

# FASE 1C - GESTIÓN DE CONTRATOS Y ALQUILER

## Propiedades Alquiladas - Contratos e Inquilinos

**Duración estimada:** 3-4 días

**Requisito previo:** Fase 1A y 1B completadas

**Objetivo:** Gestionar contratos de alquiler y calcular días alquilados (base para amortización)

## ¿QUÉ ES TIPO "ALQUILER"?

Cuando una propiedad está **alquilada** a un inquilino:

- ✓ Hay contratos con fechas inicio/fin
- ✓ Se cobran rentas mensuales (ingresos)
- ✓ Hay gastos deducibles (IBI, comunidad, reparaciones...)
- ✓ Se calcula amortización (3% del valor amortizable)
- ✓ Declaración puede ser **trimestral o anual**

Es **MÁS COMPLEJO** que **imputación** pero genera el Modelo 210 real de alquiler.

## MODIFICACIONES BASE DE DATOS

### TABLA NUEVA 1: contratos\_alquiler

```
CREATE TABLE contratos_alquiler (  
    id_contrato SERIAL PRIMARY KEY,  
  
    -- Relación con propiedad  
    id_propiedad INT NOT NULL REFERENCES propiedades(id_propiedad),  
  
    -- Fechas del contrato (CRÍTICAS)  
    fecha_inicio DATE NOT NULL,  
    fecha_fin DATE NOT NULL,  
  
    -- Información financiera  
    renta_mensual DECIMAL(10,2) NOT NULL CHECK (renta_mensual > 0),  
    forma_pago VARCHAR(50), -- 'transferencia', 'efectivo', etc.  
    dia_pago INT CHECK (dia_pago BETWEEN 1 AND 31), -- Día del mes que paga  
  
    -- Depósito/fianza  
    deposito DECIMAL(10,2),  
  
    -- Datos del inquilino  
    nombre_inquilino VARCHAR(200) NOT NULL,
```

```

apellidos_inquilino VARCHAR(200),
dni_nie_inquilino VARCHAR(12),
email_inquilino VARCHAR(150),
telefono_inquilino VARCHAR(20),

-- Documento del contrato
ruta_contrato_pdf VARCHAR(500),

-- Estado
estado VARCHAR(20) DEFAULT 'activo',
/* Valores:
'activo'      = Contrato vigente
'finalizado'  = Contrato terminado normalmente
'cancelado'   = Contrato roto antes de tiempo
'renovado'    = Renovado a nuevo contrato
*/

motivo_cancelacion TEXT, -- Si cancelado, explicar por qué

-- Auditoría
fecha_alta TIMESTAMP DEFAULT NOW(),
usuario_alta VARCHAR(100),

-- Notas
observaciones TEXT,

-- Validación fechas
CONSTRAINT chk_fechas_contrato CHECK (fecha_fin >= fecha_inicio)
);

CREATE INDEX idx_contrato_propiedad ON contratos_alquiler(id_propiedad);
CREATE INDEX idx_contrato_fechas ON contratos_alquiler(fecha_inicio, fecha_fin);
CREATE INDEX idx_contrato_estado ON contratos_alquiler(estado);
CREATE INDEX idx_contrato_inquilino ON contratos_alquiler(nombre_inquilino);

```

**¿Qué guarda?** Todos los contratos de alquiler con fechas exactas (necesario para calcular días)

## TABLA NUEVA 2: pagos\_alquiler (opcional pero recomendada)

```

CREATE TABLE pagos_alquiler (
    id_pago SERIAL PRIMARY KEY,

    -- Relación con contrato
    id_contrato INT NOT NULL REFERENCES contratos_alquiler(id_contrato),

    -- Fecha y periodo del pago
    fecha_pago DATE NOT NULL,
    mes_correspondiente INT CHECK (mes_correspondiente BETWEEN 1 AND 12),
    ano_correspondiente INT NOT NULL,

    -- Importe
    importe DECIMAL(10,2) NOT NULL CHECK (importe > 0),

    -- Estado

