



Universidad
del Valle de México

LAUREATE INTERNATIONAL UNIVERSITIES®

ACTIVIDAD 3: Proyecto integrador etapa 1

Por siempre responsable de lo que se ha cultivado

Materia: ARQ PARA EL DIS DE SOFTWARE



José Emiliano Jauregui Guzmán

Docente: Mtro. JOSE ANTONIO URQUIDEZ

RAMIREZ

Semana 3: 21 – 28 de Julio



ETAPA 1, 2 Y 3: PROYECTO INTEGRADOR

Índice

I. Análisis de la problemática interna

- 1.1 Introducir y presentar la problemática
- 1.2 Definiciones raíz de los sistemas pertinentes

II. Modelos conceptuales y definición de cambios

- 2.1 Construcción del modelo conceptual
- 2.2 Propuesta de cambio

III. Selección de metodología ágil complementaria

- 3.1 Metodología ágil complementaria

Conclusiones

Fuentes documentales



Introducción

Esta actividad consiste en aplicar los conocimientos adquiridos a lo largo del curso y retomar lo aprendido en cada una de las actividades realizadas, lo que garantiza la transversalidad de los contenidos revisados para fortalecer el desarrollo de competencias y lograr el fin de formación planteado.

Objetivo

El objetivo del Proyecto integrador es construir el diseño arquitectónico de un sistema aplicando una metodología ágil, a partir del planteamiento de un problema dentro de una organización integrada a la industria de los servicios de TI. La finalidad de resolver una problemática derivada de la ausencia de sistemas de información, generando la documentación necesaria para desarrollar software que cubra las necesidades identificadas.

Planteamiento

Un hotel requiere un sistema de administración general que se apegue a las directrices de la industria de la hospitalidad, para lo cual deberá contar con las siguientes características:

1. Reserva de habitaciones: Este deberá poseer un sistema para realizar reservas anticipadas, controlar el número de habitaciones, organizar listas de espera, etc. tomando en cuenta el overbook (sobre venta) que indica el porcentaje de habitaciones que pueden reservarse de cada tipo.
2. Recepción del cliente: Existen dos pasos en este proceso, la asignación de habitación y la recepción. En el primer paso se le indica al cliente el número de habitación y se registra en la máquina, en el segundo se le toman los datos personales al cliente.



3. Facturación: Es el que más varía según el tamaño del hotel. Hay diversas cuentas a incluir en el huésped dependiendo las áreas que tenga el hotel y lo que consume el cliente.
4. Estadísticas: Es de suma importancia en el apartado de gestión interna, nos ofrece una visión general sobre la visión diaria del hotel o durante un periodo determinado. Entre los elementos informáticos de un hotel estaría:
 - a) La base de datos, con la información de las instalaciones, personal, huéspedes, etc.
 - b) Terminales de pantalla, con teclado de recepción, administración, dirección y aquellos puntos que deban introducir datos.
 - c) Impresoras en administración, teletexto, caja y factura a huéspedes.

Fuente: Planteamiento adaptado de la tesis “Diseño de un sistema informático para el c

I. Análisis de la problemática interna

1.1 Introducir y presentar la problemática (Situación actual)

•Con base en el planteamiento y desde tu experiencia particular identificando los conceptos a partir de los cuales podrá realizar el análisis de requerimientos: datos de entrada y de salida, operaciones, flujo de trabajo, los usuarios que van a interactuar con el sistema/aplicación y características de calidad.

Actualmente, muchos hoteles enfrentan dificultades en la gestión integral de sus procesos operativos debido a la falta de un sistema de administración digital eficaz. Estas dificultades incluyen problemas al realizar reservas, errores en la asignación de habitaciones, pérdida o mal registro de los datos de los huéspedes, retrasos en el proceso de facturación y una escasa o nula capacidad de generar reportes estadísticos útiles para la toma de decisiones. Esta situación afecta directamente la experiencia del cliente, la eficiencia del personal y la competitividad del hotel en la industria de la hospitalidad.



Desde una perspectiva técnica, se puede identificar que este tipo de sistema requiere analizar y definir detalladamente los datos de entrada como: datos personales de los huéspedes, fechas de entrada y salida, tipo de habitación, métodos de pago y consumos realizados.

Los datos de salida esperados incluyen: confirmaciones de reserva, reportes de ocupación, facturas detalladas y estadísticas de rendimiento del hotel.

Las operaciones clave incluyen la gestión de reservas, incluyendo listas de espera y sobreventa, el check-in o check-out, la facturación personalizada, y la generación de reportes estadísticos.

En cuanto al flujo de trabajo, se sigue una secuencia que va desde la reserva inicial, la asignación de habitación, la recepción del huésped, su estancia y consumo, hasta la facturación final.

Los usuarios del sistema abarcan desde recepcionistas, personal de administración, dirección general, hasta los propios clientes a través de plataformas en línea. Finalmente, las características de calidad más relevantes para el sistema incluyen la fiabilidad, la disponibilidad continua, la facilidad de uso, la seguridad de los datos y la capacidad de adaptarse a diferentes escalas de operación del hotel.

