

	<b>RESUMEN PROGRAMACIÓN DIDÁCTICA</b> <b>DEPARTAMENTO DE INFORMÁTICA</b>	<b>Curso:</b> <b>2020/21</b>
		<b>Revisión:</b> <b>1</b>
<b>Módulo: Entornos de desarrollo</b> <b>Ciclo: Desarrollo de aplicaciones multiplataforma</b>		

**ÍNDICE:**

0. Identificación del módulo.
1. Criterios de evaluación y calificación del modulo
2. Los resultados de aprendizaje mínimos exigibles para obtener la evaluación positiva en el modulo
3. Plan de refuerzo de los contenidos que no pudieron impartirse el curso pasado

**1.- CRITERIOS DE EVALUACIÓN Y CALIFICACIÓN DEL MÓDULO.**

La calificación se basará en dos pruebas:

1. Examen de preguntas con respuesta múltiple y solución única
2. Realización de los trabajos por parte de los alumnos y su exposición pública

La calificación del examen será de APTO/NO APTO. En caso de obtener una calificación de NO APTO en una evaluación, dicha evaluación se considerará no aprobada, independientemente de la calificación obtenida en la otra prueba. El/la alumno/a podrá realizar un nuevo examen, de las mismas características, en el momento de la realización del examen de la primera y/o segunda prueba ordinaria de junio, para su recuperación, debiendo obtener una calificación de APTO.

Los trabajos a realizar se corresponden de la siguiente manera con las unidades didácticas:

- Trabajo 1: unidad 1
- Trabajo 2: parte teórica de la unidad 2
- Trabajo 3: parte práctica de la unidad 2
- Trabajo 4: unidad 3
- Trabajo 5: unidad 4
- Trabajo 6: unidades 5 y 6

El trabajo, será calificado según una rúbrica específica, que asignará los siguientes pesos a cada criterio de calificación:

10% Presentación

- Participación, conocimiento, compostura...
- Relación con trabajo escrito

5% Debate

- Conocimiento, participación, apoyo en materiales...

20% Trabajo

- Originalidad, unicidad, completitud...

5% Cuestiones

- Realización, originalidad (no copiarse o presentar cuestiones de otros)...

#### 50% Contenidos

- Los que marca el currículo para la parte correspondiente al trabajo

#### 10% Contenidos adicionales

- Otros contenidos adicionales significativos

En caso de que un alumno no participe en la realización de algún trabajo en grupo determinado, deberá realizar un trabajo sobre todos los contenidos de la unidad de trabajo a la que pertenezca el trabajo en el que no ha colaborado, debiendo realizar también una pequeña exposición del mismo al profesor, que demuestre que ha asimilado los contenidos; dicha exposición puede ser un vídeo con locución del alumno.

Si un alumno no puede participar activamente en la exposición pero sí ha lo ha hecho en la elaboración del trabajo, verá mermada su calificación en un 15%, correspondiente a los criterios de presentación y debate. No obstante, si lo solicita al profesor, podrá realizar una pequeña exposición del trabajo, que demuestre que ha asimilados los contenidos; como antes, dicha exposición puede ser un vídeo con locución del alumno. Caso particular de esta situación es aquella en la que alguno/a de los miembros del grupo tenga concedida la conciliación de las vidas laboral y académica, en cuyo caso, se aceptará que ese/a miembro pueda participar en la exposición mediante un vídeo o grabación de su locución.

Si la presentación del trabajo no se realizara por inasistencia de sus componentes, la calificación correspondiente al trabajo será cero para todos los componentes del grupo, debiendo realizar, cada uno de ellos, un trabajo sobre todos los contenidos de la unidad de trabajo a la que pertenezca dicho trabajo y su correspondiente exposición; como antes, dicha exposición puede ser un vídeo con locución del alumno.

Los alumnos que pierdan el derecho a la evaluación continua, desde ese momento, no podrán participar en ningún grupo de trabajo, debiendo realizar trabajos individuales, y su exposición, sobre cada unidad de trabajo e, igualmente, obtener la calificación de APTO en el examen de cada evaluación.

