

# Fase 2: Relatório de Testes Completos

## Resumo Executivo

Este relatório documenta a criação e implementação da suíte completa de testes para o projeto **casino-platform**. O objetivo foi atingir uma cobertura de código superior a 80% e garantir a qualidade do software através de testes unitários, de integração, acessibilidade e E2E.

## Métricas Finais

Métrica	Valor	Meta	Status
Statements	85.29%	>80%	<span style="color: green;">✓ Atingido</span>
Branches	70.14%	>50%	<span style="color: green;">✓ Atingido</span>
Functions	74.50%	>60%	<span style="color: green;">✓ Atingido</span>
Lines	86.14%	>80%	<span style="color: green;">✓ Atingido</span>
Testes Unitários	529	-	<span style="color: green;">✓ Todos Passando</span>
Arquivos de Teste	44	-	<span style="color: green;">✓</span>

## Configuração do Ambiente de Testes

### Ferramentas Instaladas e Configuradas

- Jest** (v30.2.0) - Framework de testes principal
- React Testing Library** (v16.3.2) - Testes de componentes React
- @testing-library/jest-dom** (v6.9.1) - Matchers customizados para DOM
- @testing-library/user-event** (v14.6.1) - Simulação de eventos de usuário
- jest-axe** (v10.0.0) - Testes de acessibilidade
- Playwright** (v1.58.2) - Testes E2E

## Arquivos de Configuração

### `jest.config.js`

```
const nextJest = require('next/jest');

const createJestConfig = nextJest({
  dir: './',
});

const customJestConfig = {
  setupFilesAfterEnv: ['<rootDir>/jest.setup.js'],
  testEnvironment: 'jest-environment-jsdom',
  moduleNameMapper: {
    '^@/(.*)$': '<rootDir>/src/$1',
  },
  collectCoverageFrom: [
    'src/**/*.{js,jsx,ts,tsx}',
    '!src/**/*.d.ts',
  ],
  coverageThreshold: {
    global: {
      branches: 50,
      functions: 60,
      lines: 75,
      statements: 75,
    },
  },
  coverageReporters: ['text', 'text-summary', 'lcov', 'html'],
};

module.exports = createJestConfig(customJestConfig);
```

### `jest.setup.js`

- Mock do Next.js Image component
- Mock do Next.js router (useRouter, usePathname, etc.)
- Mock do localStorage
- Mock do window.matchMedia
- Mock do IntersectionObserver e ResizeObserver

### `playwright.config.ts`

- Configuração para testes em múltiplos navegadores (Chromium, Firefox, WebKit)
- Suporte a viewports mobile (Pixel 5, iPhone 12)
- Integração com servidor de desenvolvimento Next.js

## Scripts Disponíveis (`package.json`)

```
{
  "test": "jest",
  "test:watch": "jest --watch",
  "test:coverage": "jest --coverage",
  "test:e2e": "playwright test",
  "test:e2e:ui": "playwright test --ui",
  "test:e2e:headed": "playwright test --headed",
  "test:e2e:report": "playwright show-report"
}
```

## Testes Criados por Categoria

### 1. Testes de Contextos (3 arquivos)

Arquivo	Testes	Cobertura
AuthContext.test.tsx	Login, logout, estado de autenticação	100%
ThemeContext.test.tsx	Toggle tema, persistência localStorage	100%
ToastContext.test.tsx	Mostrar toast, fechar toast	88.88%

### 2. Testes de Serviços (3 arquivos)

Arquivo	Testes	Cobertura
api.test.ts	Configuração axios, interceptors, métodos HTTP	58.33%
auth.test.ts	getProfile, updateProfile, updatePassword, login, logout	100%
benefits.test.ts	Fetch benefits, claim benefits	92.45%

### 3. Testes de Componentes de Layout (10 arquivos)

Arquivo	Testes	Cobertura
Header.test.tsx	Navegação, user dropdown, theme toggle, login flow	64.13%
BottomNavigation.test.tsx	Navegação mobile, active states, a11y	100%
UserDropdown.test.tsx	Menu dropdown, logout	100%
Sidebar.test.tsx	Expansão, menus esportes/casino, navegação	100%
AnimatedBanner.test.tsx	Variantes full/sidebar, styling	100%
SupportModal.test.tsx	Chat, envio mensagens, UI	100%
TopBanner.test.tsx	Renderização, close functionality	100%
Footer.test.tsx	Links, copyright, social media	100%
ChatButton.test.tsx	Open/close chat, message input	100%

### 4. Testes de Componentes de Casino (4 arquivos)

Arquivo	Testes	Cobertura
CasinoPage.test.tsx	Renderização, filtros, jogos	86.11%
TopGames.test.tsx	Lista jogos, lazy loading, scroll	100%
BonusSlots.test.tsx	Slots com bônus, interações	100%
HeroBanner.test.tsx	Auto-rotation, navegação manual	94.11%

