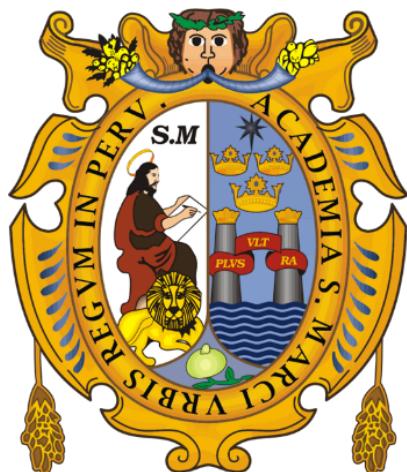


# **UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS**

Facultad de Ingeniería de Sistemas e Informática

## **Escuela Académica Profesional Ingeniería de Software**



**“Entrega parcial 2”**

### **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**

## **Informe de Desempeño del Equipo**

Durante el desarrollo de la Entrega Parcial 2, todos los integrantes del equipo: Bayona Vera, Elizabeth Ashley; Giron Altamirano, Miguel Alejandro; Melendez Cava, Andre Ivan; Padilla Arellano, Alejandro Manuel; y Torres Lezama, Mathias James, han demostrado un alto nivel de compromiso y responsabilidad. La participación ha sido activa y constante por parte de todos los miembros, cumpliendo satisfactoriamente con las tareas asignadas en los plazos establecidos y contribuyendo de manera equitativa a los objetivos de esta fase del proyecto.

## **Modelo de Datos Lógico (Versión Corregida)**

### **A. Introducción**

Esta sección presenta la versión corregida del Modelo de Datos Lógico (MLD). Su propósito es representar la estructura de la información del negocio de forma normalizada, detallando las entidades, sus atributos y las relaciones que existen entre ellas, de manera independiente del motor de base de datos.

El modelo ha sido diseñado para dar soporte a una plataforma de alquiler de alojamientos, conectando anfitriones (**hosts**) con usuarios (**users**). La estructura es capaz de gestionar todo el ciclo de vida de una reserva, desde la búsqueda de propiedades hasta la realización de pagos y la publicación de reseñas.

