

# Experiencias en el aula virtual como mediación pedagógica para el apoyo al aprendizaje en el espacio académico de lenguaje cliente servidor

---

Experiences in the virtual classroom as pedagogical mediation to support learning in the academic space of the client server language



---

## Katherine Roa Banquez

Universidad Santo Tomás, Bogotá, Colombia.

Correo electrónico: [katherinroa@ustadistancia.edu.co](mailto:katherinroa@ustadistancia.edu.co)

## Crisman Martínez Barrera

Universidad Santo Tomás, Bogotá, Colombia.

Correo electrónico: [crismanmartinez@ustadistancia.edu.co](mailto:crismanmartinez@ustadistancia.edu.co)

## Carlos Cabrera Martínez

Universidad Santo Tomás, Bogotá, Colombia.

Correo electrónico: [carlos.cabrera@misena.edu.co](mailto:carlos.cabrera@misena.edu.co)

## Resumen

El presente artículo responde a una serie de preguntas e inquietudes formuladas en el espacio académico Lenguaje Cliente Servidor (LCS), respecto a un tema específico: los sockets de comunicación. El espacio académico LCS pertenece al programa de ingeniería en informática de la Facultad de Ciencias y Tecnologías de la Decanatura de división de educación abierta y a distancia (DUAD) de la Universidad Santo Tomás (USTA).

Para el desarrollo de la investigación se aplicó un test de aprendizaje con el fin de identificar el canal de aprendizaje más influyente en el grupo de estudiantes del espacio académico de Lenguaje Cliente Servidor; de este se identificó que el tipo de aprendizaje audiovisual es el que más prevalecía, por ende, todo diálogo sostenido con los estudiantes por diferentes medios sincrónicos (chat), asincrónicos (foros y correo electrónico) y después de transcurrida una sesión presencial de tutoría, la interacción del docente se daba por medio de ejemplos, donde su respuesta se caracterizaba en la forma de representar la información por medio de ejemplos gráficos, resultando más eficaz y comprensible para los estudiantes, y de esta forma se les facilita el aprendizaje y se les incita a pensar.

**Palabras clave:** aprendizaje, Lenguaje Cliente Servidor, sockets de comunicaciones, modalidad distancia.

## Abstract

This article responds to a series of questions and concerns formulated in the Academic Client Server Language (LCS), regarding a specific topic: communication sockets. The academic space LCS belongs to the computer engineering program of the Faculty of Sciences and Technologies of the Department of Open and Distance Education Division (DUAD) of the University of Santo Tomás (USTA).

For the development of the research, a learning test was applied in order to identify the most influential learning channel in the group of students in the academic area of Client Server Language; from this it was identified that the type of audiovisual learning is the most prevalent, therefore, all dialogue held with students by different synchronous (chat), asynchronous (forums and email) means and after a face-to-face tutoring session, The interaction of the teacher was given by means of examples, where his response was characterized in the way of representing the information through graphic examples, being more effective and understandable for the students, and in this way the learning is facilitated and they are incites to think.

**Keywords:** Learning, Client Server Language, communications sockets, distance mode.

## Introducción

Una de las frases más icónicas de Robinson (2012) en su conferencia *Cambiando paradigmas* fue que “el sistema educativo fue diseñado, concebido y pensado para una época diferente (revolución industrial), donde la economía y la cultura eran completamente diferentes de los estudiantes actuales”; desde este punto de vista, debemos repensar la forma en que continuamos enseñando a nuestros estudiantes y la forma en que ellos realmente aprenden.

Como lo expresa Robinson (2012) los estudiantes del siglo XXI deben ser educados con un pensamiento divergente, el cual consiste en un proceso de pensamiento de generar ideas creativas mediante la exploración de muchas posibles soluciones; sin embargo, hay algo que debemos admitir, nuestras escuelas, institutos y universidades siguen priorizando en su metodología un tipo de pensamiento claramente convergente; donde el modelo actual de educación, está concebido en un conjunto de saberes y competencias que promueven en la mayoría de las ocasiones, comportamientos, habilidades y competencias enajenantes, individualizantes y excluyentes (por fuera de la realidad e intereses de las personas).