```

```

estado VARCHAR(20) DEFAULT 'pendiente',
/* Valores:
    'pendiente' = No pagado aún
    'pagado'     = Pagado
    'atrasado'   = Con retraso
    'impagado'   = No va a pagar
*/

-- Método de pago
metodo_pago VARCHAR(50), -- 'transferencia', 'efectivo', etc.
referencia_bancaria VARCHAR(100),

-- Justificante
ruta_justificante VARCHAR(500),

fecha_registro TIMESTAMP DEFAULT NOW()
);

CREATE INDEX idx_pago_contrato ON pagos_alquiler(id_contrato);
CREATE INDEX idx_pago_fecha ON pagos_alquiler(fecha_pago);
CREATE INDEX idx_pago_estado ON pagos_alquiler(estado);

```

¿Qué guarda? Registro de cada pago mensual del inquilino (para control de ingresos)

## TRIGGER: VALIDAR NO SOLAPAMIENTO DE CONTRATOS

**CRÍTICO:** Una propiedad no puede tener 2 contratos activos al mismo tiempo.

```

CREATE OR REPLACE FUNCTION validar_no_solapamiento()
RETURNS TRIGGER AS $$
DECLARE
    num_solapados INT;
BEGIN
    -- Contar contratos que se solapan con el nuevo
    SELECT COUNT(*) INTO num_solapados
    FROM contratos_alquiler
    WHERE id_propiedad = NEW.id_propiedad
    AND id_contrato != COALESCE(NEW.id_contrato, -1)
    AND estado IN ('activo', 'renovado')
    AND (
        -- Nuevo contrato empieza durante otro contrato
        (NEW.fecha_inicio BETWEEN fecha_inicio AND fecha_fin)
        OR
        -- Nuevo contrato termina durante otro contrato
        (NEW.fecha_fin BETWEEN fecha_inicio AND fecha_fin)
        OR
        -- Nuevo contrato contiene completamente a otro
        (NEW.fecha_inicio <= fecha_inicio AND NEW.fecha_fin >= fecha_fin)
    );

    IF num_solapados > 0 THEN
        RAISE EXCEPTION 'ERROR: Hay % contrato(s) que se solapan con estas fechas', num_s
    END IF;

```

```
        RETURN NEW;
    END;
$$ LANGUAGE plpgsql;

CREATE TRIGGER trg_validar_no_solapamiento
BEFORE INSERT OR UPDATE ON contratos_alquiler
FOR EACH ROW EXECUTE FUNCTION validar_no_solapamiento();
```

## API ENDPOINTS NUEVOS

### 1. Crear Contrato de Alquiler

```
POST /api/propiedades/{id}/contratos
```

#### Request:

```
{
  "fecha_inicio": "2024-01-01",
  "fecha_fin": "2024-12-31",
  "renta_mensual": 1200.00,
  "dia_pago": 5,
  "deposito": 1200.00,
  "nombre_inquilino": "Juan García",
  "apellidos_inquilino": "Martínez",
  "dni_nie_inquilino": "12345678A",
  "email_inquilino": "juan@example.com",
  "telefono_inquilino": "+34612345678"
}
```

#### Response:

```
{
  "id_contrato": 15,
  "id_propiedad": 2,
  "fecha_inicio": "2024-01-01",
  "fecha_fin": "2024-12-31",
  "dias_contrato": 366,
  "renta_mensual": 1200.00,
  "ingresos_totales_estimados": 14400.00,
  "nombre_inquilino": "Juan García Martínez",
  "estado": "activo"
}
```

## 2. Listar Contratos de una Propiedad

```
GET /api/propiedades/{id}/contratos
```

### Response:

```
{
  "propiedad": "Av. Barcelona 45",
  "contratos": [
    {
      "id_contrato": 15,
      "fecha_inicio": "2024-01-01",
      "fecha_fin": "2024-12-31",
      "dias": 366,
      "renta_mensual": 1200.00,
      "inquilino": "Juan García",
      "estado": "activo"
    },
    {
      "id_contrato": 12,
      "fecha_inicio": "2023-01-01",
      "fecha_fin": "2023-12-31",
      "dias": 365,
      "renta_mensual": 1100.00,
      "inquilino": "María López",
      "estado": "finalizado"
    }
  ],
  "total_contratos": 2
}
```

