

FASE 1A - GESTIÓN DE CLIENTES (GESTORÍA)

Programa Modelo 210 para Gestoría - Primera Entrega

Duración estimada: 3-5 días

Objetivo: Crear sistema básico de gestión de CLIENTES (tu enfoque principal)

LO QUE VAS A CREAR

Un programa donde TÚ (la gestoría) gestionas a TUS CLIENTES polacos.

Flujo de trabajo:

1. Entro al programa → Veo lista de MIS CLIENTES
2. Click en un cliente → Veo SUS propiedades
3. Click en una propiedad → Veo si es IMPUTACIÓN o ALQUILER

BASE DE DATOS - SOLO 2 TABLAS PARA EMPEZAR

TABLA 1: clientes (TUS CLIENTES POLACOS)

```
CREATE TABLE clientes (  
  id_cliente SERIAL PRIMARY KEY,  
  
  -- Datos básicos  
  nie VARCHAR(12) UNIQUE NOT NULL,  
  nombre VARCHAR(100) NOT NULL,  
  apellidos VARCHAR(200) NOT NULL,  
  email VARCHAR(150) UNIQUE NOT NULL,  
  telefono VARCHAR(20) NOT NULL,  
  
  -- Dirección en Polonia  
  ciudad_polonia VARCHAR(100),  
  direccion_polonia TEXT,  
  
  -- Control  
  activo BOOLEAN DEFAULT TRUE,  
  fecha_alta TIMESTAMP DEFAULT NOW(),  
  
  -- Notas tuyas  
  notas TEXT  
);  
  
CREATE INDEX idx_nie ON clientes(nie);  
CREATE INDEX idx_nombre ON clientes(nombre);
```

¿Qué guarda? Tus clientes polacos (Piotr, Anna, Marek...)

TABLA 2: propiedades (PROPIEDADES DE TUS CLIENTES)

```
CREATE TABLE propiedades (  
    id_propiedad SERIAL PRIMARY KEY,  
  
    -- De quién es (IMPORTANTE)  
    id_cliente INT NOT NULL REFERENCES clientes(id_cliente),  
  
    -- Identificación  
    referencia_catastral VARCHAR(20) UNIQUE NOT NULL,  
    direccion VARCHAR(500) NOT NULL,  
    provincia VARCHAR(100),  
    municipio VARCHAR(100),  
    tipo VARCHAR(50) NOT NULL, -- 'vivienda', 'garaje', etc.  
  
    -- CRÍTICO: ¿Qué tipo de declaración?  
    tipo_declaracion VARCHAR(20) NOT NULL DEFAULT 'imputacion',  
    /* Valores posibles:  
        'imputacion' = Propiedad vacía/uso propio (más simple)  
        'alquiler'   = Propiedad alquilada (más complejo)  
        'mixta'      = Parte año alquilada, parte vacía  
    */  
  
    -- Datos de compra (necesarios siempre)  
    fecha_compra DATE NOT NULL,  
    precio_compra DECIMAL(12,2) NOT NULL,  
  
    -- Valores del IBI  
    valor_catastral_total DECIMAL(12,2),  
    valor_catastral_suelo DECIMAL(12,2),  
    valor_catastral_construccion DECIMAL(12,2),  
  
    -- Control  
    activa BOOLEAN DEFAULT TRUE,  
    fecha_alta TIMESTAMP DEFAULT NOW(),  
  
    -- Notas tuyas  
    notas TEXT  
);  
  
CREATE INDEX idx_cliente ON propiedades(id_cliente);  
CREATE INDEX idx_tipo_declaracion ON propiedades(tipo_declaracion);
```

