

Documentación “Siste Turismo”

Models:

Cliente.java:

```
package models;
```

```
/**
 * La clase {@code Cliente} representa a un cliente dentro del sistema.
 * Un cliente tiene atributos básicos como su nombre, dirección de correo electrónico,
 * número de teléfono y un medio de pago opcional.
 *
 * <p>Esta clase proporciona métodos para acceder a la información del cliente,
 * como su nombre, email y teléfono. El atributo {@code medioPago} está presente,
 * aunque no se usa directamente en el constructor actual.
 *
 * <p>Ejemplo de uso:
 * <pre>
 *     Cliente cliente = new Cliente("Juan Pérez", "juan@example.com", "+1234567890");
 *     System.out.println(cliente.getNombre()); // Imprime: Juan Pérez
 * </pre>
 *
 * @author erick
 */
public class Cliente {

    /** El nombre del cliente */
    private String nombre;

    /** El identificador único del cliente (no utilizado en el constructor actual) */
    private String idCliente;

    /** El correo electrónico del cliente */
    private String email;

    /** El número de teléfono del cliente */
    private String telefono;

    /** El medio de pago del cliente (no inicializado en el constructor actual) */
    private Pago medioPago;

    /**
     * Constructor de la clase {@code Cliente}.
     * Crea un cliente con el nombre, correo electrónico y número de teléfono proporcionados.
     */
}
```

```

*
* @param nombre El nombre del cliente.
* @param email El correo electrónico del cliente.
* @param telefono El número de teléfono del cliente.
*/
public Cliente(String nombre, String email, String telefono){
    this.nombre = nombre;
    this.email = email;
    this.telefono = telefono;
}

/**
 * Obtiene el nombre del cliente.
 *
 * @return El nombre del cliente.
 */
public String getNombre() {
    return nombre;
}

/**
 * Obtiene el correo electrónico del cliente.
 *
 * @return El correo electrónico del cliente.
 */
public String getEmail() {
    return email;
}

/**
 * Obtiene el número de teléfono del cliente.
 *
 * @return El número de teléfono del cliente.
 */
public String getTelefono() {
    return telefono;
}
}

```

Habitacion.java:

```
package models;
```

```

/**
 * La clase {@code Habitacion} representa una habitación dentro de un hotel u hospedaje.
 * Cada habitación tiene un número único, un tipo, un precio por noche y un estado

```

```

* de disponibilidad.
*
* <p>Esta clase modela la información básica de una habitación, que incluye su número
* de habitación, tipo de habitación, precio por noche y si está disponible o no para ser
reservada.</p>
*
* <p>Ejemplo de uso:
* <pre>
*   Habitacion habitacion = new Habitacion("101", "Doble", 100, true);
*   System.out.println(habitacion.getNumeroHabitacion()); // Imprime: 101
* </pre>
* </p>
*
* @author erick
*/
public class Habitacion {

    /** El número de la habitación, que debe ser único dentro del hotel */
    private String numeroHabitacion;

    /** El tipo de habitación (e.g., "Individual", "Doble", "Suite") */
    private String tipo;

    /** El precio por noche de la habitación en la moneda local */
    private int precioNoche;

    /** Indica si la habitación está disponible para ser reservada */
    private boolean disponible;

    /**
     * Constructor de la clase {@code Habitacion}.
     * Inicializa una nueva habitación con los atributos especificados.
     *
     * @param numeroHabitacion El número único de la habitación.
     * @param tipo El tipo de la habitación (e.g., "Doble", "Suite").
     * @param precioNoche El costo por noche de la habitación.
     * @param disponible El estado de disponibilidad de la habitación.
     */
    public Habitacion(String numeroHabitacion, String tipo, int precioNoche, boolean
disponible) {
        this.numeroHabitacion = numeroHabitacion;
        this.tipo = tipo;
        this.precioNoche = precioNoche;
        this.disponible = disponible;
    }

    /**
     * Obtiene el número de la habitación.

