

UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Facultad de Ingeniería de Sistemas e Informática

Escuela Académica Profesional Ingeniería de Software



“Entrega parcial 1”

Autores:

- Bayona Vera, Elizabeth Ashley
- Giron Altamirano, Miguel Alejandro
- Melendez Cava, Andre Ivan
- Padilla Arellano Alejandro Manuel
- Torres Lezama, Mathias James

Profesor:

Chávez Soto, Jorge Luis

Curso:

Base de Datos II

Lima, Perú

2025

Perfil del Proyecto:

El presente proyecto, denominado "*Implementación de Sistema Multi-Tenant para Alquiler de Recintos Temporales*" (Código SMART-001), tiene como objetivo principal digitalizar, agilizar y facilitar el proceso de reserva de alojamientos temporales y experiencias turísticas. El sistema permitirá el registro de propiedades y anfitriones, la asignación a usuarios mediante cuentas de cliente, y el seguimiento de reservas, fechas de ingreso y salida, así como la disponibilidad de cada alojamiento. Se busca evitar el desorden de procesos manuales y optimizar la administración de estancias y experiencias mediante la integración de información de propiedades, anfitriones y clientes. La implementación de IA generativa para agilizar procesos es un componente clave. El proyecto tiene una duración estimada de 4 meses, con un costo total de S/ 802,720, y se espera que inicie el 23 de agosto de 2025. Se prevé un impacto significativo en la eficiencia operativa, la experiencia del usuario y la gestión de recursos.

Además, el proyecto incorpora un enfoque de responsabilidad social al promover la inclusión digital de pequeños anfitriones y emprendedores turísticos, quienes muchas veces no cuentan con plataformas tecnológicas para ofrecer sus servicios. De este modo, se contribuye a democratizar el acceso a la economía colaborativa y a fortalecer el turismo local, generando oportunidades de ingresos en comunidades que dependen de esta actividad. Asimismo, el sistema busca garantizar la transparencia en las transacciones y la seguridad de huéspedes y anfitriones, fomentando un turismo responsable y confiable. Este impacto social se alinea con los Objetivos de Desarrollo Sostenible (ODS) de las Naciones Unidas, en particular con el ODS 8, que promueve el trabajo decente y el crecimiento económico inclusivo, y el ODS 11, que impulsa la construcción de ciudades y comunidades sostenibles. La implementación de esta solución no solo busca mejorar procesos empresariales, sino también contribuir al desarrollo social y económico de los actores involucrados en el sector turístico.

Gestión del Proyecto:

La gestión de este proyecto se realizará bajo un ciclo de vida híbrido, combinando un enfoque predictivo con componentes iterativos. Se utilizará el marco del PMBOK para la gestión, abarcando las fases de Inicio, Planificación, Ejecución/Desarrollo, Monitoreo y Control, y Cierre. El progreso del proyecto será supervisado por un Jefe de Proyecto, quien asegurará el cumplimiento de los objetivos. El equipo de desarrollo incluirá programadores Front-end y Back-end, diseñadores UX/UI, ingenieros de IA, administradores de bases de datos e ingenieros QA. Se realizarán pruebas continuas y capacitaciones a usuarios clave. La comunicación con los stakeholders se mantendrá mediante reuniones semanales y reportes de progreso. Se hará seguimiento del cronograma y costos utilizando técnicas como el Valor Ganado (EV, PV, CV, SPI, CPI).

Los principales hitos programados son:

- Presentación y Aprobación del Project Charter: 27/08/2025
- Adquisición de equipamiento: 03/09/2025
- Diseño y conceptualización de la arquitectura: 10/09/2025
- Desarrollo del primer módulo (Gestión de propiedades y anfitriones): 17/09/2025
- Desarrollo del segundo módulo (Reserva y Disponibilidad): 01/10/2025
- Desarrollo del tercer módulo (Administración Multi-Tenant y Reportes): 22/10/2025
- Pruebas internas del sistema: 29/10/2025
- Capacitación de usuarios clave: 12/11/2025
- Despliegue y cierre del proyecto: 05/12/2025

Para el registro y seguimiento de estas actividades, se empleará un software de gestión de proyectos que permita al equipo documentar el progreso, gestionar tareas y monitorear el cumplimiento de los hitos y entregables. A continuación se muestra uno elaborado en el sistema OpenProject.

