

# GENERACIÓN AUTOMÁTICA DE CÓDIGO

Curso 2013/2014

(Código: 3110501-)

## 1. PRESENTACIÓN

La estructura modular del programa de la asignatura Generación Automática de Código ofrece una formación teórica de base y una profundización de algunas temáticas específicas que se refieren a las diferentes alternativas tecnológicas disponibles y propuestas para la creación de sistemas informáticos a partir de mecanismos de especificación diversos y que permiten la eliminación o reducción a la mínima tarea de la fase de codificación.

## 2. CONTEXTUALIZACIÓN

La creación de mecanismos que permitan la generación de sistemas informáticos sin la necesidad de un proceso de codificación exhaustivo es uno de los objetivos fundamentales de diferentes líneas de investigación y técnicas de producción de software existentes. La eliminación de esta fase creativa de la producción de software se motiva por dos razones fundamentales: la productividad y la calidad. La disminución o desaparición de una fase de codificación asegura una capacidad de producir software siguiendo las líneas maestras de la producción industrial en línea: si somos capaces de introducir en un mecanismo la suficiente información de especificación y definición obtendremos al final de la cadena de producción el software acabado. Además esta automatización de la producción de software permitirá establecer el estándar de calidad que se deseé en la medida que los generadores de código serán elementos definidos, controlados y medibles en cuanto a sus resultados.

La asignatura se encuentra enmarcada en la materia de "Ingeniería del desarrollo de software" del itinerario de "Ingeniería de Software" del Master. En cuanto a las competencias específicas de la asignatura que se alcanzan al superar la misma son básicamente dos. Una respecto al conocimiento de los mecanismos, las técnicas y las herramientas de transformación y generación automática de código y la segunda respecto a cómo construir un sistema de transformación de código o como plantear, diseñar, construir y aplicar una herramienta de generación de código.

Además y respecto a las competencias genéricas incluidas en los objetivos del Master, esta asignatura hace hincapié especial en que los estudiantes sean capaces de realizar un análisis crítico, una evaluación y una síntesis de ideas nuevas y complejas.

## 3. REQUISITOS PREVIOS RECOMENDABLES

La asignatura no requiere de requisitos previos específicos, si bien para su adecuado seguimiento y aprovechamiento se precisan conocimientos a nivel de grado universitario o ingeniería en Informática.

## 4. RESULTADOS DE APRENDIZAJE

Al finalizar el curso el alumno será capaz de:

- Reconocer la importancia del uso de los generadores de código en la producción moderna de software.
- Conocer y utilizar las principales técnicas de generación automática de código.

- Conocer y utilizar alguna de las principales herramientas de generación automática de código.
- Analizar, seleccionar y aplicar la técnica de generación automática de código más adecuada según el tipo de problema a resolver.
- Plantear, diseñar y construir un sistema para la transformación de código.
- Plantear, diseñar y construir un sistema de generación automática de código.

## 5. CONTENIDOS DE LA ASIGNATURA

Módulo 1. Conceptos Generales sobre la Automatización de código

- Introducción a la GAC (Generación automática de Código)
- Representaciones de lenguajes
- Herramientas para la GAC

Módulo 2. GAC dirigida por el objetivo

- Generadores de interfaces
- Generadores de documentos
- Generadores de pruebas
- Generadores de gestión BBDD
- Generadores de servicios web
- Generadores de reglas de negocio

Módulo 3. GAC dirigida por el paradigma

- Generadores orientados por modelos. MDA y MDD
- Generadores orientados a los aspectos
- Intentional Programming
- Generación multiparadigmática

## 6. EQUIPO DOCENTE

- [ISMAEL ABAD CARDIEL](#)
- [JAVIER ARELLANO ALAMEDA](#)
- [EUGENIO ARELLANO ALAMEDA](#)

## 7. METODOLOGÍA



Desde el punto de vista metodológico las unidades integradas en este curso, si bien presentan los contenidos secuenciados de manera progresiva, tienen un carácter modular. Es decir, son unidades independientes organizadas de manera que tanto el docente como el alumno puedan utilizarlas en situaciones diferentes.

