

Motel Cupido

Kevin Santiago Rachez, Juan Felipe Hincapié

Profesor: German Rodríguez

UCundinamarca

Ingeniería de Sistemas y Computación

Fusagasugá

2025

Contenido

Resumen (Abstract)..... 3

Implementacion de SCRUM en “Motel Cupido”..... 4

Objetivos Específicos 6

Ejecución del Proyecto 7

Diagrama de Casos de Uso 8

Descripción de Actores 8

 Recepcionista..... 9

Especificación de Casos de Uso 10

Diagrama de Secuencia 12

 Escenarios: 14

Diagrama de Procesos 15

Diagrama de Comunicación..... 16

Diagrama de Actividades ¡Error! Marcador no definido.

Diagrama de Clases ¡Error! Marcador no definido.

Diagrama MER..... 17

 Tablas Principales 17

Diagrama de componentes 20

Resumen (Abstract)

This project focuses on automating common database tasks for a business by designing a relational database system with tables for customers, products, orders, and order details. Key stored procedures were implemented to handle order insertion, stock and customer updates, order deletion, order history retrieval, and automated backups. The solution ensures data integrity through transactional operations, optimizes efficiency in routine processes, and provides a structured approach to managing business data. Documentation covers database schema design, sample data insertion, stored procedure logic, and practical usage examples, offering a scalable and maintainable system for business automation. ´

diarias.

.

Metodología incremental

La metodología Incremental es una metodología clave para desarrollar proyectos de manera iterativa e incremental, se basa en “Sprints” que son iteraciones cortas de 2 a 3 semanas en las que se desarrollan partes específicas del programa, cada día se hacen reuniones rápidas para sincronizar los avances

En el proyecto “Motel Cupido” se usó esta metodología considerando 16 semanas de plazo que se tenían como máximo de tiempo

Fases de metodología en incremental:

Análisis, diseño, código, prueba

Implementación de Incremental en “Motel Cupido”:

16 semanas permiten dividir el proyecto en 4 sprints de 4 semanas, ideal para entregas incrementales como:

- *Sprint 1*: Sistema de login + roles (admin/recepcionista).
- *Sprint 2*: Módulo de habitaciones + cronómetros.
- *Sprint 3*: Gestión de productos + validación de saldo.
- *Sprint 4*: Integración de todos los sprints + pruebas.

1. Flexibilidad para cambios: En un proyecto donde se usan componentes en tiempo real como el cronometro de cada habitación, las reuniones diarias permiten reorientar el trabajo.
2. Gestión de prioridades: El Product Backlog prioriza funcionalidades claves como por ejemplo el login con sus dos usuarios.
3. Pruebas continuas: La restricción “*no hay check-out con saldo > 0*” requiere validación constante. En Incremental, el *Sprint Review* permite probar estas

reglas de negocio en cada entre

Objetivos Específicos

1.Optimizar la gestión operativa: Desarrollar módulos integrados para la administración eficiente de la información, el control de inventario y la gestión de cada habitación.

2.Mejorar la experiencia del cliente: Implementar un sistema de alertas automatizadas que notifique a los clientes sobre el tiempo restante de uso.

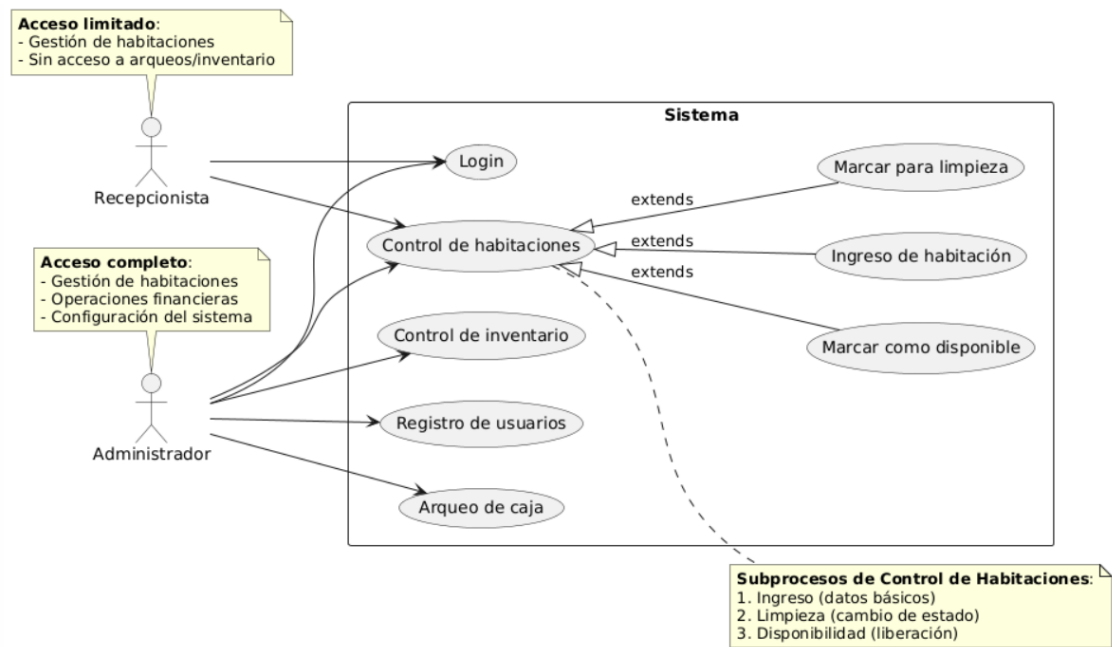
3.Digitalizar el cargo de cada habitación: Desarrollar apartado de cargos de productos consumidos para cada habitación.

