

COMPUTACIÓN DISTRIBUIDA EN JAVA

Curso 2011/2012

(Código:31102079)

1.PRESENTACIÓN

En la actualidad la diversificación de los recursos de computación asociados a los diferentes proyectos de desarrollo está íntimamente ligada a la estructura de red disponible y a su evolución. Internet, y el protocolo TCP/IP, marcó un crecimiento exponencial en la disponibilidad de dichos recursos obligando a modificar la visión tradicional del modelo de programación sobre un computador único. El objetivo principal de la asignatura es mostrar las diferentes alternativas de computación distribuida desde el punto de vista práctico, mediante el uso de un lenguaje de programación como Java que dispone de una gran cantidad de ejemplos y desarrollos que el programador y/o analista puede emplear en sus propios proyectos. Para ello, se presentarán las diferentes tecnologías existentes: RMI y CORBA, que representan los ejemplos de middlewares aplicados en el paradigma cliente-servidor, finalizando el estudio con las últimas tendencias en definición e implementación de servicios Web.

2.CONTEXTUALIZACIÓN

La diversificación de los recursos de computación en el momento actual es un hecho que traspasa los límites de la computación tradicional, basada en potentes y caros computadores. Estos recursos se encuentran distribuidos a lo largo de redes de comunicación en diferentes localizaciones, listos para ser usados en diferentes ámbitos. Sin embargo, desde el punto de vista de la programación, la accesibilidad a los recursos implica un conjunto de reglas de acceso que permita de manera transparente el uso de dichos recursos. De esta manera surge de manera natural el concepto de middleware (capa de servicios) que proporciona un conjunto de servicios a los programadores que les permite diseñar y construir aplicaciones software distribuidas en diferentes medios de comunicación, principalmente redes. Esta capa de servicios puede ser muy compleja o casi inexistente, dependiendo del tipo de paradigma de computación distribuida empleado en la construcción del software.

Los middlewares de computación distribuida más conocidos son CORBA (Common Object Request Broker Agent, definido por el OMG) y RMI (Remote Method Invocation, de Sun Microsystems). CORBA se caracteriza por su independencia de lenguaje y plataforma, pero a la vez por su complejidad de implementación. RMI, por el contrario, es más simple pero está limitado al mundo Java, lo que hace que sea una solución más limitada. Ambos middlewares disponen de normas de funcionamiento y de protocolos propios que pueden resultar complejos de comprender y "pesados" desde el punto de vista de la transmisión de información. En la actualidad ha surgido otra tecnología de computación distribuida, la arquitectura SOA (Service Oriented Architecture) que pretende estandarizar las normas de funcionamiento para que la transparencia de ubicación sea completa (independiente de lenguaje de programación y/o sistema operativo) y la comunicación se realice mediante el estándar XML bajo distintas normas (SOAP, WSDL, UDDI),

Esta asignatura pretende centrarse en los conocimientos y mecanismos necesarios para abordar una aproximación profesional al diseño e implementación de soluciones software en el ámbito de la computación distribuida, presentando las bases y conceptos necesarios para usar la terminología adecuada en los diferentes paradigmas existentes. También se realiza una presentación detallada de los diferentes middlewares, CORBA y RMI, incidiendo en dichas tecnologías y presentando sus antecedentes (sockets y RPC). Finalmente se introduce de manera breve y concisa la tecnología SOA y se trabaja en la definición e implementación de servicios Web. Complementa de manera efectiva el perfil profesional del resto de asignaturas sobre computación centrada en la red del módulo VI (Desarrollo de aplicaciones distribuidas basadas en Web) ya que en el resto de asignaturas se cubre información adicional sobre desarrollo con paradigmas como J2EE (DA2) y los



propios servicios Web (DA3).

La distribución temporal de la asignatura se ha adecuado de forma que se premia el componente práctico mediante la realización de varias actividades que complementan de manera concreta diferentes áreas del temario de la asignatura. En concreto para cada módulo de la asignatura se desarrollará un conjunto de preguntas de autoevaluación que se elaborarán en la propia plataforma de formación y que permitirá comprobar la efectividad del aprendizaje. Junto a los cuestionarios se fomentará el uso de los foros de discusión sobre los contenidos teóricos que promueva de forma colaborativa la generación de preguntas frecuentes y debates interesantes sobre diferentes aspectos teóricos. Adicionalmente se han programado varias actividades prácticas que pretenden reforzar el aprendizaje de diferentes tecnologías de computación distribuida, desde sus principios (Sockets) hasta los middlewares más usados como paradigmas cliente-servidor: CORBA y RMI. Para finalizar el curso se deberá realizar un proyecto final que consistirá en una implementación de un caso real de entorno de computación distribuida con aplicación profesional.

