



Curso:

Curso Integrador I: Sistemas Software

SECCIÓN:

46870

Tarea:

(AC-S012-APF3) Avance de Proyecto Final III

DOCENTE:

Jaime Jose SAENZ DEDIOS

INTEGRANTE Y CÓDIGOS:

Vela Bardales Robinson U21213497

Palomino Luis Eleazar - U20220894 (NO PARTICIPO)

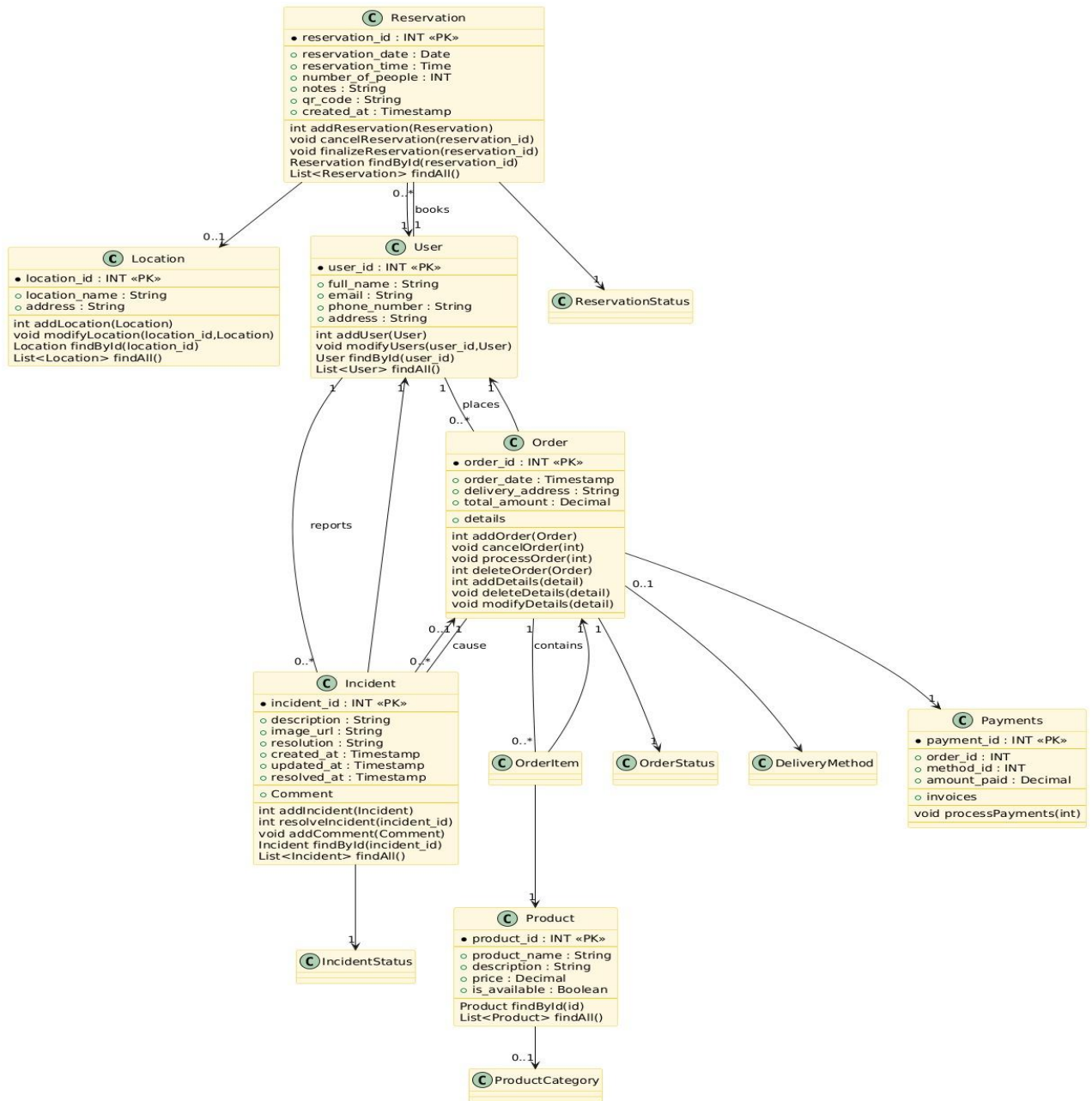
Cruz Mauricio Douglas johao - U21221659

Choccelahua Poma, Nimecio - U21306415

Fonseca Flores Luis Antonio -U21202913

PERÚ-2025

DIAGRAMA DE CLASES



1. Location

a. Atributos

- i. location_id: clave primaria.
- ii. location_name, address: nombre y dirección de la sala o local.

