

GUÍA DOCENTE DE

Curso 2011-2012

TITULACION

Grado en Ingeniería de Computadores

GUIA DOCENTE DE LA ASIGNATURA

Lenguajes de Programación

Profesores

Nombre y apellidos:

Francisco Gortázar Bellas

Raúl Cabido Valladolid

Coordinador/a de la asignatura:

Raúl Cabido Valladolid

I.- Identificación de la asignatura

| | |
|-----------------------------|---|
| Tipo | Obligatoria |
| Materia | Lenguajes de Programación |
| Período de impartición | Segundo semestre |
| Nº Créditos | 6 |
| Idioma en el que se imparte | Español |
| Departamento | Ciencias de la Computación |
| Asignaturas llave | - |
| Tasa de éxito | Este dato será incluido por el Vicerrectorado de Profesorado, Titulaciones, Ordenación Académica, Coordinación y Campus |

II.- Presentación

El objetivo de la asignatura Lenguajes de Programación es presentar una panorámica general de las diferentes tendencias en los lenguajes de programación modernos que no quedan cubiertas en otras asignaturas de la titulación.

La asignatura se estructura en tres partes claramente diferenciadas. En la primera parte se estudian los conceptos de la programación orientada a objetos con C++ y el desarrollo basado en plantillas. En la segunda parte se presentan los conceptos de la programación con lenguajes dinámicos. En la tercera y última parte se introduce el paradigma de la programación funcional.

La base de la programación orientada a objetos con C++ es el paradigma de la orientación a objetos que se explica en la asignatura de Programación Orientada a Objetos. Por tanto, se presupone que el alumno ha superado esta asignatura.

Además, para la realización de las tres partes de la asignatura se requieren conocimientos de programación básicos, así como conocimientos de algoritmia, de forma que se le suponen al alumno superadas las asignaturas de Introducción a la Programación, Estructuras de Datos y Fundamentos de Diseño de Software.

III.- Competencias

| | |
|----------------------------|--|
| Competencias transversales | |
| G4 | Capacidad para definir, evaluar y seleccionar plataformas hardware y software para el desarrollo y la ejecución de sistemas, servicios y aplicaciones informáticas |
| G5 | Capacidad para concebir, desarrollar y mantener sistemas, servicios y aplicaciones informáticas empleando los métodos de la ingeniería del software como instrumento para el aseguramiento de su calidad |
| G8 | Conocimiento de las materias básicas y tecnologías, que capaciten para el aprendizaje y desarrollo de nuevos métodos y tecnologías, así como las que les doten de una gran versatilidad para adaptarse a nuevas situaciones. |

| | |
|--------------------------|--|
| G9 | Capacidad para resolver problemas con iniciativa, toma de decisiones, autonomía y creatividad. Capacidad para saber comunicar y transmitir los conocimientos, habilidades y destrezas de la profesión de Ingeniero Técnico en Informática. |
| B4 | Conocimientos básicos sobre el uso y programación de los ordenadores, sistemas operativos, bases de datos y programas informáticos con aplicación en ingeniería. |
| C6 | Conocimiento y aplicación de los procedimientos algorítmicos básicos de las tecnologías informáticas para diseñar soluciones a problemas, analizando la idoneidad y complejidad de los algoritmos propuestos. |
| C7 | Conocimiento, diseño y utilización de forma eficiente los tipos y estructuras de datos más adecuados a la resolución de un problema. |
| C8 | Capacidad para analizar, diseñar, construir y mantener aplicaciones de forma robusta, segura y eficiente, eligiendo el paradigma y los lenguajes de programación más adecuados. |
| C14 | Conocimiento y aplicación de los principios fundamentales y técnicas básicas de la programación paralela, concurrente, distribuida y de tiempo real. |
| C15 | Conocimiento y aplicación de los principios fundamentales y técnicas básicas de los sistemas inteligentes y su aplicación práctica. |
| Competencias específicas | |
| E3 | Capacidad de analizar y evaluar arquitecturas de computadores, incluyendo plataformas paralelas y distribuidas, así como desarrollar y optimizar software para las mismas. |
| E4 | Capacidad de diseñar e implementar software de sistema y de comunicaciones. |

