## TITULO

## ### NetWorking

Es un concepto de conectividad entre 2 o mas dispositivos juntos, que comparten recusos.

Programacion de socnherts proportciona la facilidad de copmártir datos entre diferentes dispositivos.

Ventajas del networking JAVA \* Compartir recursos \* centralizar la gstion del software

Conceptos clave:

DireccionIp: numero dinamico asignado a un nodo de iuna red, compuesto por ovcteros que ban del 0 al 255. Es una direccion logica que pouede sser cambiada.

Protocolo: conunto de reglas para establecer la comunicacion: Ej. FTP, TCP, Telnet, SMTP, POP

Puerto: Esta asociado a la direccion ip para la comunicacion entre aplicaciones.

MAC: (Media access control) Identificador unico de NIC (Network Interface Controller). Un nodo de una red puede tener multiples NIC pero solo uina MAC. Direccion fisica estatica.

Oriented Connection Protocol: Es un protocolo de tipo TCP, TCP/IP, confiable pero lento. El acuse de recibo es enviado por el receptor.

Connection Less Protocol: El receptor no envia el acuse de recibo. Es rrapido, sin embargo no es confgiable, pueden o no llegar a los paquetes. Ej. UDP.

CSocket: Punto final en una conexion bidireccional.

La programacion de sockets se utiliza para la comuinicacion entre alicaciones que se ejecutan en diferentes JREs (Java Runtime Enviroment).

La programacion de Java Socket ouede estar bajo ambos protocolos (OCP, OCL). Las clases Java Socket y ServerSocket se utilizan para la programación orientada a la conexion.

Las clases DatagramSocket y DatagramPacket son usadas para la programación de Socket sin conexión.

El cliente en la programacion necesita saber: \* Ip Address del server. \* Port Number \* Socket Class

## ### Socket Class

El socket es un punto final para las comunicaciones entre dispositivos.

Metodos:

public inputStream getInpútStream() : Regresa el is adjunto con el socket. public OutputStream getOutputStream: Regresa el os adjuntyo al socket.

public syncronized void close(): Cirrra el socket.

Server Socket

Metodos:

public Socket accept(): Establece la conexin entrante. public synchronized void close(): Cierra la conexion del servidor.