

Documento de Arquitectura de Software

IEEE-1471-2000

Control del documento

Proyecto

Sistema Restaurant

Título

Arquitectura del Sistema – [v1.0 al 02 de Julio de 2009]

Generado por

Magister en Informática - [Juan José González Faúndez]

Aprobado por

[Acá tiene que poner la firma el profe Veloso]

1. Introducción

1.1 Propósito

Este documento proporciona una descripción comprensiva arquitectónica del sistema, usando un número finito de vistas diferentes para representar los distintos aspectos que se requieren para capturar y transportar las decisiones significativas que han sido hechas sobre el sistema.

1.2 Alcance

El presente documento contiene el diseño elaborado para el proyecto **Sistema Restaurant**, el cual es producto de un análisis minucioso de los requisitos del sistema, según estos pueden ser satisfechos con las tecnologías y características discutidas con los clientes y usuarios.

El documento está organizado alrededor de tres ideas principales.

1. Las características generales del diseño
2. Los requisitos atendidos por el diseño
3. Los modelos y vistas que lo detallan

Al contrario de muchas otras actividades técnicas, el desarrollo de sistemas intensivos en software dedica la mayoría de sus esfuerzos a la especificación y modelado.

Los modelos son utilizados tanto para el análisis de requisitos, como para el diseño de la solución, así como para la especificación, construcción y despliegue del sistema en su ambiente de explotación.

Los modelos son presentados por vistas o diagramas, generalmente utilizando notaciones gráficas como el UML.

Por otro lado, los programas de computadora son construidos por medio del uso de herramientas de traducción automáticas llamados compiladores, para los cuales es construida la forma lineal y más detallada del software del sistema: el código fuente.

La última sección del documento indica la forma en que se puede obtener el código fuente del proyecto así como las instrucciones de compilación necesarias para lograr la ejecución de los componentes que este código detalla.

Este documento ha sido generado directamente del análisis del sistema RESTAURANT y el modelo de diseño puesto e implementado en Rational Rose Versión 7.0. La mayoría de las secciones ha sido extraída del Modelo de Rational Rose Version 7.0 y la utilización de plantillas de referencia de ATAM (*Architecture Tradeoff Analysis Method*) y del modelo 4+1 de Kruchten.

1.3 Usuarios Interesados

Este documento de Arquitectura de Software (DAS), puede ser usado por todos aquellos usuarios que deseen comprender el diseño y construcción de la aplicación de Restaurant, y sirve como base para que los desarrolladores de software puedan construir el bajo nivel de la aplicación usando el lenguaje que más les acomode.

1.4 Recomendaciones de conformidad con esta práctica.

N/A.

2. Referencias

Las referencias aplicables a este documento son:

- *IEEE 830-1998 ST*
- *Architecture Tradeoff Analysis Method*
- *ISO 9126 -2001 Calidad del Software y Métricas de evaluación*
- *The 4+1 View .Kruchten - 1009*

3. Definiciones, acrónimos y abreviaciones.

DAS: Documento de Arquitectura de Software

RESTAURANT: sistema de gestión de reserva de mesas para restaurantes.

HTTP: Protocolo de Trasferencia de Hipertexto.

TCP: Protocolo de control de transmisión.

ARQUITECTURA DE SOFTWARE: conjunto de elementos estáticos, propios del diseño intelectual del sistema, que definen y dan forma tanto al código fuente, como al comportamiento del software en tiempo de ejecución. Naturalmente este diseño arquitectónico ha de ajustarse a las necesidades y requisitos del proyecto.

DESCRIPCION DE ARQUITECTURA: colección de productos de documentación.

VISTAS: es una representación de un área de interés o perspectiva del sistema en alto nivel.

TIPOS DE VISTAS: especificación de una convención de cómo construir y usar una vista. Deben satisfacer la capacidad de creación y análisis de una vista.

STAKEHOLDER: Individuo, equipo u organización con intereses relativos al sistema.

ESCENARIO: especifica el comportamiento y limita el interés de un área específica del sistema para uno o varios stakeholders.

MODULO O COMPONENTE: cualquier elemento estructural abstracto, visible, externo, de alto nivel, analizable, que pueda constituir una funcionalidad de la solución del sistema.

ATRIBUTOS DE CALIDAD: un atributo de calidad, es una cualidad deseable de la solución, que pueda manifestarse en forma de requerimiento no funcional, que pueda ser medible, testeable y finalmente evaluable.

