

**[AutoServiceExpress]
(SAD) Software Architecture Document
Versión 1.0**

Identificación de Documento

Identificación	
Proyecto	AutoServiceExpress
Versión	1.0

Documento mantenido por	Felipe Carrillo Aburto. Jorge Cavieres Rojas. Cristian González Valenzuela.
Fecha de última revisión	08-11-2024
Fecha de próxima revisión	08-11-2024

Documento aprobado por	
Fecha de última aprobación	

Historia de Revisiones

Fecha	Versión	Descripción	Autor

Tabla de Contenidos

1	3	
1.1	3	
1.2	3	
1.3	¡Error! Marcador no definido.	
1.4	¡Error! Marcador no definido.	
1.5	¡Error! Marcador no definido.	
1.6	3	
2	¡Error! Marcador no definido.	
2.1	3	
3	4	
	A CONTINUACIÓN SE REVISAN LAS METAS Y RESTRICCIONES DE LA ARQUITECTURA.	
3.1	5	6
3.2	5	
3.3	¡Error! Marcador no definido.	
4	¡Error! Marcador no definido.	
4.1	6	
4.2	7	
4.3	Atributos de calidad.	
4.4	11	
5	VISTA LÓGICA	11¡Error! Marcador no definido.
6	26	
7	28	
8	28	
9	28	

1.1 Contexto del Problema
ServiExpress es un taller mecánico para gestionar los servicios de mantenimiento y reparación de

los vehículos de sus clientes.

En la actualidad, el proceso de gestión de clientes, órdenes de pedido y facturación se realiza de forma manual, lo cual genera ineficiencias en términos de tiempo, precisión y capacidad de respuesta. Con el crecimiento del taller y el aumento de clientes derivados de estas compañías, se ha identificado la necesidad de implementar un **sistema de software** que permita automatizar y optimizar estos procesos.

Este sistema busca no solo mejorar la organización interna del taller, sino también ofrecer a sus clientes una forma rápida y segura de acceder a sus datos, gestionar servicios y realizar seguimiento de las órdenes de trabajo en tiempo real.

El propósito de este sistema es crear una plataforma de gestión digital para **ServiExpress** que facilite las operaciones de administración de clientes, órdenes de servicio y facturación, cumpliendo con los requisitos de eficiencia, seguridad y escalabilidad.

El sistema se desarrollará para ser accedido únicamente desde computadoras, permitiendo la interacción de tres tipos de usuarios: clientes, trabajadores y administradores, cada uno con permisos específicos.

●.1 Resumen ejecutivo

●.2 Representación

El sistema **AutoServiceExpress** será documentado y desarrollado utilizando el modelo de cinco vistas de Kruchten (4+1), que permite estructurar la arquitectura del sistema de manera modular y fácil de comprender. Este modelo proporciona cinco perspectivas que cubren diferentes aspectos del diseño del sistema, ayudando a comunicar y entender la arquitectura desde diferentes puntos de vista.

- **Vista Lógica:** Proporciona una representación estructural del sistema, organizando los componentes en módulos y clases. En el caso de AutoServiceExpress, esta vista se centrará en modelar la estructura de los datos y la relación entre entidades como “Cliente”, “Orden de Pedido” y “Boleta/Factura”.
- **Vista de Desarrollo:** Muestra cómo se organizan los componentes del software desde el punto de vista del desarrollador, permitiendo comprender cómo se estructura el código. Para AutoServiceExpress, esto incluye la implementación modular y la organización de archivos y paquetes en el frontend (Angular) y el backend (Django).
- **Vista de Procesos:** Describe el comportamiento dinámico del sistema y las interacciones entre procesos. Esta vista es fundamental para entender la concurrencia y sincronización, lo cual es relevante para gestionar pedidos simultáneos y solicitudes de acceso en tiempo real.
- **Vista Física:** Representa la infraestructura de hardware y la distribución de los componentes del sistema en distintos nodos físicos, describiendo la configuración de servidores, bases de datos y la red. En este caso, se detalla el despliegue en un servidor web y el almacenamiento

en una base de datos MySQL.

- **Vista de Escenarios:** Esta vista conecta todas las anteriores a través de casos de uso clave, asegurando que la arquitectura del sistema soporta los requisitos funcionales y no funcionales en situaciones de uso reales, como la creación de órdenes de servicio o la generación de informes.

● Metas y Restricciones de la Arquitectura

A continuación se revisan las metas y restricciones de la arquitectura.

●.1 Metas de la arquitectura

1. **Rendimiento:** El sistema debe ser capaz de procesar todas las transacciones críticas (como la creación de órdenes de servicio y la consulta de historial) con un tiempo de respuesta inferior a 2 segundos, incluso en momentos de alta demanda.
2. **Disponibilidad:** El sistema debe estar disponible el 99.9% del tiempo durante el horario hábil del taller (lunes a sábado de 8:00 a 20:00 hrs.), para permitir la gestión de pedidos en los momentos de mayor actividad.
3. **Seguridad:** Los datos del cliente y las transacciones deben estar protegidos mediante cifrado de datos (AES-256) y controles de acceso, cumpliendo con las regulaciones de privacidad y seguridad.
4. **Escalabilidad:** La arquitectura debe permitir que el sistema se expanda a medida que el taller crece, permitiendo la incorporación de nuevas sedes sin necesidad de modificar el código base.

●.2 Restricciones de la Arquitectura

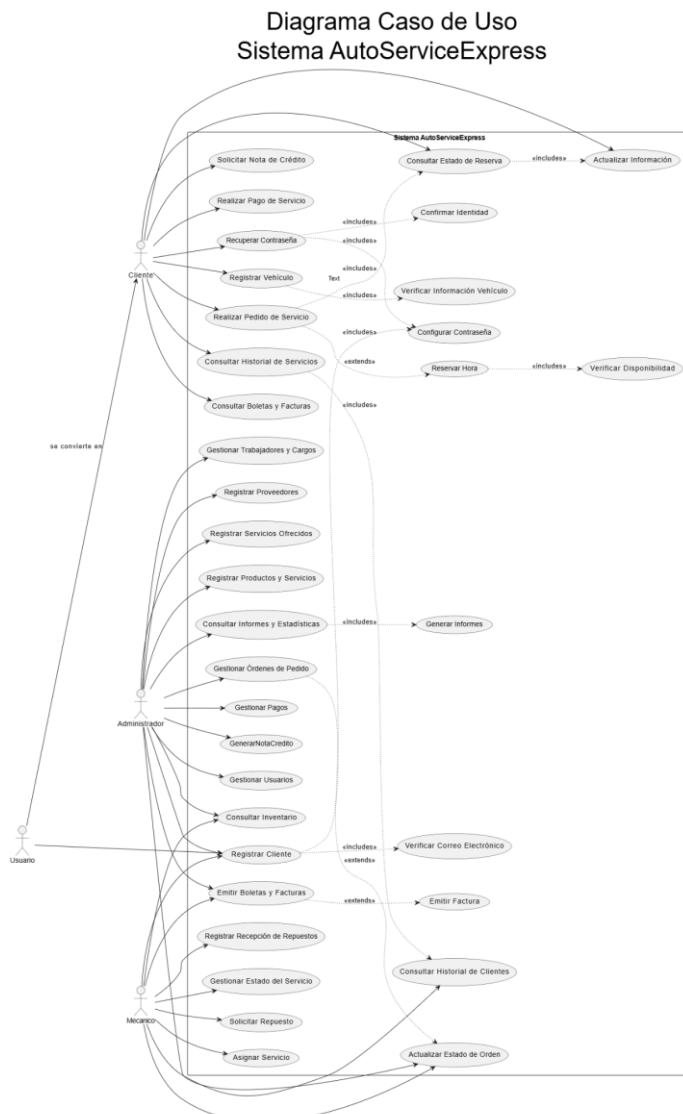
- **Tecnologías de Desarrollo:** El sistema se desarrollará usando Django para el frontend Y backend y MySQL como base de datos. Esto asegura un entorno de desarrollo conocido y soportado por el equipo.
- **Dispositivos de Acceso:** El acceso al sistema estará limitado a computadoras de escritorio, excluyendo dispositivos móviles.
- **Compatibilidad y Conectividad:** Debe asegurar la integración con los sistemas de las compañías de seguros, permitiendo la interoperabilidad para la gestión de datos de clientes.
- **Presupuesto y Tiempo:** La solución debe estar implementada dentro de los plazos y presupuesto establecidos por AutoServiExpress

