Universidad Siglo 21

Licenciatura en Informática

Seminario de Práctica de Informática

Sistema de Gestión Hotelera para Formación Práctica de Alumnos en carreras de Turismo y Hotelería

Sistema de Gestión de reservas, estadías y habitaciones para hotelería.

Alumno: Lambach Aguirre, Luis Lautaro

Legajo: VINF09474

Profesor titular Disciplinar: Pablo Alejandro Virgolini

Profesor Titular Experto: Ana Carolina Ferreyra

Contenido

SECCION 1: INICIO 5
INTRODUCCION Y RESEÑA HISTORICA 5
ANTECEDENTES 5
PROBLEMÁTICA 5
OBJETIVO DEL PROYECTO 6
OBJETIOS ESPECIFICOS 6
DIAGRAMA DE GANTT 7
OBJETIVOS DEL SISTEMA 7
LIMITES DEL SISTEMA 7
ALCANCES DEL SISTEMA 7
RESTRICCIONES DEL SISTEMA 8
ELICITACION 8
CONOCIMIENTO DEL NEGOCIO 9
DIAGRAMA DE DOMINIO DE NEGOCIO 10
ANALISIS DE PROCESOS 10
PROPUESTA DE SOLUCION 12
SOLUCIÓN FUNCIONAL 12
PROPUESTA TÉCNICA 12
ARQUITECTURA DE DESPLIEGUE 12
REQUERIMIENTOS FUNCIONALES 12
REQUERIMIENTOS NO FUNCIONALES 14
SECCION 2: ANALISIS 16
CASOS DE USO Y ELECCION DE REQUERIMINETOS CANDIDATOS 16
DIAGRAMA DE CASOS DE USO 16
TRAZABILIDAD ENTRE REQUERIMIENTOS FUNCIONALES Y CASOS DE USOS 17
ELECCION DE REQUERIMIENTOS CANDIDATOS 18
IDENTIFICACION DE ACTORES 20
DIAGRAMA DE SECUENCIA 20
DIAGRAMA DE CLASES DEL ANALISIS 25
DIAGRAMA DE PAQUETES DEL ANALISIS 27
Paquete: Gestión de Usuarios 27
Paquete: Gestión de Clientes 28
Paquete: Gestión de Reservas 28
Paquete: Gestión de Habitaciones 28
Paquete: Infraestructura de Datos 29

SECCION 3: DISEÑO 29
DIAGRAMA CLASES DE DISEÑO 29
DIAGRAMAS DE SECUENCIA 31
SUBSISTEMA DE DISEÑO 31
Diagrama de Secuencia para CU-001: Crear Usuarios 31
Diagrama de Secuencia para CU-002: Modificar Usuarios 32
Diagrama de Secuencia para CU-003: Listar Habitaciones 33
Diagrama de Secuencia para CU-004: Modificar Habitación 33
Diagrama de Secuencia para CU-005: Gestionar Perfil Cliente 34
Diagrama de Secuencia para CU-006: Crear Reserva 34
Diagrama de Secuencia para CU-007: Modificar Reserva 35
SECCION 4: IMPLEMENTACION 36
DESPLIEGUE 36
REQUISITOS CONSIDERADOS 36
DIAGRAMA DE DESPLIEGUE 37
NODOS FISICOS 37
COMPONENTES 38
RELACIONES 38
PERSISTENCIA DE DATOS 39
DEFINICION DE BASE DE DATOS PARA PROTOTIPO 39
DIAGRAMA ENTIDAD-RELACION 39
EXPLICACION DETALLADA DER 40
CREACION DE TABLAS 41
INSERCION, CONSULTA Y BORRADO DE REGISTROS 44
COMUNICACIÓN 48
DEFINICION DE LAS COMUNICACIONES DEL SISTEMA 48
REQUERIMIENTOS DE COMUNICACIÓN 48
INTERACCIONES ENTRE COMPONENTES 49
PROTOCOLOS Y ESTANDARES PARA LA COMUNICACION 50
SECCION 5: PRUEBAS 51
PLAN DE PRUEBAS 52
PROCEDIMIENTO DE PRUEBA 54
TABLERO DE EVALUACION DE PRUEBAS 55
TABLERO DE TRATAMIENTO DE DEFECTOS 56

Ilustración 1Diagrama de dominio de negocio 10

Ilustración 2 Diagrama de casos de uso 16
Ilustración 3 Tabla de trazabilidad 17
Ilustración 4 Diagrama de secuencia CU-001 20
Ilustración 5 Diagrama de secuencia CU-002 21
Ilustración 6 Diagrama de secuencia CU-003 21
Ilustración 7 Diagrama de secuencia CU-004 22
Ilustración 8 Diagrama de secuencia CU-005 23
Ilustración 9 Diagrama de secuencia CU-006 23
Ilustración 10 Diagrama de secuencia CU-007 24
Ilustración 11 Diagrama de clases del analisis 25
Ilustración 12 Diagrama de paquetes del análisis 27
Ilustración 13 Diagrama de clases del diseño 30
Ilustración 14 Diagrama de secuencia de diseño CU-001 32
Ilustración 15 Diagrama de secuencia de diseño CU-002 32
Ilustración 16 Diagrama de secuencia de diseño CU-003 33
Ilustración 17 Diagrama de secuencia de diseño CU-004 33
Ilustración 18Diagrama de secuencia de diseño CU-005 34
Ilustración 19 Diagrama de secuencia de diseño CU-006 34
Ilustración 20 Diagrama de secuencia de diseño CU-007 35
Ilustración 21Tabla de Trazabilidad Requisitos Funcionales Candidatos - Casos de Uso 37
Ilustración 22 Diagrama de despliegue 37
Ilustración 23Diagrama Entidad Relación 40
Ilustración 24 Creación Tabla Habitaciones 41
Ilustración 25 Poblar Tabla Habitaciones 41
Ilustración 26 Creacion Tabla Usuarios 42
Ilustración 27Poblar Tabla Usuarios 42
Ilustración 28 Creacion Tabla Clientes 42
Ilustración 29 Creacion Tabla Reservas 43
Ilustración 30 Inserción de registros tabla Clientes 44
Ilustración 31Consulta de Tabla Clientes 44
Ilustración 32 Borrado de Registro Tabla clientes 45
Ilustración 33 Confirmación de borrado de registro tabla Clientes 45
Ilustración 34Inserción de registros tabla Reservas 45
Ilustración 35Consulta Tabla Reservas 46
Ilustración 36 Borrado de registro tabla Reservas 46
Ilustración 37Confirmacion de borrado de registro tabla Reservas 46
Ilustración 38 Consulta tabla Usuarios 47
Ilustración 39 Consulta tabla Habitaciones 47

Tabla 1 Diagrama de Gantt 7
Tabla 2 Requerimientos funcionales 14
Tabla 3 Requerimientos no funcionales 15
Tabla 4 Requerimientos Candidatos 18

Tabla 5 Tabla de trazabilidad plan de prueba - casos de uso - requisitos funcionales candidatos 54
Tabla 6 Tabla de evaluación de pruebas 56
Tabla 7 Tabla de tratamiento de defectos 57

SECCION 1: INICIO

INTRODUCCION Y RESEÑA HISTORICA

Este proyecto tiene como objetivo desarrollar un sistema de tipo TPS orientado a la industria hotelera con fines educativos, comúnmente conocido en la industria como Sistema de Gestión Hotelera o Property Management System (PMS).

La Escuela Internacional de Turismo y Hotelería Vatel, ubicada en la ciudad de Salta, provincia de Salta, es una institución con una sólida trayectoria en la formación de profesionales en el sector Hotelero. Desde su fundación en 2008, Vatel ha sido reconocida por su enfoque práctico y riguroso en la enseñanza, preparando a sus alumnos para enfrentar los desafíos de la industria hotelera a nivel global.

A lo largo de los años, Vatel ha evolucionado para adaptarse a las demandas cambiantes de la industria, incorporando nuevas tecnologías y metodologías de enseñanza para mantener su currículum actualizado y relevante. No obstante, uno de los desafíos recurrentes ha sido la integración efectiva de herramientas tecnológicas avanzadas, como los Sistemas de Gestión Hotelera (PMS), en el proceso educativo.

ANTECEDENTES

Los PMS son herramientas esenciales en la operación diaria de cualquier hotel, gestionando aspectos críticos como reservas, disponibilidad de habitaciones, check-in/check-out, coordinación de housekeeping, y mantenimiento. La familiaridad con estos sistemas es fundamental para cualquier profesional de la industria hotelera, ya que la eficiencia y la capacidad de respuesta del personal dependen en gran medida del manejo adecuado de estas plataformas.

En la Escuela Internacional de Turismo y Hotelería Vatel, se ha identificado que los egresados, aunque cuentan con una sólida formación teórica, carecen de la experiencia práctica necesaria en el uso de PMS debido a las limitaciones presupuestarias de la institución. Las licencias de los PMS comerciales son costosas, y trasladar estos costos a los estudiantes haría que la educación fuera inaccesible para muchos, lo que va en contra de la misión de la escuela de ofrecer una educación de calidad a un costo razonable. Además, en Argentina, las nuevas generaciones muestran una baja alfabetización informática en computadoras, con solo el 61% de niños y adolescentes urbanos accediendo a una PC y apenas el 7% usándolas diariamente para fines educativos, debido a brechas socioeconómicas, currículos desactualizados y falta de capacitación docente (UNICEF, INDEC, 2023). En contraste, el 95% tiene celulares con internet desde los 9.6 años, usando dispositivos móviles intensivamente (90% para tareas escolares), lo que resalta la necesidad de un PMS educativo accesible, como el propuesto, para cerrar estas brechas tecnológicas y prácticas en un entorno controlado.

PROBLEMÁTICA

La falta de acceso a un PMS durante la formación académica, de los alumnos de VATEL, ha generado una brecha significativa entre el conocimiento teórico y las habilidades prácticas de los egresados. Al ingresar al mercado laboral, muchos de ellos se encuentran con la dificultad de adaptarse rápidamente al uso de estos sistemas en un entorno real, lo que puede afectar su desempeño inicial y su competitividad en el mercado laboral.

Esta situación también afecta la reputación de la escuela, ya que los empleadores esperan que los graduados de una institución de prestigio como VATEL estén completamente preparados para las demandas de la industria. Además, la falta de formación práctica en PMS limita la capacidad de los alumnos para participar en prácticas profesionales de alto nivel, donde el manejo de estas herramientas es un requisito fundamental.

OBJETIVO DEL PROYECTO

Este proyecto busca desarrollar un Sistema de Gestión Hotelera (PMS) educativo que sirva como una herramienta práctica y transformadora para la formación de estudiantes en la Escuela Internacional de Turismo y Hotelería Vatel. Diseñado para emular las funciones esenciales de un PMS comercial, el sistema permitirá a los alumnos dominar la gestión de reservas, check-in/check-out, housekeeping y análisis de ocupación en un entorno seguro y controlado. Al ofrecerse como un obsequio a la institución, elimina los elevados costos de licencias comerciales, garantizando una solución accesible y sostenible que se alinea con la misión de Vatel de brindar educación de calidad a un costo razonable. Más allá de Vatel, el proyecto aspira a extender su impacto donando el software a instituciones educativas de bajos recursos, como el Colegio Alfarcito, ubicado en la remota Quebrada del Toro, en la Puna salteña. Fundado en 2006 por el Padre Sigfrido Moroder (Padre Chifri) como un pilar de desarrollo comunitario, este colegio atiende a 160 estudiantes de 25 comunidades kollas dispersas en 5.200 km², enfrentando desafíos de pobreza, aislamiento y necesidades básicas como útiles escolares, ropa y becas. Con orientaciones en turismo, agropecuaria y oficios, Alfarcito impulsa el turismo rural comunitario como motor de crecimiento económico, y este PMS fortalecería su formación técnica, cerrando brechas educativas y empoderando a jóvenes en regiones marginadas.

A través de este sistema, los estudiantes de Vatel practicarán desde su primer año, adquiriendo competencias prácticas que los prepararán para destacar en el exigente mercado laboral hotelero. El proyecto no solo elevará la calidad educativa de Vatel, consolidándola como referente en la formación de profesionales, sino que también promoverá un impacto social profundo al democratizar el acceso a herramientas tecnológicas en comunidades vulnerables, fomentando un turismo inclusivo y sostenible que impulse el desarrollo regional.

.

OBJETIOS ESPECIFICOS

Analizar el Modelo de Negocio: Evaluar las necesidades y definir los requisitos del sistema basado en las problemáticas educativas de la Institución Educativa VATEL.

Diseñar el Sistema: Desarrollar el diseño técnico y conceptual del sistema, incluyendo arquitectura, interfaz y principales módulos.

Construir el Sistema: Implementar el sistema de forma iterativa, cumpliendo con los requerimientos prioritarios y realizando pruebas continuas.

Gestionar Versiones de Software: Implementar un plan de versionado para mantener, actualizar y mejorar el sistema a lo largo del tiempo.

DIAGRAMA DE GANTT

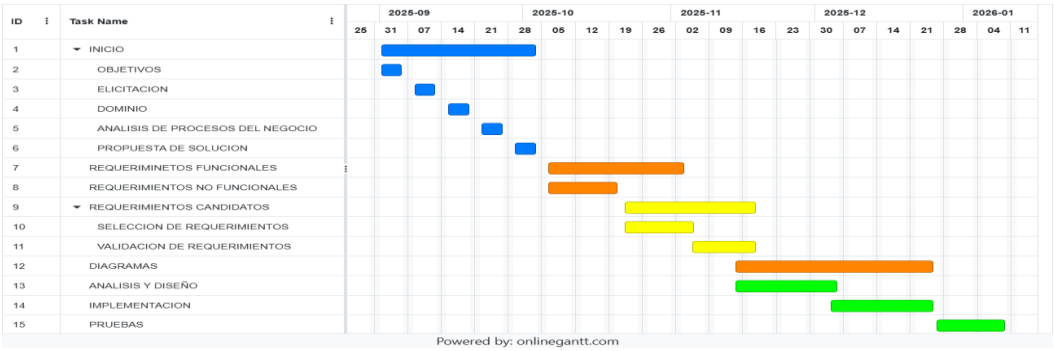

Tabla 1 Diagrama de Gantt

OBJETIVOS DEL SISTEMA

LIMITES DEL SISTEMA

Entorno Educativo: El sistema está diseñado exclusivamente para su uso en un entorno educativo, inicialmente dentro de la Escuela Internacional de Turismo y Hotelería Vatel, con la posibilidad de extenderse a otras instituciones educativas de bajos recursos, como el Colegio Alfarcito. No está destinado para operaciones comerciales en hoteles reales.

Funcionalidades Específicas: El sistema se limitará a las funciones básicas de un PMS necesarias para la formación práctica de los estudiantes, como gestión de reservas, check-in/check-out, perfiles de clientes, análisis de ocupación y módulo de housekeeping, excluyendo módulos avanzados o específicos como facturación, gestión de inventarios o integración con sistemas de reservas en línea.

Configuración de habitaciones: En esta primera iteración del software, la plaza de habitaciones y su configuración (como número, tipo y características) serán predefinidas y no accesibles para los usuarios finales (estudiantes, profesores o personal administrativo). Dado que el sistema es de práctica educativa, estas modificaciones no son necesarias para los objetivos de aprendizaje iniciales.

Usuarios: Los usuarios del sistema serán exclusivamente estudiantes, profesores y personal administrativo de las instituciones educativas donde se implemente. No se contemplará el uso por parte de clientes externos ni la integración con plataformas comerciales de reservas en línea.

ALCANCES DEL SISTEMA

Gestión de Reservas: El sistema permitirá realizar operaciones CRUD (crear, leer, actualizar, eliminar) sobre las reservas, gestionar las asignaciones de habitaciones predefinidas y mantener un registro de estas para el análisis de ocupación futura.

Perfiles de Clientes: Se podrán crear, leer, actualizar y eliminar perfiles de clientes, almacenando información relevante para la gestión de reservas, como nombre completo, documento de identidad, correo electrónico y número de teléfono.

Creación de Usuarios: El sistema incluirá un módulo simple para la creación de usuarios, permitiendo registrar cuentas para estudiantes, profesores y personal

administrativo con roles específicos (como recepcionista, encargado de reservas o housekeeping), asegurando un acceso controlado al sistema.

Check-in y Check-out: Los usuarios podrán realizar el check-in y check-out de las reservas, gestionando el flujo de clientes y actualizando la disponibilidad de habitaciones predefinidas en el sistema.

Análisis de Ocupación: El sistema permitirá analizar las reservas futuras para estimar la ocupación y la disponibilidad de habitaciones en fechas determinadas, generando reportes accesibles para usuarios autorizados.

Módulo de Housekeeping: Incluirá funcionalidades para que el personal de housekeeping pueda actualizar el estado de las habitaciones predefinidas (limpia, sucia, en mantenimiento, inspeccionada, lista para ocupar), sin posibilidad de modificar su configuración o tipo en esta iteración.

RESTRICCIONES DEL SISTEMA

No Facturación: El sistema no incluirá un módulo de facturación, limitándose únicamente a la gestión operativa de las reservas y el control de ocupación y disponibilidad de habitaciones.

Licencia y Distribución: El sistema estará restringido a su uso dentro de las instituciones educativas, como la Escuela Internacional de Turismo y Hotelería Vatel y otras de bajos recursos, como el Colegio Alfarcito, y no se distribuirá ni comercializará externamente.

Recursos Técnicos: El desarrollo y la implementación del sistema estarán limitados por los recursos disponibles, incluyendo el tiempo, presupuesto y la infraestructura tecnológica de la institución educativa.

Módulo administrador de habitaciones: En esta primera iteración, el sistema no incluirá un módulo para la administración de habitaciones, por lo que la creación, modificación o eliminación de habitaciones no estará disponible para los usuarios, ya que las habitaciones serán predefinidas para fines educativos.

