

```

pŷ ===== .dockerignore =====
pŷ node_modules
pŷ npm-debug.log
pŷ .git
pŷ Dockerfile
pŷ .dockerignore
pŷ
pŷ ===== .gitignore =====
pŷ node_modules/
pŷ dist/
pŷ
pŷ ===== Dockerfile =====
pŷ FROM node:20-alpine
pŷ
pŷ WORKDIR /app
pŷ COPY package*.json ./
pŷ # Usar install (nao 'ci') pois nao temos pack
pŷ RUN npm install
pŷ
pŷ COPY . .
pŷ
pŷ # Gera o build estatico do Vite. Mantemos as
pŷ # instaladas para permitir reexecuçoes de bui
pŷ # container final sem a necessidade de reinst
pŷ RUN npm run build
pŷ
pŷ ENV NODE_ENV=production
pŷ ENV PORT=3000
pŷ EXPOSE 3000
pŷ
pŷ CMD [ "npm", "start" ]
pŷ
pŷ ===== docker-compose.yml =====
pŷ services:
pŷ   sitepix:
pŷ     build:
pŷ       context: .
pŷ       dockerfile: Dockerfile
pŷ     restart: always
pŷ     environment:
pŷ       NODE_ENV: production
pŷ       PORT: ${PORT:-3000}
pŷ       PUSHINPAY_TOKEN: ${PUSHINPAY_TOKEN}
pŷ     healthcheck:
pŷ       test: [ "CMD", "wget", "-qO-", "http://1
pŷ       interval: 10s
pŷ       timeout: 3s
pŷ       retries: 5
pŷ
pŷ ===== package.json =====
pŷ {
pŷ   "name": "sitepix",
pŷ   "private": true,
pŷ   "version": "1.0.0",
pŷ   "type": "module",

```

```

pŷ "scripts": {
pŷ   "dev": "vite",
pŷ   "build": "vite build",
pŷ   "start": "node server.js"
pŷ },
pŷ "devDependencies": {
pŷ   "vite": "^5.4.0"
pŷ },
pŷ "dependencies": {
pŷ   "express": "^4.18.2",
pŷ   "cors": "^2.8.5",
pŷ   "axios": "^1.6.7"
pŷ }
pŷ }
pŷ
pŷ ===== public/index.html =====
pŷ <!doctype html>
pŷ <html lang="pt-BR">
pŷ <head>
pŷ   <meta charset="utf-8" />
pŷ   <meta name="viewport" content="width=device
pŷ   <title>Gerar PIX</title>
pŷ   <style>
pŷ     :root {
pŷ       color-scheme: light dark;
pŷ       --border-radius: 12px;
pŷ       --border-color: rgba(0, 0, 0, 0.12);
pŷ       --ok: #0a7a3a;
pŷ       --err: #b00020;
pŷ       --bg-muted: rgba(0, 0, 0, 0.04);
pŷ       --transition: 150ms ease;
pŷ     }
pŷ
pŷ     body {
pŷ       font-family: system-ui, -apple-system,
pŷ       max-width: 760px;
pŷ       margin: 40px auto;
pŷ       padding: 0 16px 64px;
pŷ       line-height: 1.45;
pŷ     }
pŷ
pŷ     header {
pŷ       margin-bottom: 24px;
pŷ     }
pŷ
pŷ     h1 {
pŷ       margin-bottom: 4px;
pŷ     }
pŷ
pŷ     p.subtle {
pŷ       margin: 0;
pŷ       font-size: 15px;
pŷ       color: rgba(0, 0, 0, 0.65);
pŷ     }
pŷ

```

```

pŷ      . card {
pŷ        border: 1px solid var(--border-color);
pŷ        border-radius: var(--border-radius);
pŷ        padding: 20px;
pŷ        margin: 20px 0;
pŷ        background: rgba(255, 255, 255, 0.9);
pŷ        backdrop-filter: blur(6px);
pŷ        box-shadow: 0 20px 40px rgba(15, 23, 42
pŷ      }
pŷ
pŷ      fieldset {
pŷ        border: none;
pŷ        padding: 0;
pŷ        margin: 0 0 18px;
pŷ      }
pŷ
pŷ      fieldset:last-of-type {
pŷ        margin-bottom: 0;
pŷ      }
pŷ
pŷ      legend {
pŷ        font-weight: 600;
pŷ        margin-bottom: 12px;
pŷ      }
pŷ
pŷ      label {
pŷ        display: block;
pŷ        margin: 10px 0 4px;
pŷ        font-weight: 500;
pŷ      }
pŷ
pŷ      input,
pŷ      button,
pŷ      textarea {
pŷ        width: 100%;
pŷ        padding: 12px;
pŷ        border: 1px solid #ccc;
pŷ        border-radius: 8px;
pŷ        font-size: 16px;
pŷ        transition: box-shadow var(--transition
pŷ      }
pŷ
pŷ      textarea {
pŷ        resize: vertical;
pŷ      }
pŷ
pŷ      input:focus,
pŷ      textarea:focus {
pŷ        outline: none;
pŷ        border-color: #2563eb;
pŷ        box-shadow: 0 0 0 3px rgba(37, 99, 235,
pŷ      }
pŷ
pŷ      button {
pŷ        cursor: pointer;

