|  |  |
| --- | --- |
| **PROGRAMACIÓN DIDACTICA**  **CURSO 2022/23** | |
| **ENTORNOS DE DESARROLLO** | |
| **1º CFGS TÉCNICO EN DESARROLLO DE APLICACIONES WEB**  **Informática y Comunicaciones** | |
| Interfaz de usuario gráfica  Descripción generada automáticamente con confianza media | |
| **ASPIRANTE**  **DAVID HORMIGO RAMÍREZ** | **CUERPO**  **PROFESORES DE ENSEÑANZA SECUNDARIA (590)** |
| **DNI**  **77453948E** | **ESPECIALIDAD**  **INFORMATICA (107)** |
| **NÚMERO DE ASPIRANTE**  **29003350** | **NÚMERO DE TRIBUNAL**  **7** |

Contenido



[Identificación y contexto 5](#_Toc138194182)

[Identificación del módulo profesional 5](#_Toc138194183)

[Marco legislativo completo 5](#_Toc138194184)

[Contextualización del modulo dentro de su entorno productivo 5](#_Toc138194185)

[Justificación 5](#_Toc138194186)

[Objetivos y Competencias 6](#_Toc138194187)

[Objetivos generales del título 6](#_Toc138194188)

[Competencias 7](#_Toc138194189)

[Competencia general 7](#_Toc138194190)

[Competencias profesionales, personales y sociales 8](#_Toc138194191)

[Resultados de aprendizaje y criterios de evaluación 9](#_Toc138194192)

[Unidades didácticas 12](#_Toc138194193)

[Contenidos 12](#_Toc138194194)

[Contenidos básicos 12](#_Toc138194195)

[Contenidos Interdisciplinares 13](#_Toc138194196)

[Temporalización 14](#_Toc138194197)

[Unidades de trabajo 17](#_Toc138194198)

[Introducción al desarrollo del software 17](#_Toc138194199)

[Instalación y uso de entornos de desarrollo integrado 17](#_Toc138194200)

[Ciclo de vida del software 17](#_Toc138194201)

[Defectos de software y depuración 17](#_Toc138194202)

[Control de versiones, repositorios remoto y colaboración 17](#_Toc138194203)

[Documentación de software 17](#_Toc138194204)

[Diagramas de comportamiento 17](#_Toc138194205)

[Diagramas de clase 17](#_Toc138194206)

[Introducción a las pruebas de software 17](#_Toc138194207)

[Pruebas unitarias en Java 17](#_Toc138194208)

[Pruebas de integración en Java 17](#_Toc138194209)

[Refactorización de código 17](#_Toc138194210)

[Metodología 17](#_Toc138194211)

[Evaluación y calificación 17](#_Toc138194212)

[Autoevaluación 17](#_Toc138194213)

[De la programación didáctica 17](#_Toc138194214)

[De la labor docente 17](#_Toc138194215)

[Atención a la diversidad 17](#_Toc138194216)

[Actividades complementarias y extraescolares 17](#_Toc138194217)

[Bibliografía 17](#_Toc138194218)

# Identificación y contexto

## Identificación del módulo profesional

El presente documento desarrolla la programación didáctica para el módulo profesional de Entornos de Desarrollo.

|  |  |
| --- | --- |
| **Familia Profesional** | Informática y Comunicaciones |
| **Nivel** | Ciclo Formativo de Grado Superior |
| **Título** | Desarrollo de aplicaciones web |
| **Referente en la Clasificación Internacional Normalizada de la educación** | Nivel 5 |
| **Nombre del módulo** | Entornos de desarrollo |
| **Código** | 0487 |
| **Equivalencia en créditos ECTS** | 6 |
| **Curso** | Primero |
| **Horas totales** | 96 |
| **Horas semanales** | 3 |

El módulo no tiene unidades de competencia o cualificaciones profesionales asociadas.

## Marco legislativo completo

REAL DECRETO 1147/2011, de 29 de julio, por el que se establece la ordenación general de la formación profesional del sistema educativo (BOE 30/07/2011)

Real Decreto 686/2010, de 20 de mayo, por el que se establece el título de Técnico Superior en Desarrollo de Aplicaciones Web y se fijan sus enseñanzas mínimas.