Requerimientos funcionales

Los requerimientos funcionales son aquellas funciones específicas que el sistema debe ser capaz de realizar.

- Gestión de reservas: Permitir realizar reservas anticipadas para diferentes tipos de habitaciones, validar disponibilidad y aplicar reglas de overbooking y gestionar listas de espera y notificaciones de disponibilidad.
- Recepción de clientes: Registrar los datos personales del huésped, asignar automáticamente o manualmente una habitación según disponibilidad y registrar la entrada y salida del cliente.
- Facturación: Generar facturas automáticas basadas en los consumos realizados, permitir diferentes métodos de pago.



- Gestión de habitaciones: Actualización del estado de la habitación.
- Generación de estadísticas y reportes: Reportes de ocupación diaria, semanal o mensual, por cliente, habitación o servicio y comportamiento del cliente.
- Gestión de usuarios del sistema: Registro de actividades por usuario

Requerimientos no funcionales

Definen las condiciones o restricciones que debe cumplir el sistema, relacionadas con su calidad y comportamiento.

- Disponibilidad: El sistema debe estar disponible las 24 horas del día.
- Usabilidad: Interfaz gráfica amigable e intuitiva para facilitar el uso por parte del personal no técnico.
- Seguridad: Protección de datos personales conforme a regulaciones de privacidad.
- Rendimiento: Capacidad de respuesta rápida en operaciones clave como reservas, facturación y consultas.
- Escalabilidad: Posibilidad de ampliar el sistema para gestionar más habitaciones, sucursales o integrar nuevos módulos.
- Compatibilidad: Compatible con navegadores modernos y dispositivos móviles.

1.2 Definiciones raíz de los sistemas pertinentes

- En esta etapa deberás establecer la definición raíz de la solución propuesta en el planeamiento, en términos de la metodología que se está utilizando y una vez identificados los aspectos relevantes para realizar el diseño arquitectónico del sistema/aplicación realiza una tabla de los requerimientos funcionales y no funcionales.

El sistema propuesto es una aplicación de administración hotelera integral, orientada a la automatización de los procesos operativos claves en la gestión de un hotel: reservas, recepción, facturación y reportes estadísticos. Esta solución estará basada en una arquitectura cliente-servidor con una base de datos relacional,



accesible a través de una interfaz web y móvil, que garantice escalabilidad, rendimiento y seguridad.

Aspectos relevantes para el diseño arquitectónico:

- Arquitectura general: Modelo cliente-servidor, con servidor web, servidor de aplicaciones.
- Base de datos: MySQL para el manejo de datos.
- Roles de usuarios: Recepcionista, administrador, gerente y clientes.
- Interfaz de usuario: Aplicación web accesible desde navegadores y dispositivos móviles.
- Lenguajes y tecnologías: Frontend con JavaScript, backend con Python.
- Requisitos de despliegue: Servidor con acceso 24/7, respaldos automáticos, alta disponibilidad.

ID	Requerimiento funcional	Prioridad	Descripción
RF1	Gestión de reservas	Alta	Permite realizar reservas anticipadas, visualizar disponibilidad.
RF2	Asignación de habitación	Alta	El sistema asigna habitación disponible al momento del check.
RF3	Registro de clientes	Alta	Captura y almacena los datos personales de los huéspedes.
RF4	Gestión de entrada y salida	Alta	Registro de entrada y salida de clientes.
RF5	Facturación automatizada	Alta	Calcula y genera la factura con base en consumo y servicios utilizados.
RF6	Gestión de devoluciones o cambios de habitación	Media	Permite registrar cambios de habitación o cancelaciones con condiciones.
RF7	Generación de estadísticas y reportes	Media	Reportes de ocupación, ingresos, comportamiento de clientes, etc.
RF8	Gestión de usuarios y control de acceso	Alta	Define roles de usuarios y controla el acceso a funcionalidades según permisos.
RF9	Administración del inventario de habitaciones	Media	Control del estado, tipo y disponibilidad de habitaciones.



RF10	Consulta de historial de clientes	Media	Permite revisar las estancias y consumos previos de un huésped.
------	-----------------------------------	-------	---

ID	Requerimiento no funcional	Categoría	Descripción
RNF1	Disponibilidad 24/7	Disponibilidad	El sistema debe estar en línea y operativo todos los días sin interrupciones.
RNF2	Interfaz amigable	Usabilidad	Interfaz gráfica clara, accesible y responsiva para uso de personal.
RNF3	Protección de datos personales	Seguridad	Cumplimiento con normativas de privacidad y encriptación de información sensible.
RNF4	Control de accesos y permisos por roles	Seguridad	Diferentes niveles de permisos según el perfil del usuario.
RNF5	Tiempo de respuesta menor a 3 segundos en operaciones	Rendimiento	Operaciones críticas como reservas y facturación deben responder de forma rápida.
RNF6	Soporte para múltiples usuarios simultáneos	Rendimiento	Capacidad de atender a varios usuarios activos al mismo tiempo sin fallos.
RNF7	Escalabilidad	Escalabilidad	Facilidad para agregar nuevas funcionalidades o módulos en el futuro.
RNF8	Soporte multiplataforma	Compatibilidad	Accesible desde navegadores web modernos y dispositivos móviles.
RNF9	Copias de seguridad diarias automáticas	Confiabilidad	Mecanismo de respaldo para evitar pérdida de datos importantes.
RNF10	Cumplimiento con normativas locales de hospitalidad	Legal	Alineación con las leyes locales de operación hotelera.



