

# FASE 1E - GESTIÓN DE GASTOS DEDUCIBLES

## Gastos Fiscalmente Deducibles del Alquiler

**Duración estimada:** 3-4 días

**Requisito previo:** Fases 1A, 1B, 1C y 1D completadas

**Objetivo:** Gestionar todos los gastos deducibles y calcular total deducible por año

## ¿QUÉ GASTOS SON DEDUCIBLES EN ALQUILER?

En el Modelo 210 para alquiler, se pueden **deducir gastos** relacionados con la propiedad alquilada.

### Tipos de gastos según su deducibilidad:

#### 1. GASTOS PROPORCIONALES (se prorratean por días alquilados)

- **IBI** (Impuesto Bienes Inmuebles)
- **Comunidad** de propietarios
- **Seguros** (hogar, alquiler)
- **Intereses hipoteca** (si hay préstamo)
- **Suministros** si los paga el propietario (agua, luz, gas)
- **Gastos de conservación menores**

**Se deducen:**  $\text{Gasto} \times (\text{Días alquiler} / 365)$

#### 2. GASTOS 100% DEDUCIBLES (sin prorratear)

- **Reparaciones** necesarias
- **Gestoría/Biuro** (declaraciones fiscales)
- **Agencia inmobiliaria** (búsqueda inquilino)
- **Abogado** (contratos, desahucios)
- **Publicidad** alquiler

**Se deducen:** 100% del gasto

#### 3. NO DEDUCIBLES

- Mejoras que aumentan valor (cocina nueva, terraza...)
- Gastos personales
- Multas

## LÍMITE IMPORTANTE: RENTAS NEGATIVAS

### Regla fiscal crítica:

**Solo REPARACIONES + INTERESES HIPOTECA** pueden generar rentas negativas compensables.

Si resultado es negativo por reparaciones/intereses:

- No se paga nada ese año
- El saldo negativo se compensa en los 4 años siguientes

**Otros gastos (IBI, comunidad, etc.):** Máximo deducible hasta que la renta = 0€. No generan rentas negativas compensables.

## MODIFICACIONES BASE DE DATOS

### TABLA NUEVA: gastos

```
CREATE TABLE gastos (
    id_gasto SERIAL PRIMARY KEY,

    -- Relación con propiedad
    id_propiedad INT NOT NULL REFERENCES propiedades(id_propiedad),

    -- Tipo de gasto (CRÍTICO para clasificación)
    tipo_gasto VARCHAR(50) NOT NULL,
    /* Valores:
        'ibi'                = IBI (proporcional)
        'comunidad'          = Comunidad propietarios (proporcional)
        'seguro'              = Seguros (proporcional)
        'intereses_hipoteca' = Intereses préstamo (proporcional, genera renta negativa)
        'suministros'        = Agua, luz, gas (proporcional)
        'conservacion'       = Mantenimiento menor (proporcional)
        'reparacion'         = Reparaciones necesarias (100%, genera renta negativa)
        'biuro'              = Gestoría/contable (100%)
        'agencia'            = Agencia inmobiliaria (100%)
        'abogado'            = Servicios jurídicos (100%)
        'publicidad'        = Anuncios alquiler (100%)
        'otro'               = Otros gastos
    */

    -- Clasificación automática
    es_proporcional BOOLEAN GENERATED ALWAYS AS (
        tipo_gasto IN ('ibi', 'comunidad', 'seguro', 'intereses_hipoteca',
            'suministros', 'conservacion')
    ) STORED,

    genera_renta_negativa BOOLEAN GENERATED ALWAYS AS (
        tipo_gasto IN ('reparacion', 'intereses_hipoteca')
    ) STORED,

    -- Información del gasto
```

```

descripcion VARCHAR(500) NOT NULL,
importe DECIMAL(10,2) NOT NULL CHECK (importe > 0),
fecha_gasto DATE NOT NULL,

-- Periodo que cubre (para gastos anuales como IBI)
fecha_inicio_periodo DATE,
fecha_fin_periodo DATE,

-- Proveedor
nombre_proveedor VARCHAR(200),

-- Factura/justificante
numero_factura VARCHAR(100),
ruta_factura VARCHAR(500),

-- Validación
validado BOOLEAN DEFAULT FALSE,
fecha_validacion TIMESTAMP,

-- Auditoría
fecha_registro TIMESTAMP DEFAULT NOW(),
usuario_registro VARCHAR(100),

-- Notas
observaciones TEXT
);

CREATE INDEX idx_gasto_propiedad ON gastos(id_propiedad);
CREATE INDEX idx_gasto_tipo ON gastos(tipo_gasto);
CREATE INDEX idx_gasto_fecha ON gastos(fecha_gasto);
CREATE INDEX idx_gasto_validado ON gastos(validado);

