🔒 Security Checklist - Timming LoveU Production

📋 Pre-Deployment Security Checklist

✅ Authentication & Authorization

- [ ] `NEXTAUTH_SECRET` é forte e único (gerado com `openssl rand -base64 32`)
- [ ] Sessões expiram após período apropriado
- [ ] NextAuth configurado corretamente para produção
- [ ] Proteção contra CSRF habilitada (padrão no NextAuth)
- [ ] Rate limiting implementado em rotas de autenticação
- [ ] Senhas são hasheadas com bcrypt (salt rounds >= 10)
- [ ] Validação de força de senha implementada

✅ Database Security

- [ ] Connection string usa credenciais fortes
- [ ] Database não está exposto publicamente (apenas VPN/whitelist)
- [ ] Connection pooling configurado apropriadamente
- [ ] SSL/TLS habilitado para conexões de banco
- [ ] Backups automáticos configurados
- [ ] Prepared statements usados (Prisma faz isso automaticamente)
- [ ] Princípio do menor privilégio aplicado (usuário DB tem apenas permissões necessárias)

✅ Environment Variables

- [ ] Arquivo `.env` adicionado ao `.gitignore`
- [ ] Sem credenciais hardcoded no código
- [ ] `.env.example` criado com valores de exemplo
- [ ] Variáveis de ambiente validadas no startup
- [ ] Valores diferentes entre dev e produção

✅ API Security

- [ ] Rate limiting implementado em todas as APIs
- [ ] Validação de input em todas as rotas
- [ ] Sanitização de dados de entrada
- [ ] Proteção contra SQL injection (Prisma)
- [ ] Proteção contra XSS
- [ ] CORS configurado apropriadamente
- [ ] Autenticação requerida em rotas protegidas
- [ ] Limites de tamanho de payload configurados

✅ HTTP Security Headers

- [ ] `Strict-Transport-Security` (HSTS)
- [ ] `X-Frame-Options: SAMEORIGIN`

- [ ] `X-Content-Type-Options: nosniff`
- [ ] `X-XSS-Protection: 1; mode=block`
- [ ] `Referrer-Policy: strict-origin-when-cross-origin`
- [ ] `Permissions-Policy` configurado
- [ ] CSP (Content Security Policy) - opcional mas recomendado

✅ HTTPS/TLS

- [ ] HTTPS habilitado em produção
- [ ] Certificado SSL válido e não expirado
- [ ] Redirecionamento HTTP → HTTPS configurado
- [ ] TLS 1.2+ usado (não TLS 1.0/1.1)
- [ ] Certificado auto-renovável (Let's Encrypt)

✅ File Upload Security

- [ ] Validação de tipo de arquivo (MIME type)
- [ ] Limite de tamanho de arquivo
- [ ] Nomes de arquivo sanitizados
- [ ] Uploads armazenados fora do webroot (se possível)
- [ ] Scan de malware (em uploads críticos)
- [ ] Proteção contra path traversal

✅ Session Management

- [ ] Sessions têm timeout apropriado
- [ ] Session IDs são aleatórios e seguros
- [ ] Logout invalida sessão completamente
- [ ] Proteção contra session fixation
- [ ] Cookies configurados com flags seguras:
- [ ] `httpOnly: true`
- [ ] `secure: true` (produção)
- [ ] `sameSite: 'lax'` ou `'strict'`

✅ Logging & Monitoring

- [ ] Logging de eventos de segurança habilitado
- [ ] Senhas/tokens nunca logados
- [ ] Monitoramento de tentativas de login falhadas
- [ ] Alertas para atividades suspeitas
- [ ] Error tracking configurado (Sentry)
- [ ] Uptime monitoring configurado

✅ Dependencies & Code

- [ ] Todas as dependências atualizadas
- [ ] Sem vulnerabilidades conhecidas (`npm audit`)
- [ ] Dependências de fontes confiáveis
- [ ] Lock file commitado (package-lock.json)
- [ ] Secrets scanning habilitado no Git
- [ ] Code review antes do deploy

