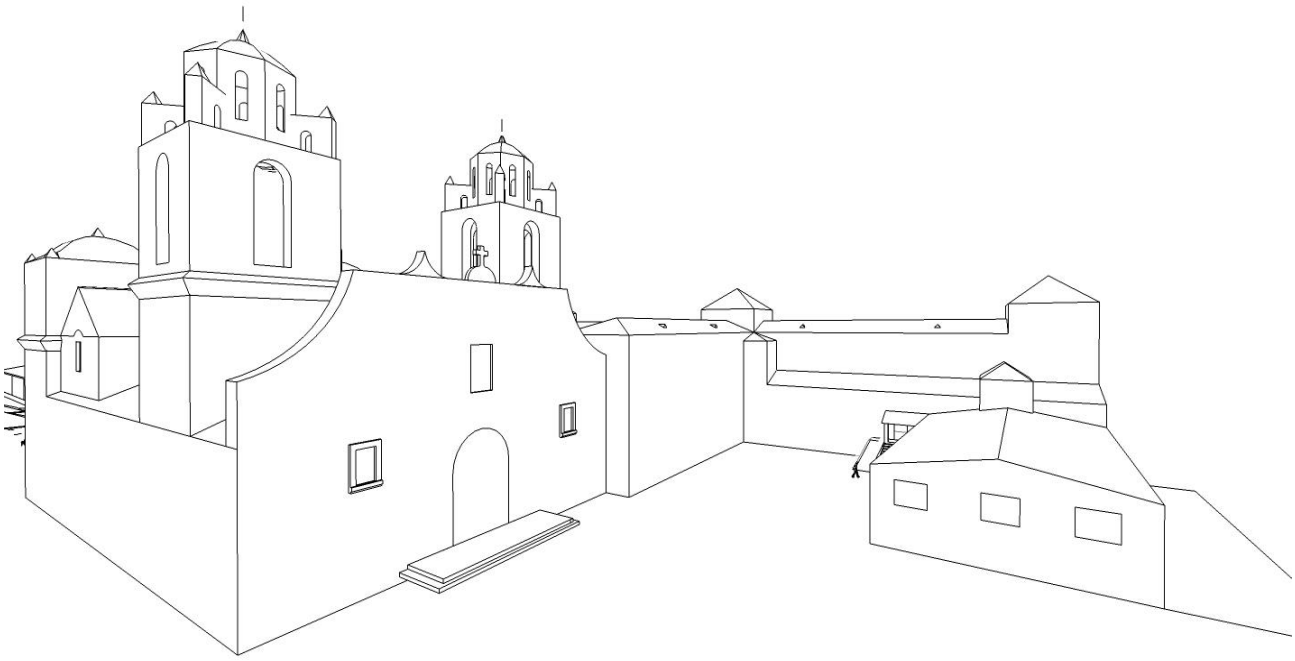
INSTITUTO SUPERIOR DE FORMACIÓN PROFESIONAL
SAN ANTONIO
UCAM

ED3DWP – ENTORNOS DE DESARROLLO

TÉCNICO SUPERIOR EN DESARROLLO DE APLICACIONES WEB

Ciclo Formativo de Grado Superior (LOE)



Índice

Entornos de Desarrollo

Código: **ED3DWP**

Nº de Créditos: **6 ECTS (90 horas)**

Unidad Temporal: **Primer curso**

Resultados de aprendizaje y criterios de evaluación

Los resultados de aprendizaje, para este módulo, establecidos en la legislación vigente (Real Decreto 686/2010, de 20 de mayo), son los siguientes:

RA1. Reconoce los elementos y herramientas que intervienen en el desarrollo de un programa informático, analizando sus características y las fases en las que actúan hasta llegar a su puesta en funcionamiento.

Criterios de evaluación:

a) Se ha reconocido la relación de los programas con los componentes del sistema informático: memoria, procesador y periféricos, entre otros.
b) Se han identificado las fases de desarrollo de una aplicación informática.
c) Se han diferenciado los conceptos de código fuente, código objeto y código ejecutable.
d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales.
e) Se han clasificado los lenguajes de programación.
f) Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en programación.