Conclusión

Esta primera etapa del proyecto ha sido fundamental para comprender la importancia de identificar correctamente las necesidades y problemáticas que enfrenta una organización real, en este caso, un hotel. Al analizar detalladamente el contexto operativo, fue posible reconocer los procesos clave como las reservas, la recepción de clientes, la facturación y el control estadístico, lo que nos permitió establecer una base sólida para el desarrollo de requerimientos precisos y realistas. Además, aprender a distinguir entre requerimientos funcionales y no funcionales nos permite visualizar mejor cómo debe comportarse el sistema y qué estándares de calidad debe cumplir.

Esta fase inicial también ha resaltado el papel que juega la ingeniería de requerimientos como herramienta para reducir errores, optimizar procesos y facilitar la toma de decisiones tanto en el diseño como en la implementación del sistema. Haber trabajado con conceptos como flujo de trabajo, actores y operaciones nos da una visión clara de cómo los futuros módulos del software deben estructurarse para cumplir con las expectativas del cliente. No me cabe duda que este ejercicio es el primer paso para un desarrollo más elaborado y enfocado en la arquitectura de un software.

Referencia

Pressman, R. (2010). Ingeniería del software. Un enfoque práctico. Recuperado el 25 de julio de 2025, de https://www.academia.edu/15231805/Ingenieria_de_software_enfoque_practico_7_ed_Pressman_PDF

Summerville, I. (2011). Ingeniería del software. Recuperado el 25 de julio de 2025, de https://gc.scalahed.com/recursos/files/r161r/w25469w/ingdelsoftwarelibro9_compressed.pdf

S/f). Researchgate.net. Recuperado el 25 de julio de 2025, de https://www.researchgate.net/profile/Perla-Velasco-Elizondo/publication/281137715_Arquitectura_de_Software_Conceptos_y_Ciclo_de_Desarrollo/links/57144e1408aeebe07c0641ab/Arquitectura-de-Software-Conceptos-y-Ciclo-de-Desarrollo.pdf