✅ Infrastructure

- [ ] Firewall configurado apropriadamente
- [ ] Apenas portas necessárias abertas (80, 443, 22)
- [ ] SSH com key-based authentication
- [ ] Root login desabilitado
- [ ] Fail2ban ou similar configurado
- [ ] Sistema operacional atualizado
- [ ] Aplicação roda com usuário não-privilegiado

✅ Data Protection

- [ ] Dados sensíveis criptografados em repouso
- [ ] Dados sensíveis criptografados em trânsito
- [ ] PII (Personal Identifiable Information) protegido
- [ ] Compliance com LGPD/GDPR (se aplicável)
- [ ] Política de retenção de dados implementada
- [ ] Backup criptografado

✅ Error Handling

- [ ] Mensagens de erro genéricas para usuários
- [ ] Detalhes de erro não expostos em produção
- [ ] Stack traces não expostos
- [ ] Error logging centralizado
- [ ] Páginas de erro customizadas

✅ Third-Party Services

- [ ] APIs de terceiros com rate limiting
- [ ] Tokens/Keys de API rotacionados regularmente
- [ ] Permissões mínimas para serviços externos
- [ ] Webhook signatures validadas
- [ ] OAuth scopes mínimos necessários

🛠️ Security Testing

Manual Testing

```
# 1. Verificar headers de segurança
curl -I https://seu-dominio.com

# 2. Test SSL/TLS
openssl s_client -connect seu-dominio.com:443 -tls1_2

# 3. Verificar rate limiting
for i in {1..100}; do curl https://seu-dominio.com/api/login; done

# 4. Testar autenticação
curl -X POST https://seu-dominio.com/api/protected-route
```

Automated Testing

```
# Security audit
npm audit

# Dependency check
npm outdated

# Type checking
npm run type-check

# Linting
npm run lint
```

Security Scanning Tools

- [ ] Mozilla Observatory (https://observatory.mozilla.org/)
- [ ] SecurityHeaders.com (https://securityheaders.com/)
- [ ] SSL Labs (https://www.ssllabs.com/ssltest/)
- [ ] OWASP ZAP (automated security testing)
- [ ] Snyk (dependency scanning)

🚨 Incident Response

Em caso de incidente de segurança:

1. **Contenção Imediata**
   - Isolar sistema afetado
   - Bloquear IPs maliciosos
   - Desabilitar contas comprometidas

2. **Investigação**
   - Revisar logs
   - Identificar vetor de ataque
   - Determinar escopo do incidente

3. **Remediação**
   - Aplicar patches/fixes
   - Rotacionar credenciais
   - Atualizar regras de firewall

4. **Comunicação**
   - Notificar stakeholders
   - Comunicar usuários afetados (se necessário)
   - Documentar incidente

5. **Recuperação**
   - Restaurar serviços
   - Verificar integridade dos dados
   - Monitorar para recorrência

6. **Post-Mortem**
   - Documentar lições aprendidas

- Atualizar procedimentos
- Implementar melhorias

📞 Contacts

- **Security Issues**: security@timming-loveu.com
- **Emergency**: +XX XXXX-XXXX

🔄 Regular Security Tasks

Diário

- [ ] Revisar logs de segurança
- [ ] Verificar alertas de monitoring

Semanal

- [ ] Revisar tentativas de login falhadas
- [ ] Verificar uptime e performance

Mensal

- [ ] Atualizar dependências
- [ ] Executar `npm audit`
- [ ] Revisar acessos e permissões
- [ ] Testar backups e recovery

Trimestral

- [ ] Security audit completo
- [ ] Penetration testing
- [ ] Revisar políticas de segurança
- [ ] Treinamento de equipe

Anual

- [ ] Renovar certificados SSL
- [ ] Audit de compliance
- [ ] Disaster recovery drill
- [ ] Revisar plano de resposta a incidentes

📚 References

- OWASP Top 10 (https://owasp.org/www-project-top-ten/)
- Next.js Security Best Practices (https://nextjs.org/docs/going-to-production)
- Prisma Security Best Practices (https://www.prisma.io/docs/guides/security)
- LGPD Compliance (https://www.gov.br/cidadania/pt-br/acesso-a-informacao/lgpd)

Last Updated: October 2024
Version: 1.0.0