IV.- Contenido

IV. A. Temario de la asignatura

| Bloque temático | Tema | Apartados |
|----------------------|----------------------|----------------------------------|
| I.- Programación C++ | Tema 1. Introducción | Programación orientada a objetos |



| | | |
|------------------------------|--|--|
| | | con C++. Herencia. Polimorfismo. Entrada/salida. |
| | Tema 2. Desarrollo basado en plantillas | Introducción a la programación basada en plantillas. Introducción a STL. Manejo de colecciones. Iteradores. |
| | Tema 3. Aspectos avanzados | Desarrollo de algoritmos sobre colecciones. Functors. Otros aspectos avanzados. |
| II.- Lenguajes Dinámicos | Tema 1. Introducción | Tipado dinámico y tipado estático. Lenguajes Dinámicos, portabilidad y ciclo de desarrollo. Máquinas virtuales, intérpretes y compiladores |
| | Tema 2. Introducción al lenguaje Ruby | Sintaxis básica de Ruby. Tipos de datos. Programación orientada a objetos con Ruby. |
| | Tema 3. Aspectos avanzados | Listas y diccionarios. Bloques e iteradores. Manejo de cadenas de caracteres y expresiones regulares. |
| III.- Programación Funcional | Tema 1. Introducción | Bases de la programación funcional. Evolución. Concepto de función. Orden de evaluación de las expresiones. |
| | Tema 2. Introducción al lenguaje Haskell | Tipos de datos. Funciones. Expresiones sencillas. Tuplas. Expresiones condicionales. Recursividad. Ajuste de patrones. Listas. Polimorfismo. |
| | Tema 3. Aspectos avanzados | Tipos de datos definidos por el programador, funciones de orden superior, mónadas. |

IV. B. Actividades obligatorias (evaluables):

1. Lecturas

| | |
|--|--|
| | |
| | |
| | |

2. Prácticas

| |
|--|
| Realización de una práctica evaluable por cada módulo (Práctica I, Práctica II, Práctica III). |
| |
| |

3. Laboratorios

| |
|--|
| |
| |
| |

4. Clínicos

| |
|--|
| |
| |
| |

5. Otras

| |
|--|
| |
| |
| |

V.- Tiempo de trabajo

| | |
|--|----------------|
| Clases teóricas | 20 |
| Clases prácticas/de resolución de problemas, casos, etc. | 0 |
| Prácticas en laboratorios tecnológicos, clínicos, etc. | 20 |
| Realización de pruebas | 8 |
| Tutorías académicas | 12 |
| Actividades relacionadas: jornadas, seminarios, etc. | 0 |
| Preparación de clases teóricas | 35 |
| Preparación de clases prácticas/problemas/casos | 35 |
| Preparación de pruebas | 20 |
| Total de horas de trabajo del estudiante | a+b+c=d |

VI.- Metodología y plan de trabajo

Clases teóricas

| Periodo | Contenidos |
|------------|------------|
| Semana 1 | Tema I.1 |
| Semanas 2 | Tema I.2 |
| Semana 3-4 | Tema I.3 |
| Semana 5 | Tema II.1 |
| Semana 5-6 | Tema II. 2 |



| | |
|--------------|------------|
| Semana 7 | Tema II.3 |
| Semana 8 | Tema II.3 |
| Semana 9 | Tema III.1 |
| Semanas 9-10 | Tema III.2 |
| Semana 11 | Tema III.3 |
| Semana 12 | Tema III.3 |

Prácticas/de resolución de problemas, casos, etc.