La Empresa:

La empresa cliente es una entidad que busca modernizar y optimizar la gestión de alquileres de recintos temporales y experiencias turísticas, buscando competir con plataformas líderes del sector como Airbnb, a través de la implementación de herramientas de Inteligencia Artificial.

- **Procesos y Supuestos:**

- **Registro de Propiedades y Anfitriones:** Los propietarios pueden registrar sus recintos y gestionarlos.
- **Gestión de Cuentas de Cliente:** Los usuarios pueden crear cuentas para buscar y reservar alojamientos.
- **Seguimiento de Reservas:** Gestión de fechas de ingreso y salida, y estado de disponibilidad.
- **Pagos:** Manejo de los pagos correspondientes a las reservas.
- **Administración de Inventario:** Los propietarios actualizan precios, disponibilidad, ubicación, imágenes y condiciones de uso.
- **Consulta de Disponibilidad y Características:** Los arrendatarios pueden consultar información detallada de los recintos.
- **Integración de IA:** Uso de IA generativa para agilizar procesos y mejorar la experiencia (ej. recomendaciones, optimización de precios).
- **Multi-tenant:** La plataforma soportará múltiples propietarios o administradores con sus respectivos inventarios.
- **Transparencia y Comodidad:** Se busca un servicio eficiente, transparente y cómodo para anfitriones y huéspedes.

- **Organización (Estructura Orgánica, Áreas, Personal, Clientes, Volumen de Ventas/Compras, etc.):**

Dado que los documentos adjuntos no detallan la estructura orgánica o el volumen de negocios específico de la empresa cliente, se estimará en base a referencias de empresas líderes del sector de alquileres temporales y plataformas tecnológicas.

- **Estructura Organizacional:** Se asume una estructura adaptable para soportar el crecimiento y la integración de nuevas tecnologías.
- **Áreas de la Empresa:** Se pueden proyectar áreas como Operaciones, Marketing, Soporte al Cliente, Tecnología (desarrollo, infraestructura), Finanzas y Recursos Humanos.
- **Locales:** Se asume un modelo de operación principalmente digital, con posible ubicación física para la administración central en Lima, Perú.
- **Personal:** El equipo de desarrollo para el proyecto incluye 1 Jefe de Proyecto, 3 Programadores Front-end, 2 Diseñadores UX/UI, 3 Programadores Back-end, 2 Ingenieros de IA, 2 Administradores de Bases de Datos y 1 Ingeniero QA. Esto implica un equipo de 14 personas dedicadas al desarrollo.
- **Clientes:** Se segmentan en arrendatarios (huéspedes) y propietarios/administradores de recintos (anfitriones). El volumen de clientes se proyectará en función de la capacidad de la plataforma y las estrategias de marketing.
- **Volumen de Ventas/Compras:** Se estima que el sistema manejará un volumen considerable de transacciones de reservas y pagos, impactando directamente en los ingresos generados por los alquileres.
- **Dimensionamiento y Carga de Datos:** El diseño del sistema debe prever una arquitectura escalable para gestionar un gran número de propiedades, usuarios y transacciones diarias, con una base de datos robusta.
- **Cálculo de Costos:** El costo total del proyecto de S/ 802,720 se desglosa en personal (S/ 600,000), máquinas (S/ 60,000), otros gastos administrativos (S/ 32,000), con reservas de contingencia (10%) y gestión (6%).

Modelo de Procesos de Negocios:

El objetivo del negocio es ofrecer una plataforma eficiente y moderna para el alquiler de recintos temporales. Los procesos de negocio clave incluyen:

1. Registro y Gestión de Anfitriones y Propiedades:

- Anfitrión se registra en la plataforma.
- Anfitrión carga información de su propiedad (descripción, fotos, precios, disponibilidad, reglas).
- Administrador revisa y aprueba la propiedad.
- Anfitrión gestiona el inventario de sus espacios (actualiza información, precios, disponibilidad).
- IA asiste en la optimización de precios y la descripción de propiedades.