4. Framework Conceptual

4.1 Descripción de la arquitectura en contexto

Este documento presenta la arquitectura como una serie de vistas basadas en la arquitectura de software del modelo 4+1 de Kruchten. Estas vistas son: la vista de escenarios, la vista lógica., la vista de desarrollos, la vista física, la vista de procesos .No hay ninguna vista separada de una misma implementación, descrita en este documento. Estas vistas están hechas sobre Lenguaje de modelo unificado (UML) en su versión 2.0 desarrolladas usando IBM Rational Rose Enterprise 7.0.

Los estilos arquitectónicos serán referenciados en este documento de arquitectura, según las recomendaciones de la Arquitectura de software del modelo 4+1 de Kruchten.

4.2 Stakeholders y sus roles

Este documento representa la identificación de Stakeholders y sus roles a partir de la interpretación de los casos de uso del Negocio.

4.3 Actividades de arquitectura en el ciclo de vida

N/A.

4.4 Usos de las descripciones de arquitectura

Las descripciones de arquitectura de este documento se usaran para referenciar el diseño del sistema de software de RESTAURANT.

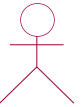
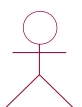

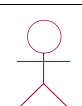

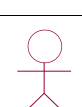
5. Descripciones prácticas de arquitectura

N/A.

5.1 Documentación de la arquitectura

N/A.

5.2 Identificación de los Stakeholders y sus responsabilidades

Stakeholder	descripción	escenario	Vistas
 Administrador	Es el usuario dueño del restaurant y está encargado de la gestión directiva del restaurant.	<ul style="list-style-type: none"> • Escenario de negocios • Escenario de diseño 	<ul style="list-style-type: none"> • CU Negocio • CU Diseño Gestionar Reserva • CU Diseño Gestionar Compra (proveedores) • CU Diseño Gestionar Cuentas
 Cliente	Es la persona que interactúa con el negocio de restaurant y hace los pedidos de menú según su preferencia.	<ul style="list-style-type: none"> • Escenario de negocios • Escenario de diseño 	<ul style="list-style-type: none"> • CU Negocio • CU Diseño Gestionar reserva • Gestionar Cuenta (Caja)
 Cajero	Es la persona encargada de hacer efectivo el pago y recibir el dinero que le proporciona el cliente.	<ul style="list-style-type: none"> • Escenario de negocios • Escenario de diseño 	<ul style="list-style-type: none"> • CU Negocio • CU Diseño Gestionar Cuenta (Caja)
 Garzon	Es la persona encargada de atender a los clientes y llenar la comanda con los pedidos.	<ul style="list-style-type: none"> • Escenario de negocios • Escenario de diseño 	<ul style="list-style-type: none"> • CU Negocio • CU Diseño Gestionar Ordenes
 Jefe cocina	Es la persona encargada de administrar las órdenes que llegan a cocina y establecer las prioridades de cada una.	<ul style="list-style-type: none"> • Escenario de negocios • Escenario de diseño 	<ul style="list-style-type: none"> • CU Diseño Gestionar Cocina • CU Diseño Gestionar Proveedores
 Manager	Es la persona encargada de administrar el local de restaurant.	<ul style="list-style-type: none"> • Escenario de negocios • Escenario de diseño 	<ul style="list-style-type: none"> • CU Negocio • CU Diseño Gestionar Cocina • CU Diseño Gestionar Cuenta • CU Diseño Gestionar Ordenes