De manera general de las competencias definidas para el módulo VI del postgrado, la asignatura cubre las siguientes:

- Ser capaz de desarrollar aplicaciones distribuidas sobre el lenguaje de programación Java.
- Ser capaz de programar sobre redes de comunicaciones mediante el uso de sockets.
- Ser capaz de diseñar aplicaciones multicapa basadas principalmente en el patrón MVC.

De forma particular, se pueden formular las siguientes competencias profesionales particulares que se asumen en esta asignatura y que el alumno deberá tener:

- Adquirir conocimientos generales sobre la computación distribuida, su historia y los paradigmas de computación existentes.
- Ser capaz de comprender los beneficios y desventajas de usar las tecnologías de computación distribuida y su correcta aplicación en los proyectos de desarrollo.
- Reconocer las reglas de funcionamiento de los conectores (sockets) y cómo implementar diferentes protocolos de intercambio de información adaptados al servicio ofrecido por UDP o TCP.
- Discernir las diferentes clasificaciones del componente servidor en el paradigma cliente-servidor, distinguiendo servidores con o sin estado u orientados a conexión
- Adquirir conocimientos sobre objetos distribuidos y las diferentes alternativas de uso de dichos objetos: procedimientos de llamada remota (RPC) o invocación de métodos remotos (RMI)
- Ser capaz de utilizar el middleware RMI para la elaboración de aplicaciones distribuidas basadas en tecnología Java y realizar proyectos con características profesionales aportadas por el propio RMI.
- Comprender la arquitectura general de CORBA y sus diferentes componentes, desarrollando las diferentes capas que conforman la estructura de CORBA, más concretamente, la definición del servicio mediante IDL.
- Reconocer una arquitectura SOA y ser capaz de especificar e implementar un servicio Web simple que emplee las tecnologías SOAP, WSDL y UDDI.
- Reconocer los múltiples métodos de desarrollar aplicaciones distribuidas y seleccionar la mejor opción de rendimiento global.
- Diseñar y gestionar un proyecto que implemente los beneficios de emplear las tecnologías de conectores, CORBA y RMI.

3.REQUISITOS PREVIOS RECOMENDABLES

Se considera imprescindible para la realización y seguimiento del curso, que el alumno posea unos sólidos fundamentos en las siguientes áreas:



- Sistemas distribuidos. Internet constituye el ejemplo más grande de sistema distribuido basado en el protocolo TCP/IP. Los sistemas que emplean el paradigma cliente servidor, como puede ser el protocolo HTTP, se basan en respuestas del servidor a mensajes desde el cliente. En el caso de esta asignatura se usa como base el protocolo HTTP por lo que es recomendable tener un conocimiento profundo del funcionamiento del protocolo y los diferentes servidores que implementan el protocolo (Apache, AOL, IIS).
- Programación con Java. Durante la práctica totalidad del temario de la asignatura, los ejemplos prácticos y las actividades planificadas presuponen un conocimiento medio del lenguaje de programación Java, es decir, la capacidad de desarrollar clases Java que implementen diferentes comportamientos y usen diferentes librerías del propio lenguaje. En particular, la asignatura enseña a emplear ciertas librerías específicas adecuadas para entornos de computación distribuida.
- Redes de computadores. Todo lo relativo a la computación distribuida se centra en el conocimiento del funcionamiento de los diferentes protocolos de comunicación. Es muy importante el conocimiento de la pila de protocolos de TCP/IP, ya que constituyen la base de los protocolos de comunicación de CORBA o RMI, y de los propios sockets.
- Sistemas operativos. Los propios sistemas operativos están pensados como piezas de software que pueden estar distribuidas en diferentes recursos físicos, por tanto, la propia programación de los sistemas operativos es un ejemplo de desarrollo de sistema distribuido. Los middlewares que se estudian en la asignatura utilizan los servicios de los sistemas operativos en los cuales se ejecutan o proporcionan a su vez servicios distribuidos, por lo que es importante conocer detalles específicos sobre los diferentes ámbitos de funcionamiento de un SO (sistemas de ficheros, memoria compartida, etc.).