Orden de 16 de junio de 2011, por la que se desarrolla el currículo correspondiente al título de Técnico Superior en Desarrollo de Aplicaciones Web.

## Contextualización del modulo dentro de su entorno productivo

El Ciclo Formativo de Grado Superior en desarrollo de aplicaciones web tiene como retos profesionales

Dentro del ciclo el módulo profesional de **Entornos de Desarrollo** es el encargado de aportar los conocimientos necesarios para el uso efectivo de las herramientas específicas para el desarrollo de software, la construcción de software fiable y la gestión de documentación de software. La evolución actual de la práctica de la profesión de desarrollador hace vital que

Las líneas de actuación en el proceso de enseñanza-aprendizaje que permite alcanzar los objetivos del módulo versarán sobre:

* Selección y uso de entornos integrados de desarrollo
* Elaboración de documentación de software
* Uso de herramientas de control de versiones
* Utilización de repositorios de software par
* Diseño y ejecución de planes de prueba de software
* Implementación de pruebas unitarias automáticas de software
* Implementación de pruebas automáticas de integración

## Justificación

La evolución de la profesión de desarrolladores de software exige de un ma

Este módulo es de vital importancia en la formación de los desarrolladores de software. El uso efectivo de las diferentes herramientas para el desarrollo de software constituye un

Además, este modulo

# Objetivos y Competencias

## Objetivos generales del título

Los objetivos generales de este ciclo formativo son los siguientes:

1. Ajustar la configuración lógica analizando las necesidades y criterios establecidos para configurar y explotar sistemas informáticos.
2. Identificar las necesidades de seguridad verificando el plan preestablecido para aplicar técnicas y procedimientos relacionados.
3. Instalar módulos analizando su estructura y funcionalidad para gestionar servidores de aplicaciones.
4. Ajustar parámetros analizando la configuración para gestionar servidores de aplicaciones.
5. Interpretar el diseño lógico, verificando los parámetros establecidos para gestionar bases de datos.
6. Seleccionar lenguajes, objetos y herramientas, interpretando las especificaciones para desarrollar aplicaciones web con acceso a bases de datos.
7. Utilizar lenguajes, objetos y herramientas, interpretando las especificaciones para desarrollar aplicaciones web con acceso a bases de datos.
8. Generar componentes de acceso a datos, cumpliendo las especificaciones, para integrar contenidos en la lógica de una aplicación web.
9. Utilizar lenguajes de marcas y estándares web, asumiendo el manual de estilo, para desarrollar interfaces en aplicaciones web.
10. Emplear herramientas y lenguajes específicos, siguiendo las especificaciones, para desarrollar componentes multimedia.
11. Evaluar la interactividad, accesibilidad y usabilidad de un interfaz, verificando los criterios preestablecidos, para Integrar componentes multimedia en el interfaz de una aplicación.
12. Utilizar herramientas y lenguajes específicos, cumpliendo las especificaciones, para desarrollar e integrar componentes software en el entorno del servidor web.
13. Emplear herramientas específicas, integrando la funcionalidad entre aplicaciones, para desarrollar servicios empleables en aplicaciones web.
14. Evaluar servicios distribuidos ya desarrollados, verificando sus prestaciones y funcionalidad, para integrar servicios distribuidos en una aplicación web.
15. Verificar los componentes de software desarrollados, analizando las especificaciones, para completar el plan de pruebas.
16. Utilizar herramientas específicas, cumpliendo los estándares establecidos, para elaborar y mantener la documentación de los procesos.
17. Establecer procedimientos, verificando su funcionalidad, para desplegar y distribuir aplicaciones.
18. Programar y realizar actividades para gestionar el mantenimiento de los recursos informáticos.
19. Analizar y utilizar los recursos y oportunidades de aprendizaje relacionadas con la evolución científica, tecnológica y organizativa del sector y las tecnologías de la información y la comunicación, para mantener el espíritu de actualización y adaptarse a nuevas situaciones laborales y personales.
20. Desarrollar la creatividad y el espíritu de innovación para responder a los retos que se presentan en los procesos y organización de trabajo y de la vida personal.
21. Tomar decisiones de forma fundamentada analizando las variables implicadas, integrando saberes de distinto ámbito y aceptando los riesgos y la posibilidad de equivocación en las mismas, para afrontar y resolver distintas situaciones, problemas o contingencias.
22. Desarrollar técnicas de liderazgo, motivación, supervisión y comunicación en contextos de trabajo en grupo para facilitar la organización y coordinación de equipos de trabajo.
23. Aplicar estrategias y técnicas de comunicación adaptándose a los contenidos que se van a transmitir, la finalidad y a las características de los receptores, para asegurar la eficacia en los procesos de comunicación.
24. Evaluar situaciones de prevención de riesgos laborales y de protección ambiental, proponiendo y aplicando medidas de prevención personales y colectivas, de acuerdo a la normativa aplicable en los procesos del trabajo, para garantizar entornos seguros.
25. Identificar y proponer las acciones profesionales necesarias para dar respuesta a la accesibilidad universal y al diseño para todos.
26. Identificar y aplicar parámetros de calidad en los trabajos y actividades realizados en el proceso de aprendizaje para valorar la cultura de la evaluación y de la calidad y ser capaces de supervisar y mejorar procedimientos de gestión de calidad.
27. Utilizar procedimientos relacionados con la cultura emprendedora, empresarial y de iniciativa profesional, para realizar la gestión básica de una pequeña empresa o emprender un trabajo.
28. Reconocer sus derechos y deberes como agente activo en la sociedad, teniendo en cuenta el marco legal que regula las condiciones sociales y laborales para participar como ciudadano democrático.