## 3. Actualizar Contrato

```
PUT /api/contratos/{id}
```

### Request:

```
{
  "fecha_fin": "2024-06-30",
  "estado": "cancelado",
  "motivo_cancelacion": "Inquilino se muda antes de tiempo"
}
```

#### 4. Calcular Días Alquilados (FUNCIÓN CRÍTICA)

```
GET /api/propiedades/{id}/dias-alquilados?ano=2024
```

##### Response:

```
{
  "id_propiedad": 2,
  "direccion": "Av. Barcelona 45",
  "ano": 2024,
  "contratos_en_ano": [
    {
      "id_contrato": 15,
      "fecha_inicio": "2024-01-01",
      "fecha_fin": "2024-12-31",
      "dias_en_ano": 366,
      "inquilino": "Juan García"
    }
  ],
  "total_dias_alquilados": 366,
  "total_dias_sin_alquilar": 0,
  "porcentaje_ocupacion": 100.0,
  "ingresos_estimados": 14400.00
}
```

#### 5. Registrar Pago de Alquiler

```
POST /api/contratos/{id}/pagos
```

##### Request:

```
{
  "fecha_pago": "2024-01-05",
  "mes_correspondiente": 1,
  "ano_correspondiente": 2024,
  "importe": 1200.00,
  "metodo_pago": "transferencia",
  "referencia_bancaria": "TRANS-2024-001"
}
```

#### 6. Listar Pagos de un Contrato

```
GET /api/contratos/{id}/pagos?ano=2024
```

##### Response:

```

{
  "contrato": {
    "id_contrato": 15,
    "inquilino": "Juan García",
    "renta_mensual": 1200.00
  },
  "ano": 2024,
  "pagos": [
    {
      "mes": "Enero 2024",
      "fecha_pago": "2024-01-05",
      "importe": 1200.00,
      "estado": "pagado"
    },
    {
      "mes": "Febrero 2024",
      "fecha_pago": "2024-02-05",
      "importe": 1200.00,
      "estado": "pagado"
    },
    {
      "mes": "Marzo 2024",
      "estado": "pendiente"
    }
  ],
  "total_cobrado": 2400.00,
  "total_pendiente": 1200.00,
  "porcentaje_cobrado": 66.67
}

```

## LÓGICA DE CÁLCULO: DÍAS ALQUILADOS

### Función principal: `calcular_dias_alquilados()`

```

from datetime import date, timedelta

def calcular_dias_alquilados(id_propiedad, ano):
    """
    Calcula el total de días que estuvo alquilada la propiedad en un año.

    CRÍTICO: Esta función es la BASE para calcular amortización proporcional.

    Args:
        id_propiedad: ID de la propiedad
        ano: Año fiscal (ej: 2024)

    Returns:
        dict con días alquilados, contratos y desglose
    """

    # Fechas límite del año
    fecha_inicio_ano = date(ano, 1, 1)