¿Qué guarda? Propiedades de cada cliente con su tipo de declaración

TABLA 3: propiedad_copropietarios (SI HAY VARIOS PROPIETARIOS)

```
CREATE TABLE propiedad_copropietarios (  
  id SERIAL PRIMARY KEY,  
  
  id_propiedad INT NOT NULL REFERENCES propiedades(id_propiedad),  
  id_cliente INT NOT NULL REFERENCES clientes(id_cliente),  
  
  -- Porcentaje de cada uno  
  porcentaje DECIMAL(5,2) NOT NULL CHECK (porcentaje > 0 AND porcentaje <= 100),  
  
  fecha_inicio DATE NOT NULL,  
  activo BOOLEAN DEFAULT TRUE  
);  
  
CREATE INDEX idx_propiedad_cop ON propiedad_copropietarios(id_propiedad);
```

¿Qué guarda? Si una propiedad tiene 2-3 propietarios, aquí va el % de cada uno

TRIGGER: VALIDAR SUMA 100%

```
CREATE OR REPLACE FUNCTION validar_copropietarios()  
RETURNS TRIGGER AS $$  
DECLARE  
  suma DECIMAL;  
BEGIN  
  SELECT SUM(porcentaje) INTO suma  
  FROM propiedad_copropietarios  
  WHERE id_propiedad = NEW.id_propiedad AND activo = TRUE;  
  
  IF suma > 100.00 THEN  
    RAISE EXCEPTION 'ERROR: Suma de porcentajes = %. Debe ser máximo 100%%', suma;  
  END IF;  
  
  RETURN NEW;  
END;  
$$ LANGUAGE plpgsql;  
  
CREATE TRIGGER trg_validar_copropietarios  
BEFORE INSERT OR UPDATE ON propiedad_copropietarios  
FOR EACH ROW EXECUTE FUNCTION validar_copropietarios();
```

API BACKEND - ENDPOINTS BÁSICOS

1. Gestión de Clientes

POST	/api/clientes	- Crear nuevo cliente
GET	/api/clientes	- Listar todos tus clientes
GET	/api/clientes/{id}	- Ver un cliente específico
PUT	/api/clientes/{id}	- Actualizar datos
DELETE	/api/clientes/{id}	- Desactivar cliente

Ejemplo crear cliente:

```
POST /api/clientes
{
  "nie": "X1234567L",
  "nombre": "Piotr",
  "apellidos": "Kowalski",
  "email": "piotr@example.com",
  "telefono": "+48123456789",
  "ciudad_polonia": "Varsovia",
  "direccion_polonia": "ul. Test 123"
}
```

2. Propiedades del Cliente

POST	/api/clientes/{id}/propiedades	- Añadir propiedad a cliente
GET	/api/clientes/{id}/propiedades	- Ver propiedades del cliente
PUT	/api/propiedades/{id}	- Actualizar propiedad
DELETE	/api/propiedades/{id}	- Desactivar propiedad

Ejemplo añadir propiedad:

```
POST /api/clientes/5/propiedades
{
  "referencia_catastral": "1234567890123456ABCD",
  "direccion": "Calle Madrid 123, 2º izq",
  "provincia": "Madrid",
  "municipio": "Madrid",
  "tipo": "vivienda",
  "tipo_declaracion": "imputacion", // &lt;-- CRÍTICO
  "fecha_compra": "2020-06-15",
  "precio_compra": 250000.00,
  "valor_catastral_total": 180000.00,
  "valor_catastral_suelo": 36000.00,
  "valor_catastral_construccion": 144000.00
}
```

3. Copropietarios (si hay varios)

```
POST /api/propiedades/{id}/copropietarios - Añadir copropietario
GET /api/propiedades/{id}/copropietarios - Ver copropietarios
DELETE /api/propiedades/{id}/copropietarios/{id_cliente} - Quitar
```

Ejemplo añadir copropietario:

```
POST /api/propiedades/10/copropietarios
{
  "copropietarios": [
    {
      "id_cliente": 5,
      "porcentaje": 60.00
    },
    {
      "id_cliente": 8,
      "porcentaje": 40.00
    }
  ]
}
```

FRONTEND - PANTALLAS BÁSICAS

PANTALLA 1: Lista de Clientes (PRINCIPAL)

MIS CLIENTES	[+ Nuevo]	
Buscar: [_____]	[0]	
Nombre	NIE	Propiedades
Piotr Kowalski	X1234L	2 →
Anna Nowak	Y5678M	1 →
Marek Lewicki	Z9012N	3 →

Al hacer click en un cliente → Ir a PANTALLA 2

PANTALLA 2: Propiedades del Cliente

← Volver

CLIENTE: Piotr Kowalski

[+ Nueva Prop.]

