

Relatório Técnico de Arquitetura e Implementação: Plataforma Web para Gestão de Matrimônio com Integração PIX e Design Minimalista

1. Introdução e Análise de Requisitos

A concepção de uma plataforma digital para a gestão de um evento matrimonial transcende a simples criação de uma página web informativa; trata-se de arquitetar um ponto central de convergência para a logística, a comunicação emocional e as transações financeiras de um dos eventos mais complexos da vida pessoal. O presente relatório detalha a engenharia necessária para desenvolver uma aplicação web de alto desempenho, utilizando as tecnologias mais modernas do ecossistema JavaScript — especificamente Next.js, Vercel e GitHub — para atender a requisitos estéticos rigorosos e funcionalidades críticas de pagamento via PIX.¹

O cenário apresentado solicita um equilíbrio delicado: a interface deve evocar a serenidade e a elegância de uma estética minimalista em tons de bege e marrom (referência visual da Imagem 1), enquanto o "motor" da aplicação deve ser robusto o suficiente para gerar payloads financeiros em conformidade com os padrões do Banco Central do Brasil (BACEN) e do EMVCo.⁴ Diferentemente de sites de casamento genéricos baseados em templates prontos, a solução proposta aqui é uma aplicação *serverless* personalizada, capaz de escalar, oferecer tempos de carregamento instantâneos via Edge Network e garantir a segurança dos dados dos convidados.¹

1.1 Definição do Escopo e Persona do Usuário

O projeto visa atender aos noivos Gabriel e Milleny, cujo casamento está agendado para 22 de

agosto de 2026.⁸ A persona do usuário final (o convidado) varia desde jovens nativos digitais até familiares idosos com menor letramento tecnológico. Portanto, a arquitetura de UX (Experiência do Usuário) deve priorizar a clareza, a acessibilidade e a responsividade móvel, dado que mais de 80% dos acessos a sites de eventos ocorrem via smartphones.

A funcionalidade de "Lista de Presentes Fictícia" introduz uma camada de complexidade técnica: a conversão de uma intenção de presente (ex: "Jantar Romântico") em uma transação financeira real e imediata via PIX, sem a necessidade de intermediários de e-commerce que cobram taxas percentuais sobre os presentes. Isso exige uma implementação precisa do padrão BR Code para geração de QR Codes Estáticos com valores definidos.⁵

2. Arquitetura Tecnológica e Infraestrutura

A escolha da *stack* tecnológica é fundamentada na necessidade de alta performance, baixo custo de manutenção e facilidade de atualização contínua. A arquitetura Jamstack (JavaScript, APIs e Markup) é a escolha ideal para este cenário.

2.1 O Ecossistema Next.js e Vercel

A utilização do framework **Next.js** (versão 14 ou superior, utilizando o App Router) sobre a infraestrutura da **Vercel** não é apenas uma conveniência, mas uma decisão estratégica de engenharia.¹

Renderização Híbrida e Performance:

Sites de casamento possuem seções estáticas (história do casal, local, fotos) e seções dinâmicas (contagem regressiva, formulário de RSVP, geração de PIX). O Next.js permite a Static Site Generation (SSG) para as páginas informativas, garantindo que o HTML seja pré-renderizado no momento do build. Isso resulta em um Time to First Byte (TTFB) excepcionalmente baixo, essencial para manter a atenção do usuário em conexões móveis instáveis (como convidados acessando o site na estrada para o local do evento).¹

Para as funcionalidades dinâmicas, como o RSVP, utilizaremos *Server Actions* ou *API Routes*. A Vercel hospeda essas funções como *Serverless Functions*, que escalam automaticamente sob demanda. O plano "Hobby" (gratuito) da Vercel oferece limites generosos (100GB de largura de banda, milhares de execuções de funções) que cobrem confortavelmente o tráfego

de um casamento médio a grande.⁷

Integração Contínua (CI/CD) com GitHub:

O repositório no GitHub atuará como a fonte da verdade. A integração nativa Vercel-GitHub dispara um novo build e deploy a cada push na branch main. Isso permite que alterações urgentes (como uma mudança de horário na cerimônia) sejam propagadas globalmente em minutos, sem a necessidade de FTP ou processos manuais de upload.³

2.2 Estratégia de Persistência de Dados

Embora soluções baseadas em planilhas (como Google Sheets via API) sejam populares em templates simples ³, uma abordagem profissional exige maior integridade de dados para o RSVP.