4.RESULTADOS DE APRENDIZAJE

El objetivo básico de la asignatura es ofrecer una descripción, lo más detallada posible, de los conceptos fundamentales sobre las diferentes tecnologías de desarrollo distribuido que se ofrecen mediante el lenguaje de programación Java. Para ello, se establecen como objetivos específicos los siguientes:

- Realizar un estudio de los fundamentos de los sistemas distribuidos y sus aplicaciones en el desarrollo de software para entornos corporativos.
- Aprender a clasificar los sistemas distribuidos de acuerdo a una tipología de paradigmas, que permite analizar los beneficios y fortalezas de dichos paradigmas de manera general.
- Comprender los conceptos fundamentales relacionados con la programación sobre redes de comunicaciones mediante el uso de sockets.
- Realizar una profunda revisión del modelo cliente servidor, modelo que en la actualidad es usado como paradigma de implementación de las denominadas Aplicaciones Web.
- Entender el concepto de objeto distribuido y las diferentes alternativas de implementación del mismo en Java mediante RMI y CORBA.
- Comprender el concepto de Arquitectura basada en servicios (SOA) y realizar implementaciones de servicios Web mediante SOAP, WDSL y UDDI.

5.CONTENIDOS DE LA ASIGNATURA

La asignatura está dividida en siete módulos que abarcan diferentes aspectos relevantes de la computación distribuida: definición y fundamentos en los que se basa la programación de sistemas distribuidos; paradigmas existentes con especial detalle en el modelo de desarrollo cliente-servidor, desarrollo de soluciones basadas en sockets y con los diferentes middlewares (RMI y CORBA); fundamentos de la arquitectura de servicios Web. El índice detallado para cada uno de los módulos se detalla a continuación:

Módulo 1. Introducción a la computación distribuida.

1.1. Definiciones.



- 1.2. Historia de la computación distribuida.
- 1.3. Tipos de computación distribuida.
- 1.4. Fortalezas y debilidades.
- 1.5. Fundamentos de sistemas operativos.
- 1.6. Fundamentos de red.
- 1.7. Fundamentos de ingeniería del software.

La computación distribuida es en la actualidad la tecnología estándar de desarrollo de aplicaciones basadas en la Web. Para ver las tendencias actuales es preciso hacer un análisis de la historia y fundamentos asociados a este paradigma global de programación, mostrando en detalle los factores que inciden en la propia computación distribuida: la red de comunicación, los sistemas operativos subyacentes y el proceso de desarrollo software.

Módulo 2. Paradigmas de la computación distribuida.

- 2.1. Comunicación entre procesos.
- 2.2. Paradigmas y abstracción.
- 2.3. Aplicación de ejemplo.
- 2.4. Paradigmas para las aplicaciones distribuidas.
- 2.5. Factores a tener en cuenta en la decisión del paradigma.

Existen diferentes aproximaciones a la computación distribuida, denominadas paradigmas. Entre ellas se pueden destacar los modelos de desarrollo basados en modelo cliente-servidor, basados en pares de iguales, basados en mensajes o los modelos más "antiguos" que usaban RPC (aunque en la actualidad hay otro modelo RPC, denominado XML-RPC que se emplea de manera extensa). El módulo pretende mostrar las características más relevantes de cada una de estas aproximaciones.

Módulo 3. Los Sockets Java.

- 3.1. Introducción.
- 3.2. El API de sockets de datagramas (UDP).
- 3.3. El API de sockets de flujo (TCP).
- 3.4. Sockets seguros.

Uno de los primeros modelos de programación distribuida que se emplearon (y que aún sigue en vigor) es el basado en sockets (conectores). Estos conectores son flujos de información entre dos servicios, usualmente en computadores diferentes, que definen su propio protocolo de intercambio de información. El módulo presenta los dos protocolos basados en



conectores: UDP y TCP, finalizando con una presentación al uso de conectores seguros en entornos donde se necesita proteger la información que se transmite entre los servicios.

Módulo 4. El paradigma cliente-servidor.

