

PROYECTO FINAL

CONSORCIOS

GESTIÓN Y ADMINISTRACIÓN

CODERHOUSE

Camilo Andrés Redondo / Profesor
Hugo Gonzalez / Tutor
57185 / Comisión

JUAN PABLO MELNIK
2024

Introducción	4
Objetivo General	5
Objetivos Específicos	6
Modelo de negocio	7
Análisis FODA para la Gestión de Consorcios	10
Tablas	12
Vistas	19
Funciones	22
Procedimientos	26
Triggers	30
Diagrama Entidad Relación (E-R)	33
Herramientas	34

Introducción

La gestión eficiente de consorcios y condominios es esencial para asegurar un ambiente ordenado y satisfactorio para los residentes. En este contexto, la administración de expensas, unidades, propietarios, y gastos comunes requiere un manejo preciso de la información. Este proyecto tiene como objetivo desarrollar una base de datos robusta que centralice todos los aspectos de la gestión de consorcios, permitiendo una administración más eficaz, transparente y automatizada.

Este documento detalla el diseño, desarrollo e implementación de una base de datos diseñada específicamente para gestionar las operaciones de un consorcio. Incluye la creación de tablas esenciales, procedimientos almacenados, funciones y triggers que automatizan procesos clave, así como la inserción de datos relevantes para su puesta en marcha.

Objetivo General

Desarrollar y desplegar una base de datos relacional que optimice la gestión de consorcios, facilitando la administración de unidades, propietarios, expensas, pagos y gastos comunes, y garantizando la integridad, consistencia y disponibilidad de la información.

Objetivos Específicos

1. **Diseñar y estructurar** una base de datos relacional que cubra todos los aspectos clave de la gestión de consorcios, incluyendo tablas para unidades, propietarios, administradores, expensas y pagos.
2. **Implementar procedimientos almacenados** y funciones que automaticen tareas comunes, como la actualización del fondo de reserva después de un pago y la inserción de gastos comunes.
3. **Desarrollar triggers** que mantengan la integridad referencial y la consistencia de los datos, evitando operaciones no deseadas como la eliminación de propietarios con unidades activas.
4. **Crear vistas y consultas** que permitan a los administradores del consorcio obtener información resumida y detallada de manera eficiente, facilitando la toma de decisiones.
5. **Probar y validar** la funcionalidad de la base de datos mediante la inserción de datos de ejemplo y la ejecución de procedimientos almacenados, funciones y triggers.

Modelo de negocio

1. Resumen Ejecutivo

El modelo de negocio para la gestión de consorcios se basa en proporcionar un software como servicio (SaaS) que centraliza y optimiza la administración de consorcios y condominios. El sistema ofrece una solución integral para gestionar unidades, propietarios, expensas, pagos, y gastos comunes, con el fin de mejorar la eficiencia operativa, reducir errores administrativos y aumentar la transparencia en la gestión.

El servicio está dirigido principalmente a administradores de consorcios, propietarios de edificios y empresas dedicadas a la administración de propiedades. A través de una suscripción mensual o anual, los clientes tendrán acceso a una plataforma segura y escalable que les permitirá gestionar múltiples propiedades desde cualquier dispositivo con acceso a internet.

2. Segmento de Clientes

- **Administradores de Consorcios:** Empresas o individuos responsables de la gestión de edificios y complejos habitacionales que necesitan una solución para centralizar y automatizar sus procesos.
- **Propietarios de Edificios:** Dueños de propiedades múltiples que buscan una forma eficiente de administrar sus unidades y mejorar la satisfacción de sus inquilinos.
- **Empresas de Administración de Propiedades:** Compañías especializadas en la gestión de propiedades que manejan grandes volúmenes de datos y requieren herramientas robustas para garantizar la eficiencia y la precisión.

3. Propuesta de Valor

- **Automatización de Procesos:** Reducción de la carga administrativa mediante la automatización de tareas repetitivas, como la generación de expensas, el seguimiento de pagos y la actualización de fondos de reserva.
- **Transparencia y Confianza:** Mejora de la transparencia en la gestión financiera a través de reportes claros y accesibles para todos los stakeholders, aumentando la confianza de los propietarios e inquilinos.
- **Acceso en Tiempo Real:** Permitir a los usuarios acceder a la información en tiempo real desde cualquier lugar, facilitando la toma de decisiones informadas y oportunas.
- **Escalabilidad:** Un sistema escalable que se adapta a las necesidades de consorcios pequeños y grandes, con la capacidad de manejar desde un edificio hasta una cartera completa de propiedades.

4. Canales

- **Plataforma Web:** Acceso a la aplicación a través de un navegador web, sin necesidad de instalaciones locales, permitiendo una adopción rápida y sin fricciones.
- **Aplicación Móvil:** Desarrollo de una aplicación móvil para que los administradores y propietarios puedan gestionar sus propiedades sobre la marcha.