Banco de Dados Relacional Serverless:

Recomenda-se o uso de um banco de dados PostgreSQL gerenciado, como o Vercel Postgres ou Supabase. A estrutura relacional garante que as confirmações de presença sejam atômicas e consistentes. Diferente de uma planilha, que pode sofrer com condições de corrida (dois usuários editando ao mesmo tempo), um banco SQL garante a integridade transacional.

Além disso, o Supabase oferece uma camada de API instantânea e autenticação se necessário, embora para um site de casamento público, a autenticação seja geralmente dispensada em favor de um formulário aberto protegido por rate limiting e honeypots.¹

2.3 Estrutura de Pastas e Organização do Código

Para manter o projeto manutenível e escalável, a estrutura de diretórios deve seguir as convenções do Next.js App Router, separando claramente a lógica de negócio da interface do usuário.

Diretório	Propósito
/app	Contém as rotas da aplicação (páginas e API).
/components	Componentes React reutilizáveis (Botões, Cards, Modals).

/lib	Lógica de negócios pura (Gerador de PIX, Validadores).
/public	Ativos estáticos (Imagens otimizadas, ícones).
/styles	Configurações globais de CSS e definições do Tailwind.

3. Engenharia Financeira: Implementação do PIX (BR Code)

A funcionalidade mais crítica e tecnicamente desafiadora deste projeto é a implementação da lista de presentes. O objetivo é permitir que o convidado selecione um item (ex: "Cotas para a Lua de Mel - R\$ 200,00") e o sistema gere instantaneamente um código PIX Copia e Cola e um QR Code correspondente para pagamento direto na conta dos noivos, sem intermediários.⁴

3.1 Anatomia do Payload EMVCo MPM

O PIX utiliza o padrão **BR Code**, que é uma implementação específica do *EMV QR Code Specification for Payment Systems - Merchant Presented Mode* (EMV QRCPS-MPM). Para gerar um código válido sem depender de APIs bancárias externas (que exigiriam autenticação complexa e certificados digitais), deve-se construir um **QR Code Estático**.

O payload do PIX é uma string de texto construída concatenando objetos de dados TLV (*Type-Length-Value*). Cada objeto possui:

1. **ID (Type):** 2 dígitos numéricos.
2. **Length:** 2 dígitos indicando o tamanho do valor.
3. **Value:** O conteúdo do dado.

A estrutura hierárquica obrigatória para um PIX Estático funcional inclui os seguintes campos⁶:

ID	Nome do Campo	Formato	Descrição Técnica
00	Payload Format Indicator	"01"	Versão fixa do protocolo. Obrigatório.
01	Point of Initiation Method	"11" ou "12"	"11" se o QR Code for reutilizável (estático), "12" se for dinâmico. Usaremos "12" se quisermos simular unicidade ou "11" para simplicidade.
26	Merchant Account Information	Aninhado	Contém a GUI e a Chave PIX.
52	Merchant Category Code	"0000"	Código de categoria do comerciante. "0000" é genérico.
53	Transaction Currency	"986"	Código ISO 4217 para o Real Brasileiro (BRL).
54	Transaction Amount	String	O valor da transação (ex: "100.00"). Opcional, mas essencial para a nossa lista de presentes.
58	Country Code	"BR"	Código ISO 3166-1 alpha 2 do país.
59	Merchant Name	Alfanumérico	Nome do beneficiário.

			Máximo 25 caracteres. Deve ser sem acentos (best practice).
60	Merchant City	Alfanumérico	Cidade do beneficiário. Máximo 15 caracteres. Sem acentos.
62	Additional Data Field	Aninhado	Campo para TXID (Identificador da Transação).
63	CRC	Hex	Checksum CRC-16 calculado sobre todo o payload.

3.2 Detalhamento Granular do Campo 26 (Merchant Account Info)

Este é o campo onde a maioria das implementações falha. O ID 26 é um contêiner para outros objetos TLV.

Ele deve conter obrigatoriamente:

- **ID 00 (GUI):** Globally Unique Identifier. Para o PIX, o valor é fixo: br.gov.bcb.pix.
 - TLV: 0014br.gov.bcb.pix (ID 00, Len 14, Val br.gov.bcb.pix).
- **ID 01 (Chave):** A chave PIX do recebedor (Email, CPF, Telefone ou Aleatória).
 - Exemplo: Chave de email noivos@email.com.
 - TLV: 0116noivos@email.com.