3. Vista de Escenario y Escenarios de Calidad

Esta sección describe en detalle el conjunto de escenarios funcionales y no funcionales que obtuvieron la mayor prioridad en el análisis. Para esto se presenta y describe el diagrama de casos de uso y los casos de uso prioritarios, así como los escenarios en que uno o más atributos de calidad se ven involucrados de manera significativa.

3.1 Modelo de Casos de Uso

Representa las interacciones clave del usuario con el sistema, destacando las principales funcionalidades, como la gestión de órdenes, reservas y consultas de servicios.



3.2 Especificación de Casos de Uso Relevantes

Los casos de uso considerados los más relevantes para el desarrollo de la arquitectura fueron determinados.

A continuación, se listan los casos de uso relevantes.

Código	Nombre	Actores	Prioridad
CU-001	Registrar Cliente	Usuario, Cliente	Alta
CU-002	Verificar Correo Electrónico	Usuario	Alta
CU-003	Configurar Contraseña	Usuario	Alta
CU-004	Confirmar Identidad	Usuario	Media
CU-005	Recuperar Contraseña	Cliente	Media
CU-006	Registrar Vehículo	Cliente	Media
CU-007	Verificar Información Vehículo	Cliente	Baja
CU-008	Realizar Pedido de Servicio	Cliente	Alta
CU-009	Consultar Historial de Servicios	Cliente	Media
CU-010	Consultar Boletas y Facturas	Cliente	Alta
CU-011	Solicitar Nota de Crédito	Cliente	Media
CU-012	Reservar Hora	Cliente	Alta
CU-013	Verificar Disponibilidad	Cliente	Media
CU-014	Realizar Pago de Servicio	Cliente	Alta
CU-015	Consultar Estado de Reserva	Cliente	Media
CU-016	Actualizar Información	Cliente, Usuario	Media
CU-017	Gestionar Estado del Servicio	Mecanico	Alta
CU-018	Emitir Boletas y Facturas	Mecanico, Administrador	Alta
CU-019	Solicitar Repuesto	Mecanico	Media
CU-020	Consultar Historial de Clientes	Mecanico	Baja
CU-021	Asignar Servicio	Mecanico	Media
CU-022	Registrar Recepción de Repuestos	Mecanico	Media
CU-023	Emitir Factura	Mecanico, Administrador	Alta

CU-024	Gestionar Usuarios	Administrador	Alta
CU-025	Gestionar Trabajadores y Cargos	Administrador	Alta
CU-026	Registrar Proveedores	Administrador	Media
CU-027	Registrar Servicios Ofrecidos	Administrador	Media
CU-028	Registrar Productos y Servicios	Administrador	Media
CU-029	Gestionar Órdenes de Pedido	Administrador, Mecanico	Alta
CU-030	Consultar Informes y Estadísticas	Administrador	Baja
CU-031	Generar Informes	Administrador	Baja
CU-032	Gestionar Pagos	Administrador	Media
CU-033	Consultar Inventario	Administrador, Mecanico	Media
CU-034	Actualizar Estado de Orden	Administrador, Mecanico	Alta
CU-035	Generar Nota de Crédito	Administrador	Media

5. 3.3 Atributos de Calidad

a. 3.3.1 Mantenibilidad

- **Descripción:** El sistema AutoServiceExpress será fácil de mantener gracias a la separación de responsabilidades proporcionada por el modelo MVC. Esta arquitectura permite modificar o actualizar uno de los componentes (Modelo, Vista o Controlador) sin afectar los demás.
- **Criterio de Aceptación:** Los cambios realizados en la lógica de negocio (Modelo) o en la interfaz de usuario (Vista) no deben requerir modificaciones en las otras capas. Se realizarán pruebas de regresión después de cada cambio para asegurar que el sistema sigue funcionando correctamente.

b. 3.3.2 Escalabilidad

- **Descripción:** La arquitectura MVC facilitará la escalabilidad del sistema, permitiendo la adición de nuevos módulos o funcionalidades sin comprometer la estructura existente.
- **Criterio de Aceptación:** Se agregarán nuevos módulos sin necesidad de realizar cambios estructurales en el sistema existente. Las pruebas de carga verificarán que el sistema sigue respondiendo con tiempos menores a 2 segundos bajo condiciones de uso pico.

c. 3.3.3 Modularidad

- **Descripción:** El sistema AutoServExpress será altamente modular, permitiendo que los módulos como la administración de clientes, boletas y órdenes de pedido se desarrollen y mantengan de manera independiente.
- **Criterio de Aceptación:** Cada módulo debe ser independiente, permitiendo su desarrollo y prueba sin afectar a los otros módulos. Se realizarán pruebas para verificar que los módulos funcionan correctamente cuando se despliegan de manera independiente.

d. 3.3.4 Rendimiento

- **Descripción:** El modelo MVC asegura un alto rendimiento en el procesamiento de solicitudes, optimizando el flujo entre la Vista y el Modelo.
- **Criterio de Aceptación:** Las operaciones de generación de boletas y gestión de pedidos deben completarse en menos de 2 segundos. Se realizarán pruebas de carga para medir el tiempo de respuesta bajo condiciones de uso intensivo.

e. 3.3.5 Seguridad

- **Descripción:** La separación de capas en el modelo MVC permite implementar mecanismos de seguridad que aíslan la lógica de negocio y los datos sensibles de los usuarios.
- **Criterio de Aceptación:** La información de los clientes y las transacciones serán protegidas mediante encriptación y controles de acceso. Las auditorías de seguridad deben confirmar que los datos están protegidos contra accesos no autorizados.

f. 3.3.6 Reusabilidad

- **Descripción:** La lógica de negocio separada del modelo MVC permitirá la reutilización de componentes del sistema en otros módulos o proyectos.
- **Criterio de Aceptación:** Los componentes del sistema podrán reutilizarse en otros módulos sin necesidad de reescribir la lógica. Las pruebas verificarán que los componentes son reutilizables sin modificaciones.

g. 3.3.7 Fiabilidad

- **Descripción:** La arquitectura MVC asegura que los errores en la Vista (interfaz de usuario) no afecten la lógica de negocio ni la integridad de los datos.
- **Criterio de Aceptación:** La consistencia de los datos será verificada mediante pruebas de integridad que confirmen que los cambios en la interfaz no afectan los datos gestionados por el Modelo.

h. 3.3.8 Usabilidad

- **Descripción:** Gracias a la separación en capas del modelo MVC, se podrá desarrollar una interfaz de usuario amigable y eficiente sin necesidad de comprometer la lógica de negocio.
- **Criterio de Aceptación:** El sistema debe ser intuitivo y fácil de usar, permitiendo que los usuarios realicen sus tareas sin asistencia. Se realizarán pruebas de usabilidad para confirmar que los usuarios pueden completar las tareas en menos de 5 minutos.