- **Servicio al Cliente:** Soporte técnico y atención al cliente a través de chat, correo electrónico y teléfono para resolver cualquier duda o inconveniente rápidamente.

5. Relación con el Cliente

- **Soporte Personalizado:** Asistencia personalizada para nuevos usuarios, incluyendo sesiones de capacitación y un soporte continuo para asegurar el éxito en el uso de la plataforma.
- **Centro de Recursos:** Una base de conocimiento en línea, tutoriales en video y documentación detallada para que los usuarios puedan resolver dudas de forma independiente.
- **Actualizaciones Regulares:** Provisión de actualizaciones y mejoras continuas en la plataforma, basadas en las necesidades y sugerencias de los usuarios.

6. Fuentes de Ingresos

- **Suscripción Mensual/Anual:** Los clientes pagarán una tarifa de suscripción basada en el número de unidades administradas o en un plan fijo con características escalables.
- **Consultoría y Personalización:** Ingresos adicionales a través de servicios de consultoría, personalización de la plataforma para clientes específicos, y formación especializada.
- **Comisiones por Transacción:** En caso de integrar un sistema de pagos, se pueden cobrar comisiones por transacciones procesadas a través de la plataforma.

7. Recursos Clave

- **Equipo de Desarrollo de Software:** Ingenieros y desarrolladores dedicados a la creación, mantenimiento y mejora continua de la plataforma.
- **Infraestructura de TI:** Servidores y bases de datos seguros y confiables para garantizar la disponibilidad y la seguridad de los datos de los clientes.
- **Equipo de Soporte al Cliente:** Personal capacitado para ofrecer soporte técnico y atención al cliente, garantizando una experiencia de usuario fluida y satisfactoria.

8. Actividades Clave

- **Desarrollo y Mantenimiento del Software:** Continuar con el desarrollo de nuevas funcionalidades y la mejora de las existentes, garantizando que la plataforma se mantenga actualizada con las necesidades del mercado.
- **Marketing y Ventas:** Promoción de la plataforma a través de campañas de marketing digital, redes sociales, y participación en eventos del sector inmobiliario.
- **Soporte Técnico y Formación:** Provisión de soporte continuo y formación para asegurar que los usuarios obtengan el máximo beneficio del software.

9. Socios Clave

- **Proveedores de Servicios de Pago:** Integración con plataformas de pago que permitan la automatización de la gestión de pagos y expensas.

- **Empresas de Hosting:** Proveedores de infraestructura en la nube para asegurar la disponibilidad y escalabilidad del sistema.
- **Asociaciones de Propietarios:** Colaboración con asociaciones para ofrecer el software como un beneficio exclusivo para sus miembros.

10. Estructura de Costos

- **Desarrollo y Mantenimiento:** Costos asociados al desarrollo del software, incluyendo salarios de desarrolladores y costos de infraestructura.
- **Marketing y Ventas:** Gastos en campañas publicitarias, eventos y comisiones de ventas.
- **Soporte al Cliente:** Costos relacionados con la atención al cliente y la formación, incluidos los salarios del equipo de soporte.

Análisis FODA para la Gestión de Consorcios

Fortalezas

1. Automatización y Eficiencia Operativa:

- La plataforma ofrece automatización de procesos clave como la generación de expensas y la actualización de fondos, lo que reduce significativamente la carga administrativa y minimiza errores humanos.

2. Accesibilidad y Escalabilidad:

- Al ser un servicio basado en la nube (SaaS), los usuarios pueden acceder a la plataforma desde cualquier lugar y en cualquier momento. Además, el sistema está diseñado para escalar, permitiendo a los consorcios de cualquier tamaño beneficiarse de sus funcionalidades.

3. Transparencia y Confianza:

- Los reportes financieros y la disponibilidad de datos en tiempo real fomentan la transparencia en la gestión del consorcio, lo que incrementa la confianza de los propietarios y residentes en la administración.

4. Integración de Funcionalidades:

- La plataforma incluye una amplia gama de funcionalidades, desde la gestión de unidades hasta la auditoría de gastos, proporcionando una solución integral para todas las necesidades de administración de un consorcio.

Oportunidades

1. Expansión de Mercado:

- Existe un gran potencial para expandir la base de clientes a nivel regional e internacional, especialmente en mercados emergentes donde la profesionalización de la gestión de consorcios está en crecimiento.

2. Adopción de Nuevas Tecnologías:

- La incorporación de tecnologías emergentes como la inteligencia artificial y el análisis predictivo podría mejorar aún más la eficiencia operativa, ofreciendo características avanzadas como la predicción de costos de mantenimiento o la automatización de decisiones financieras.

3. Alianzas Estratégicas:

- Formar alianzas con asociaciones de propietarios, empresas de administración de propiedades y proveedores de servicios de pago puede aumentar la visibilidad de la plataforma y ofrecer un valor añadido a los clientes.

4. Ampliación de Servicios:

- Ofrecer servicios adicionales como gestión de mantenimiento, programación de eventos comunitarios y herramientas de comunicación puede diversificar las fuentes de ingresos y mejorar la retención de clientes.