2. Búsqueda y Reserva de Recintos:

- Arrendatario busca propiedades según criterios (ubicación, fechas, tipo de propiedad, precio).
- Sistema muestra resultados relevantes (posiblemente con IA para recomendaciones personalizadas).
- Arrendatario selecciona una propiedad y verifica su disponibilidad.
- Arrendatario procede a la reserva, ingresa detalles y realiza el pago.
- Sistema confirma la reserva y notifica a ambas partes.

3. Gestión de Reservas y Disponibilidad:

- Sistema actualiza la disponibilidad del recinto automáticamente.
- Anfitrión y Arrendatario pueden ver el estado de sus reservas.
- Proceso de check-in/check-out.
- Generación de reportes de ocupación y rendimiento para anfitriones.

4. Administración Multi-Tenant:

- La plataforma gestiona múltiples anfitriones y sus propiedades de forma independiente pero centralizada.
- Cada anfitrión tiene su propio panel de control.
- El sistema asegura la segregación de datos entre diferentes anfitriones.

5. Soporte y Atención al Cliente:

- Provisión de canales de comunicación entre arrendatarios, anfitriones y soporte.

- Gestión de incidencias y consultas.

Reglas de Negocios:

Las reglas de negocios que gobernarán la operación de la plataforma incluyen:

- Toda propiedad registrada debe ser aprobada por un administrador antes de ser listada públicamente.
- La disponibilidad de un recinto se actualizará en tiempo real tras cada reserva o cancelación.
- Los precios de los recintos pueden ser configurados por los anfitriones, con sugerencias o ajustes asistidos por IA.
- Las reservas solo serán confirmadas una vez que el pago haya sido procesado exitosamente.
- Los arrendatarios deben crear una cuenta de cliente para realizar reservas.
- Los anfitriones son responsables de mantener actualizada la información de sus propiedades.
- El sistema debe cumplir con las leyes de protección de datos para todos sus clientes.
- El sistema debe gestionar la información de propiedades, anfitriones y clientes de manera independiente para cada "tenant".
- Las cancelaciones y modificaciones de reservas seguirán políticas predefinidas por la plataforma y/o el anfitrión.
- El sistema de IA debe mejorar la experiencia del usuario y optimizar procesos sin intervención humana para decisiones críticas no supervisadas.

Modelo de Datos Conceptual:

El modelo conceptual de datos representará las entidades clave del negocio y sus relaciones, sin preocuparse por la implementación técnica específica.

- **Entidades:**
 - **USUARIO:** Representa a las personas que interactúan con la plataforma (puede ser Anfitrión o Arrendatario).

- Atributos: ID_Usuario, Nombre, Apellido, Email, Teléfono, DNI, Rol (Anfitrión/Arrendatario).
- **ANFITRIÓN:** Un tipo de USUARIO que posee propiedades.
 - Atributos: ID_Anfitrión (FK de USUARIO), RUC, Información_Contacto_Adicional.
- **ARRENDATARIO:** Un tipo de USUARIO que busca y reserva propiedades.
 - Atributos: ID_Arrendatario (FK de USUARIO), Información_Preferencia.
- **PREFERENCIA:** Catálogo de posibles preferencias que un arrendatario puede tener.
 - Atributos: ID_Preferencia, Descripción.
- **PROPIEDAD:** Representa un recinto disponible para alquiler temporal.
 - Atributos: ID_Propiedad, Nombre_Propiedad, Descripción, Dirección, Ciudad, País, Tipo_Recinto (apartamento, casa, habitación), Capacidad_Huéspedes, Número_Habitaciones, Número_Baños, Servicios (lista de servicios), Precio_Base_Noche, Estado_Aprobación, Anfitrión_ID (FK de ANFITRIÓN).
- **IMAGEN:** Fotos asociadas a una PROPIEDAD.
 - Atributos: ID_Imagen, URL_Imagen, Descripción_Imagen, Propiedad_ID, (FK de PROPIEDAD), Orden.
- **DISPONIBILIDAD:** Calendario de fechas en que una PROPIEDAD está disponible.
 - Atributos: ID_Disponibilidad, Fecha_Inicio, Fecha_Fin, Tipo (Reserva, Bloqueo Manual, Promoción, Mantenimiento), Precio_Override (un precio que está más elevado que el usual por algo en especial), Propiedad_ID (FK de PROPIEDAD).
- **RESERVA:** Registro de una reserva realizada por un ARRENDATARIO para una PROPIEDAD.
 - Atributos: ID_Reserva, Fecha_Inicio, Fecha_Fin, Numero_Huespedes, Monto_Total, Estado_Reserva (Pendiente, Confirmada, Cancelada, Completada), Fecha_Solicitud, Fecha_Aceptacion, Fecha_Checkin,

