

Desenvolvimento de Software em Colaboração: Desafios para o Registro de Programas de Computador

Joelson Gomes Pequeno¹

Resumo

O desenvolvimento colaborativo de software tornou-se uma prática essencial na indústria da tecnologia, impulsionada pelo crescimento de iniciativas de código aberto, plataformas **no-code** e **low-code**, além da adoção de metodologias ágeis e ferramentas de colaboração em nuvem. Embora esse modelo de criação fomente a inovação e reduza custos, ele também traz desafios jurídicos significativos, especialmente no que se refere à titularidade dos direitos de propriedade intelectual, à governança dos projetos e ao registro de programas de computador.

Este artigo analisa os principais obstáculos enfrentados no registro de software junto ao Instituto Nacional da Propriedade Industrial (INPI), destacando as implicações legais das contribuições coletivas e os limites do marco normativo brasileiro. Além disso, são discutidas as tendências futuras desse setor, considerando o impacto da inteligência artificial, dos contratos inteligentes (*smart contracts*) e das novas formas de licenciamento de software. A partir de uma abordagem multidisciplinar, o estudo examina a interação entre inovação tecnológica e segurança jurídica, com foco na proteção dos direitos dos desenvolvedores e das empresas envolvidas.

1. Introdução

A colaboração no desenvolvimento de software tem se consolidado como um modelo predominante na indústria tecnológica, permitindo que programadores, empresas e comunidades acadêmicas criem soluções inovadoras de maneira descentralizada e interconectada. O avanço das plataformas **open source**, a popularização de repositórios como **GitHub** e **GitLab**, e a ascensão de ferramentas **no-code/low-code** têm viabilizado um

¹ Técnico em Propriedade Industrial; Chefe Substituto da Divisão de Programas de Computador e Topografias de Circuitos Integrados do INPI; Graduado em Direito pela Universidade Federal do Estado do Rio de Janeiro.

ambiente de produção altamente dinâmico, onde o compartilhamento de conhecimento e a cooperação entre diferentes atores são fundamentais para o progresso da tecnologia.

Apesar das vantagens dessa abordagem, como a aceleração do desenvolvimento e a democratização do acesso ao conhecimento, o modelo colaborativo levanta questões jurídicas e regulatórias complexas. A titularidade dos direitos sobre o código, a compatibilidade entre licenças de software e as regras para o registro no **Instituto Nacional da Propriedade Industrial (INPI)** são apenas alguns dos desafios enfrentados por desenvolvedores e empresas que atuam nesse setor.

Este artigo tem como objetivo examinar os desafios jurídicos do desenvolvimento colaborativo de software, com ênfase na proteção da propriedade intelectual e nos critérios para o registro de programas de computador no Brasil. Além disso, serão discutidas as tendências emergentes nesse cenário, incluindo o papel da inteligência artificial na automação de códigos e os impactos da descentralização na governança dos projetos tecnológicos. A análise se baseia em uma abordagem interdisciplinar, considerando aspectos legais, técnicos e econômicos que influenciam a inovação no setor de software.

2. Modelos de Desenvolvimento Colaborativo

O desenvolvimento de software pode ocorrer em diferentes modelos colaborativos, cada um com características próprias, vantagens e desafios. A colaboração pode se dar entre indivíduos, empresas e comunidades, variando desde iniciativas de código aberto até parcerias empresariais estratégicas. Os principais modelos incluem:

Código Aberto (Open Source)

O modelo de código aberto baseia-se na disponibilização pública do código-fonte, permitindo que qualquer pessoa o utilize, modifique e redistribua sob determinadas condições, geralmente estabelecidas por licenças específicas, como a GNU General Public License (GPL), a Apache License e a MIT License.

Esse modelo fomenta a inovação ao permitir a colaboração de desenvolvedores de diferentes partes do mundo, resultando em soluções mais seguras e robustas, uma vez que qualquer

programador pode revisar o código e identificar vulnerabilidades. Projetos como **Linux**, **Apache**, **Mozilla Firefox** e **LibreOffice** são exemplos notáveis desse modelo.

