercicios, supuestos prácticos relativos a la aplicación de normas técnicas o resolución de problemas. Los seminarios tratan en profundidad temáticas concretas relacionadas con la materia. Incorpora actividades basadas en la indagación, el debate, la reflexión y el intercambio.
- Propósito: Desarrollo en el alumnado de las habilidades instrumentales y de las competencias cognitivas y procedimentales de la materia.
- Contenido en ECTS: 1,4
- Competencias: CG5, CB5, CE24, CE25, CT1, CT5, CT6, CT7

3. ACTIVIDAD FORMATIVA: Evaluación (Realización de Exámenes).
- Descripción: Prueba tipo test de conocimientos y resolución de problemas de programación.

- Propósito: Evaluar los conocimientos adquiridos por el alumno, tanto teóricos como de programación.
- Contenido en ECTS: 0,1
- Competencias: CG5, CB5, CT1, CT7

4. ACTIVIDAD FORMATIVA: Actividades no presenciales.
Actividades no presenciales individuales (Actividad no presencial de aprendizaje mediante el estudio de la materia, el análisis de documentos, la elaboración de memorias...)
- Descripción:
1) Estudio de la Teoría – Estudio individualizado de los contenidos de la materia. Contenido en ECTS: 0,8.
2) Programación – Implementación de algoritmos y codificación de los mismos para la resolución de problemas propuestos por el profesor (durante las sesiones prácticas, o bien como paso final en la resolución de ejercicios propuestos), con carácter individual que permitan al alumno adquirir los conocimientos de programación necesarios para construir sus propios programas. Contenido en ECTS: 1,8.
3) Resolución de ejercicios – Actividades (guiadas y no guiadas) propuestas por el profesor a través de las cuales y de forma individual se profundiza en aspectos concretos de la materia posibilitando al estudiante avanzar en la adquisición de determinados conocimientos y procedimientos de la misma. Contenido en ECTS: 1.
- Propósito: Favorecer en el estudiante la capacidad para autorregular su aprendizaje, planificándolo, diseñándolo, evaluándolo y adecuándolo a sus especiales condiciones e intereses.
- Competencias: TODAS menos CT4

EVALUACIÓN (INSTRUMENTOS DE EVALUACIÓN, CRITERIOS DE EVALUACIÓN Y PORCENTAJE SOBRE LA CALIFICACIÓN FINAL, ETC.)

Se llevará a cabo una evaluación mixta que incluye la realización de un test de conocimientos adquiridos, la presentación de una colección de ejercicios resueltos y un examen de programación básica, consistente en el desarrollo controlado de uno o varios de los ejercicios resueltos durante el curso por el estudiante.

- EVALUACIÓN ORDINARIA:
  - Prueba tipo test de conocimientos: 40% (competencias evaluadas CG5,CB5,CE24,CE25,CT6)
  - Examen de programación: 30 % (competencias evaluadas CG5,CB5,CE24,CT1,CT6,CT7)
  - Colección de ejercicios resueltos: 30 % (competencias evaluadas CG5,CB5,CE24,CE25,CT1,CT5,CT6,CT7)

- EVALUACIÓN EXTRAORDINARIA:

  En el examen extraordinario, el 30% de la "Colección de ejercicios resueltos" (prácticas realizadas durante el curso) quedará incluida dentro del "Examen de programación", que en tal caso será un 60% de la nota y que por tanto incluirá ejercicios del mismo tipo que los resueltos en la "Colección de ejercicios resueltos". De esta manera la estructura del examen extraordinario es la siguiente: "Prueba tipo test de conocimientos" (40%) y "Examen de programación" (60 %).

DESCRIPCIÓN DE LAS PRUEBAS QUE FORMARÁN PARTE DE LA EVALUACIÓN ÚNICA FINAL ESTABLECIDA EN LA "NORMATIVA DE EVALUACIÓN Y DE CALIFICACIÓN DE LOS ESTUDIANTES DE LA UNIVERSIDAD DE GRANADA"

- Según la Normativa de Evaluación y de Calificación de los Estudiantes de la Universidad de Granada (modificada en Consejo de Gobierno el 26 de octubre de 2016) la evaluación será preferentemente continua, entendiendo por tal la evaluación diversificada que se establezca en las Guías Docentes de las asignaturas. No obstante, las Guías Docentes contemplarán la realización de una evaluación única final a la que podrán acogerse

aquellos estudiantes que no puedan cumplir con el método de evaluación continua por motivos laborales, estado de salud, discapacidad o cualquier otra causa debidamente justificada que les impida seguir el régimen de evaluación continua" (artículo 6.2).

