24-25

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

UNED

24-25

DESARROLLO DE LÍNEAS DE PRODUCTO SOFTWARE MEDIANTE UN ENFOQUE GENERATIVO

CÓDIGO 31105043

ÍNDICE

Nombre de la asignatura	DESARROLLO DE LÍNEAS DE PRODUCTO SOFTWARE MEDIANTE UN ENFOQUE GENERATIVO
Código	31105043
Curso académico	2024/2025
Título en que se imparte	MÁSTER UNIVERSITARIO EN INVESTIGACIÓN EN INGENIERÍA DE SOFTWARE Y SISTEMAS INFORMÁTICOS
Tipo	CONTENIDOS
Nº ETCS	9
Horas	225
Periodo	ANUAL
Idiomas en que se imparte	CASTELLANO

PRESENTACIÓN Y CONTEXTUALIZACIÓN

Presentación

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.

Contextuatlización

La asignatura “Desarrollo de Líneas de Producto Software Mediante un Enfoque Generativo” es anual, de 9 ECTS (dedicación estimada de 225 horas), de carácter optativo y perteneciente al Bloque de Ingeniería de Software. Concretamente, esta asignatura es una de las seis que forman la materia de Ingeniería del Desarrollo de Software, y tiene una gran relación con cuatro de las otras cinco asignatura de la materia:

- La asignatura “Especificación de los Sistemas Software” trata la especificación formal de sistemas software. Los contenidos de esta asignatura se aplicarán en (i) el tema 2 de “desarrollo de familias de productos de software desde un enfoque generativo”, donde estudiaremos cómo especificar formalmente la variabilidad de una familia de productos, y (ii) el tema 3, donde se explicará como metamodelar un lenguaje específico de dominio.
- La asignatura “Arquitecturas para Sistemas Software” está íntimamente relacionada con “desarrollo de familias de productos de software desde un enfoque generativo”. De hecho, la arquitectura de una línea de productos software es la generalización de las distintas arquitecturas de cada producto particular.
- La asignatura “Generación Automática de Código” trata en profundidad la producción generativa de software. Dicha asignatura puede considerarse una ampliación de parte del contenido de los temas 4 y 5 del presente curso.
- La asignatura “Arquitecturas Orientadas a Servicios” presenta la reutilización de componentes software distribuidos en Internet. El principal objetivo de una línea de productos software es la reutilización sistemática de software. Por lo tanto, la relación entre ambas asignaturas es evidente.

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 **requerido** o necesario es el correspondiente a las asignaturas:

- Fundamentos de Programación
- Programación Orientada a Objetos
- Introducción a la Ingeniería de Software
- Diseño del Software

El conocimiento **recomendado** es el correspondiente a las asignaturas:

- Autómatas, Gramáticas y Lenguajes

•Teoría de los Lenguajes de Programación

EQUIPO DOCENTE

Nombre y Apellidos	RUBEN HERADIO GIL (Coordinador de asignatura)
Correo Electrónico	rheradio@issi.uned.es
Teléfono	91398-8242
Facultad	ESCUELA TÉCN.SUP INGENIERÍA INFORMÁTICA
Departamento	INGENIERÍA DE SOFTWARE Y SISTEMAS INFORMÁTICOS

HORARIO DE ATENCIÓN AL ESTUDIANTE

En la metodología a distancia de la UNED, los **foros** del curso virtual son el principal recurso de atención colectiva los estudiantes. La comunicación a través de los foros tiene una doble vertiente en el aprendizaje: el enriquecimiento en el ejercicio de la dialéctica y del diálogo entre los estudiantes, por un lado, y la exposición del profesor a todos los alumnos (atención colectiva), junto con el debate que ello pueda suscitar.
En la atención colectiva de los foros del curso virtual, ante cualquier cuestión concreta, planteada sobre los contenidos o el funcionamiento de la asignatura, la respuesta será inferior a 5 días del calendario lectivo.
En cuanto a la atención individual, el equipo docente dará respuesta a través del teléfono (en el horario lectivo indicado) y, en horario laboral peninsular, por correo electrónico:
Horario de atención presencial y telefónica (*guardia*):
Jueves lectivos de 10:00 a 14:00 horas.
Profesorado:
Rubén Heradio Gil.
Telf.: +34 91398 8242
Correo electrónico: rheradio@issi.uned.es
Dirección postal:
ETS de Ingeniería Informática de la UNED
Dpto. de Ingeniería de Software y Sistemas Informáticos. Despacho 2.21.
C/ Juan del Rosal, 16
28040 Madrid

