

Desenvolvimento Aberto



Licenças de Software

Versão 2022/2: Fabrício Barth (fabricaojb@insper.edu.br)

Até agora

- Ferramentas de colaboração
- Modificações de código
- Tradução de UI e documentação
- Distribuição de software
- Teste de software

Conteúdo técnico: ferramentas, fluxos de trabalho, etc

Hoje

Licenças de software

Licenças de software

- Governam a **distribuição** de um software
- Cessão de direitos do autor para outras pessoas
- Listam quais permissões um usuário (não) tem

Disclaimer

Eu não sou especialista em direito autoral brasileiro. Toda informação disponibilizada foi fruto de pesquisa e não constitui aconselhamento legal.

Distribuição de software

Escrever software é considerado um *trabalho intelectual*, estando então sujeito às leis de direitos autorais.

- Essas leis são específicas de cada país
- Autoria vs exploração econômica
- Engloba tanto código fonte como binários

Direito autoral brasileiro

Não é necessário registro. Não vale para ideias.

- **Direito moral:**
 - assegura a autoria da obra ao autor original
 - integridade da obra
 - intransferível e irrevogável
- **Direito patrimonial:**
 - exploração econômica da obra.
 - licenciado ou cedido
 - **copyright** em inglês

Por que isto é importante?

O dono dos direitos autorais (patrimoniais) pode controlar:

1. Distribuição, parcial ou total;
2. Criação de obras derivadas
3. Reprodução (execução)

Por que isto é importante?

De quem é o código nos seguintes cenários?

Um funcionário trabalhando para uma empresa produz software durante o expediente.

Por que isto é importante?

De quem é o código nos seguintes cenários?

Um funcionário trabalhando para uma empresa produz software fora do expediente e decide lançar um produto não relacionado.

Por que isto é importante?

De quem é o código nos seguintes cenários?

Um funcionário trabalhando para uma empresa produz software fora do expediente e decide lançar um produto concorrente.

Por que isto é importante?

De quem é o código nos seguintes cenários?

Um programador faz um freelance e entrega um projeto para uma empresa. Ele pode reutilizar parte do código em outros projetos (para outras empresas)?

Por que isto é importante?

De quem é o código nos seguintes cenários?

Um prestador de serviços (contratado como empresa via CNPJ) participa de um projeto terceirizado, junto com outros PJs.

Por que isto é importante?

Nenhum dos casos acima tem respostas fixas. Tudo depende de um monte de fatores jurídicos e do quê cada um assinou. Alguns guias interessantes:

- [Association of Software Professionals](#)
- [Start-se](#)
- [Legal Nature - EULA](#)

Contexto 1: Propriedade intelectual e *work-for-hire*

- Contexto brasileiro: "Pela nossa lei de software, a não ser que expressamente dito o contrário, aquele que contrata o desenvolvimento é titular original do software resultante" [1]
- Contexto americano: *work-for-hire* determina que software feito sob encomenda pertence a quem contratou o serviço [2]

Fontes [1](#) [2](#)

Contexto 2: projetos *open source*

Contributor License Agreement é um documento de cessão de copyright (direitos de exploração comercial).

