

DOCUMENTACIÓN TÉCNICA

SISTEMA DE GESTIÓN HOTELERA

Universidad Mariano Gálvez de Guatemala
Facultad de Ingeniería en Sistemas de Información

Curso: Programación II

Sección: A

Catedrático: Ing. MBA. Juan Carlos Méndez N.

Semestre: 4to

ÍNDICE

1. Introducción
2. Objetivos
3. Tecnologías Utilizadas
4. Arquitectura del Sistema
5. Base de Datos
6. Estructura del Proyecto
7. Componentes del Sistema
8. Flujo de Procesos
9. Seguridad
10. Requisitos del Sistema
11. Instalación y Configuración

1. INTRODUCCIÓN

El Sistema de Gestión Hotelera es una aplicación web desarrollada en Java que permite administrar de manera integral las operaciones de un hotel, incluyendo la gestión de reservas, habitaciones, check-in/check-out, cargos adicionales, pagos y bloqueo de habitaciones para mantenimiento.

1.1 Alcance del Proyecto

El sistema cubre los siguientes módulos:

- **Autenticación de usuarios** con sistema de login
- **Gestión de reservas** (crear, buscar, modificar, cancelar)
- **Control de habitaciones** (disponibilidad, ocupación, bloqueo)
- **Check-in y Check-out** de huéspedes
- **Registro de cargos adicionales** (minibar, lavandería, etc.)
- **Control de pagos** múltiples por reserva
- **Módulo de reportes** (ocupación, ingresos, disponibilidad)
- **Gestión de bloqueos** para mantenimiento

2. OBJETIVOS

2.1 Objetivo General

Desarrollar un sistema web integral para la gestión de operaciones hoteleras utilizando tecnologías Java EE, que permita controlar reservas, habitaciones, pagos y generar reportes estadísticos.

2.2 Objetivos Específicos

1. Crear un módulo de gestión de reservas con validación de datos
2. Desarrollar un sistema de control de disponibilidad de habitaciones
3. Implementar el registro y control de cargos adicionales y pagos
4. Generar reportes estadísticos de ocupación e ingresos
5. Aplicar principios de Programación Orientada a Objetos
6. Implementar validaciones robustas de datos

3. TECNOLOGÍAS UTILIZADAS

3.1 Lenguajes de Programación

- **Java 8+** - Lenguaje principal del backend
- **HTML5** - Estructura de páginas web
- **CSS3** - Estilos y diseño responsive
- **JavaScript** - Validaciones del lado del cliente
- **JSP (JavaServer Pages)** - Páginas dinámicas

3.2 Frameworks y Librerías

- **Java Servlets** - Controladores del backend
- **JDBC (Java Database Connectivity)** - Conexión a base de datos
- **MySQL Connector/J** - Driver de MySQL para Java

3.3 Herramientas de Desarrollo

- **NetBeans IDE 12+** - Entorno de desarrollo integrado
- **Apache Tomcat 9.0** - Servidor de aplicaciones
- **MySQL 8.0** - Sistema gestor de base de datos
- **phpMyAdmin** - Administración de base de datos
- **MySQL Workbench** - Diseño de diagramas ER

3.4 Justificación de Tecnologías

Java: Elegido por su robustez, seguridad, portabilidad y amplio soporte empresarial.

MySQL: Base de datos relacional confiable, de código abierto y con excelente rendimiento.

