

	RESUMEN PROGRAMACIÓN DIDÁCTICA DEPARTAMENTO DE INFORMÁTICA	Curso: 2020/21
		Revisión: 1
Módulo: Programación de servicios y procesos Ciclo: Desarrollo de aplicaciones multiplataforma		

ÍNDICE:

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.

Se realizarán dos pruebas objetivas, de carácter teórico/práctico, una por evaluación, que proporcionará la calificación de la respectiva evaluación. La nota final será la media aritmética de las calificaciones de las dos evaluaciones, siempre y cuando la calificación de cada una de las dos evaluaciones sea superior a cuatro (4). Se considerará que el módulo está aprobado cuando la media de las calificaciones alcance o supere el cinco (5).

Aquellos alumnos que no superen una de las evaluaciones podrán realizar un examen de recuperación, que coincidirá en el tiempo con el examen de la evaluación siguiente; en el caso de la segunda evaluación, la prueba de recuperación se realizará coincidiendo con la primera prueba final ordinaria. Si el alumno no se presentara al examen de recuperación u obtuviera una calificación de cero en la misma, no se calcularía la media y se consideraría que la nota final del módulo es uno.

Tanto si no alcanza el cinco de media, como si el/la alumno/a suspende las dos evaluaciones, o si ha obtenido un uno por la circunstancia antes mencionada, o si ha perdido el derecho a la evaluación continua, se examinará de todos los contenidos del módulo en la primera prueba final ordinaria y/o en la segunda prueba final ordinaria.

Debido a la situación de excepcionalidad de este curso, la evaluación, en lugar de mediante la realización de pruebas objetivas, se realizará en base a ejercicios que los alumnos deberán entregar. Queda al criterio del profesor solicitar al alumno que explique detalladamente el trabajo llevado a cabo, que modifique algún aspecto del código o que realice algún ejercicio adicional en situación presencial.

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

RESULTADO DE APRENDIZAJE	CRITERIOS DE EVALUACIÓN.	MINIMOS EXIGIBLES
Desarrolla aplicaciones compuestas por varios procesos reconociendo y aplicando principios de programación paralela	<p>a) Se han reconocido las características de la programación concurrente y sus ámbitos de aplicación.</p> <p>b) Se han identificado las diferencias entre programación paralela y programación distribuida, sus ventajas e inconvenientes.</p> <p>c) Se han analizado las características de los procesos y de su ejecución por el sistema operativo.</p> <p>d) Se han caracterizado los hilos de ejecución y descrito su relación con los procesos.</p> <p>e) Se han utilizado clases para programar aplicaciones que crean subprocesos.</p> <p>f) Se han utilizado mecanismos para sincronizar y obtener el valor devuelto por los subprocesos iniciados.</p> <p>g) Se han desarrollado aplicaciones que gestionen y utilicen procesos para la ejecución de varias tareas en paralelo.</p> <p>h) Se han depurado y documentado las aplicaciones desarrolladas.</p>	<ul style="list-style-type: none"> - Ejecutables. Procesos. Servicios. - Estados de un proceso. - Planificación de procesos por el sistema operativo. - Hilos. - Programación concurrente. - Programación paralela y distribuida. - Creación y finalización de procesos. - Comunicación entre procesos. - Gestión de procesos. - Sincronización entre procesos. - Programación de aplicaciones multiproceso. - Documentación. - Depuración.
Desarrolla aplicaciones compuestas por varios hilos de ejecución analizando y aplicando librerías específicas del lenguaje de programación	<p>a) Se han identificado situaciones en las que resulte útil la utilización de varios hilos en un programa.</p> <p>b) Se han reconocido los mecanismos para crear, iniciar y finalizar hilos.</p> <p>c) Se han programado aplicaciones que implementen varios hilos.</p> <p>d) Se han identificado los posibles estados de ejecución de un hilo y programado aplicaciones que los gestionen.</p> <p>e) Se han utilizado mecanismos para compartir información entre varios hilos de un mismo proceso.</p> <p>f) Se han desarrollado programas formados por varios hilos</p>	<ul style="list-style-type: none"> - Recursos compartidos por los hilos. - Estados de un hilo. Cambios de estado. - Elementos relacionados con la programación de hilos. Librerías y clases. - Creación y finalización de hilos. - Gestión de hilos. - Sincronización de hilos. - Compartición de información entre