Ejecución del Proyecto

| Característica de Motel Cupido | Cómo Scrum se adapta |
|--|---|
| Roles de usuario (admin/recepcionista) | Se implementan en sprints tempranos para definir permisos base. (Semana 1-4) |
| Cronómetro por habitación | Desarrollo iterativo: versión básica en Sprint 2, mejoras (ej: notificaciones) luego. (Semana 4- 8) |
| Carga de productos + validación de saldo | Historias de usuario independientes, probadas en cada sprint. (Semana 8-12) |
| Integración final | El <i>Sprint Review</i> garantiza que lo módulos funcionen juntos progresivamente. (Semana 12-16) |

Fase 1º metodología incremental, Diseño

Diagrama de Casos de Uso



Descripción de Actores

| | | |
|-----------------|--|-----------------------|
| Actor | ADMINISTRADOR | Identificador: 001 |
| Descripción | Usuario con privilegios para gestionar diferentes opciones | |
| Características | Acceso a funciones financieras y de configuración | |
| Relación | Comparte el caso de uso login con el recepcionista | |
| Referencias | UC1,UC3,UC4,UC5 | |

| Atributos | | |
|--------------|---------------------|--------|
| Nombre | Descripción | Tipo |
| ID | Identificador único | String |
| Nivel acceso | Privilegios | String |
| | | |
| | | |

Recepcionista

| | | |
|-----------------|--|-----------------------|
| Actor | Recepcionista | Identificador: 002 |
| Descripción | Gestión de habitaciones | |
| Características | Acceso limitado a registro de habitaciones | |
| Relación | Comparte login con administrador | |
| Referencias | Uc1,uc2 | |

| Atributos | | |
|-----------|----------------|--------|
| Nombre | Descripción | Tipo |
| Id | Identificaodor | string |
| | | |
| | | |

Especificación de Casos de Uso

| | | |
|---------------|---|------------------------------|
| Caso de Uso | Login | Identificador: UC1 |
| Actores | Administrador , recepcionista | |
| Tipo | primario | |
| Referencias | Todos los casos de uso | |
| Precondición | El sistema debe estar operativo | |
| Postcondición | Usuario autenticado | |
| Descripción | Permite a los usuarios validar su identidad para acceder al sistema | |
| Resumen | | |

| | | |
|---------------|---|------------------------------|
| Caso de Uso | Control de habitaciones | Identificador: UC2 |
| Actores | Recepcionista y administrador | |
| Tipo | Primario | |
| Referencias | UC2_1 (Ingreso), UC2_2 (Limpieza), UC2_3 (Disponibilidad) | |
| Precondición | Usuario debe haber iniciado sesión | |
| Postcondición | Estado de habitación actualizado en sistema | |
| Descripción | Permite gestionar el ciclo de vida de las habitaciones | |
| Resumen | | |

| | | |
|---------------|-----------------------|------------------------------|
| Caso de Uso | Arqueo de caja | Identificador: UC3 |
| Actores | Administrador | |
| Tipo | Primario | |
| Referencias | | |
| Precondición | Usuario administrador | |
| Postcondición | Reporte generado | |
| Descripción | | |
| Resumen | | |

| | | |
|---------------|---|------------------------------|
| Caso de Uso | Control de inventario | Identificador: UC4 |
| Actores | Administrador | |
| Tipo | Primario | |
| Referencias | | |
| Precondición | Usuario administrador | |
| Postcondición | Actualización y control de inventario | |
| Descripción | Generar diferentes estados de los inventarios | |
| Resumen | | |

| | | |
|---------------|---|------------------------------|
| Caso de Uso | Registro de usuarios | Identificador: UC5 |
| Actores | Administrador | |
| Tipo | Primario | |
| Referencias | | |
| Precondición | Usuario administrador | |
| Postcondición | Generación de usuarios | |
| Descripción | Creación de usuarios dentro del sistema | |
| Resumen | | |