Debilidades

1. Dependencia de la Infraestructura de TI:

- La plataforma depende en gran medida de la infraestructura de TI, lo que significa que cualquier fallo en los servidores o en la conectividad de la red puede afectar la disponibilidad del servicio.

2. Curva de Aprendizaje:

- Los administradores de consorcios que no están familiarizados con herramientas tecnológicas pueden encontrar una curva de aprendizaje al principio, lo que podría limitar la adopción inicial del software.

3. Costos Iniciales de Desarrollo:

- Los costos asociados al desarrollo y mantenimiento de una plataforma robusta pueden ser elevados, especialmente en las primeras etapas del proyecto.

4. Competencia en el Mercado:

- Existen otras soluciones en el mercado que ofrecen servicios similares, lo que podría dificultar la captación de clientes y requerir estrategias de diferenciación sólidas.

Amenazas

1. Ciberseguridad:

- La amenaza de ataques cibernéticos siempre está presente, y cualquier brecha en la seguridad podría comprometer la confianza de los clientes y dañar la reputación de la plataforma.

2. Regulaciones Gubernamentales:

- Cambios en las regulaciones gubernamentales relacionadas con la protección de datos y la gestión de propiedades podrían aumentar los costos operativos o requerir modificaciones en la plataforma.

3. Competencia Creciente:

- La aparición de nuevos competidores en el mercado, especialmente aquellos con recursos significativos, puede poner en riesgo la cuota de mercado de la plataforma.

4. Resistencia al Cambio:

- Algunos administradores de consorcios pueden ser reacios a adoptar nuevas tecnologías, prefiriendo métodos tradicionales de gestión, lo que podría limitar el crecimiento de la base de clientes.

Conclusión del Análisis FODA

El análisis FODA revela que la plataforma de gestión de consorcios tiene una sólida base de fortalezas y muchas oportunidades en el mercado, lo que la posiciona bien para el éxito. Sin embargo, debe ser consciente de las debilidades y amenazas, como la competencia y los riesgos de seguridad, y debe adoptar estrategias proactivas para mitigarlas. La clave será continuar innovando, expandiendo los servicios, y fortaleciendo la relación con los clientes y socios estratégicos para mantener una ventaja competitiva sostenible.

Tablas

1. Tabla: administrators

Descripción: Almacena información sobre los administradores encargados de la gestión de los consorcios.

Nombre del Campo	Tipo de Dato	Descripción
admin_id	INT	Identificador único del administrador (Clave primaria).
first_name	VARCHAR(255)	Nombre del administrador.
last_name	VARCHAR(255)	Apellido del administrador.
phone	VARCHAR(50)	Número de teléfono del administrador.
email	VARCHAR(255)	Dirección de correo electrónico del administrador.

2. Tabla: condominiums

Descripción: Registra los detalles de cada condominio administrado, incluyendo su nombre, dirección y fondo de reserva.

Nombre del Campo	Tipo de Dato	Descripción
------------------	--------------	-------------

condo_id	INT	Identificador único del condominio (Clave primaria).
condo_name	VARCHAR(255)	Nombre del condominio.
address	VARCHAR(255)	Dirección del condominio.
unit_count	INT	Número de unidades en el condominio.
creation_date	DATE	Fecha de creación del condominio.
admin_id	INT	Identificador del administrador responsable (Clave foránea).
contact_phone	VARCHAR(50)	Teléfono de contacto del condominio.
contact_email	VARCHAR(255)	Correo electrónico de contacto del condominio.
reserve_fund	DECIMAL(10,2)	Fondo de reserva disponible del condominio.
status	VARCHAR(50)	Estado actual del condominio (e.g., Activo, Inactivo).
last_meeting_date	DATE	Fecha de la última reunión del consorcio.

3. Tabla: units

Descripción: Contiene la información de las unidades dentro de los condominios, incluyendo el tipo de unidad y su propietario.

Nombre del Campo	Tipo de Dato	Descripción
unit_id	INT	Identificador único de la unidad (Clave primaria).
condo_id	INT	Identificador del condominio al que pertenece la unidad (Clave foránea).
number	VARCHAR(50)	Número de la unidad.
type	VARCHAR(50)	Tipo de unidad (e.g., Apartamento, Oficina).
square_meters	DECIMAL(10,2)	Tamaño de la unidad en metros cuadrados.
owner_id	INT	Identificador del propietario de la unidad (Clave foránea).

4. Tabla: owners

Descripción: Guarda la información de los propietarios de las unidades dentro de los condominios.

Nombre del Campo	Tipo de Dato	Descripción
owner_id	INT	Identificador único del propietario (Clave primaria).
first_name	VARCHAR(255)	Nombre del propietario.

last_name	VARCHAR(255)	Apellido del propietario.
phone	VARCHAR(50)	Número de teléfono del propietario.
email	VARCHAR(255)	Dirección de correo electrónico del propietario.