- 4.1. Definición y características.
- 4.2. Implementación de un servicio de red.
- 4.3. Servidores orientados y no orientados a conexión.
- 4.4. Servidores iterativos y concurrentes.
- 4.5. Servidores con estado.

El paradigma más empleado en la computación distribuida es, sin lugar a duda, el modelo cliente-servidor. En dicho modelo un cliente accede a un servicio proporcionado por un servidor y este le proporciona la información que necesita a través de dicho servicio. Ejemplos de este modelo puede ser la propia Web, a través de su protocolo HTTP donde el servicio ofrecido en el acceso a la información almacenada y la información transmitida es la propia página HTML o equivalente. EL módulo muestra los diferentes componentes involucrados la comunicación y presenta varias clasificaciones de los servidores relativa a diferentes aspectos del paradigma cliente-servidor.

Módulo 5. Objetos distribuidos en Java: RMI.

- 5.1. Paso de mensajes versus Objetos distribuidos.
- 5.2. Arquitectura de un sistema de objetos distribuidos: RPC y RMI.
- 5.3. Invocaciones remotas mediante RMI: Arquitectura y API.
- 5.4. Ejemplo de aplicación RMI.
- 5.5. RMI avanzado.

RMI (Remote Method Invocation) es una capa de servicios distribuidos (búsqueda de objetos remotos, invocación de interfaces y métodos, etc.) que permite desarrollar aplicaciones distribuidas basadas en el modelo cliente-servidor y centradas en la metodología de desarrollo orientada a objetos. El módulo presenta los fundamentos de RMI, además de características avanzadas que pueden ser muy útiles en entornos profesionales.

Módulo 6. CORBA en Java.

- 6.1. Introducción a CORBA.
- 6.2. Java IDL.
- 6.3. Factores de diseño con CORBA.



CORBA, al igual que RMI, proporciona la capa de servicios distribuidos que permite implementar soluciones software distribuidas. La diferencia fundamental radica en no emplear objetos para el desarrollo de la solución (lo que permite usar cualquier tipo de lenguaje de programación) ni un solo lenguaje de programación (RMI usa solo Java), lo que obliga a definir un lenguaje intermedio (IDL, Interface Definition Language) que debe ser traducido al lenguaje que solicitará/implementará el servicio distribuido correspondiente. El módulo muestra los fundamentos de CORBA, su implementación Java mediante Java IDL y los factores de diseño del software distribuido que se deben tener en cuenta a la hora de emplear esta tecnología.

Módulo 7. Servicios Web.

- 7.1. ¿Qué es un servicio Web?
- 7.2. Funcionalidad de transporte de WSA: SOAP.
- 7.3. Funcionalidad de descripción de WSA: WSDL.
- 7.4. Funcionalidad de descubrimiento de WSA: UDDI.
- 7.5. Implementación de la arquitectura WSA.

Los servicios Web son parte de la definición de la arquitectura basada en servicios (SOA). Dicha arquitectura define unas reglas de comportamiento estándar que permiten la definición de servicios mediante XML (WDSL), la búsqueda y localización de servicios (UDDI) y las invocaciones a dichos servicios (SOAP). Todas estas reglas de funcionamiento se implementan mediante el uso de estándares basados en XML, lo que permiten que cualquier tipo marco de desarrollo pueda exportar sus servicios con independencia de la plataforma en la que se desarrollen. El módulo muestra las características de SOA en los ámbitos de comunicación, descripción y descubrimiento.

6.EQUIPO DOCENTE

- [RAFAEL PASTOR VARGAS](#)
- [MIGUEL ROMERO HORTELANO](#)

7.METODOLOGÍA

De forma resumida la metodología docente se concreta en:

- Adaptada a las directrices del EEES.
- La asignatura no tiene clases presenciales. Los contenidos teóricos se impartirán a distancia, de acuerdo con las normas y estructuras de soporte telemático de la enseñanza en la UNED. Esta asignatura se impartirá a distancia, utilizando una plataforma de educación a través de Internet. Se organizarán foros de discusión para dudas y debates.
- El material docente incluye cuestionarios de autoevaluación sobre los contenidos de cada tema y distintos tipos de actividades relacionadas con la asignatura: consulta bibliográfica, consulta de información en Internet, trabajos de análisis y resumen, y uso avanzado de herramientas software.
- Tratándose de un master orientado de forma profesional, las actividades de aprendizaje se estructuran en torno al estado del arte en cada una de las materias del curso y a los problemas en los que se va a focalizar en el trabajo final, sobre el que se realizará la evaluación.