Cálculo do Tamanho do Campo 26:

O tamanho total do campo 26 é a soma dos caracteres dos sub-campos.

No exemplo acima: 0014br.gov.bcb.pix (18 chars) + 0116noivos@email.com (20 chars) = 38 caracteres.

Logo, o TLV do campo 26 completo seria: 26380014br.gov.bcb.pix0116noivos@email.com.6

3.3 O Algoritmo CRC-16/CCITT-FALSE

O último campo (ID 63) é o CRC (Cyclic Redundancy Check). O padrão EMVCo exige especificamente o algoritmo **CRC-16/CCITT-FALSE**. Muitas bibliotecas de CRC padrão (como CRC-16/ARC ou XMODEM) gerarão códigos inválidos, pois utilizam parâmetros de inicialização diferentes.¹⁴

Parâmetros Matemáticos:

- **Polinômio:** $0x1021$ ($x^{16} + x^{12} + x^5 + 1$)
- **Valor Inicial:** $0xFFFF$ (Todos os bits em 1).
- **XOR de Saída:** $0x0000$ (Sem inversão final).
- **Reflexão de Entrada/Saída:** False (Bits processados do mais significativo para o menos significativo).

Implementação em JavaScript Puro:

Para evitar dependências pesadas de NPM no client-side (o que aumentaria o bundle size e prejudicaria a performance mobile), implementamos o cálculo bit a bit manualmente.¹⁴

JavaScript

```
/**
 * Calcula o CRC-16/CCITT-FALSE para o payload do PIX.
 * @param {string} payload - A string do payload completa (incluindo "6304" no final).
 * @returns {string} - O CRC calculado em formato Hexadecimal (4 caracteres).
 */
function calculateCRC16(payload) {
  const polynomial = 0x1021;
  let crc = 0xFFFF; // Inicialização conforme ISO/IEC 13239

  for (let i = 0; i < payload.length; i++) {
    // O byte de entrada é deslocado para o topo do registrador (bits 8-15)
    // e XORado com o byte alto atual do CRC.
    let c = payload.charCodeAt(i);
    crc ^= (c << 8);

    for (let j = 0; j < 8; j++) {
      // Verifica se o bit mais significativo (MSB) é 1
      if ((crc & 0x8000) !== 0) {
        // Se sim, desloca à esquerda e aplica o polinômio

```

```

    crc = (crc << 1) ^ polynomial;
  } else {
    // Se não, apenas desloca à esquerda
    crc = crc << 1;
  }
}
}

// Máscara final para garantir 16 bits e formatação Hex Uppercase
return (crc & 0xFFFF).toString(16).toUpperCase().padStart(4, '0');
}

```

Processo de Geração do Payload Final:

1. Concatene todos os campos de 00 a 62.
2. Adicione o ID do CRC e o tamanho: 6304.
3. Execute a função calculateCRC16 sobre toda essa string.
4. Anexe o resultado (4 caracteres hex) ao final.

Esta lógica deve residir em um utilitário isolado (/lib/pix.ts) para ser reutilizada em qualquer componente.¹⁷

3.4 Mecânica de "Copia e Cola" e QR Code Visual

A experiência do usuário no mobile exige atenção especial. Se o convidado está acessando o site pelo celular onde o aplicativo do banco está instalado, ele **não consegue escanear o QR Code** exibido na tela. Portanto, a funcionalidade "Pix Copia e Cola" é prioritária.¹⁹

- **QR Code Visual:** Utilize a biblioteca qrcode.react ou qrcode para renderizar o payload gerado como um canvas ou SVG. SVGs são preferíveis por serem vetoriais e escalarem sem perda de qualidade em telas Retina.
- **Copia e Cola:** Utilize a API navigator.clipboard.writeText(payload). É crucial fornecer feedback visual imediato (ex: ícone de check ou *toast notification* "Código copiado!") para confirmar a ação ao usuário.

