Registrar Pago de Factura

Permite registrar el pago de una factura existente en el sistema. Si no se especifica la forma de pago, el sistema intentará determinarla automáticamente basándose en el cliente OAuth2 autenticado o utilizará la forma de pago por defecto configurada (Consignación bancaria).

Endpoint: https://***{{instancia}}***.arrendasoft.co/service/v2/public/invoices/register-payment

{{instancia}}: Hace referencia a la instancia de cada inmobiliaria.

Petición

Método	POST
Content-Type	application/json
Authorization	**Bearer token,** Token obtenido al consumir el servicio Login

Parámetros del Body (JSON)				
Parámetro	**Tipo**	**Obligatorio**	**Valor por defecto**	**Descripción**
factura_id	integer	Sí	-	Identificador único de la factura a la cual se registrará el pago. Debe ser un número entero positivo y coincidir con una factura en el sistema.
monto	float	Sí	-	Monto del pago a registrar. Debe ser un número positivo y no puede exceder el saldo pendiente de la factura.
fecha_pago	string	Sí	-	Fecha y hora en que se realizó el pago. Formato: Y-m-d H:i:s (ej: 2026-02-05 14:30:00).

forma_pago_id	integer	No	Auto	ID de la forma de pago. Si no se proporciona, se determina automáticamente según el cliente o la configuración por defecto.
n_comprobante	string	No	-	Número de comprobante o referencia del pago realizado. Máximo 100 caracteres. Solo se permiten letras, números, guiones (-), guiones bajos (_), puntos (.) y espacios.

Ejemplo de petición (Body JSON)

```
{
    "factura_id": 366,
    "monto": 50000,
    "fecha_pago": "2026-02-05 14:30:00",
    "forma_pago_id": 1,
    "n_comprobante": "REF123456"
}
```

Ejemplo mínimo (solo campos requeridos):

```
{
    "factura_id": 366,
    "monto": 50000,
    "fecha_pago": "2026-02-05 10:00:00"
}
```

Respuestas

Respuesta Exitosa (200)

Cuando el pago se registra correctamente, la respuesta incluye los detalles completos del pago registrado, incluyendo los IDs de los registros generados en el sistema (pago, recibo de ingreso y documento contable) así como el estado actualizado de la factura.

```
{
    "success": true,
    "status": 200,
    "message": "Pago registrado exitosamente.",
    "body": {
        "factura_id": 366,
        "pago_id": 142,
        "recibo_id": 589,
        "documento_contable_id": 1023,
        "monto_pagado": 50000,
        "fecha_pago": "2026-02-05 14:30:00",
        "forma_pago_id": 1,
        "forma_pago": "Consignación bancaria",
        "n_comprobante": "REF123456",
        "saldo_anterior": 150000,
        "saldo_actual": 100000,
        "estado_factura": "Facturada",
        "gateway": null,
        "client_name": "MiAplicacion"
    }
}
```


Claves y Descripciones del JSON de Respuesta

Clave	Tipo	Descripción
`success`	Booleano	Indica si la operación fue exitosa (true) o fallida (false).
`status`	Entero	Código HTTP de la respuesta: 200 (exitoso), 400 (error de validación), 404 (no encontrado), 422 (error de negocio), 500 (error interno).
`message`	Texto	Mensaje descriptivo del resultado de la operación.
`body`	Objeto	Contiene los detalles del pago registrado cuando la operación es exitosa. Es nulo cuando hay error.
`errors`	Array	Lista de errores de validación. Solo presente cuando status es 400.

Cuerpo de Respuesta Exitosa (body)

Cuando la operación es exitosa, el objeto `body` contiene las siguientes claves:

Clave	Tipo	Descripción
`factura_id`	Entero	ID de la factura a la que se aplicó el pago.
`pago_id`	Entero	ID del registro de pago creado en el sistema.
`recibo_id`	Entero	ID del recibo de caja generado.
`documento_contable_id`	Entero	ID del documento contable creado.
`monto_pagado`	Numérico	Monto del pago registrado.
`fecha_pago`	Texto	Fecha y hora del pago en formato Y-m-d H:i:s.
`forma_pago_id`	Entero	ID de la forma de pago utilizada.
`forma_pago`	Texto	Nombre de la forma de pago utilizada.
`n_comprobante`	Texto	Número de comprobante o referencia del pago. Máximo 100 caracteres. Nulo si no se proporcionó.
`saldo_anterior`	Numérico	Saldo de la factura antes de aplicar el pago.
`saldo_actual`	Numérico	Saldo de la factura después de aplicar el pago. Si es 0, la factura quedó totalmente pagada.
`estado_factura`	Texto	Estado de la factura después de aplicar el pago (ej: "Facturada", "Pagada").
`gateway`	Texto	Nombre de la pasarela de pago si aplica. Nulo si no se usó pasarela.
`client_name`	Texto	Nombre del cliente OAuth autenticado que realizó la petición.