Universidad
del Valle de México

LAUREATE INTERNATIONAL UNIVERSITIES®

ACTIVIDAD 5: Proyecto integrador etapa 2

Por siempre responsable de lo que se ha cultivado

Materia: ARQ PARA EL DIS DE SOFTWARE



José Emiliano Jauregui Guzmán

Docente: Mtro. JOSE ANTONIO URQUIDEZ

RAMIREZ

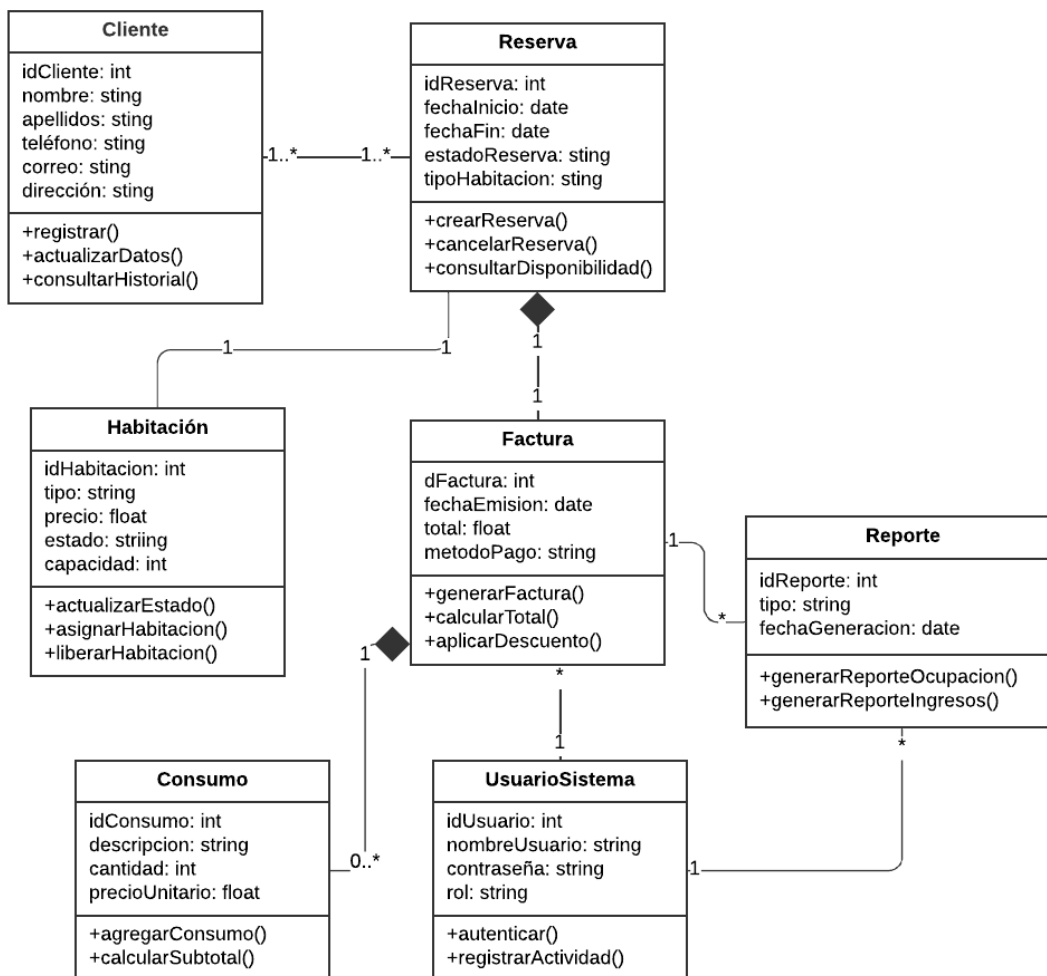
Semana 4: 28 – 24 de Agosto



II. Modelos conceptuales y definición de cambios

2.1 Construcción del modelo conceptual

•A partir de la definición de requerimientos de la primera etapa del proyecto, construye un diagrama estructural UML del sistema, puedes seleccionar entre diagramas de clases, objetos, paquetes, despliegue u otro.



Para este modelo conceptual del sistema hotelero opte por la opción de un diagrama de clases el cual está basado en la identificación de las principales entidades que participan en la operación del sistema y las sus relaciones que tiene.

•Cliente: Representa a los usuarios que realizan reservas en el hotel. Contiene atributos para almacenar su información personal y métodos para registrar y actualizar datos y para consultar su historial de reservas.

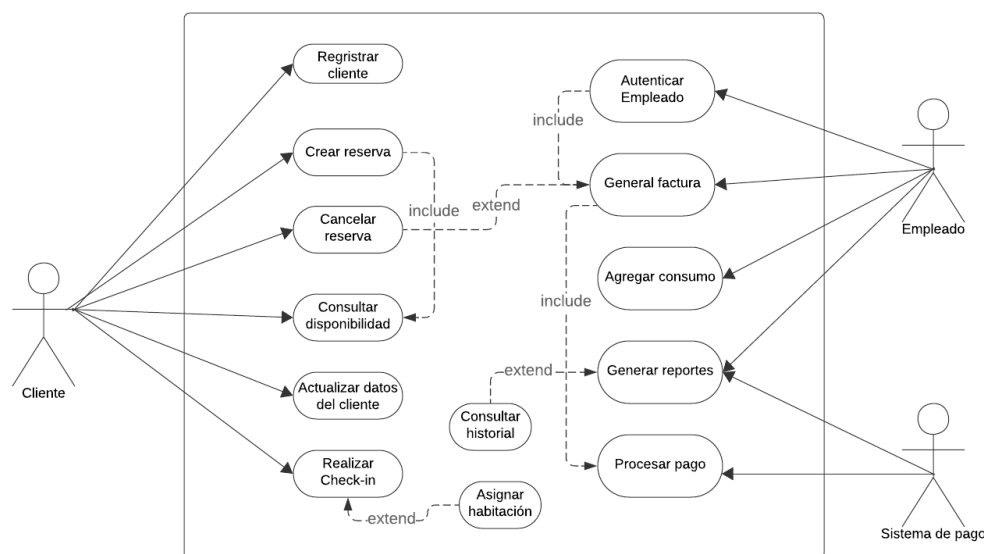


- Reserva: Registra la información sobre las reservaciones hechas por los clientes, incluyendo fechas, estado y tipo de habitación.
- Habitación: Describe las características de las habitaciones disponibles, como tipo, precio, estado y capacidad, tiene métodos para actualizar su estado así como asignarlas.
- Factura: Se encarga de representar los documentos de facturación generados por cada reserva.
- Consumo: Registra los consumos adicionales asociados a una factura.
- UsuarioSistema: Representa a un empleado, con métodos para autenticarse y registrar sus actividades.

Tiene relaciones como que un cliente puede tener varias reservas, una reserva está asociada a una habitación y esta genera facturas con consumos. El modelo tiene la integridad y coherencia de sus datos, facilitando la implementación a futuro.

2.2 Propuesta de cambio

- A partir de la definición de requerimientos de la primera etapa del proyecto, construyecun diagrama de comportamiento UML del sistema, puedes seleccionar entre diagramas de actividades, casos de uso, secuencia, estados, comunicación u otro.