Fase 2º Metodología, Diseño

Diagrama de Clases

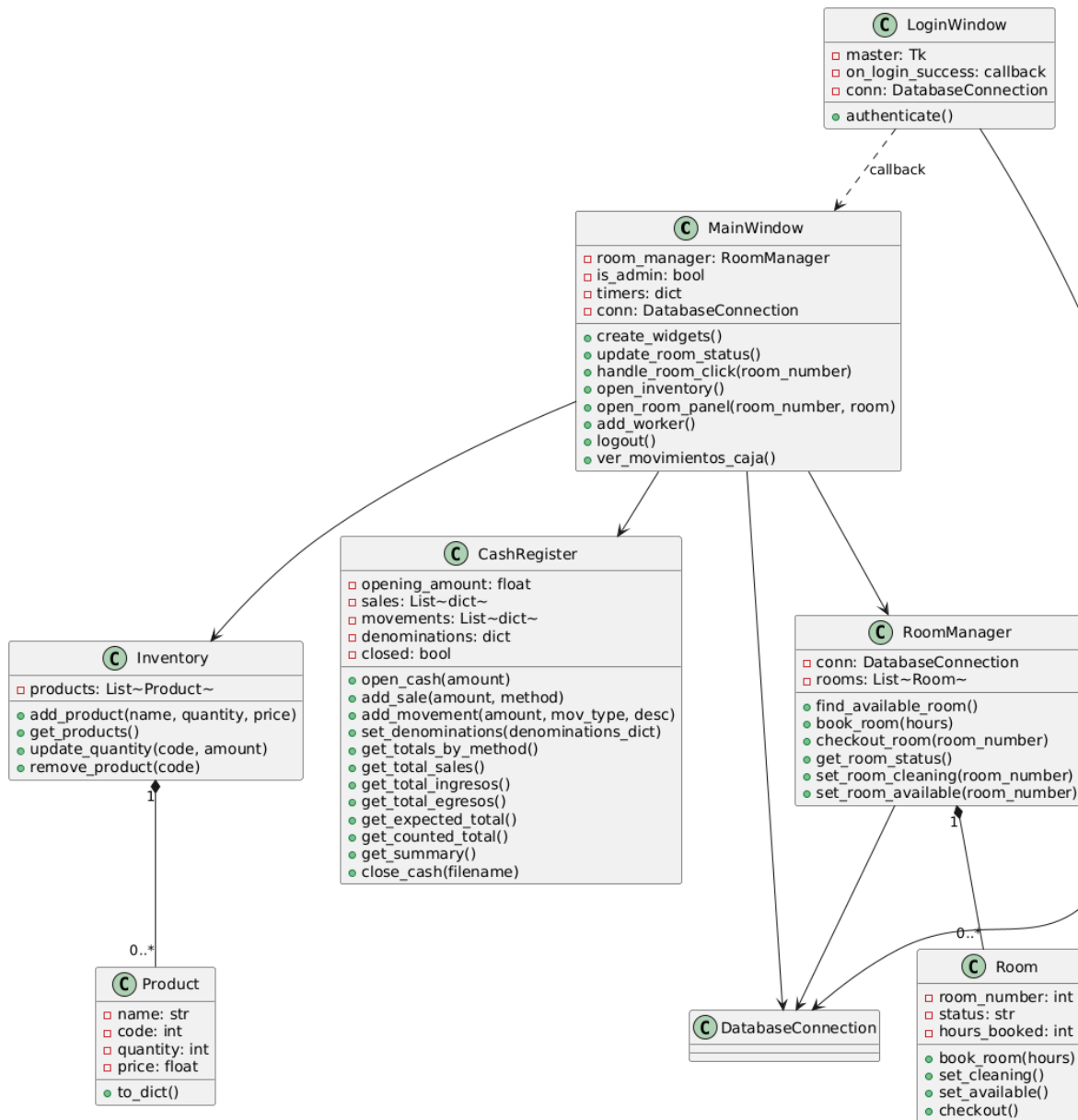