| Periodo | Contenidos |
|---------|------------|
| | |
| | |

Laboratorios

| Periodo | Contenidos |
|--------------|------------|
| Semana 1 | Tema I.1 |
| Semana 2 | Tema I.2 |
| Semana 3-4 | Tema I.3 |
| Semanas 5-6 | Tema II.2 |
| Semana 7 | Tema II.3 |
| Semana 8 | Tema II.3 |
| Semanas 9-10 | Tema III.2 |
| Semana 11 | Tema III.3 |
| Semana 12 | Tema III.3 |

Seminarios y trabajos colectivos



| | |
|---------|------------|
| Periodo | Contenidos |
| | |
| | |

Tutorías académicas

| | |
|--------------|---------------------|
| Periodo | |
| Semanas 1-12 | Tutorías académicas |

Otras actividades

| | |
|---------|--|
| Periodo | |
| | |
| | |

Pruebas

| | |
|-----------|----------------------------|
| Fecha | Contenidos |
| Semana 8 | Defensa de la práctica I |
| Semana 12 | Defensa de la práctica II |
| Semana 14 | Defensa de la práctica III |

VII.- Métodos de evaluación

VII. A. Ponderación para la evaluación continua

| Actividad evaluadora | Tipo | Ponderación | Periodo | Contenido |
|----------------------|--------------------------------------|-------------|---------|-----------|
| Prácticas fuera del | <input type="checkbox"/> Acumulativa | | | |



| | | | | | |
|---------------------------|---|---|--|-----|--------|
| aula: | | | | | |
| | <input checked="" type="checkbox"/> Resolución de problemas | <input checked="" type="checkbox"/> Liberatoria Puntuación mínima (de 1 a 10): 5 | <input checked="" type="checkbox"/> Reevaluable <input type="checkbox"/> No reevaluable | 34% | Sem 8 |
| Prácticas fuera del aula: | <input type="checkbox"/> Acumulativa | | | | |
| | <input checked="" type="checkbox"/> Resolución de problemas | <input checked="" type="checkbox"/> Liberatoria Puntuación mínima (de 1 a 10): 5 | <input checked="" type="checkbox"/> Reevaluable <input type="checkbox"/> No reevaluable | 33% | Sem 12 |
| Prácticas fuera del aula: | <input type="checkbox"/> Acumulativa | | | | |
| | <input checked="" type="checkbox"/> Resolución de problemas | <input checked="" type="checkbox"/> Liberatoria Puntuación mínima (de 1 a 10): 5 | <input checked="" type="checkbox"/> Reevaluable <input type="checkbox"/> No reevaluable | 33% | Sem 13 |
| Total | | | 100% | | |

Se considerará que el estudiante ha adquirido las competencias evaluadas en cada prueba práctica si su calificación en dicha prueba es mayor o igual a 5 sobre 10 puntos. En caso contrario se considera que el estudiante no ha adquirido dichas competencias. Las competencias evaluadas en la prueba práctica son reevaluables en convocatoria extraordinaria (Junio).

En caso de detectarse copias entre alumnos, el conjunto de la asignatura estará suspenso hasta la siguiente convocatoria. Todos los alumnos involucrados (tanto los que copian como los que son copiados) tendrán la asignatura suspenso hasta la siguiente convocatoria.

VII. B. Ponderación para la evaluación de alumnos a tiempo parcial

Para que un alumno pueda optar a esta evaluación, tendrá que obtener la "Dispensa Académica" para la asignatura, que habrá solicitado al Decano o Director/a del Centro que imparte su titulación.

La "Dispensa Académica" no excluye de la evaluación continua. Dicha evaluación se acomodará por el profesor, asistido por el coordinador de grado, estableciéndose la adaptación curricular según las características de cada caso concreto.

VII. C. Revisión de las pruebas de evaluación.

La revisión de las pruebas de evaluación se llevará a cabo en clase o en el horario de tutorías de la asignatura.