## Competencias

### Competencia general

La competencia general de este título consiste en desarrollar, implantar, y mantener aplicaciones web, con independencia del modelo empleado y utilizando tecnologías específicas, garantizando el acceso a los datos de forma segura y cumpliendo los criterios de accesibilidad, usabilidad y calidad exigidas en los estándares establecidos.

El módulo profesional de Entornos de Desarrollo contribuye a alcanzar esta competencia general contribuye al desarrollo e implantación de aplicaciones web haciendo uso de las herramientas, aplicación de métodos y técnicas para asegurar la calidad del software.

### Competencias profesionales, personales y sociales

Las competencias profesionales, personales y sociales de este título son las que se relacionan a continuación:

1. Configurar y explotar sistemas informáticos, adaptando la configuración lógica del sistema según las necesidades de uso y los criterios establecidos.
2. Aplicar técnicas y procedimientos relacionados con la seguridad en sistemas, servicios y aplicaciones, cumpliendo el plan de seguridad.
3. Gestionar servidores de aplicaciones adaptando su configuración en cada caso para permitir el despliegue de aplicaciones web.
4. Gestionar bases de datos, interpretando su diseño lógico y verificando integridad, consistencia, seguridad y accesibilidad de los datos.
5. Desarrollar aplicaciones web con acceso a bases de datos utilizando lenguajes, objetos de acceso y herramientas de mapeo adecuados a las especificaciones.
6. Integrar contenidos en la lógica de una aplicación web, desarrollando componentes de acceso a datos adecuados a las especificaciones.
7. Desarrollar interfaces en aplicaciones web de acuerdo con un manual de estilo, utilizando lenguajes de marcas y estándares web.
8. Desarrollar componentes multimedia para su integración en aplicaciones web, empleando herramientas específicas y siguiendo las especificaciones establecidas.
9. Integrar componentes multimedia en el interface de una aplicación web, realizando el análisis de interactividad, accesibilidad y usabilidad de la aplicación.
10. Desarrollar e integrar componentes software en el entorno del servidor web, empleando herramientas y lenguajes específicos, para cumplir las especificaciones de la aplicación.
11. Desarrollar servicios para integrar sus funciones en otras aplicaciones web, asegurando su funcionalidad.
12. Integrar servicios y contenidos distribuidos en aplicaciones web, asegurando su funcionalidad.
13. Completar planes de pruebas verificando el funcionamiento de los componentes software desarrollados, según las especificaciones.
14. Elaborar y mantener la documentación de los procesos de desarrollo, utilizando herramientas de generación de documentación y control de versiones.
15. Desplegar y distribuir aplicaciones web en distintos ámbitos de implantación, verificando su comportamiento y realizando modificaciones.
16. Gestionar y/o realizar el mantenimiento de los recursos de su área en función de las cargas de trabajo y el plan de mantenimiento.
17. Adaptarse a las nuevas situaciones laborales, manteniendo actualizados los conocimientos científicos, técnicos y tecnológicos relativos a su entorno profesional, gestionando su formación y los recursos existentes en el aprendizaje a lo largo de la vida y utilizando las tecnologías de la información y la comunicación.
18. Resolver situaciones, problemas o contingencias con iniciativa y autonomía en el ámbito de su competencia, con creatividad, innovación y espíritu de mejora en el trabajo personal y en el de los miembros del equipo.
19. Organizar y coordinar equipos de trabajo, supervisando el desarrollo del mismo, con responsabilidad, manteniendo relaciones fluidas y asumiendo el liderazgo, así como, aportando soluciones a los conflictos grupales que se presentan.
