

Guía docente

270402 - PA1 - Programación y Algoritmia I

Última modificación: 10/07/2024

Unidad responsable: Facultad de Informática de Barcelona
Unidad que imparte: 723 - CS - Departamento de Ciencias de la Computación.
Titulación: GRADO EN INTELIGENCIA ARTIFICIAL (Plan 2021). (Asignatura obligatoria).
Curso: 2024 **Créditos ECTS:** 6.0 **Idiomas:** Catalán, Castellano

PROFESORADO

Profesorado responsable: JORDI DELGADO PIN
Otros: Primer quadrimestre:
JORDI DELGADO PIN - 11, 12

CAPACIDADES PREVIAS

Las definidas en el bachillerato científico-técnico o equivalentes

COMPETENCIAS DE LA TITULACIÓN A LAS QUE CONTRIBUYE LA ASIGNATURA

Específicas:

CE02. Dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para el tratamiento automático de la información por medio de sistemas computacionales y su aplicación para la resolución de problemas.
CE03. Identificar i aplicar els procediments algorítmics bàsics de les tecnologies informàtiques per a dissenyar solucions a problemes, analitzant la idoneïtat i complexitat dels algorismes proposats.
CE04. Diseñar y utilizar de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.
CE10. Analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.
CE12. Dominar los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la inteligencia artificial.
CE13. Evaluar la complejidad computacional de un problema, identificar estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

Genéricas:

CG2. Utilizar los conocimientos fundamentales y metodologías de trabajo sólidas adquiridos durante los estudios para adaptarse a los nuevos escenarios tecnológicos del futuro.
CG4. Razonar, analizando la realidad y diseñando algoritmos y formulaciones que la modelen. Identificar problemas y construir soluciones algorítmicas o matemáticas válidas, eventualmente nuevas, integrando el conocimiento multidisciplinar necesario, valorando distintas alternativas con espíritu crítico, justificando las decisiones tomadas, interpretando y sintetizando los resultados en el contexto del dominio de aplicación y estableciendo generalizaciones metodológicas a partir de aplicaciones concretas.
CG8. Observar un ejercicio ético de la profesión en todas sus facetas, aplicando criterios éticos en el diseño de sistemas, algoritmos, experimentos, utilización de datos, de acuerdo con los sistemas éticos recomendados por los organismos nacionales e internacionales, con especial énfasis en seguridad, robustez, privacidad, transparencia, trazabilidad, prevención de sesgos (de raza, género, religión, territorio, etc.) y respeto a los derechos humanos.

Transversales:

CT4. Trabajo en equipo. Ser capaz de trabajar como miembro de un equipo interdisciplinar, ya sea como un miembro más o realizando tareas de dirección, con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.
CT6. Aprendizaje autónomo. Detectar deficiencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar dicho conocimiento.

METODOLOGÍAS DOCENTES

La docencia de la asignatura está estructurada en clases de teoría y clases de laboratorio.

En las clases de teoría los profesores presentan los contenidos esenciales de la asignatura. En las clases de laboratorio se practican los contenidos de la asignatura (los presentados en clase y los adquiridos autónomamente) mediante la realización de problemas prácticos. Las clases de laboratorio serán una continuación de las clases teóricas, donde los conceptos nuevos se implementarán a medida que vayan apareciendo.

OBJETIVOS DE APRENDIZAJE DE LA ASIGNATURA

1. Conocer los conceptos más básicos de la programación con funciones (sencillas y de orden superior) y su utilización como herramientas para adecuar un lenguaje de programación al dominio del problema que se quiere resolver
2. Conocer el concepto de recursividad y cómo distinguir procesos iterativos y recursivos definidos con funciones recursivas
3. Empezar a entender el concepto de diseño del software
4. Poder trabajar con la idea de error en un programa y la utilización de excepciones para tratar estos errores
5. Conocer el concepto de contenedor y las situaciones en que es pertinente utilizarlos. Conocer el coste de su uso (informalmente)
6. Introducir al estudiante en el cálculo de la complejidad de los algoritmos
7. Involucrar al estudiante en el diseño e implementación en grupo de un problema sencillo

