

# Requisitos, Arquitectura, Diseño Detallado y Código a través de un pequeño ejemplo: registro de clientes, agencia de viajes.

Versión 0.1

Autor: Julio Ariel Hurtado Alegría – Profesor Departamento de Sistemas – Universidad del Cauca

---

## Problema

Se desea mantener una lista de clientes (nombre e identificación) de la agencia de viajes visible al agente que está alimentando al sistema con los clientes con los siguientes requerimientos especiales:

1. El sistema será extendido con muchos elementos estadísticos visuales que requiere que se actualicen permanentemente, es decir se requiere una solución altamente interactiva. Para este caso este requerimiento se

simplifica en tener dos vistas, una con la lista de cliente y otra con un contador de clientes agregados.

2. Cuando el agente no conoce el nombre, pero si la identificación del cliente, entonces puede acceder al servidor de la registraduría para completar la información.

---

## Solución básica a nivel requisitos

En principio se identifica un caso de uso, correspondiente a agregar clientes en el que no se considera la persistencia.

Caso de uso: Agregar Cliente

Descripción: El Agente de viajes desea agregar un cliente al sistema, ingresa la identificación y el nombre, el cliente es creado y agregado a la agencia.

Actores: Agente de viajes, Registraduría

Precondiciones: Lista de Clientes Existente

Poscondiciones: Lista de Clientes V Lista de Clientes U {nuevo\_cliente}

Frecuencia: Media

Flujo Normal de Eventos

---

**Agente de Viajes**

**Sistema Agencia de Viajes**

---

**1**

Despliega interface de  
agregación

	Agente de Viajes	Sistema Agencia de Viajes
	<b>2</b>	Solicita Agregar Nuevo Cliente suministrando nombre del cliente e identificación
	<b>3</b>	Crea Cliente y lo agrega a la lista
	<b>4</b>	Visualiza nueva lista de clientes y el número de clientes registrados

Flujos Alternativos:

En el paso 2 y 3 el Agente sólo suministra la identificación del cliente

	Agente de Viajes	Sistema Agencia de Viajes	Registraduría
	<b>2.a</b>	El agente suministra solo la identificación del cliente	
	<b>2.b</b>	El sistema consulta en la registraduría el nombre suministrando la identificación	
	<b>2.c</b>		La registraduría, si existe un ciudadano con la identificación, devuelve el nombre completo
	<b>3.a</b>	Crea como en el paso 3 sólo si la	

**Agente de Viajes**

**Sistema Agencia de Viajes**

**Registraduría**

---

registraduría  
encuentra un  
ciudadano con la  
identificación

#### Flujos Excepcionales

En el paso 2.c. Excepción 1. No es posible establecer conexión con el servidor. El sistema informa al agente que hay problemas con la conexión.

Suposiciones:

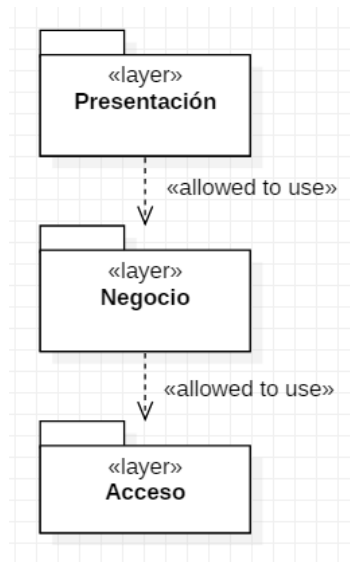
El servidor de la Registraduría trabaja con tecnología TCP/IP a través de sockets.

---

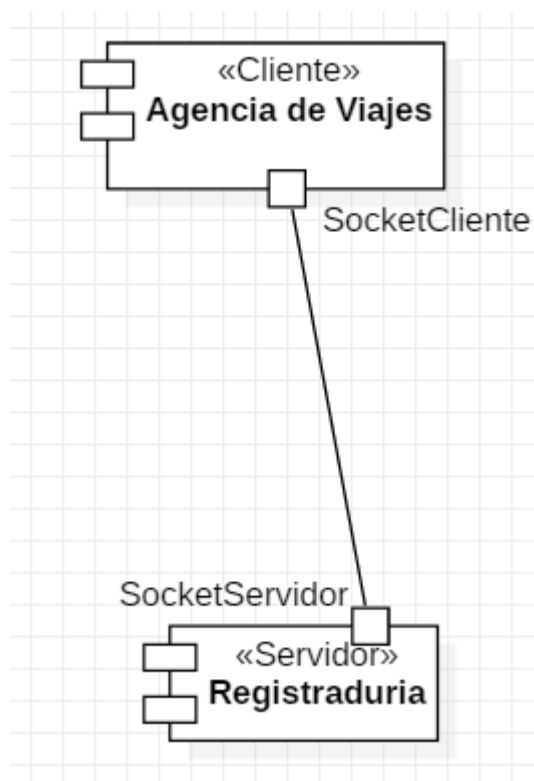
## Solución a nivel diseño arquitectónico