RA2. Evalúa entornos integrados de desarrollo, analizando, sus características para editar código fuente y generar ejecutables.

Criterios de evaluación:

a) Se han instalado entornos de desarrollo, propietarios y libres.
b) Se han añadido y eliminado módulos en el entorno de desarrollo.
c) Se ha personalizado y automatizado el entorno de desarrollo.
d) Se ha configurado el sistema de actualización del entorno de desarrollo.
e) Se han generado ejecutables a partir de código fuente de diferentes lenguajes en un mismo entorno de desarrollo.
f) Se han generado ejecutables a partir de un mismo código fuente con varios entornos de desarrollo.
g) Se han identificado las características comunes y específicas de diversos entornos de desarrollo.

RA3. Verifica el funcionamiento de programas, diseñando y realizando pruebas.

Criterios de evaluación:

a) Se han identificado los diferentes tipos de pruebas.
b) Se han definido casos de prueba.
c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.
d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento.
e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.
f) Se han efectuado pruebas unitarias de clases y funciones.
g) Se han implementado pruebas automáticas.

h) Se han documentado las incidencias detectadas.

RA4. Optimiza código empleando las herramientas disponibles en el entorno de desarrollo.

Criterios de evaluación:

a) Se han identificado los patrones de refactorización más usuales.
b) Se han elaborado las pruebas asociadas a la refactorización.
c) Se ha revisado el código fuente usando un analizador de código.
d) Se han identificado las posibilidades de configuración de un analizador de código.
e) Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.
f) Se ha realizado el control de versiones integrado en el entorno de desarrollo.
g) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.

RA5. Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.

Criterios de evaluación:

a) Se han identificado los conceptos básicos de la programación orientada a objetos.
b) Se ha instalado el módulo del entorno integrado de desarrollo que permite la utilización de diagramas de clases.
c) Se han identificado las herramientas para la elaboración de diagramas de clases.
d) Se ha interpretado el significado de diagramas de clases.
e) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.
f) Se ha generado código a partir de un diagrama de clases.
g) Se ha generado un diagrama de clases mediante ingeniería inversa.

RA6. Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.

Criterios de evaluación:

a) Se han identificado los distintos tipos de diagramas de comportamiento.
b) Se ha reconocido el significado de los diagramas de casos de uso.
c) Se han interpretado diagramas de interacción.
d) Se han elaborado diagramas de interacción sencillos.
e) Se ha interpretado el significado de diagramas de actividades.
f) Se han elaborado diagramas de actividades sencillos.
g) Se han interpretado diagramas de estados.
h) Se han planteado diagramas de estados sencillos.

Contenidos Básicos

Los contenidos básicos que se impartirán en este módulo son los establecidos en por la Conserjería de Educación, Formación y Empleo para el Currículo del Ciclo Formativo de Grado Superior correspondiente al Título de Técnico Superior en Desarrollo de Aplicaciones Web (Orden de 12 de marzo de 2013):

- **Desarrollo de software:**
  - Concepto de programa informático.
  - Instrucciones y datos.
  - Ejecución de programas en ordenadores:
    - Datos y programas.
    - Hardware frente a software.

        - Estructura funcional de un ordenador: procesador, memoria.
        - Tipos de software. BIOS. Sistema. Aplicaciones.
        - Código fuente, código objeto y código ejecutable; máquinas virtuales.
    - Lenguajes de programación:
        - Tipos de lenguajes de programación.
        - Características de los lenguajes más difundidos.
    - Introducción a la ingeniería del software:
        - Proceso software y ciclo de vida del software.
        - Fases del desarrollo de una aplicación: análisis, diseño, codificación, pruebas, documentación, explotación y mantenimiento, entre otras.
        - Modelos de proceso de desarrollo software (cascada, iterativo, evolutivo).
        - Metodologías de desarrollo software. Características. Técnicas.
        - Objetivos. Tipos de metodologías.
        - Herramientas CASE (Computer Aided Software Engineering).
    - Proceso de obtención de código ejecutable a partir del código fuente. Herramientas implicadas: editores, compiladores, enlazadores, etc.
    - Errores en el desarrollo de programas.
    - Importancia de la reutilización de código.

