

Juan Diego García Chavarriaga

C.C. 1152446888

# FINCAS LOTUS

## REQUERIMIENTO DEL NEGOCIO

El presente proyecto consiste en la creación de una empresa de hospedaje especializada en ofrecer experiencias de descanso y bienestar en un entorno con temática oriental budista. El negocio cuenta con **cuatro fincas** exclusivas, cada una diseñada con un estilo y ambiente únicos que proporcionan un espacio de tranquilidad y relajación a los huéspedes. Las fincas tienen características diferenciadas, pero todas comparten la misma misión: ofrecer un refugio ideal para retiros espirituales, vacaciones familiares y eventos especiales. Las fincas disponibles son:

### 1. Finca Flower

- **Capacidad:** Hasta 10 personas
- **Características:** Piscina privada, sala de meditación, jardín zen
- **Ideal:** Para retiros de bienestar y desconexión total
- **Precio por noche:** \$350 USD

### 2. Finca Zen

- **Capacidad:** Hasta 8 personas
- **Características:** Jardín japonés, espacio para yoga, terraza panorámica
- **Ideal:** Para escapadas de meditación y talleres de yoga
- **Precio por noche:** \$300 USD

### 3. Finca Nirvana

- **Capacidad:** Hasta 12 personas
- **Características:** Caminos naturales, espacio para fogatas, zona de lectura al aire libre
- **Ideal:** Para vacaciones familiares y retiros espirituales en la naturaleza
- **Precio por noche:** \$400 USD

### 4. Finca Mandala

- **Capacidad:** Hasta 15 personas
- **Características:** Piscina, jardines en forma de mandala, áreas de masaje

- **Ideal:** Para retiros holísticos y eventos al aire libre
- **Precio por noche:** \$450 USD

## Relación entre Cliente y Finca

Los clientes tienen la posibilidad de reservar cualquiera de las cuatro fincas, pudiendo hacer múltiples reservas en diferentes fechas. Las fincas pueden ser reservadas por diferentes clientes, pero nunca en las mismas fechas, lo que permite que las reservas no se superpongan.

Del cliente se desea saber los datos personales de los clientes como id, teléfono, email, nombre, fecha de nacimiento y edad ya que solo podemos realizar contratos comerciales con mayores de edad.

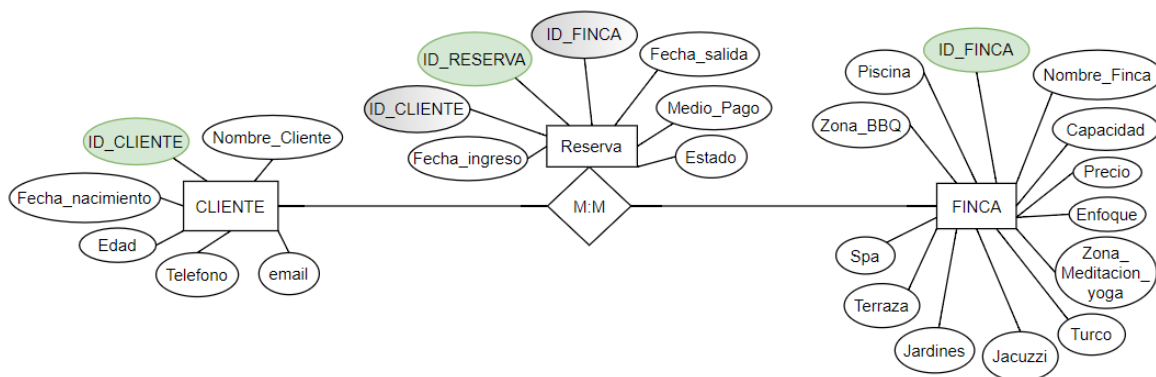
Todas las fincas cuentan con servicios básicos como agua, luz, internet, habitaciones, sala comedor, cocina, baños, sin embargo cada una cuenta con atributos particulares como piscina, zona BBQ, capacidad, precio por noche, enfoque, zona de meditación y yoga, turco, jacuzzi, jardines, terraza, spa.

Para gestionar las reservas, se ha modelado una entidad intermedia llamada **Reserva**, que permite relacionar a los clientes con las fincas en fechas específicas. Esta entidad contiene información sobre la fecha de entrada, la fecha de salida, el estado de la reserva, y medio de pago, garantizando que una finca no pueda ser ocupada por más de un cliente simultáneamente.