5. Tabla: expenses

Descripción: Almacena las expensas asociadas a cada unidad, incluyendo el monto y el estado de pago.

Nombre del Campo	Tipo de Dato	Descripción
expense_id	INT	Identificador único de la expensa (Clave primaria).
unit_id	INT	Identificador de la unidad a la que corresponde la expensa (Clave foránea).
month_year	DATE	Mes y año de la expensa.
total_amount	DECIMAL(10,2)	Monto total de la expensa.
payment_status	VARCHAR(50)	Estado del pago de la expensa (e.g., Pagado, Pendiente).

6. Tabla: payments

Descripción: Registra los pagos realizados por los propietarios para cubrir las expensas de sus unidades.

Nombre del Campo	Tipo de Dato	Descripción
payment_id	INT	Identificador único del pago (Clave primaria).
expense_id	INT	Identificador de la expensa relacionada con el pago (Clave foránea).
payment_date	DATE	Fecha en la que se realizó el pago.
amount	DECIMAL(10,2)	Monto pagado.
payment_method	VARCHAR(50)	Método de pago utilizado (e.g., Transferencia, Tarjeta).

7. Tabla: common_expenses

Descripción: Contiene los gastos comunes que se aplican a todo el condominio, como mantenimiento, servicios y reparaciones.

Nombre del Campo	Tipo de Dato	Descripción
expense_id	INT	Identificador único del gasto común (Clave primaria).

condo_id	INT	Identificador del condominio al que corresponde el gasto común (Clave foránea).
description	VARCHAR(255)	Descripción del gasto común.
date	DATE	Fecha en la que se registró el gasto.
amount	DECIMAL(10,2)	Monto del gasto común.
category	VARCHAR(50)	Categoría del gasto (e.g., Mantenimiento, Servicios, Reparaciones).

8. Tabla: common_expenses_audit

Descripción: Tabla de auditoría que registra los cambios realizados en los gastos comunes, permitiendo un seguimiento detallado de las modificaciones.

Nombre del Campo	Tipo de Dato	Descripción
audit_id	INT	Identificador único del registro de auditoría (Clave primaria).
expense_id	INT	Identificador del gasto común que fue modificado (Clave foránea).
condo_id	INT	Identificador del condominio asociado al gasto común (Clave foránea).

old_amount	DECIMAL(10,2)	Monto anterior del gasto común antes de la modificación.
new_amount	DECIMAL(10,2)	Monto nuevo del gasto común después de la modificación.
change_date	TIMESTAMP	Fecha y hora en que se realizó la modificación.

Vistas

Resumen

unit_details_view

Descripción: Proporciona un resumen de las unidades, incluyendo información sobre el condominio, el propietario y los detalles específicos de cada unidad.

expense_payment_status_view

Descripción: Muestra los detalles de las expensas de cada unidad, incluyendo el estado de pago y el método de pago utilizado.

upcoming_maintenance_view

Descripción: Detalla los próximos trabajos de mantenimiento programados para cada condominio, proporcionando una referencia fácil para la administración.

Ampliación

1. Vista de Detalles de Unidad

Esta vista proporciona un resumen de las unidades, incluyendo información sobre el consorcio, el propietario y detalles de la unidad.

```
CREATE VIEW unit_details AS
SELECT
    u.unit_id,
    u.number AS unit_number,
    u.type AS unit_type,
    u.square_meters,
    c.condo_name,
    o.first_name AS owner_first_name,
    o.last_name AS owner_last_name,
    o.phone AS owner_phone,
    o.email AS owner_email
FROM
    units u
JOIN
    condominiums c ON u.condo_id = c.condo_id
JOIN
    owners o ON u.owner_id = o.owner_id;
```

2. Vista de Estado de Pagos de Expensas

Esta vista muestra los detalles de las expensas de cada unidad, incluyendo el estado del pago y el método de pago.

```
CREATE VIEW expense_payment_status AS
SELECT
    e.expense_id,
    u.number AS unit_number,
    c.condo_name,
    e.month_year,
    e.total_amount,
    e.payment_status,
    p.payment_method,
    p.payment_date
FROM
    expenses e
JOIN
    payments p ON e.expense_id = p.expense_id
JOIN
    units u ON e.unit_id = u.unit_id
JOIN
    condominiums c ON u.condo_id = c.condo_id;
```

3. Vista de Mantenimiento Programado

Esta vista detalla los próximos trabajos de mantenimiento programados para cada consorcio, proporcionando una fácil referencia para la administración.

```
CREATE VIEW upcoming_maintenance AS
SELECT
    m.maintenance_id,
    c.condo_name,
    m.description AS maintenance_description,
    m.scheduled_date,
    m.provider,
    m.estimated_cost
FROM
    maintenance m
JOIN
    condominiums c ON m.condo_id = c.condo_id
WHERE
    m.scheduled_date >= CURRENT_DATE()
ORDER BY
    m.scheduled_date;
```

Uso de las Vistas

Estas vistas pueden ser utilizadas para simplificar consultas comunes y para proporcionar reportes fáciles de entender para la administración y los residentes del consorcio.