La calificación final del módulo vendrá dada por la media aritmética de la obtenida por el/la alumno/a en las seis unidades, a través de los trabajos realizados, si éste/a ha tenido una calificación de APTO en los tres exámenes correspondientes a las tres evaluaciones. Si la calificación de alguno de los exámenes es NO APTO, la nota que se le asignará será 1.

**2.- RESULTADOS DE APRENDIZAJE MÍNIMOS EXIGIBLES PARA OBTENER LA EVALUACIÓN POSITIVA DEL MÓDULO.**

RESULTADO DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN.	MINIMOS EXIGIBLES
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.	<p>a) Se ha reconocido la relación de los programas con los componentes del sistema informático: memoria, procesador, periféricos, entre otros.</p> <p>b) Se han identificado las fases de desarrollo de una aplicación informática.</p> <p>c) Se han diferenciado los conceptos de código fuente, objeto y ejecutable.</p> <p>d) Se han reconocido las características de la generación de código intermedio para su ejecución en máquinas virtuales.</p> <p>e) Se han clasificado los lenguajes de programación.</p> <p>f) Se ha evaluado la funcionalidad ofrecida por las herramientas utilizadas en programación.</p>	<ul style="list-style-type: none"> <li>- Concepto de programa informático</li> <li>- Código fuente, código objeto y código ejecutable; máquinas virtuales.</li> <li>- Tipos de lenguajes de programación.</li> <li>- Características de los lenguajes más difundidos.</li> <li>- Ingeniería del software. Ciclo de vida. Fases del desarrollo de una aplicación: análisis, diseño, codificación, pruebas, documentación, explotación y mantenimiento, entre otras. Modelos del proceso de desarrollo de software. Agentes participantes.</li> <li>- Arquitectura de una aplicación. Patrones Diseño.</li> <li>- Proceso de obtención de código ejecutable a partir del código fuente; herramientas implicadas.</li> </ul>
Evalúa entornos integrados de desarrollo analizando sus características para editar código	<p>a) Se han instalado entornos de desarrollo, propietarios y libres.</p> <p>b) Se han añadido y eliminado módulos en el entorno de desarrollo.</p> <p>c) Se ha personalizado y automatizado el entorno de desarrollo.</p> <p>d) Se ha configurado el sistema de actualización del entorno de desarrollo.</p> <p>e) Se han generado ejecutables a partir de código fuente de diferentes</p>	<ul style="list-style-type: none"> <li>- Funciones de un entorno de desarrollo. Componentes.</li> <li>- Herramientas y asistentes.</li> <li>- Instalación de un entorno de desarrollo.</li> <li>- Instalación y desinstalación de</li> </ul>

fuente y generar ejecutables.	<p>lenguajes en un mismo entorno de desarrollo.</p> <p>f) Se han generado ejecutables a partir de un mismo código fuente con varios entornos de desarrollo.</p> <p>g) Se han identificado las características comunes y específicas de diversos entornos de desarrollo.</p>	<p>módulos adicionales.</p> <ul style="list-style-type: none"> <li>- Mecanismos de actualización.</li> <li>- Entornos de desarrollo libres y comerciales más usuales.</li> <li>- Uso básico de un entorno de desarrollo:</li> <li>- Edición de programas.</li> <li>- Generación de ejecutables.</li> </ul>
Verifica el funcionamiento de programas diseñando y realizando pruebas.	<p>a) Se han identificado los diferentes tipos de pruebas.</p> <p>b) Se han definido casos de prueba.</p> <p>c) Se han identificado las herramientas de depuración y prueba de aplicaciones ofrecidas por el entorno de desarrollo.</p> <p>d) Se han utilizado herramientas de depuración para definir puntos de ruptura y seguimiento.</p> <p>e) Se han utilizado las herramientas de depuración para examinar y modificar el comportamiento de un programa en tiempo de ejecución.</p> <p>f) Se han efectuado pruebas unitarias de clases y funciones.</p> <p>g) Se han implementado pruebas automáticas.</p> <p>h) Se han documentado las incidencias detectadas.</p>	<ul style="list-style-type: none"> <li>- Planificación de Pruebas.</li> <li>- Tipos de pruebas: Funcionales, estructurales, regresión...</li> <li>- Procedimientos y casos de prueba.</li> <li>- Herramientas de depuración (puntos de ruptura, tipos de ejecución, examinadores de variables, entre otras.)</li> <li>- Validaciones.</li> <li>- Pruebas de Código: Cubrimiento, valores límite, clases de equivalencia, entre otras.</li> <li>- Normas de calidad.</li> <li>- Pruebas unitarias; herramientas.</li> <li>- Automatización de pruebas</li> <li>- Documentación de pruebas.</li> </ul>
Optimiza código empleando las herramientas disponibles en el entorno de desarrollo.	<p>a) Se han identificado los patrones de refactorización más usuales.</p> <p>b) Se han elaborado las pruebas asociadas a la refactorización.</p> <p>c) Se ha revisado el código fuente usando un analizador de código.</p> <p>d) Se han identificado las posibilidades de configuración de un analizador de código.</p> <p>e) Se han aplicado patrones de refactorización con las herramientas que proporciona el entorno</p>	<ul style="list-style-type: none"> <li>- Refactorización. Concepto. Limitaciones. Patrones de refactorización más usuales. Analizadores de código; uso; configuración. Refactorización y pruebas. Herramientas de ayuda a la refactorización.</li> </ul>