4. Design System e Estética Minimalista

A referência visual fornecida (Imagem 1) aponta para uma estética "Beige/Brown" minimalista,

frequentemente associada a estilos como *Organic Modern* ou *Japandi*. A tradução dessa estética para código CSS exige rigor na escolha de tipografia, espaçamento e paleta de cores.²¹

4.1 Psicologia das Cores e Paleta

O bege e o marrom evocam sentimentos de conforto, estabilidade e sofisticação natural. No entanto, no web design, cores claras podem causar problemas de contraste e legibilidade se não forem calibradas corretamente.

Definição da Paleta (Tailwind CSS Config):

Não utilizaremos cores padrão. Criaremos um tema estendido no `tailwind.config.ts` para garantir consistência.

TypeScript

```
// tailwind.config.ts
export default {
  theme: {
    extend: {
      colors: {
        wedding: {
          bg: '#F5F1E6', // Um bege claro "Off-white" quente, menos severo que #FFF.
          card: '#EBE5CE', // Bege médio para superfícies e cartões.
          primary: '#5D4037', // Marrom café profundo para texto principal (Alto Contraste).
          accent: '#A67B5B', // Terracota suave para botões e destaques.
          muted: '#8C7C71', // Taupe para textos secundários e bordas sutis.
        }
      },
    },
    fontFamily: {
      serif: ['var(--font-playfair)', 'serif'], // Para títulos (Elegância).
      sans: ['var(--font-lato)', 'sans-serif'], // Para corpo (Legibilidade).
    }
  }
}
```

A escolha de #5D4037 para o texto principal sobre o fundo #F5F1E6 garante conformidade com as diretrizes WCAG AA para acessibilidade, algo frequentemente negligenciado em designs "pastel".²²

4.2 Textura e Ruído Visual

A Imagem 1 sugere uma textura de papel ou granulação ("grainy texture"). Em CSS, isso pode ser emulado para adicionar profundidade e evitar que o design pareça "plástico" ou excessivamente digital.

Podemos aplicar um pseudo-elemento `::before` fixo na tela com uma imagem de ruído SVG e opacidade reduzida (2-4%) e `mix-blend-mode: overlay`. Isso unifica todo o site com uma sensação tátil de papelaria de luxo.²³

4.3 Tipografia Hierárquica

A tipografia é o elemento de design mais forte em convites de casamento.

- **Títulos (Display):** Utilizaremos a fonte *Playfair Display* ou *Cormorant Garamond*. Essas fontes serifadas de alto contraste imitam a gravação tradicional. Devem ser usadas em tamanhos grandes para nomes e datas.
- **Corpo de Texto (Sans):** Para informações logísticas (endereços, instruções de RSVP), uma fonte sem serifa geométrica e limpa como *Lato* ou *Montserrat* é necessária para legibilidade em telas pequenas.²⁴

5. Implementação das Funcionalidades Específicas

5.1 Funcionalidade 1: Informações do Casamento (Baseadas no Convite)

A seção "Hero" (topo do site) deve refletir digitalmente o convite físico (Imagem 2).

- **Componente Hero:** Deve ocupar 100% da altura da viewport (min-h-screen no Tailwind) para criar impacto.
- **Dados:** Nomes "Gabriel & Milleny", Data "Sábado, 22 de Agosto de 2026", Horário "14:30".
- **Design Responsivo:** Em desktop, pode-se usar um layout de tela dividida (foto à esquerda, texto à direita). No mobile, o layout empilhado (imagem topo, texto abaixo) é mandatório para manter a hierarquia visual.²⁵

5.2 Funcionalidade 2: Confirmação de Presença (RSVP)

O formulário de RSVP deve ser simples, mas seguro contra spam.

Campos:

- Nome Completo (Input texto).
- Email (Input email).
- Confirmação (Radio Button: "Sim, estarei lá" / "Infelizmente não poderei ir").
- Número de Acompanhantes (Select: 0 a 4).

Validação e Segurança:

Utilizaremos a biblioteca Zod para validação de esquema no lado do servidor. Isso previne injeção de dados maliciosos.

- *Honeypot:* Um campo oculto (input type="text" name="fax" class="hidden") deve ser incluído. Se preenchido (por bots que preenchem tudo automaticamente), o backend rejeita a submissão silenciosamente.
- *Persistência:* Os dados são enviados para a API Route do Next.js, que conecta ao Supabase para inserção. O feedback visual ("Obrigado por confirmar!") deve ser imediato, sem recarregar a página (AJAX).²⁶

5.3 Funcionalidade 3: Lista de Presentes (A Lógica Fictícia)

Esta é a aplicação prática da engenharia PIX detalhada na Seção 3.