### **B. Áreas de Información**

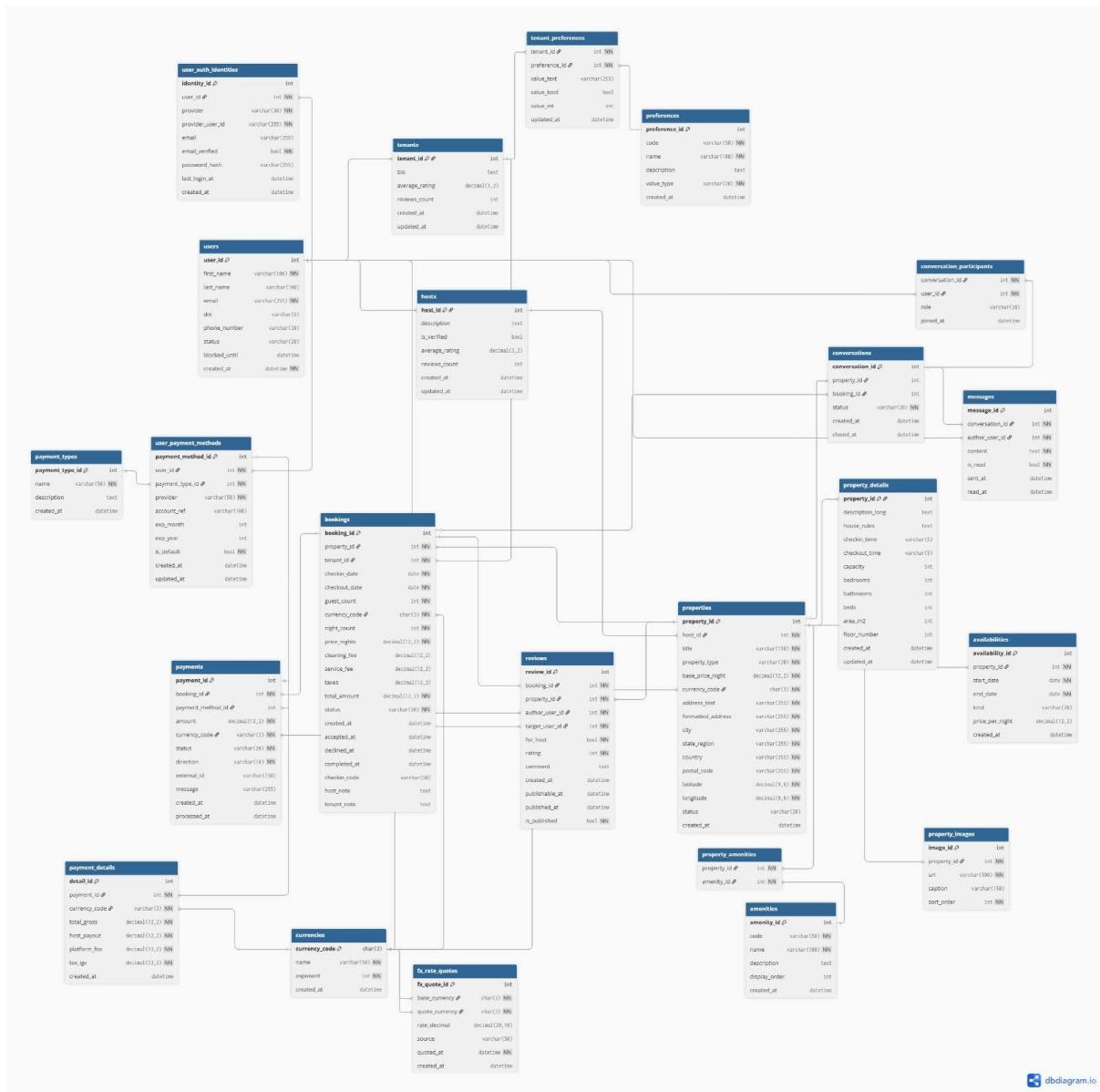
Para facilitar su comprensión, el modelo se ha organizado en las siguientes áreas de información principales:

- **Gestión de Usuarios y Perfiles:** Es el núcleo del sistema. Incluye la entidad **users** (huéspedes) y **hosts** (anfitriones). Se vincula con **user\_auth\_identities** para la autenticación y con **tenant\_preferences** y **host\_preferences** para almacenar las configuraciones de cada rol.

- **Gestión de Propiedades:** Centrada en la entidad properties, esta área describe los alojamientos. Incluye entidades de detalle como property\_details (número de camas, baños, etc.), property\_images (galería de fotos), amenities (servicios como Wi-Fi, piscina) y availabilities (fechas y precios disponibles).
- **Gestión de Reservas (Bookings):** El área transaccional principal. La entidad bookings conecta a un user con una property para un rango de fechas específico (check\_in\_date, check\_out\_date) y almacena el estado de la reserva.
- **Gestión de Pagos y Moneda:** Esta área da soporte a los requisitos de múltiples monedas. Se centra en la entidad payments (vinculada a un booking) y se detalla en payment\_details. Incluye user\_payment\_methods para los métodos de pago guardados y un módulo de divisas con currencies y fx\_rate\_quotes (cotizaciones de tipos de cambio).
- **Comunicación y Reseñas:** Modela la interacción entre usuarios. Incluye conversations (mensajes entre users y hosts sobre una property) y reviews (reseñas que un user deja para una property después de una estadía).

A continuación, se presenta el diagrama del Modelo de Datos Lógico (MLD) elaborado en diagram.io.

Este diagrama ilustra la estructura de las entidades del negocio, sus atributos con tipos de datos lógicos y las relaciones que existen entre ellas, representando la información y las reglas de negocio de la organización. Es la base conceptual sobre la cual se construye el modelo físico. Para una mejor visualización y análisis detallado del diagrama, se puede consultar el archivo de imagen en alta resolución en el repositorio de GitHub del proyecto.