Por ende, una de las formas de cambiar esta situación (el modelo actual de educación), es buscar nuevas formas de enseñar y aprender; es así como el presente estudio desarrolló una mediación pedagógica para el apoyo al aprendizaje del espacio académico de Lenguaje Cliente Servidor, donde a través de la aplicación de un test de aprendizaje se identificó el canal de aprendizaje más relevante del grupo de estudiantes, y teniendo en cuenta

este, se establecía la forma de dar respuesta a las dudas de los estudiantes; y de esta forma dar cumplimiento al objetivo de la investigación que es identificar a través de los diferentes medios de comunicación sincrónicos y asincrónicos del aula virtual de aprendizaje, estrategias para el fortalecimiento de los conocimientos adquiridos en el espacio académico de Lenguaje Cliente servidor mediados por la tecnología.

## Lenguaje Cliente Servidor

El concepto de Lenguaje Cliente Servidor se deriva de: Lenguaje y Cliente Servidor que agrupa los conocimientos que fundamentan los conceptos para diseñar, modelar y construir servicios con capacidad de inteligencia, estos servicios pueden o no entregar funcionalidades computacionales propias de la arquitectura Cliente Servidor.

Lenguaje es un conjunto de reglas que dan la capacidad de comunicación entre dos o más elementos. Esta interacción posee como características principales la semántica (conjunto de símbolos agrupados que dan un significado) y la temporalización (símbolos sincronizados en el tiempo).

Cliente/servidor es una arquitectura de red en la que cada ordenador o proceso en la red es cliente o servidor, por ejemplo, las aplicaciones web son un tipo especial de aplicaciones cliente/servidor (Luján-Mora, 2002); por ende, es importante que antes de aprender a programar aplicaciones web, es pertinente conocer las características básicas de las arquitecturas cliente/servidor.

Según Castillo (2018) se puede distinguir dos clases de clientes: los clientes inteligentes (rich client) y los clientes tontos (thin client); los primeros son ordenadores completos, con todo el hardware y software necesarios para poder funcionar de forma independiente, y los segundos, son terminales que no pueden funcionar de forma independiente, ya que necesitan de un servidor para ser operativos.

Esta arquitectura implica la existencia de una relación entre procesos que solicitan servicios (clientes) y procesos que responden a estos servicios (servidores). Estos dos tipos de procesos pueden ejecutarse en el mismo procesador o en distintos (Luján-Mora, 2002).

La arquitectura cliente/servidor permite la creación de aplicaciones distribuidas. La principal ventaja de esta arquitectura es que facilita la separación de las funciones según su servicio, permitiendo situar cada función en la plataforma más adecuada para su ejecución. Además, también presenta las siguientes ventajas: las redes de ordenadores permiten que múltiples procesadores puedan ejecutar partes distribuidas de una misma aplicación, logrando concurrencia de procesos. Existe la posibilidad de migrar aplicaciones de un procesador a otro con modificaciones mínimas en los programas. Se obtiene una escalabilidad de la aplicación. Permite la ampliación horizontal o vertical de las aplicaciones.

Ahora, viéndolo desde el punto de vista del propósito del espacio académico Lenguaje Cliente Servidor, es construir componentes con la arquitectura cliente servidor haciendo uso de las diferentes tecnologías de construcción de aplicaciones y que incorporen técnicas de validación de seguridad de credenciales de

terceros, que usen tecnologías básicas, medias y avanzadas de desarrollo y construcción de aplicaciones web. Entre las técnicas básicas de construcción de soluciones informáticas está la técnica de uso de sockets de comunicaciones, entre las técnicas medias de construcción de aplicaciones está la intermediación de servidores tercerizados y entre las técnicas avanzadas de construcción de aplicaciones está la verificación de seguridad de terceros en dos pasos. Nos enfocamos en nuestro estudio de investigación en el tema que incorpora los sockets de comunicaciones.

En consecuencia, con la abstracción del conocimiento del lenguaje cliente servidor puede el estudiante desarrollar e implementar componentes de software genéricos y/o específicos, para nuestro caso con sockets de comunicaciones, entre muchos otros.

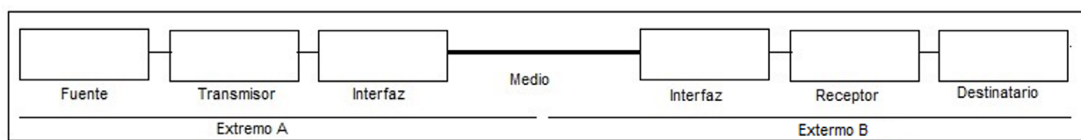
## Socket de comunicaciones

Según Gómez, Machado y Cabrera (2017) un socket es una tecnología de comunicaciones y transferencia de datos que facilita la creación de aplicaciones distribuidas para intercambiar datos entre programas sin requerir entender las diferentes capas de los protocolos de comunicación.