HORAS TOTALES DE DEDICACIÓN DEL ESTUDIANTE

Tipo	Horas	Porcentaje
Horas grupo pequeño	30,0	20.00
Horas aprendizaje autónomo	90,0	60.00
Horas grupo grande	30,0	20.00

Dedicación total: 150 h

CONTENIDOS

Construyendo Abstracciones con Funciones: Funciones (simples y de orden superior). Entornos. Control del flujo de ejecución.

Descripción:

Se introduce al estudiante en el mecanismo principal de solución de problemas a partir del diseño bottom-up de funciones que acerquen el lenguaje de programación al dominio del problema. Se estudian los mecanismos de definición de funciones, paso de parámetros, entorno y alcance, y las principales estructuras de control del flujo de ejecución (condicional, iterativas)

Construir Abstracciones con Funciones: Recursividad y ejemplos.

Descripción:

Se introduce el hecho de que una función se puede llamar a sí misma, la recursividad como mecanismo conceptual para resolver problemas y la idea de proceso. Funciones recursivas definen procesos iterativos y recursivos.

Construyendo Abstracciones con Funciones: Diseño. Tratamiento de errores con excepciones.

Descripción:

Se introduce el concepto de diseño de un programa para resolver un problema. Se introduce la idea de error en un programa y cómo tratarlos con el mecanismo de excepciones.

Construyendo abstracciones con Funciones: Ejemplos de resolución de problemas

Descripción:

Se refuerzan los conceptos introducidos hasta el momento con numerosos problemas de dificultad baja y media.

Construyendo Abstracciones con Funciones: Contenedores

Descripción:

Se empieza a trabajar con contenedores proporcionados por el lenguaje de programación: Secuencias, Listas, Diccionarios y con otros de implementados dentro del curso, como las Matrices o los Árboles.

Construyendo Abstracciones con Funciones: Mutabilidad. Iteradores y Generadores.

Descripción:

El concepto de mutabilidad se ve por primera vez relacionado con listas, y se discuten las ventajas e inconvenientes de tener estructuras de datos inmutables. Se introducen los iteradores y generadores como nuevas estructuras de control.

Introducción a la Complejidad de los Algoritmos

Descripción:

Se hace una pequeña introducción a la notación asintótica y el cálculo de la complejidad en caso peor de algunos algoritmos notables vistos durante el curso.

ACTIVIDADES

Construyendo Abstracciones con Funciones: Funciones y Estructuras de Control

Descripción:

Es necesario que el estudiante esté atento en clase y realice los ejercicios propuestos.

Objetivos específicos:

1, 2, 3, 4

Competencias relacionadas:

CE12. Dominar los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la inteligencia artificial.

CE03. Identificar i aplicar els procediments algorítmics bàsics de les tecnologies informàtiques per a dissenyar solucions a problemes, analitzant la idoneïtat i complexitat dels algorismes proposats.

CE10. Analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

CG4. Razonar, analizando la realidad y diseñando algoritmos y formulaciones que la modelen. Identificar problemas y construir soluciones algorítmicas o matemáticas válidas, eventualmente nuevas, integrando el conocimiento multidisciplinar necesario, valorando distintas alternativas con espíritu crítico, justificando las decisiones tomadas, interpretando y sintetizando los resultados en el contexto del dominio de aplicación y estableciendo generalizaciones metodológicas a partir de aplicaciones concretas.

CG2. Utilizar los conocimientos fundamentales y metodologías de trabajo sólidas adquiridos durante los estudios para adaptarse a los nuevos escenarios tecnológicos del futuro.

CT6. Aprendizaje autónomo. Detectar deficiencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar dicho conocimiento.

Dedicación: 52h

Aprendizaje autónomo: 30h

Grupo grande/Teoría: 10h

Grupo pequeño/Laboratorio: 12h

Construyendo Abstracciones con Funciones: Contenedores, Iteradores, Generadores.