## Modelo de Datos Físico

#### **A. Introducción al Modelo Físico**

El Modelo Físico de Datos (MFD) es la traducción del modelo lógico a una implementación específica para el SGBD Oracle Database. Este modelo no solo define la estructura, sino que también optimiza el rendimiento y la integridad de los datos.

En esta transición, se han aplicado las convenciones de nomenclatura definidas por la consultora y se han definido los tipos de datos físicos de Oracle, índices, secuencias y restricciones de integridad.

## B. Convenciones de Nomenclatura

Se ha adoptado una política de nomenclatura estricta y profesional, manteniendo todo el código y los objetos en idioma inglés, tal como se especificó en los requisitos del proyecto.

Las convenciones principales son:

- Tablas: Nombres en plural, mayúsculas (ej. USERS, PROPERTIES).
- Columnas: Un prefijo de 3-4 letras de la tabla, guion bajo y el nombre del atributo (ej. USR\_FIRST\_NAME, PROP\_TITLE).
- Claves Primarias (PK): Prefijo PK\_ seguido del nombre de la tabla (ej. PK\_USERS).
- Claves Foráneas (FK): Prefijo FK\_ seguido de la tabla origen y la tabla destino (ej. FK\_BOOKINGS\_USERS).
- Índices: Prefijo IDX\_ seguido de la tabla y columnas (ej. IDX\_USERS\_EMAIL).
- Secuencias: Prefijo SEQ\_ seguido del nombre de la tabla (ej. SEQ\_USERS).

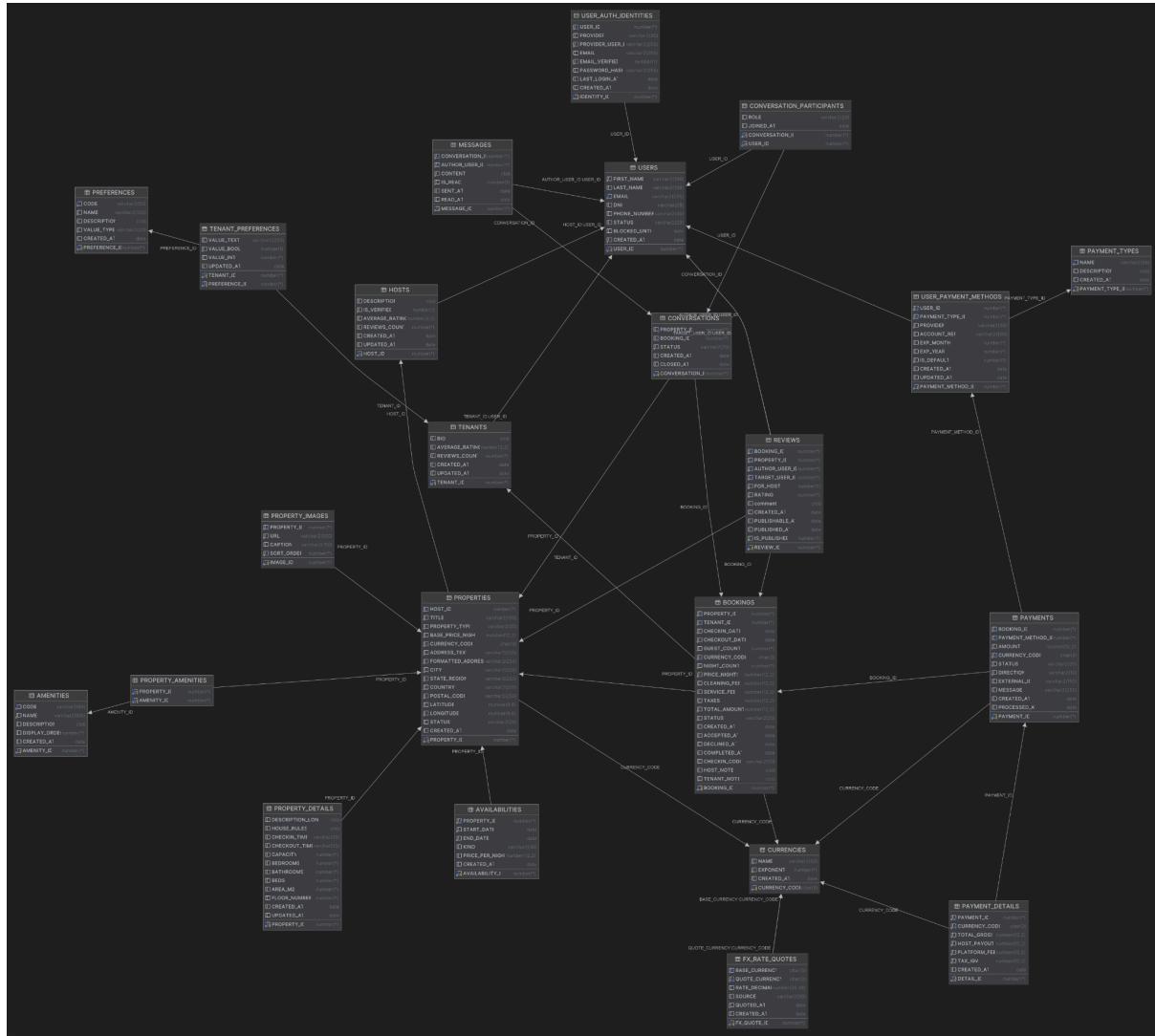
## C. Definiciones Físicas (Implementación en Oracle)