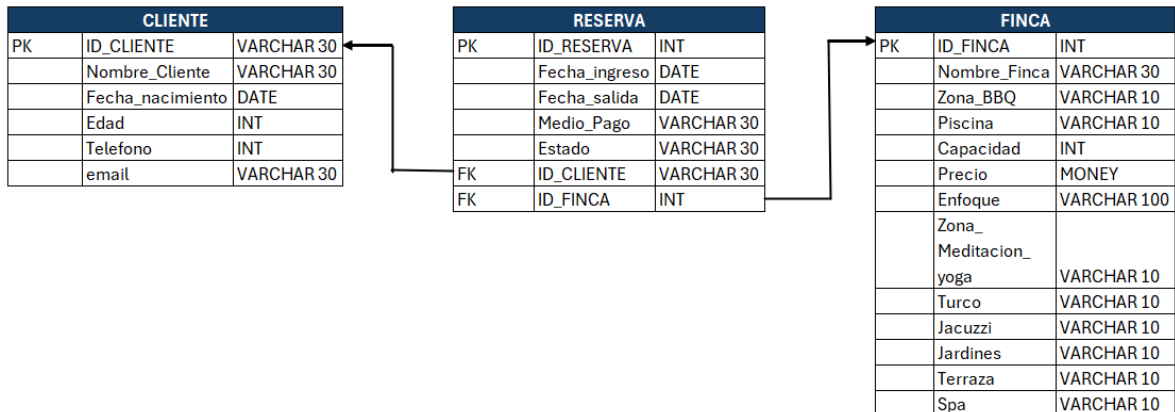
## Objetivos del Proyecto

El objetivo es implementar un sistema de gestión de hospedaje que permita a los clientes visualizar las fincas disponibles, realizar reservas, y gestionar la disponibilidad de las fincas a través de un sistema eficiente que considere las características específicas de cada finca.

## MODELO ENTIDAD RELACION



## MODELO RELACIONAL



## SQL

### Creación de la base de datos Lotus

```
create database Lotus;  
use Lotus;
```

### Creación de la tabla cliente

```
create table cliente(  
  ID_CLIENTE int,  
  constraint PK_ID_CLIENTE primary key (ID_CLIENTE),  
  Nombre_Cliente varchar(30) not null,  
  Fecha_nacimiento date not null,  
  Edad int not null,  
  telefono int not null,  
  email int not null  
);
```

### Modificaciones en la tabla cliente

```
ALTER TABLE cliente  
  ALTER COLUMN email VARCHAR(100) not null;  
  
alter table cliente  
  add constraint UQ_email unique(email);
```

## Creación de la tabla finca

```
create table finca(  
  ID_FINCA int,  
  constraint PK_ID_FINCA primary key(ID_FINCA),  
  Nombre_Finca varchar(30)not null unique,  
  Zona_BBQ varchar(10)not null,  
  Piscina varchar(10)not null,  
  Capacidad int not null,  
  Precio money not null,  
  Enfoque varchar(100),  
  Zona_yoga varchar(10)not null,  
  Turco varchar(10)not null,  
  Jacuzzi varchar(10)not null,  
  Jardines varchar(10)not null,  
  Terraza varchar(10)not null,  
  Spa varchar(10)not null  
);
```

## Creación de la tabla reserva

```
create table reserva(  
  ID_RESERVA int,  
  constraint PK_ID_RESERVA primary key(ID_RESERVA),  
  Fecha_ingreso date not null,  
  Fecha_salida date not null,  
  Medio_pago varchar(30)not null,  
  Estado_reserva varchar(30)not null,  
  ID_CLIENTE int,  
  constraint FK_ID_CLIENTE foreign key (ID_CLIENTE) references cliente(ID_CLIENTE) on delete cascade,  
  ID_FINCA int,  
  constraint FK_ID_FINCA foreign key (ID_FINCA) references finca(ID_FINCA) on delete cascade  
);
```

## Uso de bulk para insertar datos mediante bases de datos externas

```
bulk insert  
  cliente  
  from 'C:\Users\juang\Desktop\PROYECTO\BD_CLIENTES.txt'  
  with (firstrow = 2);  
  
bulk insert  
  reserva  
  from 'C:\Users\juang\Desktop\PROYECTO\BD_RESERVAS.txt'  
  with (firstrow = 2);
```