```

```

*
* @return El número de la habitación.
*/
public String getNumeroHabitacion() {
    return numeroHabitacion;
}

/**
* Obtiene el tipo de habitación (e.g., "Doble", "Suite").
*
* @return El tipo de la habitación.
*/
public String getTipo() {
    return tipo;
}

/**
* Obtiene el precio por noche de la habitación.
*
* @return El precio por noche de la habitación.
*/
public int getPrecioNoche() {
    return precioNoche;
}

/**
* Verifica si la habitación está disponible.
*
* @return {@code true} si la habitación está disponible; {@code false} en caso contrario.
*/
public boolean isDisponible() {
    return disponible;
}

/**
* Establece si la habitación está disponible o no.
*
* @param disponible El nuevo estado de disponibilidad de la habitación.
*/
public void setDisponible(boolean disponible) {
    this.disponible = disponible;
}
}

```

Hotel.java:

```

package models;

import java.util.List;

/**
 * La clase {@code Hotel} representa un hotel que contiene listas de clientes,
 * reservas, trabajadores y habitaciones.
 *
 * <p>Esta clase gestiona los principales componentes de un hotel: los clientes
 * registrados, las reservas realizadas, los trabajadores empleados y las habitaciones
 * disponibles.</p>
 *
 * <p>Los componentes del hotel están organizados en listas, lo que permite acceder
 * y manipular múltiples entidades de manera eficiente.</p>
 *
 * <p>Ejemplo de uso:
 * <pre>
 *   Hotel hotel = new Hotel();
 *   List<Cliente> listaClientes = hotel.getClientes();
 *   // Realizar operaciones con las listas.
 * </pre>
 * </p>
 *
 * @author erick
 */
public class Hotel {

    /** Lista de clientes registrados en el hotel */
    private List<Cliente> clientes;

    /** Lista de reservas realizadas en el hotel */
    private List<Reserva> reservas;

    /** Lista de trabajadores que laboran en el hotel */
    private List<Trabajador> trabajadores;

    /** Lista de habitaciones disponibles en el hotel */
    private List<Habitacion> habitaciones;

    /**
     * Obtiene la lista de clientes registrados en el hotel.
     *
     * @return La lista de clientes.
     */
    public List<Cliente> getClientes() {
        return clientes;
    }
}

```

```

/**
 * Establece la lista de clientes registrados en el hotel.
 *
 * @param clientes La nueva lista de clientes.
 */
public void setClientes(List<Cliente> clientes) {
    this.clientes = clientes;
}

/**
 * Obtiene la lista de reservas realizadas en el hotel.
 *
 * @return La lista de reservas.
 */
public List<Reserva> getReservas() {
    return reservas;
}

/**
 * Establece la lista de reservas realizadas en el hotel.
 *
 * @param reservas La nueva lista de reservas.
 */
public void setReservas(List<Reserva> reservas) {
    this.reservas = reservas;
}

/**
 * Obtiene la lista de trabajadores del hotel.
 *
 * @return La lista de trabajadores.
 */
public List<Trabajador> getTrabajadores() {
    return trabajadores;
}

/**
 * Establece la lista de trabajadores del hotel.
 *
 * @param trabajadores La nueva lista de trabajadores.
 */
public void setTrabajadores(List<Trabajador> trabajadores) {
    this.trabajadores = trabajadores;
}

/**
 * Obtiene la lista de habitaciones disponibles en el hotel.
 *

```

```

    * @return La lista de habitaciones.
    */
    public List<Habitacion> getHabitaciones() {
        return habitaciones;
    }

    /**
     * Establece la lista de habitaciones disponibles en el hotel.
     *
     * @param habitaciones La nueva lista de habitaciones.
     */
    public void setHabitaciones(List<Habitacion> habitaciones) {
        this.habitaciones = habitaciones;
    }
}

```

Pago.java:

```

package models;

import java.text.DateFormat;

/**
 * La clase {@code Pago} representa la información relacionada con un pago realizado
 * por un cliente en el sistema de un hotel u otro servicio.
 *
 * <p>Esta clase modela los atributos esenciales de un pago, incluyendo el tipo de pago,
 * un identificador único del pago y la fecha en que se realizó.</p>
 *
 * <p>Ejemplo de uso:
 * <pre>
 *     Pago pago = new Pago("Tarjeta de crédito", "P12345", DateFormat.getDateInstance());
 *     System.out.println(pago.getTipoPago()); // Imprime: Tarjeta de crédito
 * </pre>
 * </p>
 *
 * @author erick
 */
public class Pago {

    /** El tipo de pago (e.g., "Tarjeta de crédito", "Efectivo", "Transferencia bancaria") */
    private String tipoPago;

    /** El identificador único del pago */
    private String idPago;