```

```

pŷ         font-weight: 600;
pŷ         border: none;
pŷ         background: #2563eb;
pŷ         color: #fff;
pŷ         display: inline-flex;
pŷ         justify-content: center;
pŷ         align-items: center;
pŷ         gap: 8px;
pŷ     }
pŷ
pŷ     button.secondary {
pŷ         background: var(--bg-muted);
pŷ         color: inherit;
pŷ         border: 1px solid var(--border-color);
pŷ     }
pŷ
pŷ     button[disabled] {
pŷ         opacity: 0.6;
pŷ         cursor: progress;
pŷ     }
pŷ
pŷ     .row {
pŷ         display: grid;
pŷ         grid-template-columns: repeat(auto-fit,
pŷ         gap: 14px;
pŷ     }
pŷ
pŷ     .ok {
pŷ         color: var(--ok);
pŷ     }
pŷ
pŷ     .err {
pŷ         color: var(--err);
pŷ         white-space: pre-wrap;
pŷ     }
pŷ
pŷ     .qr {
pŷ         display: grid;
pŷ         place-items: center;
pŷ         margin-top: 20px;
pŷ     }
pŷ
pŷ     .qr img {
pŷ         max-width: min(260px, 100%);
pŷ         border-radius: 16px;
pŷ         border: 1px solid var(--border-color);
pŷ         padding: 12px;
pŷ         background: #fff;
pŷ     }
pŷ
pŷ     .actions {
pŷ         display: flex;
pŷ         flex-wrap: wrap;
pŷ         gap: 12px;
pŷ         margin-top: 12px;

```

```

pŷ    }
pŷ
pŷ    code.inline {
pŷ        font-size: 0.95em;
pŷ        padding: 2px 6px;
pŷ        border-radius: 6px;
pŷ        background: var(--bg-muted);
pŷ    }
pŷ
pŷ    small.helper {
pŷ        display: block;
pŷ        margin-top: 4px;
pŷ        color: rgba(0, 0, 0, 0.55);
pŷ    }
pŷ    </style>
pŷ </head>
pŷ <body>
pŷ     <header>
pŷ         <h1>Gerar PIX</h1>
pŷ         <p class="subtle">
pŷ             Servidor: <code class="inline" id="heal
pŷ             • Token: <code class="inline" id="token
pŷ         </p>
pŷ     </header>
pŷ
pŷ     <div class="card">
pŷ         <form id="pixForm" novalidate>
pŷ             <fieldset>
pŷ                 <legend>Dados do pagamento</legend>
pŷ                 <label for="valor">Valor (R$)</label>
pŷ                 <input id="valor" type="text" inputmo
pŷ                 <small class="helper">Informe apenas
pŷ
pŷ                 <div class="row">
pŷ                     <div>
pŷ                         <label for="webhook">Webhook (opc
pŷ                         <input id="webhook" type="url" pl
pŷ                     </div>
pŷ                     <div>
pŷ                         <label for="splitRules">Split rul
pŷ                         <textarea id="splitRules" rows="3
pŷ                         <small class="helper">Deixe em br
pŷ                     </div>
pŷ                 </div>
pŷ             </fieldset>
pŷ
pŷ             <div class="actions">
pŷ                 <button id="btnSubmit" type="submit">
pŷ                 <button id="btnLimpar" class="seconda
pŷ             </div>
pŷ         </form>
pŷ     </div>
pŷ
pŷ     <div id="resultado" class="card" style="dis
pŷ         <p><strong>Status:</strong> <span id="sta