## Inserción de datos manualmente

```
insert into finca values (
1, 'flower', 'no', 'si', 10,350, 'Retiro de bienestar, desconexión total, reuniones familiares', 'si', 'no', 'no', 'si', 'no', 'no'
);

insert into finca values (
2, 'zen', 'no', 'no', 8,300, 'Retiro de bienestar, desconexión total, reuniones familiares', 'si', 'no', 'no', 'si', 'si', 'no'
);

insert into finca values (
3, 'nirvana', 'no', 'no', 12,400, 'Vacaciones familiares, retiros espirituales, aventuras en la naturaleza', 'si', 'no', 'no', 'si', 'si', 'no'
);

insert into finca values (
4, 'mandala', 'si', 'si', 15,450, 'Retiros holísticos, eventos al aire libre, fines de semana en familia', 'si', 'si', 'si', 'si', 'no', 'si'
);
```

## Actualización campo enfoque de la finca Zen

```
/*Actualizacion campo enfoque de la finca Zen*/

UPDATE finca
SET Enfoque = 'Escapadas de meditación, talleres de yoga, retiros de silencio'
WHERE nombre_finca = 'zen';
```

## Creación del TRIGGER

### Creacion de la tabla para registrar los cambios

```
create table bitacora_reservas (
    id_bitacora int identity(1,1) primary key,
    id_reserva int,
    accion nvarchar(50), /*Tipo de acción (INSERT, UPDATE, DELETE)*/
    detalles nvarchar(MAX), /*Descripción del cambio*/
    fecha DATETIME DEFAULT GETDATE() /*Fecha y hora del cambio*/
);
```

## Trigger para registrar los cambios al insertar sobre la tabla reserva

```
CREATE TRIGGER trigger_insert_reserva
ON reserva
AFTER INSERT
AS
BEGIN
    INSERT INTO bitacora_reservas (id_reserva, accion, detalles)
    SELECT
        ID_RESERVA,
        'INSERT',
        CONCAT(
            'Nueva reserva creada. Fecha ingreso: ', CAST(Fecha_ingreso AS NVARCHAR),
            ', Fecha salida: ', CAST(Fecha_salida AS NVARCHAR),
            ', Medio de pago: ', Medio_pago,
            ', Estado: ', Estado_reserva,
            ', Cliente: ', CAST(ID_CLIENTE AS NVARCHAR),
            ', Finca: ', CAST(ID_FINCA AS NVARCHAR)
        )
    FROM INSERTED; /*INSERTED contiene las nuevas filas agregadas a la tabla*/
END;
```

## Prueba de inserción y consulta de registros

```
INSERT INTO reserva (ID_RESERVA, Fecha_ingreso, Fecha_salida, Medio_pago, Estado_reserva, ID_CLIENTE, ID_FINCA)
VALUES (2501, '2025-03-15', '2025-03-20', 'efectivo', 'pendiente pago', 1100, 1);

select * from bitacora_reservas;
```

Trigger para registrar cambios al actualizar estado de reserva en la tabla reserva

```
CREATE TRIGGER trg_UpdateEstadoReserva
ON reserva
AFTER UPDATE
AS
BEGIN
    /* Verificamos si el estado fue modificado */
    IF UPDATE(Estado_reserva)
    BEGIN
        INSERT INTO bitacora_reservas (id_reserva, accion, detalles, fecha)
        SELECT
            i.ID_RESERVA AS id_reserva,
            'Actualización de Estado' AS accion, /*Tipo de acción*/
            CONCAT(
                'Estado antiguo: ', d.Estado_reserva, '; ',
                'Estado nuevo: ', i.Estado_reserva, '; ',
                'Fecha de ingreso: ', FORMAT(i.Fecha_ingreso, 'yyyy-MM-dd'), '; ',
                'Fecha de salida: ', FORMAT(i.Fecha_salida, 'yyyy-MM-dd'), '; ',
                'ID finca: ', i.ID_FINCA, '; ',
                'ID cliente: ', i.ID_CLIENTE, '; ',
                'Total días: ', DATEDIFF(DAY, i.Fecha_ingreso, i.Fecha_salida)
            ) AS detalles,
            GETDATE() AS fecha /*Fecha y hora del cambio*/
        FROM
            inserted i
        INNER JOIN
            deleted d ON i.ID_RESERVA = d.ID_RESERVA;
    END
END;
GO
```

Prueba de actualización estado de reserva

```
select * from reserva;
UPDATE reserva set Estado_reserva = 'pagado' where ID_RESERVA = 2501;
select * from bitacora_reservas;
```