Fecha_Checkout, Arrendatario_ID (FK de ARRENDATARIO),

Propiedad_ID (FK de PROPIEDAD).

- **PAGO:** Transacciones de pago asociadas a una RESERVA.
 - Atributos: ID_Pago, Monto, Fecha_Pago, Estado_Pago (Pendiente, Completado, Fallido), Reserva_ID (FK de RESERVA), Metodo_Pago_ID (FK de METODO_PAGO)
- **METODO_PAGO:** Formas de pago registradas por los usuarios.
 - Atributos: ID_MetodoPago, Numero_Cuenta, Fecha_Vencimiento, Proveedor, Es_Predeterminado, Usuario_ID (FK de USUARIO), TipoPago_ID (FK de TIPO_PAGO).
- **RESEÑA:** Comentarios y valoraciones de ARRENDATARIOS sobre PROPIEDADES y ANFITRIONES sobre ARRENDATARIOS.
 - Atributos: ID_Reseña, Puntuacion, Comentario, Fecha_Creada, Fecha_Publicada, Fecha_Publicable, Publicada, Para_Arrendatario, ID_Reserva (FK de RESERVA), ID_Autor_Usuario (FK de USUARIO), ID_Receptor_Usuario (FK de USUARIO), ID_Propiedad (FK de PROPIEDAD).
- **CONVERSACIÓN:** Canal de comunicación entre arrendatarios y anfitriones respecto a una propiedad.
 - Atributos: ID_Conversacion, ID_Propiedad (FK de PROPIEDAD), ID_Arrendatario (FK de ARRENDATARIO), Fecha_Creada, Fecha_Cierre, Estado.
- **MENSAJE:** Mensajes enviados dentro de una conversación.
 - Atributos: ID_Mensaje, ID_Conversacion (FK de CONVERSACION), ID_Autor_Usuario (FK de USUARIO), Contenido, Fecha_Envio.
- **RELACIONES:**
 - USUARIO "es" ANFITRIÓN / ARRENDATARIO (Herencia/Especialización).
 - ARRENDATARIO “tiene” PREFERENCIAS

- ANFITRIÓN "posee" PROPIEDAD (1 a muchos).
- PROPIEDAD "tiene" IMAGENES (1 a muchos).
- PROPIEDAD "tiene" DISPONIBILIDAD (1 a muchos).
- ARRENDATARIO "realiza" RESERVA (1 a muchos).
- RESERVA "es para" PROPIEDAD (1 a 1).
- RESERVA "tiene" PAGO (1 a muchos).
- ARRENDATARIO "escribe" RESEÑA (1 a muchos).
- RESEÑA "es sobre" PROPIEDAD (1 a 1).

Modelo de Datos Lógico:

El modelo lógico de datos traduce el conceptual a una estructura más cercana a la implementación, generalmente relacional, organizada por áreas de información.