Capacitación y Soporte: La capacitación para el uso del sistema será interna, y el soporte técnico estará limitado a lo que el equipo de desarrollo pueda ofrecer, sin servicios externos de soporte.

Portabilidad y Recursos Limitados: El sistema está diseñado para ser instalado localmente en una única PC con recursos técnicos mínimos, sin requerir conexión en red, priorizando la portabilidad y la facilidad de uso en entornos educativos con infraestructura limitada.

ELICITACION

Se llevó a cabo una entrevista con la Rectora de la Escuela Internacional de Turismo y Hotelería Vatel, Sra. Verónica Velardo, para comprender las necesidades que el sistema a desarrollar debe cubrir. La idea inicial es implementar el sistema en una PC que se utilizará durante las cátedras de Housekeeping, Mantenimiento Hotelero, Front Office, Reservas y Ventas. En estas clases, los alumnos realizarán simulaciones de roles, asumiendo funciones de housekeeping, mantenimiento, recepcionista y coordinador de reservas.

Para complementar la información obtenida de la Rectora Velardo, se coordinó una entrevista con la Sra. Marianela Salva Storniolo, Gerente Operativa del Hotel Sheraton

Salta, con el fin de entender en profundidad las operaciones necesarias para las funciones del sistema.

CONOCIMIENTO DEL NEGOCIO

Tras la entrevista, se definieron los siguientes requerimientos para el PMS:

Configuración de Habitaciones: El sistema debe permitir modificar el estado (clean, dirty, maintenance, inspected, pick up) de las habitaciones predeterminadas (twin, king, superior, suite), los cuales serán actualizados exclusivamente por el personal de housekeeping y mantenimiento.

Gestión de Tarifas: Cada tipo de habitación tendrá una tarifa específica que solo podrá ser modificada por el agente de reservas.

Generación de Reportes: El PMS debe generar un reporte para una fecha dada, indicando la cantidad de reservas por tipo de habitación y, en función de estas reservas, mostrar la disponibilidad de habitaciones por tipo y el porcentaje de ocupación esperado. Este reporte debe estar accesible para todos los usuarios.

Gestión de Perfiles y Reservas: El encargado de reservas debe poder crear y gestionar perfiles de clientes, realizar reservas, y llevar a cabo operaciones CRUD. Además, debe existir una opción de "trace" o notas donde el agente de reservas pueda agregar observaciones sobre particularidades de la reserva.

Asignación de Habitaciones y Check-in: Los recepcionistas deben poder asignar habitaciones a las reservas y realizar el check-in cuando la habitación asignada esté en estado "clean". También deben poder ver la tarifa asignada a la reserva y agregar cargos extras, sumando al total de la misma.

Check-out: El sistema debe permitir realizar el check-out de reservas que estén en estado de check-in. Al hacer check-out, el recepcionista debe poder imprimir un reporte detallado de la reserva, y la habitación asignada debe cambiar su estado a "dirty".

DIAGRAMA DE DOMINIO DE NEGOCIO

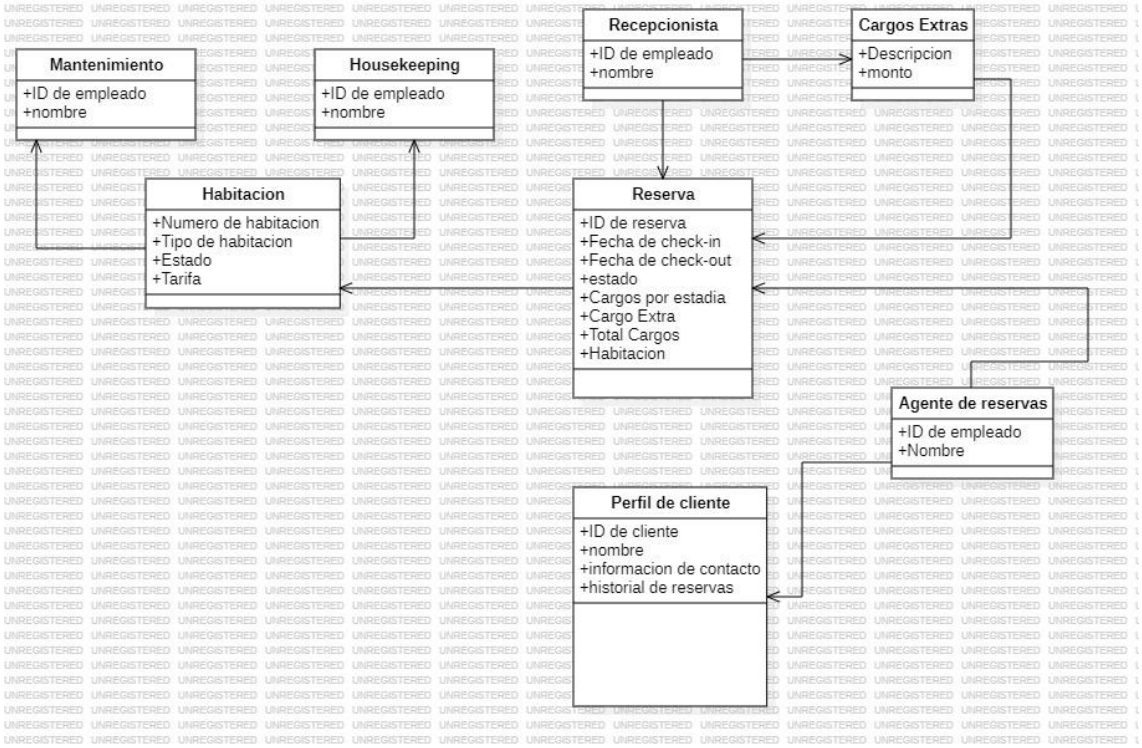

Ilustración 1Diagrama de dominio de negocio

ANALISIS DE PROCESOS

Proceso N° 1: Gestión de Reservas

Roles: Recepcionista, Encargado de Reservas

Pasos:

El cliente se comunica para realizar una reserva (presencial, telefónica o en línea).

El recepcionista o encargado de reservas crea un perfil de cliente (si es un cliente nuevo).

Se verifica la disponibilidad de habitaciones según el tipo solicitado (twin, king, suite, etc.).

El encargado de reservas crea una reserva en el sistema asociada al perfil del cliente.

Se confirma la reserva y se asigna una tarifa específica a la habitación seleccionada.

Se genera un número de confirmación y se envía al cliente.

El sistema actualiza la disponibilidad de habitaciones en tiempo real.

Proceso N° 2: Check-in de Reserva

Roles: Recepcionista

Pasos:

El cliente llega al hotel en la fecha de su reserva.

El recepcionista verifica la reserva en el sistema.

Se comprueba el estado de la habitación asignada (debe estar en estado "clean").

Si la habitación está lista, se procede al check-in en el sistema.

El sistema actualiza el estado de la reserva a "check-in" y el estado de la habitación a "ocupada".

El cliente recibe la llave y es guiado a su habitación.

El sistema registra la tarifa y las condiciones de la estancia.

Proceso N° 3: Check-out de Reserva

Roles: Recepcionista

Pasos:

El cliente informa su salida y se dirige a la recepción.

El recepcionista accede a la reserva en el sistema y revisa los cargos adicionales (si los hay).

Se genera un reporte de la estancia, incluyendo el detalle de la tarifa y los cargos extra.

El cliente revisa el total y se procede a su liquidación (sin incluir facturación en el sistema).

Se realiza el check-out en el sistema, cambiando el estado de la habitación a "dirty".

El sistema actualiza la reserva como completada y libera la habitación para futuras reservas.

Proceso N° 4: Actualización del Estado de Habitaciones

Roles: Housekeeping, Mantenimiento

Pasos:

El equipo de housekeeping o mantenimiento accede al sistema para verificar el estado de las habitaciones.

Se selecciona la habitación y se actualiza su estado según el trabajo realizado (clean, dirty, maintenance, inspected, pick up).

El sistema actualiza el estado de la habitación en tiempo real, disponible para los otros roles (recepcionista, encargado de reservas).

Si la habitación está lista y limpia, queda disponible para nuevas reservas.

Proceso N° 5: Generación de Reportes de Ocupación

Roles: Encargado de Reservas, Recepcionista, Dirección

Pasos:

El usuario solicita un reporte de ocupación desde el sistema para una fecha específica.

El sistema genera un reporte que incluye:

Número de reservas por tipo de habitación.

Cantidad de habitaciones disponibles por tipo.

Porcentaje de ocupación proyectada.

El reporte es consultado y descargado por el usuario.

El reporte se utiliza para tomar decisiones sobre futuras asignaciones o tarifas.

PROPUESTA DE SOLUCION

Una vez realizado el diagnóstico, se presenta una propuesta integral que abarca la solución funcional, la propuesta técnica, la arquitectura del sistema y los requerimientos a incluir.

SOLUCIÓN FUNCIONAL

El sistema propuesto es un PMS educativo que permitirá a los alumnos realizar simulaciones de roles hoteleros, abarcando áreas clave como housekeeping, mantenimiento, reservas, front office, y ventas. El sistema cubrirá las necesidades operativas identificadas, como la gestión de reservas, asignación de habitaciones, check-in/check-out, actualización de estados de habitaciones, y generación de reportes de ocupación.

PROPUESTA TÉCNICA

El sistema se desarrollará utilizando el lenguaje de programación Java, con MySQL como base de datos. Este enfoque garantiza un alto grado de flexibilidad y escalabilidad para futuras mejoras. Además, Java permite una implementación sólida y multiplataforma.

ARQUITECTURA DE DESPLIEGUE

Dado que la Escuela Vatel no cuenta con la infraestructura de red necesaria para implementar un sistema cliente-servidor, el sistema se diseñará para ser instalado de manera local en una única PC, donde la base de datos también residirá. Esto significa que no habrá conexión en tiempo real entre múltiples PC, ya que la escuela no dispone de la estructura necesaria para ello. La solución está optimizada para funcionar en un entorno educativo cerrado sin depender de conexiones de red complejas.

REQUERIMIENTOS FUNCIONALES

CODIGO	REQUERIMIENTO
RQF-001	EL SISTEMA DEBE PERMITIR CREAR USUARIOS CON SUS ROLES Y PERMISOS
RQF-002	EL SISTEMA DEBE PERMITIR MODIFICAR LA INFORMACION DEL USUARIO, ROLES Y PERMISOS
RQF-003	El sistema debe permitir listar todas las habitaciones registradas.
RQF-004	El sistema debe permitir asignar un estado a la habitación (clean, dirty, maintenance, inspected, pick up) que será modificado solo por el personal de housekeeping y mantenimiento.
RQF-005	El sistema debe mostrar en tiempo real el estado actual de las habitaciones en la interfaz de usuario.

RQF-006	El sistema debe permitir crear un perfil de cliente con los siguientes datos: nombre completo, documento de identidad, correo electrónico, número de teléfono.
RQF-007	El sistema debe permitir listar todos los perfiles de clientes existentes.
RQF-008	El sistema debe permitir modificar la información de un perfil de cliente existente.
RQF-009	El sistema no debe permitir borrar perfiles de clientes, excepto al administrador del sistema.
RQF-010	El sistema debe almacenar el historial de reservas del cliente en su perfil.
RQF-011	El sistema debe permitir crear una reserva asociada a un perfil de cliente existente.
RQF-012	El sistema debe permitir especificar las fechas de check-in y check-out en la creación de la reserva.
RQF-013	El sistema debe permitir asignar una habitación disponible a la reserva en el momento de su creación.
RQF-014	El sistema debe permitir modificar una reserva existente, incluyendo la habitación asignada y las fechas de estancia.
RQF-015	El sistema debe permitir cancelar reservas.
RQF-016	El sistema debe permitir agregar notas o "trace" a la reserva, en las que el agente de reservas pueda detallar particularidades.
RQF-017	El sistema debe permitir que el personal de housekeeping actualice el estado de las habitaciones (clean, dirty, inspected).
RQF-018	El sistema debe permitir que el personal de mantenimiento actualice el estado de las habitaciones a "maintenance" cuando sea necesario.
RQF-019	El sistema debe permitir que el personal de housekeeping asigne habitaciones como "pick up" cuando estén listas para ser ocupadas.
RQF-020	Solo el personal autorizado (housekeeping y mantenimiento) podrá modificar el estado de las habitaciones.
RQF-021	El sistema debe permitir que los recepcionistas realicen el check-in de una reserva cuando la habitación asignada esté en estado "clean, inspected o pick up".
RQF-022	El sistema debe permitir al recepcionista ver la tarifa asignada a la reserva al momento del check-in.
RQF-023	El sistema debe permitir agregar cargos extras a la reserva durante el check-in o a lo largo de la estancia del cliente.
RQF-024	El sistema debe permitir que los recepcionistas realicen el check-out de una reserva activa.
RQF-025	Al realizar el check-out, el sistema debe cambiar automáticamente el estado de la habitación a "dirty".
RQF-026	El sistema debe generar un reporte detallado al check-out con el resumen de cargos totales, incluyendo la tarifa y cualquier cargo extra.
RQF-027	El sistema debe permitir generar un reporte detallado de cada reserva, que incluya Fechas de la reserva (check-in, check-out). Cliente asociado a la reserva. Habitación asignada. Tarifa de la habitación. Cargos extras agregados durante la estancia. Total de la reserva.
RQF-028	El sistema debe permitir a los recepcionistas imprimir este reporte al realizar el check-out.
RQF-029	El reporte debe estar disponible para cualquier usuario autorizado que lo solicite.

RQF-030	El sistema debe permitir generar un reporte que muestre la ocupación esperada en una fecha dada.
RQF-031	El reporte debe ser accesible para todos los usuarios con permisos adecuados (reservas, recepción, dirección).
RQF-032	El reporte debe poder ser exportado en formato PDF o impreso directamente desde el sistema.

Tabla 2 Requerimientos funcionales

REQUERIMIENTOS NO FUNCIONALES

CODIGO	REQUERIMIENTO
RQNF-001	El sistema debe estar diseñado para ser escalable, permitiendo en el futuro una arquitectura cliente-servidor que soporte múltiples conexiones a la base de datos desde diferentes dispositivos.
RQNF-002	La base de datos debe estar preparada para admitir conexiones concurrentes desde múltiples clientes cuando la infraestructura lo permita.
RQNF-003	El sistema debe ser compatible con Windows 10 y Windows 11.
RQNF-004	La interfaz gráfica y las funcionalidades deben estar optimizadas para funcionar correctamente en estas versiones del sistema operativo.
RQNF-005	El sistema debe ser desarrollado en Java, garantizando portabilidad y flexibilidad en futuras expansiones o cambios de plataforma.
RQNF-006	El sistema debe utilizar MySQL para gestionar la base de datos, asegurando integridad y confiabilidad en el almacenamiento de datos.
RQNF-007	MySQL debe ser configurado inicialmente para una instalación local, pero con la posibilidad de migrar a un entorno de red en un futuro, utilizando conexiones cliente-servidor.
RQNF-008	El sistema debe permitir la conexión con impresoras láser para la impresión de reportes, facturas, y otros documentos relevantes desde cualquier módulo del sistema.
RQNF-009	La configuración de las impresoras debe ser sencilla y compatible con impresoras estándar conectadas a la PC local.
RQNF-010	El sistema debe proporcionar una interfaz de usuario intuitiva y fácil de usar, diseñada para que el personal administrativo, operativo, y docente de la escuela pueda interactuar con el sistema sin dificultades.
RQNF-011	Debe permitir una curva de aprendizaje rápida para los usuarios no técnicos.
RQNF-012	El sistema debe ser capaz de manejar un volumen moderado de datos (hasta 1,000 reservas activas simultáneamente) sin afectar su rendimiento.
RQNF-013	Los tiempos de respuesta de las consultas a la base de datos y operaciones del sistema deben ser menores a 2 segundos en condiciones normales.
RQNF-014	El sistema debe garantizar la seguridad de los datos almacenados, incluyendo el uso de autenticación para acceder a áreas críticas del sistema.
RQNF-015	Las cuentas de administrador deben tener privilegios adicionales para realizar tareas como la eliminación de registros o la configuración del sistema.

RQNF-016	El sistema debe estar documentado adecuadamente para que futuras mejoras, actualizaciones o modificaciones puedan ser realizadas de manera eficiente por otros desarrolladores.
RQNF-017	Debe utilizar una estructura modular para facilitar el mantenimiento y futuras integraciones.
RQNF-018	El sistema debe estar diseñado para ser confiable, con mecanismos de recuperación ante fallos que aseguren que la información crítica no se pierda en caso de fallas técnicas.
RQNF-019	Debe realizar copias de seguridad periódicas de la base de datos para evitar la pérdida de datos en caso de corrupción o daños.

