



Relatório de Diagnóstico - Casino Platform

Data: 13 de Fevereiro de 2026

Fase: 1 - Análise e Diagnóstico Inicial

Repositório: <https://github.com/azoneupbr-netizen/casino-platform>



Resumo Executivo

O **casino-platform** é uma aplicação Next.js 16 com TypeScript que implementa uma plataforma de cassino online e apostas esportivas. O projeto utiliza arquitetura moderna com App Router, Tailwind CSS v4 e React 19.

Principais Problemas Encontrados:

- **2 erros críticos de TypeScript** que impedem compilação correta
- **2 vulnerabilidades de alta severidade** em dependências
- **11 erros de ESLint** incluindo uso excessivo de `any`
- **25+ avisos de ESLint** (variáveis não usadas, hooks com dependências faltando)
- **Problemas graves de acessibilidade** (apenas 1 componente com `aria-label`)
- **11 instâncias de `` ao invés de `next/image`**

Estrutura do Projeto

```

casino-platform/
├── src/
│   ├── app/
│   │   ├── layout.tsx          # App Router (Next.js 13+)
│   │   ├── page.tsx            # Layout raiz
│   │   ├── globals.css          # Página inicial (Cassino)
│   │   ├── account/
│   │   │   ├── auth/callback/  # Estilos globais
│   │   │   ├── bets/           # Página de conta
│   │   │   ├── casino/          # Callback OAuth
│   │   │   ├── notifications/ # Página de apostas
│   │   │   ├── promotions/    # Página do cassino
│   │   │   ├── rewards/         # Página de notificações
│   │   │   ├── sports/          # Página de promoções
│   │   │   └── wallet/          # Página de recompensas
│   │   ├── components/
│   │   │   ├── auth/            # Página de apostas
│   │   │   ├── casino/          # Página do cassino
│   │   │   ├── layout/           # Página de layout (11)
│   │   │   ├── payment/          # Página de pagamento (1)
│   │   │   ├── promotions/       # Página de promoções (1)
│   │   │   ├── sports/           # Página de esportes (13)
│   │   │   └── ui/               # Página de UI (2)
│   │   ├── context/
│   │   ├── contexts/           # Contexto de tema
│   │   ├── services/            # Contextos (Auth, Toast)
│   │   └── public/              # Serviços de API (3)
│   ├── next.config.ts          # Arquivos estáticos
│   ├── tsconfig.json           # Configuração Next.js
│   ├── eslint.config.mjs       # Configuração TypeScript
│   └── package.json            # Configuração ESLint
                                # Dependências

```

Arquitetura:

- **Framework:** Next.js 16.1.4 com App Router
- **Linguagem:** TypeScript 5.x com modo strict
- **Estilização:** Tailwind CSS v4
- **Gerenciamento de Estado:** React Context API
- **HTTP Client:** Axios 1.13.3

Stack Tecnológico

Dependências de Produção:

Pacote	Versão	Status
next	16.1.4	 Vulnerável (3 CVEs)
react	19.2.3	 Desatualizado
react-dom	19.2.3	 Desatualizado
axios	1.13.3	 Vulnerável (1 CVE)
lucide-react	0.563.0	 Desatualizado

Dependências de Desenvolvimento:

Pacote	Versão	Status
typescript	^5	 OK
tailwindcss	^4	 OK
eslint	^9	 Desatualizado (v10 disponível)
@types/node	^20	 Desatualizado
@types/react	^19	 Desatualizado

Erros Críticos

Erros de TypeScript (2)

Arquivo	Linha	Erro
Header.tsx	16	Property 'toast' does not exist on type 'ToastContextData'
PromotionsPage.tsx	134	Cannot find name 'toast'

Causa: O `ToastContext` exporta `showToast`, mas os componentes tentam usar `toast`.

Erros de ESLint (11)

Tipo	Quantidade	Descrição
@typescript-eslint/no-explicit-any	9	Uso de tipo <code>any</code> não específico
react-hooks/set-state-in-effect	1	setState dentro de useEffect causa re-renders



Vulnerabilidades de Segurança

Alta Severidade (2)

1. Axios (<=1.13.4)

- **CVE:** GHSA-43fc-jf86-j433
- **Descrição:** Vulnerável a DoS via `__proto__` key em mergeConfig
- **Correção:** `npm audit fix`

2. Next.js (15.6.0-canary.0 - 16.1.4)

- **CVE 1:** GHSA-9g9p-9gw9-jx7f - DoS via Image Optimizer
- **CVE 2:** GHSA-h25m-26qc-wcjf - DoS via HTTP request deserialization
- **CVE 3:** GHSA-5f7q-jpqc-wp7h - Unbounded Memory Consumption
- **Correção:** `npm audit fix --force` (instala next@16.1.6)

Problemas de Segurança no Código

Problema	Arquivo	Descrição
URL de API Hardcoded	<code>api.ts</code>	URL de produção hardcoded no código

```
// src/services/api.ts
baseUrl: process.env.NEXT_PUBLIC_API_URL || 'https://n8n-casino-api.hzkun.easypanel.host'
```

⚡ Problemas de Performance

Uso de `` ao invés de `next/image` (11 instâncias)

Arquivo	Linha
Header.tsx	268, 277
UserDropdown.tsx	39
DepositModal.tsx	99
BonusSlots.tsx	67
TopGames.tsx	102
CasinoPage.tsx	161
PopularBetsWidget.tsx	103, 108
MiniGamesSidebar.tsx	66
wallet/page.tsx	170

Impacto: LCP (Largest Contentful Paint) mais lento, maior consumo de banda.