Respuestas de Error

Error de Validación (400)

Se retorna cuando los parámetros enviados no cumplen con las validaciones requeridas.


```
{
    "success": false,
    "status": 400,
    "message": "Errores de validación",
    "body": null,
    "errors": [
        "El parámetro \"factura_id\" es requerido.",
```

```
        "El parámetro \"monto\" debe ser un número positivo.",
        "El parámetro \"fecha_pago\" debe tener el formato Y-m-d H:i:s (ej: 2026-02-02
14:30:00).",
        "El parámetro \"n_comprobante\" no puede exceder 100 caracteres.",
        "El parámetro \"n_comprobante\" contiene caracteres no permitidos. Solo se aceptan
letras, números, guiones, guiones bajos, puntos y espacios."
    ]
}
```

Factura No Encontrada (404)

Se retorna cuando no existe una factura con el ID proporcionado.

```
{
    "success": false,
    "status": 404,
    "message": "No se encontró la factura con el ID proporcionado.",
    "body": null
}
```

Error de Negocio (422)

Se retorna cuando la factura existe pero no puede recibir el pago por razones de negocio.

Factura no activa:

```
{
    "success": false,
    "status": 422,
    "message": "La factura no está en estado activo/pendiente de pago. Estado actual: Pagada",
    "body": null
}
```

Monto excede saldo:

```
{
    "success": false,
    "status": 422,
    "message": "El monto a pagar (100000) excede el saldo de la factura (50000).",
    "body": null
```

```
}
```


Forma de pago no determinada:


```
{
    "success": false,
    "status": 422,
    "message": "No se pudo determinar la forma de pago automáticamente. Por favor especifique el parámetro \"forma_pago_id\".",
    "body": null
}
```


Factura sin detalles:


```
{
    "success": false,
    "status": 422,
    "message": "La factura no tiene detalles asociados. No es posible registrar el pago.",
    "body": null
}
```


Forma de pago sin cuenta contable:


```
{
    "success": false,
    "status": 422,
    "message": "La forma de pago seleccionada no tiene una cuenta contable configurada. No es posible registrar el pago.",
    "body": null
}
```


Periodo contable no activo:


```
{
    "success": false,
    "status": 422,
    "message": "No existe un periodo contable activo para la fecha 2026-01-15. Verifique que el periodo esté abierto en el módulo de contabilidad.",
    "body": null
}
```

Error Interno (500)

Se retorna cuando ocurre un error inesperado en el servidor o cuando el proceso de registro del pago falla.

Error general:

```
{
    "success": false,
    "status": 500,
    "message": "Error interno del servidor al registrar el pago.",
    "body": null
}
```

Error al generar recibo de ingreso:

```
{
    "success": false,
    "status": 500,
    "message": "Error al crear el recibo de ingreso para el pago de la factura.",
    "body": null
}
```

Pago procesado sin confirmación de ID:

```
{
    "success": false,
    "status": 500,
    "message": "El pago se procesó pero no se pudo obtener el ID del registro de pago.
Verifique manualmente en el sistema.",
    "body": null
}
```

Códigos de Respuesta HTTP

Código	Descripción
200	Pago registrado exitosamente. La respuesta incluye los IDs del pago, recibo y documento contable generados.
400	Error de validación en los parámetros enviados, forma de pago inexistente, o n_comprobante con formato inválido.

Código	Descripción
404	La factura especificada no existe.
422	Error de regla de negocio: factura no activa, monto excede saldo, factura sin detalles, forma de pago sin cuenta contable, periodo contable no activo, o no se pudo determinar forma de pago.
500	Error interno del servidor, error al generar recibo de ingreso, o pago procesado sin confirmación de ID.

Notas Importantes

1. **Determinación Automática de Forma de Pago:** Si no se proporciona `forma_pago_id`, el sistema intentará:
   - Primero, determinar la forma de pago basándose en el cliente OAuth autenticado (integración con pasarelas de pago).
   - Si no encuentra una pasarela asociada, buscará la forma de pago por defecto configurada con código DIAN 42 (Consignación bancaria).
2. **Estados de Factura:** Solo se pueden registrar pagos en facturas con estado "Facturada" (estado_id = 1). Los estados posibles son:
   - 1: Facturada (permite pagos)
   - 2: Anulada
   - 3: Pagada
   - 5: Borrador
   - 6: Anulada por NC
3. **Pagos Parciales:** El sistema permite pagos parciales. El monto debe ser mayor a cero y no puede exceder el saldo pendiente de la factura. Tras un pago parcial, la respuesta incluye `saldo_anterior` y `saldo_actual` para verificar la diferencia.
4. **Formas de Pago:** Para obtener el listado de formas de pago disponibles, consulte el endpoint Listar Formas de Pago.
5. **Verificación del Pago:** La respuesta exitosa incluye `pago_id`, `recibo_id` y `documento_contable_id` que permiten verificar que el pago, el recibo de ingreso y el asiento contable se crearon correctamente en el sistema.
6. **Estado de la Factura:** El campo `estado_factura` en la respuesta indica el estado actualizado de la factura tras el pago. Si el saldo queda en 0, la factura pasará automáticamente a estado "Pagada".
7. **Validación de n_comprobante:** El número de comprobante tiene las siguientes restricciones:
   - Máximo 100 caracteres.
   - Solo se permiten: letras (a-z, A-Z), números (0-9), guiones (-), guiones bajos (_), puntos (.) y espacios.
   - No se permiten caracteres especiales como comillas, signos de mayor/menor, punto y coma, etc.