- **Instalación y uso de entornos de desarrollo:**
    - Funciones de un entorno de desarrollo.
    - Tipos de entornos de desarrollo. Entornos de desarrollo libres y propietarios. Características.
    - Instalación de un entorno de desarrollo.
    - Uso básico de un entorno de desarrollo:
        - Uso de herramientas y asistentes.
        - Creación de proyectos.
        - Incorporación de elementos a proyectos.
        - Edición de programas. Sintaxis y formateo de código.
        - Compilación de programas. Detección de errores.
        - Generación de programas ejecutables.
        - Ejecución de programas.
        - Paneles y vistas.
        - Importación y exportación de ficheros.
        - Personalización.
        - Acceso a documentación.
        - Instalación y desinstalación de aplicaciones, módulos y plugins adicionales.
        - Configuración de actualizaciones.
        - Automatización de tareas.

- **Diseño y realización de pruebas:**
    - Pruebas en el proceso de desarrollo de software:
        - Planificación de pruebas a lo largo del ciclo de desarrollo.
        - Tipos de pruebas: funcionales, estructurales, regresión, caja negra, etc.
        - Procedimientos y casos de prueba.
    - Pruebas de código:
        - Cubrimiento, valores límite, clases de equivalencia, etc.

        - Pruebas unitarias de clases y funciones.
        - Uso de herramientas integradas en los entornos de desarrollo para realizar pruebas unitarias.
        - Automatización de pruebas unitarias.
        - Pruebas de integración.
        - Diseño y documentación casos de prueba.
    - Depuración de programas:
        - Herramientas de depuración integradas en los entornos de desarrollo.
        - Puntos de ruptura y seguimiento en tiempo de ejecución.
        - Examinadores de variables.
        - Normas de calidad.

- **Optimización y documentación:**
    - Refactorización:
        - Concepto. Limitaciones.
        - Patrones de refactorización más usuales.
        - Analizadores de código; uso; configuración.
        - Refactorización y pruebas.
        - Herramientas de ayuda a la refactorización.

- **Control de versiones:**
        - Desarrollos colectivos.
        - Estructura de las herramientas de control de versiones. Utilidad.
        - Características. Estructura (Cliente/Servidor). Repositorio.
        - Herramientas de control de versiones. Clientes de control de versiones integrados en el entorno de desarrollo. Descarga de ficheros inicial. Modificación de ficheros. Actualización de ficheros en local. Actualización de ficheros en el repositorio. Diferencias entre versiones. Restauración de versiones anteriores. Resolución de conflictos. Historial de versiones.
    - Documentación:
        - Uso de comentarios.
        - Herramientas integradas en el entorno de desarrollo para generar documentación automáticas de clases.
        - Alternativas.

- **Introducción al lenguaje unificado de modelado (UML Unified Modeling Language):**
    - Características.
    - Versiones.
    - Diagramas UML.
    - Utilización en metodologías de desarrollo orientado a objetos.
    - Herramientas CASE con soporte UML.
    - Elaboración de diagramas de clases.
    - Notación de los diagramas de clases.
        - Clases. Atributos, métodos y visibilidad.
        - Objetos. «Instanciación».
        - Relaciones. Asociación, herencia, composición, agregación, dependencia, navegabilidad.

        - Clases abstractas. Interfaces.
        - Paquetes.
        - Grado de detalle.
    - Utilización de herramientas CASE para elaborar diagramas de clases con UML
    - Módulos integrados en entornos de desarrollo para elaborar diagramas de clases.
    - Creación de código a partir de diagramas de clases.
    - Generación de diagramas de clases a partir de código (ingeniería inversa).

- **Elaboración de diagramas de comportamiento:**
    - Tipos. Campo de aplicación.
    - Diagramas de casos de uso.
    - Actores, escenario, casos de uso, asociaciones (relación de comunicación entre actores y casos de uso), relaciones entre casos de uso.
    - Diagramas de secuencia.
    - Línea de vida de un objeto/actor, activación, envío de mensajes.
    - Diagramas de colaboración.
    - Objetos/actores, mensajes.
        - Diagramas de actividades.
        - Actividades, transiciones, decisiones y combinaciones.
    - Diagramas de estado.
        - Estados, eventos, señales, transiciones.
    - Utilización de herramientas CASE para elaborar diagramas de comportamiento.
    - Módulos integrados en entornos de desarrollo para elaborar diagramas de comportamiento.