	<p>sincronizados mediante técnicas específicas.</p> <p>g) Se ha establecido y controlado la prioridad de cada uno de los hilos de ejecución.</p> <p>h) Se han depurado y documentado los programas desarrollados.</p>	<p>hilos.</p> <ul style="list-style-type: none"> - Prioridades de los hilos. Gestión de prioridades. - Programación de aplicaciones multihilo. - Documentación. - Depuración y rendimiento contra aplicaciones multiproceso.
<p>Programa mecanismos de comunicación en red empleando sockets y analizando el escenario de ejecución</p>	<p>a) Se han identificado escenarios que precisan establecer comunicación en red entre varias aplicaciones.</p> <p>b) Se han identificado los roles de cliente y de servidor y sus funciones asociadas.</p> <p>c) Se han reconocido librerías y mecanismos del lenguaje de programación que permiten programar aplicaciones en red.</p> <p>d) Se ha analizado el concepto de socket, sus tipos y características.</p> <p>e) Se han utilizado sockets para programar una aplicación cliente que se comunique con un servidor.</p> <p>f) Se ha desarrollado una aplicación servidor en red y verificado su funcionamiento.</p> <p>g) Se han desarrollado aplicaciones que utilizan sockets para intercambiar información.</p> <p>h) Se han utilizado hilos para implementar los procedimientos de las aplicaciones relativos a la comunicación en red.</p>	<ul style="list-style-type: none"> - Protocolos de comunicaciones. - Comunicación entre aplicaciones. - Roles cliente y servidor. - Elementos de programación de aplicaciones en red. Librerías. - Sockets. Tipos de sockets. Características. - Creación de sockets. - Enlazado y establecimiento de conexiones. - Utilización de sockets para la transmisión y recepción de información. - Programación de aplicaciones cliente y servidor. - Utilización de hilos en la programación de aplicaciones en red.
<p>Desarrolla aplicaciones que ofrecen servicios en red, utilizando librerías de clases y aplicando criterios</p>	<p>a) Se han analizado librerías que permitan implementar protocolos estándar de comunicación en red.</p> <p>b) Se han programado clientes de protocolos estándar de comunicaciones y verificado su funcionamiento.</p> <p>c) Se han desarrollado y probado servicios de comunicación en red.</p> <p>d) Se han analizado los requerimientos necesarios para crear servicios capaces de gestionar varios clientes concurrentes.</p>	<ul style="list-style-type: none"> - Protocolos estándar de comunicación en red a nivel de aplicación (telnet, ftp, http, pop3, smtp, entre otros). - Librerías de clases y componentes. - Utilización de objetos predefinidos. - Establecimiento y finalización de

de eficiencia y disponibilidad	<p>e) Se han incorporado mecanismos para posibilitar la comunicación simultánea de varios clientes con el servicio.</p> <p>f) Se ha verificado la disponibilidad del servicio.</p> <p>g) Se han depurado y documentado las aplicaciones desarrolladas.</p>	<p>conexiones.</p> <ul style="list-style-type: none"> - Transmisión de información. - Programación de aplicaciones cliente. - Programación de servidores. - Programación de servicios: SOA. - Programación mediante RMI. - Implementación de comunicaciones simultáneas. - Documentación. - Depuración. - Monitorización de tiempos de respuesta y disponibilidad de los servicios generados.
Protege las aplicaciones y los datos definiendo y aplicando criterios de seguridad en el acceso, almacenamiento y transmisión de la información	<p>a) Se han identificado y aplicado principios y prácticas de programación segura.</p> <p>b) Se han analizado las principales técnicas y prácticas criptográficas.</p> <p>c) Se han definido e implantado políticas de seguridad para limitar y controlar el acceso de los usuarios a las aplicaciones desarrolladas.</p> <p>d) Se han utilizado esquemas de seguridad basados en roles.</p> <p>e) Se han empleado algoritmos criptográficos para proteger el acceso a la información almacenada.</p> <p>f) Se han identificado métodos para asegurar la información transmitida.</p> <p>g) Se han desarrollado aplicaciones que utilicen sockets seguros para la transmisión de información.</p> <p>h) Se han depurado y documentado las aplicaciones desarrolladas.</p>	<ul style="list-style-type: none"> - Prácticas de programación segura. - Control de accesos y limitación de privilegios. - Criptografía de clave pública y clave privada. - Principales aplicaciones de la criptografía. - Protocolos criptográficos. - Política de seguridad. - Programación de mecanismos de control de acceso. - Encriptación de información. - Protocolos seguros de comunicaciones. - Programación de aplicaciones con comunicaciones seguras. - Prueba y depuración.

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

Los alumnos deben hacer un trabajo relativo a los resultados de aprendizaje cinco y seis del módulo **Entornos de desarrollo**. Este trabajo debe ser individual y entregado el día 31 de enero del año 2021 o antes. El trabajo no influirá en la calificación del presente módulo, pero éste no podrá superarse si aquél no cumple las condiciones necesarias para el aprobado. Aquellos alumnos que entregaron el curso pasado parte del mismo, pueden reutilizar dicho material, completando las partes ausentes.