Por lo anterior, es importante reconocer el concepto de comunicación entre sistemas, donde este hace referencia a todo elemento que desee comunicarse con otro dispositivo de la misma o diferente arquitectura en hardware y/o software requiere una interfaz de comunicaciones para poder intercambiar datos. Estas interfaces permiten realizar el cambio

de medio de comunicaciones a otro, como se muestra en la Figura 1.

Figura 1. Arquitectura general de comunicaciones.



Fuente. Elemento generador de datos.

En la arquitectura general de comunicaciones se pueden observar los siguientes elementos:

- Transmisor: elemento que permite procesar la información para poderla difundir. En general siempre realiza alguna transformación en los datos para poderla difundir o irradiar por el medio de comunicaciones.
- Interfaz (extremo A): dispositivo que permite recibir los datos del elemento transmisor para ser transferidos por el elemento medio.
- Interfaz (extremo B): dispositivo que permite recibir los datos del elemento medio para entregarlos al elemento receptor.
- Medio: puede ser visible o no visible. Sistema que permite transportar la información del origen al destino.
- Receptor: elemento que permite procesar la información y realizar la inversa del proceso que realizó el transmisor para entregársela al elemento destinatario.

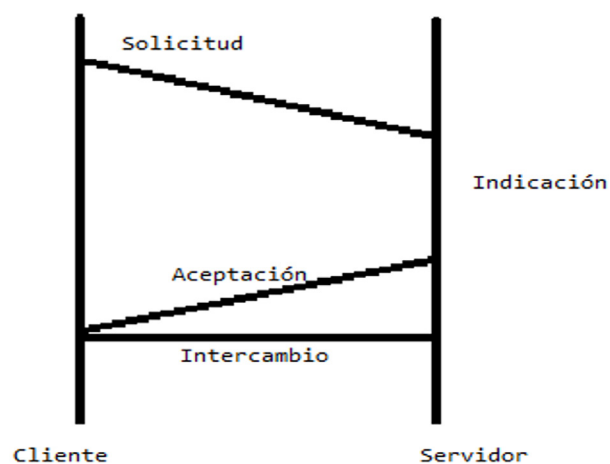
- Destinatario: elemento consumidor de los datos.

En los diferentes momentos cada entidad trasmisora requiere intercambiar datos con el destinatario, quien la convierte en información útil. Estas técnicas de intercambio de datos permiten transferir información de un lugar a otro. Entre las técnicas están los conectores de comunicaciones.

## Conectores de comunicaciones

Corresponde a una tecnología de comunicaciones que permite transportar datos a diferentes componentes con el principio de uso de interfaces que se disponen en el momento de establecer la comunicación, ver Figura 2.

Figura 2. Arquitectura de los conectores de comunicaciones.



El proceso de la arquitectura de conectores de comunicaciones muestra los siguientes procesos (ver Figura 2).

*Solicitud:* el cliente realiza una petición al servidor o servidores de transferencia de información.

*Indicación:* el servidor recibe la solicitud del cliente y de acuerdo a los recursos de hardware, software y/o firmware disponibles acepta o rechaza la solicitud.

*Aceptación:* el servidor envía un ACK (aceptación) al cliente de establecimiento de la comunicación, en caso contrario, algunas veces se envía un NACK (no aceptación) de no aceptación.

*Intercambio:* tanto transmisor (cliente / servidor) como receptor (servidor / cliente) transfieren datos.

## Metodología

El estudio se trabajó bajo un enfoque cualitativo, el cual “utiliza la recolección de datos sin medición numérica para descubrir o afinar preguntas de investigación en el proceso de la investigación” (Hernández, Fernández & Bautista, 2016); es decir, se pretende observar y analizar cualitativamente los efectos producidos de la experiencia de aprendizaje visual en el espacio académico de lenguaje cliente servidor.

## Resultados

Los resultados se presentarán en dos momentos, el primero desde el análisis del test de aprendizaje y el siguiente, desde la experiencia en el aula virtual como mediación pedagógica para el apoyo al aprendizaje en el espacio académico de lenguaje cliente servidor.