A nivel arquitectónico se identifican cuatro vistas (más la vista de escenarios):

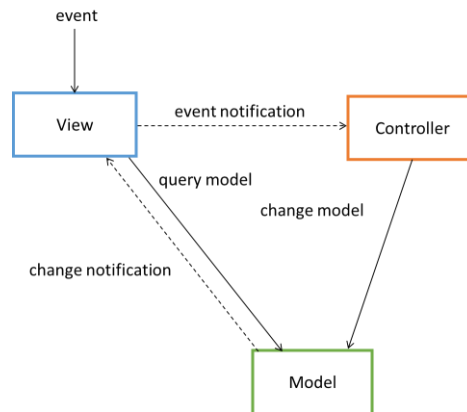
1. Vista de Módulos: se utiliza el estilo de capas permitiendo una separación de preocupaciones entre las lógicas de la presentación (interface gráfica y controladores), el negocio (manejo de la información de clientes) y el acceso (acceso a la información de la registraduría).



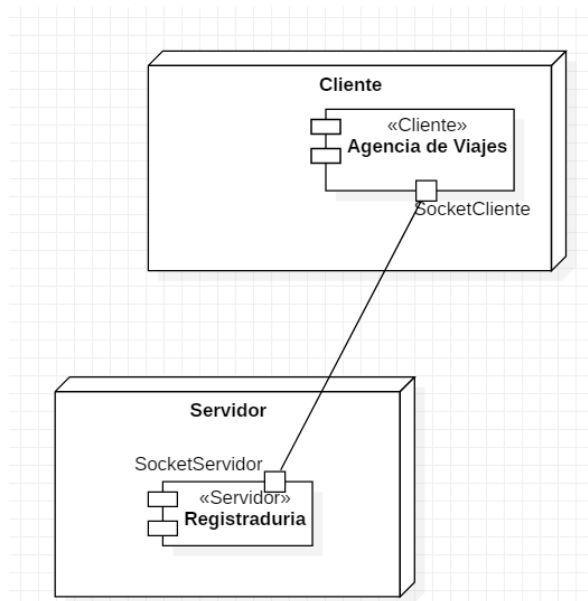
2. Vista de Componentes y Conectores: se utiliza el estilo cliente-servidor. El software de la agencia de viaje es cliente del software de la registraduría que hace el rol de servidor. El conector es una solución basada en sockets TCP/IP.



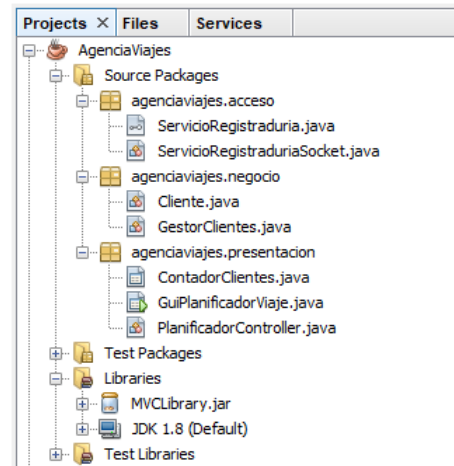
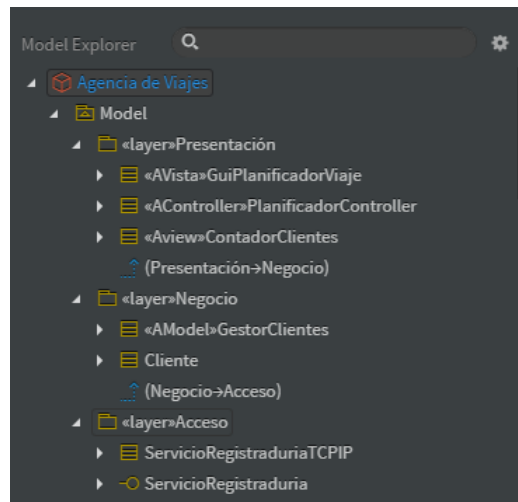
El componente cliente (agencia de viajes) para lograr la interactividad, trabajará bajo la dinámica del patrón de micro-arquitectura MVC y el framework de java.swing.