```

```

pŷ      < p   i d = " v a l o r R e s u m o "   c l a s s = " s u b t l e " > < / p >
pŷ      < d i v   c l a s s = " q r "   i d = " q r W r a p p e r "   h i d d e n >
pŷ      <   i m g   i d = " q r I m g "   a l t = " Q R   C o d e   P I X "   / >
pŷ      < / d i v >
pŷ      < l a b e l   s t y l e = " m a r g i n - t o p : 1 2 p x "   f o r = " q r T e x
pŷ      < t e x t a r e a   i d = " q r T e x t "   r o w s = " 5 "   r e a d o n l y > <
pŷ      < d i v   c l a s s = " a c t i o n s " >
pŷ      <   b u t t o n   i d = " b t n C o p i a r "   t y p e = " b u t t o n " > C o
pŷ      <   b u t t o n   i d = " b t n N o v o "   c l a s s = " s e c o n d a r y "
pŷ      < / d i v >
pŷ      < p   i d = " e r r o "   c l a s s = " e r r "   s t y l e = " d i s p l a y : n
pŷ      < / d i v >
pŷ
pŷ      < s c r i p t   t y p e = " m o d u l e " >
pŷ      c o n s t   $   =   ( i d )   =>   d o c u m e n t . g e t E l e m e n t B y I d ( i
pŷ
pŷ      c o n s t   e l s   =   {
pŷ      f o r m :   $ ( " p i x F o r m " ) ,
pŷ      s u b m i t :   $ ( " b t n S u b m i t " ) ,
pŷ      c l e a r :   $ ( " b t n L i m p a r " ) ,
pŷ      n o v o :   $ ( " b t n N o v o " ) ,
pŷ      c o p i a r :   $ ( " b t n C o p i a r " ) ,
pŷ      r e s u l t a d o :   $ ( " r e s u l t a d o " ) ,
pŷ      s t a t u s :   $ ( " s t a t u s " ) ,
pŷ      e r r o :   $ ( " e r r o " ) ,
pŷ      q r I m g :   $ ( " q r I m g " ) ,
pŷ      q r W r a p p e r :   $ ( " q r W r a p p e r " ) ,
pŷ      q r T e x t :   $ ( " q r T e x t " ) ,
pŷ      v a l o r :   $ ( " v a l o r " ) ,
pŷ      v a l o r R e s u m o :   $ ( " v a l o r R e s u m o " ) ,
pŷ      w e b h o o k :   $ ( " w e b h o o k " ) ,
pŷ      s p l i t :   $ ( " s p l i t R u l e s " ) ,
pŷ      h e a l t h :   $ ( " h e a l t h " ) ,
pŷ      t o k e n S t a t u s :   $ ( " t o k e n S t a t u s " ) ,
pŷ      } ;
pŷ
pŷ      c o n s t   c u r r e n c y F o r m a t t e r   =   n e w   I n t l . N u m b e r F o
pŷ      s t y l e :   " c u r r e n c y " ,
pŷ      c u r r e n c y :   " B R L " ,
pŷ      m i n i m u m F r a c t i o n D i g i t s :   2 ,
pŷ      } ) ;
pŷ
pŷ      c o n s t   s e t L o a d i n g   =   ( i s L o a d i n g )   =>   {
pŷ      e l s . s u b m i t . d i s a b l e d   =   i s L o a d i n g ;
pŷ      e l s . s u b m i t . t e x t C o n t e n t   =   i s L o a d i n g   ?   " G e r
pŷ      } ;
pŷ
pŷ      c o n s t   s h o w E r r o r   =   ( m e s s a g e )   =>   {
pŷ      e l s . s t a t u s . t e x t C o n t e n t   =   " e r r o " ;
pŷ      e l s . e r r o . s t y l e . d i s p l a y   =   " b l o c k " ;
pŷ      e l s . e r r o . t e x t C o n t e n t   =   m e s s a g e ;
pŷ      } ;
pŷ
pŷ      c o n s t   r e a i s P a r a C e n t a v o s   =   ( t x t )   =>   {
pŷ      i f   ( ! t x t )   r e t u r n   0 ;