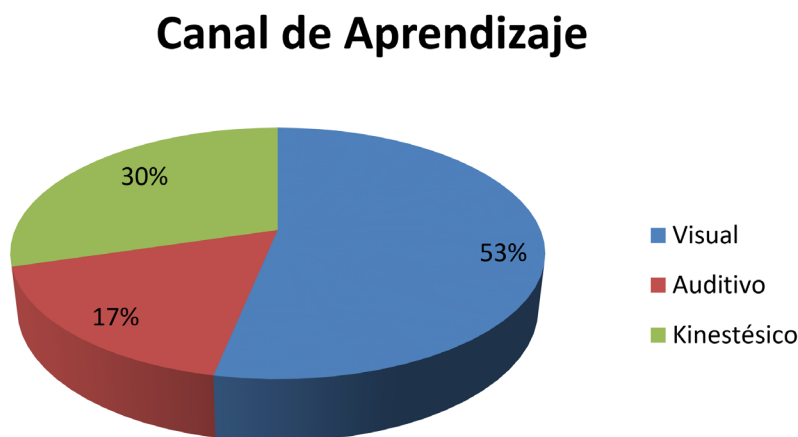
## Análisis del test de aprendizaje

Se aplicó el test del autor O'Brien (1990), para determinar el canal de aprendizaje de preferencia, el cual permite identificar la forma más adecuada en la que un estudiante aprende fácilmente los diferentes conceptos.

Este test se aplicó a un grupo de estudiantes y se identificó que el 53% de los encuestados

aprenden de forma visual, es decir, que su canal de aprendizaje se da por medio de gráficos, mapas conceptuales, mentales, organigramas, o a través de ejemplos muy visuales; el 30% es kinestésico, es decir que aprenden con la manipulación de las cosas, que para nuestro caso puntual, se podría identificar que el programar un lenguaje específico, permitiría afianzar su conocimiento; y finalmente, un 17% es auditivo.

Figura 3. Canal de aprendizaje.



Este test se aplicó a un grupo de estudiantes y se identificó que el 53% de los encuestados aprenden de forma visual, es decir, que su canal de aprendizaje se da por medio de gráficos, mapas conceptuales, mentales, organigramas, o a través de objetos virtuales de aprendizaje que incorporen elementos visuales; el 30% es kinestésico, es decir que aprenden con la manipulación de las cosas, que para nuestro caso puntual, se podría identificar el uso de un laboratorio, práctica o guía para construir una solución informática con el uso de sockets de comunicaciones para el espacio académico lenguaje cliente servidor. Esto permite que el

estudiante afiance su conocimiento en temas específicos; y finalmente, un 17% es auditivo. Ver Figura 5.

Los autores Fernández, Server y Cepero (2001) recomiendan que cuando se trata de formas de aprendizaje “debemos analizar si se trata de cambios e innovaciones en términos de los procesos cognitivos del individuo o de nuevos procedimientos, metodologías y modelos para promover el aprendizaje, aprovechando para ello diversos recursos y estrategias a nuestro alcance”, esta investigación permitió utilizar el análisis del tipo de aprendizaje para

el grupo de estudiantes del espacio académico lenguaje cliente servidor con su tema sockets de comunicación, y sigue afirmando Fernández, Server y Cepero “un elemento importante es la práctica pedagógica tendiente a generar espacios para producir conocimientos a través de diversos métodos; es decir, es la práctica pedagógica el elemento decisivo para hacer los nuevos modelos”

## Experiencia en el aula virtual como mediación pedagógica para el apoyo al aprendizaje.

Una vez identificado el tipo de aprendizaje con los que cuentan los estudiantes en el espacio académico de lenguaje cliente servidor, se procede a analizar cada una de las dudas dejadas en los diferentes espacios de comunicación del aula virtual y así identificar la mejor forma de contestar dichas inquietudes utilizando ejemplos visuales; las problemáticas que se espera que solucionen los estudiantes es construir una aplicación utilizando la arquitectura de sockets.

Para responder a las preguntas formuladas por el estudiante se propuso este artículo, de tal forma que oriente al estudiante nuevamente, después de tener a la mano diferentes recursos OVA en el ambiente virtual.