```

¿Qué guarda? Todos los gastos deducibles de cada propiedad.

## TABLA NUEVA: rentas\_negativas

```

CREATE TABLE rentas_negativas (
    id_renta_negativa SERIAL PRIMARY KEY,

    -- Relación
    id_propiedad INT NOT NULL REFERENCES propiedades(id_propiedad),
    id_cliente INT NOT NULL REFERENCES clientes(id_cliente),

    -- Año origen
    ano_origen INT NOT NULL,

    -- Importe negativo
    importe_negativo DECIMAL(12,2) NOT NULL CHECK (importe_negativo > 0),

    -- Concepto que generó la renta negativa
    concepto VARCHAR(50) NOT NULL, -- 'reparaciones', 'intereses', 'mixto'

    -- Control de compensación
    importe_compensado DECIMAL(12,2) DEFAULT 0.00,

```

```

importe_pendiente DECIMAL(12,2) GENERATED ALWAYS AS (
    importe_negativo - importe_compensado
) STORED,

-- Límite temporal (4 años)
ano_limite INT GENERATED ALWAYS AS (ano_origen + 4) STORED,

-- Estado
estado VARCHAR(20) DEFAULT 'pendiente',
/* Valores:
    'pendiente' = Aún se puede compensar
    'compensado' = Totalmente compensado
    'vencido'    = Pasaron 4 años, se pierde
*/

-- Auditoría
fecha_creacion TIMESTAMP DEFAULT NOW(),

-- Notas
observaciones TEXT
);

CREATE INDEX idx_rn_propiedad ON rentas_negativas(id_propiedad);
CREATE INDEX idx_rn_cliente ON rentas_negativas(id_cliente);
CREATE INDEX idx_rn_ano ON rentas_negativas(ano_origen);
CREATE INDEX idx_rn_estado ON rentas_negativas(estado);

```

**¿Qué guarda?** Rentas negativas pendientes de compensar (máximo 4 años).

## **TABLA NUEVA: compensaciones\_rentas\_negativas**

```

CREATE TABLE compensaciones_rentas_negativas (
    id_compensacion SERIAL PRIMARY KEY,

    -- Renta negativa que se compensa
    id_renta_negativa INT NOT NULL REFERENCES rentas_negativas(id_renta_negativa),

    -- Declaración donde se aplica la compensación
    id_declaracion INT NOT NULL REFERENCES declaraciones_210(id_declaracion),

    -- Año en que se compensa
    ano_compensacion INT NOT NULL,

    -- Importe compensado
    importe_compensado DECIMAL(12,2) NOT NULL CHECK (importe_compensado > 0),

    -- Auditoría
    fecha_compensacion TIMESTAMP DEFAULT NOW(),
    usuario VARCHAR(100)
);

CREATE INDEX idx_comp_rn ON compensaciones_rentas_negativas(id_renta_negativa);
CREATE INDEX idx_comp_decl ON compensaciones_rentas_negativas(id_declaracion);

```

¿Qué guarda? Histórico de compensaciones aplicadas.