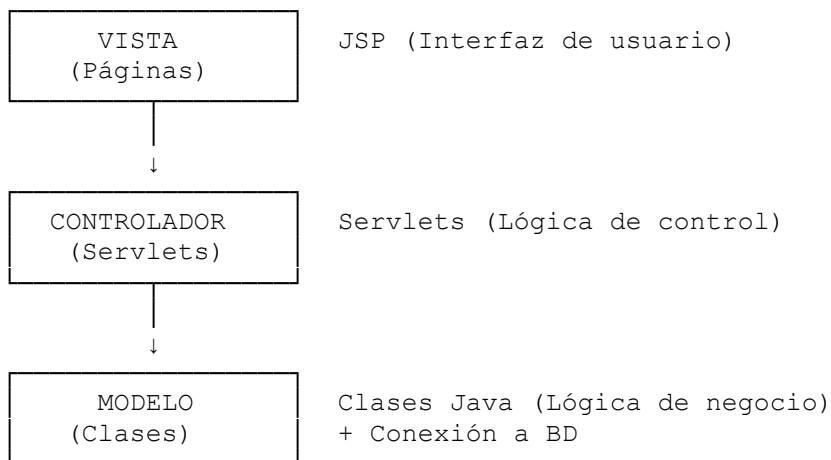
Servlets/JSP: Tecnología estándar de Java EE para aplicaciones web, con amplia documentación y soporte.

Apache Tomcat: Servidor ligero y eficiente, ideal para aplicaciones Java web.

4. ARQUITECTURA DEL SISTEMA

4.1 Patrón de Arquitectura

El sistema utiliza el patrón **MVC (Modelo-Vista-Controlador)** adaptado a Java EE:



4.2 Componentes Principales

1. **Capa de Presentación (Vista)**
 - Páginas JSP
 - CSS para estilos
 - JavaScript para validaciones
2. **Capa de Lógica de Negocio (Controlador)**
 - Servlets para cada módulo
 - Validaciones de datos
 - Procesamiento de solicitudes
3. **Capa de Acceso a Datos (Modelo)**
 - Clases de conexión a BD
 - Modelos de entidades
 - Operaciones CRUD
4. **Capa de Utilidades**
 - Sistema de logging
 - Validadores

- Excepciones personalizadas

5. BASE DE DATOS

5.1 Diagrama Entidad-Relación

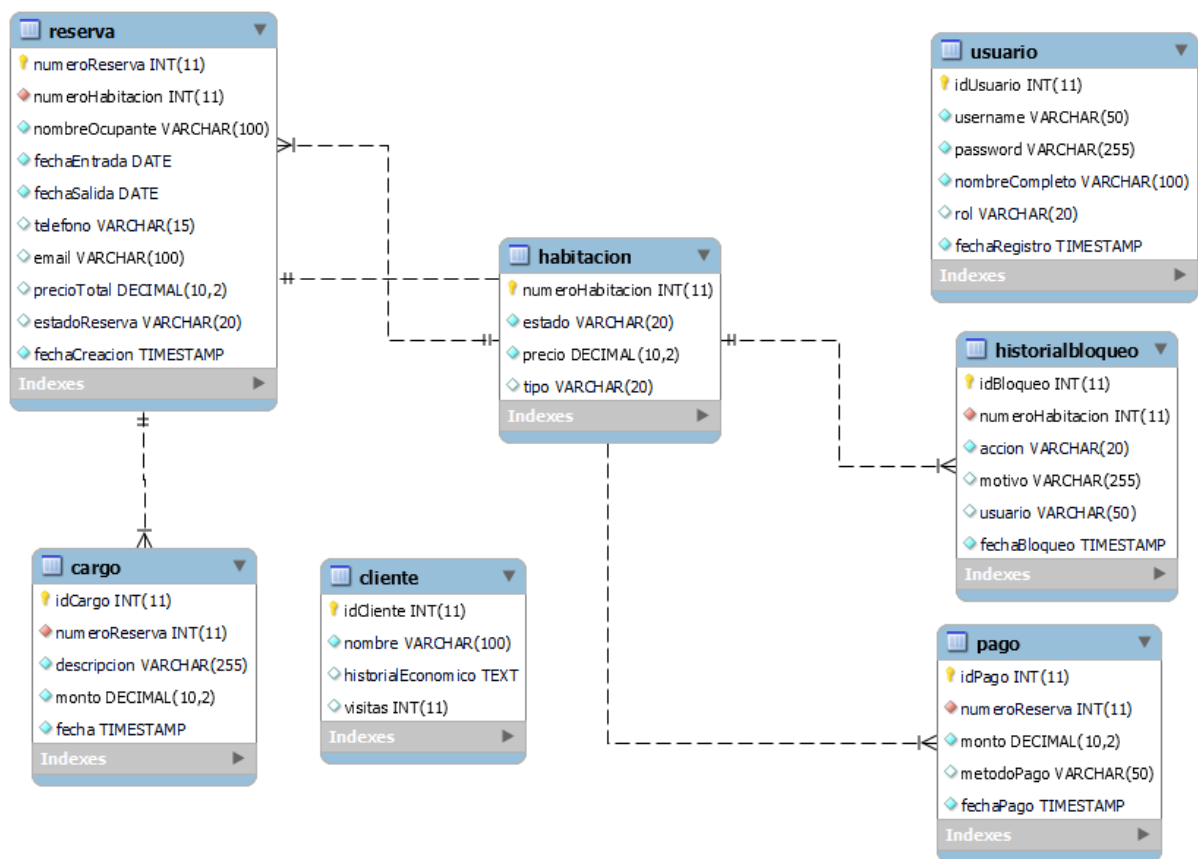
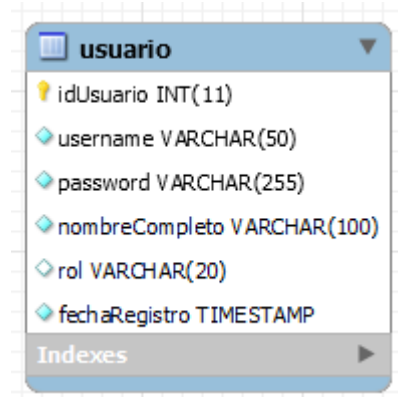