La metodología docente se desarrolla de acuerdo con los siguientes principios:



- Además de adoptar la metodología docente general del programa de postgrado, y en coherencia con el propósito de utilizar los sistemas interactivos de educación con fines pedagógicos y/o formativos, la asignatura diseñada se apoya en gran medida en los recursos educativos de este medio.
- La metodología del trabajo de la asignatura se basa en una planificación temporal de las actividades. Existirán diferentes módulos o unidades didácticas. Cada uno de éstos tendrá asociado unas unidades de aprendizaje y un material asignado (capítulos del libro base, artículos relacionados, direcciones adicionales de Internet, o cualquier otro material que se proporcione). Se asignará un período para cada módulo, en el que deberán realizar las actividades relacionadas con el mismo.

8.BIBLIOGRAFÍA BÁSICA

ISBN(13): 9788478290666

Título: COMPUTACIÓN DISTRIBUIDA: FUNDAMENTOS Y APLICACIONES

Autor/es: Liu, Mei-Ling ;

Editorial: PEARSON ADDISON-WESLEY

Buscarlo en librería virtual UNED

Buscarlo en bibliotecas UNED

Buscarlo en la Biblioteca de Educación

Buscarlo en Catálogo del Patrimonio Bibliográfico

Comentarios y anexos:

Bibliografía recomendada

- Mei-Ling Liu. "Computación distribuida. Fundamentos y Aplicaciones". Ed. Addison Wesley, 2004.
- Apuntes sobre el módulo 7, servicios Web
- Enlaces y recursos sobre los diferentes módulos del curso proporcionados por el equipo docente.

El libro base de Liu recoge la mayor parte del temario de la asignatura y presenta de una forma clara y concisa todos los conceptos clave. En cada capítulo se introducen los conceptos de una manera sencilla, progresiva y acompañados de ejemplos aclaratorios finalizando con un resumen que recoge los conceptos fundamentales presentados al alumno. El único tema no detallado de forma completa en el libro de texto base se corresponde con el módulo 7 y, por tanto, se proporcionará a los alumnos un documento de aproximadamente 15 páginas con el contenido de dicho tema que será complementado con parte del capítulo 11 del libro. Adicionalmente se le entregarán al estudiante una colección de enlaces y recursos relevantes que completarán la bibliografía básica con contenidos de actualidad sobre la programación distribuida.

9.BIBLIOGRAFÍA COMPLEMENTARIA

Comentarios y anexos:

El alumno puede consultar la siguiente bibliografía con el fin de aclarar o extender los conocimientos que debe adquirir a lo largo del curso, y más concretamente en lo concerniente a conectores, RMI y CORBA para la realización de las actividades prácticas:



- Dick Steflik, Prashant Sridharan, Richard Steflik. "Advanced Java Networking" (2nd Ed.) Prentice Hall PTR (2000).

El libro realiza una introducción detallada a las tecnologías de programación en red con Java, abarcando tecnologías como IDL/CORBA, RMI, JDBC/SQL, JavaBeans, JMAPI, servlets y JavaOS, entre otras. Sin embargo, no es sólo un libro para iniciarse en la computación en red con Java porque proporciona al programador la información necesaria para realizar implementaciones sólidas. Por ejemplo, en el caso de RMI, se muestran las ventajas e inconvenientes de la especificación mientras se está desarrollando una aplicación de calendario distribuido.

- David Reilly, Michael Reilly. "Java™ Network Programming and Distributed Computing". Ed. Pearson Ed. 2002.

El libro ha sido escrito como una descripción extensa de las características de computación en red de Java, demostrando porqué este lenguaje es único en este ámbito de la programación. Se muestran y explican los APIs correspondientes además de los conceptos relativos a la programación en red y las capacidades necesarias para ser un programador en red efectivo. Para ello se muestran en profundidad varias características: la arquitectura de Internet y TCP/IP, el sistema de entrada/salida de Java, la tecnología de conectores UDP y TCP para la implementación de clientes y servidores, las ventajas de las aplicaciones multitarea, el protocolo HTTP, las tecnologías Java para el desarrollo de servidores y las tecnologías de computación distribuida CORBA y RMI.