Orientaciones pedagógicas

Este módulo profesional contiene parte de la formación necesaria para desempeñar la función de desarrollador de aplicaciones.

La función de desarrollador de aplicaciones incluye aspectos como:

- La utilización de las herramientas software disponibles.
- La elaboración de documentación interna y técnica de la aplicación.
- La elaboración y ejecución de pruebas.
- La optimización de código.

Las actividades profesionales asociadas a esta función se aplican en el desarrollo de aplicaciones.

La formación del módulo contribuye a alcanzar los objetivos generales d), e), h), i) y j) del ciclo formativo y las competencias profesionales, personales y sociales d), f), h), i) y j) del título.

Las líneas de actuación en el proceso de enseñanza-aprendizaje que permiten alcanzar los objetivos del módulo están relacionados con:
- La interpretación de documentación técnica.
- La instalación, configuración y personalización de diversos entornos de desarrollo.
- La utilización de distintos entornos de desarrollo para la edición y prueba de aplicaciones.

- La utilización de herramientas de depuración, optimización y documentación de aplicaciones.
- La generación de diagramas técnicos.
- La elaboración de la documentación interna de la aplicación.

Metodología

Metodología	Horas de trabajo presencial
Teoría	90 horas
Prácticas en clase	
Trabajo en equipo	
Evaluación	
Preparación de prácticas	
Realización de trabajos	
Búsquedas bibliográficas	

Temario

Unidad 1 – Desarrollo de Software

- El programa informático
- Lenguajes de programación
- Obtención de código ejecutable
- Proceso de desarrollo
  - Tipos de código (fuente, objeto y ejecutable)
  - Compilación
- Introducción a la ingeniería del software
  - Análisis
  - Diseño
  - Codificación
  - Pruebas
  - Documentación
  - Explotación
  - Mantenimiento
- Importancia de la reutilización

Unidad 2 – Instalación y Uso de Entornos de Desarrollo

- Características
  - Extensiones y herramientas
  - Personalización y configuración
- Criterios de elección de un IDE
  - Sistema operativo
  - Lenguaje de programación y framework
  - Herramientas y disponibilidad
- Uso básico de un IDE
  - Edición de programas y generación de ejecutables
  - Desarrollo colaborativo
- Nuestra elección Visual Studio
  - Instalación
  - Recorrido por las ventanas y paletas principales
  - Personalización y configuración

Unidad 3 – Depuración y Realización de Pruebas

- Herramientas de depuración
  - Puntos de ruptura
  - Puntos de seguimiento
  - Inspecciones
- Análisis de código
  - Analizador estático de código
- Casos de prueba
  - Caja blanca
  - Caja negra
  - Rendimiento
  - Coherencia
- Pruebas unitarias
  - Metodología
  - JUnit

Unidad 4 – Optimización y Documentación

- Refactorización
  - Tabulación
  - Patrones de refactorización más usuales
  - Malos olores
  - Refactorización y pruebas
  - Herramientas de Visual Studio
- Control de versiones
  - Repositorios
  - Herramientas de control de versiones
- Documentación
  - Uso de comentarios
  - Herramientas

Unidad 5 – Introducción al Modelado de Sistemas Orientados a Objetos con UML.

- Introducción a UML
  - Historia de UML. Versiones.
  - Objetivos y características.
  - Uso de UML en metodologías de desarrollo del software.
  - Modelado y vistas soportadas por UML.
  - Elementos de UML
  - Tipos de diagramas UML
  - Herramientas CASE con soporte UML.
- Modelado de Requisitos
  - Introducción
  - Casos de uso, actores, escenarios.
  - Relaciones entre Casos de Uso: Inclusión, Extensión.
  - Elaboración de diagramas de Casos de Uso
- Modelado de Estructura
  - Introducción
  - Elementos estructurales
    - Clases, clases abstractas e interfaces.
    - Relaciones entre clases: herencia, asociación, dependencia, implementación.
    - Objetos. Instancias prototípicas.
  - Elaboración de diagramas de clases
    - Creación de código a partir de diagramas de clases
    - Generación de diagramas de clase a partir de código.
  - Elaboración de diagramas de objetos