## API ENDPOINTS NUEVOS

### 1. Registrar Gasto

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

**Request:**

```
{
  "tipo_gasto": "ibi",
  "descripcion": "IBI 2024",
  "importe": 850.00,
  "fecha_gasto": "2024-01-15",
  "fecha_inicio_periodo": "2024-01-01",
  "fecha_fin_periodo": "2024-12-31",
  "nombre_proveedor": "Ayuntamiento Madrid",
  "numero_factura": "IBI-2024-001"
}
```

**Response:**

```
{
  "id_gasto": 45,
  "tipo_gasto": "ibi",
  "descripcion": "IBI 2024",
  "importe": 850.00,
  "es_proporcional": true,
  "genera_renta_negativa": false,
  "mensaje": "Gasto registrado correctamente"
}
```

### 2. Listar Gastos de Propiedad

```
GET /api/propiedades/{id}/gastos?ano=2024
```

**Response:**

```
{
  "id_propiedad": 2,
  "direccion": "Av. Barcelona 45",
  "ano": 2024,
  "gastos": [
    {
      "id_gasto": 45,
      "tipo": "ibi",

```

```

        "descripcion": "IBI 2024",
        "importe": 850.00,
        "fecha": "2024-01-15",
        "es_proporcional": true
    },
    {
        "id_gasto": 46,
        "tipo": "comunidad",
        "descripcion": "Comunidad propietarios",
        "importe": 1200.00,
        "fecha": "2024-12-15",
        "es_proporcional": true
    },
    {
        "id_gasto": 47,
        "tipo": "reparacion",
        "descripcion": "Reparación caldera",
        "importe": 3500.00,
        "fecha": "2024-06-10",
        "es_proporcional": false,
        "genera_renta_negativa": true
    }
],
"resumen": {
    "total_gastos": 5550.00,
    "gastos_proporcionales": 2050.00,
    "gastos_100_deducibles": 3500.00
}
}

```

### 3. Calcular Gastos Deducibles

POST /api/propiedades/{id}/calcular-gastos-deducibles

#### Request:

```

{
    "ano": 2024
}

```

#### Response:

```

{
    "id_propiedad": 2,
    "ano": 2024,
    "dias_alquilados": 300,
    "dias_sin_alquilar": 66,

    "gastos_proporcionales": {
        "ibi": {
            "importe_total": 850.00,

```

```

        "importe_deducible": 698.63
      },
      "comunidad": {
        "importe_total": 1200.00,
        "importe_deducible": 986.30
      },
      "seguro": {
        "importe_total": 300.00,
        "importe_deducible": 246.58
      },
      "subtotal": 1931.51
    },
    "gastos_100_deducibles": {
      "reparacion": 3500.00,
      "biuro": 250.00,
      "subtotal": 3750.00
    },
    "total_gastos_deducibles": 5681.51,
    "formula": "Proporcionales: 2350€ × (300/365) = 1931.51€ | 100%: 3750€"
  }

```

#### 4. Detectar Renta Negativa

POST /api/propiedades/{id}/verificar-renta-negativa

##### Request:

```

{
  "ano": 2024,
  "ingresos_alquiler": 14400.00
}

```

##### Response:

```

{
  "id_propiedad": 2,
  "ano": 2024,
  "ingresos": 14400.00,
  "gastos_deducibles": 5681.51,
  "amortizacion": 5586.41,
  "resultado_antes_limites": 3132.08,
  "tiene_renta_negativa": false,
  "base_imponible_final": 3132.08,
  "cuota_pagar": 595.10,
}

```

```
    "observaciones": "Renta positiva. No hay rentas negativas."
  }
```

### Ejemplo con renta negativa:

```
{
  "ingresos": 10000.00,
  "gastos_deducibles": 5000.00,
  "amortizacion": 4000.00,

  "reparaciones": 3000.00,
  "intereses_hipoteca": 500.00,

  "resultado_antes_limites": -2500.00,

  "tiene_renta_negativa": true,
  "importe_renta_negativa": 2500.00,
  "base_imponible_final": 0.00,
  "cuota_pagar": 0.00,

  "renta_negativa_guardada": {
    "id_renta_negativa": 8,
    "importe": 2500.00,
    "ano_origen": 2024,
    "ano_limite": 2028,
    "concepto": "reparaciones + intereses"
  },

  "observaciones": "Renta negativa de 2.500€. Se puede compensar hasta 2028."
}
```

