

## Caso 2

Se podría utilizar tanto el patron Cliente-Servidor, como el MVC

En el caso del Cliente-Servidor, los componentes interactúan con otros componentes solicitando servicios entre si. Puede haber varias instancias de clientes ejecutándose concurrentemente de modo que como dice el enunciado permite la alta frecuencia de operaciones, de modo que la terminal nunca quede bloqueada

Ventajas: Cambios de funcionalidad centralizados, Testeable, Mantenible.

Desventajas: El servidor puede ser un cuello de botella

En el caso del MVC, que esta compuesta por un modelo (clases que manipulan los datos y reglas de negocios) una vista (Muestran distintas representaciones) y un controlador (maneja logica del programa), permite principalmente la ultima oracion del enunciado, el soporte de multiples formatos de representacion de datos y la flexibilidad para integrar diferentes fabricantes de terminales

Ventajas: Flexibilidad. Ya que es posible tener diferentes vistas para un mismo modelo y es posible construir sin necesidad de modificar modelo existente

Desventajas: Complejidad y el costo actualizaciones (ya que el modelo de las vistas no son totalmente independientes)