Ejemplos:

1. Detalles de una unidad en específico

```
SELECT * FROM unit_details WHERE unit_number = '101';
```

2. Detalle de los pagos de las expensas

```
SELECT * FROM expense_payment_status WHERE condo_name = 'Condo One';
```

3. Detalle de la Programación de Mantenimiento de los consorcios

```
SELECT * FROM upcoming_maintenance;
```

Funciones

Resumen

total_expenses_paid_by_owner

Descripción: Calcula el total de expensas pagadas por un propietario para todas las unidades que posee en un año específico.

count_units_in_condominium

Descripción: Devuelve el número de unidades asociadas a un condominio específico.

total_common_expenses_for_year

Descripción: Calcula el monto total de los gastos comunes realizados en un condominio durante un año específico.

get_admin_phone

Descripción: Devuelve el número de teléfono del administrador asociado con un condominio específico.

Ampliación

1. Función para Calcular el Total de Expensas Pagadas por Unidad

Esta función calcula el total de expensas pagadas para una unidad específica en un periodo dado.

DELIMITER //

```
CREATE FUNCTION total_expenses_paid(unit_id_param INT, year_param YEAR)
RETURNS DECIMAL(10, 2)
BEGIN
    DECLARE total_paid DECIMAL(10, 2);

    SELECT SUM(p.amount) INTO total_paid
    FROM payments p
    JOIN expenses e ON p.expense_id = e.expense_id
    WHERE e.unit_id = unit_id_param AND YEAR(e.month_year) = year_param AND
    e.payment_status = 'Paid';

    RETURN IFNULL(total_paid, 0);
END //
```

DELIMITER ;

Uso de la función

Parámetros: **unit_id** de la unidad y el **año** para el cual deseas conocer el total de expensas pagadas. Por ejemplo:

```
SELECT total_expenses_paid(1, 2023) AS Total_Paid;
```

2. Función para Obtener el Número de Contacto del Administrador de un Consorcio

Esta función devuelve el número de teléfono del administrador asociado con un consorcio específico. Muy útil para un contacto rápido.

```
DELIMITER //
```

```
CREATE FUNCTION get_admin_contact(condo_id_param INT)
RETURNS VARCHAR(50)
BEGIN
    DECLARE contact_phone VARCHAR(50);

    SELECT a.phone INTO contact_phone
    FROM administrators a
    JOIN condominiums c ON a.admin_id = c.admin_id
    WHERE c.condo_id = condo_id_param;

    RETURN contact_phone;
END //
```

```
DELIMITER ;
```

Uso de la Función

Para usar esta función, simplemente hay que pasar el **condo_id** del consorcio cuyo número de contacto del administrador necesitas. Por ejemplo:

```
SELECT get_admin_contact(1) AS Admin_Contact;
```

3. Función para Calcular el Total de Expensas Pagadas por un Propietario

Esta función calcula el total de expensas que un propietario ha pagado para todas las unidades que posee.

```
DELIMITER //
```

```

CREATE FUNCTION total_expenses_paid_by_owner(owner_id_param INT)
RETURNS DECIMAL(10, 2)
BEGIN
    DECLARE total_paid DECIMAL(10, 2);

    SELECT SUM(p.amount) INTO total_paid
    FROM payments p
    JOIN expenses e ON p.expense_id = e.expense_id
    JOIN units u ON e.unit_id = u.unit_id
    WHERE u.owner_id = owner_id_param AND e.payment_status = 'Paid';

    RETURN IFNULL(total_paid, 0);
END //

DELIMITER ;

```

Explicación:

- **Parámetro de entrada:** owner_id_param (ID del propietario).
- **Resultado:** Total de expensas pagadas por ese propietario.
- **Uso:** Se puede utilizar para obtener un resumen financiero por propietario.

2. Función para Obtener el Número de Unidades Asociadas a un Condominio

Esta función devuelve el número de unidades que están asociadas a un condominio específico.

```

DELIMITER //

CREATE FUNCTION count_units_in_condominium(condo_id_param INT)
RETURNS INT
BEGIN
    DECLARE unit_count INT;

    SELECT COUNT(*) INTO unit_count
    FROM units
    WHERE condo_id = condo_id_param;

    RETURN unit_count;
END //

DELIMITER ;

```

Explicación:

- **Parámetro de entrada:** `condo_id_param` (ID del condominio).
- **Resultado:** Cantidad de unidades en el condominio.
- **Uso:** Útil para obtener rápidamente un conteo de unidades en cualquier condominio.

3. Función para Calcular el Monto Total de Gastos Comunes en un Año

Esta función calcula el monto total de los gastos comunes realizados en un condominio durante un año específico.