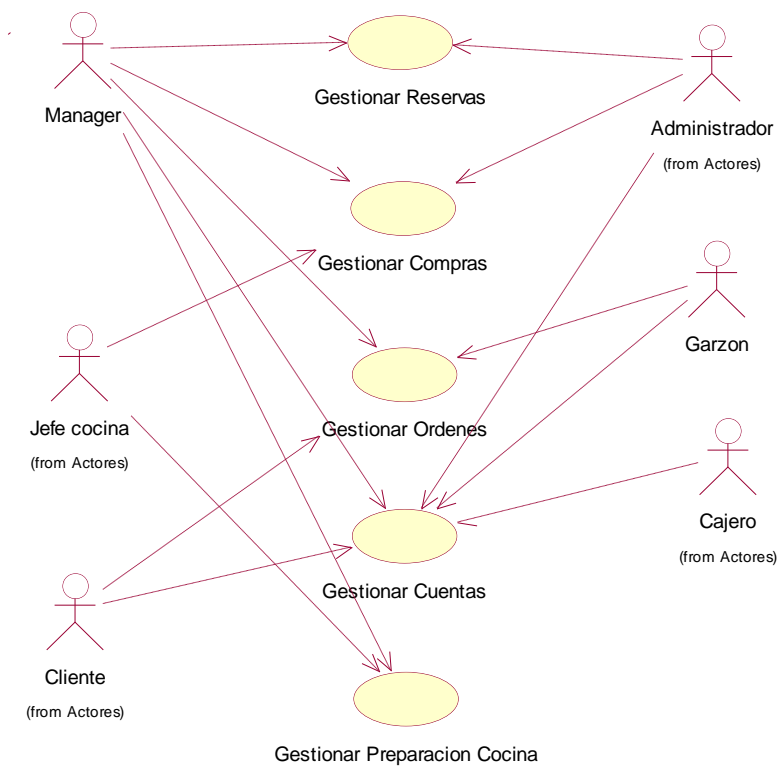
5.3 Selección de puntos de vista de la arquitectura.

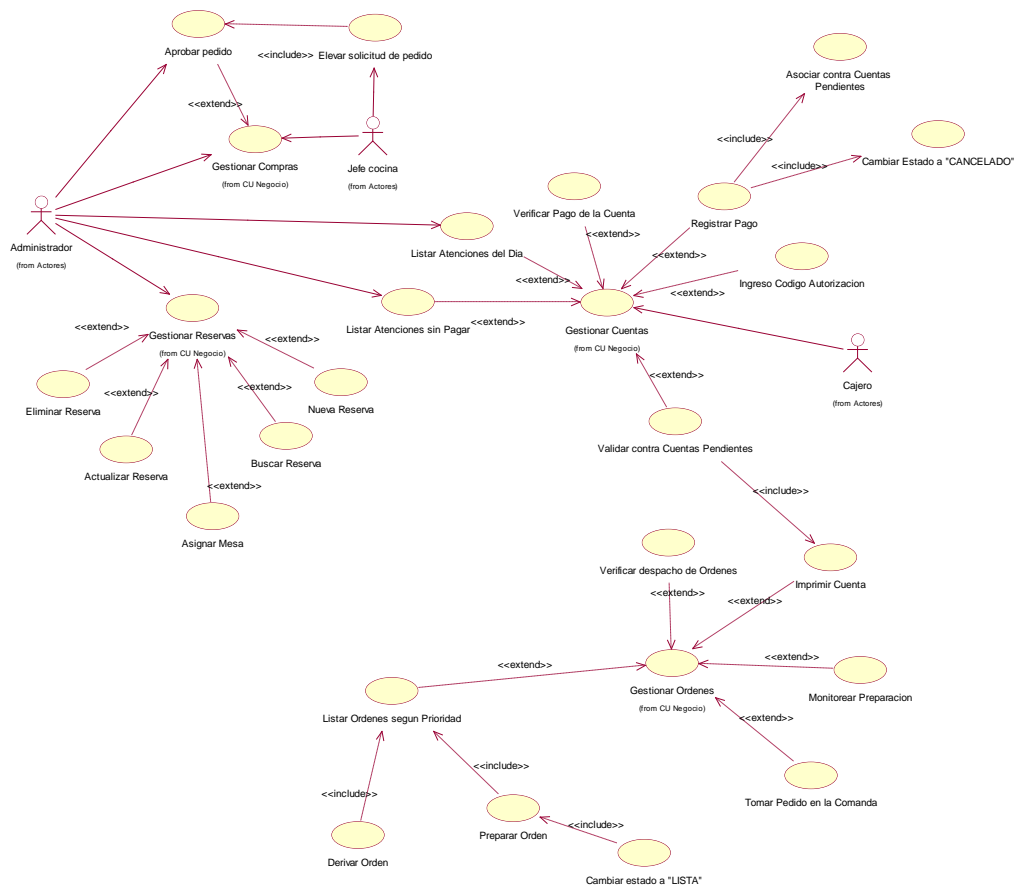
Vistas	UML
Escenarios	Casos de uso
Lógica	Clases
Desarrollo	Componentes
Física	Despliegue
Procesos	Secuencia