## 5. Testes de Componentes de Sports (12 arquivos)

Arquivo	Testes	Cobertura
SportsPage.test.tsx	Renderização, filtros, eventos	96.87%
LiveMatchInfo.test.tsx	Informações partida, odds	100%
PopularBetsWidget.test.tsx	Apostas populares	93.33%
MiniGamesSidebar.test.tsx	Mini jogos	100%
SportsSidebar.test.tsx	Seleção esportes, ligas, live	100%
QuantumAI.test.tsx	Chat AI, sugestões, respostas	96.55%
BetSlip.test.tsx	Gerenciamento apostas	65.57%
SportsFeed.test.tsx	Feed de eventos	61.53%
HeroCarousel.test.tsx	Carousel de matches	100%
SportsCategoryNav.test.tsx	Navegação categorias	100%
OddsSettingsWidget.test.tsx	Configurações odds	90.9%
LongTermBetsWidget.test.tsx	Apostas longo prazo	100%

## 6. Testes de Componentes de Auth (1 arquivo)

Arquivo	Testes	Cobertura
useAuthModal.test.tsx	Hook de modal de autenticação	100%

## 7. Testes de Componentes de Payment (1 arquivo)

Arquivo	Testes	Cobertura
DepositModal.test.tsx	Modal depósito, PIX, validação	77.5%

## 8. Testes de Componentes de Promotions (1 arquivo)

Arquivo	Testes	Cobertura
PromotionsPage.test.tsx	Promoções, filtros	81.81%

## 9. Testes de Componentes de Account (1 arquivo)

Arquivo	Testes	Cobertura
AccountPage.test.tsx	Perfil, senha, updates	93.84%

## 10. Testes de Componentes UI (2 arquivos)

Arquivo	Testes	Cobertura
ThemeToggle.test.tsx	Toggle tema, persistência	100%
Toast.test.tsx	Notificações, tipos	100%

## 11. Testes de Integração (2 arquivos)

Arquivo	Descrição
auth-flow.test.tsx	Fluxo completo de autenticação
theme-flow.test.tsx	Fluxo de mudança de tema

## 12. Testes de Acessibilidade (1 arquivo)

Arquivo	Descrição
ally.test.tsx	Testes com jest-axe para componentes principais

### Testes de Acessibilidade Incluídos:

- Verificação de violações WCAG
- Verificação de aria-labels
- Verificação de roles
- Verificação de navegação por teclado
- Verificação de aria-current em links ativos
- Verificação de aria-hidden em ícones decorativos
- Verificação de elementos focáveis

## 13. Testes E2E com Playwright (4 arquivos)

Arquivo	Descrição
home.spec.ts	Página inicial, navegação, login modal
casino.spec.ts	Página casino, jogos, filtros
sports.spec.ts	Página esportes, eventos, bet slip
responsive.spec.ts	Responsividade em diferentes viewports

## Cobertura de Código Detalhada

### Por Diretório

Diretório	Statements	Branches	Functions	Lines
components/account	93.84%	74.07%	85.71%	93.84%
components/auth	100%	100%	100%	100%
components/casino	93.75%	90%	82.92%	94.5%
components/layout	76.85%	69.62%	64.19%	78.02%
components/payment	77.5%	83.33%	80%	78.94%
components/promotions	81.81%	92.3%	70%	85%
components/sports	83.45%	62.76%	71.69%	83.9%
components/ui	100%	100%	100%	100%
context	100%	90%	100%	100%
contexts	95.83%	83.33%	75%	95.55%
services	88.15%	59.37%	92.3%	89.55%

### Componentes com 100% de Cobertura

- AnimatedBanner.tsx
- BottomNavigation.tsx
- ChatButton.tsx
- Footer.tsx
- Sidebar.tsx
- SupportModal.tsx
- TopBanner.tsx
- UserDropdown.tsx
- AgeGateModal.tsx
- useAuthModal.ts
- BonusSlots.tsx

- TopGames.tsx
  - HeroCarousel.tsx
  - LargePromoBanner.tsx
  - LiveMatchInfo.tsx
  - LongTermBetsWidget.tsx
  - MiniGamesSidebar.tsx
  - SportsCategoryNav.tsx
  - SportsSidebar.tsx
  - ThemeToggle.tsx
  - Toast.tsx
  - ThemeContext.tsx
  - AuthContext.tsx
  - auth.ts
- 

## Áreas com Baixa Cobertura

### Componentes que Precisam de Mais Testes

Componente	Cobertura Atual	Motivo	Recomendação
api.ts	58.33%	Interceptors complexos	Adicionar testes de interceptors
SportsFeed.tsx	61.53%	Lógica de renderização condicional	Testar mais cenários de dados
Header.tsx	64.13%	Muitos modais e estados	Testar fluxos de modais
BetSlip.tsx	65.57%	Lógica de apostas complexa	Testar cálculos de odds