- **Área de Gestión de Usuarios:**

- **Tabla USUARIOS:**

- id_usuario (PK, INT)
- nombre (VARCHAR)
- apellido (VARCHAR)
- telefono (VARCHAR)
- fecha_nacimiento (DATE)
- tipo_usuario (ENUM 'Anfitrión', 'Arrendatario')
- fecha_registro (DATETIME)

- **Tabla CUENTAS:**

- id_cuenta(PK)
- id_usuario(FK a USUARIOS.ID_Usuario,INT)
- correo(VARCHAR)
- correo_verificado(BOOL)
- contraseña(VARCHAR)

- proveedor(VARCHAR)
 - id_proveedor (INT)
 - fecha_ultimo_login(DATETIME)
 - bloqueado_hasta(DATETIME)
 - fecha_creacion(DATETIME)
- **Tabla ANFITRION:**
 - id_anfitrion (PK, FK a USUARIOS.ID_Usuario, INT)
 - ruc (VARCHAR, UNIQUE)
 - direccion_contacto (TEXT)
- **Tabla ARRENDATARIO:**
 - id_arrendatario (PK, FK a USUARIOS.ID_Usuario, INT)
- **Tabla PREFERENCIA:**
 - id_preferencia (PK, INT)
 - nombre (VARCHAR)
 - descripcion (VARCHAR)
- **Tabla ARRENDATARIO_PREFERENCIA:**
 - id_arrendatario (FK a ARRENDATARIO.ID_Arrendatario, INT)
 - id_preferencia (FK a PREFERENCIA.ID_Preferencia, INT)
 - valor (VARCHAR)
- **Área de Gestión de Propiedades:**
 - **Tabla PROPIEDAD:**
 - id_propiedad (PK, INT)
 - id_anfitrion (FK a ANFITRION .ID_Anfitrion, INT)
 - disponible (BOOL)
 - nombre_propiedad (VARCHAR)
 - descripcion (TEXT)
 - direccion (VARCHAR)

- codigo_postal (VARCHAR)
- tipo_recinto (VARCHAR)
- capacidad_huespedes (INT)
- numero_habitaciones (INT)
- numero_banios (INT)
- estado_aprobacion (ENUM 'Pendiente', 'Aprobada', 'Rechazada')
- precio_base_noche (DECIMAL)
- fecha_creacion (DATETIME)

○ **Tabla DETALLES_PROPIEDAD:**

- id_propiedad (PK, FK → PROPIEDAD.id_propiedad)
- hora_checkin (TIME)
- hora_checkout (TIME)
- capacidad (INT)
- num_habitaciones (INT)
- num_baños (INT)
- num_camás (INT)
- area_m2 (DECIMAL)
- piso_numero (INT)
- fecha_creacion (DATETIME)
- fecha_actualizacion (DATETIME)

○ **Tabla SERVICIO:** (Para un manejo más flexible de servicios)

- id_servicio (PK, INT)
- nombre_servicio (VARCHAR)
- descripcion (TEXT)

○ **Tabla PROPIEDAD_SERVICIO:** (Tabla de unión para la relación M:M entre Propiedad y Servicio)

- id_propiedad (FK a PROPIEDAD.ID_Propiedad, INT)

- id_servicio (FK a SERVICIOS.ID_Servicio, INT)
 - (PK compuesta por Propiedad_ID, Servicio_ID)
- **Tabla IMAGENES_PROPIEDAD:**
 - id_imagen (PK, INT)
 - url_imagen (VARCHAR)
 - descripcion_imagen (TEXT)
 - orden (INT)
 - id_propiedad (FK a PROPIEDADES.ID_Propiedad, INT)
- **Tabla DISPONIBILIDAD:**
 - id_rango (PK INT)
 - id_propiedad (FK a PROPIEDADES.ID_Propiedad, INT)
 - tipo (ENUM 'Reserva', 'BloqueoManual', 'Promocion', 'Mantenimiento')
 - fecha_inicio (DATETIME)
 - fecha_fin (DATETIME)
 - precio_override (FLOAT)
- **Área de Gestión de Reservas y Pagos:**
 - **Tabla RESERVA:**
 - id_reserva (PK, INT)
 - fecha_inicio (DATE)
 - fecha_fin (DATE)
 - numero_huespedes (INT)
 - precio_noche (DECIMAL)
 - monto_total (DECIMAL)
 - estado_reserva (ENUM 'Pendiente', 'Confirmada', 'Cancelada', 'Completada')
 - fecha_solicitud (DATETIME)
 - fecha_aceptacion (DATETIME)
 - fecha_checkin (DATETIME)