COMPETENCIAS QUE ADQUIERE EL ESTUDIANTE

Competencias Básicas:
CB6 - Poseer y comprender conocimientos que aporten una base u oportunidad de ser originales en el desarrollo y/o aplicación de ideas, a menudo en un contexto de investigación
CB7 - Que los estudiantes sepan aplicar los conocimientos adquiridos y su capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios (o multidisciplinares) relacionados con su área de estudio
CB8 - Que los estudiantes sean capaces de integrar conocimientos y enfrentarse a la

complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios
CB9 - Que los estudiantes sepan comunicar sus conclusiones y los conocimientos y razones últimas que las sustentan a públicos especializados y no especializados de un modo claro y sin ambigüedades
CB10 - Que los estudiantes posean las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

Competencias Generales:
CG01 - Saber aplicar los conocimientos adquiridos y la capacidad de resolución de problemas en entornos nuevos o poco conocidos dentro de contextos más amplios y multidisciplinares relacionados con la Ingeniería de Sistemas y la Ingeniería de Software.
CG02 - Demostrar una comprensión sistemática del campo de estudio de la Ingeniería de Software o de la Ingeniería de Sistemas, y el dominio de las habilidades y métodos de investigación relacionados con dicho campo.
CG03 - Demostrar la capacidad de concebir, diseñar, poner en práctica y adoptar un proceso sustancial de investigación con seriedad académica.
CG04 - Ser capaz de realizar un análisis crítico, evaluación y síntesis de ideas nuevas y complejas.
CG05 - Saber comunicar sus conclusiones -y los conocimientos y razones últimas que las sustentan- a públicos especializados y no especializados, a sus colegas, a la comunidad académica en su conjunto y a la sociedad, de un modo claro y sin ambigüedades.
CG06 - Ser capaz de fomentar, en contextos académicos y profesionales, el avance tecnológico dentro de una sociedad basada en el conocimiento.
CG07 - Ser capaz de integrar conocimientos y enfrentarse a la complejidad de formular juicios a partir de una información que, siendo incompleta o limitada, incluya reflexiones sobre las responsabilidades sociales y éticas vinculadas a la aplicación de sus conocimientos y juicios.
CG08 - Realizar una contribución a través de una investigación original que amplíe las fronteras del conocimiento desarrollando un corpus sustancial, del que parte merezca la publicación referenciada a nivel nacional o internacional.
CG09 - Poseer las habilidades de aprendizaje que les permitan continuar estudiando de un modo que habrá de ser en gran medida autodirigido o autónomo.

Competencias Específicas:
CE01 - Incorporar mejoras cualitativas sustanciales, bien sea en la elaboración de software o bien en el desarrollo e implantación de sistemas robóticos.
CE02 - Concebir, implementar implantar y supervisar nuevas soluciones a los problemas específicos que se le planteen en el ámbito de la investigación, innovación y desarrollo de software o de la robótica.

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 de 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

Tema 1: Introducción

1.1. Motivación:
1.1.a. Reutilización
1.1.b. Abstracción
1.2. Ejemplo práctico de motivación: “Diccionarios en Java”.
1.3. Organización del curso.

Tema 2: Aspectos metodológicos

2.1. Ingeniería de Dominio e Ingeniería de Aplicación.
2.2. Análisis de Dominio: Diagramas de características.

Tema 3: Lenguajes específicos de dominio

3.1. Cualidades deseables de un Lenguaje Específico de Dominio (Domain Specific Language, DSL).
3.2. Sintaxis y Semántica de un DSL:

3.2.a. Gramáticas BNF
3.2.b. Metamodelos
3.3. Cómo implementar un analizador para DSLs:
3.3.a. Metaparsers, Lenguajes de Transformación.
3.3.b. XML
3.3.c. DSLs embebidos en Lenguajes de Propósito General.
3.3.d. Herramientas de Metamodelado (DSL Tools de Microsoft, GME+EMF de Eclipse).

