

# Documentação Técnica: Gerador de Pix Simplificado (Apenas Frontend)

**Autor:** Manus AI **Data:** 15 de Novembro de 2025 **Versão:** 3.0 (Foco em Uso Interno - Frontend Only)

## 1. Aviso de Segurança Crítico

**ATENÇÃO:** A API da Black Cat Pagamentos exige autenticação com sua **Chave Secreta (SECRET\_KEY)**. Ao implementar a comunicação diretamente no Frontend (JavaScript no navegador), a Chave Secreta será **exposta** no código-fonte acessível a qualquer usuário.

*Esta abordagem é altamente insegura e deve ser utilizada APENAS em ambientes de uso interno e controlado, onde o risco de comprometimento da chave é mitigado pela restrição de acesso. Para qualquer aplicação pública ou externa, o uso de um Backend para proteger a Chave Secreta é obrigatório.*

## 2. Objetivo e Componentes

Este guia demonstra como criar um **Gerador de Pix Simplificado** que exige **apenas o valor** do usuário, implementado exclusivamente com **HTML e JavaScript**.

## 2.1. Componentes Utilizados

Componente	Função	Observação
HTML	Estrutura da interface.	Simples formulário e área de resultado.
JavaScript	Lógica de autenticação, comunicação com a API e manipulação do DOM.	<b>Contém a Chave Secreta exposta.</b>
qrcode.js	Biblioteca JavaScript para gerar o QR Code visual no lado do cliente.	Evita a necessidade de um Backend para gerar a imagem.

## 3. Estratégia de Dados Mínimos

Para que o usuário forneça apenas o valor, o código JavaScript deve preencher os campos obrigatórios (`customer` e `items`) com dados fixos e genéricos, conforme a estratégia anterior.

Campo	Valor Fixo (Preenchido pelo JS)
<code>paymentMethod</code>	"pix"
<code>items[0].title</code>	"Pagamento Pix Interno"
<code>customer.name</code>	"Cliente Interno"
<code>customer.document.number</code>	<b>CPF/CNPJ Válido Genérico</b>

## 4. Implementação (HTML e JavaScript)

O código a seguir deve ser salvo em um único arquivo HTML (ex: `gerador_pix_interno.html`).

## 4.1. Código Completo (HTML + JavaScript)

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <title>Gerador de Pix Interno (Frontend Only)</title>
    <script
src="https://cdn.jsdelivr.net/npm/qrcode.js@0.0.2/qrcode.min.js"></script>
    <style>
        body { font-family: Arial, sans-serif; padding: 20px; }
        #result-area { border: 1px solid #ccc; padding: 20px; margin-top: 20px; display: none; }
        #pix-code { width: 100%; resize: none; margin-top: 10px; }
        .warning { color: red; font-weight: bold; border: 2px solid red; padding: 10px; margin-bottom: 20px; }
    </style>
</head>
<body>
    <div class="warning">
        AVISO DE SEGURANÇA: Esta ferramenta expõe a Chave Secreta da Black
        Cat no código-fonte. Use APENAS para fins internos.
    </div>

    <h1>Gerador de Pix Simplificado</h1>

    <div id="input-area">
        <label for="amount">Valor (R$):</label>
        <input type="number" id="amount" step="0.01" min="0.01"
placeholder="Ex: 10.50">
        <button onclick="generatePix()">Gerar Pix</button>
    </div>

    <div id="result-area">
        <h2>Pix Gerado</h2>
        <div id="qrcode-container"></div>

        <p><strong>Pix Copia e Cola:</strong></p>
        <textarea id="pix-code" rows="3" readonly></textarea>
        <button onclick="copyPix()">Copiar Código</button>

        <p id="expiration-info"></p>
        <p id="status-info">Aguardando pagamento...</p>
    </div>
```

```
<script>
  // =====
  // ATENÇÃO: CHAVES EXPOSTAS! Mantenha este arquivo em ambiente
seguro.
  // =====
  const PUBLIC_KEY = 'SUA_CHAVE_PUBLICA'; // Substitua pela sua chave
  const SECRET_KEY = 'SUA_CHAVE_SECRETA'; // Substitua pela sua chave
  const API_URL =
'https://api.blackcatpagamentos.com/v1/transactions';