```

```

fecha_fin_ano = date(ano, 12, 31)

# Calcular días del año (365 o 366 si bisiestro)
dias_totales_ano = (fecha_fin_ano - fecha_inicio_ano).days + 1

# Obtener contratos que intersectan con el año
contratos = ContratoAlquiler.objects.filter(
    id_propiedad=id_propiedad,
    estado__in=['activo', 'finalizado', 'renovado'],
    fecha_inicio__lte=fecha_fin_ano,
    fecha_fin__gte=fecha_inicio_ano
).order_by('fecha_inicio')

if not contratos.exists():
    return {
        'id_propiedad': id_propiedad,
        'ano': ano,
        'contratos': [],
        'total_dias_alquilados': 0,
        'total_dias_sin_alquilar': dias_totales_ano,
        'porcentaje_ocupacion': 0.0,
        'mensaje': 'No hay contratos en este año'
    }

# Calcular días por cada contrato
contratos_info = []
total_dias_alquilados = 0

for contrato in contratos:
    # Intersección del contrato con el año
    inicio_efectivo = max(contrato.fecha_inicio, fecha_inicio_ano)
    fin_efectivo = min(contrato.fecha_fin, fecha_fin_ano)

    # Calcular días (inclusivo: inicio y fin cuentan)
    dias_en_ano = (fin_efectivo - inicio_efectivo).days + 1

    # Acumular
    total_dias_alquilados += dias_en_ano

    # Calcular ingresos del contrato en este año
    meses_en_ano = dias_en_ano / 30.44 # Promedio días por mes
    ingresos_estimados = contrato.renta_mensual * meses_en_ano

    contratos_info.append({
        'id_contrato': contrato.id_contrato,
        'fecha_inicio': contrato.fecha_inicio,
        'fecha_fin': contrato.fecha_fin,
        'fecha_inicio_efectiva': inicio_efectivo,
        'fecha_fin_efectiva': fin_efectivo,
        'dias_en_ano': dias_en_ano,
        'renta_mensual': contrato.renta_mensual,
        'ingresos_estimados': round(ingresos_estimados, 2),
        'inquilino': f"{contrato.nombre_inquilino} {contrato.apellidos_inquilino or ' '}",
        'estado': contrato.estado
    })

```



```

# VALIDACIÓN: Total no puede exceder días del año
if total_dias_alquilados > dias_totales_ano:
    raise ValidationError(
        f"ERROR: Días alquilados ({total_dias_alquilados}) > "
        f"días del año ({dias_totales_ano}). "
        f"Revise contratos por solapamiento."
    )

# Calcular días sin alquilar
dias_sin_alquilar = dias_totales_ano - total_dias_alquilados

# Porcentaje de ocupación
porcentaje_ocupacion = (total_dias_alquilados / dias_totales_ano) * 100

# Ingresos totales estimados
ingresos_totales = sum([c['ingresos_estimados'] for c in contratos_info])

return {
    'id_propiedad': id_propiedad,
    'ano': ano,
    'dias_totales_ano': dias_totales_ano,
    'contratos': contratos_info,
    'num_contratos': len(contratos_info),
    'total_dias_alquilados': total_dias_alquilados,
    'total_dias_sin_alquilar': dias_sin_alquilar,
    'porcentaje_ocupacion': round(porcentaje_ocupacion, 2),
    'ingresos_totales_estimados': round(ingresos_totales, 2)
}

```

## Función auxiliar: detectar\_solapamientos()

```

def detectar_solapamientos(id_propiedad):
    """
    Detecta si hay contratos solapados (error de datos)
    """

    contratos = ContratoAlquiler.objects.filter(
        id_propiedad=id_propiedad,
        estado__in=['activo', 'finalizado', 'renovado']
    ).order_by('fecha_inicio')

    solapamientos = []

    for i, c1 in enumerate(contratos):
        for c2 in contratos[i+1:]:
            # Verificar si se solapan
            if c1.fecha_fin >= c2.fecha_inicio and c1.fecha_inicio <= c2.fecha_fin:
                dias_solapados = min(c1.fecha_fin, c2.fecha_fin) - max(c1.fecha_inicio, c2.fecha_inicio)
                dias_solapados = dias_solapados.days + 1

                solapamientos.append({
                    'contrato_1': {
                        'id': c1.id_contrato,
                        'fechas': f"{c1.fecha_inicio} - {c1.fecha_fin}",

```

```

        'inquilino': c1.nombre_inquilino
    },
    'contrato_2': {
        'id': c2.id_contrato,
        'fechas': f"{c2.fecha_inicio} - {c2.fecha_fin}",
        'inquilino': c2.nombre_inquilino
    },
    'dias_solapados': dias_solapados
})

return solapamientos