5.4 Vistas de la arquitectura

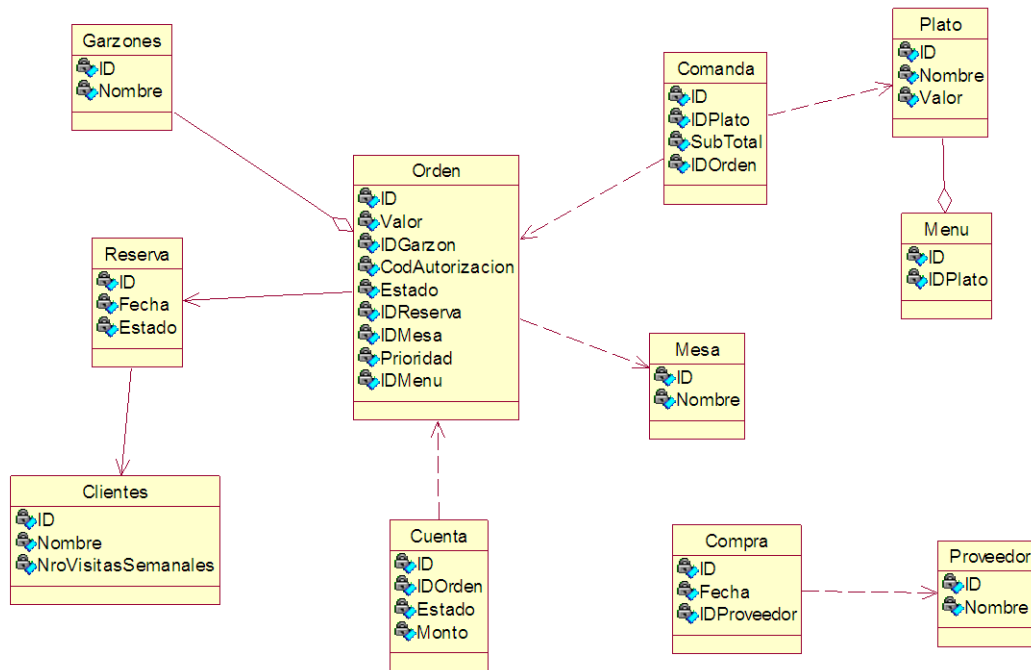
Vistas.- escenarios

Diagrama.- Caso de uso del negocio – Caso de uso de diseño

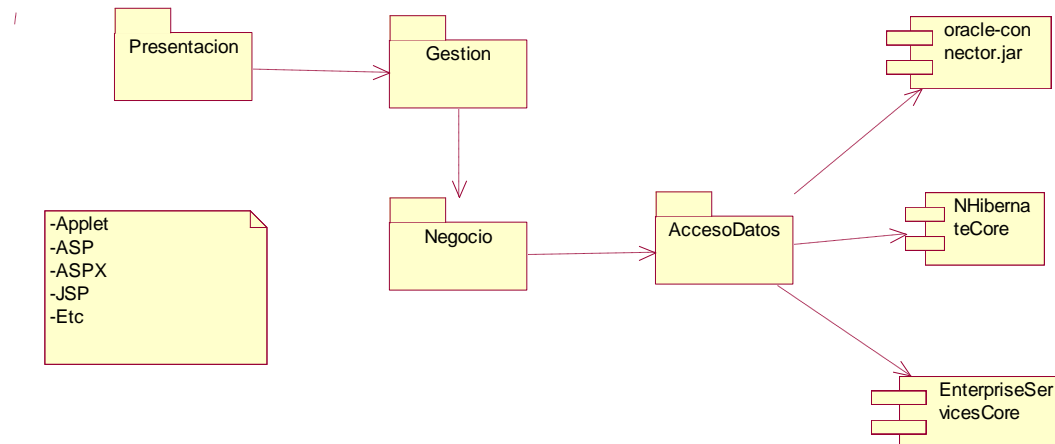




Vista.- Lógica Diagramas.- Clases

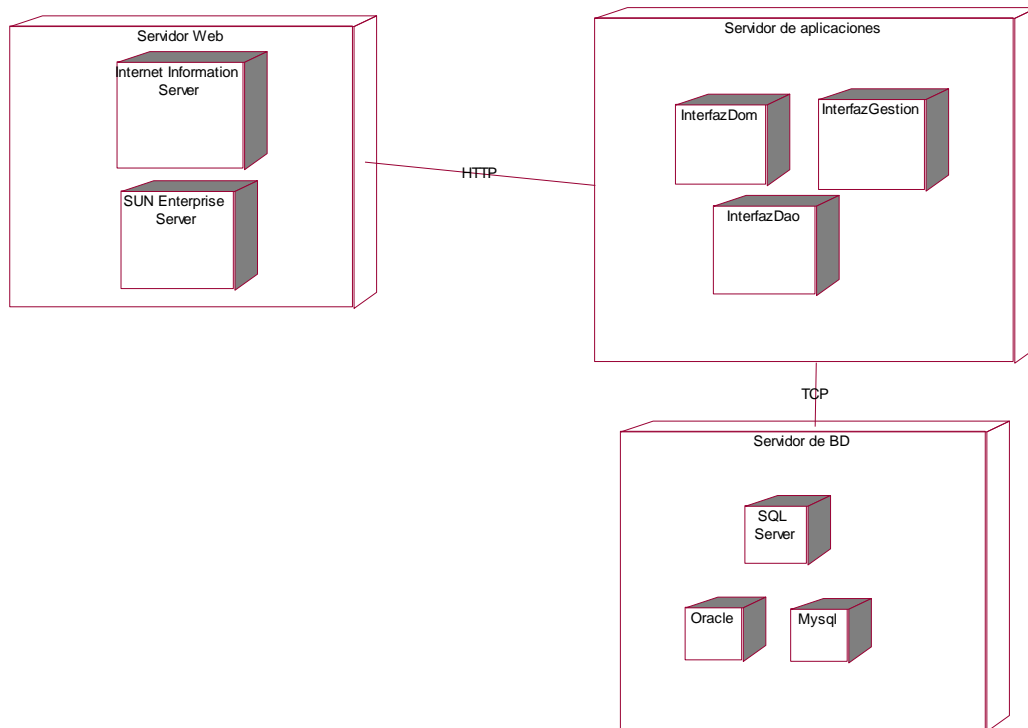


Vista.- Desarrollo Diagrama de componentes general, estilo arquitectónico “N-Tiers / Orientación a objetos”



Vista.- Física

Diagrama.- Despliegue



Vista.- Procesos

Diagrama.- Secuencia (Buscar Reserva)