```

```

/** La fecha en que se realizó el pago */
private DateFormat fechaPago;

/**
 * Constructor de la clase {@code Pago}.
 * Inicializa un nuevo pago con los atributos especificados.
 *
 * @param tipoPago El tipo de pago utilizado (e.g., "Tarjeta de crédito", "Efectivo").
 * @param idPago El identificador único del pago.
 * @param fechaPago La fecha en que se realizó el pago.
 */
public Pago(String tipoPago, String idPago, DateFormat fechaPago) {
    this.tipoPago = tipoPago;
    this.idPago = idPago;
    this.fechaPago = fechaPago;
}

/**
 * Obtiene el tipo de pago utilizado (e.g., "Tarjeta de crédito", "Efectivo").
 *
 * @return El tipo de pago.
 */
public String getTipoPago() {
    return tipoPago;
}

/**
 * Establece el tipo de pago.
 *
 * @param tipoPago El nuevo tipo de pago.
 */
public void setTipoPago(String tipoPago) {
    this.tipoPago = tipoPago;
}

/**
 * Obtiene el identificador único del pago.
 *
 * @return El identificador del pago.
 */
public String getIdPago() {
    return idPago;
}

/**
 * Establece el identificador único del pago.
 *

```



```

    * @param idPago El nuevo identificador del pago.
    */
    public void setIdPago(String idPago) {
        this.idPago = idPago;
    }

    /**
     * Obtiene la fecha en que se realizó el pago.
     *
     * @return La fecha del pago.
     */
    public DateFormat getFechaPago() {
        return fechaPago;
    }

    /**
     * Establece la fecha en que se realizó el pago.
     *
     * @param fechaPago La nueva fecha del pago.
     */
    public void setFechaPago(DateFormat fechaPago) {
        this.fechaPago = fechaPago;
    }
}

```

Reserva.java:

```

package models;

import java.text.DateFormat;

/**
 * La clase {@code Reserva} representa una reserva de una habitación en un hotel.
 *
 * <p>Esta clase modela la información esencial relacionada con una reserva, incluyendo
 * el identificador de la reserva, las fechas de inicio y fin, el estado de la reserva,
 * la habitación asignada y el cliente que realizó la reserva.</p>
 *
 * <p>Ejemplo de uso:
 * <pre>
 *   Reserva reserva = new Reserva("R12345", DateFormat.getDateInstance(),
 *   DateFormat.getDateInstance(), "Confirmada");
 *   System.out.println(reserva.getEstado()); // Imprime: Confirmada
 * </pre>
 * </p>
 */

```

```

* @author erick
*/
public class Reserva {

    /** Identificador único de la reserva */
    private String idReserva;

    /** Fecha de inicio de la reserva */
    private DateFormat fechaInicio;

    /** Fecha de finalización de la reserva */
    private DateFormat fechaFinal;

    /** Estado de la reserva (e.g., "Confirmada", "Cancelada", "Pendiente") */
    private String estado;

    /** Habitación asociada a la reserva */
    private Habitacion habitacion;

    /** Cliente que realizó la reserva */
    private Cliente cliente;

    /**
     * Constructor para la clase {@code Reserva}.
     *
     * @param idReserva El identificador único de la reserva.
     * @param fechaInicio La fecha de inicio de la reserva.
     * @param fechaFinal La fecha de finalización de la reserva.
     * @param estado El estado actual de la reserva.
     * @param habitacion La habitación asignada a la reserva.
     * @param cliente El cliente que realizó la reserva.
     */
    public Reserva(String idReserva, DateFormat fechaInicio, DateFormat fechaFinal, String
estado, Habitacion habitacion, Cliente cliente) {
        this.idReserva = idReserva;
        this.fechaInicio = fechaInicio;
        this.fechaFinal = fechaFinal;
        this.estado = estado;
        this.habitacion = habitacion;
        this.cliente = cliente;
    }

    /**
     * Obtiene el identificador único de la reserva.
     *
     * @return El identificador de la reserva.
     */
    public String getIdReserva() {