- fecha_checkout (DATETIME)
- id_arrendatario (FK a ARRENDATARIOS.ID_Arrendatario, INT)
- id_propiedad (FK a PROPIEDADES.ID_Propiedad, INT)
- **Tabla PAGO:**
 - id_pago (PK, INT)
 - monto (DECIMAL)
 - fecha_pago (DATETIME)
 - estado_pago (ENUM 'Pendiente', 'Completado', 'Fallido')
 - id_reserva (FK a RESERVA.ID_Reserva, INT)
 - id_metodo_pago (FK a METODO_PAGO.id_metodo_pago, INT)
- **Tabla DETALLE_PAGO:**
 - id_detalle (PK, INT)
 - pago_total (DECIMAL)
 - pago_arrendador (DECIMAL)
 - pago_comision (DECIMAL)
 - pago_igv (DECIMAL)
 - id_pago (FK a PAGO.id_pago, INT)
- **Tabla METODO_PAGO:**
 - id_metodo_pago (PK, INT)
 - id_tipo_pago (INT)
 - numero_cuenta (INT)
 - fecha_vencimiento (DATETIME)
 - proveedor (VARCHAR)
 - es_predeterminado (BOOLEAN)
 - id_usuario (FK a USUARIO.id_usuario, INT)
- **Tabla TIPO_PAGO:**
 - id_tipo_pago (PK, INT)

- nombre (VARCHAR)
- **Tabla MONEDAS:**
 - codigo_moneda (PK, CHAR(3))
 - nombre (VARCHAR)
 - decimales (INT)
 - fecha_creacion (DATETIME)
- **Tabla COTIZACIONES_DIVISAS:**
 - id_cotizacion (PK, INT, autoincremental)
 - moneda_base (FK → MONEDAS.codigo_moneda)
 - moneda_cotizada (FK → MONEDAS.codigo_moneda)
 - tipo_cambio (DECIMAL)
 - fuente (VARCHAR)
 - fecha_cotizacion (DATETIME)
 - fecha_creacion (DATETIME)
- **Área de Feedback, Calificación y Mensajería:**
 - **Tabla CONVERSACIÓN:**
 - id_conversacion (PK, INT)
 - id_propiedad (FK a PROPIEDAD.ID_Propiedad, INT)
 - id_arrendatario (FK a ARRENDATARIO.ID_Arrendatario, INT)
 - fecha_creada (DATETIME)
 - fecha_cierre (DATETIME)
 - estado (ENUM 'abierta','cerrada')
 - **Tabla CONVERSACION_PARTICIPANTE:**
 - id_conversacion (FK a CONVERSACION.ID_Conversacion, INT)
 - id_usuario (FK a USUARIOS.ID_Usuario, INT)
 - rol (ENUM 'anfitrión','arrendatario')
 - (PK compuesta por id_conversacion, id_usuario)