Tabla 3 Requerimientos no funcionales

SECCION 2: ANALISIS

CASOS DE USO Y ELECCION DE REQUERIMINETOS CANDIDATOS

DIAGRAMA DE CASOS DE USO

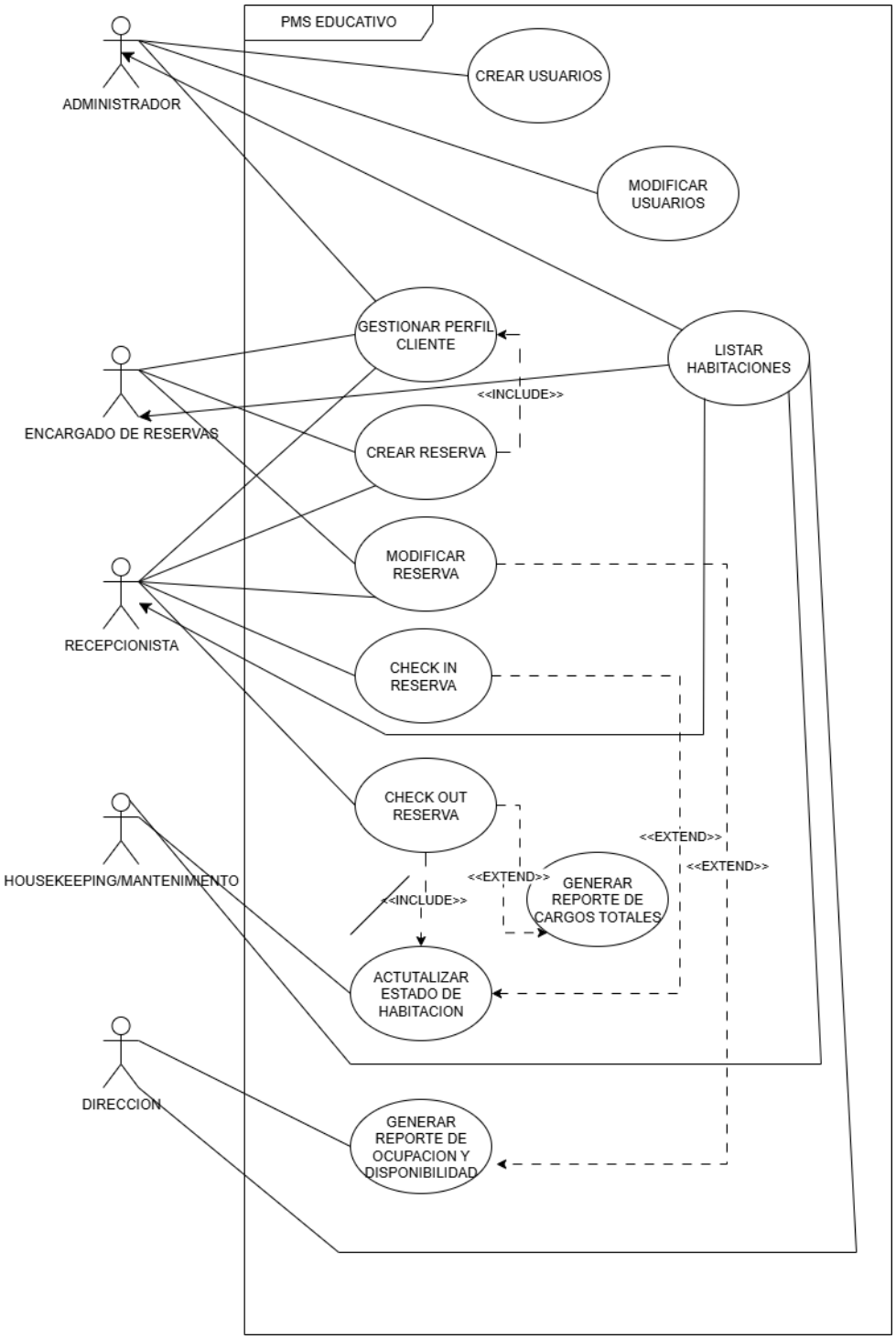

Ilustración 2 Diagrama de casos de uso

TRAZABILIDAD ENTRE REQUERIMIENTOS FUNCIONALES Y CASOS DE USOS

CODIGO	REQUERIMIENTO	CODIGO	CASOS DE USO
RQF-001	EL SISTEMA DEBE PERMITIR CREAR USUARIOS CON SUS ROLES Y PERMISOS	CU-001	CREAR USUARIOS
RQF-002	EL SISTEMA DEBE PERMITIR MODIFICAR LA INFORMACION DEL USUARIO, ROLES Y PERMISOS	CU-002	MODIFICAR USUARIOS
RQF-003	El sistema debe permitir listar todas las habitaciones registradas.	CU-003	LISTAR HABITACIONES
RQF-004	El sistema debe permitir asignar un estado a la habitación (clean, dirty, maintenance, inspected, pick up) que será modificado solo por el personal de housekeeping y mantenimiento.	CU-004	MODIFICAR HABITACION
RQF-005	El sistema debe mostrar en tiempo real el estado actual de las habitaciones en la interfaz de usuario.		
RQF-006	El sistema debe permitir crear un perfil de cliente con los siguientes datos: nombre completo, documento de identidad, correo electrónico, número de teléfono.	CU-005	GESTIONAR PERFIL CLIENTE
RQF-007	El sistema debe permitir listar todos los perfiles de clientes existentes.		
RQF-008	El sistema debe permitir modificar la información de un perfil de cliente existente.		
RQF-009	El sistema no debe permitir borrar perfiles de clientes, excepto al administrador del sistema.		
RQF-010	El sistema debe almacenar el historial de reservas del cliente en su perfil.		
RQF-011	El sistema debe permitir crear una reserva asociada a un perfil de cliente existente.	CU-006	CREAR RESERVA
RQF-012	El sistema debe permitir especificar las fechas de check-in y check-out en la creación de la reserva.		
RQF-013	El sistema debe permitir asignar una habitación disponible a la reserva en el momento de su creación.		
RQF-014	El sistema debe permitir modificar una reserva existente, incluyendo la habitación asignada y las fechas de estancia.	CU-007	MODIFICA RESERVA
RQF-015	El sistema debe permitir cancelar reservas.		
RQF-016	El sistema debe permitir agregar notas o "trace" a la reserva, en las que el agente de reservas pueda detallar particularidades.		
RQF-017	El sistema debe permitir que el personal de housekeeping actualice el estado de las habitaciones (clean, dirty, inspected).	CU-008	ACTUALIZAR ESTADO DE HABITACIONES
RQF-018	El sistema debe permitir que el personal de mantenimiento actualice el estado de las habitaciones a "maintenance" cuando sea necesario.		
RQF-019	El sistema debe permitir que el personal de housekeeping asigne habitaciones como "pick up" cuando estén listas para ser ocupadas.		
RQF-020	Solo el personal autorizado (housekeeping y mantenimiento) podrá modificar el estado de las habitaciones.		
RQF-021	El sistema debe permitir que los recepcionistas realicen el check-in de una reserva cuando la habitación asignada esté en estado "clean, inspected o pick up".	CU-009	CHECK IN RESERVA
RQF-022	El sistema debe permitir al recepcionista ver la tarifa asignada a la reserva al momento del check-in.		
RQF-023	El sistema debe permitir agregar cargos extras a la reserva durante el check-in o a lo largo de la estancia del cliente.		
RQF-024	El sistema debe permitir que los recepcionistas realicen el check-out de una reserva activa.	CU-010	CHECK OUT RESERVA
RQF-025	Al realizar el check-out, el sistema debe cambiar automáticamente el estado de la habitación a "dirty".		
RQF-026	El sistema debe generar un reporte detallado al check-out con el resumen de cargos totales, incluyendo la tarifa y cualquier cargo extra.		
RQF-027	El sistema debe permitir generar un reporte detallado de cada reserva, que incluya Fechas de la reserva (check-in, check-out). Cliente asociado a la reserva. Habitación asignada. Tarifa de la habitación. Cargos extras agregados durante la estancia. Total de la reserva.		
RQF-028	El sistema debe permitir a los recepcionistas imprimir este reporte al realizar el check-out.		
RQF-029	El reporte debe estar disponible para cualquier usuario autorizado que lo solicite.		
RQF-030	El sistema debe permitir generar un reporte que muestre la ocupación esperada en una fecha dada.	CU-011	GENERAR REPORTE DE OCUPACION Y DISPONIBILIDAD
RQF-031	El reporte debe ser accesible para todos los usuarios con permisos adecuados (reservas, recepción, dirección).		
RQF-032	El reporte debe poder ser exportado en formato PDF o impreso directamente desde el sistema.		

Ilustración 3 Tabla de trazabilidad

ELECCION DE REQUERIMIENTOS CANDIDATOS

Seleccionar los requerimientos candidatos es una parte crucial del proceso de desarrollo, especialmente en enfoques iterativos e incrementales como el Proceso Unificado de Desarrollo (PUD). Esta selección implica identificar cuáles de los requerimientos funcionales serán abordados en las primeras iteraciones del desarrollo del sistema, tomando en cuenta las prioridades del negocio, la complejidad técnica y la viabilidad.

Para la primera iteración en la que se desarrollara el MVP se seleccionaron los siguientes requerimientos funcionales:

CODIGO	REQUERIMIENTO
RQF-001	EL SISTEMA DEBE PERMITIR CREAR USUARIOS CON SUS ROLES Y PERMISOS
RQF-002	EL SISTEMA DEBE PERMITIR MODIFICAR LA INFORMACION DEL USUARIO, ROLES Y PERMISOS
RQF-003	El sistema debe permitir listar todas las habitaciones registradas.
RQF-004	El sistema debe permitir asignar un estado a la habitación (clean, dirty, maintenance, inspected, pick up) que será modificado solo por el personal de housekeeping y mantenimiento.
RQF-005	El sistema debe mostrar en tiempo real el estado actual de las habitaciones en la interfaz de usuario.
RQF-006	El sistema debe permitir crear un perfil de cliente con los siguientes datos: nombre completo, documento de identidad, correo electrónico, número de teléfono.
RQF-007	El sistema debe permitir listar todos los perfiles de clientes existentes.
RQF-008	El sistema debe permitir modificar la información de un perfil de cliente existente.
RQF-009	El sistema no debe permitir borrar perfiles de clientes, excepto al administrador del sistema.
RQF-010	El sistema debe almacenar el historial de reservas del cliente en su perfil.
RQF-011	El sistema debe permitir crear una reserva asociada a un perfil de cliente existente.
RQF-012	El sistema debe permitir especificar las fechas de check-in y check-out en la creación de la reserva.
RQF-013	El sistema debe permitir asignar una habitación disponible a la reserva en el momento de su creación.
RQF-014	El sistema debe permitir modificar una reserva existente, incluyendo la habitación asignada y las fechas de estancia.
RQF-015	El sistema debe permitir cancelar reservas.
RQF-016	El sistema debe permitir agregar notas o "trace" a la reserva, en las que el agente de reservas pueda detallar particularidades.

Tabla 4 Requerimientos Candidatos

RQF-001	El sistema debe permitir crear habitaciones con sus características (número, tipo de habitación: twin, king, superior, suite).
RQF-002	El sistema debe permitir listar todas las habitaciones registradas.
RQF-003	El sistema debe permitir modificar las características de una habitación existente (número, tipo).

RQF-004	El sistema debe permitir eliminar habitaciones únicamente si no están asignadas a una reserva activa.
RQF-005	El sistema debe permitir asignar un estado a la habitación (clean, dirty, maintenance, inspected, pick up) que será modificado solo por el personal de housekeeping y mantenimiento.
RQF-006	El sistema debe mostrar en tiempo real el estado actual de las habitaciones en la interfaz de usuario.
RQF-007	El sistema debe permitir crear un perfil de cliente con los siguientes datos: nombre completo, documento de identidad, correo electrónico, número de teléfono.
RQF-008	El sistema debe permitir listar todos los perfiles de clientes existentes.
RQF-009	El sistema debe permitir modificar la información de un perfil de cliente existente.
RQF-010	El sistema no debe permitir borrar perfiles de clientes, excepto al administrador del sistema.
RQF-011	El sistema debe almacenar el historial de reservas del cliente en su perfil.
RQF-012	El sistema debe permitir crear una reserva asociada a un perfil de cliente existente.
RQF-013	El sistema debe permitir especificar las fechas de check-in y check-out en la creación de la reserva.
RQF-014	El sistema debe permitir asignar una habitación disponible a la reserva en el momento de su creación.
RQF-015	El sistema debe permitir modificar una reserva existente, incluyendo la habitación asignada y las fechas de estancia.
RQF-016	El sistema debe permitir cancelar reservas.
RQF-017	El sistema debe permitir agregar notas o "trace" a la reserva, en las que el agente de reservas pueda detallar particularidades.

IDENTIFICACION DE ACTORES

Administrador del Sistema: Responsable de la creación y modificación de usuarios, además tiene permiso para listar habitaciones y listar clientes

Encargado de Reservas: Responsable de la creación y modificación de perfiles de clientes y reservas.

Recepcionista: Encargado de realizar el check-in, check-out, y generar reportes de cargos y ocupación.

Personal de Housekeeping/Mantenimiento: Encargado de actualizar el estado de las habitaciones (clean, dirty, maintenance, etc.).

Dirección: Responsable de generar reportes de ocupación y disponibilidad.

DIAGRAMA DE SECUENCIA

Se realizarán diagramas de secuencia para los casos de uso seleccionados, basados en los requerimientos funcionales candidatos para esta iteración del MVP. Los casos de uso son: CU-001 (Crear Usuarios), CU-002 (Modificar Usuarios), CU-003 (Listar Habitaciones), CU-004 (Modificar Habitación), CU-005 (Gestionar Perfil Cliente), CU-006 (Crear Reserva) y CU-007 (Modificar Reserva).

CU-001 Crear Usuarios

Actor: Administrador

Lifelines: Administrador, Sistema PMS, Base de Datos MySQL

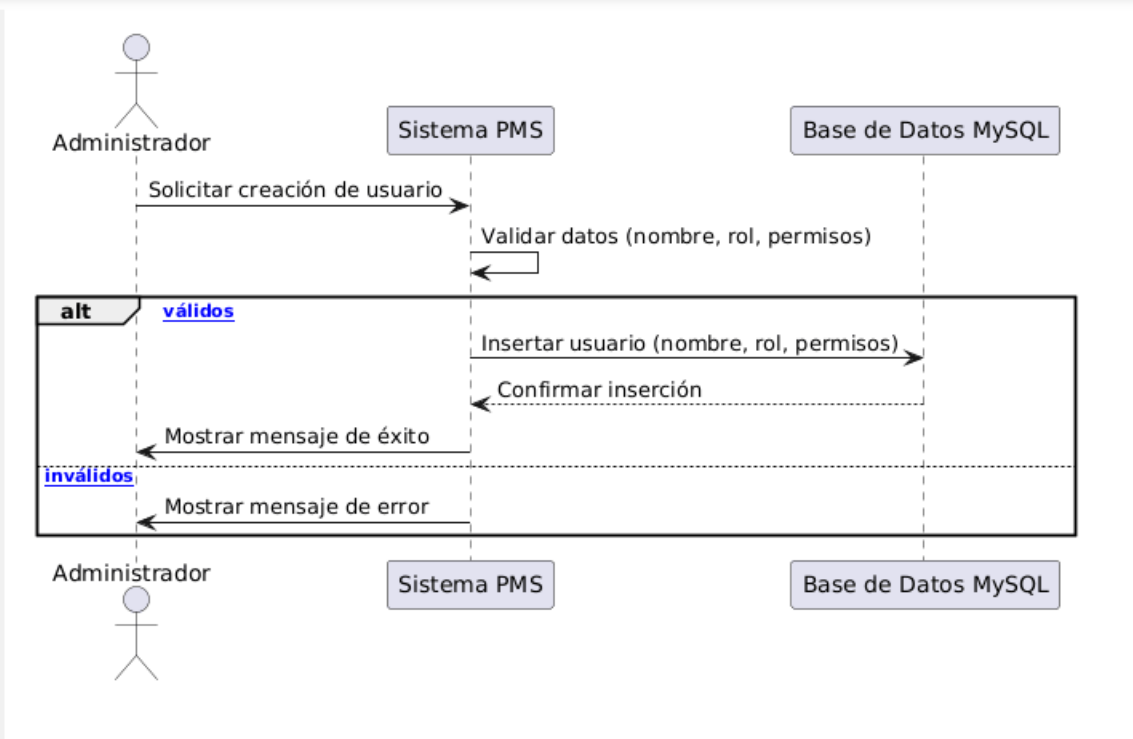

Ilustración 4 Diagrama de secuencia CU-001

Notas: La validación incluye verificar que el rol (e.g., recepcionista, housekeeping) sea válido.