Eric S. Raymond, em *The Cathedral and the Bazaar* (2001), argumenta que o desenvolvimento em código aberto se assemelha a um “bazar”, onde muitas mentes trabalham simultaneamente no aprimoramento do software, contrastando com o modelo fechado tradicional, a “catedral”, onde apenas um grupo restrito de desenvolvedores tem acesso ao código. Essa abordagem permite a rápida identificação e correção de falhas, além de promover um ambiente de inovação contínua.

Desenvolvimento Empresarial Compartilhado

Empresas frequentemente estabelecem parcerias para o desenvolvimento conjunto de softwares proprietários, combinando recursos, expertise e infraestrutura. Esse modelo pode ocorrer entre grandes corporações ou entre startups e empresas consolidadas, buscando inovação e competitividade.

Um exemplo desse modelo é o **Consórcio Open Invention Network (OIN)**, formado por empresas como IBM, Google e Red Hat, que compartilham tecnologias para o avanço do software de código aberto, ao mesmo tempo em que protegem suas inovações contra litígios.

Outro caso é o **desenvolvimento de soluções empresariais SaaS (Software as a Service)**, onde múltiplas empresas colaboram para criar plataformas integradas que atendem a nichos específicos, como softwares de gestão empresarial (ERP) e ferramentas de automação de processos.

Plataformas No-Code e Low-Code

As plataformas **No-Code** e **Low-Code** representam uma revolução na colaboração no desenvolvimento de software, permitindo que usuários sem conhecimentos avançados de programação criem aplicações de forma visual e interativa.

Ferramentas como **Bubble**, **OutSystems**, **Microsoft Power Apps** e **Mendix** permitem que empresas desenvolvam soluções internas sem a necessidade de um time extenso de programadores, agilizando a inovação e a adaptação a novas demandas.

O modelo facilita a colaboração entre desenvolvedores tradicionais e usuários finais, democratizando o acesso à tecnologia e promovendo a participação de profissionais de áreas diversas na construção de soluções digitais. No entanto, o uso dessas plataformas pode gerar desafios relacionados à propriedade intelectual, interoperabilidade e dependência de fornecedores.

Crowdsourcing em Desenvolvimento de Software

O **crowdsourcing** envolve a terceirização de tarefas para uma comunidade externa, permitindo que múltiplos desenvolvedores colaborem na resolução de problemas específicos.

Empresas como **Topcoder**, **Kaggle** e **Gitcoin** adotam esse modelo, promovendo competições e desafios em que desenvolvedores e cientistas de dados contribuem para a criação e aprimoramento de algoritmos, softwares e soluções tecnológicas.

Esse modelo acelera a inovação e reduz custos, pois as empresas podem acessar talentos altamente especializados sem a necessidade de contratação formal. No entanto, ele apresenta desafios como controle de qualidade, definição de titularidade dos direitos sobre o código e necessidade de regras claras para a remuneração dos colaboradores.

Desenvolvimento em Comunidades Virtuais

O crescimento de plataformas como **GitHub**, **GitLab** e **Bitbucket** impulsionou um novo paradigma de desenvolvimento colaborativo, no qual programadores de diferentes partes do mundo trabalham juntos em projetos open-source e privados.

Essas plataformas oferecem ferramentas para **controle de versão (Git)**, **gestão de issues**, **automação de testes** e **integração contínua**, permitindo que times distribuídos colaborem de maneira eficiente. Empresas como **Microsoft**, **Google** e **Facebook** utilizam esse modelo para gerenciar seus projetos de código aberto, promovendo contribuições da comunidade enquanto mantêm governança sobre o desenvolvimento.

Projetos como o **Kubernetes**, um dos principais orquestradores de contêineres do mundo, e o **TensorFlow**, uma biblioteca de machine learning da Google, são exemplos de software desenvolvido colaborativamente nessas plataformas, com milhares de contribuidores ao redor do mundo.

Os modelos de desenvolvimento colaborativo permitem uma abordagem mais flexível e inovadora para a criação de software, seja por meio de comunidades open-source, parcerias empresariais, crowdsourcing ou plataformas no-code/low-code. Cada modelo apresenta vantagens e desafios, especialmente no que diz respeito à governança, licenciamento, segurança e propriedade intelectual.

O avanço dessas formas de colaboração transforma não apenas a maneira como o software é produzido, mas também sua relação com os mercados, os usuários finais e os direitos de propriedade intelectual. A escolha do modelo mais adequado dependerá dos objetivos estratégicos dos desenvolvedores e das empresas envolvidas no processo de inovação tecnológica.