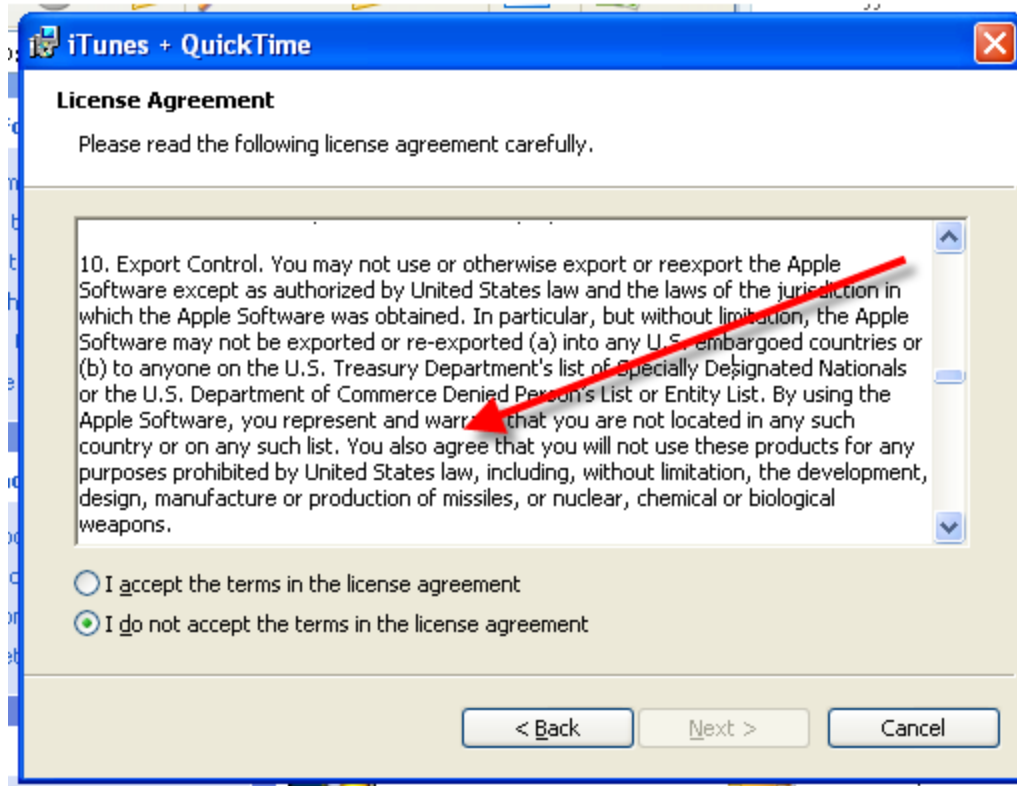
- Dá ao projeto uma licença não exclusiva (normalmente) e irrevogável para o projeto explorar sua contribuição
- Mantém autoria
- Isenta o projeto de responsabilidades sobre suas contribuições

Contexto 3: comercialização de software

End User License Agreement (EULA): Termo usado para aquelas condições que (não lemos) aceitamos ao instalar software proprietário ou nos registramos em um site/serviço.

Ignorado por aproximadamente 90% dos usuários ([fonte](#))

End User License Agreement



E por algumas empresas... [Fonte](#)

End User License Agreement

Restrições típicas encontradas em EULAs:

- Software é licenciado, não vendido;
- É proibido engenharia reversa
- Não pode modificar ou explorar limitações / vulnerabilidades
- Isenta responsabilidades pelo (mau) uso do software

EULA (serviços)

Informações típicas encontradas em EULAs de serviços:

- Restrições de idade;
- Uso dos dados coletados;
- Término unilateral do serviço
- Garantias de disponibilidade e responsabilidade

Legal Nature - EULA

EULA (AWS)

57.10 Acceptable Use; Safety-Critical Systems. Your use of the Lumberyard Materials must comply with the [AWS Acceptable Use Policy](#). The Lumberyard Materials are not intended for use with life-critical or safety-critical systems, such as use in operation of medical equipment, automated transportation systems, autonomous vehicles, aircraft or air traffic control, nuclear facilities, manned spacecraft, or military use in connection with live combat. However, this restriction will not apply in the event of the occurrence (certified by the United States Centers for Disease Control or successor body) of a widespread viral infection transmitted via bites or contact with bodily fluids that causes human corpses to reanimate and seek to consume living human flesh, blood, brain or nerve tissue and is likely to result in the fall of organized civilization.

fonte: <https://aws.amazon.com/service-terms/>

Um pouco de história

Um pouco de história



Richard Stallman fez um interpretador Lisp (MIT ~ 1985)

Um pouco de história



- Uma empresa chamada *Symbolics* licenciou o LISP do MIT para rodar em máquinas como a acima

Um pouco de história

- Uma empresa chamada *Symbolics* licenciou o LISP do MIT para rodar em máquinas como a acima
- Fez modificações no software
- Deu um jeito de não devolvê-las para o MIT....
- E eliminou a competição...