b. Métodos

- i. addLocation(Location loc) → int
Crea un nuevo registro de ubicación en la base de datos y retorna su location_id.
- ii. modifyLocation(location_id, Location loc) → void
Actualiza nombre y/o dirección de la ubicación identificada por location_id.
- iii. findById(location_id) → Location
Devuelve la instancia de Location cuyo location_id coincida.
- iv. findAll() → List<Location>
Obtiene todas las ubicaciones registradas.

2. User

a. Atributos

Información de contacto y dirección del cliente.

b. Métodos

- i. addUser(User u) → int
Inserta un nuevo usuario y devuelve su user_id.
- ii. modifyUsers(user_id, User u) → void
Actualiza los datos (full_name, email, etc.) del usuario dado.
- iii. findById(user_id) → User
Recupera un usuario concreto por su clave.
- iv. findAll() → List<User>
Lista todos los usuarios registrados.

3. Order

a. Atributos

Fecha del pedido, dirección de envío y monto total calculado.

b. Métodos

- i. addOrder(Order o) → int
Registra un nuevo pedido (estado "Pendiente") y retorna su order_id.
- ii. cancelOrder(order_id) → void
Cambia el estado del pedido a "Cancelado" y detiene su procesamiento.
- iii. processOrder(order_id) → void
Avanza el estado (por ejemplo de "Pending" a "Shipped"), lanza avisos/notificaciones.
- iv. deleteOrder(Order o) → int
Borra físicamente la orden (usualmente sólo usada en pruebas).
- v. addDetails(detail) → int
Agrega un ítem (OrderItem) al pedido y devuelve su identificador.
- vi. deleteDetails(detail) → void
Elimina un ítem concreto del pedido.
- vii. modifyDetails(detail) → void
Actualiza cantidad o instrucciones de un ítem existente.

4. Payments

a. **Atributos**

Relación con Order, método de pago y monto cobrado.

b. **Métodos**

- i. processPayments(payment_id) → void

Valida y confirma la transacción ante la pasarela, actualiza estado, genera facturas.

5. Product

a. Atributos

Nombre, descripción, precio y disponibilidad para pedidos.

b. Métodos

- i. findById(id) → Product

ii. Obtiene un producto según su identificador.

- iii. findAll() → List<Product>

- iv. Lista todos los productos (o filtra por is_available = true)

6. Incident

a. **Atributos**

Detalle de la queja, enlaces a imágenes, fecha/hora de creación y resolución.

b. **Métodos**

- i. addIncident(Incident inc) → int

Registra una nueva incidencia y devuelve su incident_id.

- ii. resolveIncident(incident_id) → int

Cambia estado a "Resuelto" y rellena resolved_at.

- iii. addComment(Comment c) → void

Inserta un comentario de seguimiento (usuario o staff).

- iv. findById(incident_id) → Incident

Recupera la incidencia.

- v. findAll() → List<Incident>

Lista incidencias (puede filtrar por estado).

7. Reservation

a. **Atributos**

Fecha, hora, número de comensales, notas especiales y QR de validación.

b. **Métodos**

- i. addReservation(Reservation r) → int

Crea la reserva (estado "Confirmed") y devuelve su ID.

- ii. cancelReservation(reservation_id) → void

Anula la reserva y notifica al usuario.

- iii. finalizeReservation(reservation_id) → void

Marca la reserva como "Attended" al consumirse en local.

- iv. findById(reservation_id) → Reservation

Obtiene los datos de una reserva concreta.

- v. findAll() → List<Reservation>

Lista todas las reservas (o por usuario/fecha).