3. Vista de Instalación: cada componente de la arquitectura cliente-servidor se instala en una respectiva máquina. Ambas máquinas deben tener la máquina virtual de java edición estándar versión 8.



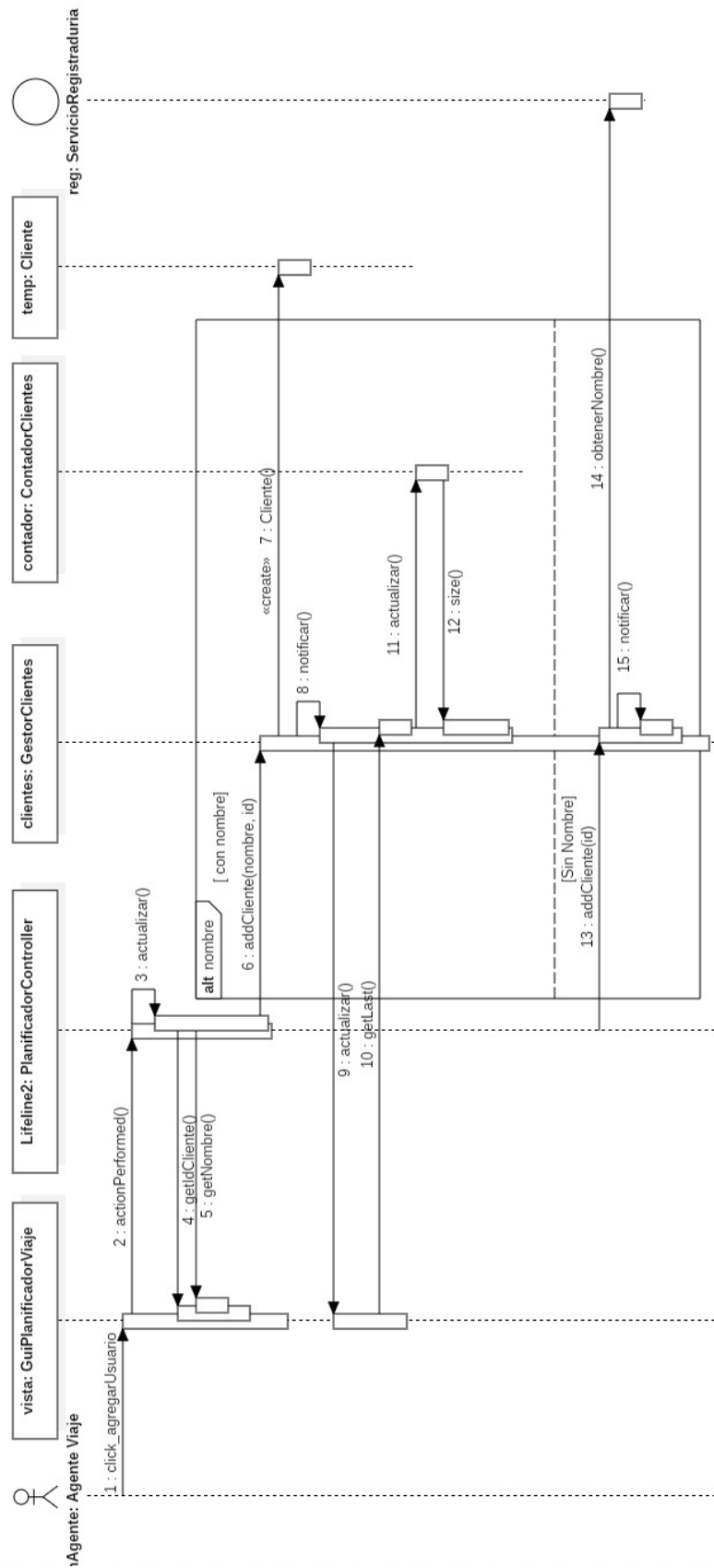
4. Vista de Desarrollo: a nivel de desarrollo, diseño y desarrollo se organizan las clases de acuerdo al empaquetado de la vista de módulos. A nivel de diseño y desarrollo las clases hacen parte del paquete correspondiente. A nivel de desarrollo se pueden ver las dependencias respecto a la librería MVCLibrary desarrollada en clase.



## Solución a nivel diseño detallado

A nivel de diseño detallado se realiza el caso de uso agregar un cliente a través de objetos instancia de clases que hacen parte del sistema. Haciendo uso del paradigma orientado a objetos se parte del diagrama de secuencia y luego se presenta el diagrama de clases unificado.

1. Realización del caso de uso registrar usuario (parte dinámica)



1. Realización del caso de uso registrar usuario (parte estática/estructural)