4. Especificación de los Escenarios de Calidad Relevantes

Define los requisitos no funcionales, enfocándose en atributos como rendimiento, disponibilidad y seguridad en situaciones reales.

Después de un análisis en conjunto con los stakeholders, los escenarios de calidad se expresan a continuación:

Escenario de Calidad N°1	
Atributo de Calidad Asociado (Característica): Rendimiento (R6, R28) Subcaracterística: Tiempo de Respuesta	
Descripción: El sistema debe procesar las operaciones críticas (generación de boletas y pedidos) en un tiempo mínimo, asegurando que el sistema no se ralentice bajo carga.	
Fuente del Estímulo:	Usuario
Estímulo:	Múltiples usuarios generando boletas y pedidos simultáneamente.
Artefacto:	Módulo de admin. Boletas /facturas y modulo de admin. De orden de Pedidos
Ambiente:	Operación normal bajo carga
Respuesta:	El sistema debe procesar las solicitudes en menos de 2 segundos.
Medida de Respuesta:	Tiempo de respuesta \leq 2 segundos en el 95% de los casos.

Escenario de Calidad N°2	
Atributo de Calidad Asociado (Característica): Seguridad (R8, R32) Subcaracterística: Control de Acceso	
Descripción: El sistema debe prevenir accesos no autorizados, protegiendo la información sensible como datos de clientes y transacciones de boletas.	
Fuente del Estímulo:	Actor externo (Hacker)
Estímulo:	Intento de acceso no autorizado a la administración del sistema.
Artefacto:	Módulo de admin. boleta o factura y ordenes de admin. De orden de pedido
Ambiente:	Operación normal bajo carga
Respuesta:	El sistema debe procesar las solicitudes en menos de 2 segundos.
Medida de Respuesta:	Tiempo de respuesta \leq 2 segundos en el 95% de los casos.

Escenario de Calidad N°3	
Atributo de Calidad Asociado (Característica): Disponibilidad (R7)	
Subcaracterística: Restricciones Horarias	
Descripción: El sistema debe restringir el acceso a los trabajadores fuera del horario laboral definido (Lun-Sáb, 8:00 a 20:00).	
Fuente del Estímulo:	Usuario (Trabajador)
Estímulo:	Intento de acceder fuera del horario laboral.
Artefacto:	Módulo admin. de boleta o factura y módulo de admin. de orden de pedido
Ambiente:	Fuera del horario laboral
Respuesta:	El acceso debe ser bloqueado para los trabajadores y administradores fuera del horario.
Medida de Respuesta:	El 100% de los accesos fuera del horario laboral deben ser bloqueados.

Escenario de Calidad N°4	
Atributo de Calidad Asociado (Característica): Mantenibilidad (R15) Subcaracterística: Modificabilidad	
Descripción: El sistema debe permitir la actualización de los módulos de boletas y pedidos sin afectar el resto del sistema.	
Fuente del Estímulo:	Equipo de desarrollo.
Estímulo:	Actualización del módulo de boletas para agregar nuevas funcionalidades.
Artefacto:	Módulo admin. de boleta
Ambiente:	Actualización en entorno de producción
Respuesta:	El resto del sistema (clientes y pedidos) debe seguir operando sin interrupciones durante la actualización.
Medida de Respuesta:	El 100% de las pruebas de regresión deben pasar sin fallos en los módulos no modificados.

Escenario de Calidad N°5	
Atributo de Calidad Asociado (Característica): Usabilidad (R17, R29)	
Subcaracterística: Aprendizaje	
Descripción: Los usuarios deben poder utilizar el sistema de forma intuitiva sin necesidad de asistencia externa.	
Fuente del Estímulo:	Usuario (Trabajador
Estímulo:	Un nuevo trabajador accede al sistema para generar boletas y consultar el estado de un pedido por primera vez.
Artefacto:	Interfaz de usuario del sistema (Vista)
Ambiente:	Uso en condiciones normales
Respuesta:	El trabajador debe poder completar las tareas sin necesidad de capacitación adicional.
Medida de Respuesta:	El 90% de los nuevos usuarios deben completar las tareas en menos de 5 minutos.

Escenario de Calidad N°6	
Atributo de Calidad Asociado (Característica): Escalabilidad (R14, R16)	
Subcaracterística: Capacidad de expansión	
Descripción: El sistema debe permitir la incorporación de nuevas sedes del taller a medida que la empresa crece, sin necesidad de modificar el código base del sistema.	
Fuente del Estímulo:	Nuevas sedes del taller
Estímulo:	Incorporación de una nueva sede al sistema, con la gestión de clientes, boletas y pedidos para la nueva ubicación.
Artefacto:	Módulos de admin. clientes, boletas y pedidos
Ambiente:	Operación normal, incorporando la nueva sede sin interrupciones para las sedes existentes
Respuesta:	El sistema debe permitir la integración de la nueva sede sin requerir cambios en el código base y sin afectar las sedes ya operativas.
Medida de Respuesta:	El sistema debe incorporar la nueva sede sin errores funcionales en el 100% de las pruebas, y debe permitir la gestión independiente de las sedes existentes y nuevas.

Escenario de Calidad N°7	
Atributo de Calidad Asociado (Característica): Fiabilidad (R30) Subcaracterística: Integridad de Datos	
Descripción: El sistema debe registrar todas las modificaciones realizadas en boletas, facturas y órdenes de pedido, incluyendo la fecha, hora y el usuario que realizó la modificación, asegurando que esta información esté disponible en cualquier momento.	
Fuente del Estímulo:	Administrador del sistema
Estímulo:	Un usuario modifica una boleta o pedido, y el administrador consulta el registro de auditoría para revisar los cambios realizados.
Artefacto:	Módulos de admin. boletas y pedidos
Ambiente:	Operación normal, con múltiples usuarios realizando modificaciones en boletas y pedidos
Respuesta:	El sistema debe registrar todas las modificaciones con precisión y mantener estos registros disponibles para consulta en cualquier momento.
Medida de Respuesta:	El 100% de las modificaciones deben registrarse correctamente con la fecha, hora y el usuario correspondiente, y estar disponibles en el módulo de auditoría para consulta en el 100% de las solicitudes.

Escenario de Calidad N°8	
Atributo de Calidad Asociado (Característica): Modularidad (R15)	
Subcaracterística: Independencia de los Módulos	
Descripción: El sistema debe permitir la actualización de un módulo (por ejemplo, el módulo de boletas) sin que se vean afectados los otros módulos (Clientes, Pedidos).	
Fuente del Estímulo:	Equipo de desarrollo
Estímulo:	Se realiza una actualización en el módulo de boletas para añadir nuevas funcionalidades sin afectar los módulos de clientes y pedidos.
Artefacto:	Módulo admin. de Boletas
Ambiente:	Entorno de desarrollo y producción
Respuesta:	El sistema debe seguir operando normalmente en los módulos de clientes y pedidos durante la actualización del módulo de boletas.
Medida de Respuesta:	El 100% de las pruebas de regresión deben pasar sin errores en los módulos de clientes y pedidos durante y después de la actualización del módulo de boletas.