Se usará el diagrama de casos de uso ya que se realizó en una actividad anterior y me resulta con mayor facilidad representar el comportamiento de un sistema hotelero desde la perspectiva de los usuarios y actores externos.

•Actores:

Cliente: Realiza operaciones como registrarse, gestionar reservas, consultar información y efectuar el check-in.

Empleado: Administra procesos internos, como la generación de facturas, agregación de consumos, generación de reportes y actualización del estado de las habitaciones.

Sistema de Pago: Actor externo encargado de procesar transacciones financieras para la facturación.

•Casos de uso clave: Se incluyen escenarios los cuales reflejan los procesos centrales del hotel.

•Relaciones: Se representan acciones obligatorias dentro de un caso de uso principal y opcionales o condicionales.

Se proporciona una visión clara de las interacciones entre actores y el sistema.

Conclusión

El desarrollo de esta segunda etapa del proyecto permitió consolidar los conocimientos sobre modelado de sistemas utilizando UML, aplicados a un entorno real como lo es la administración de un hotel. A través de la construcción del diagrama de clases y del diagrama de casos de uso, fue posible visualizar la estructura del sistema, identificar los actores involucrados y representar las interacciones principales que permiten gestionar reservas, facturación y reportes de manera eficiente. Este proceso también facilitó la comprensión de cómo los requerimientos funcionales y no funcionales definidos en la primera etapa se transforman en representaciones gráficas que guían el diseño e implementación del software.



Además me muestra la importancia de planificar y documentar correctamente el funcionamiento de un sistema antes de programarlo, asegurando que cumpla con las necesidades reales del usuario final. Aprender a identificar relaciones en los casos de uso, así como las asociaciones entre actores y escenarios.

Referencia

Teatro de la Abadía (2021). Diagramas del UML Recuperado el 1 de agosto de 2025, de https://www.teatroabadia.com/es/uploads/documentos/iagramas_del_uml.pdf

SAVUNISEVILLA (Productor). (27 de mayo de 2020) ¿Qué es UML? Recuperado el 1 de agosto de 2025, de <https://www.youtube.com/watch?v=QYzUnWliA1U>

Universitat Politècnica de València - UPV (Productor). (04 de octubre de 2017). Diagrama de clases (UPV). Recuperado el 1 de agosto de 2025, de <https://www.youtube.com/watch?v=JioEGJllg88&list=PLhAp9C4WUHwSld0QtKW5k--7CZ0moxdk1>

Universitat Politècnica de València - UPV (Productor). (04 de octubre de 2017) Diagrama de casos de uso (UPV) Recuperado el 1 de agosto de 2025, de <https://www.youtube.com/watch?v=orvAkFFWo5o&list=PLhAp9C4WUHwSld0QtKW5k--7CZ0moxdk1&index=2>

Universitat Politècnica de València - UPV (Productor). (19 de octubre de 2017) ¿Cómo construir un diagrama de secuencia? Recuperado el 1 de agosto de 2025, de <https://www.youtube.com/watch?v=Q1kH7XKxK5l&list=PLhAp9C4WUHwSld0QtKW5k--7CZ0moxdk1&index=3>

Universitat Politècnica de València - UPV (Productor). (18 de noviembre de 2013). Ingeniería del software II - Tipos de relaciones en UML (UPV). Recuperado el 1 de agosto de 2025, de <https://www.youtube.com/watch?v=zvVzW4LzwTc&list=PLhAp9C4WUHwSld0QtKW5k--7CZ0moxdk1&index=5>





Universidad
del Valle de México

LAUREATE INTERNATIONAL UNIVERSITIES®

ACTIVIDAD 8: Proyecto integrador etapa 3

Por siempre responsable de lo que se ha cultivado

Materia: ARQ PARA EL DIS DE SOFTWARE



José Emiliano Jauregui Guzmán

Docente: Mtro. JOSE ANTONIO URQUIDEZ

RAMIREZ

Semana 6: 11 – 18 de Agosto



III. Selección de metodología ágil complementaria

3.1 Metodología ágil complementaria

En esta etapa se requiere la elaboración de un diseño arquitectónico que amalgame las actividades efectuadas en las primeras dos etapas del proyecto integrador.

- Con base en el material consultado y de acuerdo con tu experiencia elabora un diagrama arquitectónico en **base al modelo C4** que se constituye como un modelo emergente para el diseño ágil de modelos arquitectónicos de *software*, basado en las herramientas de UML y la metodología 4+1.
- En la fase 1 de esta actividad, establece el **contexto** general del sistema identificando los actores que intervienen en los procesos previamente identificados y la manera en que se relacionan entre sí o con el *software*.