Diagrama ER generado con MySQL Workbench

5.2 Descripción de Tablas

5.2.1 Tabla USUARIO

Almacena los usuarios del sistema con credenciales de acceso.



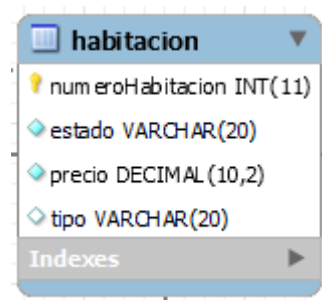
The screenshot shows the 'usuario' table structure with the following fields:

Field Name	Field Type
idUsuario	INT(11)
username	VARCHAR(50)
password	VARCHAR(255)
nombreCompleto	VARCHAR(100)
rol	VARCHAR(20)
fechaRegistro	TIMESTAMP

Below the fields, there is an 'Indexes' section with a right-pointing arrow.

5.2.2 Tabla HABITACION

Información de las habitaciones del hotel.



The screenshot shows the 'habitacion' table structure with the following fields:

Field Name	Field Type
numeroHabitacion	INT(11)
estado	VARCHAR(20)
precio	DECIMAL(10,2)
tipo	VARCHAR(20)

Below the fields, there is an 'Indexes' section with a right-pointing arrow.

5.2.3 Tabla RESERVA

Reservas realizadas por los clientes.

reserva	
num eroReserva	INT(11)
num eroHabitacion	INT(11)
nombreOcupante	VARCHAR(100)
fechaEntrada	DATE
fechaSalida	DATE
telefono	VARCHAR(15)
email	VARCHAR(100)
precioTotal	DECIMAL(10,2)
estadoReserva	VARCHAR(20)
fechaCreacion	TIMESTAMP
Indexes	

5.2.4 Tabla CARGO

Cargos adicionales a las reservas.

cargo	
idCargo	INT(11)
num eroReserva	INT(11)
descripcion	VARCHAR(255)
monto	DECIMAL(10,2)
fecha	TIMESTAMP
Indexes	

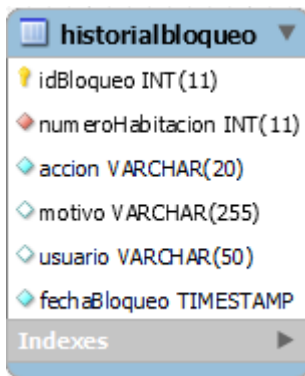
5.2.5 Tabla PAGO

Pagos realizados por los clientes.

pago	
idPago	INT(11)
num eroReserva	INT(11)
monto	DECIMAL(10,2)
metodoPago	VARCHAR(50)
fechaPago	TIMESTAMP
Indexes	

5.2.6 Tabla HISTORIALBLOQUEO

Registro de bloqueos de habitaciones.