- **UX da Lista:** Apresentar os presentes como "Cartões" com foto, título e valor. O botão de ação deve ser "Presentear" ou "Contribuir".
- **Fluxo do Usuário:**
 1. Clique em "Presentear".
 2. Abre-se um Modal (para não perder o contexto da página).

3. O Modal exibe: "Para presentear com este valor, utilize o PIX abaixo".
 4. Exibe o QR Code e o código copia-e-cola gerados dinamicamente com o valor do item.
 5. Instrução clara: "O valor será transferido diretamente para a conta dos noivos."
- **Gerenciamento de Estado:** Usar React Context ou Zustand para gerenciar qual presente está selecionado e controlar a abertura do modal, evitando "prop drilling" excessivo.³

5.4 Funcionalidade 4: Contagem Regressiva

O desafio técnico aqui é a "Hidratação" do React. O servidor (Vercel) e o cliente (Navegador) podem ter horários ligeiramente diferentes, causando erros de renderização (Hydration Mismatch).

Solução Técnica:

O componente Countdown deve ser um Client Component ('use use client') que só renderiza o tempo após a montagem inicial (useEffect).

- Utilize a biblioteca date-fns para manipulação robusta de datas e fusos horários. É crucial definir o fuso horário do casamento (Brasília -03:00) explicitamente para que a contagem esteja correta para convidados acessando de outros países.

5.5 Funcionalidade 5: Fotos e Localização (Google Maps)

A integração de mapas deve ser funcional e estética.

Google Maps Embed API:

A forma mais eficiente e econômica é usar a Embed API.

URL:

https://www.google.com/maps/embed/v1/place?key=API_KEY&q=Spazio+Giardino+Londrina.

- **Otimização:** Adicionar o atributo loading="lazy" ao iframe para não bloquear o carregamento inicial da página.
- **Estilização:** O mapa padrão do Google (azul/verde/amarelo) pode conflitar com o design bege. Uma técnica avançada é aplicar um filtro CSS no iframe: filter: grayscale(100%) sepia(20%) contrast(90%);. Isso dessatura o mapa e adiciona um tom sépia, integrando-o perfeitamente ao tema visual sem custos adicionais de estilização de API.²⁸

Fotos:

Utilizar o componente `<Image />` do Next.js é obrigatório. Ele converte automaticamente as fotos pesadas do casamento para formatos modernos (WebP/AVIF) e dimensiona conforme o dispositivo, economizando dados móveis dos convidados e acelerando o LCP (Largest Contentful Paint).²⁵

5.6 Funcionalidade 6: Área de Recados (Email)

Para o envio de mensagens, não é necessário um banco de dados complexo. A integração direta com um serviço de envio de emails transacionais é mais eficiente.

Resend (Serviço de Email):

A Vercel recomenda o uso do Resend. Com o SDK do Resend, a API Route pode receber a mensagem do formulário e despachar um email formatado em HTML para a caixa de entrada dos noivos.

- Limites: O plano gratuito permite até 3.000 emails/mês, o que é mais que suficiente para qualquer casamento (mesmo com 300 convidados enviando múltiplas mensagens).²⁶

6. Estrutura de Código e Implementação Mobile/Web

O desenvolvimento deve seguir a filosofia *Mobile-First*. O código CSS (Tailwind) é escrito pensando primeiro na tela pequena, e depois expandido para telas maiores usando prefixos (md:, lg:).

6.1 Responsividade e Breakpoints

O design da Imagem 1, com muito espaço em branco, traduz-se bem para o mobile se os espaços forem ajustados.

- **Padrão de Margem:** px-4 (16px) no mobile, px-8 ou px-12 em tablets, e um container centralizado com max-w-5xl no desktop.
- **Navegação:**
 - *Desktop:* Menu horizontal no topo, fixo ou estático.
 - *Mobile:* Menu "Hambúrguer" ou, preferencialmente para este caso, uma barra de navegação inferior (*Bottom Navigation*) fixa com ícones para as seções principais

(Início, Local, Presentes, RSVP), facilitando o uso com uma mão.