Actores principales y relaciones con el sistema

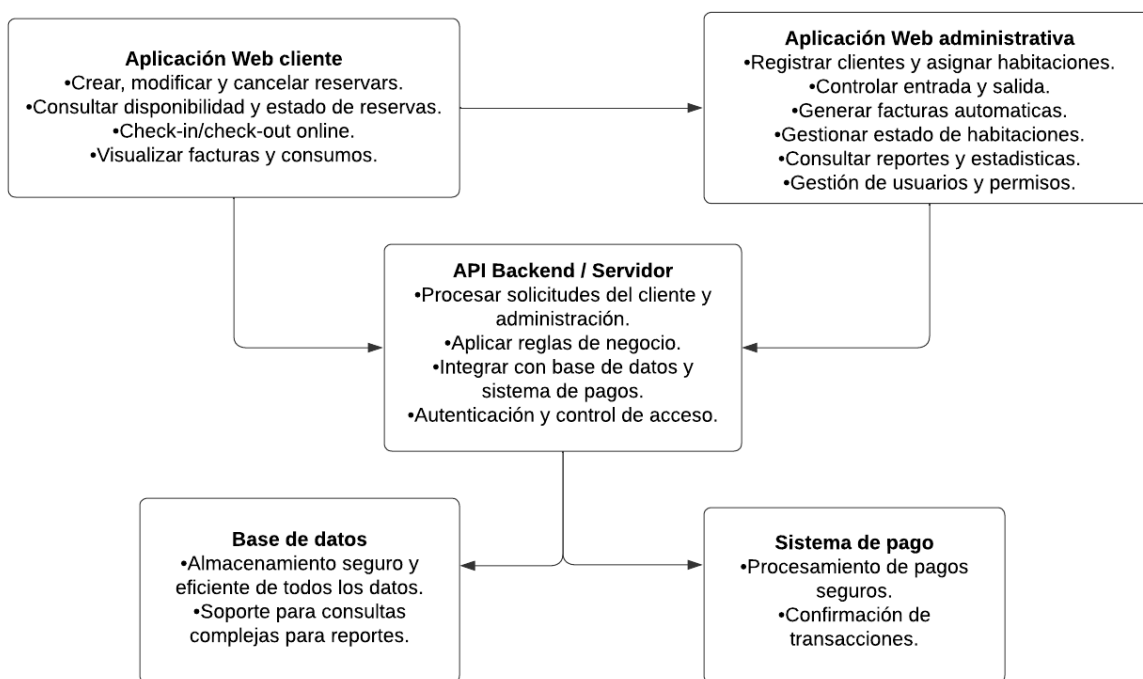
Actor	Descripción	Interacciones con el sistema
Cliente	Persona que realiza reservas y utiliza los servicios del hotel	Reserva habitaciones, consulta disponibilidad, realiza check-in/check-out, consulta facturas y realiza pagos.
Empleado	Personal que administra el proceso de llegada y salida, asigna habitaciones y registra consumos	Asigna habitaciones, registra datos de clientes, genera facturas, gestiona el estado de habitaciones.
Administrador	Personal encargado de supervisar operaciones, generar reportes y gestionar usuarios	Consulta estadísticas, controla usuarios del sistema, configura políticas.
Sistema de Pago	Servicio externo que procesa transacciones financieras	Recibe solicitudes de pago, procesa y confirma transacciones.

Relación entre actores y el sistema

- El **Cliente** accede al sistema vía web o móvil para hacer reservas y gestionar su estancia.



- El **Empleado** interactúa con la interfaz administrativa para gestionar las operaciones diarias.
 - El **Administrador** supervisa y controla el sistema a nivel administrativo.
 - El **Sistema de Pago** se integra para procesar pagos electrónicos.
- En la fase 2, construye los **contenedores** y las funciones, actividades o servicios que el usuario puede utilizar.



Contenedores principales del sistema.

Contenedor	Descripción	Ejemplos
Aplicación web cliente	Interfaz para que los clientes hagan reservas, consultas y gestión de su estancia.	React, vue, HTML, Javascript
Aplicación web administrativa	Interfaz para recepcionistas, administradores y gerentes para gestionar operaciones internas.	React, vue, HTML, Javascript
API Backend / Servidor	Lógica de negocio central: gestión de reservas, facturación, usuarios, reportes, etc.	Python, Node.js, Java, .net



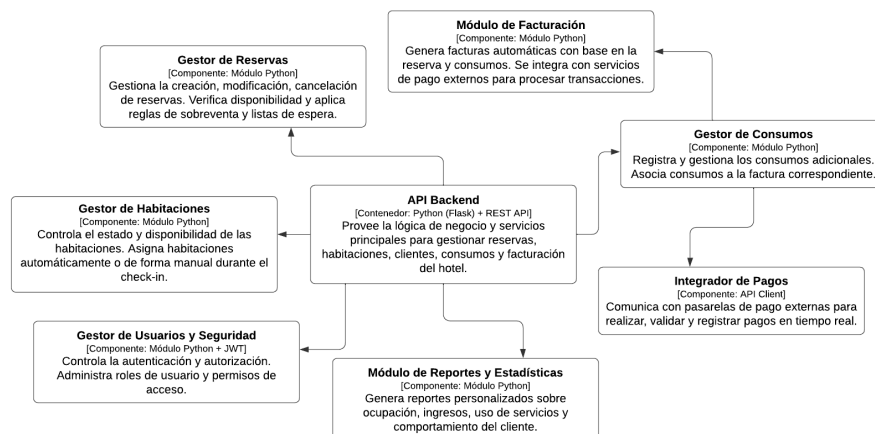
Base de datos	Almacena datos persistentes: clientes, reservas, habitaciones, facturas, consumos, usuarios.	MySQL, PostgreSQL, SQL Server
Sistema de pago	Servicio externo que procesa las transacciones financieras.	PayPas, app bancarias

Funciones y servicios.

