UNIVERSIDADE DA CORUÑA

Guía Docente					
Datos Identificativos					2014/15
Asignatura (*)	Programación I			Código	614G01001
Titulación	Grao en Enxeñaría Informática				
Descriptores					
Ciclo	Período	Curso		Tipo	Créditos
Grao	1º cuadrimestre	Primeiro		Formación básica	6
Idioma	Castelán				
Prerrequisitos					
Departamento	Tecnoloxías da Información e as Comunicacións				
Coordinación	Garcia Martin, Esteban		Correo electrónico	esteban.garcia@udc.es	
Profesorado	Arcay Varela, Bernardino		Correo electrónico	bernardino.arcay@udc.es	
	Boveda alvarez, Maria del Carmen			carmen.boveda@udc.es	
	Castro Martinez, Alfonso			alfonso.castro@udc.es	
	Garcia Martin, Esteban			esteban.garcia@udc.es	
	Rabuñal Dopico, Juan Ramon			juan.rabunal@udc.es	
Web					
Descrición xeral					

Competencias da titulación	
Código	Competencias da titulación
A3	Capacidade para comprender e dominar os conceptos básicos de matemática discreta, lóxica, algorítmica e complexidade computacional e a súa aplicación para a resolución de problemas propios da enxeñaría.
A4	Coñecementos básicos sobre o uso e a programación dos ordenadores, sistemas operativos, bases de datos e programas informáticos con aplicación na enxeñaría.
A5	Coñecemento da estrutura, organización, funcionamento e interconexión dos sistemas informáticos, os fundamentos da súa programación e a súa aplicación para a resolución de problemas propios da enxeñaría.
A13	Coñecemento, deseño e utilización de forma eficiente dos tipos e estruturas de datos máis adecuados á resolución dun problema.
B1	Capacidade de resolución de problemas
B2	Traballo en equipo
B3	Capacidade de análise e síntese
B4	Capacidade para organizar e planificar
B5	Habilidades de xestión da información
B6	Toma de decisións
B7	Preocupación pola calidade
C3	Utilizar as ferramentas básicas das tecnoloxías da información e as comunicacións (TIC) necesarias para o exercicio da súa profesión e para a aprendizaxe ao longo da súa vida.
C4	Desenvolverse para o exercicio dunha cidadanía aberta, culta, crítica, comprometida, democrática e solidaria, capaz de analizar a realidade, diagnosticar problemas, formular e implantar solucións baseadas no coñecemento e orientadas ao ben común.
C6	Valorar criticamente o coñecemento, a tecnoloxía e a información dispoñible para resolver os problemas cos que deben enfrontarse.
C7	Asumir como profesional e cidadán a importancia da aprendizaxe ao longo da vida.
C8	Valorar a importancia que ten a investigación, a innovación e o desenvolvemento tecnolóxico no avance socioeconómico e cultural da sociedade.

Resultados da aprendizaxe	
Competencias de materia (Resultados de aprendizaxe)	Competencias da titulación

Conocer y comprender la importancia de los objetivos de la programación. Conocer los aspectos generales sobre los lenguajes y paradigmas de la programación. Conocer pseudocódigo y la sintaxis del lenguaje Pascal ISO10206 utilizado para describir algoritmos y programas. Conocer los pasos para la realización de un programa y sus principales componentes. Conocer los tipos de datos básicos usando Pascal ISO_10206. Conocer las estructuras de control de la programación estructurada y las diferencias entre ellas. Conocer todos los aspectos relacionados con la realización de funciones y procedimientos.	A4 A5		
Ser capaz de realizar el seguimiento de un algoritmo (en pseudocódigo) o programa (en Pascal ISO-10206), explicar qué realiza y encontrar posibles errores. Ser capaz de resolver pequeños algoritmos y programas. A partir del planteamiento de un problema de pequeña-mediana envergadura saber realizar el programa para resolverlo: Teniendo en cuenta los objetivos de la programación. eligiendo y utilizando los tipos y estructuras de datos adecuadas.Saber elegir y utilizar las estructuras de control y estructuras de datos convenientes.Realizar la descomposición adecuada e implementar las funciones y procedimientos necesarios correctamente.Utilizar un estilo de programación apropiado. Saber hacer buen uso de identificadores, los comentarios adecuados, el establecimiento de precondiciones y postcondiciones el buen diseño de las interfaces de procedimientos y funciones y saber hacer buen conocimiento de la parte del lenguaje que se explique.	A3 A5 A13	B1 B2 B3 B4 B5 B6 B7	C3
Aprendizaje autónomo Planificación de las actividades a desarrollar Capacidad de abstracción Toma de decisiónes Capacidad de iniciativa y participación			C3 C4 C6 C7 C8