El MFD incluye las siguientes definiciones técnicas:

- Tipos de Datos Específicos: Se han ajustado los tipos de datos lógicos a los tipos nativos de Oracle. Por ejemplo:
  - varchar se convierte en VARCHAR2(n).
  - int se convierte en NUMBER(p, s).
  - datetime se convierte en TIMESTAMP o DATE.
  - bool se implementa como NUMBER(1) con una restricción CHECK (columna IN (0, 1)).

- text se convierte en CLOB para almacenar grandes volúmenes de texto (como descripciones o mensajes).
- Índices: Se han creado índices (B-Tree por defecto) en todas las claves foráneas para optimizar las operaciones de JOIN. Además, se han añadido índices en columnas de alta frecuencia de búsqueda, como USERS(EMAIL) o PROPERTIES STATUS), para mejorar el rendimiento de las consultas.
- Secuencias y Disparadores (Triggers): Para manejar las claves primarias autoincrementales (ej. users\_id), se ha creado una SEQUENCE de Oracle para cada tabla (SEQ\_USERS). Se utiliza un disparador (trigger) BEFORE INSERT para asignar automáticamente el siguiente valor de la secuencia a la columna ID en cada nueva inserción.
- Integridad de Datos: Se han implementado todas las restricciones de integridad:
  - PRIMARY KEY para garantizar la unicidad de las filas.
  - FOREIGN KEY con la cláusula ON DELETE (ej. RESTRICT o SET NULL) para mantener la integridad referencial.
  - NOT NULL en columnas obligatorias.
  - UNIQUE para campos que no pueden repetirse (ej. USERS(EMAIL)).
  - CHECK para validar reglas de negocio a nivel de base de datos (ej. BOOKINGS(CHECK\_OUT\_DATE > CHECK\_IN\_DATE)).

A continuación, se presenta el diagrama del Modelo Físico de Datos (MFD). Este diagrama ilustra la estructura final de las tablas implementadas en Oracle Database, sus columnas con tipos de datos específicos, las relaciones (claves foráneas), índices y secuencias, aplicando las convenciones de nomenclatura definidas. Representa la traducción técnica del modelo lógico. Para una mejor visualización y análisis detallado del diagrama, se puede consultar el archivo de imagen en alta resolución en el repositorio de GitHub del proyecto.



## Esquema de Base de Datos (Oracle Database)

El esquema de la base de datos ha sido completamente implementado en Oracle Database. Todos los archivos .sql necesarios para su creación, carga de datos y definición de lógica de negocio se encuentran gestionados y versionados. Se ha centralizado todo el código en el repositorio del proyecto en GitHub, donde se pueden encontrar los scripts de generación de esquemas, creación de objetos (DDL), carga de datos ficticios (DML) y los objetos de programación almacenados (PL/SQL).