CU-002 Modificar Usuarios

Actor: Administrador

Lifelines: Administrador, Sistema PMS, Base de Datos MySQL

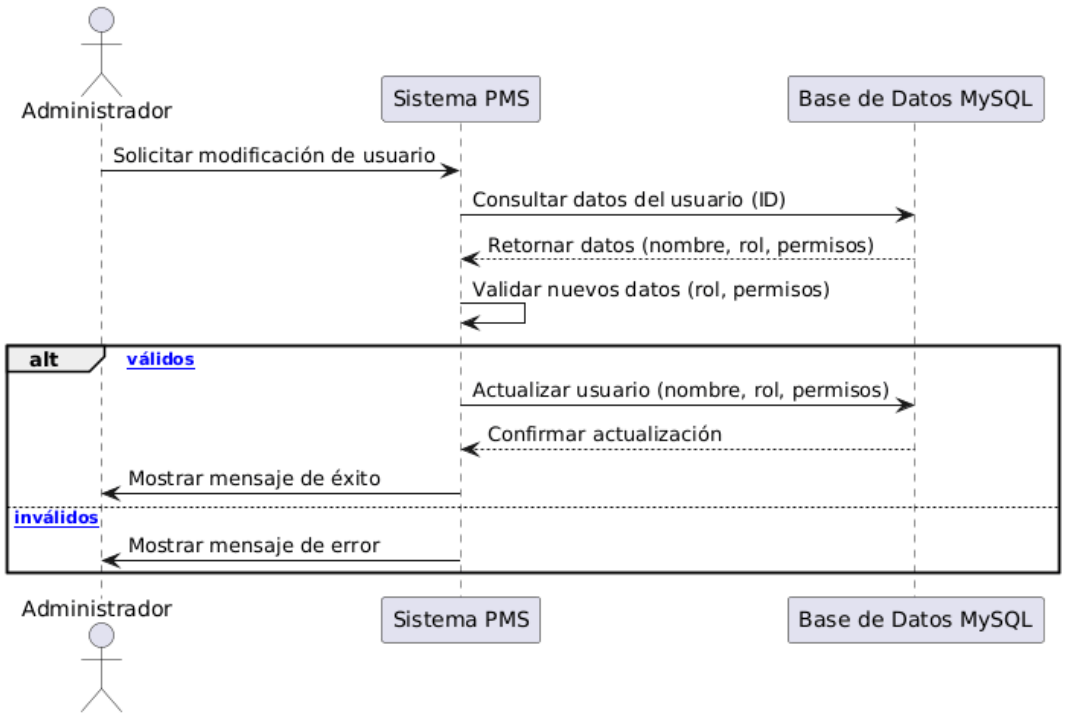

Ilustración 5 Diagrama de secuencia CU-002

CU-003 Listar Habitaciones

Actor: Administrador, Encargado de Reservas, Recepcionista

Lifelines: Actor, Sistema PMS, Base de Datos MySQL

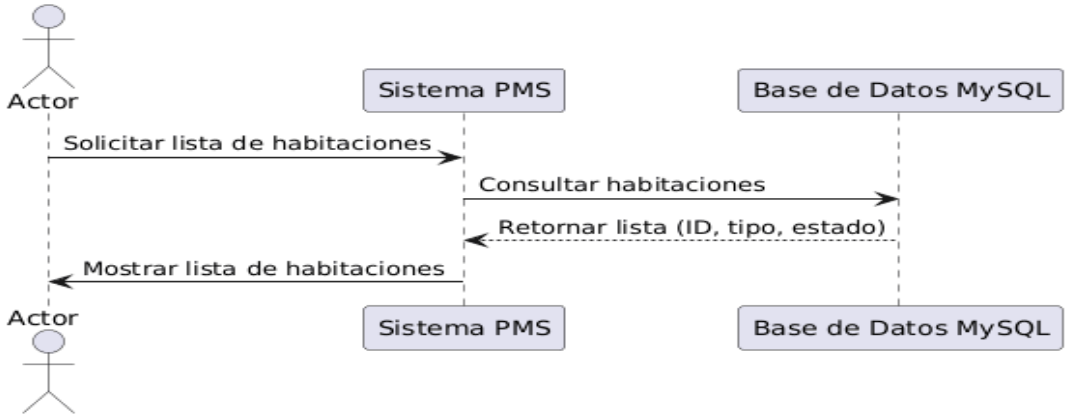

Ilustración 6 Diagrama de secuencia CU-003

Notas: Dado que las habitaciones son predefinidas, la consulta devuelve un conjunto estático. No se permite modificación de datos aquí.

CU-004 Modificar habitación

Actor: Personal de Housekeeping/Mantenimiento

Lifelines: Personal, Sistema PMS, Base de Datos MySQL

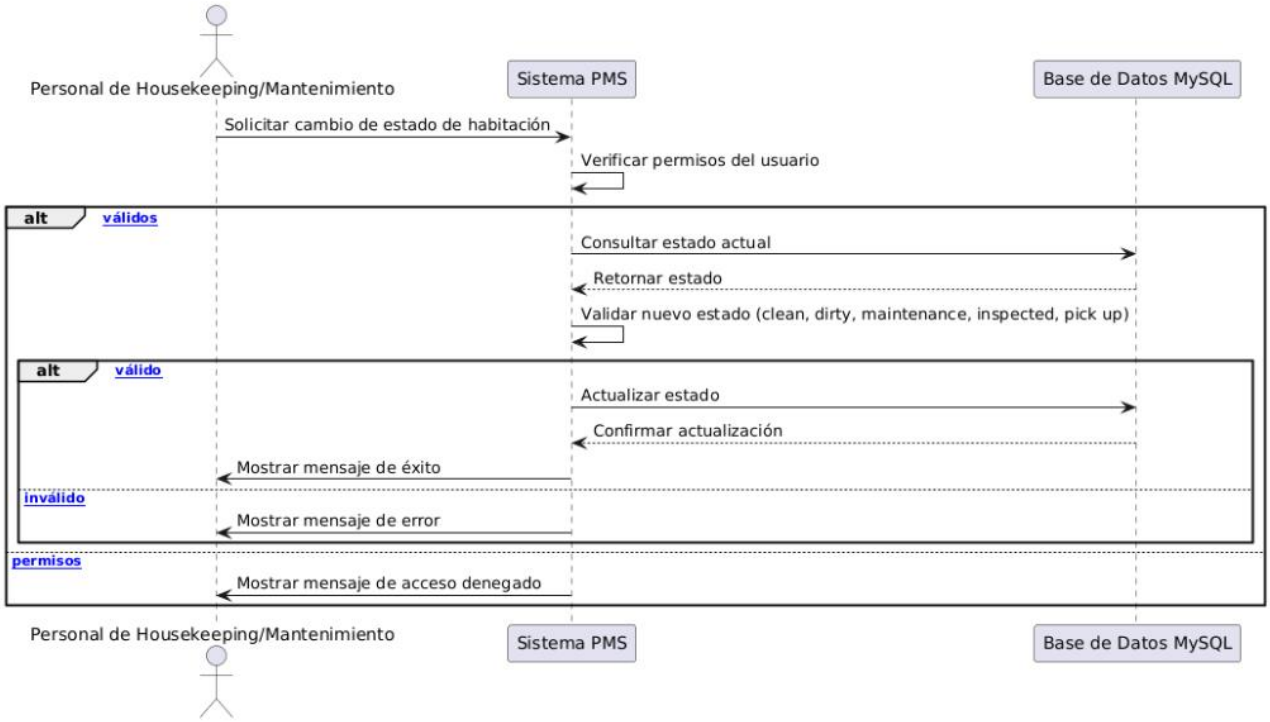

Ilustración 7 Diagrama de secuencia CU-004

Notas: La restricción RQF-020 asegura que solo housekeeping/mantenimiento pueda modificar estados.

CU-005 Gestionar Perfil cliente

Actor: Encargado de Reservas

Lifelines: Encargado, Sistema PMS, Base de Datos MySQL

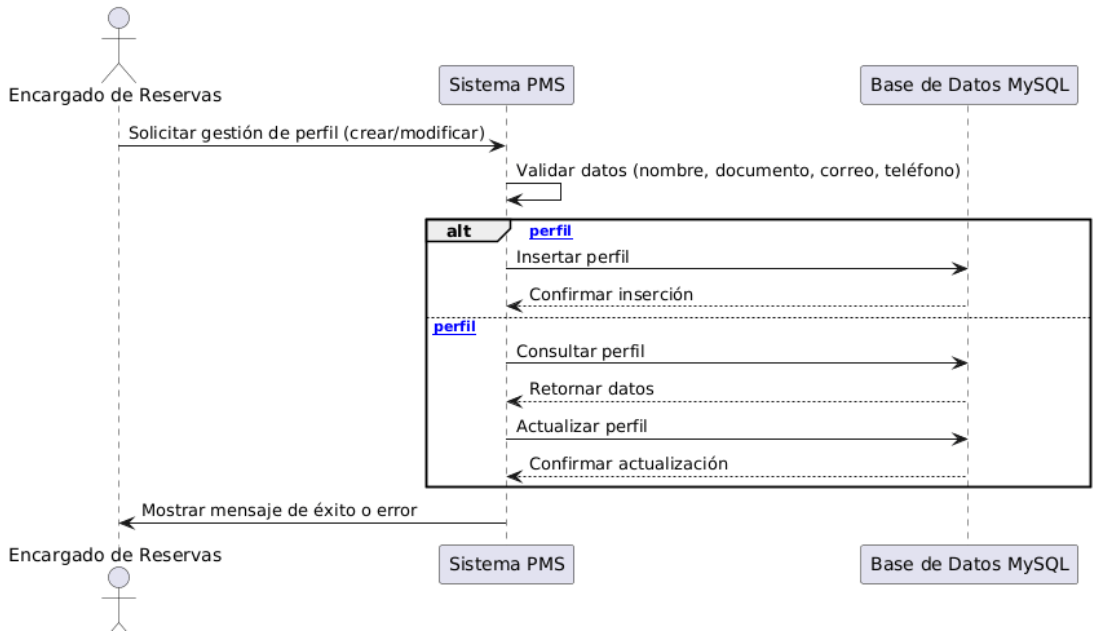

Ilustración 8 Diagrama de secuencia CU-005

Notas: RQF-009 (no borrar) se implementara como restricción en la interfaz;

CU-006 Crear Reserva

Actor: Encargado de Reservas

Lifelines: Encargado, Sistema PMS, Base de Datos MySQL

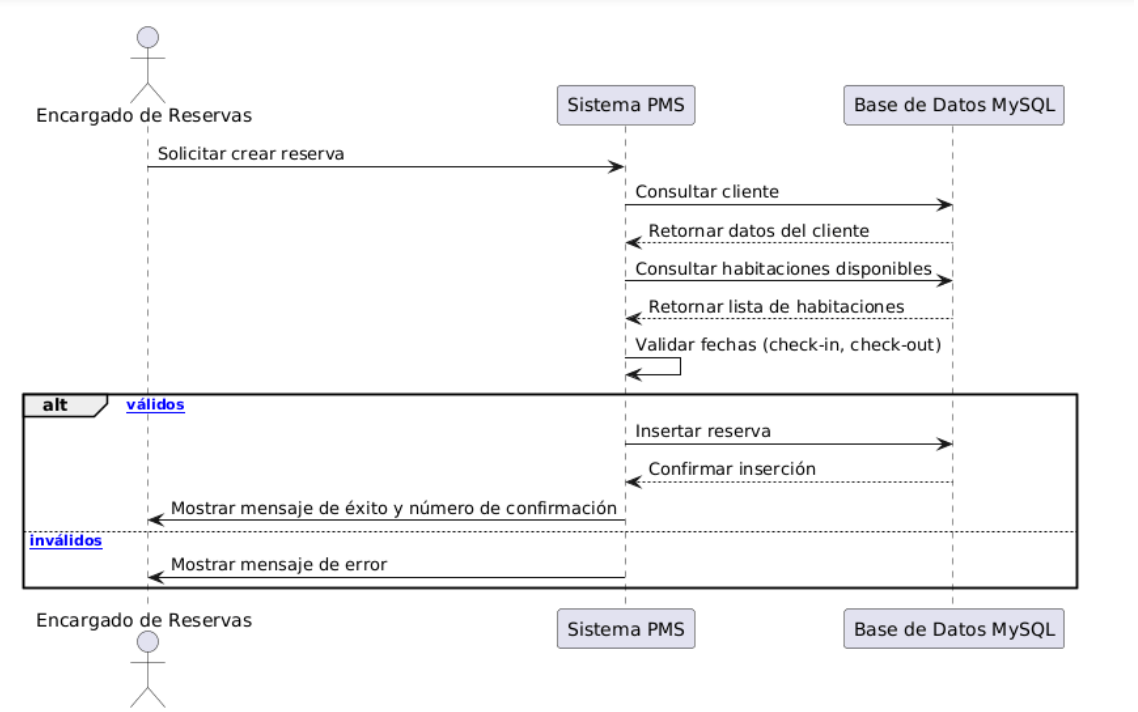

Ilustración 9 Diagrama de secuencia CU-006

Notas: RQF-012 y RQF-013 se implementaran en la validación; las habitaciones predefinidas simplifican la consulta.

CU-007 Modificar Reserva

Actor: Encargado de Reservas

Lifelines: Encargado, Sistema PMS, Base de Datos MySQL

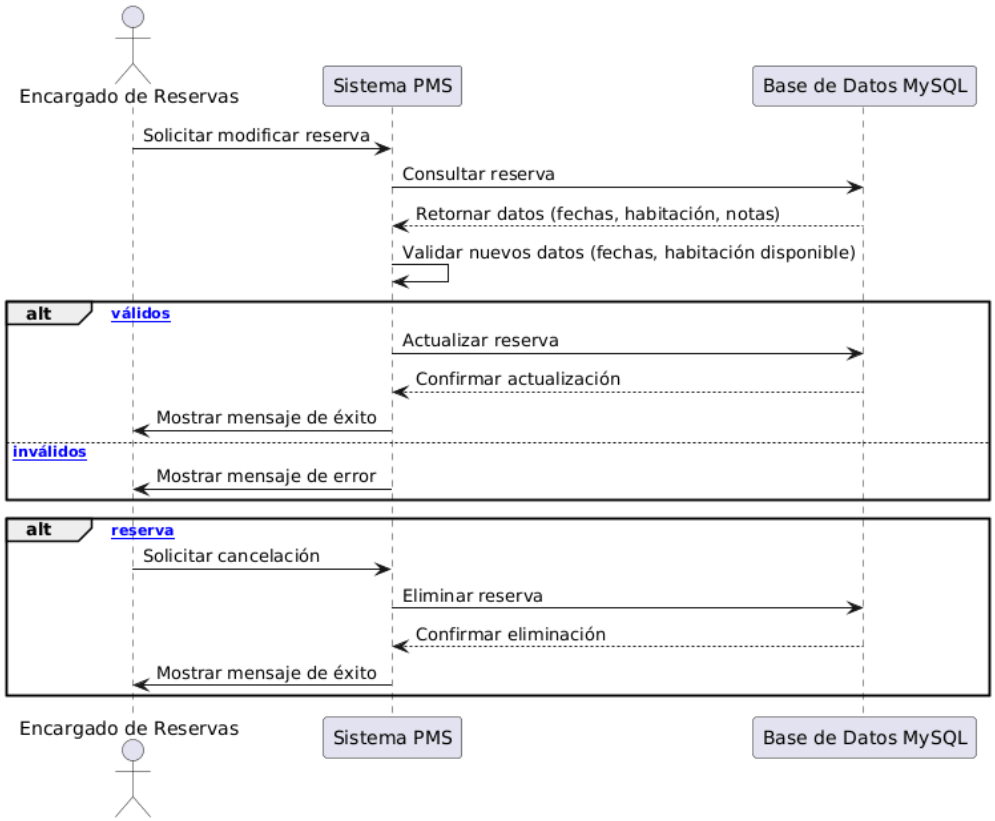

Ilustración 10 Diagrama de secuencia CU-007

Notas: RQF-014 (modificación), RQF-015 (cancelación) y RQF-016 (notas) se manejan en este flujo.

DIAGRAMA DE CLASES DEL ANALISIS

Se realizara el diagrama de clases del análisis teniendo en cuenta las clases definidas en UML de entidad, control e interface para los Requerimientos funcionales candidatos que corresponden a los CU-001 al CU-007 designados para esta iteraccion del MVP.

Adjunto enlace para visualizar optimizda de la imagen: Enlace

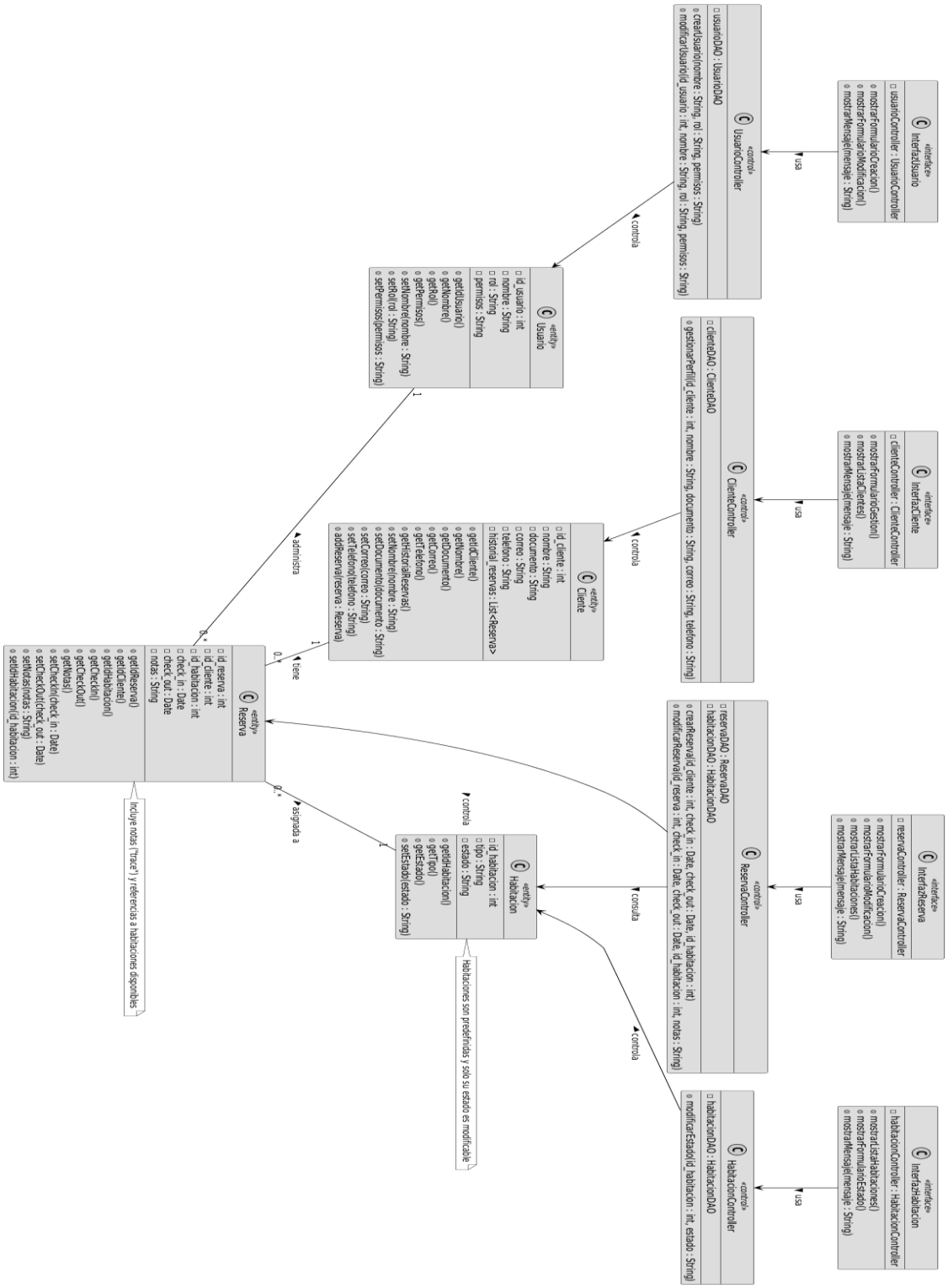

Ilustración 11 Diagrama de clases del analisis

Clases de Entidad

Usuario: Representa a los actores del sistema (Administrador, Encargado de Reservas, etc.) con atributos como id_usuario, nombre, rol y permisos. Relacionada con Reserva para rastrear administraciones.