8. **Periodo Contable:** La fecha del pago debe corresponder a un periodo contable activo en el sistema. Si la fecha está fuera de un periodo activo, se retornará un error 422 indicando que debe verificarse la configuración de periodos en el módulo de contabilidad.
9. **Validación de forma_pago_id:** Si se proporciona el parámetro `forma_pago_id`, debe ser un número entero positivo.

Ejemplos

Curl

```
curl -X POST "https://{{instancia}}.arrendasoft.co/service/v2/public/invoices/register-payment" \
  -H "Content-Type: application/json" \
  -H "Authorization: Bearer YOUR_AUTHENTICATION_TOKEN" \
  -d '{
    "factura_id": 366,
    "monto": 50000,
    "fecha_pago": "2026-02-05 10:00:00",
    "n_comprobante": "REF123456"
  }'
```

PHP Curl

```
<?php

$instance = 'your_instance'; // Reemplace con su instancia actual
$token = 'YOUR_AUTHENTICATION_TOKEN'; // Reemplace con el token obtenido desde el servicio de login

// Datos del pago a registrar
$paymentData = [
    'factura_id' => 366,
    'monto' => 50000,
    'fecha_pago' => '2026-02-05 14:30:00',
    'forma_pago_id' => 1, // Opcional
    'n_comprobante' => 'REF123456' // Opcional
```

```
];

$url = "https://{$instance}.arrendasoft.co/service/v2/public/invoices/register-payment";

$ch = curl_init($url);
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
curl_setopt($ch, CURLOPT_POST, true);
curl_setopt($ch, CURLOPT_POSTFIELDS, json_encode($paymentData));
curl_setopt($ch, CURLOPT_HTTPHEADER, [
    'Content-Type: application/json',
    'Authorization: Bearer ' . $token
]);

$response = curl_exec($ch);
$http_code = curl_getinfo($ch, CURLINFO_HTTP_CODE);

if (curl_errno($ch)) {
    echo 'Error: ' . curl_error($ch);
} else {
    echo "HTTP Status Code: " . $http_code . "\n";
    echo "Response: " . $response . "\n";

    // Procesar la respuesta
    $result = json_decode($response, true);

    if ($result['success']) {
        echo "Pago registrado exitosamente.\n";
        echo "Factura: " . $result['body']['factura_id'] . "\n";
        echo "Pago ID: " . $result['body']['pago_id'] . "\n";
        echo "Recibo ID: " . $result['body']['recibo_id'] . "\n";
        echo "Documento Contable ID: " . $result['body']['documento_contable_id'] . "\n";
        echo "Monto pagado: " . $result['body']['monto_pagado'] . "\n";
        echo "Forma de pago: " . $result['body']['forma_pago'] . "\n";
        echo "Saldo anterior: " . $result['body']['saldo_anterior'] . "\n";
        echo "Saldo actual: " . $result['body']['saldo_actual'] . "\n";
        echo "Estado factura: " . $result['body']['estado_factura'] . "\n";
    } else {
        echo "Error: " . $result['message'] . "\n";
        if (isset($result['errors'])) {
```

```
            foreach ($result['errors'] as $error) {
                echo "- " . $error . "\n";
            }
        }
    }
}
curl_close($ch);

?>
```

JavaScript Fetch

```
const instance = 'your_instance'; // Reemplace con su instancia actual
const token = 'YOUR_AUTHENTICATION_TOKEN'; // Reemplace con el token obtenido

const paymentData = {
    factura_id: 366,
    monto: 50000,
    fecha_pago: '2026-02-05 14:30:00',
    n_comprobante: 'REF123456'
};

fetch(`https://${instance}.arrendasoft.co/service/v2/public/invoices/register-payment`, {
    method: 'POST',
    headers: {
        'Content-Type': 'application/json',
        'Authorization': `Bearer ${token}`
    },
    body: JSON.stringify(paymentData)
})
.then(response => response.json())
.then(data => {
    if (data.success) {
        console.log('Pago registrado exitosamente');
        console.log('Factura:', data.body.factura_id);
        console.log('Pago ID:', data.body.pago_id);
        console.log('Recibo ID:', data.body.recibo_id);
```

```
        console.log('Documento Contable ID:', data.body.documento_contable_id);
        console.log('Monto pagado:', data.body.monto_pagado);
        console.log('Forma de pago:', data.body.forma_pago);
        console.log('Saldo anterior:', data.body.saldo_anterior);
        console.log('Saldo actual:', data.body.saldo_actual);
        console.log('Estado factura:', data.body.estado_factura);
    } else {
        console.error('Error:', data.message);
        if (data.errors) {
            data.errors.forEach(error => console.error('-', error));
        }
    }
})
.catch(error => console.error('Error de conexión:', error));
```

Revisión #9
Creado el 5 febrero 2026 16:29:27 por Isabel Higuita
Actualizado el 2 julio 2026 21:22:57 por Isabel Higuita