

17-18

MÁSTER UNIVERSITARIO EN  
INVESTIGACIÓN EN INGENIERÍA DE  
SOFTWARE Y SISTEMAS  
INFORMÁTICOS

# GUÍA DE ESTUDIO PÚBLICA



## DESARROLLO DE LÍNEAS DE PRODUCTO SOFTWARE MEDIANTE UN ENFOQUE GENERATIVO

CÓDIGO 31105043



Ámbito: GUJ - La autenticidad, validez e integridad de este documento puede ser verificada mediante el "Código Seguro de Verificación (CSV)" en la dirección <https://sede.uned.es/valida/>



1EA9CC9654B904873CDF31E823EBC5D4

**17-18**

**DESARROLLO DE LÍNEAS DE PRODUCTO  
SOFTWARE MEDIANTE UN ENFOQUE  
GENERATIVO  
CÓDIGO 31105043**

# ÍNDICE

PRESENTACIÓN Y CONTEXTUALIZACIÓN  
REQUISITOS Y/O RECOMENDACIONES PARA CURSAR ESTA ASIGNATURA  
EQUIPO DOCENTE  
HORARIO DE ATENCIÓN AL ESTUDIANTE  
COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE  
RESULTADOS DE APRENDIZAJE  
CONTENIDOS  
METODOLOGÍA  
SISTEMA DE EVALUACIÓN  
BIBLIOGRAFÍA BÁSICA  
BIBLIOGRAFÍA COMPLEMENTARIA  
RECURSOS DE APOYO Y WEBGRAFÍA



Nombre de la asignatura	DESARROLLO DE LÍNEAS DE PRODUCTO SOFTWARE MEDIANTE UN ENFOQUE GENERATIVO
Código	31105043
Curso académico	2017/2018
Títulos en que se imparte	MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INGENIERÍA DE SOFTWARE Y SISTEMAS INFORMÁTICOS
Tipo	CONTENIDOS
Nº ETCS	9
Horas	225.0
Periodo	ANUAL
Idiomas en que se imparte	CASTELLANO

## PRESENTACIÓN Y CONTEXTUALIZACIÓN

Aunque algunas estimaciones realizadas en los años 80, pronosticaban que el 60% de cualquier aplicación informática se desarrollaría ensamblando componentes reutilizables, el nivel de reutilización alcanzado hoy día es bastante inferior. Muchos autores consideran que este fracaso se debe a que la mayoría de los procesos de desarrollo de software, ya sean formales o ágiles, persiguen la construcción de productos aislados. Al no disponerse de contextos suficientemente amplios como para detectar con precisión qué elementos son reutilizables y cuáles son las situaciones donde puede sacarse más partido a la reutilización, se desemboca en una reutilización oportunista del software. Para que la reutilización del software fuera sistemática, los procesos de desarrollo deberían abordar la construcción colectiva de familias de productos relacionados por un dominio.

Otros autores han llegado a conclusiones similares al tratar de aplicar en la fabricación de software los principios de economía de escala y de alcance, comúnmente utilizados en la industria para reducir los costes y tiempos de fabricación y mejorar la calidad de los productos. La economía de escala se refiere a la fabricación de múltiples unidades de un mismo producto. Cuanto más se produce, menores son los costes. Se logra por diversas causas: reparto de los costes fijos entre más unidades producidas (disminución del coste medio), *rappel* sobre compras, mejora tecnológica, incremento de la racionalidad en el trabajo (especialización y división del trabajo)... La economía de alcance se da en la fabricación colectiva de productos similares. Se consigue principalmente porque los problemas comunes en la fabricación de los diversos productos se resuelven una sola vez. La fabricación industrial de un producto consta fundamentalmente de dos etapas: (i) la fase de desarrollo, donde se crean el diseño del producto y unos pocos prototipos para la validación del diseño; y (ii) la fase de producción, donde se crean de forma masiva instancias del producto. La economía de escala ocurre sobre todo durante la fase de producción. La naturaleza esencialmente lógica del software hace que los costes se concentren en la etapa de desarrollo (el coste de producir las copias de un sistema informático es despreciable comparado con el coste de desarrollo del sistema) y, por tanto, sea la economía de alcance el principio más aplicable en la fabricación de software.

