

|                             |                              |
|-----------------------------|------------------------------|
| <b>Código-Materia:</b>      | 09728 – PROGRAMACIÓN EN RED  |
| <b>Programa – Semestre:</b> | Ingeniería Telemática – 5    |
| <b>Requisitos:</b>          | 09687 – ESTRUCTURAS DE DATOS |
| <b>Período académico:</b>   | 2021-2                       |
| <b>Intensidad semanal:</b>  | 4 horas                      |
| <b>Créditos:</b>            | 3                            |

### **Descripción**

En este curso se busca que cada estudiante aplique los conocimientos adquiridos en los cursos del bloque de algoritmos junto con nuevos conceptos de programación en red para elaborar aplicaciones que trabajen con dispositivos conectados entre sí, empleando como herramienta de trabajo el lenguaje de programación Java en diferentes entornos de desarrollo y despliegue.

Durante el curso, el estudiante dará solución a problemas complejos que requieran elaboración de algoritmos para trabajo en red, adquiriendo con ello conocimientos sobre manejo de direcciones IP, sockets, arquitectura cliente/servidor, administración de bases de datos y utilidades para el manejo de interfaces web.

### **Objetivos**

#### **General:**

Al finalizar el curso los estudiantes estarán en capacidad de desarrollar, de forma individual o como parte de un equipo efectivo de trabajo, una solución distribuida en red con datos persistentes, utilizando un modelo de ciclo de vida para el desarrollo del sistema y los conceptos de programación orientada a objetos.

#### **Terminales:**

Al finalizar el semestre el estudiante estará en capacidad de:

- Explicar el concepto de ciclo de vida, las fases y las actividades que permiten realizar la ingeniería del sistema.
- Aplicar el ciclo de vida para un sistema cuya solución considere persistencia de la información, la arquitectura Cliente/Servidor, el paradigma de programación orientada a objetos y el patrón MVC.
- Evaluar diferentes alternativas para la arquitectura, el diseño y la realización de la solución.
- Efectuar un análisis de causa raíz de un evento o error, utilizando herramientas tales como logs, analizadores de protocolos, debuggers, entre otras.
- Planificar las actividades del ciclo de vida para la ingeniería del sistema considerando las restricciones establecidas, los roles y responsabilidades asociadas (trabajo individual/trabajo en equipo).

## Específicos

### De formación académica:

**Unidad 0:** Ciclo de vida para la ingeniería del sistema y Formación/Operación de equipos de trabajo.

- Identificar el problema concreto que requiere solución y recopilar los datos necesarios para el planteamiento de una solución apropiada.
- Proponer soluciones creativas utilizando las técnicas propuestas por el método de ingeniería y evaluar su factibilidad.
- Identificar métodos y herramientas disponibles para el diseño de la solución de un problema.
- Describir los conceptos básicos de la formación y operación de equipos de trabajo en ingeniería.
- Reconocer los elementos básicos de por lo menos una metodología para la formación y operación de equipos de trabajo en ingeniería.

**Unidad 1:** Fundamentos de Redes.

- Clasificar una red determinada según la distancia geográfica de sus nodos.
- Reconocer la importancia de los modelos de referencia e interpretar la función de cada nivel para explicar el comportamiento de una red.
- Describir la utilidad de los protocolos, direcciones IP y servicios de red para la comunicación efectiva a través de diferentes nodos.
- Desarrollar aplicaciones que consulten y procesen información de red haciendo uso del lenguaje de programación JAVA
- Describir los conceptos básicos de las redes de comunicaciones.
- Reconocer los elementos y el funcionamiento de la arquitectura cliente/servidor.
- Realizar pruebas de conectividad a través de ping (protocolo ICMP).

**Unidad 2:** Cliente/Servidor protocolo TCP.

- Analizar, diseñar e implementar aplicaciones con el patrón MVC (Modelo-Vista-Control)
- Enumerar las principales características del protocolo TCP y describir su funcionamiento en la red.
- Diseñar e implementar programas de arquitectura Cliente/Servidor donde se utilicen objetos de las clases Socket y ServerSocket para transferir información entre dos procesos independientes.
- Identificar errores de conectividad, bloqueo de puertos, envío y recepción de datos a través de la depuración del programa.

**Unidad 3:** Cliente/Servidor protocolo UDP

- Enumerar las principales características del protocolo UDP y describir su funcionamiento en la red diferenciando del protocolo TCP.
- Diseñar e implementar programas donde utilicen objetos de las clases DatagramSocket y DatagramPacket para transferir información entre procesos independientes utilizando UDP.
- Reconocer la diferencia entre los tipos de comunicación: unicast, multicast y broadcast.
- Escribir programas donde se utilice la clase MulticastSocket para realizar comunicaciones multicast.
- Utilizar serialización para simplificar el envío de objetos.