20. Comunicarse con sus iguales, superiores, clientes y personas bajo su responsabilidad utilizando vías eficaces de comunicación, transmitiendo la información o conocimientos adecuados, y respetando la autonomía y competencia de las personas que intervienen en el ámbito de su trabajo.
21. Generar entornos seguros en el desarrollo de su trabajo y el de su equipo, supervisando y aplicando los procedimientos de prevención de riesgos laborales y ambientales de acuerdo con lo establecido por la normativa y los objetivos de la empresa.
22. Supervisar y aplicar procedimientos de gestión de calidad, de accesibilidad universal y de diseño para todos, en las actividades profesionales incluidas en los procesos de producción o prestación de servicios.
23. Realizar la gestión básica para la creación y funcionamiento de una pequeña empresa y tener iniciativa en su actividad profesional con sentido de la responsabilidad social.
24. Ejercer sus derechos y cumplir con las obligaciones derivadas de su actividad profesional, de acuerdo con lo establecido en la legislación vigente, participando activamente en la vida económica, social y cultural.

# Resultados de aprendizaje y criterios de evaluación

Según la orden del 16 de julio de 2011, los resultados de aprendizaje del módulo son:

1. 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:

1. Se ha reconocido la relación de los programas con los componentes del sistema informático, memoria, procesador, periféricos, entre otros.
2. Se han clasificado los lenguajes de programación.
3. Se han diferenciado los conceptos de código fuente, objeto y ejecutable.
4. Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales.
5. Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en programación.
6. Se ha diferenciado el funcionamiento de los distintos tipos de traductores de lenguajes ante el código fuente de un programa.
7. Se han identificado las fases de desarrollo de una aplicación informática.

2. Evalúa entornos integrados de desarrollo analizando sus características para editar código fuente y generar ejecutable.

Criterios de evaluación:

1. Se han instalado entornos de desarrollo, propietarios y libres.
2. Se han añadido y eliminado módulos en el entorno de desarrollo.
3. Se ha personalizado y automatizado el entorno de desarrollo.
4. Se ha configurado el sistema de actualización del entorno de desarrollo.
5. Se han generado ejecutables a partir de código fuente de diferentes lenguajes en un mismo entorno de desarrollo.
6. Se han generado ejecutables a partir de un mismo código fuente con varios entornos de desarrollo.
7. Se han identificado las características comunes y específicas de diversos entornos de desarrollo.
8. Se han identificado las funciones más usuales de las herramientas CASE para el desarrollo, prueba y documentación de código.

3. Verifica el funcionamiento de programas diseñando y realizando pruebas.

Criterios de evaluación:

1. Se han identificado los diferentes tipos de pruebas.
2. Se han definido casos de prueba.
3. Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.
4. Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento.
5. Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.
6. Se ha documentado el plan de pruebas.
7. Se han efectuado pruebas unitarias de clases y funciones.
8. Se han efectuado pruebas de integración, de sistema y de aceptación.
9. Se han implementado pruebas automáticas.
10. Se han documentado las incidencias detectadas.
11. Se han aplicado normas de calidad a los procedimientos de desarrollo de software.
12. Se han realizado medidas de calidad sobre el software desarrollado.