Cliente: Almacena datos de clientes (RQF-006) y su historial de reservas (RQF-010), con atributos como id_cliente, nombre, documento, correo, telefono y historial_reservas.

Habitacion: Modela habitaciones predefinidas, con id_habitacion, tipo (twin, king, etc.) y estado (clean, dirty, etc.), modificable solo por housekeeping (RQF-004).

Reserva: Representa reservas (RQF-011 a RQF-016), con id_reserva, id_cliente, id_habitacion, check_in, check_out y notas, vinculando clientes y habitaciones.

Clases de Control:

UsuarioController: Coordina la creación y modificación de usuarios (CU-001, CU-002), interactuando con un DAO (Data Access Object) para persistencia.

ClienteController: Gestiona perfiles de clientes (CU-005), validando y actualizando datos.

ReservaController: Maneja creación y modificación de reservas (CU-006, CU-007), consultando habitaciones disponibles.

HabitacionController: Controla la modificación de estados de habitaciones (CU-004), restringida a housekeeping/mantenimiento.

Clases de Interfaz

InterfazUsuario: Proporciona la UI para crear y modificar usuarios (CU-001, CU-002), interactuando con UsuarioController.

InterfazCliente: Ofrece la UI para gestionar perfiles (CU-005), mostrando listas y formularios.

InterfazReserva: Facilita la creación y modificación de reservas (CU-006, CU-007), mostrando habitaciones disponibles.

InterfazHabitacion: Permite listar y actualizar estados de habitaciones (CU-003, CU-004).

Relaciones:

Asociaciones: Usuario administra Reserva, Cliente tiene Reserva, Habitacion es asignada a Reserva.

Dependencias: Controladores dependen de entidades para operaciones CRUD, y las interfaces dependen de controladores para interacción con actores.

Multiplicidades: 1 a 0..* reflejan que un cliente puede tener múltiples reservas, y una habitación puede estar asignada a varias reservas a lo largo del tiempo.

Notas y Restricciones

Las habitaciones son predefinidas (nota en Habitacion)

Reserva incluye soporte para notas ("trace", RQF-016).

La lógica de validación (e.g., permisos, estados válidos) se delega a los controladores, reflejando los diagramas de secuencia.

DIAGRAMA DE PAQUETES DEL ANALISIS

A continuación, se describen los paquetes de análisis, considerando los diagramas de secuencia, el diagrama de clases y los casos de uso del MVP (CU-001 a CU-007), definimos los paquetes de análisis para organizar los artefactos del modelo de análisis. Estos paquetes facilitan la gestión de clases de análisis, realizaciones de casos de uso y otros elementos relacionados, alineándose con las fases del PUD. La estructura se organiza en piezas manejables que reflejan las funcionalidades principales y los actores involucrados (Administrador, Encargado de Reservas, Personal de Housekeeping/Mantenimiento).

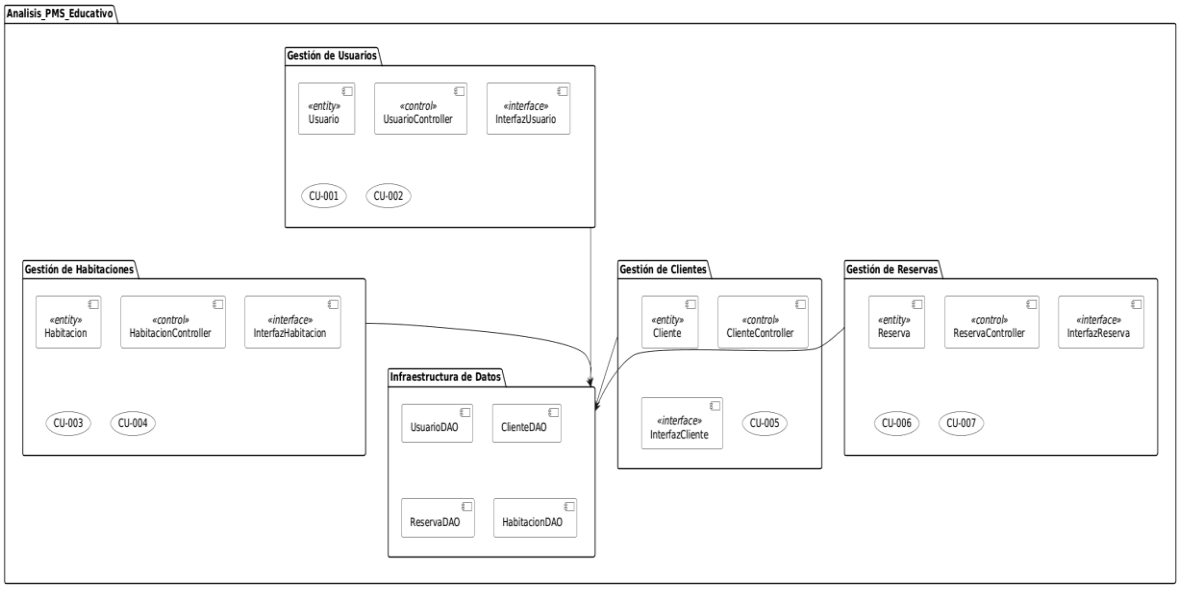

Ilustración 12 Diagrama de paquetes del análisis

Paquete: Gestión de Usuarios

Descripción: Contiene artefactos relacionados con la administración de usuarios, incluyendo la creación y modificación de perfiles de usuario con roles y permisos.

Contenido:

Clases de Análisis: Usuario (entidad), UsuarioController (control), InterfazUsuario (interfaz).

Realizaciones de Casos de Uso: CU-001 (Crear Usuarios), CU-002 (Modificar Usuarios).

Artefactos Adicionales: Diagrama de secuencia para CU-001 y CU-002, especificaciones de requerimientos funcionales (RQF-001, RQF-002).

Este paquete agrupa la lógica de autenticación y control de acceso, esencial para el Administrador, y facilita la trazabilidad de las funcionalidades iniciales del MVP.

Paquete: Gestión de Clientes

Descripción: Incluye artefactos asociados con la gestión de perfiles de clientes, como creación, modificación y mantenimiento del historial de reservas.

Contenido:

Clases de Análisis: Cliente (entidad), ClienteController (control), InterfazCliente (interfaz).

Realizaciones de Casos de Uso: CU-005 (Gestionar Perfil Cliente).

Artefactos Adicionales: Diagrama de secuencia para CU-005, especificaciones de requerimientos funcionales (RQF-006 a RQF-010).

Este paquete organiza la gestión de datos de clientes, una responsabilidad del Encargado de Reservas, y soporta la integración con el módulo de reservas.

Paquete: Gestión de Reservas

Descripción: Agrupa artefactos relacionados con la creación y modificación de reservas, incluyendo la asignación de habitaciones predefinidas y notas.

Contenido:

Clases de Análisis: Reserva (entidad), ReservaController (control), InterfazReserva (interfaz).

Realizaciones de Casos de Uso: CU-006 (Crear Reserva), CU-007 (Modificar Reserva).

Artefactos Adicionales: Diagramas de secuencia para CU-006 y CU-007, especificaciones de requerimientos funcionales (RQF-011 a RQF-016).

Este paquete centraliza las operaciones de reserva, clave para el Encargado de Reservas y Recepcionista, y refleja la interacción con habitaciones predefinidas.

Paquete: Gestión de Habitaciones

Descripción: Contiene artefactos para la visualización y modificación de estados de habitaciones predefinidas, limitada a housekeeping/mantenimiento.

Contenido:

Clases de Análisis: Habitacion (entidad), HabitacionController (control), InterfazHabitacion (interfaz).

Realizaciones de Casos de Uso: CU-003 (Listar Habitaciones), CU-004 (Modificar Habitación).

Artefactos Adicionales: Diagramas de secuencia para CU-003 y CU-004, especificaciones de requerimientos funcionales (RQF-003 a RQF-005).

Este paquete organiza la gestión estática de habitaciones, alineada con las restricciones del TP2, y soporta simulaciones educativas.

Paquete: Infraestructura de Datos

Descripción: Agrupa artefactos relacionados con la persistencia de datos y la interacción con la base de datos MySQL, que soporta todos los módulos.

Contenido:

Clases de Análisis: UsuarioDAO, ClienteDAO, ReservaDAO, HabitacionDAO (entidades de acceso a datos, implícitas en controladores).

Artefactos Adicionales: Especificaciones de conexiones JDBC, modelo conceptual de datos (preliminar para el DER).

Este paquete proporciona una capa de abstracción para la base de datos local, esencial para la implementación y pruebas del prototipo.

SECCION 3: DISEÑO

DIAGRAMA CLASES DE DISEÑO

El diagrama de clases de diseño es una abstracción que permite una traducción directa a la implementación en Java, el lenguaje de programación elegido para el proyecto. Este diagrama incorpora atributos, operaciones, parámetros y tipos definidos con visibilidad explícita (public, protected, private), alineándose con las convenciones de Java. Las clases de diseño se basan en las clases de análisis (entidad, control, interfaz) y los casos de uso del MVP (CU-001 a CU-007).

Adjunto enlace para visualización optimizada: Enlace

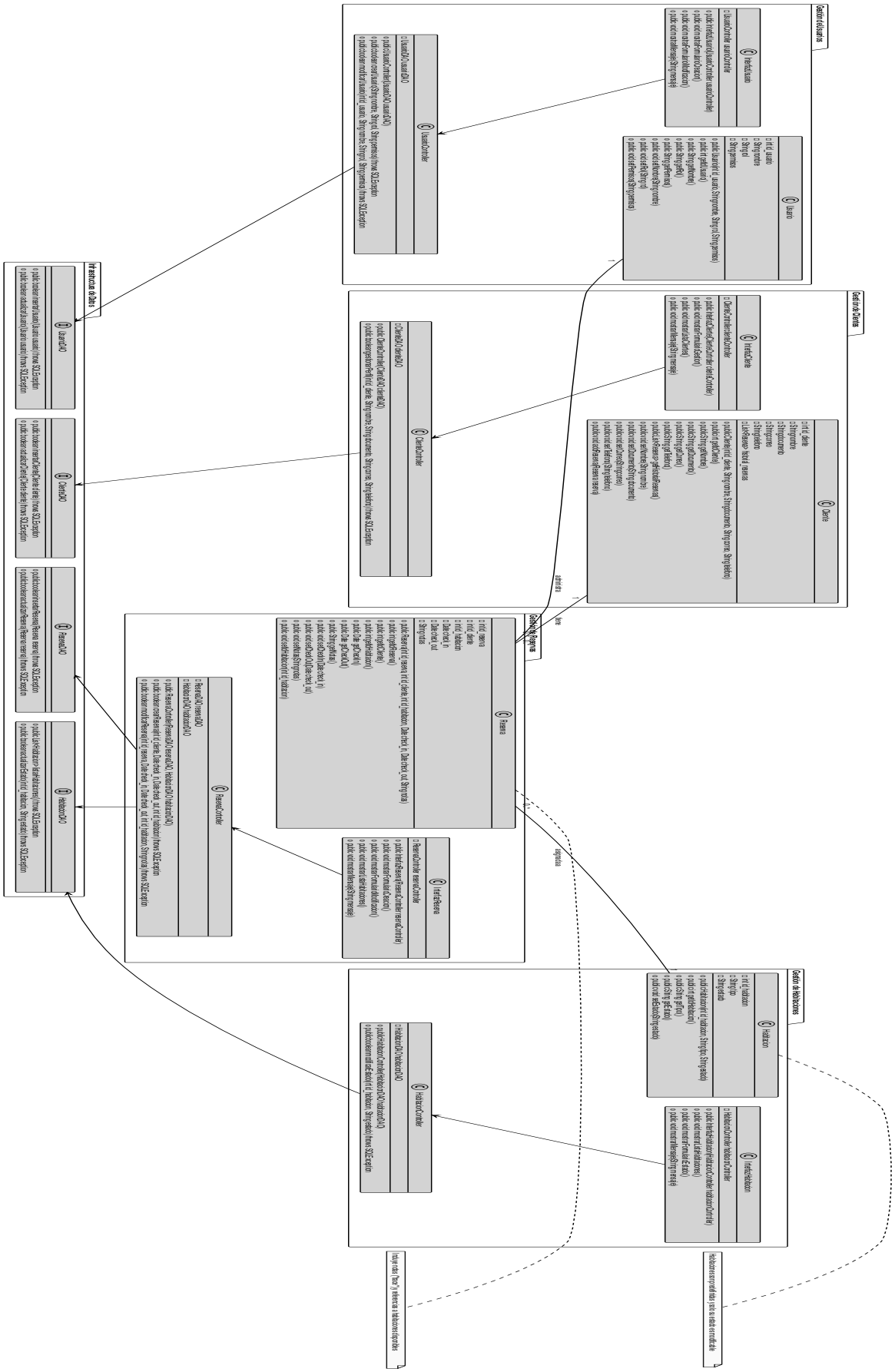

Ilustración 13 Diagrama de clases del diseño

DIAGRAMAS DE SECUENCIA

Con base en la etapa de análisis y los casos de uso seleccionados para el MVP (CU-001 a CU-007) del sistema PMS educativo, se elaborarán diagramas de secuencia que visualicen los mensajes entre los objetos involucrados, siguiendo el orden en que ocurren. Estos diagramas incorporarán el subsistema de diseño, que representa las capas o componentes principales del sistema (e.g., interfaz, control, persistencia) para reflejar la arquitectura del diseño. Dado que el lenguaje de implementación es Java y el sistema se basa en una instalación local con MySQL, los diagramas incluirán objetos correspondientes a las clases de diseño definidas previamente (entidades, controladores, interfaces) y el subsistema de persistencia (DAOs y base de datos).

A continuación, se describen los diagramas de secuencia para cada caso de uso (CU-001 a CU-007.

SUBSISTEMA DE DISEÑO

El subsistema de diseño se estructura en tres capas principales:

Capa de Interfaz: Representada por las clases InterfazUsuario, InterfazCliente, InterfazReserva, InterfazHabitacion, que interactúan con los actores.

Capa de Control: Incluye los controladores (UsuarioController, ClienteController, ReservaController, HabitacionController), que coordinan la lógica de negocio.

Capa de Persistencia: Compuesta por las interfaces DAO (UsuarioDAO, ClienteDAO, ReservaDAO, HabitacionDAO) y la base de datos MySQL, manejando el acceso a datos.

Este subsistema se reflejará en los diagramas como lifelines, mostrando la interacción entre capas.

Diagrama de Secuencia para CU-001: Crear Usuarios

Actor: Administrador

Lifelines: Administrador, InterfazUsuario, UsuarioController, UsuarioDAO, Base de Datos MySQL

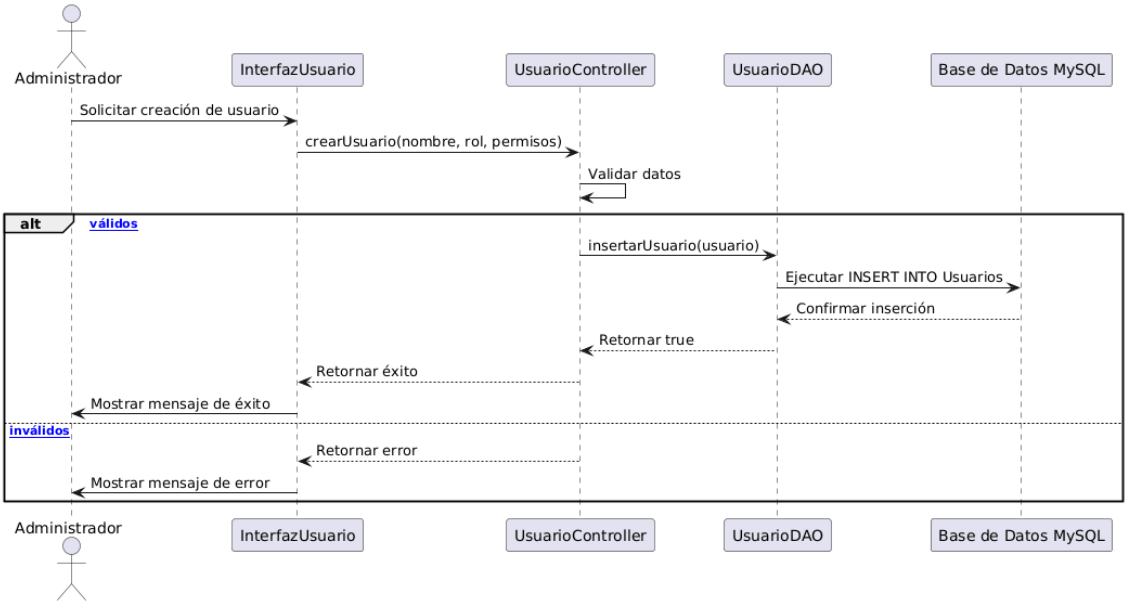

Ilustración 14 Diagrama de secuencia de diseño CU-001

Notas: El subsistema muestra la interacción entre la capa de interfaz (IU), control (UC) y persistencia (UD, DB). La validación se realiza en el controlador.