Para acogerse a la evaluación única final, el estudiante, en las dos primeras semanas de impartición de la asignatura, o en las dos semanas siguientes a su matriculación si ésta se ha producido con posterioridad al inicio de la asignatura, lo solicitará, a través del procedimiento electrónico, al Director del Departamento, alegando y acreditando las razones que le asisten para no poder seguir el sistema de evaluación continua. El Director del Departamento al que se dirigió la solicitud, oído el profesorado responsable de la asignatura, resolverá la solicitud en el plazo de 10 días hábiles. Transcurrido dicho plazo sin que el estudiante haya recibido respuesta expresa por escrito, se entenderá estimada la solicitud. En caso de denegación, el estudiante podrá interponer, en el plazo de un mes, recurso de alzada ante el Rector, quien podrá delegar en el Decano o Director del Centro, según corresponda, agotando la vía administrativa. No obstante lo anterior, por causas excepcionales sobrevenidas y justificadas (motivos laborales, estado de salud, discapacidad, programas de movilidad, representación o cualquier otra circunstancia análoga), podrá solicitarse la evaluación única final fuera de los citados plazos, bajo el mismo procedimiento administrativo. (NCG112/3: Instrucción relativa a la aplicación del artículo 8.2). La normativa se puede consultar en http://secretariageneral.ugr.es/pages/normativa/fichasugr/ncg7121.

- La evaluación única final se realizará en un solo acto académico el mismo día y hora de la convocatoria oficial de examen ordinario para la asignatura (es decir, se realiza de forma conjunta). Dicha prueba (evaluada de 0 a 10) tendrá la misma estructura de un examen extraordinario (40-60%) teniendo en cuenta que las preguntas del "Examen de programación" (60%) deben suplir el trabajo realizado durante el curso mediante la "Colección de ejercicios resueltos" y garantizando que el alumno ha adquirido la totalidad de las competencias descritas en esta misma guía docente.

INFORMACIÓN ADICIONAL

- La asistencia a las clases teóricas no será obligatoria, aunque dicha asistencia, la participación activa en clase y la entrega de ejercicios planteados por el profesor se consideran muy importantes a la hora de entender la asignatura para poder abordar las prácticas semanales (que componen la "colección de ejercicios resueltos") así como el "examen de programación".

- La asistencia a las clases prácticas no será obligatoria, exceptuando las sesiones en las que el alumno decida entregar los ejercicios que ha resuelto para su evaluación. En cualquier caso, la asistencia (entendida como el trabajo continuo para resolver los ejercicios propuestos en presencia del profesor, así como el tiempo dedicado a la entrega de dichos ejercicios) junto con la participación activa en clase, se consideran muy importantes para poder superar las prácticas (que componen la "colección de ejercicios resueltos") así como el "examen de programación".

El programa de actividades (orientativo) se muestra a continuación.

PROGRAMA DE ACTIVIDADES orientativo

SEMESTRE 3º	Tema	ACTIVIDADES PRESENCIALES					ACTIVIDADES NO PRESENCIALES		
		Teoría (horas)	Clases prácticas de laboratorio y/o de informática (horas)		Exámenes (horas)	Contenidos	Estudio de teoría (horas)	Programación (horas)	Resolución de ejercicios (horas)
			Laboratorio (horas)	Seminarios (horas)					
SEMANA 1 13-15 sep									
SEMANA 2 18-22 sep	1-2	2h				- Introd./Fund. Inf.	1		
SEMANA 2 25sep-29sep	2-3	2h	1h(P1)	1h(S1)		- Fund. Inf/Leng. - SOs Introd. - SOs	1		
SEMANA 3 2-6 oct	3-4	2h	2h(P2)	1h(S2)		- Leng./Progr. - Entorno Python - 1ºs programas	1	4	4
SEMANA 4 9-13 oct	4	2h	3h(P3)			- Progr. - Estruc. control	1	4	
SEMANA 5 16-20 oct	4	2h	3h(P3)			- Progr. - Estruc. control	1	4	
SEMANA 6 23-27 oct	4	2h				- Progr.	1	2	4
SEMANA 7 30oct-3 nov	4	2h	1h(P3)-3h(P4)			- Progr. - Estruc. Control / Dat. Complejos	1	6	
SEMANA 8 6-10 nov	4-5	2h	4h(P4)			- Progr./Bases de Datos - Dat. Complejos	1	4	
SEMANA 9 13-17 nov	5	1h	1h(P4)-3h(P5)			- Bases de Datos - Dat. Complejos / Modular	1	4	4
SEMANA 10 20-24 nov	6	2h	1h(P5)-2h(P6)	1h(S3)		- I.A. y Bioquím. - Dif. e Integr. numérica - Modular/Dif. Int. Num	1	4	
SEMANA 11 27nov - 1dic	6-7	2h	1h(P6)-2h(P7)	1h(S4)		- I.A. y Bioquím. /Algs. Bioinsp. - Busqueda Inte. - Dif.Int.N/BDs	1	4	2
SEMANA 12 4-8 dic	7	1h				- Algs. Bioinsp.	1	2	2

SEMANA 13 11-15 dic	7	1h	3h(P8)	1h(S5)		- Algs. Bioinsp. - Biotools. - Biopython.		5	4
RESTO					2,5h	**Prueba tipo test Conoc. y Examen Programación:** **- Ordinario: (22Ene)** **- Extraordinario (13Feb)**	8	2	5
Total hs		**23h**	30h **35**	5h	**2,5h**		**20**	**45**	**25**