VIII.- Recursos y materiales didácticos

General

| | |
|-----------|---|
| Título | El lenguaje de programación C++ |
| Autor | Bjarne Stroustrup |
| Editorial | Addison-Wesley |
| Título | Generic programming and the STL. Using and extending the C++ standard template library. |
| Autor | Matthew H. Austern |
| Editorial | Addison-Wesley |
| Título | Beginning Ruby |
| Autor | Peter Cooper |
| Editorial | Apress |
| Título | Haskell. The craft of functional programming |
| Autor | Simon Thompson |
| Editorial | Addison-Wesley |

Complementaria

| | |
|-----------|--|
| Título | Beyond the C++ standard library: an introduction to boost |
| Autor | Bjorn Karlsson |
| Editorial | Addison-Wesley |
| Título | Everyday scripting with Ruby |
| Autor | Brian Marick |
| Editorial | The Pragmatic Bookshelf |
| Título | Programming Ruby (http://www.ruby-doc.org/docs/ProgrammingRuby/) |
| Autor | David Thomas, Andrew Hunt |
| Editorial | Addison-Wesley |

Direcciones web de interés

| |
|---|
| Boost: http://boost.org |
| C++: http://www2.research.att.com/~bs/C++.html |
| Lenguaje de programación Ruby: http://www.ruby-lang.org/es/ |
| Documentación del núcleo de Ruby: http://www.ruby-doc.org/core/ |
| Haskell: http://www.haskell.org/ |
| EclipseGavab: http://code.sidelab.es/projects/eclipsegavab/wiki/ |

IX.- Profesorado

| | |
|---------------------|-------------------------|
| Nombre y apellidos | Raúl Cabido Valladolid |
| Horario de tutorías | Se publicarán más tarde |

| | |
|--|--|
| académicas | |
| Correo electrónico | Raul.cabido@urjc.es |
| Departamento/área de conocimiento | Ciencias de la computación |
| Categoría | Profesor Ayudante Doctor |
| Titulación Académica | Doctor Ingeniero Informático |
| Experiencia Docente (Indicar la antigüedad en el área y en la asignatura. Incluir tramos de docencia.) | Profesor de la Universidad Rey Juan Carlos desde febrero de 2007. Ha impartido asignaturas de grado y postgrado, incluyendo asignaturas de programación en C y C++, estructuras de datos, visión dinámica y tratamiento digital de imagen. |
| Experiencia profesional (Indicar la actividad profesional y la antigüedad en la misma) | Ha sido becario de investigación de la Universidad Rey Juan Carlos (2005-2007) y ha participado en dos proyectos de transferencia tecnológica con Endesa y Mediapro. Tiene más de 10 años de experiencia en desarrollo C y tecnologías relacionadas (C++, CUDA, CG, ...) |

| | |
|--|---|
| Nombre y apellidos | Francisco Gortázar Bellas |
| Horario de tutorías académicas | Se publicarán más tarde |
| Correo electrónico | francisco.gortazar@urjc.es |
| Departamento/área de conocimiento | Ciencias de la computación |
| Categoría | Profesor Ayudante Doctor |
| Titulación Académica | Doctor Ingeniero Informático |
| Experiencia Docente (Indicar la antigüedad en el área y en la asignatura. Incluir tramos de docencia.) | <p>Profesor de la Universidad Rey Juan Carlos desde octubre de 2003. Ha impartido asignaturas de grado y postgrado, tanto troncales como obligatorias, incluyendo asignaturas de compiladores, paradigmas de programación, optimización, ...</p> <p>Tiene cinco años de experiencia impartiendo docencia de programación con lenguajes dinámicos y programación funcional.</p> <p>Ha obtenido en el año 2011 la evaluación positiva del programa Docencia de la URJC.</p> |



| | |
|---|--|
| Experiencia profesional (Indicar la actividad profesional y la antigüedad en la misma) | Tiene un año de experiencia profesional en la empresa Investigación y Programas, S.A.. Además ha sido becario del programa de doctorado de la Universidad Rey Juan Carlos. |
|---|--|