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

Criterios de evaluación:

1. Se han identificado los patrones de refactorización más usuales.
2. Se han elaborado las pruebas asociadas a la refactorización.
3. Se ha revisado el código fuente usando un analizador de código.
4. Se han identificado las posibilidades de configuración de un analizador de código.
5. Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno de desarrollo.
6. Se ha realizado el control de versiones integrado en el entorno de desarrollo.
7. Se ha documentado el código fuente mediante comentarios.
8. Se han utilizado herramientas del entorno de desarrollo para documentar los procesos, datos y eventos.
9. Se han utilizado herramientas del entorno de desarrollo para documentar las clases.

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

1. Se han identificado los conceptos básicos de la programación orientada a objetos.
2. Se ha instalado el módulo del entorno integrado de desarrollo que permite la utilización de diagramas de clases.
3. Se han identificado las herramientas para la elaboración de diagramas de clases.
4. Se ha interpretado el significado de diagramas de clases.
5. Se han trazado diagramas de clases a partir de las especificaciones de las mismas.
6. Se ha generado código a partir de un diagrama de clases.
7. Se ha generado un diagrama de clases mediante ingeniería inversa.

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

Criterios de evaluación:

1. Se han identificado los distintos tipos de diagramas de comportamiento.
2. Se ha reconocido el significado de los diagramas de casos de uso.
3. Se han interpretado diagramas de interacción.
4. Se han elaborado diagramas de interacción sencillos.
5. Se han interpretado diagramas de estados.
6. Se han planteado diagramas de estados sencillos.
7. Se ha interpretado el significado de diagramas de actividades.
8. Se han elaborado diagramas de actividades sencillos.

# Unidades didácticas

## Contenidos

### Contenidos básicos

* Reconocimiento de elementos del desarrollo de software:
  + Conceptos de programa informático y de aplicación informática.
  + Concepto de lenguaje de programación.
  + Tipos de lenguajes de programación.
  + Características de los lenguajes más difundidos.
  + Código fuente, código objeto y código ejecutable; máquinas virtuales.
  + Proceso de obtención de código ejecutable a partir del código fuente; herramientas implicadas.
    - Traductores de lenguajes.
    - Depuradores.
  + Fases del desarrollo de una aplicación, análisis, diseño, codificación, pruebas, documentación, explotación y mantenimiento, entre otras.
* Evaluación de entornos integrados de desarrollo:
  + Funciones de un entorno de desarrollo.
  + Componentes de un entorno de desarrollo.
  + Instalación de un entorno de desarrollo.
  + Mecanismo de actualización de un entorno de desarrollo.
  + Uso básico de un entorno de desarrollo.
    - Edición de programas.
    - Generación de ejecutables.
  + Entornos de desarrollo libres y comerciales más usuales.
  + Uso de herramientas CASE en el desarrollo de software.
* Diseño y realización de pruebas:
  + Tipos de pruebas, funcionales, estructurales, regresión.
  + Procedimientos y casos de prueba.
  + Pruebas de Código, cubrimiento, valores límite, clases de equivalencia.
  + Herramientas de depuración de código.
  + Planificación de Pruebas.
    - Pruebas unitarias; herramientas.
    - Pruebas de integración.
    - Pruebas del sistema.
    - Pruebas de aceptación.
    - Automatización de pruebas.
  + Calidad del software.
    - Normas y certificaciones.
    - Medidas de calidad del software.
* Optimización y documentación:
  + Refactorización.
    - Concepto y limitaciones.
    - Patrones de refactorización más usuales.
    - Refactorización y pruebas.
    - Herramientas de ayuda a la refactorización.
  + Control de versiones.
    - Concepto y características.
    - Tipos.
    - Herramientas.
    - Repositorio.
  + Documentación.
    - Uso de comentarios.
    - Alternativas.
    - Herramientas.
* Elaboración de diagramas de clases:
  + Notación de los diagramas de clases.
    - Clases. Atributos, métodos y visibilidad.
    - Objetos. Instanciación.
    - Relaciones. Herencia, composición, agregación, asociación y uso.
  + Herramientas para la elaboración de diagramas de clases. Instalación.
  + Generación de código a partir de diagramas de clases.
  + Generación de diagramas de clases a partir de código.
* Elaboración de diagramas de comportamiento:
  + Tipos. Campo de aplicación.
  + Diagramas de casos de uso. Actores, escenario, relación de comunicación.
  + Diagramas de interacción.
    - Diagramas de secuencia. Línea de vida de un objeto, activación, envío de mensajes.
    - Diagramas de colaboración. Objetos, mensajes.
    - Diagramas de estados. Estados, eventos, señales, transiciones.
    - Diagramas de actividades. Actividades, transiciones, decisiones y combinaciones.

