

---

# Ejercicio 1

---

Leganés  
Ingeniería Informática  
Sistemas Distribuidos

**uc3m**

---

Universidad  
**Carlos III**  
de Madrid

Alejandro Prieto Macías   NIA 100383428   Gr. 81  
Laura Sánchez Cerro   NIA 100383419   Gr. 81

## Índice

---

<b>1. Introducción</b>	<b>2</b>
<b>2. Modelo</b>	<b>3</b>
2.1. Cliente . . . . .	3
2.1.1. Biblioteca Dinámica . . . . .	3
2.2. Servidor . . . . .	3
2.2.1. Modelo de datos . . . . .	3
2.3. Implementación . . . . .	3
<b>3. Conclusiones</b>	<b>4</b>

## Introducción

---

El motivo de este ejercicio se basa en la idea de poder aprender las técnicas para poder crear una aplicación cliente-servidor de forma concurrente. Además, se pretende aprender a administrar sistema de comunicación para sistemas distribuidos, es decir, para sistemas que no comparten el espacio de memoria.

## Modelo

---

Este problema pretende que el cliente sea capaz de crear vectores de tamaño  $N$ , con un nombre y poder administrarlos a partir de una aplicación. Estos vectores serán almacenados en el servidor con el que se comunica el cliente.

### Cliente

La función del cliente en el sistema es la de utilizar la interfaz gráfica, en este caso el main del archivo *cliente.c* para administrar los datos que el cliente quiere utilizar, es decir, los vectores mencionados anteriormente.

### Biblioteca Dinámica

En la biblioteca dinámica se han implementado las funciones que se comunicarán con el sistema servidor. La inclusión de estructuras de este estilo, es la de facilitar la posible mejora de los sistemas de comunicación sin afectar a la interfaz del usuario, incluso pudiendo tener varias interfaces.

### Servidor

El servidor, será el encargado de recibir las peticiones y que con ayuda de hilos bajo demanda se puedan ir ejecutando las acciones que requiere el cliente para administrar los vectores.

### Modelo de datos

Para administrar la pequeña base de datos que almacena los vectores, se ha decidido establecer una lista enlazada implementada a través de un *struct* del lenguaje de programación **C**.

### Implementación

Las funciones que manejan la lista enlazada se encuentran en un archivo distinto al del propio servidor para facilitar los cambios en el sistema, dando prioridad a la modularización del sistema.

## Conclusiones

---