### *Contexto del espacio académico*

El espacio académico de la experiencia aquí presentada es Lenguaje Cliente Servidor, de naturaleza 100% práctico, donde estas se ca-

racterizan porque “son aquellas que por su dominio de conocimiento requieren un desarrollo equitativo en sus niveles de conceptualización, contextualización y aplicación de las mismas” (Reglamento particular estudiantil de pregrado de la VUAD, 2013).

La finalidad de una de las prácticas en las que se puso a prueba el canal de aprendizaje visual es la construcción de una aplicación utilizando la arquitectura de sockets. A continuación, se presenta un conjunto de inquietudes que se recuperan del foro de dudas del aula virtual, tanto del docente a cargo del espacio académico como del estudiante.

La pregunta formulada por el estudiante es: “¿obligatoriamente se debe manejar el código que usted nos provee o se puede crear otro código que cumpla con la misma función?”. En este caso, el código puesto como ejemplo en el aula virtual realiza la transferencia de datos básicos utilizando el protocolo TCP IP sobre internet. A lo cual se le envió la siguiente respuesta “para la actividad de sockets es necesario que programe con sockets una aplicación de comercio electrónico de elementos de robótica. ...puede basarse en ese o en otro código”. A lo cual el estudiante sutilmente realiza la siguiente pregunta “¿se podría realizar una aplicación tipo chat?” y se le contestó con lo siguiente: “un chat utilizando sockets, no es el tipo de aplicación que se utiliza para transacciones automáticas de comercio electrónico de productos de robótica” y adiciono lo siguiente explícitamente “...por favor, puede realizar antes el análisis de la aplicación que quieres presentar sobre comercio electrónico, posterior a ello, haces el bosquejo (dibujo, diagrama, esquema) de cómo sería el dialogo entre el cliente y el servidor”. Finalmente, retorna



el siguiente comentario: “en ese caso seguiré investigando, pero le soy sincero y no tengo la más mínima idea de cómo realizar una aplicación para transacciones automáticas de comercio electrónico, en lo que he podido apreciar en diversos tutoriales solo muestran aplicaciones simples de escucha cliente servidor, o chats estilo Messenger, igualmente gracias por la respuesta”.

*Respuesta pedagógica del docente:*

Una vez analizada la interacción del estudiante y el docente, el tutor responde y refuerza puntualmente la inquietud del estudiante con el siguiente paper:

A continuación, se presenta el concepto de sockets de comunicaciones y el concepto de comercio electrónico, atributos básicos para desarrollar una aplicación referente a comercio electrónico.

Los sockets corresponden a un principio de intercambio de datos, que cualquier herramienta de desarrollo de aplicaciones de tercera generación soporta, este concepto de conectores de servicios de transferencia de datos se implementa en el nivel de RED del modelo de referencia OSI.

*Comercio electrónico:* son los servicios que se prestan vía electrónica utilizando una red de comunicaciones para el intercambio de bienes, productos y/o servicios. Estos servicios pueden ser completamente automáticos y/o semiautomáticos.

*Primitivas de comunicación:* las primitivas de intercambio, se pueden ver en la Figura 2, donde:

*Solicitud:* el cliente realiza una petición al servidor o servidores de transferencia de información.

*Indicación:* el servidor recibe la solicitud del cliente y de acuerdo a los recursos de hardware, software y/o firmware disponibles acepta o rechaza la solicitud.

*Aceptación:* el servidor envía un ACK (aceptación) al cliente de establecimiento de la comunicación, en caso contrario, algunas veces se envía un NACK (no aceptación) de no aceptación.

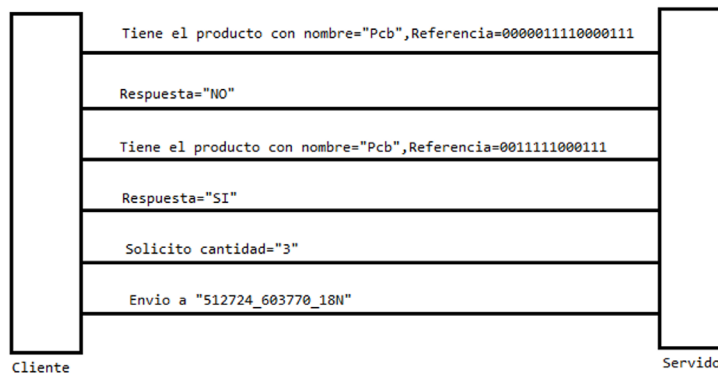
*Intercambio:* tanto transmisor (cliente / servidor) como receptor (servidor / cliente) transfieren datos.