5. Vista Lógica

A continuación, se presenta una vista lógica de la aplicación. Muestra las entidades principales del sistema y sus relaciones. Describe la estructura del sistema mediante clases como Cliente, Vehículo, Orden de Trabajo, entre otras.

Define los datos y funciones principales de cada entidad, asegurando una clara separación entre las responsabilidades de cada módulo del sistema.

Diagrama de Clase

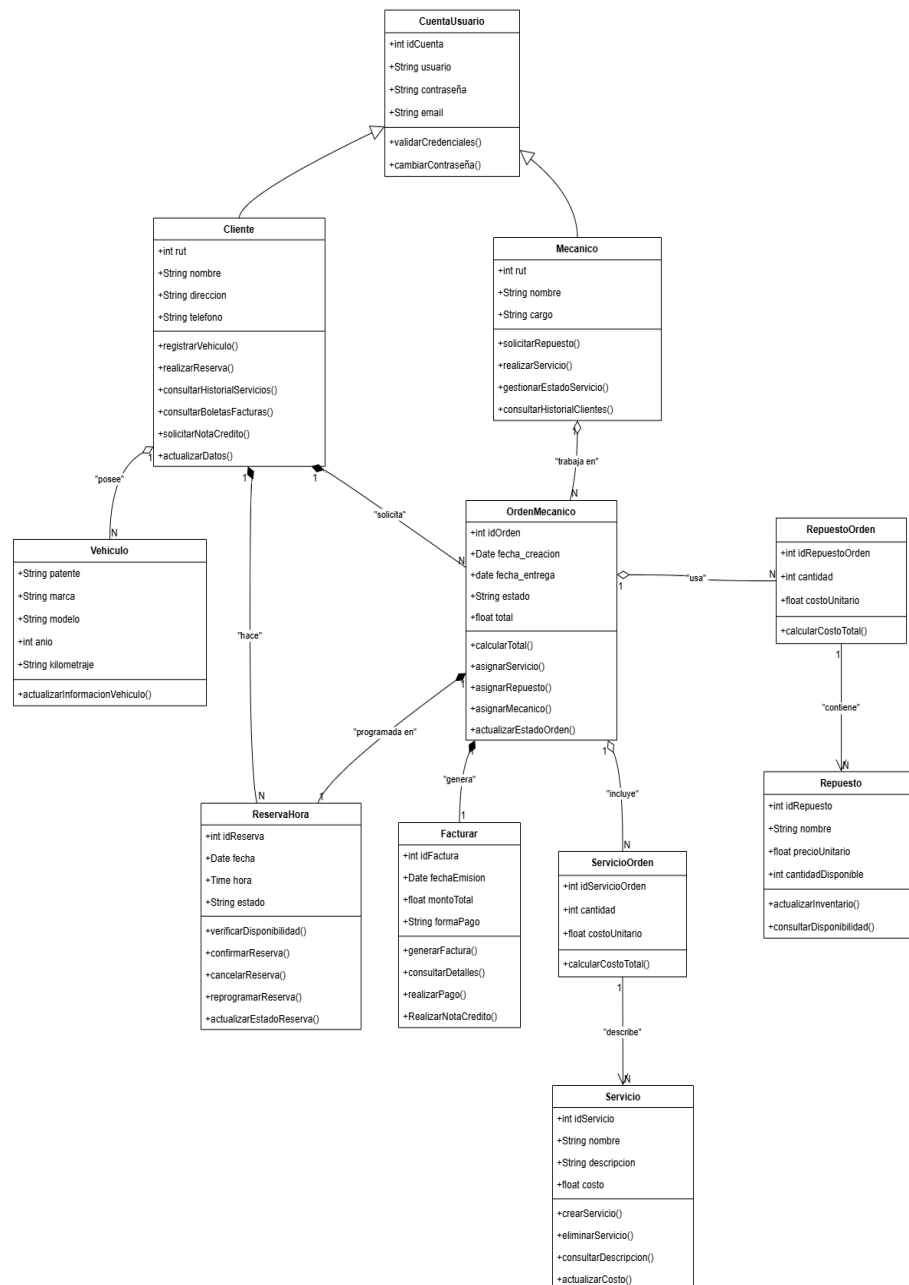


Diagrama Entidad-Relación

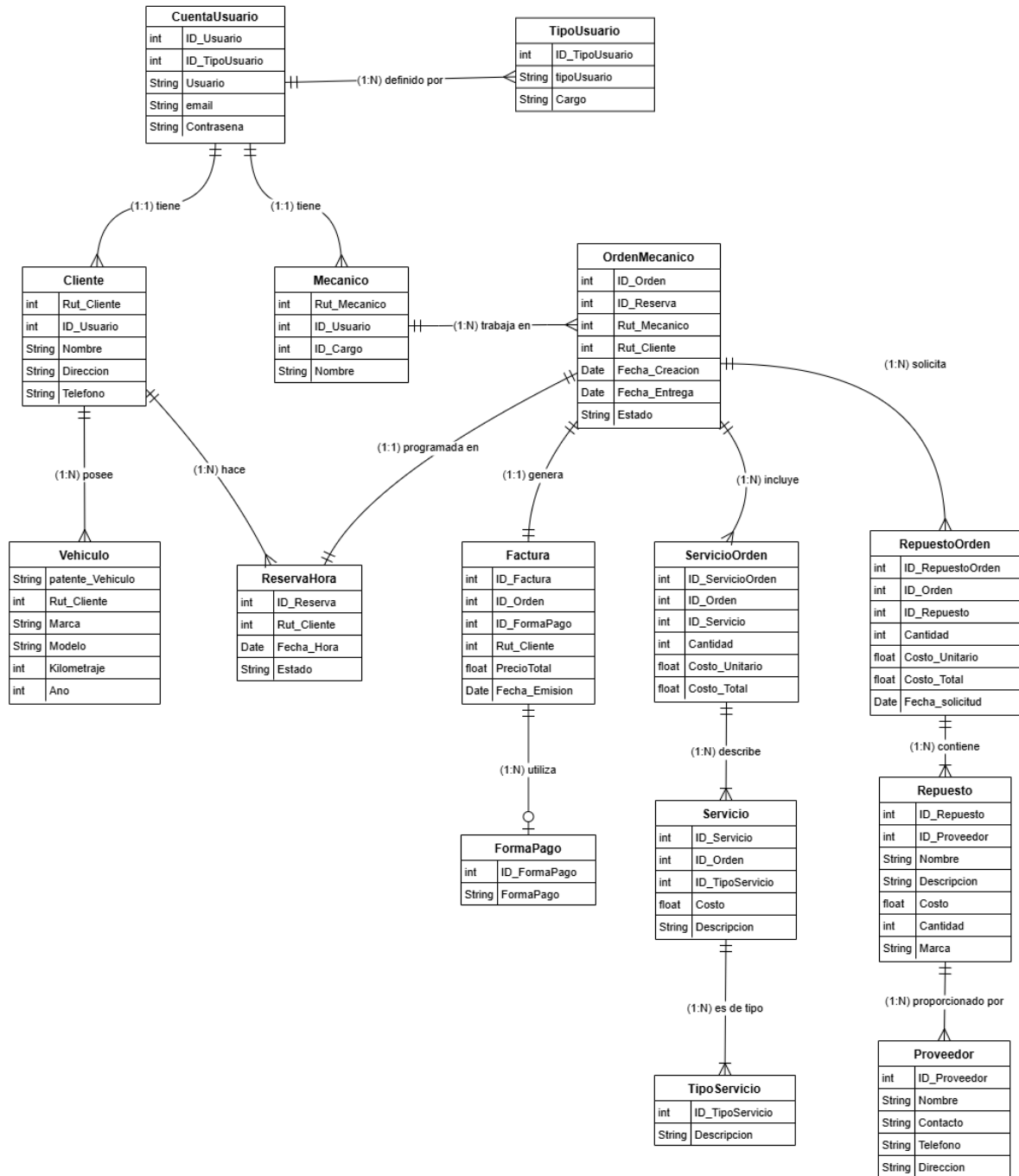


Diagrama de Secuencia

Realizar Pago de Servicio Mecánico

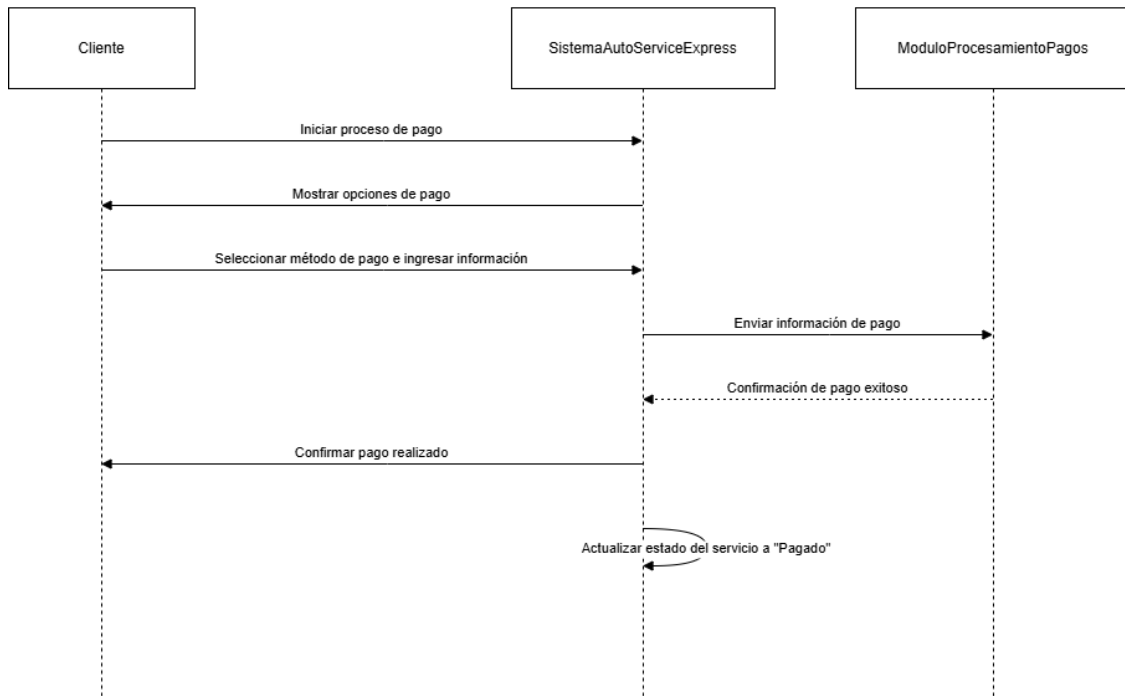


Diagrama de Secuencia

Gestionar Orden de Servicio Mecánico

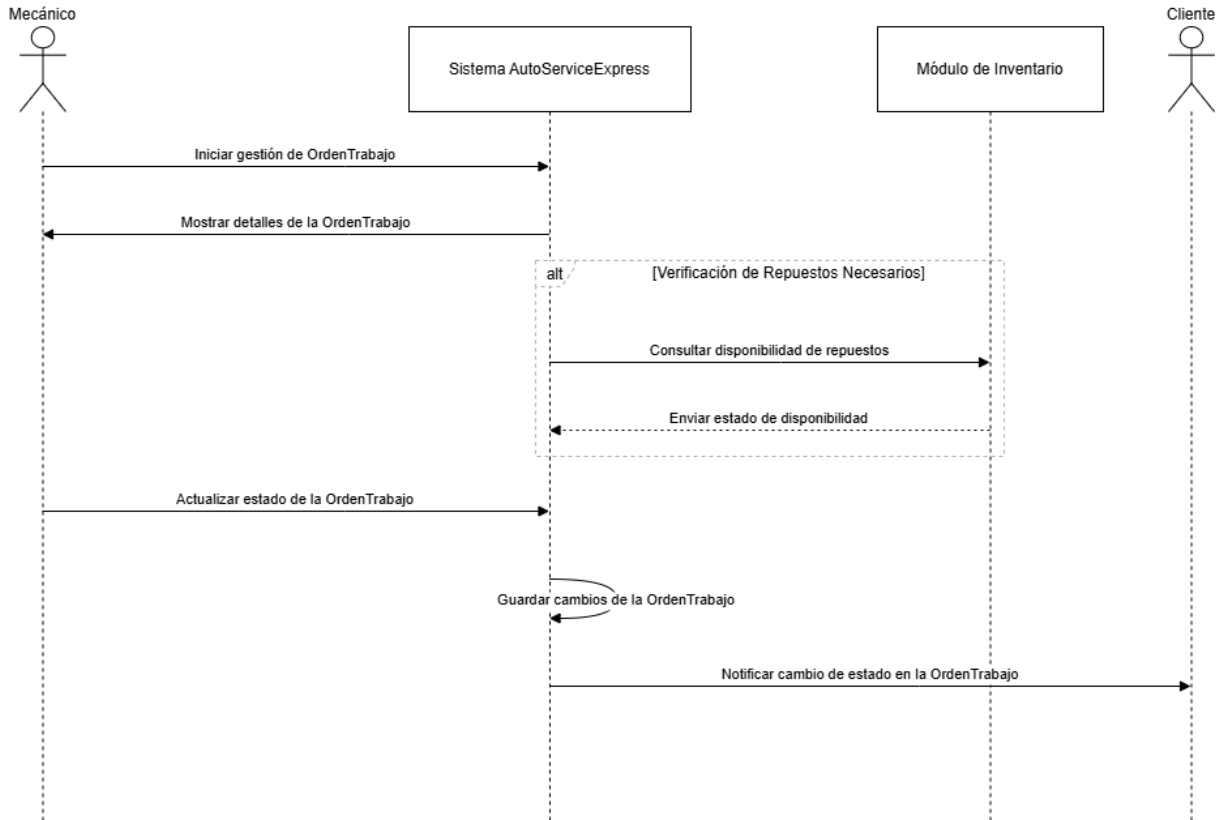


Diagrama de Secuencia

Reservar Hora de Servicio Mecánico