	<p>de desarrollo.</p> <p>f) Se ha realizado el control de versiones integrado en el entorno de desarrollo.</p> <p>g) Se han utilizado herramientas del entorno de desarrollo para documentar las clases.</p>	<p>- Control de versiones. Estructura de las herramientas de control de versiones. Repositorio.</p> <p>Herramientas de control de versiones. Clientes de control de versiones integrados en el entorno de desarrollo.</p> <p>- Documentación. Uso de comentarios. Alternativas. Documentación de clases.</p> <p>Herramientas.</p>
<p>Genera diagramas de clases valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas disponibles en el entorno.</p>	<p>a) Se han identificado los conceptos básicos de la programación orientada a objetos.</p> <p>b) Se ha instalado el módulo del entorno integrado de desarrollo que permite la utilización de diagramas de clases.</p> <p>c) Se han identificado las herramientas para la elaboración de diagramas de clases.</p> <p>d) Se ha interpretado el significado de diagramas de clases.</p> <p>e) Se han trazado diagramas de clases a partir de las especificaciones de las mismas.</p> <p>f) Se ha generado código a partir de un diagrama de clases.</p> <p>g) Se ha generado un diagrama de clases mediante ingeniería inversa.</p>	<p>- Clases. Atributos, métodos y visibilidad</p> <p>- Objetos. Instanciación.</p> <p>- Relaciones. Herencia, composición, agregación.</p> <p>- Notación de los diagramas de clases.</p>
<p>Genera diagramas de comportamiento valorando su importancia en el desarrollo de aplicaciones y empleando las herramientas</p>	<p>a) Se han identificado los distintos tipos de diagramas de comportamiento.</p> <p>b) Se ha reconocido el significado de los diagramas de casos de uso.</p> <p>c) Se han interpretado diagramas de interacción.</p> <p>d) Se han elaborado diagramas de interacción sencillos.</p> <p>e) Se ha interpretado el significado de diagramas de actividades.</p> <p>f) Se han elaborado diagramas de actividades sencillos.</p> <p>g) Se han interpretado diagramas de estados.</p> <p>h) Se han planteado diagramas de estados sencillos.</p>	<p>- Tipos. Campo de aplicación</p> <p>- Diagramas de casos de uso. Actores, escenario, relación de comunicación.</p> <p>- Diagramas de secuencia. Línea de vida de un objeto, activación, envío de mensajes</p> <p>- Diagramas de colaboración. Objetos, mensajes.</p>

disponibles en el entorno.		- Diagramas de actividades. Actividades, transiciones, decisiones y combinaciones - Diagramas de estado. Estados, eventos, señales, transiciones.
----------------------------	--	---

### **3.- Plan de refuerzo de los contenidos que no pudieron impartirse el curso pasado**

No se contempla refuerzo al ser un módulo de primer curso.