# Modelo de Casos de Uso



## Manejo de usuarios



## Reservación y devolución de vehículos



## Control de empleados



## Manejo de catálogo



## Reporte y facturas



# Modelo de dominio inicial















# Especificación de requerimientos no funcionales

## Usabilidad

Uno de los requerimientos explícitos para el sistema fue un diseño sencillo de usar y adecuado a la imagen institucional, lo cual, además de agregar valor al producto, puede ser un factor importante para el éxito del sistema pues sus potenciales usuarios no requieren un alto nivel de manejo de la computadora y podrían no estar exentos de algún problema físico que impida el uso adecuado del producto. Esto implica que el tiempo de prototipado aumente un 20% y se dediquen mayores recursos -hombre- al análisis y diseño de la interfaz, así mismo, en la etapa de pruebas, validación y verificación se tendrá que dedicar más recursos para evaluar la interfaz, con un 25% de tiempo extra calculado. El tiempo de desarrollo se extenderá 1 semana debido a tales implicaciones y los costos están dentro del presupuesto calculado para el proyecto.

* El tiempo de aprendizaje del sistema no debe ser mayor a 4 horas.
* La taza de errores durante las transacciones cometidas por el usuario debe ser inferiores al 5%.
* Cualquier mensaje debe ser tan explícito como sea posible.
* Debe contemplar la accesibilidad para grupos con alguna deficiencia.
* El diseño debe ser responsivo para garantizar la buena usabilidad en distintos dispositivos.
* La línea de diseño debe poseer colores que contrasten adecuadamente.

Que el sistema sea suficientemente usable provoca un diseño más complejo y con un nivel de abstracción mayor, puede impactar en el tipo de tecnologías seleccionadas, el hardware, tiempos de desarrollo y el modelo planteado, causan una mayor cantidad de clases y uso de patrones y capas de diseño.

## Rendimiento

El modelo web de negocio está teniendo ventajas como ser independiente de la plataforma, por ello, un sitio que esté disponible el 100% del tiempo puede traer muchas ventajas a las empresas. El poseer un servidor con las características para garantizar acceso permanente a los usuarios tiene implicaciones principalmente monetarias y de funcionalidad. Los recursos invertidos para colocar el servidor incrementarán en $100,000.00 más costos de hardware (por seleccionar), dónde un 10% del personal se moverá al área para garantizar cumplir con los plazos planteados. El tiempo de desarrollo del sistema se extenderá 1 mes con el fin de lograr una implementación más cuidada.

* La infraestructura de red deberá cumplir con las normas de la IEEE para conexión de equipos con el fin de garantizar tiempos de respuesta mínimo.
* El sitio web debe ser capaz de atender hasta 100,000 peticiones simultaneas sin pérdida de rendimiento.
* Las transacciones no deben demorar más de 2 segundos en ser realizadas.
* El servidor debe poseer redundancia de datos para garantizar el acceso a cualquier usuario.
* El diseño se considerará exitoso si el 90% de los stakeholders pertinentes lo aprueban.

Esto causa que el diseño se vuelva más específico en las áreas de mayor importancia para lograr los requerimientos y haya -posiblemente- que agregar más vistas para lograr un buen seguimiento.

# Patrón arquitectónico

El patrón arquitectónico que elegiría sería el de Microservicios

Las ventajas que éste patrón ofrece son que permite el despliegue del sistema en unidades separadas, que para un sistema como el planteado, podría ayudarle a la empresa a lograr ganancias pronto gracias a una fácil liberación en las fechas de entrega. El patrón también hace el software fácilmente escalable, proporciona bajo acoplamiento y es distribuida, por ello, si el sistema creciera (tiene el potencial, a mi parecer), el sistema sería fácilmente extensible en software y hardware y su tiempo de operatividad en esos escenarios sería el máximo posible. Su topología basada en API REST permite pequeños servicios expuestos en un sitio web que pueden trabajar en conjunto lo cual ayuda a que el mantenimiento sea mínimo por compartir funcionalidad, pero al mismo tiempo, si algún problema hubiera con un módulo, este no debería afectar a los demás. Finalmente, debido a la baja complejidad de las peticiones que se realizarán al servidor, el problema de la velocidad de las solicitudes puede solventarse con una buena estructura de hardware.

# Diagrama del patrón arquitectónico seleccionado