En resumen, el desarrollo de familias de productos, frente a la construcción individual de productos aislados, es un paso decisivo hacia la reutilización sistemática de software y la obtención de economía de alcance. Esta asignatura cubre las distintas fases de desarrollo



de una familia de productos software.

## REQUISITOS Y/O RECOMENDACIONES PARA CURSAR ESTA ASIGNATURA

La formación previa que deberían tener los alumnos para el adecuado seguimiento de esta asignatura son los propios de ingreso al posgrado, haciendo especial recomendación en conocimientos de ingeniería de software, lenguajes de programación y compiladores. Usando como referencia el GRADO EN INGENIERÍA INFORMÁTICA impartido por la UNED, el conocimiento aconsejable es el correspondiente a las asignaturas:

- Fundamentos de Programación
- Programación Orientada a Objetos
- Autómatas, Gramáticas y Lenguajes
- Teoría de los Lenguajes de Programación
- Introducción a la Ingeniería de Software
- Diseño del Software

## EQUIPO DOCENTE

Nombre y Apellidos  
Correo Electrónico  
Teléfono  
Facultad  
Departamento

RUBEN HERADIO GIL  
rheradio@issi.uned.es  
91398-8242  
ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA  
ING.DE SOFTWARE Y SISTEMAS INFORMÁTICOS

## HORARIO DE ATENCIÓN AL ESTUDIANTE

La tutorización de los alumnos se llevará a cabo fundamentalmente a través de correo electrónico y consultas telefónicas los Jueves de 16:00 a 20:00:

- *Rubén Heradio Gil: rheradio@issi.uned.es, 913988242, horario jueves de 10 a 14 h.*

## COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE

### RESULTADOS DE APRENDIZAJE

Los resultados de aprendizaje que se espera alcanzar con esta asignatura por parte del estudiante son:

- Comprender el impacto que tienen los conceptos de reutilización y abstracción en la producción de software (competencias CG1, CED1, CED5)



- Aprender los principios metodológicos que guían el desarrollo de una línea de productos software, es decir, los fundamentos de la ingeniería de dominio e ingeniería de aplicación (competencias CG7, CED2)
- Ser capaz de modelar mediante un diagrama de características el dominio de una línea de productos software (competencias CED3, CEP1)
- Conocer distintos enfoques para desarrollar una familia de productos, distinguiendo (i) sus puntos fuertes y débiles, y (ii) las herramientas informáticas que los soportan (competencias CG2, CG3, CED6, CEP2, CEP3)
- Ser capaz de implementar una línea de productos software mediante el lenguaje de programación Ruby (competencias CG5, CG6, CED4, CED7, CEP4)
- Ser capaz de sintetizar el trabajo realizado en un documento que siga el formato de un artículo científico (competencias CG4, CED8)

## CONTENIDOS

## METODOLOGÍA

La docencia de esta asignatura se impartirá a distancia, siguiendo el modelo educativo propio de la UNED adaptado al EEES. El principal instrumento docente será un curso virtual dentro de las plataformas educativas para la enseñanza a distancia, complementado con la asistencia personalizada del equipo docente y la tutela presencial y telemática.

Dentro del curso virtual el alumnado dispondrá de:

- Página de bienvenida, donde se indica el concepto general de la asignatura y se presenta el equipo docente.
- Calendario, donde se establece el orden temporal de actividades y sugerencias sobre el reparto temporal de la materia, para que el estudiante los adapte a su disponibilidad y necesidades.
- Materiales:
  - Guía didáctica del curso, donde se establecen los objetivos concretos y los puntos de interés.
  - Programa, donde se especifica la división del contenido por capítulos.
  - Procedimiento, donde se sugieren al alumno las tareas que debe realizar.
  - Ejemplos de trabajos, donde se orienta sobre las pruebas escritas y se muestran ejemplos de trabajos de cursos anteriores.
- Comunicación:
  - Correo para comunicaciones individuales.
  - Foros de Debate donde se intercambian conocimientos y se resuelven dudas de tipo académico general.



- Grupos de trabajo para intercambiar información dentro de los grupos.