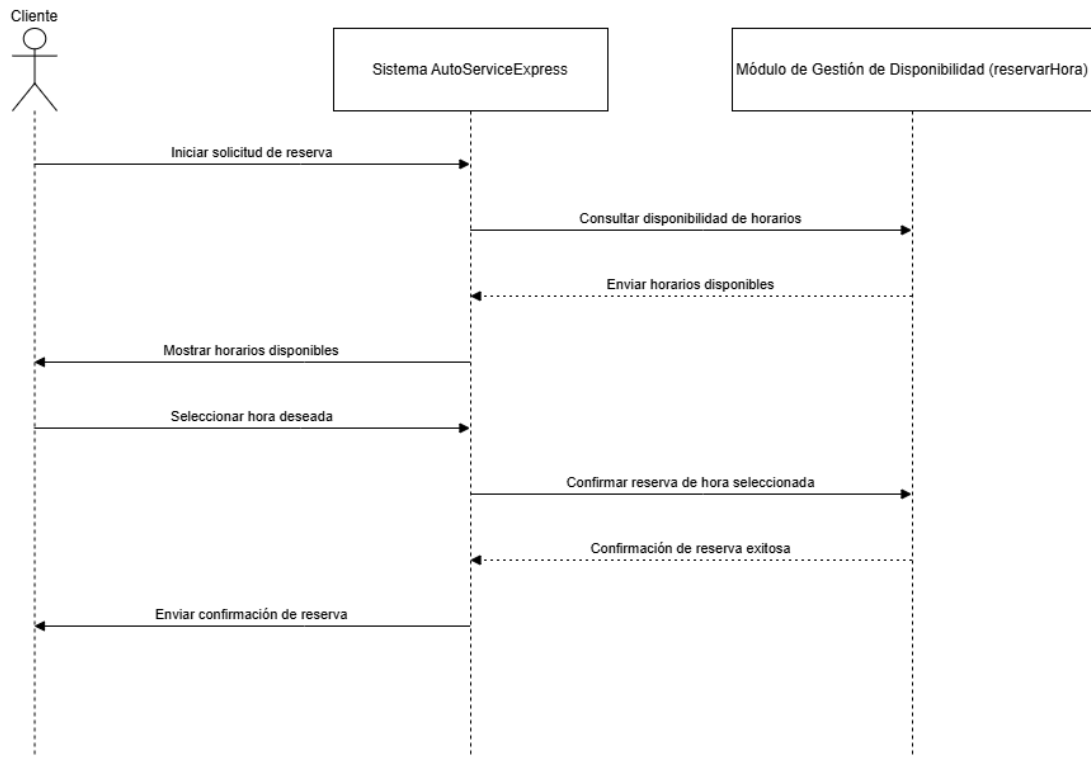
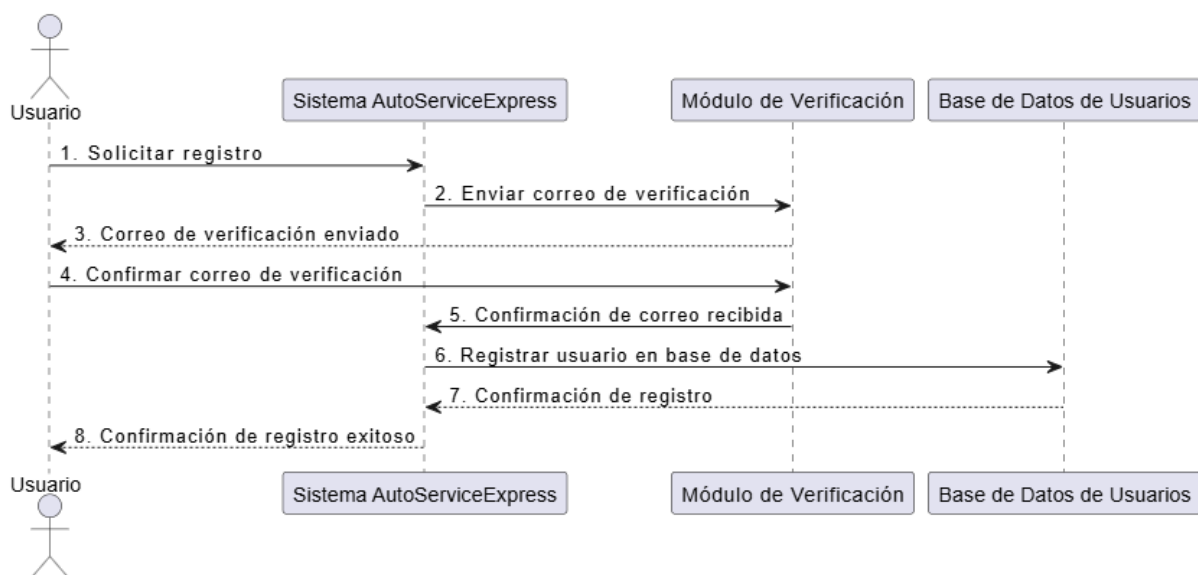


Diagrama de Comunicación

Registro al Sistema



6 Vista de Procesos

A continuación, se muestra una vista de procesos, en la cual se observa que modela los flujos de trabajo principales, como el proceso de Reservar Servicio Mecánico, detallando cada paso en la secuencia de actividades.