Um pouco de história

- Stallman ficou `#$@#$$@#$$@`
- e criou a primeira licença de software *livre*: *Emacs General Public License*

Principal característica:

1. Toda cópia do programa deverá estar acompanhada do código fonte;
2. Se uma modificação do software for distribuída, ela deverá estar sujeita aos mesmos termos do software original;

Um pouco de história - 1985



- [GNU Manifesto](#): descrevia um esforço em criar um sistema completamente livre cujo destino estivesse nas mãos dos usuários
- GNU/Hurd: todas as ferramentas de linha de comando das distribuições modernas vieram desse projeto

Um pouco de história - 1990



- Linus Torvalds tinha um kernel **experimental** "pronto".
- Liberava uma versão nova toda semana;
- Recebia contribuições de outros interessados.

Um pouco de história

Leitura complementar: Free Software, Free Society: Selected Essays of Richard M. Stallman.



Um pouco de história

Leitura complementar: Just for Fun: The Story of an Accidental Revolutionary, Linus Torvalds and David Diamond



Licenças de software (livre)

Criadas pela Free Software Foundation visando garantir 4 liberdades para um usuário ao usar um software:

0. Execução sem restrições
1. Estudo e modificação do software;
2. Redistribuir cópias;
3. Redistribuir cópias das suas modificações.

Distribuir os fontes é pré-requisito dos itens 1 e 3!

Chamadas também de *recíprocas*.

Licenças de software (livre)



0. RUN



1. STUDY



2. SHARE



3. IMPROVE

Licenças de software livre - recíprocas



Copyleft: Símbolo é o contrário do copyright, cujo objetivo seria restringir os usuários

Open Source Initiative



- Fundada para difundir um discurso menos anti-comercial e ideológico
- Também difunde benefícios "pragmáticos" do uso de software "de código aberto"
- *Free* causa confusão em inglês
 - Grátis (*free as in beer*)
 - Livre (*free as in freedom*)
- Fundada por Eric Raymond e Bruce Perens

Licenças de software de código aberto

Mais associadas a [Open Source Initiative](#), são consideradas menos "ideológicas".

- Muitas vezes chamadas de *permissivas*;
- Normalmente não garantem a liberdade 3;
- Podem fazer outras pequenas exigências ou restrições

Exemplos: MIT, BSD

- Permitem a distribuição de software com partes proprietárias
- Permitem a venda e completo *rebranding* do software sem nenhuma contribuição para o autor original.

Licenças: qual usar?

<https://choosealicense.com/licenses/>

Atividade

Cultura livre: fazer uma pesquisa profunda e apresentar seminário sobre licenças de software e casos recentes em que licenças tem papel de destaque

Sua apresentação precisa ter alguma forma de interação com a audiência

- debate
- painel com votações

Atividade - sugestões de temas

- [The new open-source economics \(Yochai Benkler, 2005\)](#). O que mudou de 2005 para 2022?
- There is no "Linux" platform.
- Telemetria é "ético"? (exemplo VSCode e outros).
- GitHub CoPilot é "ético"? Respeita licenças de software?
- Regulamentações recentes na Europa sobre durabilidade de smartphones (bateria, right to repair, etc).
- Maintainer burnout.

Atividade - datas de apresentação

- 17/11 - 2 trabalho
- 22/11 - 2 trabalhos
- 24/11 - 2 trabalhos
- 29/11 - 2 trabalhos

45 minutos de apresentação/atividade + 10 minutos para perguntas

Hoje

- Definição de temas e datas de apresentação
- Prévia de 5 minutos do seu tema na próxima aula

Registrar seu grupo, escolha de tema e dia de apresentação na issue

#665

Desafio de tradução

Skill "Tradução aceita!" (5XP): Ao menos 40 string traduzidas

Desenvolvimento Aberto



Licenças de Software

Versão 2022/2: Fabrício Barth (fabricaojb@insper.edu.br)