```

```

pŷ      const normalized = txt
pŷ      .toString()
pŷ      .trim()
pŷ      .replace( / \s+ / g , " " )
pŷ      .replace( / \. / g , " " )
pŷ      .replace( " , " , " . " )
pŷ      .replace( / [ ^ \d . \ - ] / g , " " ) ;
pŷ
pŷ      const value = Number.parseFloat(normalized)
pŷ      if (!Number.isFinite(value)) return NaN;
pŷ      return Math.round(value * 100);
pŷ  } ;
pŷ
pŷ  const parseSplitRules = (text) => {
pŷ      const trimmed = text.trim();
pŷ      if (!trimmed) return [];
pŷ      try {
pŷ          const parsed = JSON.parse(trimmed);
pŷ          if (!Array.isArray(parsed)) {
pŷ              throw new Error("O JSON precisa ser u
pŷ          }
pŷ          return parsed;
pŷ      } catch (err) {
pŷ          throw new Error('Split rules inválidas:
pŷ      }
pŷ  } ;
pŷ
pŷ  const resetResultado = () => {
pŷ      els.resultado.style.display = "none";
pŷ      els.erro.style.display = "none";
pŷ      els.erro.textContent = "";
pŷ      els.status.textContent = "";
pŷ      els.qrWrapper.hidden = true;
pŷ      els.qrImg.src = "";
pŷ      els.qrText.value = "";
pŷ      els.valorResumo.textContent = "";
pŷ  } ;
pŷ
pŷ  const atualizarHealth = async () => {
pŷ      try {
pŷ          const response = await fetch("/health")
pŷ          if (!response.ok) throw new Error("Falh
pŷ          const payload = await response.json();
pŷ          els.health.textContent = payload.ok ? "
pŷ          els.tokenStatus.textContent = payload.h
pŷ          els.tokenStatus.className = payload.has
pŷ      } catch (err) {
pŷ          els.health.textContent = "indisponível"
pŷ          els.tokenStatus.textContent = "indispon
pŷ          els.tokenStatus.className = "inline err
pŷ      }
pŷ  } ;
pŷ
pŷ  els.form.addEventListener("submit", async (
pŷ      event.preventDefault();

```

```

pŷ      setLoading( true );
pŷ      els.resultado.style.display = "block" ;
pŷ      els.erro.style.display = "none" ;
pŷ      els.erro.textContent = " " ;
pŷ      els.status.textContent = "gerando&" ;
pŷ      els.qrWrapper.hidden = true;
pŷ      els.qrImg.removeAttribute( "src" );
pŷ      els.qrText.value = " " ;
pŷ
pŷ      const valueInCents = reaisParaCentavos( el
pŷ      if ( !Number.isFinite( valueInCents ) || val
pŷ          setLoading( false );
pŷ          showError( "Informe um valor válido maio
pŷ          return;
pŷ      }
pŷ
pŷ      let splitRules;
pŷ      try {
pŷ          splitRules = parseSplitRules( els.split.
pŷ      } catch (err) {
pŷ          setLoading( false );
pŷ          showError( err.message );
pŷ          return;
pŷ      }
pŷ
pŷ      try {
pŷ          const response = await fetch( "/pix", {
pŷ              method: "POST",
pŷ              headers: { "Content-Type": "applicati
pŷ              body: JSON.stringify( {
pŷ                  value: valueInCents,
pŷ                  webhook_url: els.webhook.value.trim
pŷ                  split_rules: splitRules,
pŷ              } ),
pŷ          } );
pŷ
pŷ      const data = await response.json();
pŷ
pŷ      if ( !response.ok ) {
pŷ          const reason = typeof data === "object
pŷ          throw new Error( reason );
pŷ      }
pŷ
pŷ      els.status.textContent = data.status ??
pŷ      els.valorResumo.textContent = `Valor so
pŷ
pŷ      if ( data.qr_code_base64 ) {
pŷ          els.qrWrapper.hidden = false;
pŷ          els.qrImg.src = data.qr_code_base64;
pŷ      }
pŷ
pŷ      if ( data.qr_code ) {
pŷ          els.qrText.value = data.qr_code;
pŷ      }
pŷ

```



```

pŷ         if (!data.qr_code && !data.qr_code_base
pŷ             showError("A API não retornou um QR C
pŷ         }
pŷ     } catch (err) {
pŷ         showError(String(err.message || err));
pŷ     } finally {
pŷ         setLoading(false);
pŷ     }
pŷ }) ;
pŷ
pŷ els.copiar.addEventListener("click", async
pŷ     const texto = els.qrText.value.trim();
pŷ     if (!texto) return;
pŷ     try {
pŷ         await navigator.clipboard.writeText(tex
pŷ         window.alert("Código PIX copiado para a
pŷ     } catch (err) {
pŷ         window.alert("Não foi possível copiar a
pŷ         console.warn("Clipboard copy failed", e
pŷ     }
pŷ }) ;
pŷ
pŷ els.novo.addEventListener("click", () => {
pŷ     resetResultado();
pŷ     els.valor.focus();
pŷ }) ;
pŷ
pŷ els.clear.addEventListener("click", () => {
pŷ     els.form.reset();
pŷ     els.split.value = " ";
pŷ     resetResultado();
pŷ }) ;
pŷ
pŷ atualizarHealth();
pŷ </script>
pŷ </body>
pŷ </html>
pŷ
pŷ ===== server.js =====
pŷ // server.js
pŷ import express from "express";
pŷ import cors from "cors";
pŷ import path from "path";
pŷ import fs from "fs";
pŷ import { fileURLToPath } from "url";
pŷ import axios from "axios";
pŷ
pŷ const __filename = fileURLToPath(import.meta.
pŷ const __dirname = path.dirname(__filename);
pŷ
pŷ const app = express();
pŷ app.use(cors());
pŷ app.use(express.json());
pŷ
pŷ const distDir = path.join(__dirname, "dist");