En telefonía se presentan tres fases, establecimiento de la conexión, fase de intercambio y fase de cierre de la comunicación.

### *Diseño del diálogo de la solución informática*

El diseño del diálogo de la solución informática, ver Figura 4, muestra la información que pueden intercambiar, finalmente los elementos cliente y servidor al momento de realizar la aplicación la aplicación y que responde a la mediación del docente con el espacio académico y el espacio académico con el estudiante.

Figura 4. Compra de un producto específico, usando sockets de comunicaciones.



La Figura 4, presenta detalladamente la fase de intercambio de información. Esta fase ocurre una vez se realicen los procesos de solicitud, indicación y aceptación. Posterior a la fase de intercambio de información se realiza el proceso de cierre de conexión. A continuación, se presenta un ejemplo de implementación computacional de sockets de comunicaciones.

### Implementación computacional de los sockets de comunicaciones

En la Figura 5 se muestra un ejemplo de código en Java, se implementa un sockets de comunicación básico y que fundamenta con la práctica la conceptualización teórica del tema sockets de comunicaciones del espacio académico lenguaje cliente servidor.

Figura 5. Código básico en Java de sockets de comunicaciones.

```

    Cliente.java: Bloc de notas
    Archivo Edición Formato Ver Ayuda
    import java.io.*;
    import java.net.*;

    class Cliente
    {
        public Cliente(String host, int puerto)
        {
            Socket skCliente;

            try {
                skCliente = new Socket(host,puerto);
                InputStream aux = skCliente.getInputStream();
                DataInputStream flujo = new DataInputStream(aux);
                System.out.println(flujo.readUTF());
                skCliente.close();
            }catch(Exception e)
            {
                System.out.println(e.getMessage());
            }
        }

        public static void main(String[] arg)
        {
            String host = "172.19.8.46";
            int puerto=4999;

            new Cliente(host,puerto+1);
            new Cliente(host,puerto+2);
        }
    }

    Servidor.java: Bloc de notas
    Archivo Edición Formato Ver Ayuda
    import java.io.*;
    import java.net.*;

    class Servidor
    {
        public Servidor(int clienteActual, int puerto)
        {
            try {
                ServerSocket skServidor = new ServerSocket(puerto);
                System.out.println("Escuchando puerto "+ puerto);
                Socket skCliente=skServidor.accept();
                System.out.println("sirvo al cliente "+clienteActual);
                OutputStream aux = skCliente.getOutputStream();
                DataOutputStream flujo = new DataOutputStream(aux);
                flujo.writeUTF("Hola cliente "+clienteActual);
                skCliente.close();
            }catch(Exception e)
            {
                System.out.println(e.getMessage());
            }
        }

        public static void main(String[] arg)
        {
            int cliente=0, topeCliente=5,puerto=4999;

            new Servidor(cliente+1,puerto+1);
            new Servidor(cliente+2,puerto+2);
        }
    }
    
```

En el código del cliente se aprecia que se requiere la dirección Ip del servidor y un número de puerto de comunicación para conectarse al servidor. El servidor a su vez define un puerto de escucha para las peticiones que se reciben desde el cliente. En TCP (Transfer Control Protocol), IP (Interfaz Protocol), se utiliza el protocolo IP y el protocolo UDP (protocolo de datagrama de usuario), para comunicaciones orientadas a la conexión y no orientadas a la comunicación, respectivamente.

En esta implementación de sockets de comunicaciones se aprecia que se surten las fases de solicitud, indicación, aceptación e intercambio, conceptos fundamentados anteriormente.

### *Implementaciones futuras utilizando sockets de comunicaciones*

En general, las aplicaciones de nivel superior del stack del protocolo que se esté utilizando pueden implementar en su capa de transferencia de datos los sockets de comunicación. Las siguientes pueden ser aplicaciones a nivel de transferencia de datos que pueden implementar este tipo de tecnología.

- Transacciones en línea.
- Transacciones fuera de línea.
- Pagos electrónicos.
- Transferencia de catálogos.
- Transferencia de información entre empresas.

- Transferencia de información entre consumidores.
- Transferencia de información entre entidades del gobierno.
- Transferencia de información entre cosas.
- Automatización de cadenas de valor.
- Interacciones bancarias.
- Transferencia de información aérea.
- Conexión de transferencia segura de datos.
- Puerta de transferencia de datos segura.
- Desarrollos propios y locales de transferencia de datos (Data transfer at home), entre otros.