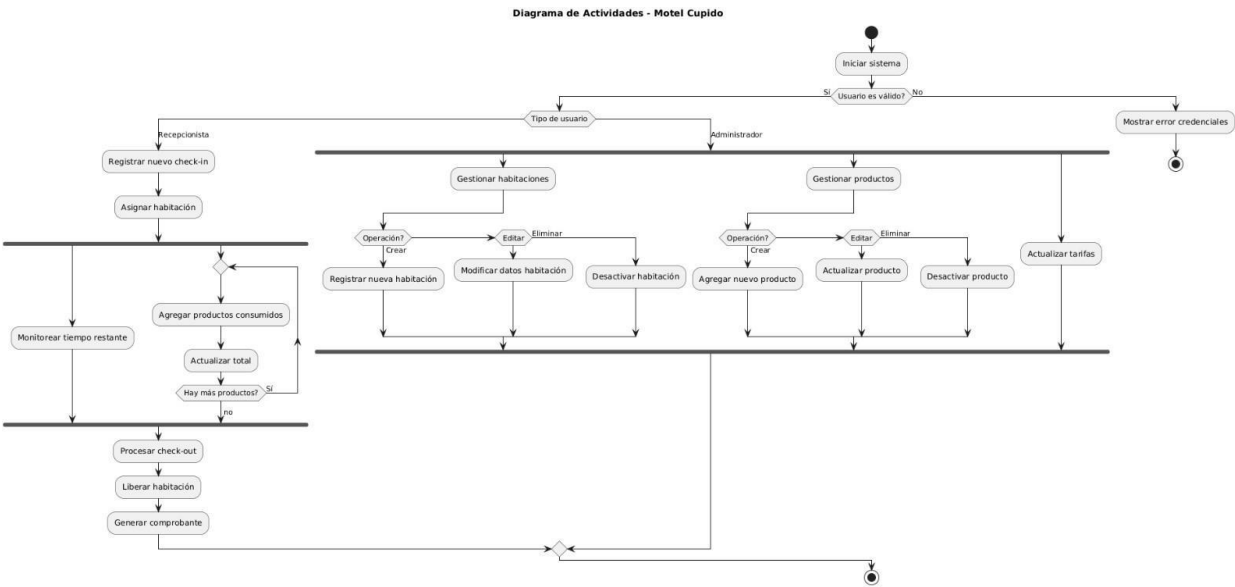
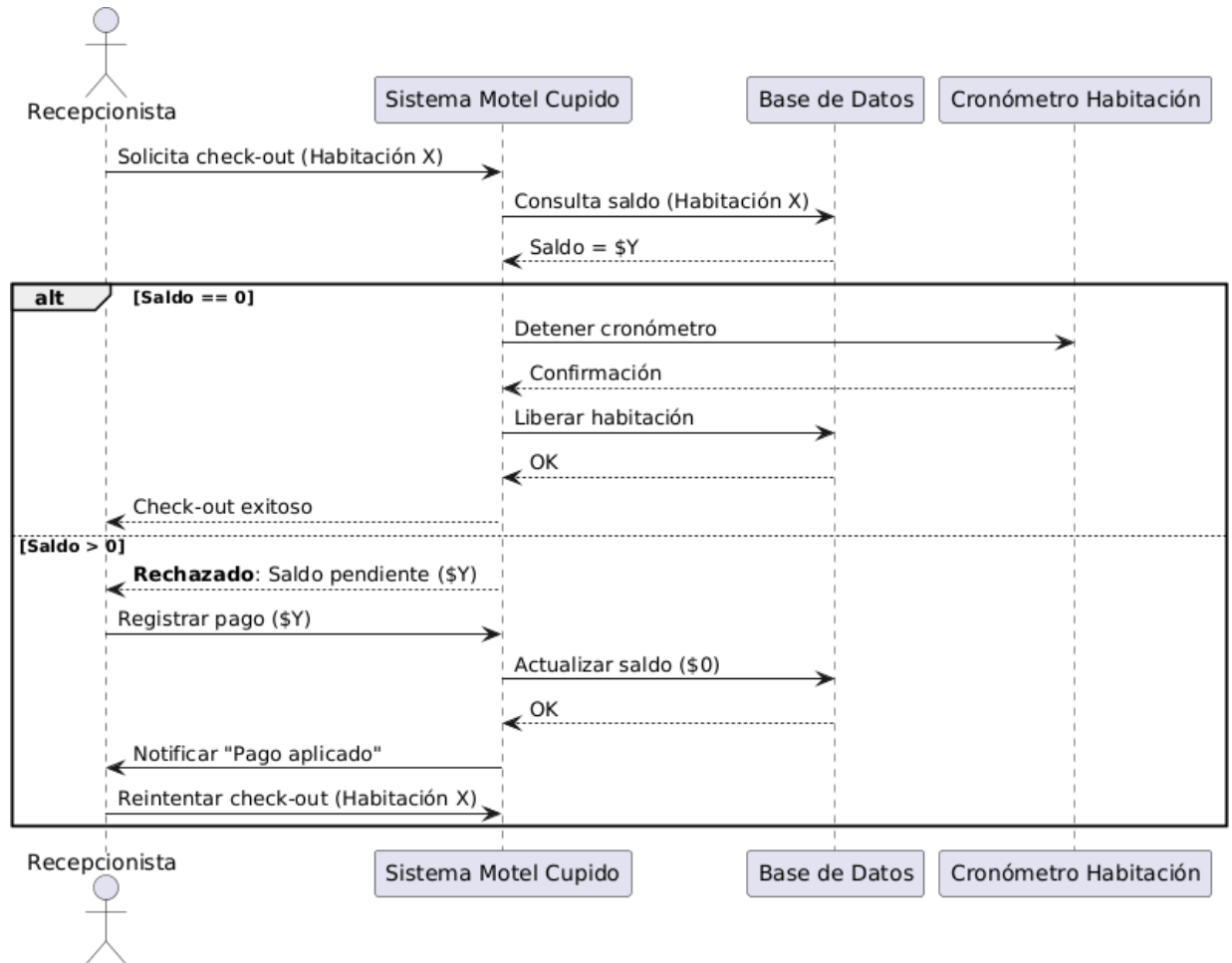