DELIMITER //

```
CREATE FUNCTION total_common_expenses_for_year(condo_id_param INT, year_param
YEAR)
RETURNS DECIMAL(10, 2)
BEGIN
    DECLARE total_expenses DECIMAL(10, 2);

    SELECT SUM(amount) INTO total_expenses
    FROM common_expenses
    WHERE condo_id = condo_id_param AND YEAR(date) = year_param;

    RETURN IFNULL(total_expenses, 0);
END //
```

DELIMITER ;

Explicación:

- **Parámetros de entrada:** `condo_id_param` (ID del condominio), `year_param` (año específico).
- **Resultado:** Total de gastos comunes en el año especificado para el condominio.
- **Uso:** Ayuda a realizar análisis financieros anuales para cada condominio.

Procedimientos

Resumen

add_common_expense_and_update_reserve

Descripción: Agrega un nuevo gasto común y actualiza el fondo de reserva del consorcio correspondiente, asegurando que los registros financieros se mantengan actualizados.

get_owner_total_expenses

Descripción: Calcula el total de expensas pagadas por un propietario para todas las unidades que posee en un año específico.

get_condo_maintenance_summary

Descripción: Proporciona un resumen de los gastos de mantenimiento realizados en un condominio durante un año específico, detallando cada actividad de mantenimiento y los pagos realizados.

Ampliación

1. Agregar un Gasto Común y Actualizar Fondo de Reserva

Este procedimiento almacenado toma varios parámetros para un nuevo gasto común y ajusta el fondo de reserva del consorcio correspondiente según el monto del gasto.

DELIMITER //

```
CREATE PROCEDURE AddCommonExpenseAndUpdateReserve(
    IN condo_id_param INT,
    IN description_param VARCHAR(255),
    IN date_param DATE,
    IN amount_param DECIMAL(10, 2),
    IN category_param VARCHAR(50)
)
BEGIN
    -- Insertar el nuevo gasto común
    INSERT INTO common_expenses (condo_id, description, date, amount, category)
    VALUES (condo_id_param, description_param, date_param, amount_param,
category_param);

    -- Actualizar el fondo de reserva del consorcio, restando el monto del gasto
    UPDATE condominiums
    SET reserve_fund = reserve_fund - amount_param
    WHERE condo_id = condo_id_param;
```

```

-- Verificar que el fondo de reserva no sea negativo y emitir una advertencia si es
necesario
SELECT reserve_fund INTO @new_reserve
FROM condominiums
WHERE condo_id = condo_id_param;

IF @new_reserve < 0 THEN
    SIGNAL SQLSTATE '45000'
    SET MESSAGE_TEXT = 'Advertencia: El fondo de reserva ha quedado en negativo.';
END IF;

END //

DELIMITER ;

```

Descripción del Procedimiento

1. Insertar un Nuevo Gasto: El procedimiento comienza insertando el nuevo gasto común en la tabla **common_expenses** usando los parámetros proporcionados.
2. Actualizar el Fondo de Reserva: A continuación, actualiza el **reserve_fund** en la tabla **condominiums**, disminuyendo el monto del nuevo gasto del fondo de reserva existente.
3. Verificación de Fondo de Reserva: Finalmente, verifica si el fondo de reserva ha caído por debajo de cero y, de ser así, emite una advertencia.

Ejemplo de Uso del Procedimiento Almacenado

```
CALL AddCommonExpenseAndUpdateReserve(1, 'Reparación de Ascensor',
'2023-10-01', 1200.00, 'Mantenimiento');
```

Nota: Este ejemplo añade un gasto para mantenimiento del ascensor y actualiza el fondo de reserva del consorcio con ID 1.

2. GetOwnerTotalExpenses

Descripción: Este procedimiento almacenado calcula el total de expensas que un propietario ha pagado para todas las unidades que posee en un año específico. Es útil para generar informes financieros detallados para los propietarios.

```
DELIMITER //
```

```
CREATE PROCEDURE GetOwnerTotalExpenses(
    IN owner_id_param INT,
    IN year_param YEAR,
```

```

    OUT total_expenses DECIMAL(10, 2)
)
BEGIN
    SELECT SUM(p.amount) INTO total_expenses
    FROM payments p
    JOIN expenses e ON p.expense_id = e.expense_id
    JOIN units u ON e.unit_id = u.unit_id
    WHERE u.owner_id = owner_id_param
    AND YEAR(e.month_year) = year_param
    AND e.payment_status = 'Paid';

    IF total_expenses IS NULL THEN
        SET total_expenses = 0;
    END IF;
END //

DELIMITER ;

```

Explicación:

- **Propósito:** Este procedimiento permite calcular el total de expensas pagadas por un propietario en un año determinado. Esto es especialmente útil para los administradores de consorcios cuando necesitan proporcionar un resumen anual a los propietarios.
- **Parámetros de Entrada:**
 - **owner_id_param** (INT): El ID del propietario para el que se desea calcular el total de expensas.
 - **year_param** (YEAR): El año específico en el que se desean sumar las expensas.
- **Parámetro de Salida:**
 - **total_expenses** (DECIMAL): El total de expensas pagadas por el propietario en el año especificado.
- **Funcionamiento:**
 - El procedimiento suma todos los montos de los pagos realizados por el propietario para todas sus unidades en el año especificado.
 - Si no se han registrado pagos para ese propietario en el año dado, el valor de salida será 0.