Diagrama de Secuencia para CU-002: Modificar Usuarios

Actor: Administrador

Lifelines: Administrador, InterfazUsuario, UsuarioController, UsuarioDAO, Base de Datos MySQL

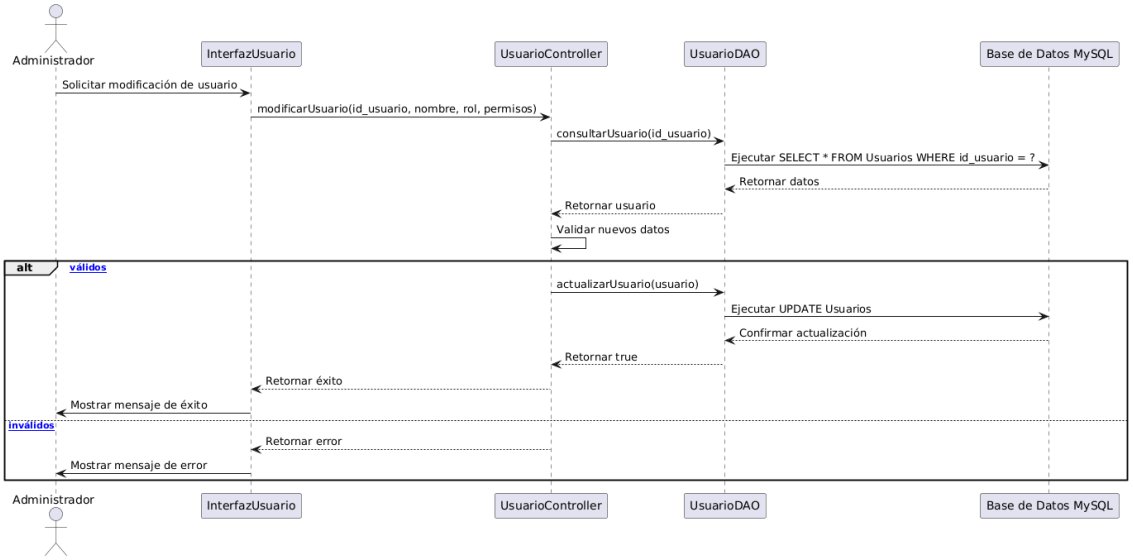

Ilustración 15 Diagrama de secuencia de diseño CU-002

Notas: El subsistema refleja la consulta inicial y la actualización en la base de datos.

Diagrama de Secuencia para CU-003: Listar Habitaciones

Actor: Administrador, Encargado de Reservas, Recepcionista

Lifelines: Actor, InterfazHabitacion, HabitacionController, HabitacionDAO, Base de Datos MySQL

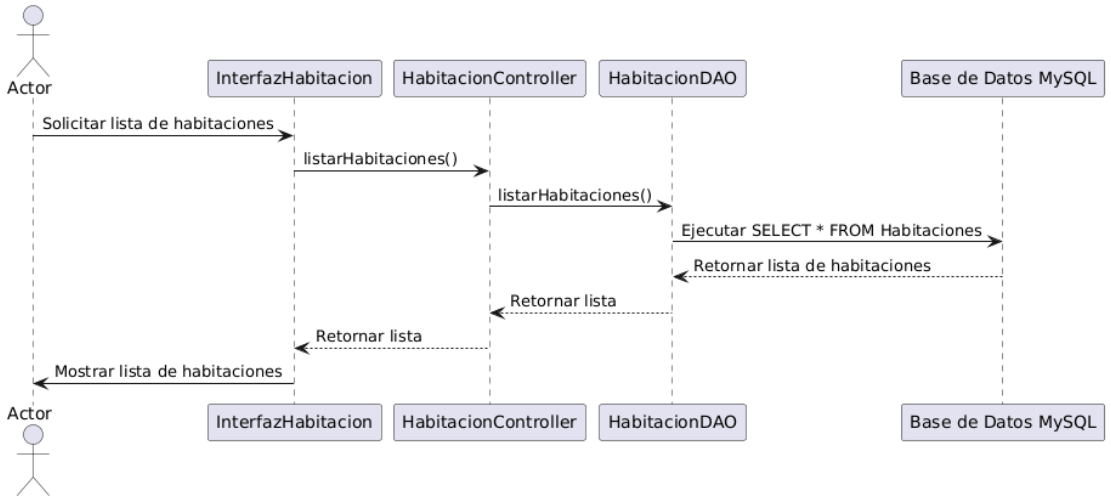

Ilustración 16 Diagrama de secuencia de diseño CU-003

Notas: El subsistema muestra una consulta simple, respetando que las habitaciones son predefinidas.

Diagrama de Secuencia para CU-004: Modificar Habitación

Actor: Personal de Housekeeping/Mantenimiento

Lifelines: Personal, InterfazHabitacion, HabitacionController, HabitacionDAO, Base de Datos MySQL

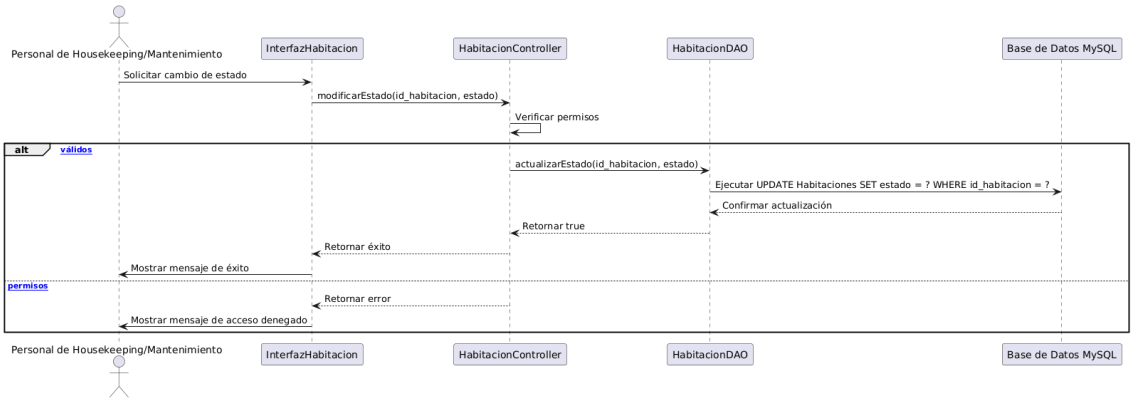

Ilustración 17 Diagrama de secuencia de diseño CU-004

Notas: El subsistema incluye validación de permisos en la capa de control.

Diagrama de Secuencia para CU-005: Gestionar Perfil Cliente

Actor: Encargado de Reservas

Lifelines: Encargado, InterfazCliente, ClienteController, ClienteDAO, Base de Datos MySQL

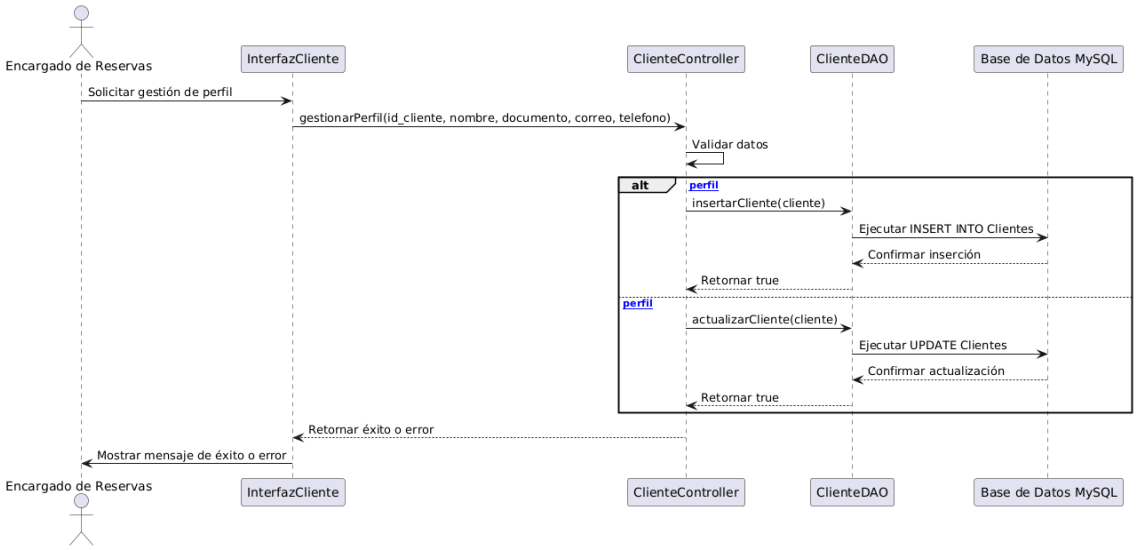

Ilustración 18Diagrama de secuencia de diseño CU-005

Notas: El subsistema maneja ambas operaciones (crear/modificar) con validación.

Diagrama de Secuencia para CU-006: Crear Reserva

Actor: Encargado de Reservas

Lifelines: Encargado, InterfazReserva, ReservaController, ReservaDAO, HabitacionDAO, Base de Datos MySQL

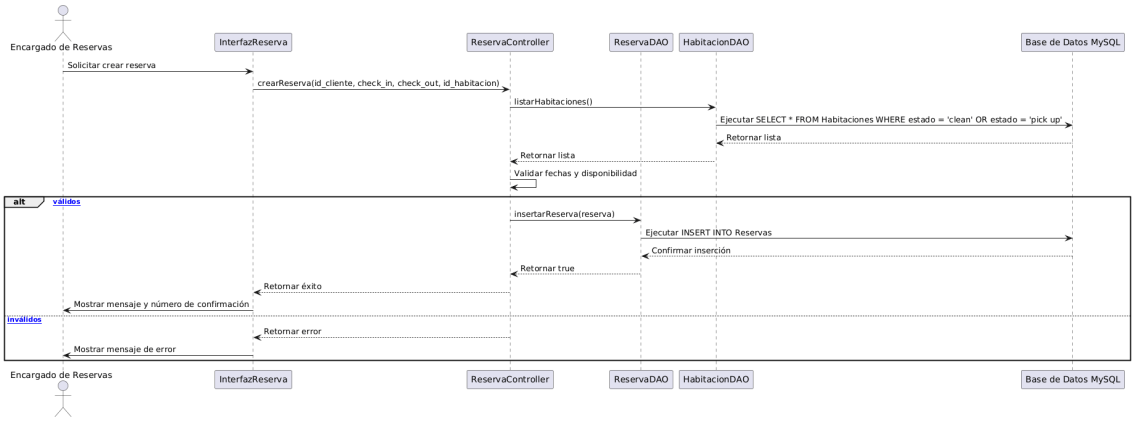

Ilustración 19 Diagrama de secuencia de diseño CU-006

Notas: El subsistema incluye consulta de habitaciones antes de la inserción.

Diagrama de Secuencia para CU-007: Modificar Reserva

Actor: Encargado de Reservas

Lifelines: Encargado, InterfazReserva, ReservaController, ReservaDAO, HabitacionDAO, Base de Datos MySQL

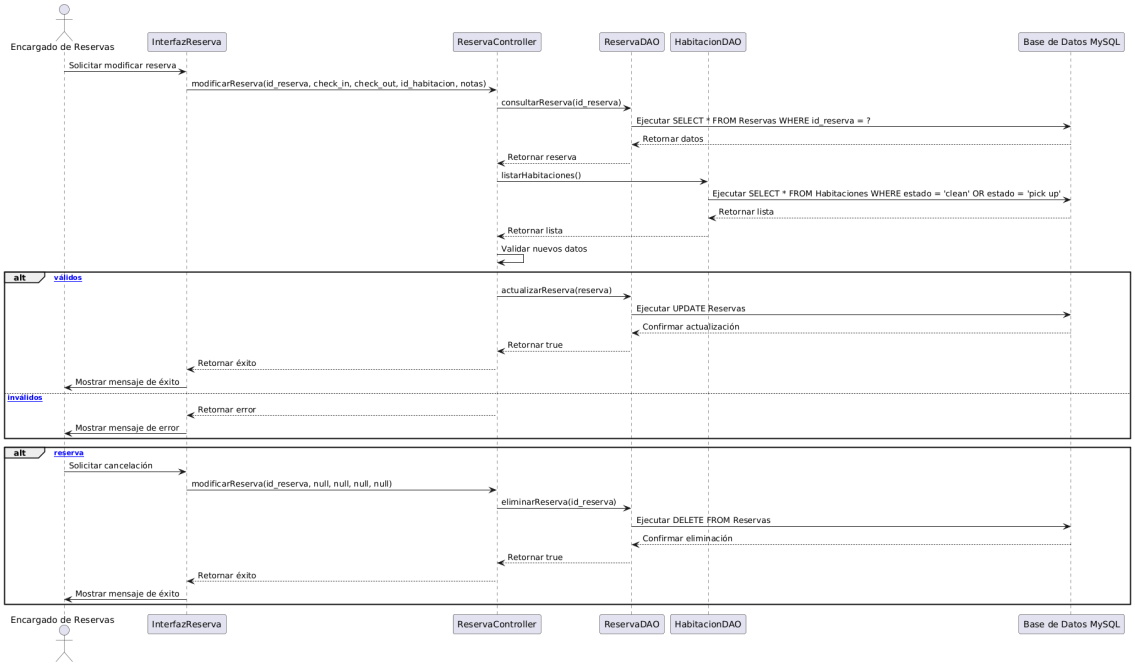

Ilustración 20 Diagrama de secuencia de diseño CU-007

Notas: El subsistema maneja modificación y cancelación, con validación de disponibilidad.

SECCION 4: IMPLEMENTACION

DESPLIEGUE

Con base en la etapa de diseño y los requisitos funcionales candidatos, se elaborará un diagrama de despliegue que represente la distribución de los componentes del sistema PMS educativo en los nodos físicos y cómo interactúan entre sí en una implementación real. Este diagrama será útil para entender cómo se desplegará el sistema en hardware físico y cómo se comunican sus componentes. El diseño refleja una arquitectura local sin conexión en red, utilizando Java, MySQL y XAMPP.

REQUISITOS CONSIDERADOS

Requisitos Funcionales Candidatos

CODIGO	REQUERIMIENTO	CODIGO	CASOS DE USO
RQF-001	EL SISTEMA DEBE PERMITIR CREAR USUARIOS CON SUS ROLES Y PERMISOS	CU-001	CREAR USUARIOS
RQF-002	EL SISTEMA DEBE PERMITIR MODIFICAR LA INFORMACION DEL USUARIO, ROLES Y PERMISOS	CU-002	MODIFICAR USUARIOS
RQF-003	El sistema debe permitir listar todas las habitaciones registradas.	CU-003	LISTAR HABITACIONES
RQF-004	El sistema debe permitir asignar un estado a la habitación (clean, dirty, maintenance, inspected, pick up) que será modificado solo por el personal de housekeeping y mantenimiento.	CU-004	MODIFICAR HABITACION
RQF-005	El sistema debe mostrar en tiempo real el estado actual de las habitaciones en la interfaz de usuario.		
RQF-006	El sistema debe permitir crear un perfil de cliente con los siguientes datos: nombre completo, documento de identidad, correo electrónico, número de teléfono.	CU-005	GESTIONAR PERFIL CLIENTE
RQF-007	El sistema debe permitir listar todos los perfiles de clientes existentes.		
RQF-008	El sistema debe permitir modificar la información de un perfil de cliente existente.		
RQF-009	El sistema no debe permitir borrar perfiles de clientes, excepto al administrador del sistema.		
RQF-010	El sistema debe almacenar el historial de reservas del cliente en su perfil.		
RQF-011	El sistema debe permitir crear una reserva asociada a un perfil de cliente existente.	CU-006	CREAR RESERVA

RQF-012	El sistema debe permitir especificar las fechas de check-in y check-out en la creación de la reserva.		
RQF-013	El sistema debe permitir asignar una habitación disponible a la reserva en el momento de su creación.		
RQF-014	El sistema debe permitir modificar una reserva existente, incluyendo la habitación asignada y las fechas de estancia.	CU-007	MODIFICA RESERVA
RQF-015	El sistema debe permitir cancelar reservas.		
RQF-016	El sistema debe permitir agregar notas o "trace" a la reserva, en las que el agente de reservas pueda detallar particularidades.		