3. Aspectos Jurídicos e Desafios no Registro de Software no INPI

O registro de software no Instituto Nacional da Propriedade Industrial (INPI) é regulamentado pela Lei 9.609/1998 e confere proteção aos direitos do autor, garantindo a exclusividade de uso, comercialização e licenciamento do programa de computador. No entanto, projetos desenvolvidos de forma colaborativa apresentam desafios específicos que não são facilmente solucionados pelas regras tradicionais de propriedade intelectual. Os principais obstáculos incluem:

- **Definição de Titularidade:** Quando um software é criado por um único desenvolvedor ou por uma empresa com equipe interna, a titularidade dos direitos é relativamente simples de determinar. No entanto, em projetos colaborativos, nos quais múltiplos indivíduos contribuem para o código-fonte, a atribuição dos direitos pode ser controversa. A titularidade pode depender de fatores como termos de contribuição, contratos de colaboração, políticas de repositórios de código e normas internas das plataformas de desenvolvimento, como GitHub e GitLab. Sem uma definição clara, há risco de disputas judiciais sobre a propriedade do software.
- **Licenciamento e Uso:** Projetos colaborativos frequentemente combinam diferentes licenças de software, o que pode gerar conflitos jurídicos. Softwares licenciados sob licenças permissivas (como MIT e Apache) podem ser incorporados a produtos comerciais sem grandes restrições, enquanto aqueles sob licenças copyleft (como GPL) impõem a obrigação de manter o código aberto em redistribuições. O desafio ocorre

quando um software incorpora componentes sob diferentes tipos de licença, tornando necessária a análise de compatibilidade para evitar violações. Além disso, a escolha inadequada de uma licença pode limitar a adoção do software por empresas ou órgãos públicos.

- **Prova de Autoria:** Em projetos de desenvolvimento individual, a comprovação de autoria pode ser feita por meio de registros de código e documentação técnica. No entanto, em ambientes colaborativos, onde diversas pessoas contribuem continuamente para o código-fonte, a identificação dos autores originais pode ser complexa. Para garantir a rastreabilidade das contribuições, práticas como versionamento de código (exemplo: Git), registro de commits e assinaturas digitais são fundamentais. No entanto, mesmo com essas ferramentas, pode haver dificuldades na comprovação de autoria em disputas legais, especialmente quando colaboradores contribuem de forma anônima ou sem contratos formais.
- **Governança e Compliance:** Empresas e comunidades que desenvolvem software colaborativo precisam estabelecer diretrizes claras para a gestão da propriedade intelectual. Questões como a definição de termos de contribuição, a política de licenciamento e a governança do código são fundamentais para evitar disputas jurídicas. Projetos comunitários bem-sucedidos, como o Linux e o Apache, possuem fundações e conselhos responsáveis por administrar a propriedade intelectual e resolver conflitos entre colaboradores. Já no setor empresarial, políticas internas de compliance devem garantir que a incorporação de software livre ocorra dentro dos limites legais, prevenindo riscos de infração de direitos autorais.
- Em suma, o desenvolvimento colaborativo de software desafia as estruturas tradicionais de proteção da propriedade intelectual, exigindo novas abordagens para a definição de direitos e responsabilidades. Sem mecanismos claros de governança, há risco de insegurança jurídica para desenvolvedores e empresas que desejam utilizar ou comercializar softwares criados em colaboração.

4. Implicações Econômicas e Sociais do Desenvolvimento Colaborativo

O desenvolvimento colaborativo de software não apenas impacta o setor tecnológico, mas também tem reflexos significativos na economia global e na inclusão digital:

- **Redução de Custos:** Softwares open-source reduzem os custos de aquisição e manutenção de soluções tecnológicas para empresas e governos, permitindo maior investimento em inovação e infraestrutura.
- **Aumento da Inovação:** Ambientes colaborativos promovem a troca de conhecimento e o desenvolvimento ágil, impulsionando a criação de soluções tecnológicas disruptivas.
- **Inclusão Digital e Educação:** O acesso aberto ao código-fonte e a ampla disponibilidade de ferramentas colaborativas facilitam o aprendizado de programação e capacitação profissional, democratizando o conhecimento tecnológico.
- **Transformação no Mercado de Trabalho:** O modelo colaborativo altera as dinâmicas do mercado, substituindo postos de trabalho tradicionais por novas oportunidades de desenvolvimento remoto, contribuindo para a descentralização do setor de tecnologia.
- **Impacto na Soberania Tecnológica:** Governos e instituições públicas podem se beneficiar da adoção de softwares de código aberto para reduzir a dependência de soluções proprietárias estrangeiras, fortalecendo a autonomia digital.