```

## PANTALLAS FRONTEND

### PANTALLA 1: Detalle Propiedad - Tab Contratos

PROPIEDAD: Av. Barcelona 45	
Tabs: [Datos] [Contratos] [Gastos] [Cálculos]	
[TAB ACTIVO: Contratos]	
Tipo declaración: ALQUILER	
[+ Nuevo]	
CONTRATOS DE ALQUILER	
Contrato actual (activo):	
<div> <div>Contrato 2024</div> <div> Inquilino: Juan García Martínez  Periodo: 01/01/2024 - 31/12/2024  Renta: 1.200€/mes  Días: 366 días (100% ocupación) </div> <div> [Ver pagos] [Editar] [Finalizar] </div> </div>	
Contratos anteriores:	
<div> 2023: María López (365 días) </div> <div> 2022: Pedro Sánchez (365 días) </div>	
RESUMEN OCUPACIÓN 2024	
Días alquilados: 366 días (100%) Días sin alquilar: 0 días (0%) Ingresos estimados: 14.400,00€	

## PANTALLA 2: Formulario Nuevo Contrato

NUEVO CONTRATO - Av. Barcelona 45	
FECHAS DEL CONTRATO *	
Fecha inicio:	[__/__/____] ▯
Fecha fin:	[__/__/____] ▯
Duración:	[calculado automático] días
RENTA *	
Renta mensual:	[_____] €
Día de pago:	[__] (1-31)
Forma de pago:	▼ Transferencia
Depósito:	[_____] € (opcional)
DATOS INQUILINO *	
Nombre:	[_____]
Apellidos:	[_____]
DNI/NIE:	[_____]
Email:	[_____]
Teléfono:	[_____]
CONTRATO PDF	
<input type="checkbox"/> Subir contrato firmado] (opcional)	
OBSERVACIONES	
[_____]	
[_____]	
⚠ Sistema validará que no se solape con contratos existentes	
[Cancelar] [Guardar]	

## PANTALLA 3: Gestión de Pagos del Contrato

PAGOS - Contrato Juan García 2024				
Renta mensual: 1.200€   Día pago: 5 de cada mes				
[+ Nuevo pago]				
Mes	Fecha	Importe	Estado	
Enero	05/01	1.200€	✓ Pagado	

Febrero	05/02	1.200€	✓ Pagado
Marzo	07/03	1.200€	⚠ Atraso
Abril	-	1.200€	□ Pend.
Mayo	-	1.200€	□ Pend.
...			

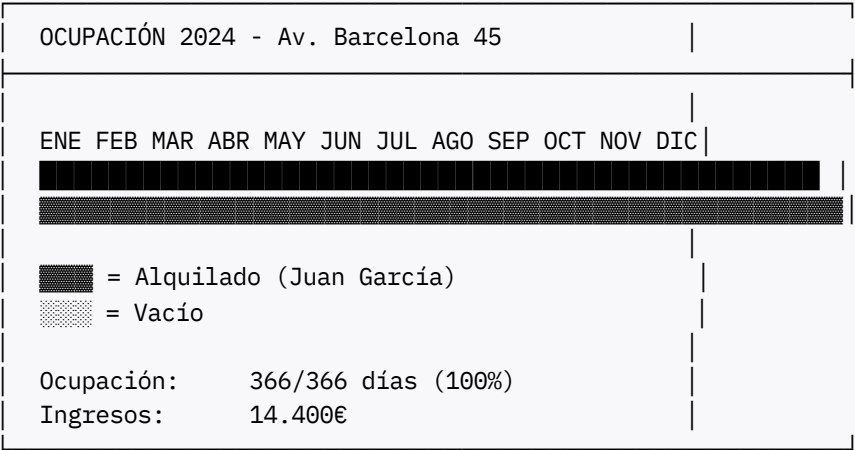
Resumen 2024:

Total cobrado: 3.600€ (3 meses)

Total pendiente: 9.600€ (8 meses)