Unidad 6 – Modelado Avanzado con UML

- Modelado de Interacciones
  - Introducción
  - Elementos de interacción: línea de vida, objeto, activación, mensajes síncronos y asíncronos.
  - Elaboración de diagrama de secuencia
  - Elaboración de diagramas de comunicación
- Modelado de Comportamiento
  - Introducción
  - Eventos y tipos de eventos
  - Elaboración de diagramas estados
  - Elaboración de diagramas de actividades
- Modelado de Arquitectura
  - Introducción
  - Arquitectura lógica
    - Paquetes
    - Elaboración de diagramas de paquetes
  - Arquitectura física
    - Elementos: Componentes, Artefactos, Nodos
    - Elaboración de diagramas de componentes
    - Elaboración de diagramas de despliegue

Sistema de evaluación

La evaluación se realizará a través de pruebas teórico-prácticas de los contenidos establecidos y evaluación continua. Por medio de la aplicación de los criterios de evaluación se medirá el grado de aprendizaje progresivo del alumno y se valorará en que medida va alcanzándose los objetivos establecidos.

La evaluación comprenderá una evaluación sumativa dividida en tres bloques que correspondan con cada una de las tres evaluaciones ordinarias de ciclo.

Para obtener una evaluación positiva se establecen los siguientes requisitos:

- Asistencia regular a clase.
- Realización de los trabajos y actividades propuestas en clase.
- Superación de las pruebas y controles realizados durante el periodo lectivo

Cuando los resultados de la evaluación sean negativos, se realizarán actividades de recuperación que consistirán en pruebas teórico-prácticas en cada evaluación en la cual el alumno no haya alcanzado las destrezas, conocimientos y habilidades requeridas.

Además, se establece una convocatoria de recuperación a final de curso en la cual el alumno podrá recuperar aquellas evaluaciones en las que ha obtenido resultados negativos. Dicha prueba final consistirá en un examen teórico-práctico de los contenidos no superados.

Si tras la recuperación de final de curso, el alumno tiene alguna evaluación negativa, en la prueba extraordinaria de septiembre tendrá la posibilidad de recuperar las evaluaciones con resultados negativos. Esta convocatoria se evaluará con un examen de tipo teórico-práctico, y un conjunto de ejercicios de recuperación que debe ser entregado en esta convocatoria respecto al enunciado indicado al estudiante si no se ha superado la parte prácticas**.**

Primera evaluación: Evaluación continua.

- Parte teórica: 70% del total de la nota.
  - 50% examen teórico-práctico
  - 20% trabajo de redacción y exposición en clase

- Parte práctica: 30% del total de la nota.
  - 20% ejercicios realizados en clase
  - 10 % participación y actitud (se tendrá en cuenta la actitud en clase, el nivel de participación y respeto hacia el trabajo y compañeros)

Segunda evaluación: Evaluación continua.

- Parte teórica: 70% del total de la nota.
  - 50% examen teórico-práctico
  - 20% trabajo de redacción y exposición en clase

- Parte práctica: 30% del total de la nota.
  - 20% ejercicios realizados en clase
  - 10 % participación y actitud (se tendrá en cuenta la actitud en clase, el nivel de participación y respeto hacia el trabajo y compañeros)

Convocatoria Ordinaria de Junio

Evaluación Final: Evaluación continua.

- Parte teórica: 70% del total de la nota.
  - 50% examen teórico-práctico
  - 20% trabajo de redacción y exposición en clase

- Parte práctica: 30% del total de la nota.
  - 20% ejercicios realizados en clase
  - 10 % participación y actitud (se tendrá en cuenta la actitud en clase, el nivel de participación y respeto hacia el trabajo y compañeros)

Recuperación: Correspondiente a la Evaluación Final. Se realizará un examen de tipo teórico-práctico más el trabajo de redacción y se recuperará la parte práctica en caso de no estar superada.