Diagrama de Actividad
Generar Boleta/Factura

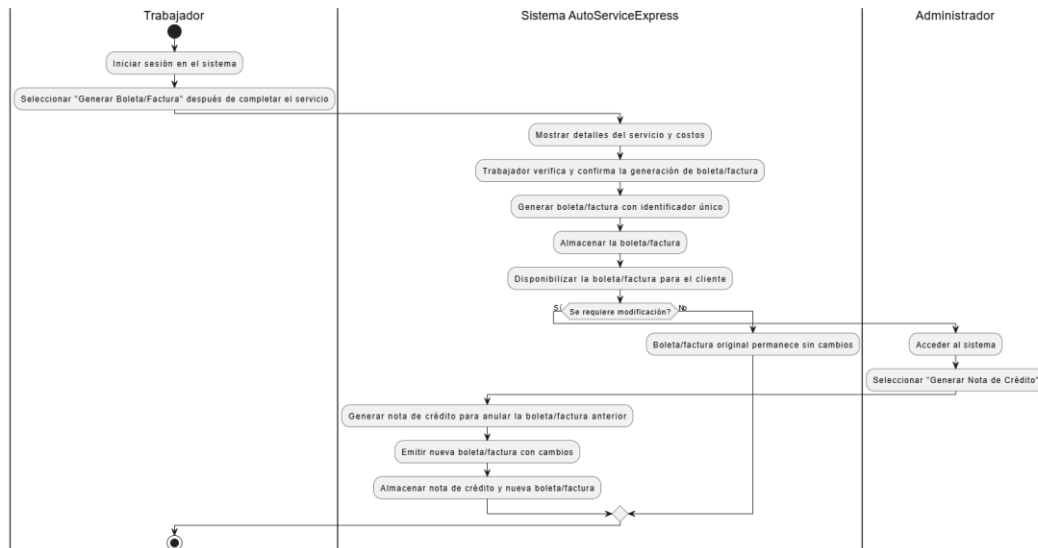


Diagrama de Actividad
Reserva de Servicio Mecánico

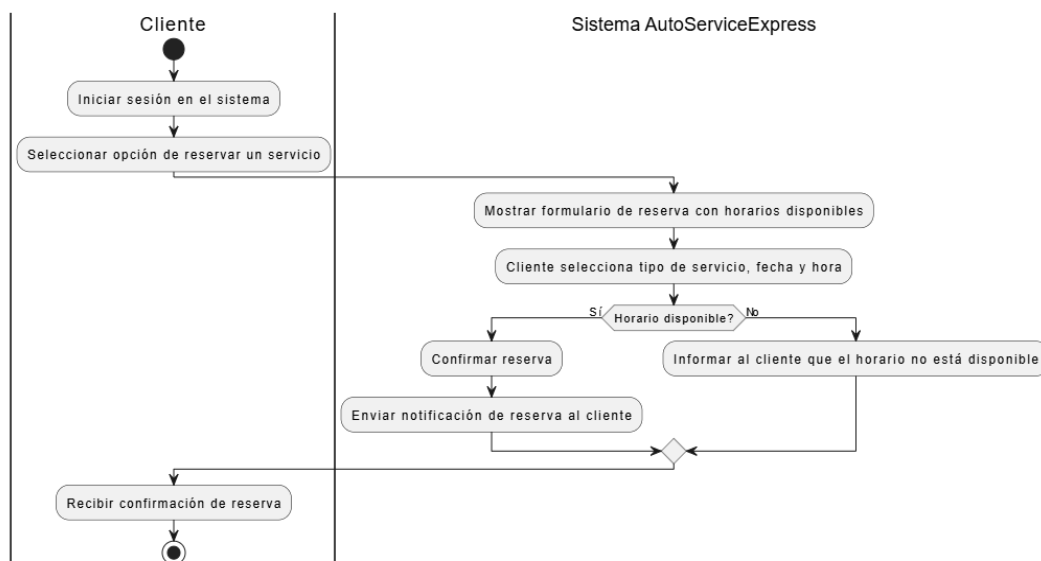
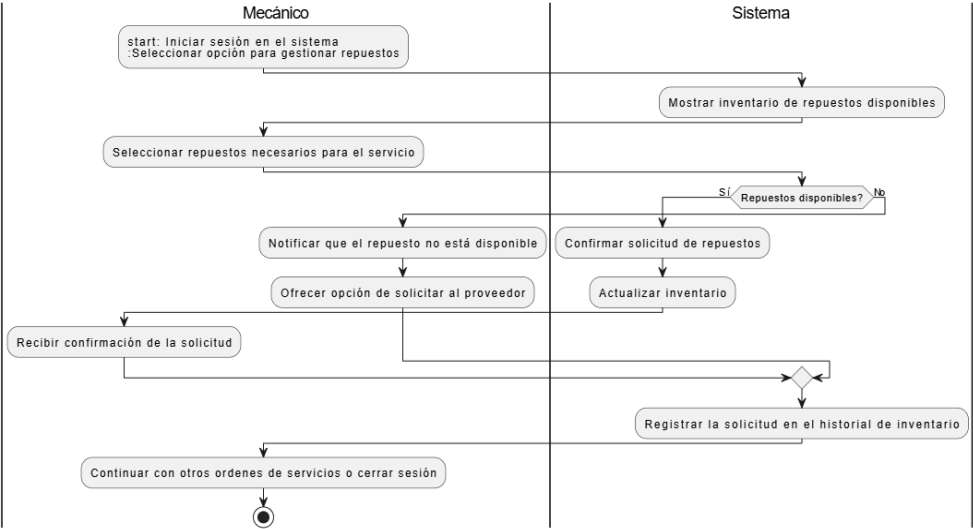


Diagrama de Actividad

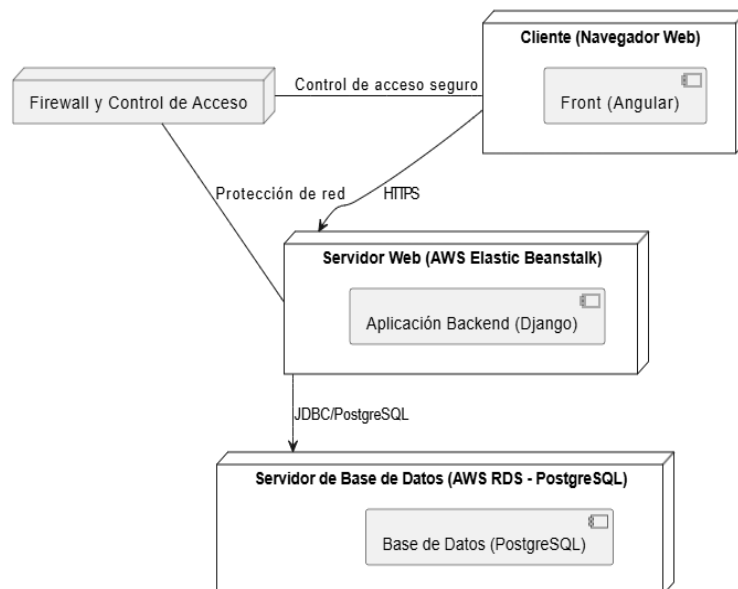
Gestión de Repuestos



7 Vista Física

Representa la infraestructura física, con servidores, bases de datos y redes involucradas. Este diagrama describe el despliegue en un entorno de servidor y el almacenamiento en PostgreSQL.

Diagrama de Despliegue Sistema AutoServiceExpress



8 Vista de Despliegue

En esta vista se aprecia que existirán dos módulos principales que contendrán distintas funcionalidades de la aplicación. A continuación se describen:

Diagrama de Componentes Sistema AutoServiceExpress

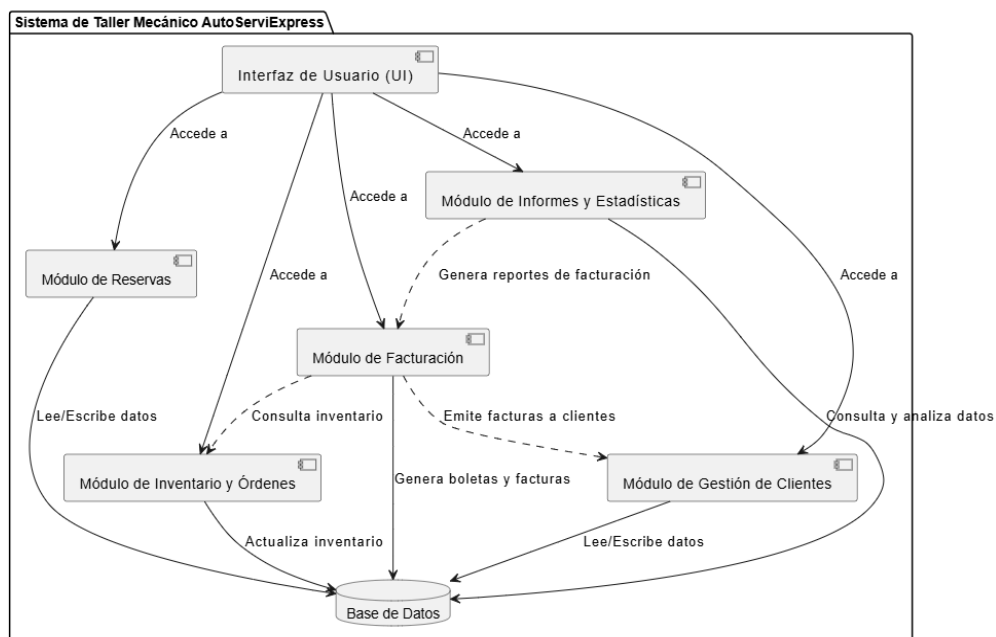
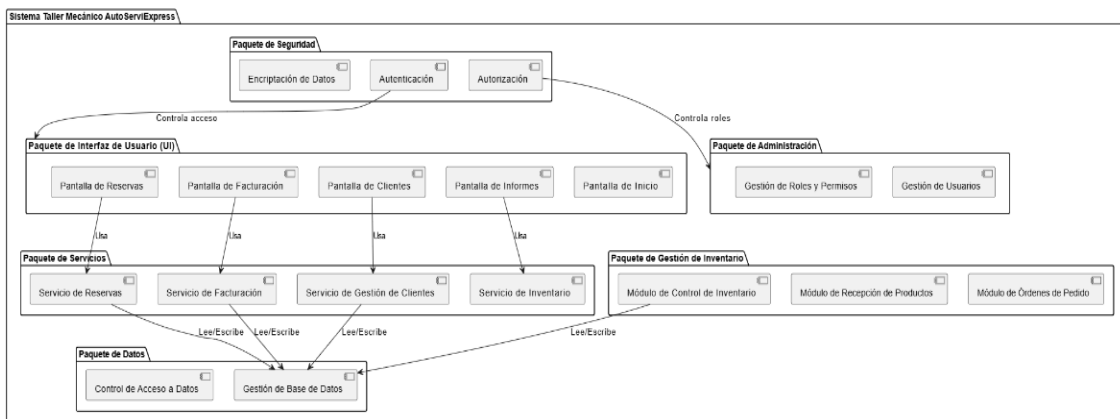


Diagrama de Paquetes Sistema AutoServiceExpress



9.