Por otra parte, el curso se adapta a la modalidad de autoaprendizaje, propia de los alumnos a distancia, a través de medios de apoyo informáticos. Además, se adapta al crédito ECTS a través de la atención al desarrollo del estudio del alumno:

- pautas de planificación incluidas en el plan de trabajo
- seguimiento del trabajo del alumno mediante tutorías individuales y personalizadas
- virtualización del curso: plataforma digital y utilización de los foros de comunicación
- realización de tareas de autoevaluación y evaluación continuadas.

## 8. BIBLIOGRAFÍA BÁSICA

ISBN(13): 9781930110977  
Título: CODE GENERATION IN ACTION (2003)  
Autor/es: Jack Herrington ;  
Editorial: MANNING

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

## 9. BIBLIOGRAFÍA COMPLEMENTARIA

ISBN(13): 9780201824674  
Título: MULTI-PARADIGM DESIGN FOR C++ (1998)  
Autor/es: James O. Coplien ;  
Editorial: ADDISON-WESLEY PU. CO.

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9780321617613  
Título: PRACTICAL CODE GENERATION IN .NET : COVERING VISUAL STUDIO 2005, 2008, AND 2010 (2010)  
Autor/es: Peter Vogel ;  
Editorial: : ADDISON-WESLEY

Buscarlo en librería virtual UNED



Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9780321712943

Título: DOMAIN-SPECIFIC LANGUAGES (2010)

Autor/es: Fowler, Martin ;

Editorial: ADDISON-WESLEY

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

ISBN(13): 9780471202844

Título: SOFTWARE FACTORIES : ASSEMBLING APPLICATIONS WITH PATTERNS, MODELS, FRAMEWORKS, AND TOOLS (2004)

Autor/es: Jack Greenfield ;

Editorial: JOHN WILLEY & SONS

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

## 10. RECURSOS DE APOYO AL ESTUDIO

La documentación necesaria para el seguimiento del curso y el resto del material didáctico que el alumno necesite durante el curso (ejercicios, guía de prácticas vía Internet, herramientas, plantillas, etc.) serán enviados o puestos a disposición del alumno según lo vaya necesitando a lo largo del mismo.

## 11. TUTORIZACIÓN Y SEGUIMIENTO

El proceso de la tutorización y seguimiento de los aprendizajes es continuo a partir de la comunicación de alumnos y los profesores a través de los diferentes canales y de los ejercicios en programación planificados a lo largo del curso. Además los alumnos podrán en todo momento contactar con los profesores vía correo electrónico o telefónicamente durante las horas de tutoría.

Horario: Lunes de 16 a 20 horas.

Prof. Ismael Abad Cardiel - Tel. 913988654 - email: [iabad@issi.uned.es](mailto:iabad@issi.uned.es)



Viernes de 16 a 20 horas.

Prof. Javier Arellano Alameda - Tel. 913988735 - email: [javier@issi.uned.es](mailto:javier@issi.uned.es)

Jueves de 9 a 13 horas.

Prof. Magdalena Arcilla Cobián

## 12.EVALUACIÓN DE LOS APRENDIZAJES

La metodología que se ha diseñado para el curso permite un seguimiento y una evaluación continua e individualizada de cada alumno, atendiendo y ponderando en cada caso su trabajo.

Asimismo, la calificación final del curso tendrá en cuenta los trabajos realizados por el alumno y su participación en el mismo, de acuerdo con los siguientes criterios:

1.- Realización de ejercicios de contenido en un 60%. Siguiendo una de las dos opciones disponibles a elegir por parte del alumno:

1.1 Trabajos de evaluación continua (6 trabajos de temario)

1.2 Trabajos de fin de módulo (3 trabajos, uno por cada módulo de la asignatura)

2.- Realización del trabajo final en un 40% orientado a la investigación y vinculado a la generación automática de código orientado al paradigma.

## 13.COLABORADORES DOCENTES

Véase equipo docente.