### Contenidos Interdisciplinares

El modulo profesional de entornos de desarrollo está íntimamente ligado al módulo profesional de Programación (0485) y es vital realizar una coordinación en para la consecución de objetivos y el desarrollo de competencias. Ambos módulos comparten ciertos contenidos como son:

* Entornos integrados de desarrollo
  + Definición y tipos. Entornos comerciales y de Software libre.
  + Instalación y descripción de entornos integrados de desarrollo.
  + Creación de proyectos. Estructura y componentes.
* Entornos de desarrollo para programación orientada a objetos.
  + Entornos específicos.
  + Plugins de integración en entornos genéricos.
* Depuración de programas.
  + El depurador como herramienta de control de errores.
  + Documentación de programas.
* Documentación interna, comentarios.
* Documentación externa, diagramas de clases, requisitos, guías, etc.

A su vez, Sistemas Informáticos para el uso de contenedores y máquinas virtuales

Con Lenguaje de Marcas y Sistemas de Información para el uso de HTML y CSS. -🡪 Pruebas de integración sele

### Temporalización

Bloque I. El desarrollo de software.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UT 1: Introducción al desarrollo de Software | | | | |
| CP | OG | RA | Contenidos Propuestos | Contenidos según normativa |
| a,q | a,f,g | 1 | ¿Qué es el desarrollo del software?  Lenguajes de programación compilados e interpretados  Proceso de compilación en C  Ejecutar un programa en Python | Conceptos de programa informático y de aplicación informática.  Concepto de lenguaje de programación.  Tipos de lenguajes de programación.  Características de los lenguajes más difundidos.  Código fuente, código objeto y código ejecutable; máquinas virtuales.  Proceso de obtención de código ejecutable a partir del código fuente; herramientas implicadas.  Traductores de lenguajes. |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UT 2: Instalación y uso de entornos de desarrollo integrado | | | |  |
| CP | OG | RA | Contenidos Propuestos | Contenidos según normativa |
|  |  | 2 | ¿Qué es un IDE?  IDEs generalistas: Visual Studio Code  IDEs para Java: IntelliJ IDEA  Configuración de un IDE  Extensiones de un IDE |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UT 3: Ciclo de vida de software | | | | |
| CP | OG | RA | Contenidos Propuestos | Contenidos según normativa |
|  |  | 1 | Ingeniería del Software  Ciclos de vida en cascada, en espiral e iterativo  Ciclos de vida agiles: Scrum, Programación Extrema, Kanban  Herramientas de gestión de proyectos ágiles: Trello |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UT 4: Defectos de software y depuración | | | | |
| CP | OG | RA | Contenidos Propuestos | Contenidos según normativa |
|  |  | 3 | El concepto de “bug”  Errores en tiempo de compilación  Errores en tiempo de ejecución  Uso del depurador |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UT 5: Control de versiones, repositorios remotos y colaboración | | | | |
| CP | OG | RA | Contenidos Propuestos | Contenidos según normativa |
|  |  | 4 | La importancia del control de versiones  Git.  Uso de Git desde el IDE  El desarrollo cómo tarea coloborativa  Repositorios Remotos: GitHub  Flujos de trabajo más comunes |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UT 6: Documentación de software | | | | |
| CP | OG | RA | Contenidos Propuestos | Contenidos según normativa |
|  |  | 4 | La importancia de la documentación  Documentar requisitos de software  Uso de comentarios  Uso de JavaDoc |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UT 7: Diagramas de comportamiento | | | | |
| CP | OG | RA | Contenidos Propuestos | Contenidos según normativa |
|  |  | 6 | Lenguaje Unificado de Modelado  Herramientas para el modelado  Diagrama de casos de uso  Diagramas de actividad  Diagramas de secuencia |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UT 8: Diagramas de clases | | | | |
| CP | OG | RA | Contenidos Propuestos | Contenidos según normativa |
|  |  | 5 | Diagramas de clase |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UT 9: Introducción a las pruebas de software | | | | |
| CP | OG | RA | Contenidos Propuestos | Contenidos según normativa |
|  |  | 3 | La paradoja de la verificación del software  Tipos de pruebas  Diseño de plan de pruebas  Enfoque actual de pruebas |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UT 10: Pruebas unitarias en Java | | | | |
| CP | OG | RA | Contenidos Propuestos | Contenidos según normativa |
|  |  | 3 | Incorporar JUnit 5 a un proyecto Java  Ejecución de pruebas desde el IDE  La necesidad de aislar los métodos  El uso de Mockito para test-doubles |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UT 11: Pruebas de integración en Java | | | | |
| CP | OG | RA | Contenidos Propuestos | Contenidos según normativa |
|  |  | 3 | Selenium WebDriver  Configuración de Web driver  Desarrollo de pruebas de integración con WebDriver |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| UT 12: Refactorización de código | | | | |
| CP | OG | RA | Contenidos Propuestos | Contenidos según normativa |
|  |  | 4 | El código Legacy y la necesidad de mantenimiento  La regla del boy-scout  Patrones de diseño  Herramientas para refactorización en el IDE |  |