**Unidad 4:** Persistencia

- Reconocer la utilidad del ODBC en la implementación de conectividad a bases de datos.
- Comparar las principales características de los motores de bases de datos más comunes (MySQL, Postgres, Oracle, SQL Server).

- Describir las propiedades de un modelo de conceptos y un modelo relacional al utilizarlos para el análisis y representación de un problema propuesto.
- Diseñar e implementar instrucciones avanzadas en el lenguaje SQL para la manipulación de bases de datos.
- Utilizar logs de SQL para encontrar errores de comunicación y sintaxis entre un programa de Java y bases de datos tipo SQL.
- Integrar herramientas de análisis, diseño e implementación para desarrollar aplicaciones con arquitectura Cliente/Servidor y bases de datos como método de persistencia.
- Diseñar e implementar bases de datos no relacionales como respuesta a un problema de persistencia, en aplicaciones con arquitectura cliente/servidor.

## **Unidad 5: HTTP y Websockets**

- Desarrollar servicios web Restful en Java.
- Pruebas unitarias de servicios web.
- Consumir servicios web desde aplicaciones escritas en Javascript.
- Utilizar servidores de aplicaciones Java para desplegar servicios en ambientes de pruebas
- Implementar aplicaciones con arquitectura Publisher/Subscriber por medio de WebSockets

### De formación en valores y capacidades:

Al terminar el curso el estudiante habrá tenido la oportunidad de trabajar para desarrollar estas competencias generales:

- **Razonamiento analítico y sintético:** Comprender una situación compleja e identificar en ella componentes más simples, estableciendo relaciones lógicas entre ellos (causales o condicionales)
  - Descomponer una situación compleja en elementos simples e identificar las relaciones entre estos.
  - Identificar relaciones causales y deducir conclusiones lógicas.
  - Proceder sistemáticamente para considerar distintas alternativas.
  - Elaborar a partir de distintos elementos de información una lectura coherente de una situación
- **Solución de problemas:** Aplicar estrategias de solución de problemas de manera intencional, tanto en situaciones donde el problema y la solución deseada son claramente evidentes como en situaciones donde el problema y la solución no aparecen estructurados.
  - Estructurar un problema en una situación dada.
  - Analizar, modelar y elaborar diferentes representaciones de una situación problema.
  - Identificar información relevante de un problema analizado.
  - Identificar alternativas de solución y sustentar una selección con criterio profesional.
  - Evaluar la solución dada a un problema, las estrategias utilizadas y el impacto de su implementación en el contexto o situación planteado.

### Metodología

Moodle y GitHub son los medios que contiene la información oficial del curso y es responsabilidad del estudiante consultar en ella todo lo referente al curso, especialmente las actualizaciones del material y actividades.

Para la estructura del curso, se cuenta con dos componentes:

- **Componente Teórica:** Consiste en la discusión de los diferentes temas por medio del desarrollo de hojas de trabajo, exposiciones, informes de investigaciones y mapas conceptuales.

- **Componente Práctica:** Es la resolución de ejercicios y su implementación en sala. Los ejercicios se presentarán en la modalidad de talleres y un proyecto final que consta de 2 entregas parciales y una entrega final.

### **Evaluación**

La evaluación de la materia se divide en nota individual y nota grupal.

La nota individual se compone de:

| <b>Evaluación</b>      | <b>Unidades</b> | <b>Porcentaje</b> |
|------------------------|-----------------|-------------------|
| 1er Parcial práctico   | 2               | 25%               |
| 2do Parcial práctico   | 3,4             | 25%               |
| Talleres               | 0-4             | 10%               |
| Proyecto de Curso      | 0-4             | 40%               |
| <b>Nota individual</b> |                 | <b>100%</b>       |

### **Bibliografía**

La materia no maneja libro guía. A continuación, se listan los recursos a utilizar:

- Redes de Computadoras. Andrew S Tanenbaum. 4ta Edición. Prentice Hall. 2003
- Introducción a las estructuras de datos. Aprendizaje Activo basado en Casos. Jorge A. Villalobos. Prentice Hall
- Fundamentos de Bases de datos. Silberschatz - Korth - Sudarshan. Mc Graw Hill. 2002.
- Java XML tutorial. Mkyong.com. recurso en línea, disponible desde <http://www.mkyong.com/tutorials/java-xml-tutorials/>

### **Profesor**

Domiciano Rincón  
[domiciano.rincon@correounivalle.edu.co](mailto:domiciano.rincon@correounivalle.edu.co)