```

```

        return idReserva;
    }

    /**
     * Establece el identificador único de la reserva.
     *
     * @param idReserva El nuevo identificador de la reserva.
     */
    public void setIdReserva(String idReserva) {
        this.idReserva = idReserva;
    }

    /**
     * Obtiene la fecha de inicio de la reserva.
     *
     * @return La fecha de inicio.
     */
    public DateFormat getFechaInicio() {
        return fechaInicio;
    }

    /**
     * Establece la fecha de inicio de la reserva.
     *
     * @param fechaInicio La nueva fecha de inicio.
     */
    public void setFechaInicio(DateFormat fechaInicio) {
        this.fechaInicio = fechaInicio;
    }

    /**
     * Obtiene la fecha de finalización de la reserva.
     *
     * @return La fecha final de la reserva.
     */
    public DateFormat getFechaFinal() {
        return fechaFinal;
    }

    /**
     * Establece la fecha de finalización de la reserva.
     *
     * @param fechaFinal La nueva fecha de finalización.
     */
    public void setFechaFinal(DateFormat fechaFinal) {
        this.fechaFinal = fechaFinal;
    }
}

```

```
/**
 * Obtiene el estado actual de la reserva.
 *
 * @return El estado de la reserva (e.g., "Confirmada", "Cancelada").
 */
public String getEstado() {
    return estado;
}
```

```
/**
 * Establece el estado actual de la reserva.
 *
 * @param estado El nuevo estado de la reserva.
 */
public void setEstado(String estado) {
    this.estado = estado;
}
```

```
/**
 * Obtiene la habitación asignada a la reserva.
 *
 * @return La habitación asociada a la reserva.
 */
public Habitacion getHabitacion() {
    return habitacion;
}
```

```
/**
 * Establece la habitación asignada a la reserva.
 *
 * @param habitacion La nueva habitación de la reserva.
 */
public void setHabitacion(Habitacion habitacion) {
    this.habitacion = habitacion;
}
```

```
/**
 * Obtiene el cliente que realizó la reserva.
 *
 * @return El cliente que hizo la reserva.
 */
public Cliente getCliente() {
    return cliente;
}
```

```
/**
 * Establece el cliente que realizó la reserva.
 *
```

```

    * @param cliente El nuevo cliente asociado a la reserva.
    */
    public void setCliente(Cliente cliente) {
        this.cliente = cliente;
    }
}

```

Trabajador.java:

```

package models;

```

```

/**
 * La clase {@code Trabajador} representa a un trabajador en el contexto de un hotel.
 *
 * <p>Esta clase modela la información básica relacionada con un trabajador, incluyendo
 * su nombre, identificador único y el cargo que desempeña dentro del hotel.</p>
 *
 * <p>Ejemplo de uso:
 * <pre>
 *   Trabajador trabajador = new Trabajador("Erick", "T001", "Recepcionista");
 *   System.out.println(trabajador.getNombre()); // Imprime: Erick
 * </pre>
 * </p>
 *
 * @author erick
 */
public class Trabajador {

    /** Nombre del trabajador */
    private String nombre;

    /** Identificador único del trabajador */
    private String idTrabajador;

    /** Cargo del trabajador dentro del hotel */
    private String cargo;

    /**
     * Constructor para la clase {@code Trabajador}.
     *
     * @param nombre El nombre del trabajador.
     * @param idTrabajador El identificador único del trabajador.
     * @param cargo El cargo que desempeña el trabajador.
     */
    public Trabajador(String nombre, String idTrabajador, String cargo) {

```

```

        this.nombre = nombre;
        this.idTrabajador = idTrabajador;
        this.cargo = cargo;
    }

    /**
     * Obtiene el nombre del trabajador.
     *
     * @return El nombre del trabajador.
     */
    public String getNombre() {
        return nombre;
    }

    /**
     * Establece el nombre del trabajador.
     *
     * @param nombre El nuevo nombre del trabajador.
     */
    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    /**
     * Obtiene el identificador único del trabajador.
     *
     * @return El identificador del trabajador.
     */
    public String getIdTrabajador() {
        return idTrabajador;
    }

    /**
     * Establece el identificador único del trabajador.
     *
     * @param idTrabajador El nuevo identificador del trabajador.
     */
    public void setIdTrabajador(String idTrabajador) {
        this.idTrabajador = idTrabajador;
    }

    /**
     * Obtiene el cargo del trabajador.
     *
     * @return El cargo que desempeña el trabajador.
     */
    public String getCargo() {
        return cargo;
    }

```

```
}

/**
 * Establece el cargo del trabajador.
 *
 * @param cargo El nuevo cargo del trabajador.
 */
public void setCargo(String cargo) {
    this.cargo = cargo;
}
}
```