## Proceso almacenado para cambiar estado de la reserva

```
CREATE PROCEDURE ModificarEstadoReserva
    @ID_RESERVA INT,
    @Nuevo_Estado VARCHAR(30)
AS
BEGIN
    SET NOCOUNT ON; -- Evita que se muestren mensajes de "filas afectadas"

    UPDATE reserva
    SET Estado_reserva = @Nuevo_Estado
    WHERE ID_RESERVA = @ID_RESERVA;
END;

EXEC ModificarEstadoReserva @ID_RESERVA = 2501, @Nuevo_Estado = 'pendiente pago';
```

## Proceso almacenado para consultar reserva por id reserva

```
CREATE PROCEDURE ConsultarReservaDetalles
    @ID_RESERVA INT
AS
BEGIN
    SET NOCOUNT ON; /*Evita que se muestren mensajes de "filas afectadas"*/

    SELECT
        reserva.ID_RESERVA,
        reserva.Fecha_ingreso,
        reserva.Fecha_salida,
        reserva.Estado_reserva,
        reserva.Medio_pago,
        cliente.Nombre_Cliente AS Nombre_Cliente,
        reserva.ID_FINCA,
        finca.Nombre_Finca,
        finca.Precio AS Precio_por_dia,
        DATEDIFF(DAY, reserva.Fecha_ingreso, reserva.Fecha_salida) AS Dias_reserva,
        DATEDIFF(DAY, reserva.Fecha_ingreso, reserva.Fecha_salida) * finca.Precio AS Recaudo_total
    FROM
        reserva
    INNER JOIN
        finca
    ON
        reserva.ID_FINCA = finca.ID_FINCA
    INNER JOIN
        cliente
    ON
        reserva.ID_CLIENTE = cliente.ID_CLIENTE
    WHERE
        reserva.ID_RESERVA = @ID_RESERVA;
END;

EXEC ConsultarReservaDetalles @ID_RESERVA = 2004;
```



## Proceso almacenado para consultar recaudo en un determinado periodo de tiempo

```
CREATE PROCEDURE ConsultarRecaudoPorFechaYEstado
    @Fecha_Inicio DATE,
    @Fecha_Fin DATE,
    @Estado_Reserva VARCHAR(30)
AS
BEGIN
    SET NOCOUNT ON; -- Evita que se muestren mensajes de "filas afectadas"

    SELECT
        SUM(DATEDIFF(DAY, reserva.Fecha_ingreso, reserva.Fecha_salida) * finca.Precio) AS Recaudo_total
    FROM
        reserva
    INNER JOIN
        finca
    ON
        reserva.ID_FINCA = finca.ID_FINCA
    WHERE
        reserva.Fecha_ingreso >= @Fecha_Inicio
        AND reserva.Fecha_salida <= @Fecha_Fin
        AND reserva.Estado_reserva = @Estado_Reserva;
END;

EXEC ConsultarRecaudoPorFechaYEstado
    @Fecha_Inicio = '2024-01-01',
    @Fecha_Fin = '2024-12-31',
    @Estado_Reserva = 'finalizado';
```

## Proceso almacenado para consultar reservas realizadas de un cliente a traves de su ID

```
CREATE PROCEDURE ConsultarReservasPorCliente
    @ID_CLIENTE INT
AS
BEGIN
    SET NOCOUNT ON; -- Evita que se muestren mensajes de "filas afectadas"

    SELECT
        cliente.ID_CLIENTE,
        cliente.Nombre_Cliente AS Nombre_Cliente,
        finca.Nombre_Finca,
        reserva.Fecha_ingreso,
        reserva.Fecha_salida,
        DATEDIFF(DAY, reserva.Fecha_ingreso, reserva.Fecha_salida) AS Total_Dias,
        finca.Precio AS Precio_por_dia,
        DATEDIFF(DAY, reserva.Fecha_ingreso, reserva.Fecha_salida) * finca.Precio AS Total_Recaudo
    FROM
        reserva
    INNER JOIN
        finca
    ON
        reserva.ID_FINCA = finca.ID_FINCA
    INNER JOIN
        cliente
    ON
        reserva.ID_CLIENTE = cliente.ID_CLIENTE
    WHERE
        cliente.ID_CLIENTE = @ID_CLIENTE;
END;

EXEC ConsultarReservasPorCliente @ID_CLIENTE = 1001;
```

Link pagina:

<https://diego17-cell.github.io/lotus/index.html>

link git hub:

<https://github.com/Diego17-cell/lotus.git>