## 5. Listar Rentas Negativas Pendientes

```
GET /api/clientes/{id}/rentas-negativas-pendientes
```

### Response:

```
{
  "id_cliente": 1,
  "nombre": "Piotr Kowalski",
  "rentas_negativas": [
    {
      "id_renta_negativa": 8,
      "propiedad": "Av. Barcelona 45",
      "ano_origen": 2024,
      "importe_negativo": 2500.00,
      "importe_compensado": 0.00,
      "importe_pendiente": 2500.00,
      "ano_limite": 2028,
      "anos_restantes": 4,
      "estado": "pendiente"
    }
  ]
}
```



```

    },
    {
      "id_renta_negativa": 5,
      "propiedad": "Calle Madrid 123",
      "ano_origen": 2022,
      "importe_negativo": 1800.00,
      "importe_compensado": 1200.00,
      "importe_pendiente": 600.00,
      "ano_limite": 2026,
      "anos_restantes": 2,
      "estado": "pendiente"
    }
  ],
  "total_pendiente": 3100.00
}

```

## 6. Aplicar Compensación Renta Negativa

POST /api/declaraciones/{id}/aplicar-compensacion

### Request:

```

{
  "id_renta_negativa": 8,
  "importe_a_compensar": 1500.00
}

```

### Response:

```

{
  "id_compensacion": 12,
  "id_renta_negativa": 8,
  "id_declaracion": 25,
  "importe_compensado": 1500.00,

  "renta_negativa_actualizada": {
    "importe_original": 2500.00,
    "importe_compensado_total": 1500.00,
    "importe_pendiente": 1000.00,
    "estado": "pendiente"
  },

  "declaracion_actualizada": {
    "base_imponible_antes": 2000.00,
    "compensacion_aplicada": -1500.00,
    "base_imponible_despues": 500.00,
    "cuota_pagar": 95.00
  },
}