Además, la asignatura contará con la web asignatura:

<http://www.issi.uned.es/doctorado/generative/index.htm>, que incluirá el siguiente material:

- Todas las referencias bibliográficas de la asignatura.
- La mayor parte de la bibliografía básica y complementaria de la asignatura.
- Videos introductorios de cada tema.
- Las diapositivas empleadas en los videos introductorios de cada tema.
- El código de los ejemplos utilizados en la asignatura.
- La plantilla de texto que deberán emplearse para redactar el trabajo del curso.

A lo largo del curso, los alumnos irán realizando un trabajo que constará de las siguientes actividades:

<b>Actividad</b>	<b>Resultado de Aprendizaje</b>
Plantear un problema real donde aplicar el paradigma estudiado en el curso	Comprender el impacto que tienen los conceptos de reutilización y abstracción en la producción de software
Identificar los beneficios y costes de abordar la línea de productos planteada	Aprender los principios metodológicos que guían el desarrollo de una línea de productos software, es decir, los fundamentos de la ingeniería de dominio e ingeniería de aplicación
Modelar con un diagrama de características el dominio de la línea de productos	Ser capaz de modelar mediante un diagrama de características el dominio de una línea de productos software
Analizar cuales son los mecanismos más idóneos para implementar la línea de productos	Conocer distintos enfoques para de desarrollar una familia de productos, distinguiendo (i) sus puntos fuertes y débiles, y (ii) las herramientas informáticas que los soportan
Codificar la línea en Ruby	Ser capaz de implementar una línea de productos software mediante el lenguaje de programación Ruby
Documentar la solución con dos informes: (i) documentación de desarrollo, (ii) artículo científico donde de manera sintética se explique el trabajo realizado	Ser capaz de sintetizar el trabajo realizado en un documento que siga el formato de un artículo científico



## SISTEMA DE EVALUACIÓN

### BIBLIOGRAFÍA BÁSICA

Tema 1:

- Heradio Gil, R. "Metodología de desarrollo de software basada en el paradigma generativo. Realización mediante la transformación de ejemplares". Ph. D. Thesis, Departamento de Ingeniería de Software y Sistemas Informáticos de la UNED, España, April 2007.

Tema 2:

- Capítulos 3 y 5 de Czarnecki, K. "Generative Programming Principles and Techniques of Software Engineering Based on Automated Configuration and Fragment-Based Component Models". Ph. D. Thesis, Department of Computer Science and Automation, Technical University of Ilmenau, October 1998.

Tema 3:

- Sección 3.2.2.2 de Heradio Gil, R. "Metodología de desarrollo de software basada en el paradigma generativo. Realización mediante la transformación de ejemplares". Ph. D. Thesis, Departamento de Ingeniería de Software y Sistemas Informáticos de la UNED, España, April 2007.
- Capítulo 8 de Greenfield, J.; Short, K. "Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools". Wiley, 2004.
- Fowler, M. "Language Workbenches: The Killer-App for Domain Specific Languages?" 12 Jun 2005. URL: <http://www.martinfowler.com/articles/languageWorkbench.html>

Tema 4:

- Capítulos 1, 2 y 3 de Pohl C. et al. "Survey of existing implementation techniques with respect to their support for the requirements identified in M3.2". AMPLE Consortium, Version 1.2, 7/30/2007. URL: <http://ample.holos.pt>.

Tema 5:

- Herrington, J. "Code Generation in Action". Manning, 2003.

Tema 6:

- R. Day. "How to Write and Publish a Scientific Paper". Cambridge University Press, 1989.



## BIBLIOGRAFÍA COMPLEMENTARIA

Resultados de aprendizaje: (a) *Comprender el impacto que tienen los conceptos de reutilización y abstracción en la producción de software*, (b) *Aprender los principios metodológicos que guían el desarrollo de una línea de productos software, es decir, los fundamentos de la ingeniería de dominio e ingeniería de aplicación*, y (c) *Ser capaz de modelar mediante un diagrama de características el dominio de una línea de productos software*

- Czarnecki, K.; Eisenecker, U. "Generative Programming: Methods, Tools, and Applications". Addison-Wesley, 2000.
- Návrat, P. A closer look at programming expertise: critical survey of some methodological issues. In Information and Software Technology, no. 38, 1996, Elsevier, pp. 37-46.
- Greenfield, J.; Short, K. "Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools". Wiley, 2004.
- Stahl, T.; Voelter, M. "Model-Driven Software Development: Technology, Engineering, Management". Wiley, 2006.
- Mellor, S. J.; Scott, K.; Uhl, A.; Weise, D. "MDA Distilled. Principles of Model-Driven Architecture". Addison-Wesley, 2004.
- Linden, F. J.; Schmid, K. Rommes, E. "Software Product Lines in Action: The Best Industrial Practice in Product Line Engineering". Springer, 2007.