- William Grosso. "Designing and Building Distributed Applications: Java RMI". Ed. O'Reilly Media, Inc.; 1ª edición, 2001.

El texto contiene ejemplos y experiencias en el diseño e implementación de Java RMI, permitiendo a los desarrolladores que se inician en esta tecnología comprobar que RMI es una tecnología muy poderosa y fácil de usar en la programación distribuida. Además, se presentan estrategias de uso para trabajar con la serialización, la multitarea, el registro RMI, conectores y factorías de conectores, activación, descarga dinámica de clases, túneles HTTP, liberación de memoria dinámica distribuida, JNDI y CORBA (todos ellos conceptos avanzados de RMI).

- Darrel Ince. "Developing Distributed and E-Commerce Applications". Ed. Pearson Ed. 2003.

El creciente interés surgido sobre los sistemas distribuidos, sobre todo en el ámbito de las necesidades de negocio de las compañías, y el uso de tecnologías de red, de forma preferencial sobre Internet, está siendo una de las áreas de interés del desarrollo de aplicaciones software. El libro describe el desarrollo de las nuevas tecnologías que han emergido o madurado durante años, como los servidores Web, CORBA, HRML, XML y Java. Además, el libro discute los paradigmas distribuidos y el modelo cliente/servidor antes de comenzar a detallar tecnologías más específicas como ejemplos de estos conceptos. Al final del libro se detallan conceptos avanzados como la seguridad en Internet, la concurrencia, los agentes y la computación ubicua/móvil.

- Max Goff. "Network Distributed Computing". Ed. Pearson Ed. 2004.

Los sistemas de computación distribuida en red (Networked distributed computing, NDC) han sido una revolución tecnológica que se ha expandido con el crecimiento de Internet transformando el mundo en un sistema global de información. El libro



intenta desmitificar las tecnologías mostrando los errores comunes que se cometen al abordar los desarrollos con los nuevos marcos de desarrollo como J2EE, .NET, CORBA o RMI. Además, aborda los protocolos de comunicación y aproximaciones de mensajería para los sistemas distribuidos interoperables. También realiza una descripción detallada de XML, los servicios Web, JINI y otras tecnologías clave.

- Esmond Pitt, Kathleen Mcniff. "JAVA.RMI The Remote Method Invocation Guide". Ed. Pearson Ed. 2001.

El libro proporciona una revisión en profundidad de todas las características de RMI, de manera que sea posible aplicar y comprender la tecnología RMI en entornos profesionales. Contiene información detallada de cómo aplicar RMI para conseguir aplicaciones distribuidas reales, mientras aclara y extiende la información oficial encontrada en la especificación RMI.

10.RECURSOS DE APOYO AL ESTUDIO

Recursos de apoyo

Curso virtual

Para alcanzar todos los objetivos propuestos, el curso se va a articular, como ya se ha comentado, a través de una plataforma especialmente diseñada para facilitar el trabajo colaborativo en Internet (basada en comunidades virtuales), desarrollada por la Sección de Innovación del Centro de Innovación y Desarrollo Tecnológico de la UNED: aLF, ubicada en <http://www.innova.uned.es>.

La plataforma de e-Learning aLF, proporcionará el soporte requerido para gestionar los procesos de enseñanza y aprendizaje, compartir documentos y enlaces de interés, crear y participar en comunidades temáticas y grupos de trabajo específicos, realizar proyectos de diversa naturaleza, organizar el trabajo mediante agendas compartidas e individuales, acceder y publicar noticias de interés, etc.

La plataforma de aprendizaje en Internet permitirá realizar el seguimiento de las actividades del curso, así como estar al tanto de cualquier información o documentación de interés relacionada con el mismo. Para poder utilizar esta plataforma y para mantener un contacto personal con el alumnado se necesitará una dirección de correo electrónico suministrada por el Centro de Servicios Informáticos de la Uned. La filosofía de uso es bien sencilla. Todas las interacciones se hacen a través de enlaces. Por lo tanto, con sólo seguir dichos enlaces se podrá acceder a foros de discusión, documentos de compañeros, etc.