  // Dados genéricos para satisfazer a API
  const GENERIC_CPF = '000000000000'; // Substitua por um CPF/CNPJ
válido para testes
  const WEBHOOK_URL = 'https://seu-dominio.com/webhook-pix'; //
Webhook não será processado aqui, mas é obrigatório no payload

  function getAuthHeader() {
    const credentials = `${PUBLIC_KEY}:${SECRET_KEY}`;
    const encoded = btoa(credentials); // btoa é a função nativa
para Base64
    return `Basic ${encoded}`;
  }

  async function generatePix() {
    const amountInput = document.getElementById('amount');
    const amount = parseFloat(amountInput.value);

    if (isNaN(amount) || amount <= 0) {
      alert('Por favor, insira um valor válido.');
      return;
    }

    const amountInCents = Math.round(amount * 100);
    const externalRef = `pix_interno_${Date.now()}`;

    const payload = {
      amount: amountInCents,
      paymentMethod: "pix",
      postbackUrl: WEBHOOK_URL,
      externalRef: externalRef,
      customer: {
        name: "Cliente Interno",
        email: "pix_interno@seuprojeto.com",
        document: {
          number: GENERIC_CPF,
          type: "cpf"
        }
      }
    }
  }
}
```

```
        }
    },
    items: [
        {
            externalRef: "servico-pix",
            title: "Pagamento Pix Interno",
            unitPrice: amountInCents,
            quantity: 1,
            tangible: false
        }
    ]
};

try {
    const response = await fetch(API_URL, {
        method: 'POST',
        headers: {
            'Authorization': getAuthHeader(),
            'Content-Type': 'application/json'
        },
        body: JSON.stringify(payload)
    });

    const data = await response.json();

    if (response.ok && data.status === 'waiting_payment' &&
data.pix && data.pix.qrcode) {
        const pixCode = data.pix.qrcode;

        // 1. Gerar o QR Code visual (usando qrcode.js)
        const qrcodeContainer = document.getElementById('qrcode-
container');
        qrcodeContainer.innerHTML = ''; // Limpa o anterior
        new QRCode(qrcodeContainer, {
            text: pixCode,
            width: 200,
            height: 200,
            colorDark : "#000000",
            colorLight : "#ffffff",
            correctLevel : QRCode.CorrectLevel.H
        });

        // 2. Exibir o Pix Copia e Cola
        document.getElementById('pix-code').value = pixCode;

        // 3. Exibir a data de expiração
        const expiration = new
Date(data.pix.expirationDate).toLocaleString('pt-BR');
    }
}
```

```

        document.getElementById('expiration-info').textContent =
`Válido até: ${expiration}`;

        // 4. Mostrar a área de resultado
        document.getElementById('result-area').style.display =
'block';
        document.getElementById('status-info').textContent =
'Aguardando pagamento... (Acompanhe a confirmação no painel Black Cat)';

    } else {
        console.error('Erro na API Black Cat:', data);
        alert('Falha ao gerar Pix. Verifique o console para
detalhes.');
    }
} catch (error) {
    console.error('Erro de conexão:', error);
    alert('Erro de conexão com a API. Verifique sua rede.');
}
}

function copyPix() {
    const pixCode = document.getElementById('pix-code');
    pixCode.select();
    document.execCommand('copy');
    alert('Código Pix Copia e Cola copiado!');
}
</script>
</body>
</html>

```

## 5. Confirmação de Pagamento (Uso Interno)

Como não há Backend, o Frontend **não pode** receber o webhook de confirmação de pagamento.

### 5.1. Rastreamento Manual

A confirmação de pagamento deve ser feita **manualmente** pela equipe interna, monitorando o painel da Black Cat ou o sistema que recebe o webhook configurado no campo `postbackUrl`.

O campo `externalRef` (ex: `pix_interno_TIMESTAMP`) gerado no JavaScript pode ser usado para localizar a transação no painel da Black Cat.

## 5.2. Fluxo de Confirmação

1. O usuário interno gera o Pix e envia para o pagador.
  2. O pagador realiza o pagamento.
  3. A equipe interna acessa o painel da Black Cat e busca a transação pelo `externalRef` ou ID.
  4. A equipe interna confirma o status `paid` (pago).
- 

## 6. Boas Práticas e Limitações

Aspecto	Detalhe
Segurança	<b>Risco Alto:</b> A Chave Secreta está exposta. Mantenha o arquivo em um servidor local ou em um ambiente de acesso restrito (ex: VPN interna).
Confirmação	<b>Limitação:</b> Não há confirmação automática em tempo real. A confirmação é manual via painel ou sistema de webhook externo.
Dependência	O código depende da biblioteca <code>qrcode.js</code> carregada via CDN.
Dados Mínimos	A estratégia de preenchimento de dados genéricos ( <code>customer</code> e <code>items</code> ) é mantida para que o usuário precise apenas do valor.
Webhook	O campo <code>postbackurl</code> é enviado na requisição, mas o Frontend não o processa. Ele é necessário para que a Black Cat envie a notificação para algum lugar (seu sistema de rastreamento).

---

## 7. Conclusão

A implementação de um Gerador de Pix Simplificado **apenas com Frontend** é tecnicamente possível, mas **compromete a segurança** ao expor a Chave Secreta. Dado que o uso é restrito à equipe interna, este guia fornece o código completo e as

instruções para que a ferramenta funcione, com a ressalva de que a confirmação de pagamento será manual.