Descripción:

Es necesario que el estudiante esté atento en clase y realice los ejercicios propuestos.

Objetivos específicos:

5, 7

Competencias relacionadas:

CE03. Identificar i aplicar els procediments algorítmics bàsics de les tecnologies informàtiques per a dissenyar solucions a problemes, analitzant la idoneïtat i complexitat dels algorismes proposats.

CE02. Dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para el tratamiento automático de la información por medio de sistemas computacionales y su aplicación para la resolución de problemas.

CE10. Analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

CE04. Diseñar y utilizar de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

CG4. Razonar, analizando la realidad y diseñando algoritmos y formulaciones que la modelen. Identificar problemas y construir soluciones algorítmicas o matemáticas válidas, eventualmente nuevas, integrando el conocimiento multidisciplinar necesario, valorando distintas alternativas con espíritu crítico, justificando las decisiones tomadas, interpretando y sintetizando los resultados en el contexto del dominio de aplicación y estableciendo generalizaciones metodológicas a partir de aplicaciones concretas.

CG8. Observar un ejercicio ético de la profesión en todas sus facetas, aplicando criterios éticos en el diseño de sistemas, algoritmos, experimentos, utilización de datos, de acuerdo con los sistemas éticos recomendados por los organismos nacionales e internacionales, con especial énfasis en seguridad, robustez, privacidad, transparencia, trazabilidad, prevención de sesgos (de raza, género, religión, territorio, etc.) y respeto a los derechos humanos.

CG2. Utilizar los conocimientos fundamentales y metodologías de trabajo sólidas adquiridos durante los estudios para adaptarse a los nuevos escenarios tecnológicos del futuro.

CT6. Aprendizaje autónomo. Detectar deficiencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar dicho conocimiento.

CT4. Trabajo en equipo. Ser capaz de trabajar como miembro de un equipo interdisciplinar, ya sea como un miembro más o realizando tareas de dirección, con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 56h

Aprendizaje autónomo: 30h

Grupo grande/Teoría: 12h

Grupo pequeño/Laboratorio: 14h

Complejidad de los Algoritmos

Descripción:

Es necesario que el estudiante esté atento en clase y realice los ejercicios propuestos.

Objetivos específicos:

6

Competencias relacionadas:

CE13. Evaluar la complejidad computacional de un problema, identificar estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

Dedicación: 17h

Aprendizaje autónomo: 10h

Grupo grande/Teoría: 3h

Grupo pequeño/Laboratorio: 4h

Examen Parcial

Objetivos específicos:

1, 2, 3, 4

Competencias relacionadas:

CE12. Dominar los principios fundamentales y modelos de la computación y saberlos aplicar para interpretar, seleccionar, valorar, modelar, y crear nuevos conceptos, teorías, usos y desarrollos tecnológicos relacionados con la inteligencia artificial.

CE03. Identificar i aplicar els procediments algorítmics bàsics de les tecnologies informàtiques per a dissenyar solucions a problemes, analitzant la idoneïtat i complexitat dels algorismes proposats.

CE10. Analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

CG4. Razonar, analizando la realidad y diseñando algoritmos y formulaciones que la modelen. Identificar problemas y construir soluciones algorítmicas o matemáticas válidas, eventualmente nuevas, integrando el conocimiento multidisciplinar necesario, valorando distintas alternativas con espíritu crítico, justificando las decisiones tomadas, interpretando y sintetizando los resultados en el contexto del dominio de aplicación y estableciendo generalizaciones metodológicas a partir de aplicaciones concretas.

CG2. Utilizar los conocimientos fundamentales y metodologías de trabajo sólidas adquiridos durante los estudios para adaptarse a los nuevos escenarios tecnológicos del futuro.

CT6. Aprendizaje autónomo. Detectar deficiencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar dicho conocimiento.