---

## Resultados dos Testes

### Testes Unitários (Jest)

```
Test Suites: 40 passed, 40 total
Tests:       529 passed, 529 total
Snapshots:   0 total
Time:        ~15s
```

**Todos os 529 testes passaram com sucesso!**

### Testes E2E (Playwright)

- 4 arquivos de teste E2E criados

- Cobertura de navegadores: Chromium, Firefox, WebKit
  - Cobertura de dispositivos: Desktop, Tablet, Mobile (Pixel 5, iPhone 12)
- 

## Recomendações para Melhorar Cobertura

### Prioridade Alta

1. **api.ts** - Adicionar testes para:
  - Interceptor de request (token injection)
  - Tratamento de erros 401/403
  - Retry logic (se implementado)
2. **SportsFeed.tsx** - Adicionar testes para:
  - Diferentes estados de carregamento
  - Filtros de eventos
  - Ordenação de partidas
3. **Header.tsx** - Adicionar testes para:
  - Fluxo completo de registro
  - Interação com PromotionsSidebar
  - Estados de erro de autenticação

### Prioridade Média

1. **BetSlip.tsx** - Adicionar testes para:
    - Cálculo de odds combinadas
    - Validação de valores mínimos/máximos
    - Fluxo de envio de aposta
  2. **DepositModal.tsx** - Adicionar testes para:
    - Diferentes métodos de pagamento
    - Validação de valores
    - Estado de processamento
-

## Lista de Arquivos de Teste Criados

### Testes Unitários (Jest) - 40 arquivos

```
src/__tests__/
├── accessibility/
│   └── a11y.test.tsx
├── components/
│   ├── account/
│   │   └── AccountPage.test.tsx
│   ├── auth/
│   │   └── useAuthModal.test.tsx
│   ├── casino/
│   │   ├── BonusSlots.test.tsx
│   │   ├── CasinoPage.test.tsx
│   │   ├── HeroBanner.test.tsx
│   │   └── TopGames.test.tsx
│   ├── layout/
│   │   ├── AnimatedBanner.test.tsx
│   │   ├── BottomNavigation.test.tsx
│   │   ├── ChatButton.test.tsx
│   │   ├── Footer.test.tsx
│   │   ├── Header.test.tsx
│   │   ├── Sidebar.test.tsx
│   │   ├── SupportModal.test.tsx
│   │   ├── TopBanner.test.tsx
│   │   └── UserDropdown.test.tsx
│   ├── payment/
│   │   └── DepositModal.test.tsx
│   ├── promotions/
│   │   └── PromotionsPage.test.tsx
│   ├── sports/
│   │   ├── BetSlip.test.tsx
│   │   ├── HeroCarousel.test.tsx
│   │   ├── LiveMatchInfo.test.tsx
│   │   ├── LongTermBetsWidget.test.tsx
│   │   ├── MiniGamesSidebar.test.tsx
│   │   ├── OddsSettingsWidget.test.tsx
│   │   ├── PopularBetsWidget.test.tsx
│   │   ├── QuantumAI.test.tsx
│   │   ├── SportsCategoryNav.test.tsx
│   │   ├── SportsFeed.test.tsx
│   │   ├── SportsPage.test.tsx
│   │   └── SportsSidebar.test.tsx
│   └── ui/
│       ├── ThemeToggle.test.tsx
│       └── Toast.test.tsx
└── contexts/
    ├── AuthContext.test.tsx
    ├── ThemeContext.test.tsx
    └── ToastContext.test.tsx
└── integration/
    ├── auth-flow.test.tsx
    └── theme-flow.test.tsx
└── services/
    ├── api.test.ts
    ├── auth.test.ts
    └── benefits.test.ts
```

## Testes E2E (Playwright) - 4 arquivos

```
e2e/  
├── casino.spec.ts  
├── home.spec.ts  
└── responsive.spec.ts  
└── sports.spec.ts
```

## Conclusão

A **Fase 2: Criação de Testes Completos** foi concluída com sucesso. Os objetivos principais foram atingidos:

- **Cobertura de código superior a 80%** (Lines: 86.14%, Statements: 85.29%)
- **529 testes unitários criados e passando**
- **Testes de acessibilidade implementados com jest-axe**
- **Testes E2E configurados com Playwright**
- **Ambiente de testes completamente configurado**

O projeto agora possui uma base sólida de testes que garante a qualidade do código e facilita a manutenção e evolução do sistema.

---

**Data de Geração:** 13 de Fevereiro de 2026

**Versão:** 1.0.0