Contidos	
Temas	**Subtemas**

1 CONCEPTOS BÁSICOS	1.1 Algoritmos 1.1.1 Representación de algoritmos 1.2 Programas 1.2.1 Tipos de programas 1.3 Lenguajes de programación 1.3.1 Una visión histórica 1.3.2 Clasificación de los lenguajes 1.3.3 Instrucciones más importantes 1.3.4 Propiedades de los lenguajes 1.4 Traductores 1.5 Descripción de los lenguajes 1.5.1 Notación BNF 1.5.2 Diagramas de Conway 1.6 Estructura de un programa 1.7 Elementos de un programa 1.7.1 Símbolos predefinidos 1.7.2 Símbolos especiales 1.7.3 Identificadores 1.7.4 Etiquetas 1.7.5 Comentarios 1.7.6 Directivas 1.7.7 Constantes 1.7.8 Números 1.7.9 Cadenas de caracteres 1.7.10 Variables: Declaración e iniciación 1.8 Salida e Entrada 1.8.1 Sentencias de salida 1.8.2 Sentencias de entrada 1.9 Tipos de datos y operadores 1.9.1 Tipos de datos 1.9.1.1 Concepto 1.9.1.2 Clases de tipos 1.9.1.3 Tipo entero 1.9.1.4 Tipo Real 1.9.1.5 Tipo Char 1.9.1.6 Tipo Boolean 1.9.1.7 Definición de tipos de usuario 1.9.1.8 Tipo enumerado 1.9.1.9 Tipo subrango 1.9.1.10 Tipos Anónimos 1.9.1.11 Compatibilidad de tipos 1.9.2 Operadores 1.9.2.1 Aritméticos

1.9.2.2 Relacionales

1.9.2.3 Lógicos

1.9.2.4 De conjunto

1.9.2.5 De cadena

1.9.2.6 Prioridad de operadores

1.9.2.7 Expresiones

2 Sentencias de control	2.1 Secuencial 2.2 Alternativa 2.2.1 La sentencia IF 2.2.2 La sentencia case 2.3 Repetitiva 2.3.1 Introducción 2.3.2 Variables asociadas a los bucles 2.3.3 Bucle WHILE 2.3.4 Ejemplos a realizar en clase 2.3.5 Bucle FOR 2.3.6 Bucle REPEAT 2.3.7 Equivalencia entre bucles 2.3.8 Ejemplos 2.3.9 Errores en los bucles 2.3.10 Diseño de bucles
3 Arquitectura de un programa	3.1 Procedimientos 3.1.1 Concepto 3.1.2 Tipos de procedimientos 3.1.3 Parámetros por valor y referencia 3.1.4 Parámetros protegidos 3.1.5 La pila de activación de procedimientos 3.1.6 Variables globales y locales: Alcance 3.1.7 Efectos laterales 3.2 Funciones 3.2.1 Concepto 3.2.2 Funciones predefinidas 3.2.3 Funciones de usuario 3.3 Recursividad 3.3.1 Naturaleza de la recursividad 3.3.2 Recursividad directa e indirecta. La directiva del lenguaje FORWARD. 3.3.3 Recursión infinita 3.3.4 Ejemplos

4 Estructuras simples de datos	4.1 Arrays 4.1.1 Tipo de dato ARRAY 4.1.2 Declaración de un Array 4.1.3 Arrays de más de una dimensión 4.1.4 Operaciones con Arrays 4.1.5 Arrays como parámetros 4.1.6 Funciones de tipo Array 4.1.7 Constantes de tipo Array 4.2 Registros 4.2.1 Tipo de dato registro 4.2.2 La sentencia with 4.2.3 Operaciones con registros 4.2.4 Registros variantes 4.2.5 Registros como parámetros 4.2.6 Constantes de tipo registro 4.3 Cadenas 4.3.1 Cadenas de longitud fija 4.3.2 Cadenas de longitud variable 4.4 Conjuntos 4.4.1 Operaciones y relaciones entre conjuntos 4.4.2 Procesamiento de conjuntos 4.5 Operaciones básicas sobre Arrays 4.6 Entrada/Salida

Planificación			
Metodoloxías / probas	Horas presenciais	Horas non presenciais / traballo autónomo	Horas totais
Sesión maxistral	30	30	60
Prácticas de laboratorio	20	50	70
Seminario	10	10	20
Atención personalizada	0		0
*Os datos que aparecen na táboa de planificación son de carácter orientativo, considerando a heteroxeneidade do alumnado			