Dedicación: 12h

Aprendizaje autónomo: 10h

Actividades dirigidas: 2h

Examen Final

Objetivos específicos:

5, 6, 7

Competencias relacionadas:

CE03. Identificar i aplicar els procediments algorítmics bàsics de les tecnologies informàtiques per a dissenyar solucions a problemes, analitzant la idoneïtat i complexitat dels algorismes proposats.

CE02. Dominar los conceptos básicos de matemática discreta, lógica, algorítmica y complejidad computacional, y su aplicación para el tratamiento automático de la información por medio de sistemas computacionales y su aplicación para la resolución de problemas.

CE10. Analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados.

CE13. Evaluar la complejidad computacional de un problema, identificar estrategias algorítmicas que puedan conducir a su resolución y recomendar, desarrollar e implementar aquella que garantice el mejor rendimiento de acuerdo con los requisitos establecidos.

CE04. Diseñar y utilizar de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema.

CG4. Razonar, analizando la realidad y diseñando algoritmos y formulaciones que la modelen. Identificar problemas y construir soluciones algorítmicas o matemáticas válidas, eventualmente nuevas, integrando el conocimiento multidisciplinar necesario, valorando distintas alternativas con espíritu crítico, justificando las decisiones tomadas, interpretando y sintetizando los resultados en el contexto del dominio de aplicación y estableciendo generalizaciones metodológicas a partir de aplicaciones concretas.

CG8. Observar un ejercicio ético de la profesión en todas sus facetas, aplicando criterios éticos en el diseño de sistemas, algoritmos, experimentos, utilización de datos, de acuerdo con los sistemas éticos recomendados por los organismos nacionales e internacionales, con especial énfasis en seguridad, robustez, privacidad, transparencia, trazabilidad, prevención de sesgos (de raza, género, religión, territorio, etc.) y respeto a los derechos humanos.

CG2. Utilizar los conocimientos fundamentales y metodologías de trabajo sólidas adquiridos durante los estudios para adaptarse a los nuevos escenarios tecnológicos del futuro.

CT6. Aprendizaje autónomo. Detectar deficiencias en el propio conocimiento y superarlas mediante la reflexión crítica y la elección de la mejor actuación para ampliar dicho conocimiento.

CT4. Trabajo en equipo. Ser capaz de trabajar como miembro de un equipo interdisciplinar, ya sea como un miembro más o realizando tareas de dirección, con la finalidad de contribuir a desarrollar proyectos con pragmatismo y sentido de la responsabilidad, asumiendo compromisos teniendo en cuenta los recursos disponibles.

Dedicación: 13h

Aprendizaje autónomo: 10h

Actividades dirigidas: 3h

SISTEMA DE CALIFICACIÓN

El método de evaluación de la asignatura consistirá en dos pruebas de carácter teórico (T1 y T2), una a mediados de curso y otra al final y una práctica de tamaño pequeño-medio (Practica).

Entonces, el método de evaluación sería:

$$0.8 * \text{Teoría} + 0.2 * \text{Practica}$$

donde:

$$\text{Teoría: } \text{MAX}(T2, 0.5 * T1 + 0.5 * T2)$$

Competencia transversal "Trabajo en equipo":

Se evalúa usando una rúbrica simple en que el tutor de cada grupo puntúa los diferentes aspectos del trabajo en equipo de cada miembro de los grupos.



BIBLIOGRAFÍA

Básica:

- DeNero, John. Composing Programs.
- Abelson, Harold; Sussman, Gerald Jay; Sussman, Julie. Structure and interpretation of computer programs. 2nd ed. Cambridge [Massachusetts] [etc.]: MIT Press [etc.], 1996. ISBN 9780262011532.
- Lutz, Mark. Python pocket reference. 5th ed. Sebastopol, California: O'Reilly Media, Inc., 2014. ISBN 9781449356941.

RECURSOS

Enlace web:

- <http://www.composingprograms.com/>- <https://cs61a.org/>- <https://docs.python.org/3/reference/index.html>