5.3 Relaciones

- **HABITACION (1) → RESERVA (N):** Una habitación puede tener múltiples reservas
- **RESERVA (1) → CARGO (N):** Una reserva puede tener múltiples cargos
- **RESERVA (1) → PAGO (N):** Una reserva puede tener múltiples pagos
- **HABITACION (1) → HISTORIALBLOQUEO (N):** Una habitación puede tener múltiples bloqueos

5. ESTRUCTURA DEL PROYECTO

HotelManagement/

```
|
|
|— src/
|   |— main/
|       |— java/
|           |— hotelmanagement/
|               |— HotelManagement.java
```

```
|
|
|   └─ models/
|   |   └─ Reserva.java
|   |   └─ Habitacion.java
|   |
|   |
|   └─ servlets/
|   |   └─ LoginServlet.java
|   |   └─ LogoutServlet.java
|   |   └─ ReservationServlet.java
|   |   └─ BusquedaServlet.java
|   |   └─ CheckInOutServlet.java
|   |   └─ DetalleReservaServlet.java
|   |   └─ CargoServlet.java
|   |   └─ PagoServlet.java
|   |   └─ ReporteServlet.java
|   |   └─ BloqueoServlet.java
|   |   └─ BloquearHabitacionServlet.java
|   |
|   |
|   └─ utils/
|   |   └─ Validator.java
|   |   └─ Logger.java
|   |
|   |
|   └─ exceptions/
|       └─ HotelException.java
|
└─ web/
```

```
| |— login.jsp
| |— dashboard.jsp
| |— agregar-cargo.jsp
| |— agregar-pago.jsp
| |— WEB-INF/
|   |— web.xml
|
|— pom.xml
```

7. COMPONENTES DEL SISTEMA

7.1 Clases Principales

7.1.1 HotelManagement.java

Descripción: Clase principal que gestiona la conexión a la base de datos y operaciones CRUD básicas.

Métodos principales:

- **verificarDisponibilidad(int numeroHabitacion):** Verifica si una habitación está disponible
- **hacerReservaCompleta(...):** Crea una reserva completa con todos los datos
- **borrarReserva(int numeroReserva):** Elimina una reserva y libera la habitación
- **cerrarConexion():** Cierra la conexión a la base de datos

7.1.2 Validator.java

Descripción: Clase de utilidades para validar datos de entrada.

Métodos principales:

- **esTextoValido(String texto):** Valida que un texto no esté vacío
- **esEmailValido(String email):** Valida formato de email
- **esTelefonoValido(String telefono):** Valida formato de teléfono
- **validarFechas(Date entrada, Date salida):** Valida coherencia de fechas
- **validarNombreOcupante(String nombre):** Valida nombre del cliente

7.1.3 Logger.java

Descripción: Sistema de logging para registrar eventos del sistema.

Métodos principales:

- `info(String mensaje)`: Registra información general
- `warning(String mensaje)`: Registra advertencias
- `error(String mensaje, Throwable e)`: Registra errores
- `logAccion(String usuario, String accion, String detalle)`: Registra acciones de usuario

7.2 Servlets del Sistema

Servlet	Descripción	URL
LoginServlet	Autenticación de usuarios	/LoginServlet
LogoutServlet	Cierre de sesión	/LogoutServlet
ReservationServlet	Gestión de reservas	/ReservationServlet
BusquedaServlet	Búsqueda de reservas	/BusquedaServlet
CheckInOutServlet	Check-in y check-out	/CheckInOutServlet
DetalleReservaServlet	Detalle completo de reserva	/DetalleReservaServlet
CargoServlet	Agregar cargos adicionales	/CargoServlet
PagoServlet	Registrar pagos	/PagoServlet
ReporteServlet	Generación de reportes	/ReporteServlet
BloqueoServlet	Gestión de bloqueos	/BloqueoServlet