Contenedor	Funciones / Servicios
Aplicación web cliente	<ul style="list-style-type: none"> • Crear, modificar y cancelar reservas. • Consultar disponibilidad y estado de reservas. • Check-in/check-out online. • Visualizar facturas y consumos.
Aplicación web administrativa	<ul style="list-style-type: none"> • Registrar clientes y asignar habitaciones. • Controlar entrada y salida. • Generar facturas automáticas. • Gestionar estado de habitaciones. • Consultar reportes y estadísticas. • Gestión de usuarios y permisos
API Backend / Servidor	<ul style="list-style-type: none"> • Procesar solicitudes del cliente y administración. • Aplicar reglas de negocio. • Integrar con base de datos y sistema de pagos. • Autenticación y control de acceso.
Base de datos	<ul style="list-style-type: none"> • Almacenamiento seguro y eficiente de todos los datos. • Soporte para consultas complejas para reportes.
Sistema de pago	Procesamiento de pagos seguros. Confirmación de transacciones y actualización de estados de facturas



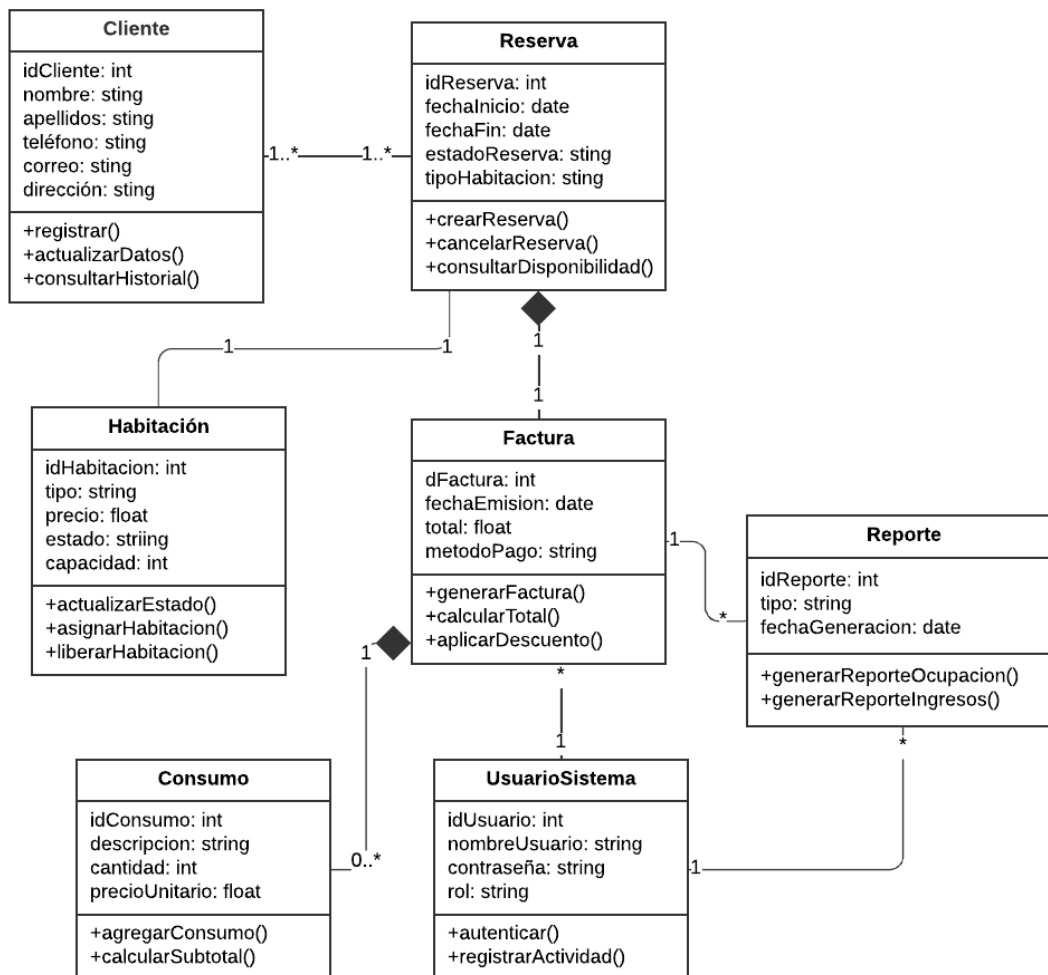
- En la fase 3, construye los **componentes** describiendo de manera detallada su rol, funcionalidad o manera de operar.