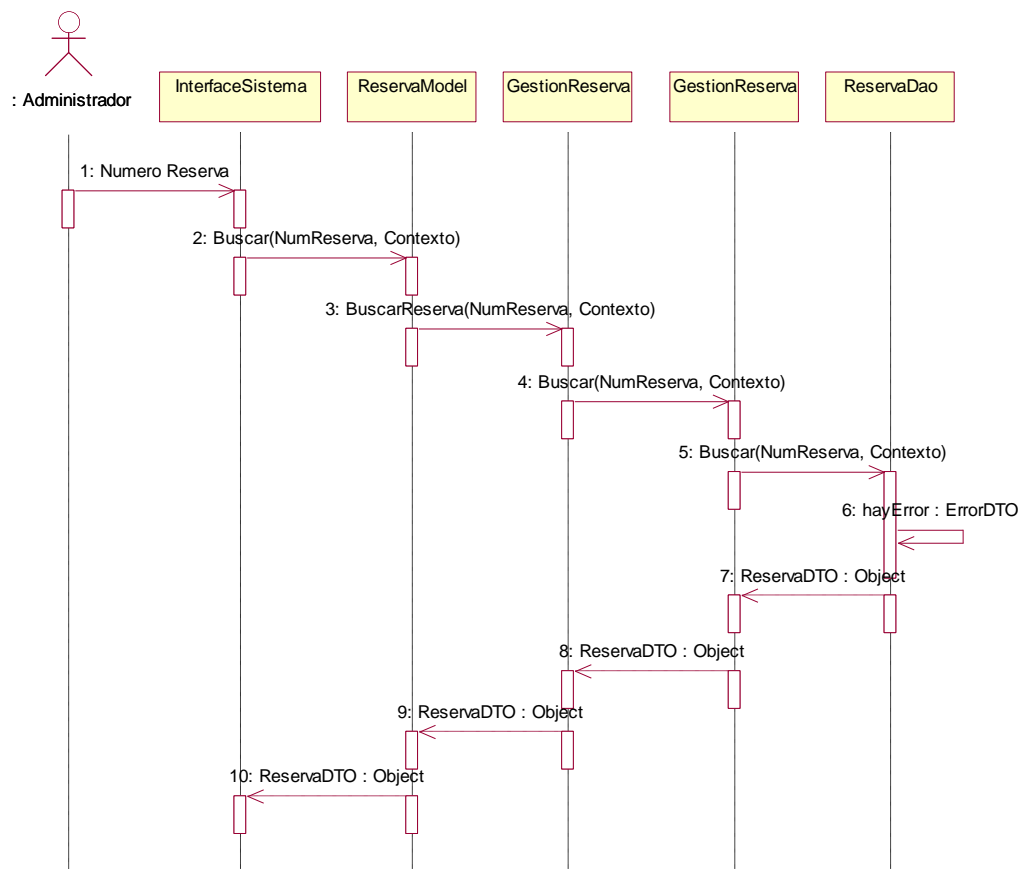


Diagrama de secuencia (Nueva reserva)

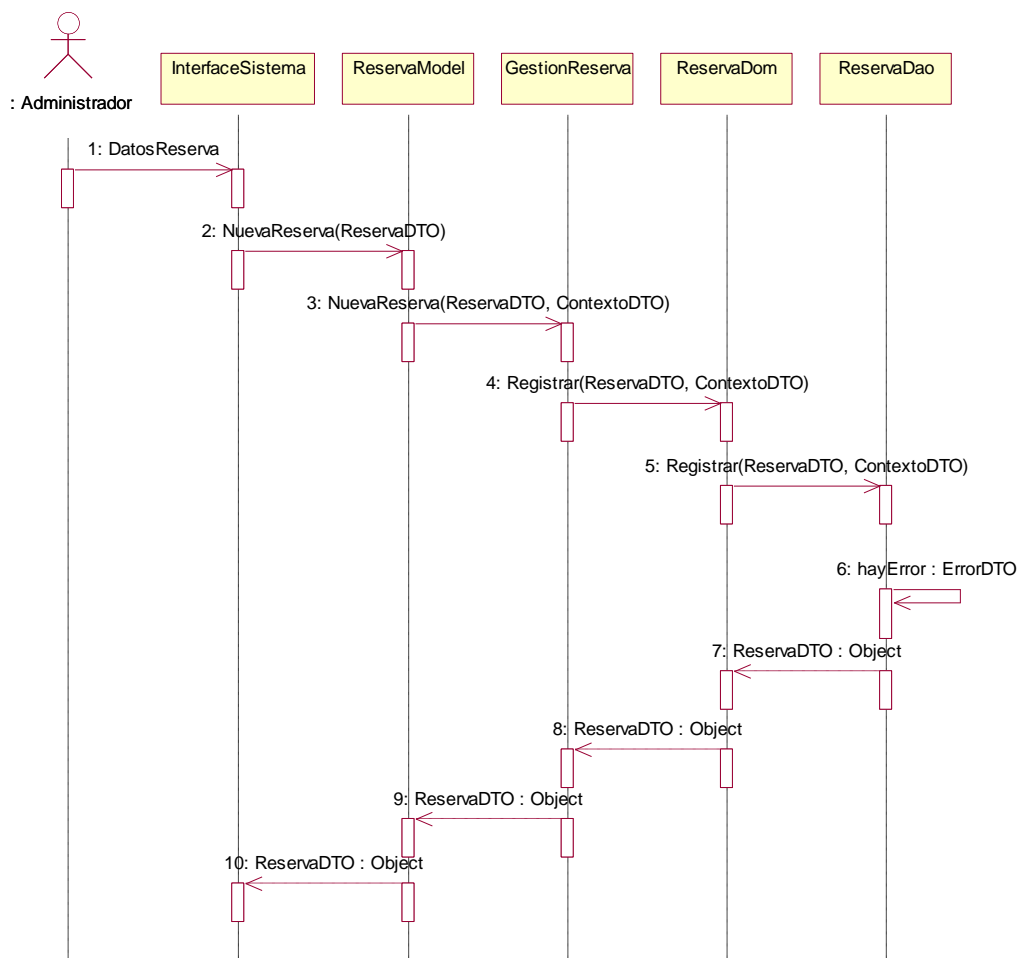
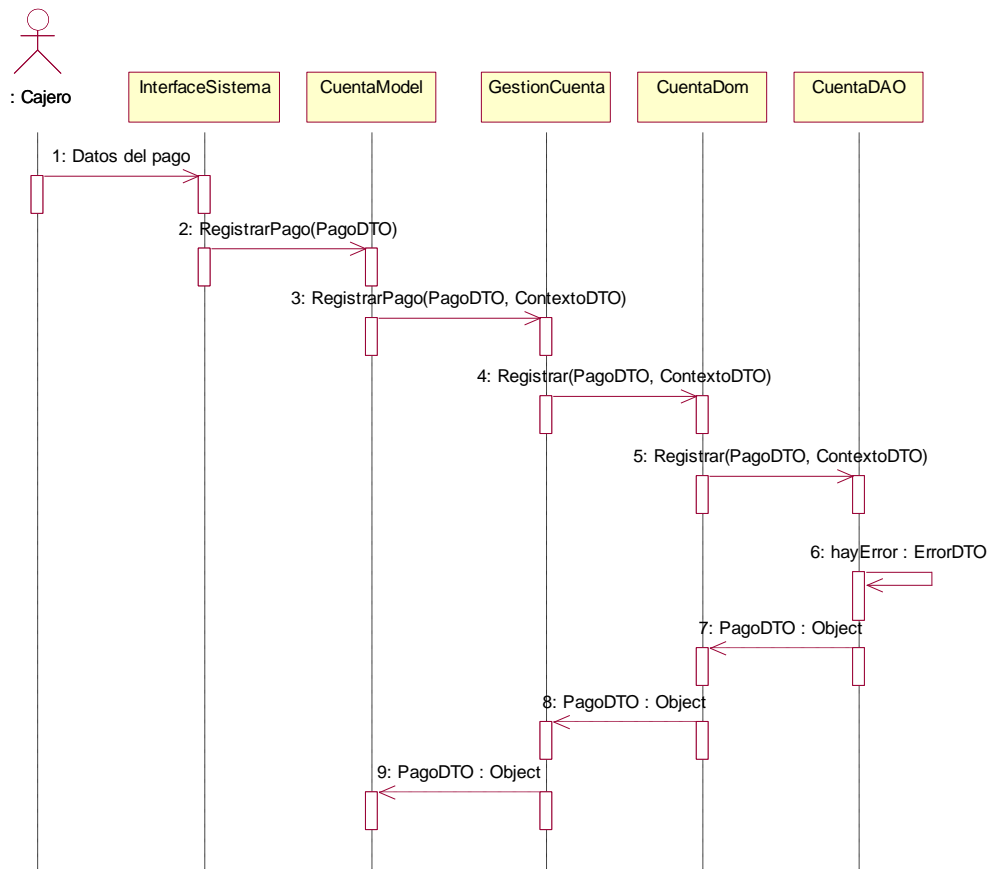
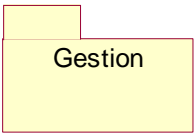
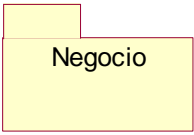
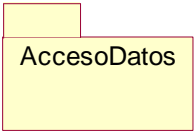