Tema 4: Implementación de la variabilidad de una línea de productos

4.1. Motivación
a. Cualidades deseables para una técnica de gestión de la variabilidad.
b. Inexistencia de una técnica de “propósito general”
4.2. Técnicas internas: composición, herencia, polimorfismo, genericidad, orientación a aspectos.
4.3. Técnicas externas: generación de código.

Tema 5: Ruby, un lenguaje para la implementación de líneas de productos desde un enfoque generativo

5.1. Justificación del uso de Ruby.
5.2. Análisis de DSLs con Ruby.
5.2.a. Metaparsers (racc, rockit)
5.2.b. XML
5.2.c. DSLs embebidos.
5.3. Generación de código con Ruby.
5.3.a. Librería de plantillas de texto ERB.

Tema 6: Enunciado del trabajo fin de curso

6.1. Enunciado.
6.2. Estructura y contenidos de un artículo científico.
6.3. Consulta de artículos desde la UNED.

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, gestionado por el equipo docente, y la supervisión personalizada del estudiante, tanto presencial como telemática.

Dentro del curso virtual, el alumnado dispondrá de:

1. Información detallada de cada uno de los temas, incluyendo:

•Uno o varios videos introductorios para cada tema

•Las diapositivas empleadas en los videos introductorios

•Ejemplos resueltos y su respectivo código

•Todas las referencias bibliográficas de cada tema

•Gran parte de la bibliografía básica y complementaria de cada tema (que, al ser de uso libre, se pone a disposición del alumnado)

2. El enunciado de un trabajo que los alumnos deberán realizar de manera progresiva a lo largo del curso. La siguiente tabla resume las actividades intermedias que componen el trabajo, junto sus respectivos resultados de aprendizaje.

Actividad	Resultado de Aprendizaje
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 linea 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

3. Un cronograma, donde se propone una secuencia temporal para estudiar la asignatura y realizar el trabajo
4. Foros de debate, organizados por cada tema del curso
5. Preguntas más frecuentes

SISTEMA DE EVALUACIÓN

TIPO DE PRIMERA PRUEBA PRESENCIAL

Tipo de examen: No hay prueba presencial

TIPO DE SEGUNDA PRUEBA PRESENCIAL

Tipo de examen2: No hay prueba presencial

CARACTERÍSTICAS DE LA PRUEBA PRESENCIAL Y/O LOS TRABAJOS

Requiere Presencialidad: No

Descripción

El alumno deberá realizar un trabajo práctico donde aplicará los conceptos estudiados en la asignatura. Se estima que la realización completa de dicho trabajo requiere 108 horas.

El trabajo se descompone en las 5 Partes siguientes:

Análisis de dominio de un problema práctico planteado por el alumno, 10 horas

Especificación del DSL que se utilizará en el supuesto práctico planteado por el alumno, 15 horas

Estudio de alternativas para realizar la implementación de la variabilidad de la línea de productos planteada por el alumno, 20 horas

Implementación de la línea de productos planteada por el alumno, 35 horas

Documentación del trabajo realizado, 28 horas

Criterios de evaluación

A lo largo del curso, los alumnos pueden enviar voluntariamente cada parte de la elaboración del trabajo para recibir feedback del equipo docente.

En cualquier caso, la evaluación se realiza exclusivamente sobre el trabajo final, que se documentará siguiendo las directrices indicadas en el Tema 6.

Los criterios de evaluación del trabajo son:

Interés del caso práctico abordado. La primera PEC tiene por objetivo que el alumno proponga un campo de aplicación de la asignatura donde desarrollará su trabajo. Además, explorará la viabilidad del trabajo, realizando el análisis de dominio correspondiente. Conviene que antes de seguir adelante, el alumno consulte con el equipo docente si vale la pena seguir desarrollando el ejemplo elegido o, por el contrario, conviene plantear un nuevo problema.

Adecuación y corrección de la especificación formal de la sintaxis y semántica del DSL el alumno utilizará en su trabajo.

Correcta evaluación y elección de las distintas alternativas para implementar la variabilidad de la línea de productos planteada por el alumno.

Corrección de la implementación final de la línea de productos planteada por el alumno.

Claridad y capacidad de síntesis en la documentación del trabajo desarrollado a lo largo del curso.