Decisiones de Diseño y Selección de Alternativas

Las principales decisiones arquitectónicas se tomaron en consideración de la restricción **Tiempo de Construcción**. Dado que el proyecto debe implementarse en un tiempo ajustado y sin holguras, se privilegió la adopción de una arquitectura conocida y que presente un bajo riesgo en su implementación.

Asimismo, la arquitectura se modularizó con el primer objetivo de separar concernimientos de forma que permita paralelización en construcción de dichos componentes, y que a su vez sea módulos testeables unitariamente de forma de asegurar (mediante la suite Junit) que cada pieza tenga una baja tasa de fallas.

Un segundo elemento fue considerado en la arquitectura, que corresponde a la restricción de **Infraestructura** con que debe cumplir la aplicación, combinado con el escenario de calidad de **Tolerancia a Fallos**, nos condiciona la modularización de la aplicación en una **aplicación web activa-activa** y una **aplicación de servicios activa-pasiva**.

El escenario de calidad relacionado con la **mantenibilidad** nos conduce al modelamiento pensando en la separación de concernimiento de los componentes y a la utilización del patrón **provider** de forma que el sistema pueda delegar sus requerimientos de información hacia sistemas externos a piezas de software no acopladas que nos permitan su extensibilidad a futuro.

Por último, se eligió que la estrategia para implementar los providers externos en aquellos servicios asíncronos de entrada se implementarían mediante un temporizador (quartz) que levantará los procesos que verifican la llegada de información a las colas de entrada (mensajería asíncrona, archivos de texto en directorios). Esta estrategia fue seleccionada para disminuir el riesgo pues es una solución simple y efectiva.

- **Decisiones de Diseño:**

- Se ha elegido la arquitectura **Modelo-Vista-Controlador (MVC)** para separar la lógica de presentación de la lógica de negocio. Esto facilita la modularidad del sistema, permitiendo actualizaciones y mantenimiento sin afectar otras partes del sistema. Esta arquitectura permite también que el sistema crezca en funcionalidades, manteniendo la claridad y organización del código.

- **Selección de Alternativas:**

- **Angular:** Seleccionado como framework de frontend por su capacidad para crear interfaces de usuario rápidas y responsivas, además de contar con una amplia comunidad de soporte y herramientas que aceleran el desarrollo.
- **Django:** Elegido como framework de backend debido a su estructura escalable y su capacidad para manejar aplicaciones con requisitos de seguridad elevados. Django ofrece herramientas integradas que facilitan la creación de una API robusta.
- **PostgreSQL:** Elegido como sistema de gestión de bases de datos relacional por su confiabilidad, facilidad de uso y soporte para aplicaciones de mediana a gran escala, lo cual se alinea con los requisitos de **AutoServiceExpress**.
-

6. 10. Matriz de Riesgos por Atributos de Calidad

Estas matrices reflejan los riesgos específicos de cada atributo de calidad, calculados en base a la probabilidad e impacto para obtener el nivel de riesgo asociado. Las clasificaciones generales sugieren qué atributos son más críticos y pueden requerir un enfoque de mitigación en fases tempranas del desarrollo del sistema.

7. Escala de Riesgo

Nivel de Riesgo Calculado	Clasificación
1 -3	Bajo
4-6	Medio
7-9	Alto

1. Matriz de Riesgo para Mantenibilidad

Riesgo	Probabilidad (P)	Impacto (I)	Nivel de Riesgo (P * I)	Clasificación
Cambio en la lógica afecta otras capas	Medio (2)	Alto (3)	6	Medio
Tiempo de inactividad prolongado en actualizaciones	Bajo (1)	Medio (2)	2	Bajo

a. Clasificación general del atributo: Medio

2. Matriz de Riesgo para Escalabilidad

Riesgo	Probabilidad (P)	Impacto (I)	Nivel de Riesgo (P * I)	Clasificación
Sobrecarga del sistema con aumento de datos	Alto (3)	Alto (3)	9	Alto
Dificultad para integrar nuevas funcionalidades	Medio (2)	Medio (2)	4	Medio

b. Clasificación general del atributo: Alto

3. Matriz de Riesgo para Modularidad

Riesgo	Probabilidad (P)	Impacto (I)	Nivel de Riesgo (P * I)	Clasificación
Interdependencia de módulos	Bajo (1)	Alto (3)	3	Bajo
Fallas en un módulo afectan a otros	Medio (2)	Medio (2)	4	Medio

c. Clasificación general del atributo: Medio

4. Matriz de Riesgo para Rendimiento

Riesgo	Probabilidad (P)	Impacto (I)	Nivel de Riesgo (P * I)	Clasificación
Aumento de tiempo de respuesta en picos	Alto (3)	Alto (3)	9	Alto
Rendimiento reducido por cambios en el sistema	Medio (2)	Medio (2)	4	Medio

d. Clasificación general del atributo: Alto

5. Matriz de Riesgo para Seguridad

8.

Riesgo	Probabilidad (P)	Impacto (I)	Nivel de Riesgo (P * I)	Clasificación
Acceso no autorizado a datos sensibles	Alto (3)	Alto (3)	9	Alto
Fallos en encriptación	Medio (2)	Alto (3)	6	Medio

a. Clasificación general del atributo: Alto

6. Matriz de Riesgo para Fiabilidad

Riesgo	Probabilidad (P)	Impacto (I)	Nivel de Riesgo (P * I)	Clasificación
Inconsistencias por errores de usuario	Medio (2)	Medio (2)	4	Medio
Pérdida de datos en caso de falla	Bajo (1)	Alto (3)	3	Bajo

b. Clasificación general del atributo: Medio

7. Matriz de Riesgo para Usabilidad

9.

Riesgo	Probabilidad (P)	Impacto (I)	Nivel de Riesgo (P * I)	Clasificación
Dificultad de uso para nuevos usuarios	Medio (2)	Medio (2)	4	Medio
Interfaz confusa afecta productividad	Bajo (1)	Alto (3)	3	Bajo

a. Clasificación general del atributo: Medio

8. Matriz de Riesgo para Compatibilidad

Riesgo	Probabilidad (P)	Impacto (I)	Nivel de Riesgo (P * I)	Clasificación
Problemas al importar/exportar datos	Medio (2)	Medio (2)	4	Medio
Incompatibilidad con otros sistemas externos	Medio (2)	Alto (3)	6	Medio

b. Clasificación general del atributo: Medio

Resumen de Clasificación de Riesgo por Atributo de Calidad

Atributo de calidad	Clasificación de Riesgo
Mantenibilidad	Medio
Escalabilidad	Alto
Modularidad	Medio
Rendimiento	Alto
Seguridad	Alto
Fiabilidad	Medio
Usabilidad	Medio
Compatibilidad	Medio

-Felipe Carrillo Aburto. 7.0

-Jorge Cavieres Rojas. 7.0

-Cristian González Valenzuela. 7.0