Diagrama de Actividades



Fase 3º Metodología, Código

Diagrama de Secuencia



1. **Inicio:** El recepcionista solicita el check-out de una habitación.
2. **Validación:** El sistema consulta el saldo en la BD.

Escenarios:

- **Saldo = \$0:** Detiene el cronómetro, libera la habitación y confirma.
- **Saldo > \$0:** Bloquea el check-out, obliga al pago y permite reintentar.

Diagrama de Procesos

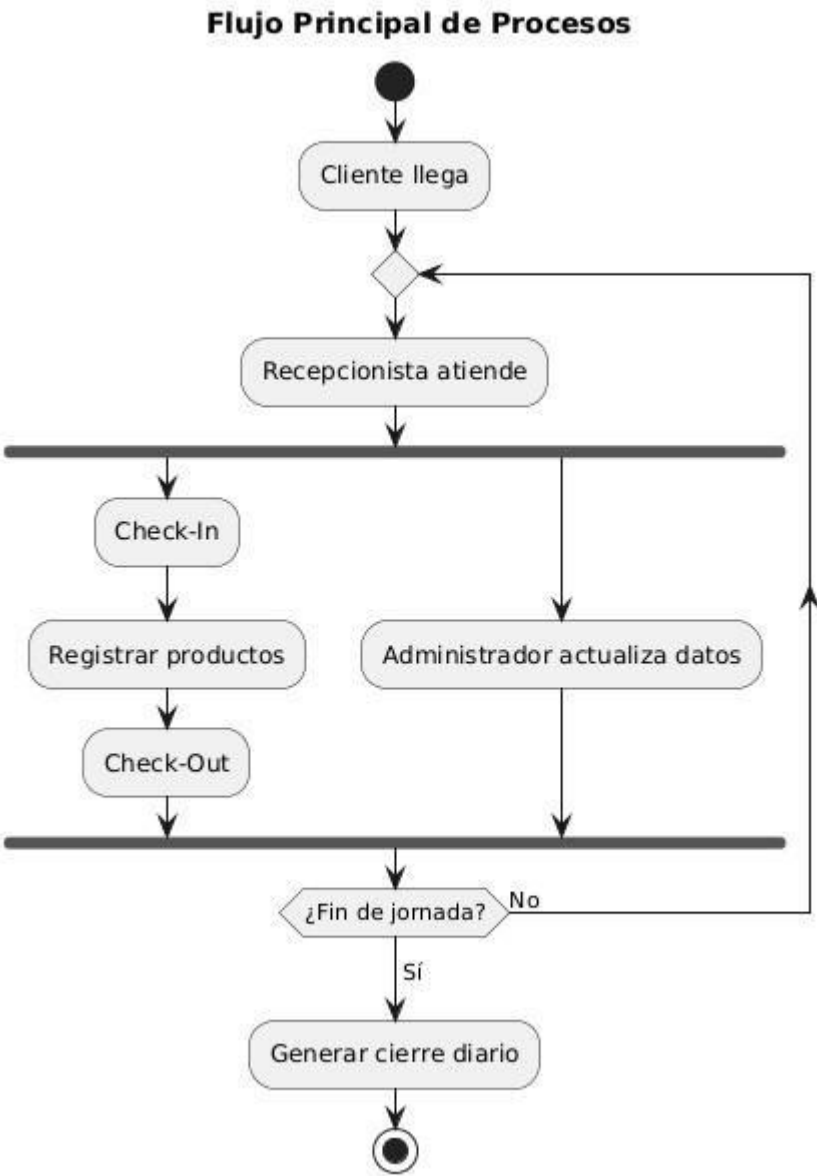


Diagrama de Comunicación

Diagrama de Comunicación - Motel Cupido (sin reportes)