Ilustración 21Tabla de Trazabilidad Requisitos Funcionales Candidatos - Casos de Uso

DIAGRAMA DE DESPLIEGUE

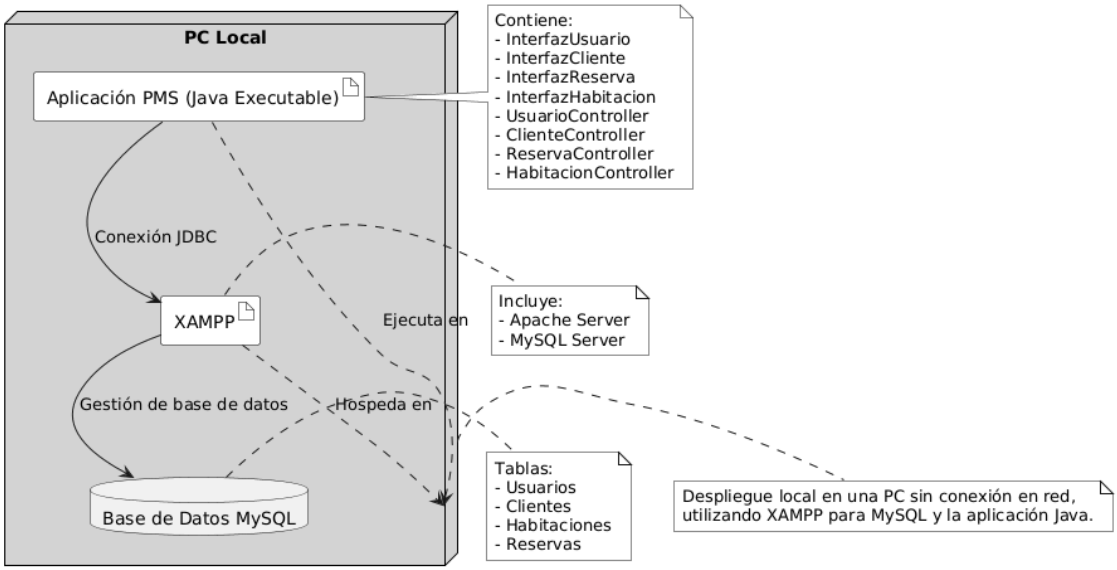

Ilustración 22 Diagrama de despliegue

NODOS FISICOS

PC Local: Representa el único nodo físico, una computadora local donde se desplegará el sistema. Esto refleja la restricción de instalación local sin conexión en red, adecuada para entornos educativos con recursos limitados (e.g., Escuela Vatel, Colegio Alfarcito).

COMPONENTES

Aplicación PMS (Java Executable):

Descripción: Archivo ejecutable Java que contiene las clases de interfaz (InterfazUsuario, InterfazCliente, InterfazReserva, InterfazHabitacion) y control (UsuarioController, ClienteController, ReservaController, HabitacionController) definidas en el diseño.

Interfaz: Proporciona servicios de interacción con los actores (Administrador, Encargado de Reservas, etc.).

Relación: Se ejecuta en la PC Local y se conecta a XAMPP via JDBC para acceder a la base de datos.

XAMPP:

Descripción: Entorno de desarrollo web de código abierto que incluye un servidor Apache y un servidor MySQL.

Interfaz: Ofrece servicios de hosting y gestión de la base de datos.

Relación: Hospeda la base de datos MySQL y facilita la conexión con la aplicación Java.

Base de Datos MySQL:

Descripción: Componente de persistencia con tablas (Usuarios, Clientes, Habitaciones, Reservas).

Interfaz: Proporciona servicios de almacenamiento y consulta de datos.

Relación: Gestionada por XAMPP y accedida por la aplicación via JDBC.

RELACIONES

Conexión JDBC: La aplicación PMS se conecta a XAMPP (específicamente al servidor MySQL) mediante JDBC, reflejando la interacción entre la capa de control y la persistencia.

Gestión de Base de Datos: XAMPP administra la base de datos MySQL, ejecutándose localmente en la PC.

Ejecución y Hospedaje: Ambos componentes (aplicación y XAMPP) se ejecutan/hospedan en la PC Local, representado por relaciones de dependencia.

PERSISTENCIA DE DATOS

DEFINICION DE BASE DE DATOS PARA PROTOTIPO

El desarrollo del sistema PMS educativo requiere la implementación de una base de datos relacional como componente central. De acuerdo con los requisitos no funcionales establecidos, el prototipo empleará MySQL, una solución que proporciona un desempeño sólido, adaptabilidad y capacidad de crecimiento. Dado que estamos trabajando en un prototipo, estas cualidades resultan esenciales para estructurar de manera eficiente el avance hacia la versión completa del proyecto.

La base de datos desempeña un rol clave al garantizar la persistencia de información crítica, como los datos de usuarios, detalles de clientes, estados de las habitaciones y registros de reservas, entre otros. Esta funcionalidad es indispensable para conservar un historial detallado de todas las operaciones y actividades llevadas a cabo en el sistema.

Además, la conservación de estos datos habilita la generación de informes que faciliten el análisis de la ocupación, la administración de reservas, el seguimiento del estado de las habitaciones y la toma de decisiones para mejorar la gestión del entorno educativo.

El prototipo se concibe como un modelo funcional que incorpora únicamente ciertas características del sistema final que abarca los requisitos funcionales candidatos representados por los casos de uso CU-001 al CU-007. Siguiendo el Proceso Unificado de Desarrollo, las clases identificadas en las etapas previas, especialmente aquellas de la etapa de diseño, sirven como base para definir las tablas que se crearán en la base de datos.

DIAGRAMA ENTIDAD-RELACION

Con base en los diagramas previamente generados (diagrama de clases de diseño, diagrama de despliegue, diagramas de secuencia de diseño y el subsistema de diseño), se elaborará un Diagrama Entidad-Relación (DER) para el prototipo del sistema PMS educativo. Este DER identificará las entidades, sus atributos, claves primarias, claves foráneas, relaciones y cardinalidades, utilizando como referencia las tablas y clases definidas en las etapas anteriores.

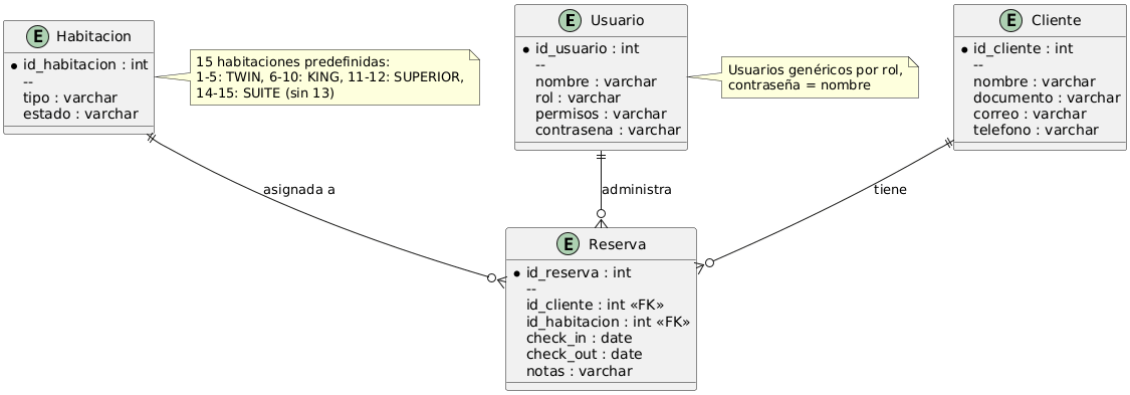

Ilustración 23Diagrama Entidad Relación

EXPLICACION DETALLADA DER

Entidades

Usuario: Incluye id_usuario como clave primaria y atributos derivados del diagrama de clases (nombre, rol, permisos, contrasena predefinida).

Cliente: Contiene id_cliente como clave primaria y atributos de perfil (nombre, documento, correo, telefono).

Habitación: Usa id_habitacion como clave primaria, con tipos y estados predefinidos (15 habitaciones según el prompt).

Reserva: Tiene id_reserva como clave primaria, y id_cliente e id_habitacion como claves foráneas, con atributos temporales y notas.

Atributos

Los atributos se derivan de las clases de diseño (e.g., Usuario, Cliente, Habitación, Reserva) y se ajustan a tipos MySQL (int, varchar, date).

Las claves primarias están marcadas con * y las foráneas con <<FK>>.

Relaciones y Cardinalidad

Administra: Relación opcional (0..*) desde Usuario a Reserva, reflejando que no todos los usuarios gestionan reservas, pero un usuario puede administrar muchas.

Tiene: Relación de 1 a 0..* entre Cliente y Reserva, indicando que un cliente puede tener varias reservas.

Asignada a: Relación de 1 a 0..* entre habitación y Reserva, permitiendo múltiples asignaciones históricas.

Las cardinalidades se basan en los diagramas de secuencia y clases, donde las reservas dependen de clientes y habitaciones.

Notas

La nota en habitación detalla las 14 habitaciones predefinidas (1-5 TWIN, 6-10 KING, 11-12 SUPERIOR, 14-15 SUITE, sin 13 debido a la costumbre en hotelería de evitar esa numeración para “buena suerte”)

CREACION DE TABLAS

TABLA HABITACIONES

Contiene 14 habitaciones predefinidas con tipos específicos (1-5 TWIN, 6-10 KING, 11-12 SUPERIOR, 14-15 SUITE, sin 13) y un estado inicial.

```sql
CREATE TABLE Habitaciones (
    id_habitacion INT PRIMARY KEY,
    tipo VARCHAR(20) NOT NULL,
    estado VARCHAR(20) NOT NULL DEFAULT 'clean'
);
```

Ilustración 24 Creación Tabla Habitaciones

Poblar con datos predefinidos tabla Habitaciones.

```sql
INSERT INTO Habitaciones (id_habitacion, tipo) VALUES
(1, 'TWIN'), (2, 'TWIN'), (3, 'TWIN'), (4, 'TWIN'), (5, 'TWIN'),
(6, 'KING'), (7, 'KING'), (8, 'KING'), (9, 'KING'), (10, 'KING'),
(11, 'SUPERIOR'), (12, 'SUPERIOR'),
(14, 'SUITE'), (15, 'SUITE');
```

Ilustración 25 Poblar Tabla Habitaciones

TABLA USUARIOS

Incluye usuarios genéricos por rol con nombres y contraseñas predefinidos.

```sql
CREATE TABLE Usuarios (
    id_usuario INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(50) NOT NULL,
    rol VARCHAR(50) NOT NULL,
    permisos VARCHAR(100) NOT NULL,
    contrasena VARCHAR(50) NOT NULL
);
```

Ilustración 26 Creacion Tabla Usuarios

Poblar con datos predefinidos

```sql
INSERT INTO Usuarios (nombre, rol, permisos, contrasena) VALUES
('Administrador', 'Administrador', 'gestion_usuarios,gestion_reservas', 'Administrador'),
('Reservas', 'Encargado de Reservas', 'gestion_clientes,gestion_reservas', 'Reservas'),
('Recepcionista', 'Recepcionista', 'gestion_reservas', 'Recepcionista'),
('Housekeeping', 'Personal de Housekeeping', 'modificar_habitaciones', 'Housekeeping'),
('Direccion', 'Direccion', 'ver_reportes', 'Direccion');
```

Ilustración 27Poblar Tabla Usuarios

TABLA CLIENTES

Almacena perfiles de clientes, inicialmente vacía para que se creen dinámicamente.

```sql
CREATE TABLE Clientes (
    id_cliente INT PRIMARY KEY AUTO_INCREMENT,
    nombre VARCHAR(100) NOT NULL,
    documento VARCHAR(20) NOT NULL,
    correo VARCHAR(100),
    telefono VARCHAR(20)
);
```

Ilustración 28 Creacion Tabla Clientes

TABLA RESERVAS

Registra reservas con referencias a clientes y habitaciones.

```sql
CREATE TABLE Reservas (
    id_reserva INT PRIMARY KEY AUTO_INCREMENT,
    id_cliente INT,
    id_habitacion INT,
    check_in DATE NOT NULL,
    check_out DATE NOT NULL,
    notas VARCHAR(255),
    FOREIGN KEY (id_cliente) REFERENCES Clientes(id_cliente),
    FOREIGN KEY (id_habitacion) REFERENCES Habitaciones(id_habitacion)
);
```

Ilustración 29 Creacion Tabla Reservas

Estructura de las Tablas

Habitaciones: id_habitacion es la clave primaria, tipo y estado son atributos obligatorios con valores predefinidos. El estado inicial es "clean" por defecto.

Usuarios: id_usuario es la clave primaria con auto-incremento, y los campos nombre, rol, permisos y contrasena son obligatorios, con datos predefinidos para cada rol.

Clientes: id_cliente es la clave primaria con auto-incremento, y los campos reflejan los atributos de perfil. Se deja vacía inicialmente.

Reservas: id_reserva es la clave primaria con auto-incremento, id_cliente e id_habitacion son claves foráneas que referencian a Clientes y Habitaciones, respectivamente. Se deja vacía inicialmente.

Claves Foráneas

En Reservas, id_cliente y id_habitacion establecen relaciones con Clientes y Habitaciones, respetando las cardinalidades del DER (1 a 0..*).

Datos Predefinidos

Habitaciones: Se insertan 14 registros (1-5 TWIN, 6-10 KING, 11-12 SUPERIOR, 14-15 SUITE, sin 13).

Usuarios: Se insertan 5 usuarios genéricos (Administrador, Encargado de Reservas, Recepcionista, Personal de Housekeeping, Dirección) con contraseñas iguales a sus nombres y permisos básicos.

Clientes y Reservas: Se dejan vacías para que los datos se generen durante el uso del prototipo.

INSERCION, CONSULTA Y BORRADO DE REGISTROS

Con base en las tablas creadas previamente para el prototipo del sistema PMS educativo en MySQL, realizaremos pruebas de inserción, consulta y borrado de registros. Se insertarán 5 registros en la tabla Clientes, se realizarán consultas y se eliminarán registros. Para la tabla Reservas, se insertarán 2 registros, se consultarán y se borrará un registro. Además, se incluirán consultas básicas para las tablas Usuarios y Habitaciones para verificar su contenido predefinido.

PRUEBA PARA LA TABLA CLIENTES

```sql
INSERT INTO Clientes (nombre, documento, correo, telefono) VALUES
('Juan Pérez', '12345678', 'juan.perez@email.com', '123-456-7890'),
('María Gómez', '87654321', 'maria.gomez@email.com', '987-654-3210'),
('Carlos López', '11223344', 'carlos.lopez@email.com', '555-123-4567'),
('Ana Martínez', '44332211', 'ana.martinez@email.com', '444-987-6543'),
('Pedro Sánchez', '55667788', 'pedro.sanchez@email.com', '666-555-4444');
```

Ilustración 30 Inserción de registros tabla Clientes

Mostrando filas 0 - 4 (total de 5, La consulta tardó 0,0003 segundos.)

```sql
SELECT * FROM Clientes;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

←T→				id_cliente	nombre	documento	correo	telefono
Editar	Copiar	Borrar		1	Juan Pérez	12345678	juan.perez@email.com	123-456-7890
Editar	Copiar	Borrar		2	María Gómez	87654321	maria.gomez@email.com	987-654-3210
Editar	Copiar	Borrar		3	Carlos López	11223344	carlos.lopez@email.com	555-123-4567
Editar	Copiar	Borrar		4	Ana Martínez	44332211	ana.martinez@email.com	444-987-6543
Editar	Copiar	Borrar		5	Pedro Sánchez	55667788	pedro.sanchez@email.com	666-555-4444

Ilustración 31Consulta de Tabla Clientes

Examinar | Estructura | SQL | Buscar | Insertar | E

Mostrar ventana de consultas SQL

1 fila eliminada. (La consulta tardó 0,0005 segundos.)

```
DELETE FROM Clientes WHERE id_cliente = 2;
```

[Editar en línea] [Editar] [Crear código PHP]

Ilustración 32 Borrado de Registro Tabla clientes

Mostrando filas 0 - 3 (total de 4, La consulta tardó 0,0006 segundos.)

```
SELECT * FROM `Clientes`;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

←T→	id_cliente	nombre	documento	correo	telefono
Editar Copiar Borrar	1	Juan Pérez	12345678	juan.perez@email.com	123-456-7890
Editar Copiar Borrar	3	Carlos López	11223344	carlos.lopez@email.com	555-123-4567
Editar Copiar Borrar	4	Ana Martínez	44332211	ana.martinez@email.com	444-987-6543
Editar Copiar Borrar	5	Pedro Sánchez	55667788	pedro.sanchez@email.com	666-555-4444

Seleccionar todo *Para los elementos que están marcados:* Editar Copiar Borrar Exportar

Ilustración 33 Confirmación de borrado de registro tabla Clientes

PRUEBA PARA LA TABLA RESERVAS

2 filas insertadas.
La Id de la fila insertada es: 2 (La consulta tardó 0,0005 segundos.)

```
INSERT INTO Reservas (id_cliente, id_habitacion, check_in, check_out, notas) VALUES (1, 1, '2025-10-03', '2025-10-05',
'Reserva para evento'), (1, 2, '2025-10-06', '2025-10-08', 'Reserva familiar');
```

[Editar en línea] [Editar] [Crear código PHP]

Ilustración 34Inserción de registros tabla Reservas

Mostrando filas 0 - 1 (total de 2, La consulta tardó 0,0003 segundos.)

```
SELECT * FROM reservas;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

	id_reserva	id_cliente	id_habitacion	check_in	check_out	notas
Editar Copiar Borrar	1	1	1	2025-10-03	2025-10-05	Reserva para evento
Editar Copiar Borrar	2	1	2	2025-10-06	2025-10-08	Reserva familiar

Seleccionar todo *Para los elementos que están marcados:* Editar Copiar Borrar Exportar

Ilustración 35Consulta Tabla Reservas

1 fila eliminada. (La consulta tardó 0,0003 segundos.)

```
DELETE FROM Reservas WHERE id_reserva = 1;
```

[Editar en línea] [Editar] [Crear código PHP]

Ilustración 36 Borrado de registro tabla Reservas

Mostrando filas 0 - 0 (total de 1, La consulta tardó 0,0003 segundos.)