- Parte teórica: 70% del total de la nota.
  - 50% examen teórico-práctico
  - 20% trabajo de redacción y exposición en clase

- Parte práctica: 30% del total de la nota.
  - 20% ejercicios realizados en clase
  - 10 % participación y actitud (se tendrá en cuenta la actitud en clase, el nivel de participación y respeto hacia el trabajo y compañeros)

Convocatoria Extraordinaria de septiembre

Esta convocatoria se evaluará con un examen de tipo teórico-práctico más el trabajo de redacción, en caso de no haber superado esta parte, y un conjunto de ejercicios de recuperación que debe ser entregado en esta convocatoria respecto al enunciado indicado al estudiante si no se ha superado la parte prácticas.

- Parte teórica: 70 % del total de la nota.
  - 50% examen teórico-práctico
  - 20 % trabajo de redacción y entrevista con el profesor

- Parte práctica: 30 % del total de la nota.
  - 30% ejercicios de recuperación y entrevista con el profesor

Bibliografía y fuentes de referencia

Bibliografía básica

- Hueso-Ibañez Galindo, L. Bases De Datos (Cfgs. Ciclos Formativos De Grado Superior). Editorial: RA-MA, ISBN 9788499641577, 2012
- M. Fowler; K. Scott; UML Distilled: A Brief Guide to the Standard Object Modelling Language. 2nd edition. Addison-Wesley Publisher. 1999. ISBN: 020165783X
- F.V. Der Heyde; L. Debrauwer. UML 2: Iniciación, Ejemplos y Ejercicios Corregidos. 2ª edición. Editorial ENI. 2011. ISBN: 9782746047419
- P. Kimmel. Manual de UML. 1ª Edición. Editorial McGraw Hill. 2011. ISBN: 9789701058992
- Pressman, R. Ingeniería del Software: Un enfoque práctico. 7ª edición. Madrid: McGraw Hill, 2010. ISBN: 9701054733.

- Booch, G.; Rumbaugh, J.; Jackobson, I. El lenguaje unificado de modelado: Guia del Usuario. 2ª edición. Madrid: Addison-Wesley, 2006. ISBN: 9788478290765.

Bibliografía complementaria

- BERTHOLD, D. (2005). PROFESIONAL ECLIPSE 3. PARA DESARROLLADORES JAVA.
- Pressman, R. S. (1997). Ingeniería del Software: Un enfoque práctico. Mikel Angoar.
- Beck, K. (2004). JUnit pocket guide. " O'Reilly Media, Inc.".

- Booch, G.; Rumbaugh, J.; Jackobson, I. Lenguaje Unificado de Modelado Manual de Referencia Uml 2.0. 1ª edición. Madrid: Addison-Wesley, 2006. ISBN: 8478290877.
- Sommerville, I.; Sawyer, P. Requirements engineering: a good practice guide. 1ª edición. Londres: Wiley, 1997. ISBN: 9780471974444.
- Booch, G.; Rumbaugh, J.; Jackobson, I. El Proceso Unificado de Desarrollo de Software. 1ª edición. Madrid: Addison Wesley, 2000. ISBN: 9788478290369.
- Weitzenfeld, A. Ingeniería del Software orientada a objetos con UML, Java e internet. 1ª edición. México: Thomson, 2004. ISBN: 9789706861900.
- Stevens, P.; Pooly, R. Utilización de UML en Ingeniería del Software con Objetos y Componentes. 1ª edición. Madrid: Addison-Wesley, 2007. ISBN: 9788478290864.

Web relacionadas

- Eclipse. https://www.eclipse.org/
- Vsual studio http://www.visualstudio.com/
- VirtualBox (para virtualización del sistema operativo): https://www.virtualbox.org/
- VMPlayer (para virtualización del sistema operativo): https://my.vmware.com/web/vmware/login
- Git. https://www.atlassian.com/es/git/
- Subversion https://subversion.apache.org/
- Scrum https://www.scrum.org/
- Unified Modeling Language: http://www.uml.org/.
- Institute of Electrical and Electronics Engineers: http://www.ieee.org/portal/site.
- Object Management Group. http://www.omg.org/

Recomendaciones para el estudio y la docencia

Esta asignatura muestra el uso y las ventajas de los entornos de desarrollos, introduce y explica los conceptos teóricos necesarios para su entendimiento y la justificación que hay detrás de la toma de las diversas decisiones durante el desarrollo de aplicaciones.