```

```
"mensaje": "Compensación aplicada correctamente"
}
```

## LÓGICA DE CÁLCULO (BACKEND)

### Función 1: calcular\_gastos\_deducibles()

```
def calcular_gastos_deducibles(id_propiedad, ano):
    """
    Calcula el total de gastos deducibles en un año.

    Aplica prorrateo a gastos proporcionales según días alquilados.

    Returns:
        dict con gastos proporcionales, 100% y total
    """

    # Obtener propiedad
    propiedad = Propiedad.objects.get(id_propiedad=id_propiedad)

    # Obtener días alquilados (de Fase 1C)
    from .contratos import calcular_dias_alquilados
    resultado_dias = calcular_dias_alquilados(id_propiedad, ano)
    dias_alquilados = resultado_dias['total_dias_alquilados']

    if dias_alquilados == 0:
        raise ValidationError(
            f"No hay días alquilados en {ano}. "
            f"No se pueden deducir gastos."
        )

    # Obtener todos los gastos del año
    gastos = Gasto.objects.filter(
        id_propiedad=id_propiedad,
        fecha_gasto__year=ano,
        validado=True
    )

    # Clasificar gastos
    gastos_proporcionales = {}
    gastos_100_deducibles = {}

    for gasto in gastos:
        if gasto.es_proporcional:
            # Prorratear por días
            importe_deducible = gasto.importe * (dias_alquilados / 365)
            importe_deducible = round(importe_deducible, 2)

            # Agrupar por tipo
            if gasto.tipo_gasto not in gastos_proporcionales:
                gastos_proporcionales[gasto.tipo_gasto] = {
                    'importe_total': 0,
                    'importe_deducible': 0,
```

```

        'gastos': []
    }

    gastos_proporcionales[gasto.tipo_gasto]['importe_total'] += gasto.importe
    gastos_proporcionales[gasto.tipo_gasto]['importe_deducible'] += importe_deducible
    gastos_proporcionales[gasto.tipo_gasto]['gastos'].append({
        'id_gasto': gasto.id_gasto,
        'descripcion': gasto.descripcion,
        'importe_total': gasto.importe,
        'importe_deducible': importe_deducible
    })
else:
    # 100% deducible
    if gasto.tipo_gasto not in gastos_100_deducibles:
        gastos_100_deducibles[gasto.tipo_gasto] = {
            'importe': 0,
            'gastos': []
        }

    gastos_100_deducibles[gasto.tipo_gasto]['importe'] += gasto.importe
    gastos_100_deducibles[gasto.tipo_gasto]['gastos'].append({
        'id_gasto': gasto.id_gasto,
        'descripcion': gasto.descripcion,
        'importe': gasto.importe,
        'genera_renta_negativa': gasto.genera_renta_negativa
    })

# Calcular subtotales
subtotal_proporcionales = sum([
    tipo['importe_deducible']
    for tipo in gastos_proporcionales.values()
])

subtotal_100 = sum([
    tipo['importe']
    for tipo in gastos_100_deducibles.values()
])

total_gastos_deducibles = subtotal_proporcionales + subtotal_100

return {
    'id_propiedad': id_propiedad,
    'direccion': propiedad.direccion,
    'ano': ano,
    'dias_alquilados': dias_alquilados,
    'dias_sin_alquilar': 365 - dias_alquilados,
    'gastos_proporcionales': gastos_proporcionales,
    'subtotal_proporcionales': round(subtotal_proporcionales, 2),
    'gastos_100_deducibles': gastos_100_deducibles,
    'subtotal_100': round(subtotal_100, 2),
    'total_gastos_deducibles': round(total_gastos_deducibles, 2)
}

