

PRUEBAS ENDPOINTS - Documento 2: Validación y Envío

Descripción

Este documento contiene pruebas para los endpoints de **validación y envío de facturas** a Hacienda.

1. POST /api/facturas/validar

Validar estructura y estado de una factura

1.1 Prueba Exitosa - Validar por Clave

Request:

```
POST /api/facturas/validar
Content-Type: application/json

{
  "clave": "50602272020110310000100010010000000001"
}
```

Response (200 OK):

```
{
  "success": true,
  "clave": "50602272020110310000100010010000000001",
  "consecutivo": null,
  "valid": true,
  "mode": "SIMULATED",
  "timestamp": "2025-11-21T10:35:00.000Z",
  "hash": "abc123def456...",
  "mensajes": [],
  "estado": "VALIDATED"
}
```

Validaciones esperadas:

- HTTP 200 OK
- valid es true
- hash generado
- mensajes vacío (sin advertencias)

1.2 Prueba Exitosa - Validar por Consecutivo

Request:

```
POST /api/facturas/validar
Content-Type: application/json
```

```
{  
  "consecutivo": "00100010010000000001"  
}
```

Response (200 OK):

```
{  
  "success": true,  
  "clave": "50602272020110310000100010010000000001",  
  "consecutivo": "00100010010000000001",  
  "valid": true,  
  "mode": "SIMULATED",  
  "timestamp": "2025-11-21T10:36:00.000Z",  
  "hash": "abc123def456...",  
  "mensajes": [],  
  "estado": "VALIDATED"  
}
```

Validaciones esperadas:

- Sistema busca la clave desde el consecutivo
- Retorna información completa
- Validación positiva

1.3 Prueba Exitosa - Validar Payload Completo

Request:

```
POST /api/facturas/validar  
Content-Type: application/json  
  
{  
  "payload": {  
    "emisor": {  
      "nombre": "Mi Empresa",  
      "identificacion": "3101234567",  
      "tipoIdentificacion": "02",  
      "correoElectronico": "facturacion@miempresa.cn"  
    },  
    "receptor": {  
      "nombre": "Cliente Test",  
      "identificacion": "123456789",  
      "tipoIdentificacion": "01"  
    },  
    "detalleServicio": [  
      {  
        "numeroLinea": 1,  
        "cantidad": 1,  
        "unidadMedida": "UNI",  
        "detalle": "Producto Test",  
      }  
    ]  
  }  
}
```

```

    "precioUnitario": 500.00,
    "montoTotalLinea": 500.00,
    "impuestos": [
        {
            "codigo": "01",
            "tarifa": 13.00,
            "monto": 65.00
        }
    ]
},
"resumenFactura": {
    "totalVenta": 500.00,
    "totalImpuesto": 65.00,
    "totalComprobante": 565.00,
    "codigoMoneda": "CRC"
},
"condicionVenta": "Contado",
"mediosPago": ["01"]
}
}

```

Response (200 OK):

```
{
    "success": true,
    "validacionEstructural": {
        "valid": true,
        "errors": []
    },
    "modo": "ESTRUCTURAL_Y_LOGICA",
    "message": "Validación estructural y de lógica exitosa",
    "timestamp": "2025-11-21T10:37:00.000Z"
}
```

Validaciones esperadas:

- Validación estructura (campos obligatorios)
- Validación lógica (totales cuadran)
- No requiere persistencia previa
- Respuesta positiva

1.4 Prueba Fallida - Clave Inválida

Request:

```

POST /api/facturas/validar
Content-Type: application/json

{

```

```
        "clave": "12345" // Formato incorrecto
    }
```

Response (400 Bad Request):

```
{
  "success": false,
  "error": "Formato de clave inválido",
  "details": [
    {
      "field": "clave",
      "message": "La clave debe tener 50 caracteres numéricos"
    }
  ]
}
```

Validaciones esperadas:

- HTTP 400 Bad Request
- Error detallado del formato esperado

1.5 Prueba Fallida - Consecutivo No Encontrado

Request:

```
POST /api/facturas/validar
Content-Type: application/json

{
  "consecutivo": "00100010010000099999"
}
```

Response (404 Not Found):

```
{
  "success": false,
  "error": "Factura no encontrada",
  "consecutive": "00100010010000099999",
  "message": "La factura con ese consecutivo no existe en el sistema"
}
```

Validaciones esperadas:

- HTTP 404 Not Found
- Mensaje claro indicando que no existe

1.6 Prueba Fallida - Sin Parámetros Requeridos

Request:

```
POST /api/facturas/validar
Content-Type: application/json

{}
```

Response (400 Bad Request):

```
{
  "success": false,
  "error": "Debe proporcionar clave, consecutivo o payload para validar"
}
```

Validaciones esperadas:

- HTTP 400 Bad Request
- Mensaje indicando parámetros faltantes

1.7 Prueba Fallida - Payload Inválido

Request:

```
POST /api/facturas/validar
Content-Type: application/json
```

```
{
  "payload": {
    "emisor": {
      "nombre": "Test"
      // Falta identificacion (requerida)
    },
    "receptor": {
      "nombre": "Cliente"
    },
    "detalleServicio": [],
    "resumenFactura": {
      "totalVenta": 0,
      "totalComprobante": 0
    }
  }
}
```

Response (400 Bad Request):

```
{
  "success": false,
  "error": "Payload inválido",
  "details": [
    {
      "field": "emisor.identificacion",
      "message": "La identificación del emisor es requerida"
    }
  ]
}
```

```
        }
    ]
}
```

2. POST /api/facturas/enviar

Enviar factura a Hacienda para procesamiento

2.1 Prueba Exitosa - Envío Simple

Request:

```
POST /api/facturas/enviar
Content-Type: application/json

{
  "clave": "50602272020110310000100010010000000001",
  "pin": "1234"
}
```

Response (200 OK):

```
{
  "success": true,
  "clave": "50602272020110310000100010010000000001",
  "consecutivo": "00100010010000000001",
  "status": "SENT",
  "estado": "ACCEPTED",
  "message": "Factura enviada exitosamente a Hacienda",
  "sentAt": "2025-11-21T10:40:00.000Z",
  "mode": "SIMULATED",
  "haciendaResponse": {
    "code": "200",
    "message": "Comprobante recibido correctamente",
    "validationCode": "VALID_ABC123"
  }
}
```

Validaciones esperadas:

- HTTP 200 OK
- status es "SENT"
- sentAt timestamp actualizado
- Respuesta de Hacienda incluida
- Metadata actualizada en storage

2.2 Prueba - Reintentar Envío Fallido

First Request (Falla simulada):

```
POST /api/facturas/enviar
Content-Type: application/json

{
  "clave": "5060227202011031000100010010000000002",
  "pin": "0000"
}
```

Response (400 Bad Request):

```
{
  "success": false,
  "status": "REJECTED",
  "error": "PIN inválido",
  "clave": "5060227202011031000100010010000000002",
  "haciendaResponse": {
    "code": "401",
    "message": "PIN provided is invalid"
  }
}
```

Second Request (Reintento con PIN correcto):

```
POST /api/facturas/enviar
Content-Type: application/json

{
  "clave": "5060227202011031000100010010000000002",
  "pin": "1234"
}
```

Response (200 OK):

```
{
  "success": true,
  "clave": "5060227202011031000100010010000000002",
  "status": "SENT",
  "message": "Factura enviada exitosamente",
  "sentAt": "2025-11-21T10:41:00.000Z"
}
```

Validaciones esperadas:

- ✅ Primer intento rechazado con código 400
- ✅ Segundo intento exitoso con mismo consecutivo
- ✅ Sistema permite reintentos

2.3 Prueba Fallida - Clave No Existe

Request:

```
POST /api/facturas/enviar
Content-Type: application/json

{
  "clave": "50602272020110310000100010010099999",
  "pin": "1234"
}
```

Response (404 Not Found):

```
{
  "success": false,
  "error": "Factura no encontrada",
  "clave": "50602272020110310000100010010099999",
  "message": "No existe factura con esa clave en el sistema"
}
```

2.4 Prueba Fallida - PIN Faltante

Request:

```
POST /api/facturas/enviar
Content-Type: application/json

{
  "clave": "50602272020110310000100010010000000001"
}
```

Response (400 Bad Request):

```
{
  "success": false,
  "error": "Parámetros incompletos",
  "details": {
    "missing": ["pin"]
  }
}
```

2.5 Prueba Fallida - Factura Ya Enviada

Request (Intento de reenvío):

```
POST /api/facturas/enviar
Content-Type: application/json

{
  "clave": "50602272020110310000100010010000000001",
```

```
        "pin": "1234"  
    }
```