## Conclusiones

Finalmente, el estudio nos permitió conocer las opiniones de un grupo de estudiantes, donde se evidenció que la forma de llevar la interacción en el aula virtual, les pareció muy acertada, dado que ellos reconocían que a través de ejemplos prácticos y representación de la información, adoptaban más los conceptos y de esta forma la construcción de aplicaciones bajo el lenguaje cliente servidor era más sencillo; así como entender lo que el docente les daba como respuesta a sus inquietudes.

Ahora como docentes del espacio académico, se logró que los estudiantes avanzaran más rápidamente en la temática y en la práctica del tema sockets de comunicación. Igualmente, se espera que esta experiencia se pueda llevar a cabo en otros espacios académicos y así comparar los resultados obtenidos entre estos.

El estudio nos reafirma que es necesario innovar en los espacios académicos tanto presenciales como virtuales y que, en todo proceso desde el momento de elegir la técnica de enseñanza, en la planificación, al momento de escribir el guion, en el desarrollo, en la publicación, en la evaluación y hasta en la retroalimentación se debe innovar.

## Referencias

- Castillo Crespín, J. F. (2018). *Desarrollo de un sistema web y móvil basado en la tecnología cliente servidor para la gestión de asistencia laboral (examen complejo)*. Machala: UTMACH, Unidad Académica de Ingeniería Civil. Recuperado de: <http://repositorio.utmachala.edu.ec/handle/48000/12572>
- Fernández, R., Server, P. & Cepero, E. (2001). El aprendizaje con el uso de las nuevas tecnologías de la información y las comunicaciones. *Revista Iberoamericana de Educación*, 25. Recuperado de: <https://dialnet.unirioja.es/servlet/articulo?codigo=6435591>.
- Flórez, G. (2016). *Diseño de un sistema multiagente de programación de tareas basado en técnicas de inteligencia artificial y simulación discreta para servicio tipo red de colas multi-clase*. Tesis de grado. Maestría en ingeniería industrial. Universidad Distrital Francisco José de Caldas. Bogotá. Recuperado de: <http://repository.udistrital.edu.co/bitstream/11349/4519/4/FlorezBecerraGustavo2016.pdf>
- Gómez-Baryolo, O., Machado-Sosa, A. & Cabrera-Mondeja, A. (2017). Arquitectura de integración basada en socket para sistemas distribuidos. *Espacios*, 38(59). Recuperado de: <https://www.revistaespacios.com/a17v38n59/a17v38n59p06.pdf>
- Luján-Mora, S. (2002). *Programación de aplicaciones web: historia, principios básicos y clientes web*. Alicante: Editorial Club Universitario. Recuperado de: <https://gplsi.dlsi.ua.es/~slujan/programacion-aplicaciones-web-historia-principios-basicos-clientes-web>
- Meza, J. (2017). *Algoritmo de optimización multiobjetivo basado en comportamientos emergentes de enjambres*. Tesis de doctorado. Facultad de Ingeniería. Universidad Distrital Francisco José de Caldas. Bogotá. Recuperado de: <http://repository.udistrital.edu.co/bitstream/11349/5695/1/MezaAlvarezJoaquinJavier2017.pdf>
- O'Brien, L. (1990). *Test para determinar el Canal de Aprendizaje de preferencia*. Recuperado de: <http://www.iafi.com.ar/pnl/ejercicios-pnl/test-canalpreferencia.pdf>
- Robinson K. (2012). *Changing Paradigms*. Recuperado de: [https://www.youtube.com/watch?v=pVYp2\\_Y6Lgw](https://www.youtube.com/watch?v=pVYp2_Y6Lgw)
- Universidad Santo Tomás. (2013). *Reglamento particular estudiantil de pregrado de la*

VUAD. Recuperado de: <https://www.ustadistancia.edu.co/images/documentos/documentos-institucionales/reglamentos/reglamento-estudiantil-pregrado.pdf>

Universidad Santo Tomás (2019). Universidad Abierta y a Distancia. Decanatura de División de Educación Abierta y a Distancia. Facultad de Ciencias y Tecnologías. Aula Virtual - Lenguaje Cliente Servidor. Recuperado de: <http://aula20191.ustadistancia.edu.co>