```

## Función 2: detectar\_renta\_negativa()

```
def detectar_renta_negativa(id_propiedad, id_cliente, ano, ingresos_alquiler):
    """
    Detecta si hay renta negativa y la gestiona según normativa.

    REGLA FISCAL:
    - Solo reparaciones + intereses generan rentas negativas compensables
    - Otros gastos: máximo hasta renta = 0€

    Returns:
        dict con resultado fiscal y renta negativa si aplica
    """

    # Obtener gastos deducibles
    gastos_resultado = calcular_gastos_deducibles(id_propiedad, ano)
    total_gastos = gastos_resultado['total_gastos_deducibles']

    # Obtener amortización (de Fase 1D)
    from .amortizacion import calcular_amortizacion_ano
    amort_resultado = calcular_amortizacion_ano(id_propiedad, ano)

    # Buscar amortización del propietario específico
    amortizacion = 0
    for coprop in amort_resultado['copropietarios']:
        if coprop['id_cliente'] == id_cliente:
            amortizacion = coprop['amortizacion_propietario']
            break

    # CÁLCULO PRELIMINAR
    resultado_preliminar = ingresos_alquiler - total_gastos - amortizacion

    # Si resultado >= 0, no hay renta negativa
    if resultado_preliminar >= 0:
        return {
            'id_propiedad': id_propiedad,
            'id_cliente': id_cliente,
            'ano': ano,
            'ingresos': ingresos_alquiler,
            'gastos_deducibles': total_gastos,
            'amortizacion': amortizacion,
            'resultado_preliminar': round(resultado_preliminar, 2),
            'tiene_renta_negativa': False,
            'base_imponible_final': round(resultado_preliminar, 2),
            'cuota_pagar': round(resultado_preliminar * 0.19, 2),
            'observaciones': 'Renta positiva. No hay rentas negativas.'
        }

    # HAY RENTA NEGATIVA
    # Separar gastos que generan renta negativa compensable
    gastos_renta_negativa = 0
    gastos_normales = 0

    for tipo, info in gastos_resultado['gastos_100_deducibles'].items():
        for gasto in info['gastos']:
            if gasto.get('genera_renta_negativa', False):
```

```

        gastos_renta_negativa += gasto['importe']
    else:
        gastos_normales += gasto['importe']

# Añadir intereses hipoteca (si hay)
if 'intereses_hipoteca' in gastos_resultado['gastos_proporcionales']:
    gastos_renta_negativa += gastos_resultado['gastos_proporcionales']['intereses_hipoteca']

# APLICAR LÍMITES
# Los gastos normales solo se deducen hasta que renta = 0
renta_antes_gastos_normales = ingresos_alquiler - amortizacion - gastos_renta_negativa

if renta_antes_gastos_normales <= 0:
    # Renta negativa por reparaciones/intereses
    importe_renta_negativa = abs(renta_antes_gastos_normales)
    base_imponible_final = 0

    # Guardar renta negativa
    renta_negativa = RentaNegativa.objects.create(
        id_propiedad=id_propiedad,
        id_cliente=id_cliente,
        ano_origen=ano,
        importe_negativo=importe_renta_negativa,
        concepto='reparaciones + intereses' if gastos_renta_negativa > 0 else 'intereses hipoteca',
        estado='pendiente'
    )

    return {
        'id_propiedad': id_propiedad,
        'id_cliente': id_cliente,
        'ano': ano,
        'ingresos': ingresos_alquiler,
        'amortizacion': amortizacion,
        'gastos_renta_negativa': gastos_renta_negativa,
        'gastos_normales': gastos_normales,
        'gastos_normales_deducidos': 0, # No se deducen
        'tiene_renta_negativa': True,
        'importe_renta_negativa': importe_renta_negativa,
        'base_imponible_final': 0,
        'cuota_pagar': 0,
        'renta_negativa_guardada': {
            'id_renta_negativa': renta_negativa.id_renta_negativa,
            'importe': importe_renta_negativa,
            'ano_origen': ano,
            'ano_limite': ano + 4,
            'concepto': renta_negativa.concepto
        },
        'observaciones': f'Renta negativa de {importe_renta_negativa:.2f}€. Se puede deducir hasta {base_imponible_final:.2f}€.'
    }
else:
    # Renta positiva, deducir gastos normales hasta límite
    gastos_normales_deducibles = min(gastos_normales, renta_antes_gastos_normales)
    base_imponible_final = renta_antes_gastos_normales - gastos_normales_deducibles

    return {
        'id_propiedad': id_propiedad,

```

```

'id_cliente': id_cliente,
'ano': ano,
'ingresos': ingresos_alquiler,
'amortizacion': amortizacion,
'gastos_renta_negativa': gastos_renta_negativa,
'gastos_normales': gastos_normales,
'gastos_normales_deducidos': gastos_normales_deducibles,
'tiene_renta_negativa': False,
'base_imponible_final': round(base_imponible_final, 2),
'cuota_pagar': round(base_imponible_final * 0.19, 2),
'observaciones': 'Renta positiva después de aplicar límites.'
}

```

## PANTALLAS FRONTEND

### PANTALLA 1: Tab Gastos

PROPIEDAD: Av. Barcelona 45			
Tabs: [...] [Gastos] [...]			
[TAB ACTIVO: Gastos Deducibles]			
Año: [2024 ▼]		[+ Nuevo Gasto]	
GASTOS PROPORCIONALES (Se prorratean por días alquilados)			
Tipo	Importe	Deducible	
IBI	850€	698,63€	
Comunidad	1.200€	986,30€	
Seguro	300€	246,58€	
Subtotal	2.350€	1.931,51€	
GASTOS 100% DEDUCIBLES			
Tipo	Importe	RN*	
Reparación	3.500€	✓ Sí	
Biuro	250€	No	
Subtotal	3.750€		
* RN = Genera renta negativa compensable			

TOTAL GASTOS DEDUCIBLES: 5.681,51€

Fórmula:  $2.350€ \times (300/365) = 1.931,51€$   
+ 3.750€ = 5.681,51€

## PANTALLA 2: Formulario Nuevo Gasto

NUEVO GASTO - Av. Barcelona 45

Tipo gasto \*:

▼ IBI

(IBI / Comunidad / Seguro / Reparación...)