7. FLUJO DE PROCESOS

1. Usuario accede al sistema (Login)



2. Ingresa al Dashboard



3. Completa formulario de reserva



4. Sistema valida datos



5. Verifica disponibilidad de habitación



6. Calcula precio total



7. Crea reserva en BD



8. Actualiza estado de habitación a "Ocupada"



9. Muestra confirmación con número de reserva

8.3 Proceso de Check-out

1. Usuario busca reserva activa



2. Revisa estado de cuenta



3. Click en "Realizar Check-out"



4. Sistema actualiza estado a "Completada"



5. Libera habitación (estado "Disponible")



6. Genera resumen final

8. SEGURIDAD

9.1 Autenticación

- Sistema de login con usuario y contraseña
- Validación de credenciales contra base de datos
- Sesiones HTTP para mantener estado de autenticación

9.2 Autorización

- Verificación de sesión en cada página protegida
- Redirección a login si no hay sesión activa
- Roles de usuario (Admin/Usuario)

9.3 Validación de Datos

- Validaciones en el cliente (JavaScript)
- Validaciones en el servidor (Java)
- Sanitización de entradas de texto
- Prevención de SQL Injection (uso de PreparedStatements)

9.4 Gestión de Contraseñas

Nota: En esta versión las contraseñas se almacenan en texto plano. En producción se realizaría:

- algoritmos de hash (BCrypt, PBKDF2)
- Implementar salt único por usuario
- Política de complejidad de contraseñas

9. REQUISITOS DEL SISTEMA

Servidor:

- Procesador: Intel Core i3 o superior
- RAM: 4 GB mínimo, 8 GB recomendado
- Disco Duro: 1 GB de espacio libre
- Conexión a Internet (opcional)

Cliente:

- Navegador web moderno (Chrome, Firefox, Edge)
- Conexión de red al servidor

10.2 Requisitos de Software

Servidor:

- Sistema Operativo: Windows 10/11, Linux o macOS
- JDK 8 o superior
- Apache Tomcat 9.0 o superior
- MySQL 5.7 o superior
- XAMPP (alternativa todo-en-uno)

Cliente:

- Navegador web actualizado con soporte para HTML5, CSS3 y JavaScript

11. INSTALACIÓN Y CONFIGURACIÓN

11.1 Instalación de Dependencias

1. **Instalar JDK:**
 - Descargar de: <https://www.oracle.com/java/technologies/downloads/>
 - Configurar variable JAVA_HOME
2. **Instalar NetBeans:**
 - Descargar de: <https://netbeans.apache.org/download/>
 - Incluir Apache Tomcat en la instalación
3. **Instalar XAMPP:**
 - Descargar de: <https://www.apachefriends.org/>
 - Iniciar Apache y MySQL

11.2 Configuración de Base de Datos

1. Abrir phpMyAdmin: <http://localhost/phpmyadmin>
2. Crear base de datos "hoteldb"
3. Ejecutar script SQL de creación de tablas
4. Insertar usuario inicial (admin/admin123)

11.3 Configuración del Proyecto

1. Abrir proyecto en NetBeans
2. Configurar biblioteca MySQL Connector/J
3. Verificar configuración de conexión en HotelManagement.java:

java

jdbc:mysql://localhost:3306/hoteldb

Usuario: root

Contraseña: (vacía)

4. Clean and Build
5. Run Project

11.4 Acceso al Sistema

- **URL:** <http://localhost:8080/Hotel/>
- **Usuario:** admin
- **Contraseña:** admin123

CONCLUSIONES

El Sistema de Gestión Hotelera cumple con los requerimientos establecidos, implementando:

1. Programación Orientada a Objetos con clases bien estructuradas
2. Patrón MVC para separación de responsabilidades
3. Base de datos relacional normalizada
4. Validaciones robustas de datos
5. Sistema de logging para auditoría
6. Interfaz de usuario intuitiva y profesional
7. Módulos completos de gestión hotelera
8. Reportes estadísticos útiles