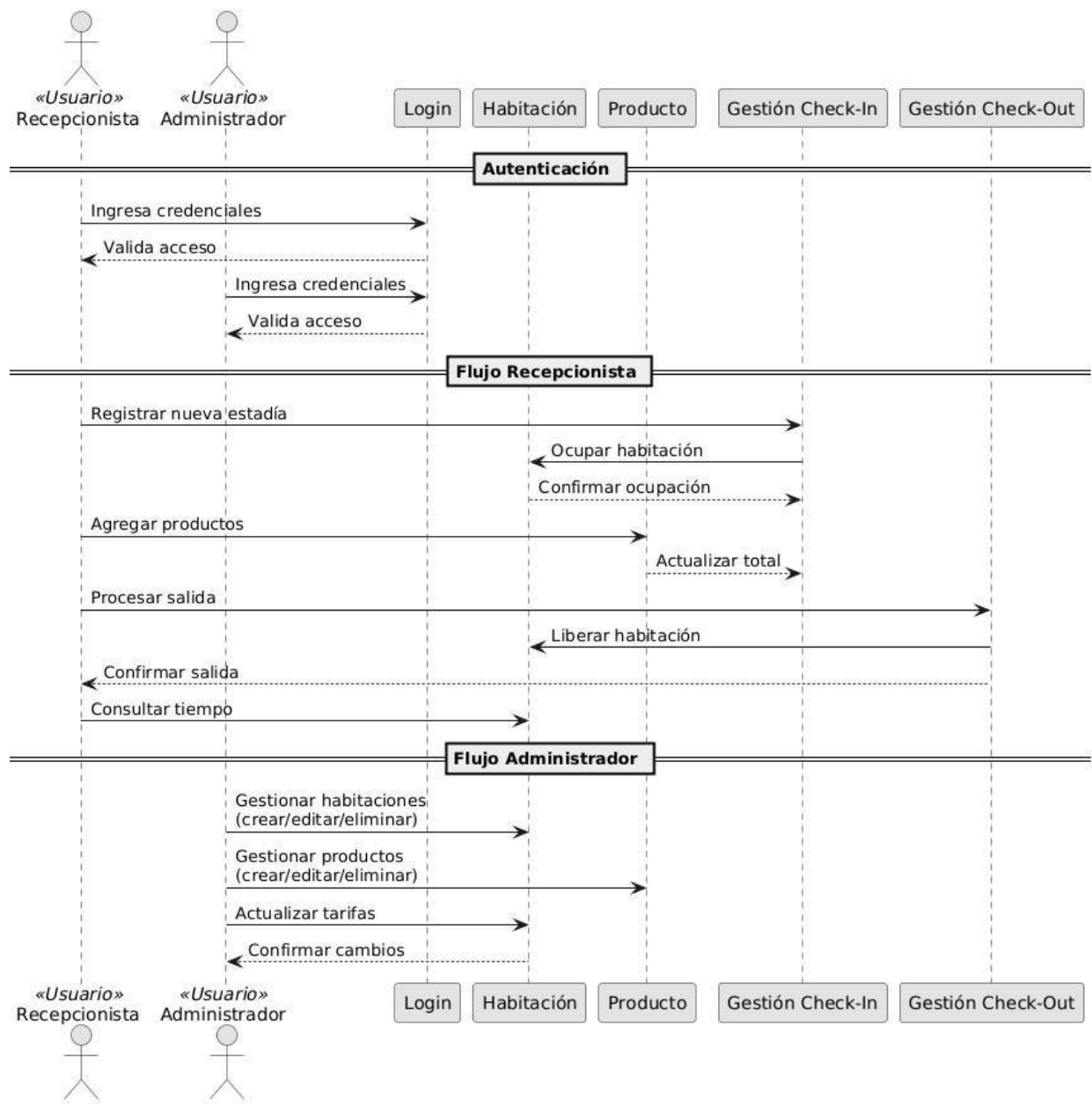
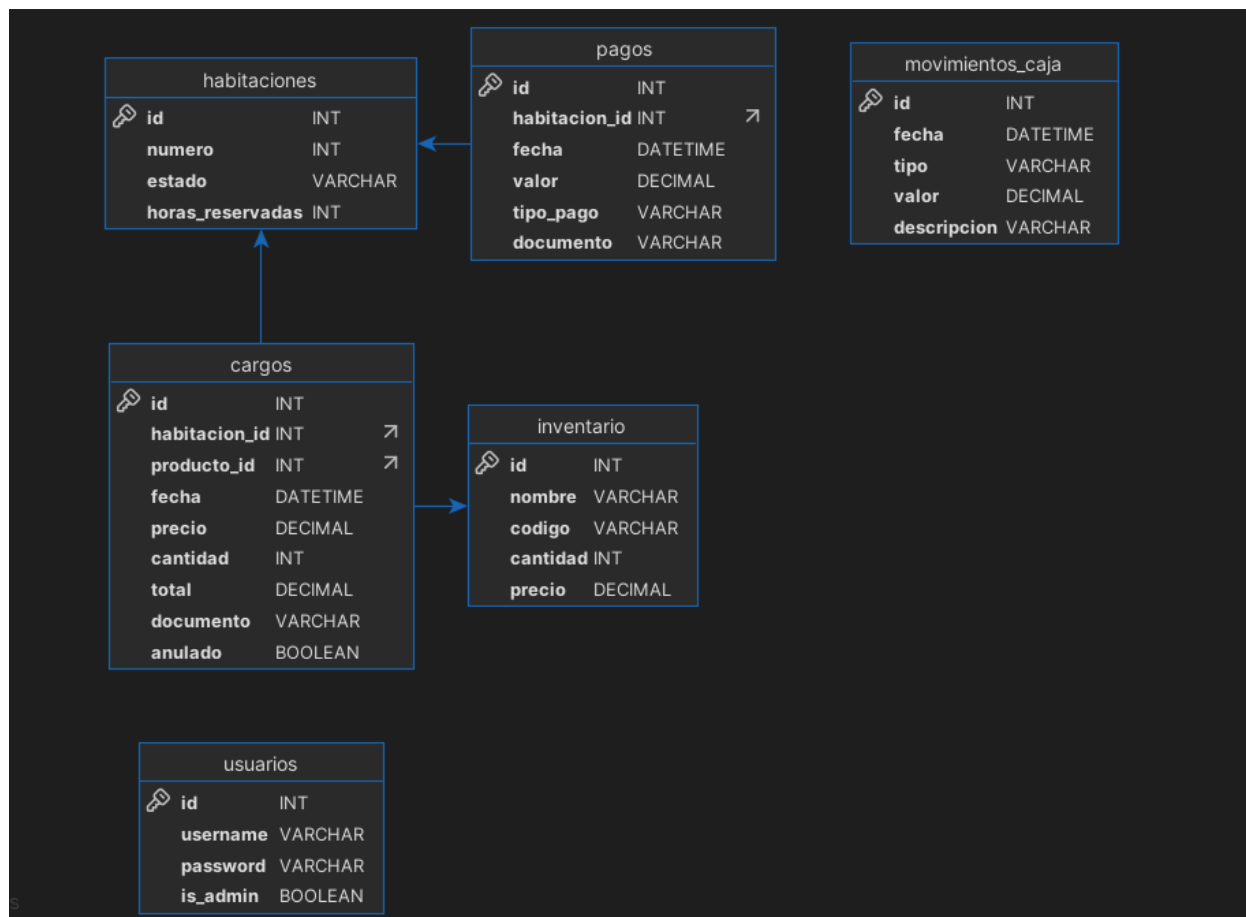