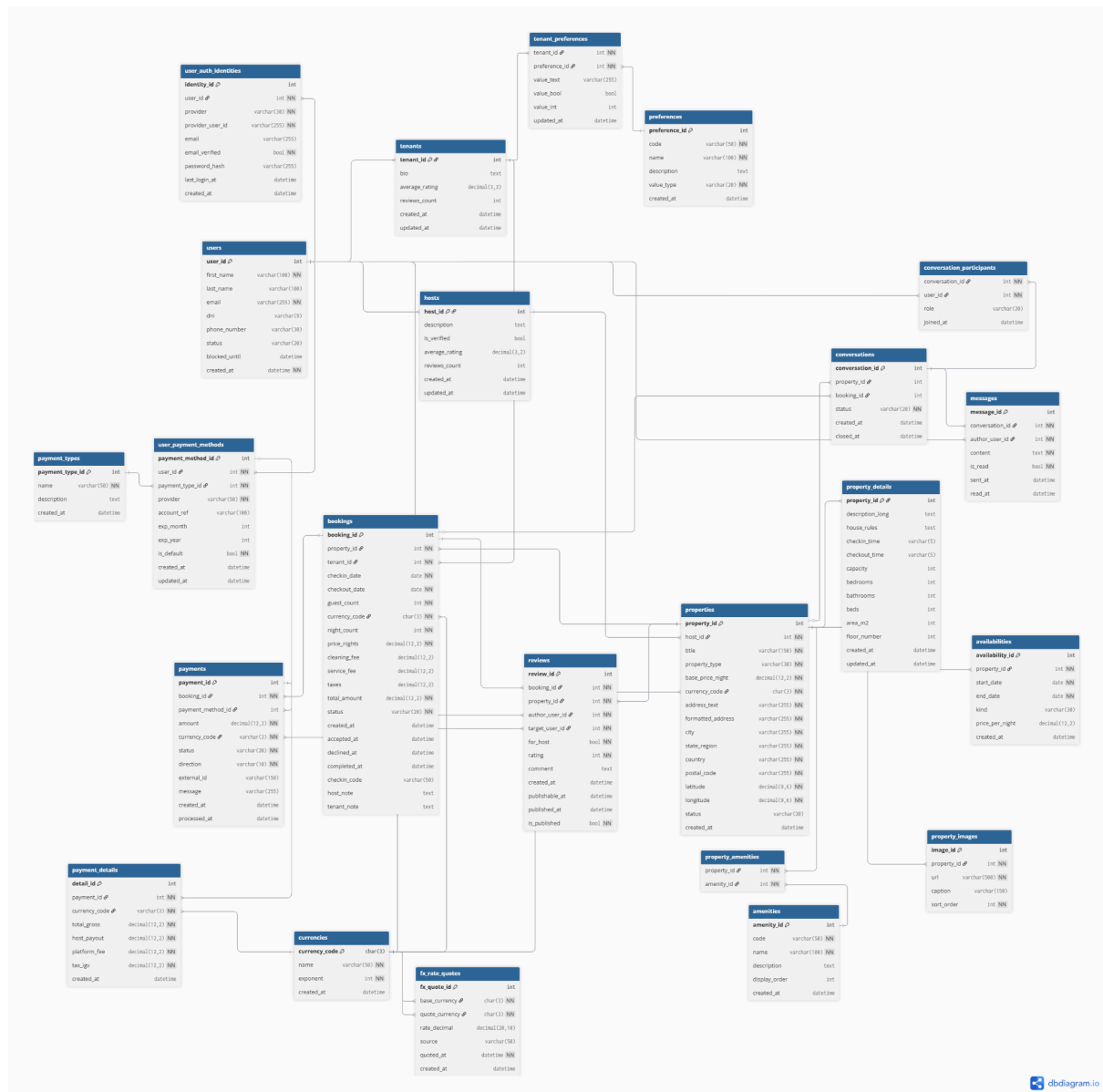
○ **Tabla MENSAJE:**

- id_mensaje (PK, INT)
- id_conversacion (FK a CONVERSACION.ID_Conversacion, INT)
- id_autor_usuario (FK a USUARIOS.ID_Usuario, INT)
- contenido (TEXT)
- fecha_envio (DATETIME)

○ **Tabla RESEÑAS:**

- id_resenia (PK, INT)
- puntuacion (INT, CHECK entre 1 y 5)
- Comentario (TEXT)
- Fecha_Creada (DATETIME)
- Fecha_Publicada (DATETIME)
- Fecha_Publicable (DATETIME)
- para_arrendatario (BOOL)
- id_reserva (FK a RESERVA.id_reserva, INT)
- id_autor_usuario (FK a USUARIO.id_usuario, INT)
- id_receptor_usuario (FK a USUARIO.id_usuario, INT)
- id_propiedad (FK a PROPIEDADES.ID_Propiedad, INT)
- Publicada (BOOL)
- (Unique Constraint: id_autor_usuario,id_receptor_usuario - para una reseña por propiedad por arrendatario).

Diagrama del Modelo de Datos Lógico:



Para una mejor visualización, ver en:

ModeloLogicoBD.pdf

Este modelo lógico servirá como base para el diseño de la base de datos relacional, permitiendo la integración y consistencia de la información de todas las áreas de dominio del negocio.