# Unidades de trabajo

## Introducción al desarrollo del software

## Instalación y uso de entornos de desarrollo integrado

## Ciclo de vida del software

## Defectos de software y depuración

## Control de versiones, repositorios remoto y colaboración

## Documentación de software

## Diagramas de comportamiento

## Diagramas de clase

## Introducción a las pruebas de software

## Pruebas unitarias en Java

## Pruebas de integración en Java

## Refactorización de código

# Metodología

# Evaluación y calificación

# Autoevaluación

## De la programación didáctica

## De la labor docente

# Atención a la diversidad

# Actividades complementarias y extraescolares

# Bibliografía

Atlassian. (s.f.). *Getting Git Right*. Obtenido de Bitbucket: https://www.atlassian.com/git/tutorials

Bechtold, S. (s.f.). *JUnit5*. Obtenido de https://junit.org/junit5/

Casado Iglesias, C. (2012). *Entornos de Desarrollo.* Paracuellos del Jarama: RA-MA.

*Dangit, Git!?!* (s.f.). Obtenido de https://dangitgit.com/es

Dudler, R. (01 de 06 de 2023). *git - la guía sencilla*. Obtenido de https://rogerdudler.github.io/git-guide/index.es.html

Faber, S. (s.f.). *Mockito*. Obtenido de https://javadoc.io/doc/org.mockito/mockito-core/latest/org/mockito/Mockito.html

Foundation, E. (s.f.). *Eclipse Documentation*. Obtenido de https://www.eclipse.org/documentation/

Jetbrains. (s.f.). *IntelliJ IDEA*. Obtenido de https://www.jetbrains.com/help/idea/getting-started.html

Microsoft. (s.f.). *Visual Studio Code Docs*. Obtenido de https://code.visualstudio.com/docs

Oracle. (02 de 2004). *Javadoc Tool*. Obtenido de https://www.oracle.com/es/technical-resources/articles/java/javadoc-tool.html

Pressman, R. S. (2010). *Ingenieria del software. Un enfoque práctico.* Ciudad de México: Mc Graw Hill.

Selenium. (7 de 12 de 2021). *WebDriver documentation*. Obtenido de https://www.selenium.dev/documentation/webdriver/

Sourcemaking. (s.f.). *Refactoring*. Obtenido de https://sourcemaking.com/refactoring

Sourcemaking. (s.f.). *UML*. Obtenido de https://sourcemaking.com/uml