Diagrama MER



Tablas Principales

1. Tabla usuarios

- **Propósito:** Gestión de acceso al sistema
- **Campos clave:**
 - username y password para autenticación
 - is_admin para control de privilegios
- **Relaciones:** No tiene relaciones directas pero es fundamental para la seguridad del sistema

2. Tabla habitaciones

- **Propósito:** Control del estado y uso de las habitaciones

- **Campos clave:**
 - numero: Identificador único de la habitación
 - estado: Control del ciclo de vida (disponible/ocupada/limpieza)
 - horas_reservadas: Registro del tiempo de uso
- **Relaciones:**
 - Relacionada con cargocargoshabitacion_ids y pagos

3. Tabla inventario

- **Propósito:** Gestión de productos y suministros
- **Campos clave:**
 - codigo: Identificador único de productos
 - cantidad: Control de stock
 - precio: Valor unitario
- **Relaciones:**
 - Relacionada con cargocargoshabitacion_ids

4. Tabla cargocargoshabitacion_ids

- **Propósito:** Registro de consumos por habitación
- **Campos clave:**
 - habitacion_id y producto_id: Relaciones con otras tablas
 - total: Cálculo del valor (precio x cantidad)
 - anulado: Para reversión de cargos
- **Relaciones:**
 - Claves foráneas a habitaciones e inventario

5. Tabla pagos

- **Propósito:** Registro de transacciones financieras
- **Campos clave:**
 - tipo_pago: Clasificación del método de pago