Response (409 Conflict):

```
{  
    "success": false,  
    "error": "Factura ya fue enviada",  
    "clave": "50602272020110310000100010010000000001",  
    "status": "ALREADY_SENT",  
    "sentAt": "2025-11-21T10:40:00.000Z",  
    "message": "Esta factura ya fue enviada previamente a Hacienda"  
}
```

Validaciones esperadas:

- HTTP 409 Conflict
- Aviso que fue enviada previamente
- Información de cuándo se envió

3. POST /api/english-invoices/validate

Validar factura en formato English

3.1 Prueba Exitosa - Validar Invoice por Key

Request:

```
POST /api/english-invoices/validate  
Content-Type: application/json  
  
{  
    "key": "50602272020110310000100010010000000003",  
    "consecutive": "00100010010000000003"  
}
```

Response (200 OK):

```
{  
    "success": true,  
    "method": "key_consecutive",  
    "valid": true,  
    "mode": "SIMULATED",  
    "timestamp": "2025-11-21T10:45:00.000Z",  
    "validation": {  
        "key": "50602272020110310000100010010000000003",  
        "consecutive": "00100010010000000003",  
        "hash": "xyz789abc456...",  
        "messages": [],  
        "status": "VALIDATED"
```

```
},
"metadata": {
    "language": "english",
    "validationMethod": "key_consecutive"
}
}
```

Validaciones esperadas:

- HTTP 200 OK
- Retorna en formato English
- Validación exitosa

3.2 Prueba Exitosa - Validar Invoice por Payload

Request:

```
POST /api/english-invoices/validate
```

```
Content-Type: application/json
```

```
{
    "payload": {
        "issuer": {
            "name": "Test Company",
            "identification": "3101234567"
        },
        "receiver": {
            "name": "Test Client",
            "identification": "123456789"
        },
        "lineItems": [
            {
                "lineNumber": 1,
                "quantity": 1,
                "description": "Test",
                "unitPrice": 100,
                "lineAmount": 100,
                "taxes": []
            }
        ],
        "summary": {
            "totalSales": 100,
            "totalAmount": 100,
            "currency": "CRC"
        }
    }
}
```

Response (200 OK):

```
{
  "success": true,
  "method": "payload",
  "valid": true,
  "validation": {
    "structuralValidation": {
      "valid": true,
      "errors": []
    }
  },
  "metadata": {
    "language": "english",
    "validationMethod": "payload"
  }
}
```

3.3 Prueba Fallida - Parámetros Inválidos

Request:

```
POST /api/english-invoices/validate
Content-Type: application/json

{}
```

Response (400 Bad Request):

```
{
  "success": false,
  "error": "Invalid validation request",
  "message": "Must provide either (key and consecutive) or (payload)",
  "expectedFormat": {
    "option1": { "key": "string", "consecutive": "string" },
    "option2": { "payload": "object" }
  }
}
```

Resumen de Pruebas - Documento 2

Prueba	Endpoint	Método	Esperado	Estado
1.1	/api/facturas/validar	POST	200 OK	✓
1.2	/api/facturas/validar	POST	200 OK	✓
1.3	/api/facturas/validar	POST	200 OK	✓
1.4	/api/facturas/validar	POST	400 Bad Request	✓

1.5	/api/facturas/validar	POST	404 Not Found	<input checked="" type="checkbox"/>
1.6	/api/facturas/validar	POST	400 Bad Request	<input checked="" type="checkbox"/>
1.7	/api/facturas/validar	POST	400 Bad Request	<input checked="" type="checkbox"/>
2.1	/api/facturas/enviar	POST	200 OK	<input checked="" type="checkbox"/>
2.2	/api/facturas/enviar	POST	400 + 200	<input checked="" type="checkbox"/>
2.3	/api/facturas/enviar	POST	404 Not Found	<input checked="" type="checkbox"/>
2.4	/api/facturas/enviar	POST	400 Bad Request	<input checked="" type="checkbox"/>
2.5	/api/facturas/enviar	POST	409 Conflict	<input checked="" type="checkbox"/>
3.1	/api/english-invoices/validate	POST	200 OK	<input checked="" type="checkbox"/>
3.2	/api/english-invoices/validate	POST	200 OK	<input checked="" type="checkbox"/>
3.3	/api/english-invoices/validate	POST	400 Bad Request	<input checked="" type="checkbox"/>

Próximo documento: Consulta y Listado de Facturas