Propiedades de Piotr:

Dirección	Tipo Decl.	
Calle Madrid 123	IMPUTACIÓN	→
Av. Barcelona 45	ALQUILER	→
Plaza Garaje 7	IMPUTACIÓN	→

Al hacer click en una propiedad → Ir a PANTALLA 3

PANTALLA 3: Detalle de Propiedad

← Volver

PROPIEDAD: Calle Madrid 123

[Editar]

Cliente: Piotr Kowalski

Ref. Catastral: 1234567890123456ABCD

Dirección: Calle Madrid 123, 28001 Madrid

Tipo inmueble: Vivienda

TIPO DECLARACIÓN: ☒ Imputación

☐ Alquiler

☐ Mixta

[Si IMPUTACIÓN → Mostrar cálculo simple]

[Si ALQUILER → Mostrar opciones contratos]

Copropietarios:

• Piotr Kowalski: 100%

[Botón: Calcular Modelo 210]

PANTALLA 4: Formulario Nuevo Cliente

NUEVO CLIENTE

NIE *:

[_____]

Nombre *:

[_____]

Apellidos *: [_____] |
Email *: [_____] |
Teléfono *: [_____] |

Ciudad (Polonia): [_____] |
Dirección: [_____] |
[_____] |

Notas: [_____] |
[_____] |

[Cancelar] [Guardar] |

PANTALLA 5: Formulario Nueva Propiedad

NUEVA PROPIEDAD PARA: Piotr Kowalski |

Ref. Catastral *: [_____] |
Dirección *: [_____] |
Provincia *: [_____] |
Municipio *: [_____] |

Tipo inmueble *: ▼ Vivienda |
(Vivienda/Garaje/...) |

TIPO DECLARACIÓN *: |
☒ Imputación (vacía/uso propio) |
☐ Alquiler (alquilada) |
☐ Mixta (parte año alquilada) |

Fecha compra *: [__/__/____] |
Precio compra *: [_____] € |

VALORES CATASTRALES (del IBI): |
Valor total: [_____] € |
Valor suelo: [_____] € |
Valor construc.: [_____] € |

¿Tiene copropietarios? |
☐ No (100% de Piotr) |
☒ Sí (varios propietarios) |

[Si Sí → Mostrar lista para añadir] |

[Cancelar] [Guardar] |

VALIDACIONES BACKEND

Validación 1: NIE único

```
def validar_nie_unico(nie):  
    if Cliente.objects.filter(nie=nie).exists():  
        raise ValidationError("Este NIE ya existe")
```

Validación 2: Email único

```
def validar_email_unico(email):  
    if Cliente.objects.filter(email=email).exists():  
        raise ValidationError("Este email ya existe")
```

Validación 3: Referencia catastral única

```
def validar_ref_catastral(ref):  
    if Propiedad.objects.filter(referencia_catastral=ref).exists():  
        raise ValidationError("Esta referencia catastral ya existe")
```

Validación 4: Valores catastrales suman

```
def validar_valores_catastrales(total, suelo, construccion):  
    suma = suelo + construccion  
    if abs(suma - total) > 0.01:  
        raise ValidationError(  
            f"Suelo ({suelo}) + Construcción ({construccion}) "  
            f"debe = Total ({total})"  
        )
```

Validación 5: Copropietarios suman 100%

```
def validar_copropietarios(copropietarios):  
    suma = sum([c['porcentaje'] for c in copropietarios])  
    if abs(suma - 100.00) > 0.01:  
        raise ValidationError(f"Suma = {suma}%. Debe ser 100%")
```