Componente	Rol	Funcionalidad Principal	Modo de Operar
Gestor de reservas	Administra el ciclo de vida de una reserva.	Crear, modificar, cancelar reservas, verificar disponibilidad y aplicar overbooking.	Se comunica con el Gestor de Habitaciones y base de datos para gestionar reservas.
Gestor de habitaciones	Controla estado y asignación de habitaciones.	Actualiza estado de habitaciones y asigna según criterios	Informa disponibilidad al gestor de reservas y actualiza base de datos.
Módulo de facturación	Automatiza cobros y generación de facturas.	Calcula cargos por estancia y consumos; genera facturas digitales	Recibe datos de reservas/consumos e integra pagos vía API externa.
Gestor de consumos	Registra servicios utilizados por el cliente.	Asocia consumos a clientes; clasifica por tipo..	Se conecta con el módulo de facturación; actualizado en tiempo real.
Gestor de usuarios y seguridad	Controla acceso y roles de usuario.	Registro/login, asignación de roles, control de accesos.	Valida credenciales, emite tokens y restringe funciones según rol.
Módulo de reportes y estadísticas	Provee datos para análisis y decisiones.	Genera reportes sobre ocupación, ingresos y desempeño.	Consulta base de datos y presenta resultados exportables o visuales
Integrador de pagos	Procesa transacciones financieras externas.	Realiza y confirma pagos digitales.	Usa APIs REST de servicios externos y comunica el estado de pago al sistema.

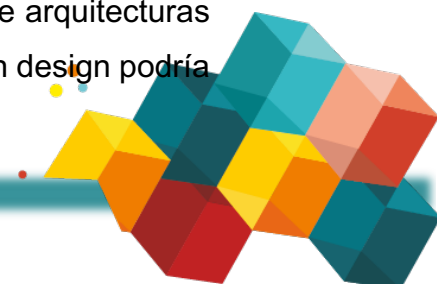


- En la última fase, crea el diagrama de **clases** correspondiente del sistema a partir de los niveles construidos previamente.



Conclusión

Desde mi perspectiva, el modelo C4 ha demostrado ser una herramienta valiosa y clara para diseñar arquitecturas de software en distintos niveles de abstracción. Su capacidad para representar el sistema desde el contexto general hasta el nivel de componentes específicos facilita la comunicación entre todos los involucrados en el desarrollo, desde stakeholders hasta desarrolladores. Por otro lado considero que en proyectos más centrados en microservicios distribuidos, el uso de arquitecturas como los vistos model driven architecture o el enfoque domain driven design podría



ofrecer ventajas más orientadas al negocio, especialmente si se busca mayor autonomía entre módulos y una evolución continua del dominio.

A lo largo de este proyecto integrador, fue posible aplicar conocimientos teóricos en un caso práctico real como el de un sistema de administración hotelera. En la primera etapa, logramos entender a fondo la problemática del cliente, definiendo requerimientos funcionales y no funcionales precisos. En la segunda, aprendimos a estructurar el sistema con modelos UML para visualizar su comportamiento y estructura. Finalmente, en la tercera etapa, pudimos integrar todo este conocimiento utilizando el modelo C4, comprendiendo la importancia de la arquitectura en el éxito de un software. Esta experiencia reforzó mi entendimiento sobre cómo diseñar soluciones sólidas, escalables y alineadas con las necesidades reales del usuario.

Referencia

SAVUNISEVILLA (Productor). (27 de mayo de 2020) ¿Qué es UML? Recuperado el 15 de agosto de 2025, de <https://www.youtube.com/watch?v=QYzUnWliA1U>

Brown, S. (2021). The C4 model for visualising software architecture. Context, containers, components, and. Recuperado el 15 de agosto de 2025, de <https://c4model.com/>

Rodríguez, R. (Productor). (24 de mayo de 2018). *Detonando la arquitectura del software con C4 - Rodrigo Negrete | SG Virtual 14*. Recuperado el 15 de agosto de 2025, de <https://www.youtube.com/watch?v=0fNXnhYfIX8>

Red Hat (2021). ¿Qué es la arquitectura orientada a los servicios (SOA). Recuperado el 15 de agosto de 2025, de <https://www.redhat.com/es/topics/cloud-native-apps/what-is-service-oriented-architecture>

Sanabria, A. y Tascón, J. (2008). Diseño e implementación de la arquitectura orientada a servicios (SOA), en el desarrollo de un software para la empresa CONCIVIN LTDA. Recuperado el 15 de agosto de 2025, de <http://biblioteca.usbbog.edu.co:8080/Biblioteca/BDigital/43335.pdf>

Tune, N. (2020). La relación entre la arquitectura de software y los modelos comerciales (y más). Recuperado el 15 de agosto de 2025, de <https://medium.com/nick-tune-tech-strategy-blog/the-relationship-between-software-architecture-and-business-models-and-more-7209dc3be83e>