```

```

pŷ const publicDir = path.join(__dirname, "public");
pŷ const hasDistBuild = fs.existsSync(path.join(
pŷ
pŷ if (!hasDistBuild) {
pŷ     console.warn(
pŷ         "&p Build do front-end não encontrado em
pŷ     );
pŷ }
pŷ
pŷ // Servir o build do Vite (pasta dist) quando
pŷ app.use(express.static(hasDistBuild ? distDir
pŷ
pŷ // Healthcheck para o front
pŷ app.get("/health", (req, res) => {
pŷ     res.json({ ok: true, hasToken: Boolean(proc
pŷ });
pŷ
pŷ // Endpoint que gera PIX via PushinPay
pŷ app.post("/pix", async (req, res) => {
pŷ     try {
pŷ         const token = process.env.PUSHINPAY_TOKEN
pŷ         if (!token) return res.status(500).json({
pŷ
pŷ         const { value, webhook_url, split_rules }
pŷ
pŷ         const valueInCents = Number.parseInt(valu
pŷ         if (!Number.isFinite(valueInCents) || val
pŷ             return res.status(400).json({ error: "V
pŷ         }
pŷ
pŷ         const body = {
pŷ             value: valueInCents,
pŷ             webhook_url: webhook_url || undefined,
pŷ             split_rules: Array.isArray(split_rules)
pŷ         };
pŷ
pŷ         const { data } = await axios.post(
pŷ             "https://api.pushinpay.com.br/api/pix/c
pŷ             body,
pŷ             {
pŷ                 headers: {
pŷ                     Authorization: `Bearer ${token}`,
pŷ                     Accept: "application/json",
pŷ                     "Content-Type": "application/json",
pŷ                 },
pŷ                 timeout: 15000,
pŷ             }
pŷ         );
pŷ
pŷ         // Tenta padronizar campos usados no fron
pŷ         res.json({
pŷ             status: "ok",
pŷ             qr_code: data?.payload || data?.qrCode
pŷ             qr_code_base64: data?.qrCodeBase64 || n
pŷ             raw: data,

```

```

pŷ      } ) ;
pŷ    } catch ( error ) {
pŷ      const status = error.response?.status ;
pŷ      const data = error.response?.data ;
pŷ      console.error( " Erro  /pix: " , status , data
pŷ      res.status( 500 ).json( { error: " Erro  ao ge
pŷ    }
pŷ  } ) ;
pŷ
pŷ // SPA fallback (opcional, se tiver rotas de
pŷ app.get( "*" , ( req , res ) => {
pŷ   res.sendFile( path.join( hasDistBuild ? distD
pŷ } ) ;
pŷ
pŷ const PORT = process.env.PORT || 3000 ;
pŷ const server = app.listen( PORT , ( ) => {
pŷ   console.log( '\033[32m Servidor rodando na porta $
pŷ } ) ;
pŷ
pŷ server.on( "error" , ( err ) => {
pŷ   if ( err?.code === " EADDRINUSE " ) {
pŷ     console.error(
pŷ       '\033[31m Porta ${PORT} j  est  em uso. Verifi
pŷ     ) ;
pŷ     process.exit( 1 ) ;
pŷ   }
pŷ
pŷ   console.error( "\033[31m Erro inesperado ao iniciar
pŷ   process.exit( 1 ) ;
pŷ } ) ;
pŷ
pŷ
pŷ ===== vite.config.js =====
pŷ import { defineConfig } from "vite" ;
pŷ
pŷ export default defineConfig( {
pŷ   root: "public" , // aqui est  seu index.html
pŷ   build: {
pŷ     outDir: "../dist" ,
pŷ     emptyOutDir: true
pŷ   }
pŷ } ) ;
pŷ

```