6.2 Organização do Repositório e Deploy

1. **Setup Inicial:** `npx create-next-app@latest casamento-gabriel-milleny --typescript --tailwind --eslint`.
 2. **Configuração de Ambiente:** Criar um arquivo `.env.local` para armazenar chaves sensíveis (Chave PIX, API Google Maps, Chaves Supabase/Resend). **Nunca comitar este arquivo no GitHub.**
 3. **Deploy na Vercel:** Conectar a conta Vercel ao GitHub. A Vercel detectará automaticamente o framework Next.js. Configurar as variáveis de ambiente no painel da Vercel (Settings > Environment Variables).
-

7. Conclusão e Visão de Futuro

A construção deste site de casamento representa um exercício sofisticado de desenvolvimento web moderno. Ao optar por uma arquitetura baseada em **Next.js** e **Vercel**, garante-se uma fundação sólida, rápida e escalável. A implementação manual do gerador de **PIX (BR Code)** demonstra domínio técnico sobre protocolos financeiros e elimina a dependência de gateways de pagamento custosos, oferecendo valor real aos noivos.

Esteticamente, a aplicação rigorosa de princípios de design minimalista, apoiada por um sistema de design **Tailwind** personalizado, assegura que o produto final não será apenas funcional, mas uma extensão digital elegante da identidade visual do casamento. O resultado é uma plataforma profissional, responsiva e perene, que serve não apenas para o dia do evento, mas como um arquivo digital das memórias do casal.

7.1 Recomendações Pós-Evento

A natureza efêmera de um site de casamento sugere um planejamento para o "Day After".

- **Modo Arquivo:** Após a data do casamento, o site pode ser convertido em uma galeria de fotos. O formulário de RSVP e a lista de presentes devem ser desativados via *feature flags* no código ou variáveis de ambiente (`NEXT_PUBLIC_ENABLE_RSVP=false`).
- **SEO e Privacidade:** É recomendável configurar o arquivo `robots.txt` para impedir a

indexação do site por motores de busca (Google), garantindo a privacidade dos dados e fotos do casal, a menos que a visibilidade pública seja desejada.

Esta arquitetura fornece aos noivos total controle, zero custos de transação financeira e uma experiência de usuário superior, cumprindo todos os requisitos técnicos e estéticos propostos.¹

Tabela Comparativa: Vercel vs. Hospedagem Tradicional para Eventos

Característica	Hospedagem Tradicional (cPanel/PHP)	Vercel (Next.js/Serverless)	Vantagem para o Casamento
Custo	Mensal fixo (R\$ 20-50/mês)	Gratuito (Tier Hobby)	Economia total para o casal.
Escalabilidade	Limitada ao servidor contratado. Pode cair com picos de acesso (envio de convites).	Escalabilidade automática (Serverless). Suporta picos massivos.	Site não cai quando todos acessam ao mesmo tempo.
Performance	Servidor em local único (latência alta).	Edge Network (CDN Global).	Carregamento instantâneo para convidados de qualquer lugar.
Atualizações	Upload manual via FTP (lento, propenso a erro).	Git Push (Automático).	Correção rápida de erros (ex: endereço errado).
Backend	Requer configuração de banco/PHP.	Serverless Functions integradas.	Menor complexidade de gestão.