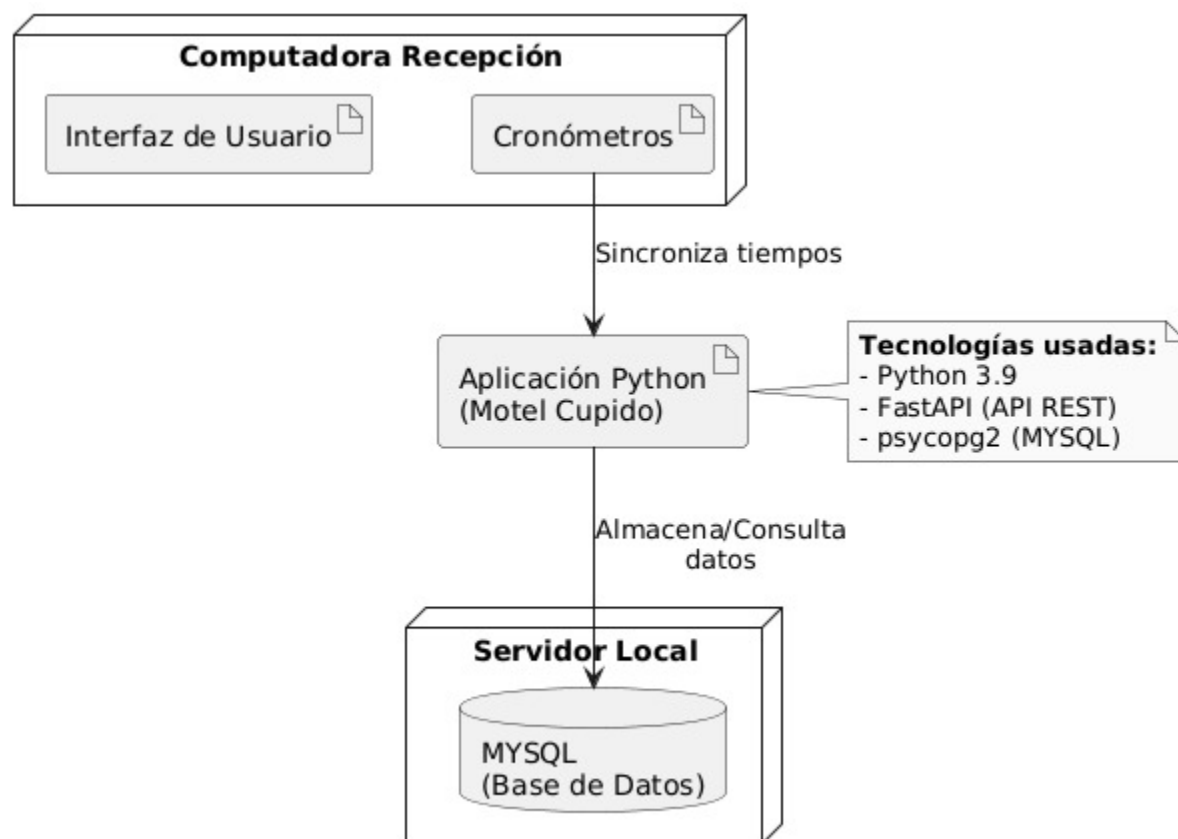
- valor: Monto de la transacción
- **Relaciones:**
 - Clave foránea a habitaciones

6. Tabla movimientos_caja

- **Propósito:** Control del flujo de efectivo
- **Campos clave:**
 - tipo: Ingreso o egreso
 - valor: Monto del movimiento
 - descripcion: Detalle de la operación

Fase 4º Prueba Metodología Prueba

Diagrama de componentes



Costo programa

1. Costo de Mano de Obra (Desarrolladores)

Sueldo mensual por desarrollador: \$2.000.000

Duración del proyecto: 16 semanas \approx 4 meses.

Costo total por desarrollador:

$\$2.000.000/\text{mes} \times 4 \text{ meses} = \$8.000.000$

Costo total para 2 desarrolladores:

$\$8.000.000 \times 2 = \$16.000.000$

2. Costos Adicionales (Aproximados)

En proyectos de software, además del salario, se deben considerar otros gastos como:

Infraestructura (licencias, herramientas, servidores): $\sim \$1.000.000 - \$3.000.000$.

Imprevistos/otros gastos: $\sim 10\text{-}20\%$ del costo total.

Ejemplo:

Si añadimos \$2.000.000 en infraestructura + 15% de imprevistos (\$2.400.000), el costo adicional sería $\sim \$4.400.000$.

3. Margen de Ganancia (Opcional)

Si el desarrollo lo realiza una empresa o freelancer, suele agregarse un margen de ganancia (20-30%).

Ejemplo con 25%:

$\$16.000.000 \text{ (salarios)} + \$4.400.000 \text{ (adicionales)} = \$20.400.000$

25% de ganancia ($\$5.100.000$) = $\$25.500.000$.

Valor Estimado del Software

Mínimo (solo salarios): \$16.000.000.