Se recomienda realizar los ejercicios de prácticas asociado a cada tema de la asignatura una vez se tengan adquiridos y comprendidos los conceptos explicados en cada tema práctico, que en ocasiones también harán referencia a la parte teórica de la asignatura.

En relación a los contenidos impartidos en las unidades 4 y 5 sobre modelado con UML, es importante realizar la siguiente aclaración. Pese a que conocer la sintaxis del lenguaje UML puede parecer sencillo y abordable en un corto periodo de tiempo, esto no debe llevar a engaño al alumno. En verdad, la dificultad radica en adquirir una correcta comprensión de cómo emplear todas las herramientas de modelado que ofrece UML, y conocer como extender el lenguaje cuando éste no se adapta a las necesidades particulares del sistema software que se aborda.

Finalmente se recomienda ampliar los conocimientos incluidos en el material didáctico proporcionado por el profesor haciendo uso de las referencias a los capítulos específicos de los libros indicados en la bibliografía y que se incluyen al final del material didáctico de cada tema.

Material necesario

Aplicaciones

El software comercial a utilizar son los entornos de desarrollo Eclipse y Visual Studio, en su versión **Juno**, que es gratuita con fines académicos. Tras registrarse con los datos básicos en la misma web. Los requisitos mínimos de dicha versión son 2 gigas de espacio en disco y 512 megas de RAM

Se debe realizar la instalación sobre una máquina virtual, utilizando software de virtualización como VirtualBox o VMPlayer, cuyas Webs están indicadas en la sección “Web relacionadas”. El alumno que lo desee puede realizar la instalación en otro S.O. (consultar con el profesor en estos casos).

Para el modelado con UML será necesario emplear algún programa que asista el proceso de modelado en UML de sistemas software. Se requiere que la herramienta soporte la versión UML 2.0 y, al menos, los siguientes tipos de diagramas: casos de uso, clases, objetos, secuencia, comunicación, estados, actividad, paquetes y despliegue.

Existe total libertad para que el alumno escoja aquella herramienta que mejor se adapte a sus necesidades. De entre el amplio abanico existente, se sugiere el uso de **Visual Paradigm**, que en su versión *Community Edition* su descarga es gratuita a través del siguiente enlace:

http://www.visual-paradigm.com/download/vpuml.jsp?edition=ce

El software está disponible para distintas plataformas: Microsoft Windows (XP/Vista/7/8), Microsoft Windows Server (2000/2003/2008/2012), Linux, Mac OS X. Los requisitos mínimos del sistema son los siguientes:

- Procesador Intel Pentium 4 a 2.0 GHz o superior.
- 512 MB de RAM. Tamaño recomendable de 1 GB.
- 1GB de espacio libre en disco duro.

Material didáctico

Además de la bibliografía recomendada, en el apartado de Recursos del Campus Virtual se proporcionará al alumno el material didáctico necesario organizado en carpetas por temas para el seguimiento de la asignatura que consistirá en:

- Apuntes sobre cada uno de los temas tratados, con indicaciones específicas a capítulos de libros o manuales en los que se puede profundizar más en los conocimientos expuestos en cada tema.
- Enlaces a páginas Web donde aumentar la información sobre los temas con ejercicios interactivos.
- Ejercicios para practicar y posteriormente las soluciones a los mismos.

Tutorías

Tutoría personal:

Es una ayuda que te ofrece el Instituto Superior de Formación Profesional San Antonio. Consiste en poner a tu disposición una persona, un tutor, dedicada a acompañarte en toda tu etapa como estudiante del Ciclo Formativo. El tutor forma parte del claustro de profesores del ciclo formativo. Los alumnos podrán mantener con su tutor personal una serie de entrevistas personales concertadas cada cierto tiempo. Estas entrevistas no son obligatorias, sino que se plantean como un derecho que tienes el alumno. Es decir, las entrevistas con el tutor personal sólo tendrán lugar si el alumno así lo desea.