Client Components em Excesso

- **34 de 48 componentes** são Client Components ('use client')
- Potencial para otimizar com Server Components

Lazy Loading Implementado ✓

- TopGames e BonusSlots já usam `dynamic()` com loading states



Problemas Específicos Identificados

1. Mobile Layout Issues

Problemas Encontrados:

a) BottomNavigation com z-index excessivo:

```
// z-[100000] é excessivamente alto
<div className="md:hidden fixed bottom-0 ... z-[100000]">
```

b) Falta de viewport meta tag verificada

c) Overflow handling:

```
html, body {
  overflow-x: hidden !important; /* !important pode causar problemas */
}
```

2. Infinite Scroll

Status: Não há implementação de infinite scroll com IntersectionObserver.

- O scroll é manual (botões de navegação) ou horizontal scroll nativo.
- Sem memory leaks relacionados a infinite scroll identificados.

3. Excessive Spacing

Análise do CSS:

- Uso de variáveis CSS para espaçamento consistente
- Tailwind v4 com configuração padrão
- Possíveis problemas em componentes específicos que precisam de inspeção visual



Problemas de Acessibilidade

Situação Crítica

Métrica	Valor	Esperado
Componentes com aria-label	1	30+
Componentes com role	0	10+
Imagens com alt	11	✓ OK

Problemas Identificados:

1. Falta de aria-labels em:

- Botões de navegação
- Modais
- Formulários
- Links interativos

2. Falta de roles em:

- Navigation landmarks
- Main content areas
- Dialog modals

3. Problemas de foco:

- Não há skip links
- Não há focus trapping em modais



Métricas do Projeto

Métrica	Valor
Total de linhas de código	~5,921
Arquivos TypeScript (.tsx)	48
Arquivos TypeScript (.ts)	4
Arquivos CSS	1
Componentes totais	~35
Páginas (App Router)	11
Serviços de API	3
Contextos React	3
Client Components	34
Console.logs no código	20
Commits no repositório	32

⚠️ Avisos de ESLint (25+)

Variáveis Não Utilizadas (15)

Arquivo	Variável
bets/page.tsx	loading
wallet/page.tsx	loading
AccountPage.tsx	loading
CasinoPage.tsx	BonusSlots
Header.tsx	typingSpeed
UserDropdown.tsx	KeyIcon, ChatIcon, GiftIcon
BetSlip.tsx	Gamelcons, isOpen, setStake, showSettings, setShowSettings, oddsOption, setOddsOption
SportsPage.tsx	QuantumAI
SportsSidebar.tsx	useState
AuthContext.tsx	api
benefits.ts	error (2x)

React Hooks com Dependências Faltando (3)

Arquivo	Hook
AccountPage.tsx	useEffect - fetchUserData, router
Header.tsx	useEffect - gameNames
SportsFeed.tsx	useEffect - fetchMatches

🔧 Recomendações Prioritárias (Top 10)

Prioridade Alta (Crítico)

1. **Corrigir erros de TypeScript** - `toast` → `showToast` em `Header.tsx` e `PromotionsPage.tsx`
2. **Atualizar Next.js** - `npm audit fix --force` para corrigir 3 CVEs
3. **Atualizar Axios** - `npm audit fix` para corrigir CVE de DoS

Prioridade Média (Importante)

1. **Substituir `` por `next/image`** - 11 instâncias para otimização de imagens

2. **Adicionar aria-labels** - Melhorar acessibilidade em todos os componentes interativos
3. **Corrigir dependências de useEffect** - 3 hooks com dependências faltando
4. **Remover variáveis não utilizadas** - 15 variáveis/imports não usados

Prioridade Baixa (Melhoria)

1. **Tipar corretamente** - Remover 9 usos de `any`
 2. **Mover URL de API para variável de ambiente** - Remover hardcoded URL
 3. **Otimizar Client Components** - Avaliar quais podem ser Server Components
-

Próximos Passos (Fase 2)

Preparação para Testes

1. **Configurar ambiente de testes:**
 - Instalar Jest/Vitest + React Testing Library
 - Configurar mocks para API e contextos
 2. **Criar testes para:**
 - Componentes críticos (Header, BottomNavigation, modais)
 - Contextos (Auth, Toast, Theme)
 - Serviços de API
 - Hooks customizados
 3. **Testes E2E:**
 - Configurar Playwright/Cypress
 - Fluxos críticos: login, navegação, apostas
-



Conclusão

O projeto casino-platform está funcional mas apresenta diversos problemas que precisam ser corrigidos antes de ir para produção:

- **Segurança:** 2 vulnerabilidades de alta severidade precisam ser corrigidas imediatamente
- **Qualidade de Código:** 2 erros de TypeScript e 11 erros de ESLint
- **Acessibilidade:** Situação crítica com falta de suporte para leitores de tela
- **Performance:** 11 oportunidades de otimização de imagens

Estimativa de esforço para correções: 8-16 horas de desenvolvimento

Relatório gerado automaticamente pela Fase 1 de Análise do Casino Platform