DATOS DE PRUEBA

Inserta estos datos para probar:

```
-- Cliente 1  
INSERT INTO clientes (nie, nombre, apellidos, email, telefono, ciudad_polonia)  
VALUES ('X1234567L', 'Piotr', 'Kowalski', 'piotr@test.com', '+48123456789', 'Varsovia');
```



```

-- Cliente 2
INSERT INTO clientes (nie, nombre, apellidos, email, telefono, ciudad_polonia)
VALUES ('Y7654321M', 'Anna', 'Nowak', 'anna@test.com', '+48987654321', 'Cracovia');

-- Propiedad 1 de Piotr (IMPUTACIÓN)
INSERT INTO propiedades (
    id_cliente, referencia_catastral, direccion, provincia, municipio,
    tipo, tipo_declaracion, fecha_compra, precio_compra,
    valor_catastral_total, valor_catastral_suelo, valor_catastral_construccion
) VALUES (
    1, '1234567890123456ABCD', 'Calle Madrid 123, 28001 Madrid',
    'Madrid', 'Madrid', 'vivienda', 'imputacion', '2020-06-15', 250000.00,
    180000.00, 36000.00, 144000.00
);

-- Propiedad 2 de Piotr (ALQUILER)
INSERT INTO propiedades (
    id_cliente, referencia_catastral, direccion, provincia, municipio,
    tipo, tipo_declaracion, fecha_compra, precio_compra,
    valor_catastral_total, valor_catastral_suelo, valor_catastral_construccion
) VALUES (
    1, '9876543210987654ZYXW', 'Av. Barcelona 45, 08001 Barcelona',
    'Barcelona', 'Barcelona', 'vivienda', 'alquiler', '2019-03-10', 180000.00,
    120000.00, 24000.00, 96000.00
);

-- Propiedad copropiedad (60% Piotr + 40% Anna)
INSERT INTO propiedades (
    id_cliente, referencia_catastral, direccion, provincia, municipio,
    tipo, tipo_declaracion, fecha_compra, precio_compra,
    valor_catastral_total, valor_catastral_suelo, valor_catastral_construccion
) VALUES (
    1, '5555555555555555QQQQ', 'Calle Valencia 100, 46001 Valencia',
    'Valencia', 'Valencia', 'vivienda', 'alquiler', '2021-09-01', 200000.00,
    140000.00, 28000.00, 112000.00
);

-- Copropietarios
INSERT INTO propiedad_copropietarios (id_propiedad, id_cliente, porcentaje, fecha_inicio)
VALUES
(3, 1, 60.00, '2021-09-01'), -- Piotr 60%
(3, 2, 40.00, '2021-09-01'); -- Anna 40%

```

PRUEBAS QUE DEBES HACER

Test 1: Listar clientes

```

GET /api/clientes
→ Debe devolver Piotr y Anna

```

Test 2: Ver propiedades de Piotr

```
GET /api/clientes/1/propiedades
→ Debe devolver 3 propiedades
→ Una IMPUTACIÓN, dos ALQUILER
```

Test 3: Ver copropietarios de propiedad 3

```
GET /api/propiedades/3/copropietarios
→ Debe devolver:
  - Piotr: 60%
  - Anna: 40%
```

Test 4: Intentar crear NIE duplicado

```
POST /api/clientes con NIE X1234567L
→ Debe dar ERROR: "Este NIE ya existe"
```

Test 5: Intentar copropietarios que sumen > 100%

```
POST /api/propiedades/3/copropietarios
Con: Piotr 70% + Anna 50%
→ Debe dar ERROR: "Suma = 120%. Debe ser 100%"
```

ENTREGABLES FASE 1A

Cuando termines, debes entregar:

1. **Script SQL** (fase1a_schema.sql) con:

- 3 tablas creadas
- 1 trigger
- Datos de prueba

2. **Backend funcionando** con:

- 10 endpoints básicos
- 5 validaciones implementadas

3. **Frontend funcionando** con:

- 5 pantallas básicas
- Formularios con validación

4. **Video demo (5 min)** mostrando:

- Crear cliente

- Añadir propiedad tipo IMPUTACIÓN
- Añadir propiedad tipo ALQUILER
- Añadir copropietarios
- Ver que validaciones funcionan