% cobrado: 30%

PANTALLA 4: Vista Calendario de Ocupación



VALIDACIONES IMPORTANTES

Validación 1: No solapamiento de contratos

```
def validar_no_solapamiento_nuevo(id_propiedad, fecha_inicio, fecha_fin, id_contrato_actu
    """
    Valida que el nuevo contrato no se solape con existentes
    """

    contratos_solapados = ContratoAlquiler.objects.filter(
        id_propiedad=id_propiedad,
        estado__in=['activo', 'renovado']
    ).exclude(
        id_contrato=id_contrato_actual # Excluir si estamos editando
    )

    for contrato in contratos_solapados:
        # Verificar solapamiento
        if (fecha_inicio <= contrato.fecha_fin and
            fecha_fin >= contrato.fecha_inicio):
```

```

        raise ValidationError(
            f"SOLAPAMIENTO: Este contrato se solapa con el contrato "
            f"de {contrato.nombre_inquilino} "
            f"({contrato.fecha_inicio} - {contrato.fecha_fin})"
        )

```

## Validación 2: Fecha fin posterior a inicio

```

def validar_fechas_contrato(fecha_inicio, fecha_fin):
    if fecha_fin <= fecha_inicio:
        raise ValidationError(
            "Fecha fin debe ser posterior a fecha inicio"
        )

    # Validar que contrato no sea demasiado largo (máx 5 años)
    dias = (fecha_fin - fecha_inicio).days
    if dias > 1825: # 5 años
        raise ValidationError(
            f"Contrato muy largo ({dias} días). Máximo 5 años (1825 días)"
        )

```

## Validación 3: Renta mensual positiva

```

def validar_renta_mensual(renta):
    if renta <= 0:
        raise ValidationError("Renta mensual debe ser mayor a 0€")

    # Validar que no sea excesiva (ej: máx 10.000€/mes)
    if renta > 10000:
        raise ValidationError(
            f"Renta mensual muy alta ({renta}€). Revisar."
        )

```

## DATOS DE PRUEBA

```

-- Contrato 2024 para Av. Barcelona 45 (propiedad 2)
INSERT INTO contratos_alquiler (
    id_propiedad, fecha_inicio, fecha_fin, renta_mensual,
    dia_pago, deposito, nombre_inquilino, apellidos_inquilino,
    dni_nie_inquilino, email_inquilino, telefono_inquilino, estado
) VALUES (
    2, '2024-01-01', '2024-12-31', 1200.00,
    5, 1200.00, 'Juan', 'García Martínez',
    '12345678A', 'juan@example.com', '+34612345678', 'activo'
);

-- Contrato 2023 (finalizado)
INSERT INTO contratos_alquiler (
    id_propiedad, fecha_inicio, fecha_fin, renta_mensual,
    dia_pago, deposito, nombre_inquilino, apellidos_inquilino,

```

```

        dni_nie_inquilino, email_inquilino, telefono_inquilino, estado
    ) VALUES (
        2, '2023-01-01', '2023-12-31', 1100.00,
        5, 1100.00, 'María', 'López Sánchez',
        '98765432B', 'maria@example.com', '+34687654321', 'finalizado'
    );

-- Pagos del contrato 2024
INSERT INTO pagos_alquiler (
    id_contrato, fecha_pago, mes_correspondiente, ano_correspondiente,
    importe, metodo_pago, estado
) VALUES
(1, '2024-01-05', 1, 2024, 1200.00, 'transferencia', 'pagado'),
(1, '2024-02-05', 2, 2024, 1200.00, 'transferencia', 'pagado'),
(1, '2024-03-07', 3, 2024, 1200.00, 'transferencia', 'pagado');

```

## PRUEBAS QUE DEBES HACER

### Test 1: Crear contrato válido

```

POST /api/propiedades/2/contratos
Body: {
  "fecha_inicio": "2024-01-01",
  "fecha_fin": "2024-12-31",
  "renta_mensual": 1200,
  "nombre_inquilino": "Juan García"
}

```

Verificar:

- ✓ Se crea contrato
- ✓ Estado = 'activo'
- ✓ Días calculados = 366

### Test 2: Validar solapamiento

```

POST /api/propiedades/2/contratos
Body: {
  "fecha_inicio": "2024-06-01",
  "fecha_fin": "2024-12-31",
  ...
}

```

Verificar:

- ✗ Da ERROR: "Se solapa con contrato existente"

### Test 3: Calcular días alquilados

GET /api/propiedades/2/dias-alquilados?ano=2024

Verificar:

- ✓ total\_dias\_alquilados = 366
- ✓ total\_dias\_sin\_alquilar = 0
- ✓ porcentaje\_ocupacion = 100%

### Test 4: Registrar pagos

POST /api/contratos/1/pagos

Body: {

```
"fecha_pago": "2024-01-05",  
"mes_correspondiente": 1,  
"ano_correspondiente": 2024,  
"importe": 1200
```

}

Verificar:

- ✓ Se registra pago
- ✓ Estado = 'pagado'

### Test 5: Listar pagos

GET /api/contratos/1/pagos?ano=2024

Verificar:

- ✓ Muestra todos los pagos del año
- ✓ Calcula total cobrado correctamente
- ✓ Muestra meses pendientes

## ENTREGABLES FASE 1C

### 1. Base de datos actualizada:

- Tabla contratos\_alquiler
- Tabla pagos\_alquiler
- Trigger no solapamiento
- Datos de prueba

### 2. Backend funcionando:

- 6 endpoints nuevos
- Función calcular\_dias\_alquilados()
- Función detectar\_solapamientos()
- 3 validaciones

### 3. Frontend actualizado:

- Tab Contratos en detalle propiedad
- Formulario nuevo contrato
- Vista pagos del contrato
- Calendario ocupación (opcional)

### 4. Tests pasando:

- Los 5 tests descritos
- Capturas de pantalla

## CRITERIOS ACEPTACIÓN

- ✓ Puedo crear contratos de alquiler
- ✓ Sistema valida no solapamiento
- ✓ Puedo ver contratos de una propiedad
- ✓ Puedo calcular días alquilados en un año
- ✓ Días calculados son correctos (incluyendo día inicio y fin)
- ✓ Puedo registrar pagos mensuales
- ✓ Puedo ver estado de pagos (cobrado/pendiente)
- ✓ Frontend muestra resumen ocupación
- ✓ No hay errores en consola

## LO QUE NO HACEMOS EN ESTA FASE

- X Cálculo de amortización (Fase 1D)
- X Gestión de gastos deducibles (Fase 1E)
- X Generación Modelo 210 alquiler (Fase 1F)
- X Rentas negativas (Fase 1G)

### Solo gestión de contratos y cálculo de días

## SIGUIENTE FASE

Cuando termines Fase 1C, me avisas y te doy:

### Fase 1D: Cálculo de Amortización

- Valor amortizable (precio + gastos compra)
- Separación suelo/construcción
- Amortización 3% anual
- Prorratio por días alquilados
- Aplicación % copropietarios

**Después:**



- Fase 1E: Gestión Gastos Deducibles
- Fase 1F: Modelo 210 Alquiler Completo
- Fase 1G: Rentas Negativas

## RESUMEN VISUAL

### PROGRAMA GESTORÍA - FASE 1C

Cliente: Piotr Kowalski

└─ Propiedad: Av. Barcelona 45

└─ Tipo: ALQUILER ✓

└─ Contratos: ← NUEVA FUNCIONALIDAD

└─ 2024: Juan García (366 días)

└─ 2023: María López (365 días)

└─ 2022: Pedro Sánchez (365 días)

└─ Ocupación 2024:

└─ Días alquilados: 366 (100%)

└─ Días vacío: 0 (0%)

└─ Ingresos: 14.400€

└─ Pagos 2024:

└─ Enero: 1.200€ ✓

└─ Febrero: 1.200€ ✓

└─ Marzo: 1.200€ ✓

└─ Pendientes: 9 meses

**Plazo:** 3-4 días

**Cuando termines:** Envíame código + tests + capturas pantallas