Descripción \*:

[\_\_\_\_\_]

Importe \*:

[\_\_\_\_\_] €

Fecha gasto \*:

[\_\_/\_\_/\_\_\_\_] ☐

Periodo que cubre (opcional):

Desde: [\_\_/\_\_/\_\_\_\_] ☐

Hasta: [\_\_/\_\_/\_\_\_\_] ☐

Proveedor:

[\_\_\_\_\_]

Nº Factura:

[\_\_\_\_\_]

Factura PDF:

☐ Subir factura (opcional)

**i** Este gasto es PROPORCIONAL

Se deducirá según días alquilados

[Cancelar] [Guardar]

PANTALLA 3: Rentas Negativas Pendientes

RENTAS NEGATIVAS - Piotr Kowalski		
⚠ Rentas negativas pendientes de compensar:		
Propiedad	Año	Pendiente
Av. Barcelona 45	2024	2.500,00€
Límite: 2028 (4 años restantes)		
Concepto: Reparaciones		
Calle Madrid 123	2022	600,00€
Límite: 2026 (2 años restantes)		
Concepto: Reparaciones		
Compensado: 1.200€		
TOTAL PENDIENTE: 3.100,00€		
ⓘ Estas rentas se pueden compensar con rentas positivas de los próximos años		

VALIDACIONES IMPORTANTES

Validación 1: Importe gasto positivo

```
def validar_importe_gasto(importe):
    if importe <= 0:
        raise ValidationError("Importe debe ser mayor a 0€")

    if importe > 50000:
        raise ValidationError(
            f"Importe muy alto ({importe:.2f}€). "
            f"Revisar antes de guardar."
        )
```

Validación 2: Fecha gasto coherente

```
def validar_fecha_gasto(fecha_gasto, id_propiedad):
    propiedad = Propiedad.objects.get(id_propiedad=id_propiedad)

    # No puede ser anterior a fecha de compra
    if fecha_gasto < propiedad.fecha_adquisicion:
        raise ValidationError(
```



```

        f"Fecha gasto ({fecha_gasto}) es anterior a "
        f"fecha de compra ({propiedad.fecha_adquisicion})"
    )

    # No puede ser futura
    if fecha_gasto > date.today():
        raise ValidationError("Fecha gasto no puede ser futura")

```

### Validación 3: Clasificación tipo gasto

```

def validar_tipo_gasto(tipo_gasto):
    tipos_validos = [
        'ibi', 'comunidad', 'seguro', 'intereses_hipoteca',
        'suministros', 'conservacion', 'reparacion',
        'biuro', 'agencia', 'abogado', 'publicidad', 'otro'
    ]

    if tipo_gasto not in tipos_validos:
        raise ValidationError(
            f"Tipo gasto '{tipo_gasto}' no válido. "
            f"Debe ser uno de: {'', '.join(tipos_validos)}"
        )

```

### DATOS DE PRUEBA

```

-- Gastos para Av. Barcelona 45 (propiedad 2) en 2024

-- Gastos proporcionales
INSERT INTO gastos (
    id_propiedad, tipo_gasto, descripcion, importe, fecha_gasto,
    fecha_inicio_periodo, fecha_fin_periodo, validado
) VALUES
(2, 'ibi', 'IBI 2024', 850.00, '2024-01-15', '2024-01-01', '2024-12-31', TRUE),
(2, 'comunidad', 'Comunidad propietarios 2024', 1200.00, '2024-12-15', '2024-01-01', '2024-12-31', TRUE),
(2, 'seguro', 'Seguro hogar 2024', 300.00, '2024-02-01', '2024-02-01', '2025-01-31', TRUE);

-- Gastos 100% deducibles
INSERT INTO gastos (
    id_propiedad, tipo_gasto, descripcion, importe, fecha_gasto, validado
) VALUES
(2, 'reparacion', 'Reparación caldera', 3500.00, '2024-06-10', TRUE),
(2, 'biuro', 'Gestoría declaración 2023', 250.00, '2024-04-01', TRUE);