5. Instrucciones instalación

CRITERIOS ACEPTACIÓN

- ✓ Puedo crear clientes
- ✓ Puedo ver lista de clientes
- ✓ Puedo añadir propiedades a un cliente
- ✓ Puedo elegir tipo declaración (imputación/alquiler/mixta)
- ✓ Puedo añadir copropietarios
- ✓ Sistema valida suma copropietarios = 100%
- ✓ Frontend responsive (funciona en móvil)
- ✓ No hay errores en consola

LO QUE NO HACEMOS AÚN

En esta fase NO implementes:

- ✗ Cálculos de amortización
- ✗ Generación Modelo 210
- ✗ Gestión de contratos
- ✗ Gestión de gastos
- ✗ Subida de documentos
- ✗ Reportes

Solo la estructura básica de clientes y propiedades.

SIGUIENTE FASE

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

Fase 1B: Cálculo Imputación Renta

- Para propiedades tipo "imputación"
- Cálculo simple: 1.1% o 2% valor catastral
- Generación Modelo 210 básico

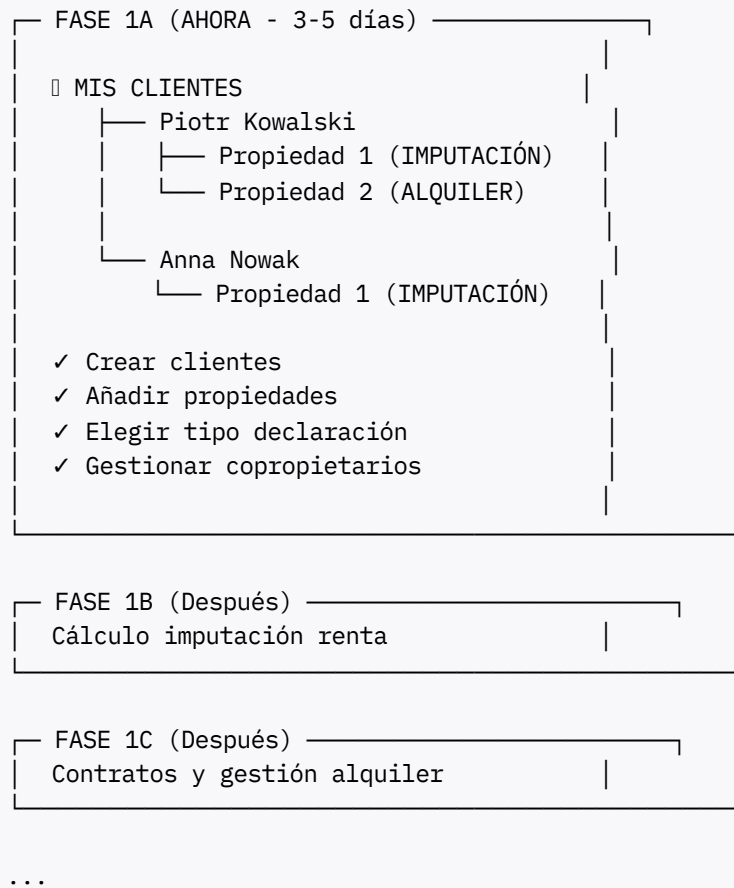
Después:

- Fase 1C: Contratos y alquiler
- Fase 1D: Gastos deducibles
- Fase 1E: Amortización

- Fase 1F: Modelo 210 completo

RESUMEN VISUAL

PROGRAMA GESTORÍA MODELO 210



INSTRUCCIONES FINALES

Tu trabajo ahora:

1. Instalar PostgreSQL
2. Crear las 3 tablas + trigger
3. Implementar los 10 endpoints
4. Crear las 5 pantallas frontend
5. Hacer las 5 pruebas
6. Grabar video demo

Plazo: 3-5 días

Cuando termines:

Me envías el código + video + instrucciones

Si todo OK:

Te doy Fase 1B (cálculo imputación)