Resultado de aprendizaje: *Conocer distintos enfoques para de desarrollar una familia de productos, distinguiendo (i) sus puntos fuertes y débiles, y (ii) las herramientas informáticas que los soportan*

- Aho A. V., Lam M. S., Sethi R.; Ullman J.D. "Compilers: Principles, Techniques, and Tools". Addison Wesley; 2nd edition (August 31, 2006).
- Parr, T. "The Definitive ANTLR Reference: Building Domain-Specific Languages". Pragmatic Bookshelf , May 17, 2007.
- Cook, S., Jones, G.; Kent, S. Wills, A. C. "Domain-Specific Development with Visual Studio DSL Tools". Addison-Wesley Professional (June 3, 2007).
- E. Visser. "Stratego: A language for program transformation based on rewriting strategies. System description of Stratego 0.5". In A. Middeldorp, editor, Rewriting Techniques and Applications (RTA'01), volume 2051 of Lecture Notes in Computer Science, pages 357-361. Springer-Verlag, May 2001. URL: <http://www.program-transformation.org/Stratego>
- Capítulos 2 y 5 de Heradio Gil, R. "Metodología de desarrollo de software basada en el paradigma generativo. Realización mediante la transformación de ejemplares". Ph. D. Thesis, Departamento de Ingeniería de Software y Sistemas Informáticos de la UNED, España, April 2007.





- Heradio, R.; Cerrada, J. A. "Software Product Line Development by Analogy". Internacional Summer School, GTTSE (Generative and Transformational Techniques in Software Engineering). Braga, Portugal, July 2-7, 2007.
- Coplien J. O. "Multi-Paradigm Design for C++". Addison-Wesley, 1999
- Gamma, E.; Helm, R.; Johnson, R.; Vlissides, J. "Design Patterns: Elements of Reusable Object-Oriented Software". Addison Wesley, 1994.
- Alexandrescu, A. Modern C++ Design. Generic Programming and Design Patterns Applied. Addison-Wesley 2001.
- Batory, D. "A Tutorial on Feature Oriented Programming and the AHEAD Tool Suite (ATS)". September 8, 2004.

Resultado de aprendizaje: *Ser capaz de implementar una línea de productos software mediante el lenguaje de programación Ruby*

Thomas, D.; Hunt, A. "Programming Ruby. The Pragmatic Programmers' Guide". Addison Wesley, 2nd edition (October 1, 2004).

## RECURSOS DE APOYO Y WEBGRAFÍA

La plataforma de e-Learning aLF, proporcionará el adecuado interfaz de interacción entre el alumno y sus profesores. aLF es una plataforma de e-Learning y colaboración que permite impartir y recibir formación, gestionar y compartir documentos, crear y participar en comunidades temáticas, así como realizar proyectos online.

Además, el equipo docente mantiene una página Web con la asignatura en la que se mantienen contenidos, información y materiales en <http://www.issi.uned.es/doctorado/generative/index.htm>

Se ofrecerán las herramientas necesarias para que, tanto el equipo docente como el alumnado, encuentren la manera de compaginar tanto el trabajo individual como el aprendizaje cooperativo. Además, el estudiante podrá realizar consultas al equipo docente a través del correo, teléfono y presencialmente en los horarios establecidos para estas actividades. También se podrán organizar videoconferencias si las necesidades docentes lo hicieran preciso.

## IGUALDAD DE GÉNERO

En coherencia con el valor asumido de la igualdad de género, todas las denominaciones que en esta Guía hacen referencia a órganos de gobierno unipersonales, de representación, o miembros de la comunidad universitaria y se efectúan en género masculino, cuando no hayan sido sustituido por términos genéricos, se entenderán hechas indistintamente en género femenino o masculino, según el sexo del titular que los desempeñe.