Pesquisas como as de Benkler (2006) demonstram que o modelo de colaboração aberta fortalece economias emergentes e incentiva a inovação descentralizada. Além disso, autores como Weber (2004) analisam a governança do software livre como um fator-chave para sua sustentabilidade econômica e seu impacto social.

5. Tendências Futuras e Recomendações

O futuro do desenvolvimento colaborativo de software aponta para avanços significativos, impulsionados por novas tecnologias e mudanças regulatórias. Dentre as principais tendências, destacam-se:

- **Maior integração de inteligência artificial** para facilitar a colaboração em larga escala, permitindo automação de tarefas repetitivas, revisão de código e assistência na programação.
- **Expansão de contratos inteligentes (smart contracts)** baseados em blockchain para gerenciar direitos de propriedade intelectual, garantindo registros imutáveis de autoria e licenciamento.
- **Reformas legislativas** para melhor acomodar novas formas de desenvolvimento de software, equilibrando proteção à propriedade intelectual e estímulo à inovação.

- **Adoção de modelos de negócios híbridos** que combinam software aberto com serviços pagos, permitindo maior sustentabilidade econômica para projetos colaborativos.
- **Melhor regulamentação da propriedade intelectual**, com a harmonização de normas entre diferentes jurisdições para evitar litígios internacionais e fomentar a cooperação global.

Essas tendências já foram apontadas por autores como Lessig (1999) e Stallman (2002), que destacaram a necessidade de adaptar a legislação ao avanço tecnológico. Eles sugerem maior flexibilidade na proteção e no compartilhamento de código, além de incentivos para a adoção de licenças abertas. Estudos mais recentes, como os de O'Mahony (2007) e West & O'Mahony (2008), ressaltam a importância da governança e da participação comunitária em projetos colaborativos, indicando que políticas mais transparentes e inclusivas podem fortalecer o ecossistema do software livre.

6. Conclusão

O desenvolvimento colaborativo de software representa uma verdadeira revolução na forma como aplicações são criadas, distribuídas e aprimoradas. Com a crescente adoção de modelos colaborativos, o impacto dessa abordagem transcende o âmbito técnico e afeta setores econômicos, sociais e jurídicos. A democratização do acesso ao desenvolvimento de software por meio de plataformas de código aberto e ferramentas no-code impulsiona a inovação e reduz barreiras de entrada para novos desenvolvedores e empresas.

No entanto, essa revolução traz desafios consideráveis no que tange à proteção jurídica dessas inovações. A titularidade dos direitos autorais em projetos colaborativos, a compatibilidade entre diferentes licenças de software e a governança sobre contribuições descentralizadas são apenas alguns dos aspectos que demandam atenção dos operadores do direito e formuladores de políticas públicas. A ausência de diretrizes claras pode gerar insegurança jurídica e entraves à adoção de modelos colaborativos, especialmente no contexto do registro de software no INPI e da compatibilidade com normas internacionais de propriedade intelectual.

Dante desse cenário, é essencial promover regulamentações adequadas e fortalecer políticas públicas que garantam segurança jurídica sem comprometer a inovação. O equilíbrio entre proteção intelectual e estímulo ao desenvolvimento colaborativo será determinante para o futuro da indústria de software e sua sustentabilidade a longo prazo.

Referências

- BENKLER, Yochai. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press, 2006.
- LESSIG, Lawrence. *Code and Other Laws of Cyberspace*. Basic Books, 1999.
- O'MAHONY, Siobhan. The Emergence of Governance in an Open Source Community. Harvard Business School, 2007.
- RAYMOND, Eric S. *The Cathedral and the Bazaar*. O'Reilly Media, 2001.
- STALLMAN, Richard. *Free Software, Free Society: Selected Essays of Richard M. Stallman*. GNU Press, 2002.
- WEBER, Steven. *The Success of Open Source*. Harvard University Press, 2005.