```
SELECT * FROM Reservas;
```

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla

Opciones extra

	id_reserva	id_cliente	id_habitacion	check_in	check_out	notas
Editar Copiar Borrar	2	1	2	2025-10-06	2025-10-08	Reserva familiar

Seleccionar todo *Para los elementos que están marcados:* Editar Copiar Borrar Exportar

Ilustración 37Confirmacion de borrado de registro tabla Reservas

CONSULTA TABLA USUARIOS

SELECT * FROM Usuarios;

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según la clave: Ninguna

Opciones extra

←T→	id_usuario	nombre	rol	permisos	contrasena
Editar Copiar Borrar	1	Administrador	Administrador	gestion_usuarios,gestion_reservas	Administrador
Editar Copiar Borrar	2	Reservas	Encargado de Reservas	gestion_clientes,gestion_reservas	Reservas
Editar Copiar Borrar	3	Recepcionista	Recepcionista	gestion_reservas	Recepcionista
Editar Copiar Borrar	4	Housekeeping	Personal de Housekeeping	modificar_habitaciones	Housekeeping
Editar Copiar Borrar	5	Direccion	Direccion	ver_reportes	Direccion

Seleccionar todo *Para los elementos que están marcados:* Editar Copiar Borrar Exportar

Ilustración 38 Consulta tabla Usuarios

CONSULTA TABLA HABITACIONES

SELECT * FROM habitaciones;

Perfilando [Editar en línea] [Editar] [Explicar SQL] [Crear código PHP] [Actualizar]

Mostrar todo | Número de filas: 25 Filtrar filas: Buscar en esta tabla Ordenar según l

Opciones extra

←T→	id_habitacion	tipo	estado
Editar Copiar Borrar	1	TWIN	clean
Editar Copiar Borrar	2	TWIN	clean
Editar Copiar Borrar	3	TWIN	clean
Editar Copiar Borrar	4	TWIN	clean
Editar Copiar Borrar	5	TWIN	clean
Editar Copiar Borrar	6	KING	clean
Editar Copiar Borrar	7	KING	clean
Editar Copiar Borrar	8	KING	clean
Editar Copiar Borrar	9	KING	clean
Editar Copiar Borrar	10	KING	clean
Editar Copiar Borrar	11	SUPERIOR	clean
Editar Copiar Borrar	12	SUPERIOR	clean
Editar Copiar Borrar	14	SUITE	clean
Editar Copiar Borrar	15	SUITE	clean

Ilustración 39 Consulta tabla Habitaciones

COMUNICACIÓN

DEFINICION DE LAS COMUNICACIONES DEL SISTEMA

Para el prototipo del sistema PMS educativo, la definición de las comunicaciones se basa en los requisitos no funcionales (RQNF) proporcionados, que enfatizan un enfoque inicial en una implementación local y portable, con preparación para escalabilidad futura. El sistema se diseña para operar en un entorno cerrado en una PC local (compatible con Windows 10/11), utilizando Java como lenguaje de desarrollo y MySQL como base de datos, todo integrado en XAMPP como entorno de desarrollo web. Las comunicaciones se centran en interacciones eficientes y seguras entre componentes internos, con mecanismos para garantizar rendimiento (tiempos de respuesta < 2 segundos), confiabilidad (recuperación ante fallos), seguridad (autenticación) y soporte para futuras expansiones (escalabilidad cliente-servidor).

A continuación, se plantean los requerimientos de comunicación, se describen las interacciones entre componentes y se definen los protocolos y estándares adecuados, considerando el contexto del prototipo (instalación local sin red inicial, pero con potencial para conexiones concurrentes y migración a entornos de red).

REQUERIMIENTOS DE COMUNICACIÓN

Se establecen los siguientes requerimientos específicos para las comunicaciones del sistema:

Comunicación Interna entre Componentes de Software: El sistema debe facilitar interacciones modulares entre la capa de interfaz (UI en Java), la capa de lógica de negocio (controladores) y la capa de persistencia (base de datos MySQL), asegurando una estructura modular para mantenimiento y futuras integraciones.

Comunicación con la Base de Datos: Debe soportar conexiones locales eficientes, con preparación para conexiones concurrentes desde múltiples clientes, garantizando integridad de datos y tiempos de respuesta óptimos. Inicialmente local , pero escalable a remoto.

Comunicación con Dispositivos Periféricos: El sistema debe permitir conexiones sencillas con impresoras láser locales para impresión de reportes, compatible con impresoras estándar conectadas a la PC local.

Comunicación en Entorno de Red (Futuro): Aunque el prototipo es local, debe estar diseñado para migrar a una arquitectura cliente-servidor, soportando conexiones TCP/IP para múltiples dispositivos, con mecanismos de autenticación y copias de seguridad periódicas.

Seguridad y Confiabilidad en Comunicaciones: Todas las interacciones deben incluir autenticación para áreas críticas, con mecanismos de recuperación ante fallos para evitar pérdida de datos.

Portabilidad y Flexibilidad: Las comunicaciones deben ser independientes de la plataforma, aprovechando Java para portabilidad y optimizadas para entornos educativos con recursos limitados.

Estos requerimientos aseguran que el sistema sea intuitivo, capaz de manejar volúmenes moderados de datos y documentado adecuadamente para futuras mejoras.

INTERACCIONES ENTRE COMPONENTES

El sistema se organiza en componentes modulares (basados en los diagramas de clases y despliegue), con interacciones definidas como sigue:

Interacción entre la Aplicación Java y la Base de Datos (Principal):

Componentes Involucrados: Aplicación PMS (controladores en Java) y Base de Datos MySQL (hospedada en XAMPP).

Los controladores (UsuarioController, ReservaController, etc.) interactúan con los DAOs para realizar operaciones CRUD en las tablas (Usuarios, Clientes, Habitaciones, Reservas). Inicialmente, esto ocurre localmente en la misma PC, sin red. En el futuro, podría escalarse a conexiones remotas para múltiples usuarios concurrentes.

Interacción entre la Interfaz de Usuario y los Controladores:

Componentes Involucrados: Interfaz gráfica (e.g., InterfazReserva en Java Swing o similar) y Controladores (lógica de negocio).

La interfaz envía solicitudes a los controladores para procesar acciones del usuario. Esto es una comunicación interna en memoria, sin red, asegurando respuestas rápidas.

Interacción con Periféricos (Impresión):

Componentes Involucrados: Aplicación PMS y Impresora Láser local (conectada via USB o red local simple).

Desde cualquier módulo, el sistema genera reportes y los envía a la impresora. En el futuro, podría extenderse a impresoras compartidas en red.

Interacción para Copias de Seguridad y Recuperación:

Componentes Involucrados: Aplicación PMS y Base de Datos MySQL.

La aplicación puede invocar scripts o comandos para respaldos periódicos, almacenados localmente o en la nube futura.

Interacción Futura en Red:

Componentes Involucrados: Múltiples PCs (clientes) y Servidor MySQL (en XAMPP o dedicado).

Para escalabilidad, los clientes se conectarían al servidor MySQL remoto, permitiendo accesos concurrentes desde diferentes dispositivos educativos.

PROTOCOLOS Y ESTANDARES PARA LA COMUNICACION

Se seleccionan protocolos y estándares que aseguren eficiencia, seguridad y compatibilidad, considerando el entorno local inicial y la escalabilidad futura:

Protocolo para Comunicación con la Base de Datos:

Protocolo Principal: JDBC (Java Database Connectivity), estándar para conectar aplicaciones Java a bases de datos relacionales como MySQL.

Proporciona una interfaz estándar para consultas SQL, transacciones y conexiones seguras. Inicialmente local, con soporte para remoto en migraciones futuras.

Entorno de Red:

Inicialmente sin red (local en PC), usando loopback (127.0.0.1) para comunicaciones internas.

En el futuro TCP/IP para conexiones cliente-servidor, permitiendo accesos desde múltiples PCs en una LAN educativa. Puerto estándar 3306 para MySQL.

TCP/IP asegura fiabilidad y orden de paquetes, adecuado para entornos educativos con bajo volumen de datos.

Infraestructura Física:

Nodo Principal: PC local con Windows 10/11, ejecutando XAMPP para hospedar Apache/MySQL.

Periféricos: Conexión USB o inalámbrica para impresoras láser estándar, usando el estándar USB 2.0/3.0 o Wi-Fi (IEEE 802.11) para configuración sencilla.

En el futuro un servidor dedicado o cloud para escalabilidad, con hardware que soporte múltiples conexiones.

Garantiza portabilidad y optimización para sistemas operativos especificados.

Control de Enlace de Datos:

No aplica directamente a nivel bajo (enlace), ya que el sistema es de aplicación; sin embargo, se usa el estándar Ethernet (IEEE 802.3) para conexiones locales futuras, con manejo de errores en JDBC (reintentos en fallos de conexión).

Asegura transmisión fiable de datos en redes futuras, con checksums para integridad.

Protocolos para Periféricos (Impresión):

Protocolo Principal dependiente de Java Printing Service (JPS), estándar para impresión en Java sumado el estándar IPP (Internet Printing Protocol) para impresoras láser compatibles, o PCL/PS para formatos de impresión.

Permite impresión directa desde módulos, con configuración sencilla en Windows.

Seguridad en Comunicaciones:

Autenticación básica en MySQL (usuario/root) y cifrado TLS para conexiones futuras.

Se aplicará el estándar SHA-256 para hashing de contraseñas en la tabla Usuarios.

Protege datos sensibles y privilegios de administrador, con copias de seguridad automáticas via scripts MySQL (mysqldump).

SECCION 5: PRUEBAS

En esta etapa de pruebas, suponemos que el código Java para el prototipo del sistema PMS educativo ya ha sido implementado, basado en los diagramas de clases de diseño, diagramas de secuencia, despliegue y el modelo de datos relacional con MySQL. Se han incorporado instrucciones de log usando System.out.println en el código para rastrear el flujo de ejecución en cada módulo y caso de uso, permitiendo pruebas de caja blanca. Las pruebas de caja negra se enfocan en validar el comportamiento del sistema desde la perspectiva del usuario, sin acceso al código interno.

Se realiza un plan de pruebas con trazabilidad, vinculando los casos de uso desde CU-001 al CU-007 a pruebas específicas. La nomenclatura de las pruebas es: CP (Caso de Prueba) + Tipo (C para componente/caja blanca, S para sistema/caja negra) + Número (dos dígitos relacionados al número de caso de uso.

PLAN DE PRUEBAS

Caso de Uso	Descripción del Caso de Uso	Prueba Asociada	Tipo de Prueba	Trazabilidad a Requerimientos Funcionales	Objetivo de la Prueba
CU-001: Crear Usuarios	Crear un usuario con rol y permisos.	CPC01	Caja Blanca (Componente)	RQF-001	Verificar flujo interno mediante logs para creación de usuario válido/inválido.
CU-001: Crear Usuarios	Crear un usuario con rol y permisos.	CPS01	Caja Negra (Sistema)	RQF-001	Validar resultado del sistema con datos válidos (éxito) e inválidos (error).
CU-002: Modificar Usuarios	Modificar información, rol y permisos de un usuario existente.	CPC02	Caja Blanca (Componente)	RQF-002	Verificar flujo interno mediante logs para modificación válida/inválida.
CU-002: Modificar Usuarios	Modificar información, rol y permisos de un usuario existente.	CPS02	Caja Negra (Sistema)	RQF-002	Validar resultado del sistema con datos válidos (éxito) e inválidos (error).
CU-003: Listar Habitaciones	Listar todas las habitaciones predefinidas.	CPC03	Caja Blanca (Componente)	RQF-003	Verificar flujo interno mediante logs para consulta de lista.
CU-003: Listar Habitaciones	Listar todas las habitaciones predefinidas.	CPS03	Caja Negra (Sistema)	RQF-003	Validar resultado del sistema (lista completa) o error si no hay datos.

CU-004: Modificar Habitación	Actualizar el estado de una habitación predefinida.	CPC04	Caja Blanca (Componente)	RQF-004, RQF-005	Verificar flujo interno mediante logs para actualización de estado válido/inválido.
CU-004: Modificar Habitación	Actualizar el estado de una habitación predefinida.	CPS04	Caja Negra (Sistema)	RQF-004, RQF-005	Validar resultado del sistema con estado válido (éxito) e inválido (error).
CU-005: Gestionar Perfil Cliente	Crear o modificar perfil de cliente.	CPC05	Caja Blanca (Componente)	RQF-006, RQF-008	Verificar flujo interno mediante logs para gestión válida/inválida.
CU-005: Gestionar Perfil Cliente	Crear o modificar perfil de cliente.	CPS05	Caja Negra (Sistema)	RQF-006, RQF-008	Validar resultado del sistema con datos válidos (éxito) e inválidos (error).
CU-006: Crear Reserva	Crear una reserva asociada a un cliente y habitación.	CPC06	Caja Blanca (Componente)	RQF-011, RQF-012, RQF-013	Verificar flujo interno mediante logs para creación válida/inválida.
CU-006: Crear Reserva	Crear una reserva asociada a un cliente y habitación.	CPS06	Caja Negra (Sistema)	RQF-011, RQF-012, RQF-013	Validar resultado del sistema con datos válidos (éxito) e inválidos (error).

CU-007: Modificar Reserva	Modificar una reserva existente, incluyendo fechas y notas.	CPC07	Caja Blanca (Componente)	RQF-014, RQF-015, RQF-016	Verificar flujo interno mediante logs para modificación válida/inválida.
CU-007: Modificar Reserva	Modificar una reserva existente, incluyendo fechas y notas.	CPS07	Caja Negra (Sistema)	RQF-014, RQF-015, RQF-016	Validar resultado del sistema con datos válidos (éxito) e inválidos (error).

Tabla 5 Tabla de trazabilidad plan de prueba - casos de uso - requisitos funcionales candidatos

PROCEDIMIENTO DE PRUEBA

El procedimiento de prueba se ejecuta en un entorno controlado (XAMPP en Windows 10/11), con el código Java implementado con logs para caja blanca y una interfaz para caja negra. Se sigue un enfoque iterativo:

Preparación: Configurar la base de datos MySQL con datos predefinidos (habitaciones, usuarios). Compilar y ejecutar la aplicación Java en modo debug para capturar logs (caja blanca). Preparar datos de prueba válidos (nombre de usuario único) e inválidos (rol no permitido).

Ejecución de Pruebas de Caja Blanca (Componente):

- Ejecutar el módulo correspondiente
- Monitorear logs para rastrear el flujo: validación de datos, llamada a DAO, inserción en MySQL, manejo de errores.
- Registrar si el flujo coincide con el esperado.

Ejecución de Pruebas de Caja Negra (Sistema):

- Usar la interfaz del sistema para ingresar datos válidos (para CPS01, crear usuario con rol válido) y verificar el resultado (éxito: mensaje "Usuario creado").
- Ingresar datos inválidos (rol vacío) y verificar el resultado (error: mensaje "Datos inválidos").
- Registrar el output del sistema sin acceder al código interno.

Evaluación y Registro: Comparar resultados con expectativas. Si se detectan defectos, documentarlos en el tablero de tratamiento. Repetir pruebas hasta cubrir todos los casos.

Finalización: Generar reportes de pruebas, incluyendo logs y capturas de pantalla, asegurando cobertura del 100% de los CU del MVP.

TABLERO DE EVALUACION DE PRUEBAS

Este tablero resume la evaluación de las pruebas con resultados simulados. Cada prueba se evalúa como "Pasada" (éxito total), "Fallida" (defecto detectado) o "Parcial" (éxito parcial).

Prueba	Caso de Uso Asociado	Tipo	Resultado	Observaciones
CPC01	CU-001	Caja Blanca (Componente)	Pasada	Logs muestran flujo correcto: validación -> inserción en MySQL.
CPS01	CU-001	Caja Negra (Sistema)	Pasada	Datos válidos: Usuario creado; datos inválidos: Error mostrado.
CPC02	CU-002	Caja Blanca (Componente)	Parcial	Logs rastrean actualización, pero excepción en permisos inválidos no registrada completamente.
CPS02	CU-002	Caja Negra (Sistema)	Pasada	Modificación exitosa con datos válidos; error con inválidos.
CPC03	CU-003	Caja Blanca (Componente)	Pasada	Logs confirman consulta a MySQL y retorno de lista.
CPS03	CU-003	Caja Negra (Sistema)	Pasada	Lista de habitaciones mostrada correctamente.
CPC04	CU-004	Caja Blanca (Componente)	Fallida	Logs indican flujo, pero no valida permisos correctamente.
CPS04	CU-004	Caja Negra (Sistema)	Parcial	Estado actualizado con válido; error con inválido, pero mensaje confuso.
CPC05	CU-005	Caja Blanca (Componente)	Pasada	Logs rastrean gestión de perfil.

CPS05	CU-005	Caja Negra (Sistema)	Pasada	Perfil creado/modificado correctamente.
CPC06	CU-006	Caja Blanca (Componente)	Pasada	Logs confirman validación de fechas y inserción.
CPS06	CU-006	Caja Negra (Sistema)	Pasada	Reserva creada con válido; error con inválido (e.g., fecha pasada).
CPC07	CU-007	Caja Blanca (Componente)	Fallida	Logs muestran flujo, pero no maneja cancelación adecuadamente.
CPS07	CU-007	Caja Negra (Sistema)	Pasada	Modificación exitosa; cancelación genera error esperado.

Tabla 6 Tabla de evaluación de pruebas

TABLERO DE TRATAMIENTO DE DEFECTOS

Este tablero registra defectos detectados durante las pruebas, con detalles para su tratamiento. Cada defecto se asigna un ID, prioridad (Alta, Media, Baja), estado (Abierto, En Progreso, Resuelto) y acciones correctivas.

ID Defecto	Prueba Asociada	Descripción del Defecto	Prioridad	Estado	Acciones Correctivas	Responsable
DEF-001	CPC02	Excepción en permisos inválidos no registrada en logs.	Media	Abierto	Añadir log adicional en bloque de error.	Desarrollador
DEF-002	CPS04	Mensaje de error confuso para estado inválido.	Baja	En Progreso	Mejorar texto de mensaje en interfaz.	Desarrollador
DEF-003	CPC04	Validación de permisos no se ejecuta correctamente en flujo.	Alta	Abierto	Corregir condicional en controlador.	Desarrollador

DEF-004	CPC07	Manejo de cancelación no registra logs completos.	Media	Abierto	Agregar logs en rama de eliminación.	Desarrollador

Tabla 7 Tabla de tratamiento de defectos