Referências citadas

1. Vercel as a hosting platform: When It's the best choice and when to look elsewhere, acessado em novembro 20, 2025, <https://focusreactive.com/when-to-host-on-vercel-and-when-not/>
2. MattiasHenders/wedding-template: The Wedding Website Template provides a simple, elegant solution for couples looking to create a wedding website with ease. Built with React and designed for easy customization, it offers features like RSVP management, gift registry display, interactive maps, and mobile-friendly design. - GitHub, acessado em novembro 20, 2025, <https://github.com/MattiasHenders/wedding-template>
3. rampatra/wedding-website - GitHub, acessado em novembro 20, 2025, <https://github.com/rampatra/wedding-website>
4. Generate PIX Payment QR Codes for Any Brazilian Bank Key | n8n workflow template, acessado em novembro 20, 2025, <https://n8n.io/workflows/3704-generate-pix-payment-qr-codes-for-any-brazilian-bank-key/>
5. Pix and QR Codes - Pismo Developers Portal, acessado em novembro 20, 2025, <https://developers.pismo.io/pismo-docs/docs/pix-and-qr-codes>
6. MANUAL BR CODE - Banco Central, acessado em novembro 20, 2025, https://www.bcb.gov.br/content/config/Documents/BR_Code_MANUAL_Version_2_May_2020.pdf
7. How, When, and Why you should switch from Vercel to a different Hosting Provider (Especially for Next.js) | by Sushrit Pasupuleti | Medium, acessado em novembro 20, 2025, <https://medium.com/@sushrit.pk21/how-when-and-why-you-should-switch-from-vercel-to-a-different-hosting-provider-especially-for-8ba25e439788>
8. Curitiba (Paróquia e Clero) | Catedral Ortodoxa, acessado em novembro 20, 2025, <https://www.arquidioceseortodoxa.com.br/curitiba-par%C3%B3quia-e-clero>
9. Creating a Charge using API - OpenPix Developers, acessado em novembro 20, 2025, <https://developers.openpix.com.br/en/docs/flows/flow-create-charge-api>
10. Next.js Starter Templates & Themes - Vercel, acessado em novembro 20, 2025, <https://vercel.com/templates/nextjs>
11. Find a plan to power your apps. - Vercel, acessado em novembro 20, 2025, <https://vercel.com/pricing>
12. QR Code Specification for Payment Systems (EMV QRCPS) - GitHub Pages, acessado em novembro 20, 2025, <https://mvallim.github.io/emv-qr-code/docs/EMVCo-Merchant-Presented-QR-Specification-v1-1.pdf>
13. rprata/pixcode: PIX typescript module - GitHub, acessado em novembro 20, 2025, <https://github.com/rprata/pixcode>

14. How to generate CRC-16 in javascript and validate the checksum on STM32F373, acessado em novembro 20, 2025, <https://community.st.com/t5/stm32-mcus-products/how-to-generate-crc-16-in-javascript-and-validate-the-checksum/td-p/620523>
15. CRC-CCITT -- 16-bit, acessado em novembro 20, 2025, <https://srecord.sourceforge.net/crc16-ccitt.html>
16. CRC16-CCITT - RIOT Documentation, acessado em novembro 20, 2025, https://doc.riot-os.org/group_sys_checksum_crc16_ccitt.html
17. Optimized CRC implementations in JavaScript - GitHub, acessado em novembro 20, 2025, <https://gist.github.com/bryc/79d1a62304773285317191f1ae5aa5b8>
18. devcarlosalberto/pix-payload: Published package on NPM ... - GitHub, acessado em novembro 20, 2025, <https://github.com/devcarlosalberto/pix-payload>
19. PIX Payments - Wise Platform API Docs, acessado em novembro 20, 2025, <https://docs.wise.com/guides/regional/brazil/pix>
20. How to create copy-paste event in JavaScript? - Stack Overflow, acessado em novembro 20, 2025, <https://stackoverflow.com/questions/64931254/how-to-create-copy-paste-event-in-javascript>
21. How to Pull Off a Brown Wedding Color Palette - The Knot, acessado em novembro 20, 2025, <https://www.theknot.com/content/brown-wedding>
22. 25 perfect wedding color combinations - Canva, acessado em novembro 20, 2025, <https://www.canva.com/learn/wedding-colors/>
23. wedding website UI design - CodePen, acessado em novembro 20, 2025, <https://codepen.io/sakshxm08/pen/oNPzQNG>
24. 15 Best HTML Wedding Website Templates for 2025 - Envato, acessado em novembro 20, 2025, <https://elements.envato.com/learn/best-html-wedding-website-templates>
25. Mylove - Wedding Next Js Template by wpoceans - ThemeForest, acessado em novembro 20, 2025, <https://themeforest.net/item/mylove-wedding-next-js-template/43553078>
26. formspree vs web3forms vs emailjs which one better for website : r/webdev - Reddit, acessado em novembro 20, 2025, https://www.reddit.com/r/webdev/comments/1oxr4kr/formspree_vs_web3forms_vs_emailjs_which_one/
27. fderuiter/wedding_website: Next.js wedding site. Gift registry. Admin panel. Vercel Hobby. Tailwind. Prisma. TypeScript . Data scraper. - GitHub, acessado em novembro 20, 2025, https://github.com/fderuiter/wedding_website
28. Embed a map | Maps Embed API - Google for Developers, acessado em novembro 20, 2025, <https://developers.google.com/maps/documentation/embed/embedding-map>
29. Learn Google Maps Embed API - Gru Singh, acessado em novembro 20, 2025, <https://www.grusingh.com/post/google-maps-embed-api/>