Procedimiento 3: GetCondoMaintenanceSummary

Descripción: Este procedimiento almacenado proporciona un resumen de los gastos de mantenimiento realizados en un condominio durante un año específico. Es útil para que los administradores puedan revisar y presentar los costos de mantenimiento de manera clara.

```

DELIMITER //

```

```

CREATE PROCEDURE GetCondoMaintenanceSummary(
    IN condo_id_param INT,
    IN year_param YEAR
)
BEGIN
    SELECT
        m.description AS Maintenance_Description,
        m.scheduled_date AS Scheduled_Date,
        m.provider AS Service_Provider,
        m.estimated_cost AS Estimated_Cost,
        IFNULL(SUM(p.amount), 0) AS Amount_Paid
    FROM maintenance m
    LEFT JOIN payments p ON m.maintenance_id = p.expense_id
    WHERE m.condo_id = condo_id_param
    AND YEAR(m.scheduled_date) = year_param
    GROUP BY m.maintenance_id;
END //

DELIMITER ;

```

Explicación:

- **Propósito:** Este procedimiento genera un resumen de todos los gastos de mantenimiento realizados en un condominio para un año específico, detallando cada actividad de mantenimiento, el proveedor del servicio, el costo estimado y el total pagado hasta la fecha.
- **Parámetros de Entrada:**
 - **condo_id_param** (INT): El ID del condominio del que se desea obtener el resumen de mantenimiento.
 - **year_param** (YEAR): El año específico para el cual se desea revisar los gastos de mantenimiento.
- **Funcionamiento:**
 - El procedimiento consulta la tabla de **maintenance** para obtener todos los registros de mantenimiento realizados en el condominio durante el año especificado.
 - Luego, calcula el total pagado para cada actividad de mantenimiento utilizando la tabla **payments**.
 - El resultado es una lista de todas las actividades de mantenimiento, junto con el proveedor de servicios, el costo estimado y el total pagado hasta la fecha.

Triggers

Resumen

after_payment_insert

Descripción: Se ejecuta después de que se inserta un nuevo pago, actualizando el fondo de reserva del consorcio asociado sumando el monto del pago.

before_owner_delete

Descripción: Previene la eliminación de un propietario si todavía tiene unidades asociadas activas en el sistema.

after_common_expense_update

Descripción: Registra cualquier cambio en la tabla common_expenses en una tabla de auditoría para mantener un historial de modificaciones.

Ampliación

1. Trigger para Actualizar el Fondo de Reserva Después de un Pago

Este *trigger* se ejecuta después de que se inserta un nuevo pago y actualiza el fondo de reserva del consorcio asociado, incrementándolo con el monto del pago.

DELIMITER //

```
CREATE TRIGGER after_payment_insert
AFTER INSERT ON payments
FOR EACH ROW
BEGIN
    DECLARE condo_id INT;

    -- Obtener el ID del consorcio asociado a la unidad que realizó el pago
    SELECT u.condo_id INTO condo_id
    FROM units u
    JOIN expenses e ON u.unit_id = e.unit_id
    WHERE e.expense_id = NEW.expense_id;

    -- Actualizar el fondo de reserva del consorcio sumando el monto del pago
    UPDATE condominiums
    SET reserve_fund = reserve_fund + NEW.amount
    WHERE condo_id = condo_id;
END //
```

DELIMITER ;

Trigger para Prevenir la Eliminación de Propietarios con Unidades Activas

Este *trigger* previene la eliminación de un propietario que todavía tiene unidades asociadas en el sistema.

DELIMITER //

```
CREATE TRIGGER before_owner_delete
BEFORE DELETE ON owners
FOR EACH ROW
BEGIN
    DECLARE unit_count INT;

    -- Contar el número de unidades asociadas al propietario que se intenta eliminar
    SELECT COUNT(*)
    INTO unit_count
    FROM units
    WHERE owner_id = OLD.owner_id;

    -- Si hay unidades asociadas, prevenir la eliminación del propietario
    IF unit_count > 0 THEN
        SIGNAL SQLSTATE '45000'
        SET MESSAGE_TEXT = 'No se puede eliminar al propietario porque todavía tiene unidades asociadas.';
    END IF;
END //
```

DELIMITER ;

Trigger para Registrar Cambios en los Gastos Comunes

Este *trigger* registra cualquier cambio en la tabla `common_expenses` en una tabla de auditoría `common_expenses_audit`.

```
-- Crear tabla de auditoría si no existe
CREATE TABLE IF NOT EXISTS common_expenses_audit (
    audit_id INT PRIMARY KEY AUTO_INCREMENT,
    expense_id INT,
    condo_id INT,
    old_amount DECIMAL(10, 2),
    new_amount DECIMAL(10, 2),
    change_date TIMESTAMP DEFAULT CURRENT_TIMESTAMP
```