```

### PRUEBAS QUE DEBES HACER

## Test 1: Registrar gastos

POST /api/propiedades/2/gastos  
Crear varios gastos de distintos tipos

Verificar:

- ✓ Se clasifican automáticamente
- ✓ es\_proporcional correcto
- ✓ genera\_renta\_negativa correcto

## Test 2: Calcular gastos deducibles

POST /api/propiedades/2/calcular-gastos-deducibles  
Body: { "ano": 2024 }

Verificar:

- ✓ Gastos proporcionales  $\times (300/365)$
- ✓ Gastos 100% sin prorratear
- ✓ Total correcto

## Test 3: Detectar renta positiva

POST /api/propiedades/2/verificar-renta-negativa  
Body: {  
 "ano": 2024,  
 "ingresos\_alquiler": 14400  
}

Verificar:

- ✓ Resultado positivo
- ✓ Cuota calculada correctamente
- ✓ No se crea renta negativa

## Test 4: Detectar renta negativa

POST /api/propiedades/2/verificar-renta-negativa  
Body: {  
 "ano": 2024,  
 "ingresos\_alquiler": 8000  
}

Verificar:

- ✓ Detecta renta negativa
- ✓ Cuota = 0€
- ✓ Guarda renta negativa en BD
- ✓ año\_limite = año\_origen + 4

## Test 5: Listar rentas negativas pendientes

GET /api/clientes/1/rentas-negativas-pendientes

Verificar:

- ✓ Lista todas las rentas negativas
- ✓ Muestra años restantes
- ✓ Total pendiente correcto

## ENTREGABLES FASE 1E

### 1. Base de datos actualizada:

- Tabla gastos
- Tabla rentas\_negativas
- Tabla compensaciones\_rentas\_negativas
- Datos de prueba

### 2. Backend funcionando:

- 6 endpoints nuevos
- Función calcular\_gastos\_deducibles()
- Función detectar\_renta\_negativa()
- 3 validaciones

### 3. Frontend actualizado:

- Tab Gastos
- Formulario nuevo gasto
- Vista rentas negativas pendientes

### 4. Tests pasando:

- Los 5 tests descritos

## CRITERIOS ACEPTACIÓN

- ✓ Puedo registrar gastos de todos los tipos
- ✓ Sistema clasifica automáticamente (proporcional/100%)
- ✓ Gastos proporcionales se prorratan por días
- ✓ Sistema detecta rentas negativas correctamente
- ✓ Solo reparaciones + intereses generan RN compensables
- ✓ Rentas negativas se guardan con límite 4 años
- ✓ Frontend muestra todos los gastos y cálculos
- ✓ Validaciones funcionan

## SIGUIENTE FASE

### Fase 1F: Modelo 210 Alquiler Completo

- Generar Modelo 210 completo con todos los datos
- Ingresos - Gastos - Amortización = Base imponible
- Aplicar compensaciones rentas negativas automáticas
- Exportar PDF Modelo 210 oficial
- Declaraciones trimestrales vs anual

## RESUMEN VISUAL

### GASTOS DEDUCIBLES 2024

#### Gastos Proporcionales (300/365 días):

└─ IBI:	850€	→	698,63€	✓
└─ Comunidad:	1.200€	→	986,30€	✓
└─ Seguro:	300€	→	246,58€	✓
SUBTOTAL: 1.931,51€				

#### Gastos 100% Deducibles:

└─ Reparación:	3.500€	(genera RN)	✓
└─ Biuro:	250€		✓
SUBTOTAL: 3.750€			

TOTAL DEDUCIBLE: 5.681,51€

#### Cálculo Renta:

Ingresos:	14.400,00€
- Gastos:	-5.681,51€
- Amortiz.:	-5.586,41€

---

BASE:	3.132,08€
Cuota 19%:	595,10€

**Plazo:** 3-4 días

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