Una vez familiarizados con su uso, es importante tener en cuenta que todas las novedades, instrucciones y actividades se van a publicar utilizando este medio, por tanto, el alumno debe entrar en el grupo frecuentemente para ver si hay alguna novedad en el curso. Si además tiene activados ciertos avisos, podrá recibir notificaciones en el correo electrónico, utilizado para acceder a la plataforma, de los mensajes republicados en los foros, los documentos subidos, las citas puestas en el calendario, por lo que tendrá una información instantánea de todo lo que acontece en la plataforma.

Para comenzar las actividades planificadas es necesario registrarse en la plataforma de aprendizaje y colaboración aLF. Por otro lado, para poder organizar adecuadamente el grupo de trabajo, es necesario conocer cuáles son los conocimientos de partida de los alumnos, preferencias y temas de interés. Por eso, al inicio del curso pondremos disponibles unos



cuestionarios y les pediremos que los rellenen.

Se ofrecerán las herramientas necesarias para que, tanto el equipo docente como el alumnado, puedan compaginar el trabajo individual y el aprendizaje colaborativo.

Software para prácticas.

Se ubicará en la propia plataforma, en el área correspondiente, o bien se darán los enlaces correspondientes de las ubicaciones originales donde descargar tanto el software como los correspondientes manuales.

11.TUTORIZACIÓN Y SEGUIMIENTO

La tutorización de los estudiantes tendrá lugar esencialmente a través de los foros de la plataforma, aunque también podrán utilizarse ocasionalmente otros medios, tales como chats interactivos, servicios de mensajería instantánea y el correo electrónico. Adicionalmente, está también previsto, para temas personales que no afecten al resto de los estudiantes, atender consultas en persona o por teléfono.

El seguimiento del aprendizaje se realizará revisando la participación de los alumnos en los distintos foros de debate, junto con las aportaciones de material nuevo. Además, de la entrega en fecha de los diferentes trabajos prácticos que se han planificado durante la evolución del curso.

12.EVALUACIÓN DE LOS APRENDIZAJES

La evaluación es un aspecto esencial del propio proceso de aprendizaje y como tal se hará uso de la misma. Esto implica que a lo largo del curso, y de acuerdo con la planificación de actividades previstas, el alumno podrá acceder tanto a los resultados de los ejercicios de auto-evaluación propuestos como a las calificaciones y valoraciones de los trabajos presentados en cada tarea y práctica.

La evaluación estará fundamentalmente centrada en la reorientación y motivación del aprendizaje, así como en facilitar la capacidad de auto-comprensión de los conocimientos y las destrezas adquiridas.

Por otro lado, la evaluación es una herramienta esencial para el control de la tarea docente y, en este sentido, se pedirán valoraciones de las tareas propuestas y del propio planteamiento de la docencia de la asignatura. Se prevé al menos una evaluación de este tipo a lo largo del curso.

En cuanto a los ejercicios que requieran trabajo colaborativo, por la propia naturaleza de la asignatura, se aprovecharán especialmente las ventajas que aporta la plataforma de colaboración de la UNED, aLF.



Los criterios de evaluación que se seguirán en las tareas de tipo colaborativo serán los siguientes:

- Garantizar la interdependencia positiva: se valorará tanto la realización de las tareas individuales como las de grupo de forma que el estudiante se sienta motivado para ayudar al resto a alcanzar los objetivos del grupo.
- Capacidad de interacción: se evaluará el grado de interacción y participación en las actividades propuestas.
- Responsabilidad individual y de grupo: se valorará la consecución de los objetivos del grupo y de las tareas individuales en las que ha participado cada miembro del mismo.
- Desarrollo de capacidades de colaboración: se evaluará de forma independiente el aprendizaje de las destrezas asociadas a la resolución de la tarea objeto de las capacidades propias de colaboración.
- Análisis del trabajo de grupo: se evaluará la propia evaluación que los alumnos hagan de la productividad del grupo, distinguiendo el valor relativo de las distintas tareas individuales y de su gestión a lo largo del tiempo, de forma que puedan tomarse medidas de corrección que ayuden a alcanzar los objetivos de tarea y de colaboración previstos.

El alumno deberá entregar una memoria en la que se concreten sus aportaciones en la realización de todas las actividades propuestas.

La evaluación global se calculará de acuerdo al siguiente polinomio:

$$\text{Nota (final)} = [\text{Nota (ejercicios-prácticos)} \times 0.7] + [\text{Nota (trabajo final)} \times 0.3]$$

13.COLABORADORES DOCENTES

Véase equipo docente.