);

DELIMITER //

```
CREATE TRIGGER after_common_expense_update
AFTER UPDATE ON common_expenses
FOR EACH ROW
BEGIN
    -- Insertar los cambios en la tabla de auditoría
    INSERT INTO common_expenses_audit (expense_id, condo_id, old_amount,
new_amount)
    VALUES (OLD.expense_id, OLD.condo_id, OLD.amount, NEW.amount);
END //
```

DELIMITER ;

Explicación de los Triggers:

Trigger after_payment_insert:

- Se activa después de que se inserta un pago.
- Actualiza el fondo de reserva del consorcio asociado sumando el monto del pago.

Trigger before_owner_delete:

- Se activa antes de que se intente eliminar un propietario.
- Previene la eliminación si el propietario todavía tiene unidades activas asociadas.

Trigger after_common_expense_update:

- Se activa después de que se actualiza un gasto común.
- Registra los cambios en una tabla de auditoría para mantener un historial de modificaciones.

Diagrama Entidad Relación (E-R)

<https://github.com/jpmelnik/coderhouse-proyecto-final.git>

Herramientas

MySQL

Descripción:

MySQL es un sistema de gestión de bases de datos relacional de código abierto que se utiliza ampliamente para aplicaciones web y de software. Es conocido por su confiabilidad, rendimiento y facilidad de uso.

Aplicación en el Proyecto:

Se utilizó MySQL para la creación, administración y consulta de la base de datos de gestión de consorcios. MySQL permitió diseñar una base de datos robusta, implementar procedimientos almacenados, funciones, vistas y triggers que automatizan y optimizan la gestión de consorcios.

MySQL Workbench

Descripción:

MySQL Workbench es una herramienta visual unificada para desarrolladores de bases de datos y administradores. Ofrece un conjunto de herramientas para modelado de datos, desarrollo de SQL y administración de servidores.

Aplicación en el Proyecto:

MySQL Workbench se utilizó para el diseño y modelado de la base de datos, incluyendo la creación de diagramas entidad-relación (ERD), la escritura y ejecución de scripts SQL, y la gestión de las tablas, vistas, funciones y procedimientos almacenados.

SQL (Structured Query Language)

Descripción:

SQL es el lenguaje estándar utilizado para interactuar con bases de datos relacionales. Permite realizar consultas, actualizaciones, inserciones, y modificaciones en los datos almacenados.

Aplicación en el Proyecto:

SQL fue el lenguaje principal utilizado para la creación de tablas, la inserción de datos, la definición de relaciones entre tablas, y la implementación de lógica avanzada a través de procedimientos almacenados, funciones y triggers.

Git y GitHub

Descripción: Git es un sistema de control de versiones distribuido utilizado para rastrear cambios en el código fuente durante el desarrollo de software. GitHub es una plataforma basada en la web que utiliza Git para el control de versiones y facilita la colaboración.

Aplicación en el Proyecto: Git y GitHub se utilizaron para gestionar las versiones del código SQL y colaborar en el desarrollo del proyecto, permitiendo un control preciso de los cambios y la colaboración entre diferentes miembros del equipo.

Visual Studio Code

Descripción:

Visual Studio Code es un editor de código fuente desarrollado por Microsoft. Es ligero y potente, con soporte para múltiples lenguajes de programación y una amplia gama de extensiones.

Aplicación en el Proyecto:

Visual Studio Code se utilizó como editor principal para escribir y editar scripts SQL, aprovechando su integración con extensiones de bases de datos y su capacidad para trabajar con Git.

Diagramas Entidad-Relación (ERD)

Descripción

Los diagramas entidad-relación son representaciones gráficas que ilustran las entidades dentro de un sistema y las relaciones entre ellas.

Aplicación en el Proyecto

Los ERD se utilizaron para planificar y visualizar la estructura de la base de datos, facilitando la comprensión de las relaciones entre tablas y el diseño general del sistema.

Google Drive o Documentos Compartidos

Descripción

Google Drive es un servicio de almacenamiento en la nube que permite el almacenamiento y la compartición de archivos y documentos de manera colaborativa.

Aplicación en el Proyecto

Google Drive se utilizó para almacenar documentos de proyecto, compartir recursos y colaborar en la redacción y revisión de los mismos.

Conclusión Final

El desarrollo de la base de datos para la gestión de consorcios ha logrado crear una solución robusta y eficiente que centraliza y automatiza la administración de propiedades. Al integrar funciones avanzadas, procedimientos almacenados, vistas y triggers, se ha optimizado significativamente la gestión de expensas, pagos, y mantenimiento, mejorando la transparencia y la eficiencia operativa. Este proyecto proporciona una base sólida sobre la cual los administradores pueden construir, adaptarse y crecer, asegurando una gestión efectiva y confiable para el futuro.