Metodoloxías	
Metodoloxías	Descrición

Sesión maxistral	En las sesiones de teoría, el profesor describe los objetivos y los contenidos de la materia, para dar una visión particular del tema a tratar y relacionarlo con otros dentro de la asignatura Después se desarrolla el tema correspondiente en la forma de sesión magistral, ayudándose de las herramientas técnicas disponibles, haciendo hincapié en ciertas cuestiones en las que el alumno debe profundizar en su autoaprendizaje. El objetivo es que el alumno aprenda a algorimizar, utilizar las estructuras básiccas de datos y resolver sencillos problemas de programación. Se utilizará como lenguaje de codificación Pascal estandard Extendido ISO 10206
Prácticas de laboratorio	En las sesiones de prácticas el alumno realizará programas en papel para después codificarlo en Pascal Estandard Extendido ISO 10206, compilarlo, ejecutarlo y comprobar su nivel de corrección. Los enunciados de los programas se proporcionará con la suficiente antelación para que los alumnos puedan aprovechar mejor su tiempo. Es misión del profesor supervisar el código generado por el alumno para resolver dudar, corregir malos estilos de programación y corregir errores, contando con que el profesor no es un compilador que busca errores.
Seminario	Las sesiones de Seminario/Tutoría se utilizarán para acercarse aún más al alumno y detectar las lagunas que presenta en los temas vistos hasta la fecha. Se fomentará la participación activa de los alumnos, la realización de pequeños ejercicios para que estas sesiones sirvan de sonda y hacer ver al alumno a que conceptos, destrezas o conocimientos debe dedicarle más tiempo de estudio y/o práctica.

Atención personalizada	
Metodoloxías	**Descrición**
Sesión maxistral Prácticas de laboratorio Seminario	Tanto en las sesiones magistrales como en los laboratorios de prácticas y en las sesiones de seminario se llevará una atención personalizada del alumno, en distintos niveles según sea el tipo de clase, detectando el nivel de asimilación y comprensión de los temas explicados y las prácticas requeridas a implantar. En las sesiones de seminario en donde se puede llegar más alumno para conocer las lagunas que presente e indicarle el camino para cubrirlas. El tratamiento diferenciado para los alumnos que tengan matrícula a tiempo parcial será la que marque el departamento o en su defecto la Facultad.

Avaliación		
Metodoloxías	**Descrición**	**Cualificación**

Sesión maxistral	La nota de la asignatura será la suma de lo obtenido en la Evaluación Continua y lo obtenido en el Examen Final. La nota de evaluación continua se divide en dos partes: 1.- A la mitad del curso se realizará una prueba que valdrá 2 puntos. 2.- En la última semana del curso se realiza una prueba en el laboratorio utilizando ordenadores que valdrá 3 puntos. El examen final constará de tres ejercicios que el alumno tendrá que desrrollar en código y tendrá un valor de 5 puntos. En examen extraordinario de julio constará de tres problemas a desarrollar en código con un valor de 5 puntos, que se sumarán a lo obtenido en la evaluación continua. Para la evaluación en la oportunidad adelantada de diciembre se realizará un único examen de tres problemas a resolver en papel y puntuando sobre 10.	70
Prácticas de laboratorio	Como se indica en la parte de sesión magistral en la última semana del curso se realizará una prueba en el laboratorio usando ordenadores que tendrá un valor de 3 puntos sobre la nota total del curso.	30

Observacións avaliación
La nota final vendrá dada por la nota obtenida por Evaluación Contínua y la obtenida en el examen final   El examen final constará de tres problemas a codificar.

Fontes de información	
Bibliografía básica	- ISO (1990). Extended Pascasl ISO 10206. ISO - Leestma, S e Nyhoff, L.. (1999). Programación en Pascal. Madrid Prentice Hall - Carmen Bóveda, Esteban García, Alejandra Martínez (2014). Programación estructurada en un lenguaje didáctico y estándar. La Coruña , Reprografia del Noroeste - Valls, J. e Camacho, D. (2004). Programación estructurado y algoritmos en Pascal. Madrid Prentice Hall
Bibliografía complementaria	- Grogono, P (). Programación en Pascal. Addison-Wesley I

Recomendacións
Materias que se recomenda ter cursado previamente
Materias que se recomenda cursar simultaneamente
Materias que continúan o temario
Observacións

El alumno debe tener en cuenta que debe realizar una labor autodidacta muy importante. Siga el siguiente esquema: Leer, atender, comprender, preguntar, estudiar y practicar.

Leer: Lea el tema a tratar antes de asistir a las sesiones teóricas.Aunque le parezca raro ES MUY IMPORTANTE.Atender: Atienda en clase, no se duerma, no se dedique sólo a tomar apuntes.
Comprender: Comprenda lo que se le dice en las sesiones de teoría, y si no
pregunte.Preguntar: Pregunte todo lo que no comprenda, tiene ese derecho,
Estudiar: para retener lo comprendido.
Practicar: Haga muchos programas, los que se le pidan y otros por su cuenta, tanto en papel como en el ordenador.

Programación es una asignatura que no se puede aprender con un empacho de estudio en dos días. El alumno debe ir madurando los conceptos, y hacer sobre el papel y en la máquina muchos programas.

Es una asignatura que, por medio del sistema de evalución continua, se puede aprobar sin más que seguir el ritmo de las distintas clases de sesiones teóricas y hacer caso a las indicaciones particulares de refuerzo de estudio que le señale el profesor.

(*)A Guía docente é o documento onde se visualiza a proposta académica da UDC. Este documento é público e non se pode modificar, salvo casos excepcionais baixo a revisión do órgano competente dacordo coa normativa vixente que establece o proceso de elaboración de guías