Diagrama de secuencia (Registrar pago)

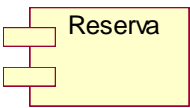
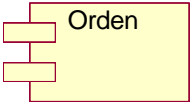


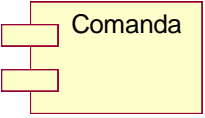
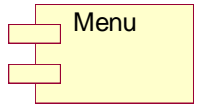
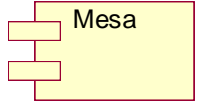
5.5 Consistencia en la cantidad de vistas de la arquitectura.

DESCRIPCION DE MODULOS

Nombre del módulo	descripción	Componentes incluidos
	Modulo que agrupa las clases e interfaces encargadas de orquestar las clases del dominio, agrupa funcionalidades que se acercan más al negocio.	<ul style="list-style-type: none">• GestionCuenta• GestionOrden• GestionComanda• GestionReserva
	Modulo que agrupa todas las clases del negocio (Dominio), cada clase contiene su propia interfaz para exponer la funcionalidad a las otras capas, ejemplo: ReservaDom / IReservaDom. Contiene el CRUD (Create, Read, Update, Delete) del negocio.	<ul style="list-style-type: none">• ReservaDom• CuentaDom• OrdenDom• ClienteDom• ComandaDom• MesaDom• PlatoDom• MenuDom• CompraDom• ProveedorDom
	Contiene las clases que hacen la persistencia a la base de datos, cada clase posee su propia interfaz para exponer la funcionalidad de cada tabla expresada en el modelo de datos.	<ul style="list-style-type: none">• DaoFactory• IDaoFactory• ReservaDao• OrdenDao• ComandaDao• CuentaDao• ClienteDao• ProveedorDao• CompraDao• MesaDao• PlatoDao• MenuDao