Ponderación de la prueba presencial y/o los trabajos en la nota final	100%
Fecha aproximada de entrega	08/06/2024 (convocatoria de junio) y 07/09/2024 (convocatoria de septiembre)

Comentarios y observaciones

El trabajo se puede entregar en las convocatorias de junio y/o septiembre. El plazo coincide con el último día de exámenes del Grado en Ingeniería Informática según el calendario publicado por la UNED (ver https://www.uned.es/universidad/inicio/estudiantes/calendario-examenes.html).

En caso de que un trabajo entregado en junio no alcance la calificación de 5, el coordinador de la asignatura informará a el/la estudiante acerca de las deficiencias encontradas y cómo corregirlas para que proceda a su nueva entrega en septiembre.

PRUEBAS DE EVALUACIÓN CONTINUA (PEC)

¿Hay PEC?	No
Descripción	
Criterios de evaluación	
Ponderación de la PEC en la nota final	
Fecha aproximada de entrega	
Comentarios y observaciones	

OTRAS ACTIVIDADES EVALUABLES

¿Hay otra/s actividad/es evaluable/s? No

Descripción

Criterios de evaluación

Ponderación en la nota final

Fecha aproximada de entrega

Comentarios y observaciones

¿CÓMO SE OBTIENE LA NOTA FINAL?

La evaluación se realiza exclusivamente sobre el trabajo final, que se documentará obligatoriamente siguiendo las directrices indicadas en el Tema 6.

La asignatura estará superada con una calificación 5 en dicho trabajo.

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, Framworks, 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

En el curso virtual el equipo docente mantiene el acceso y gran parte de los contenidos del material de apoyo para el estudio de la asignatura.
Además, también se 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
Con el fin de introducir cada tema del curso, el equipo docente ha desarrollado 7 videos disponibles en UNED INTECCA y en CANAL UNED:

Tema 1:
https://www.intecca.uned.es/portalavip/grabacion.php?ID_Sala=3&ID_Grabacion=67055&hashData=305370828e3762e881049ac224ed2e04¶msToCheck=SURfR3JhYmFjaW9uLElEX1NhbGEs

·Tema 2:
https://www.intecca.uned.es/portalavip/grabacion.php?ID_Sala=3&ID_Grabacion=67056&hashData=9525149f8e58b06f11f41224a98a61a6¶msToCheck=SURfR3JhYmFjaW9uLElEX1NhbGEs

·Tema 3:
https://www.intecca.uned.es/portalavip/grabacion.php?ID_Sala=3&ID_Grabacion=67057&hashData=2353215673386479 54e8a05af13a623c¶msToCheck=SURfR3JhYmFjaW9uLElEX1NhbGEs

·Tema 4:
https://www.intecca.uned.es/portalavip/grabacion.php?ID_Sala=3&ID_Grabacion=67059&hashData=ed492119b8a774a114963f8213fae096¶msToCheck=SURfR3JhYmFjaW9uLElEX1NhbGEs

·Tema 5:
o
https://www.intecca.uned.es/portalavip/grabacion.php?ID_Sala=3&ID_Grabacion=67060&hashData=5b0aba80eeb0c0718f6d5a145f21bb45¶msToCheck=SURfR3JhYmFjaW9uLElEX1NhbGEs
o https://canal.uned.es/iframe/5aa63462b1111f026e8b4573
o https://canal.uned.es/iframe/5aa63466b1111f026e8b4579
o https://canal.uned.es/iframe/5aa63460b1111f026e8b456d
·Tema 6:
https://www.intecca.uned.es/portalavip/grabacion.php?ID_Sala=3&ID_Grabacion=67061&hashData=02ead9f49b7cb4a266acf8be954d67fc¶msToCheck=SURfR3JhYmFjaW9uLElEX1NhbGEs

·Tema de carácter voluntario donde se comentan distintas posibilidades para realizar un futuro Trabajo Fin de Máster en el área de las líneas de producto software:
https://www.intecca.uned.es/portalavip/grabacion.php?ID_Sala=3&ID_Grabacion=67062&hashData=14b903edc70e65ae1595c41c031165a6¶msToCheck=SURfR3JhYmFjaW9uLElEX1NhbGEs

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 se hayan sustituido por términos genéricos, se entenderán hechas indistintamente en género femenino o masculino, según el sexo del titular que los desempeñe.