DESCRIPCION DE COMPONENTES

Nombre del componente	descripción	Componentes relacionados
	Contiene la lógica para: Nuevas reservas, buscar reservas, eliminar reservas, actualizar reservas.	<ul style="list-style-type: none">• Cliente• Orden Con sus correspondientes interfaces.
	Contiene la lógica para: nuevas órdenes y el CRUD necesario de acuerdo a la funcionalidad del negocio.	<ul style="list-style-type: none">• Reserva• Garzon• Cuenta• Comanda• Mesa

 <p>Comanda</p>	<p>Contiene la lógica para almacenar los pedidos realizados por el cliente. Por cada orden nueva del cliente se genera una nueva comanda. Asociada a una misma Orden y Reserva.</p>	<ul style="list-style-type: none"> • Plato • Orden
 <p>Menu</p>	<p>Contiene la lógica para almacenar el menú del día del restaurant.</p>	<ul style="list-style-type: none"> • Plato
 <p>Mesa</p>	<p>Contiene la lógica de cada una de las mesas del local.</p>	<ul style="list-style-type: none"> • Orden

DESCRIPCION DE CONECTORES

5.6 Arquitectura lógica.

Performances

La arquitectura de software escogida apoya a los requerimientos no funcionales y requerimientos de arquitectura de sistemas descritos en los anexos de este documento.

1. El sistema apoyará hasta 2000 usuarios simultáneos contra la base de datos central en cualquier tiempo dado, y hasta 500 usuarios simultáneos contra los servidores locales en un momento dado.
2. El sistema proporcionará el acceso a la base de datos de catálogo de curso de herencia sin más que una 10 segunda latencia.
3. El sistema debe ser capaz de completar el 80 % de todas las transacciones dentro de 2 minutos.
4. La parte de cliente requerirá el espacio de disco de menos de 20 MB y la RAM de 32 MB.

Calidad

La arquitectura de software apoya las exigencias de calidad, como estipulado en la especificación anexa a este documento.

1. El interfaz de usuario será WEB.
2. El interfaz de usuario del Sistema RESTAURANT será diseñado para la facilidad de uso y será apropiado para asegurar las normas de usabilidad universal establecidas por ISO 9126.
3. Cada despliegue de opciones de pantalla, tendrá la ayuda en línea para el usuario. La ayuda En línea incluirá paso a paso instrucciones en la utilización del Sistema. La ayuda En línea incluirá definiciones para términos y acrónimos.

5.7 Ejemplo de uso.

N/A.

5.8 Detalles de la implementación

La especificación de un sistema intensivo en software tiene como última representación al código fuente de los componentes. Este código indica los más finos detalles del software, por medio de un lenguaje preciso, capaz de ser traducido automáticamente a instrucciones de la maquina. Acompaña al código, las llamadas *previsiones de compilación*, constituidos por todos los elementos de soporte necesarios para realizar la construcción de los componentes a partir del conjunto de códigos. Esta

sección detalla la obtención y uso del paquete de código fuente para el proyecto. De manera de facilitar el uso de este, para las futuras ampliaciones o correcciones del sistema.

5.8.1 Lenguajes y plataformas

La lógica de diseño arquitectónico aplicada en este documento, abre la posibilidad de que la implementación de bajo nivel sea efectuada con lenguajes que solamente cumpla con la característica de Orientación a Objetos (Punto NET, Java, SmallTalk, etc.). Y eso va a depender directamente de las características de los desarrolladores, capacidad de aprendizaje, y en muchos casos opciones propias de la empresa para la cual se efectúa el diseño. Si la implementación se desea desarrollar bajo lenguajes que no cumplan las características mencionadas, se deberá confeccionar una nueva vista que cumpla con los requerimientos funcionales y no funcionales de los stakeholders que lo solicitan.