



# Por que você deve estudar isso?

---

- As leis se aplicam a **você**
- Batalhas judiciais
- Reputação do projeto
- Preocupação crescente nas organizações



# Por que proteger o software?

---

- Investimento financeiro
- Futuros desenvolvimentos
- Incentivo para a divulgação
- Base para o comércio
- Vulnerabilidade do software

# Aspectos Jurídicos

---

- Necessidade de uma legislação específica
- Software é um *fato* jurídico de caráter único
  - mercadoria (*res*)
  - bem imaterial
- Produto ou serviço?



# Código Fonte x Código Binário

---

- Fonte
  - obra literária
- Binário
  - produzido pelo computador
  - sem legibilidade humana
- cópia do binário é uma cópia do fonte?



# Histórico no Brasil

---

- 1983: relatório do escritório brasileiro de patentes
  - benefício da sociedade no longo prazo
  - legislação de outras criações tecnológicas
- 1984: anteprojeto de lei redigido por SEI/INPI/CNDA
  - proteção mediante registro
  - algumas semelhanças não constitui violação

# Histórico no Brasil

---

- 1987: Lei 7.646: escolha do direito autoral

# Histórico no Mundo

---

- 1970: OMPI
  - “Disposições Tipo para a Proteção do Software”
- Década de 1980: vários países adotando o direito autoral
  - proteção rápida e eficaz
  - adoção de regras específicas



# Histórico no Mundo

---

- 1991: UE - Diretiva 91/250/CEE
  - *droit d'auteur* -> *common law*
- Década de 1990: acordo TRIPs
  - programas de computador (fonte ou objeto) protegidos como obra literária pela Convenção de Berna
    - mais restritivo do que obra de arte aplicada

# Lei do Software

---

*Programa de computador é a expressão de um conjunto organizado de instruções em linguagem natural ou codificada, contida em suporte físico de qualquer natureza, de emprego necessário em máquinas automáticas de tratamento da informação, dispositivos, instrumentos ou equipamentos periféricos, baseados em técnica digital ou análoga, para fazê-los funcionar de modo e para fins determinados.*

- <http://www.planalto.gov.br/ccivil/Leis/L9609.htm>
- <http://www.planalto.gov.br/ccivil/Leis/L9610.htm>
- Código Civil

# Direito Autoral e a Lei do Software

---

- **Justificativa:**
  - balancear os interesses do detentor dos direitos com os do público
  - benefício de toda a sociedade no longo prazo
  - funcionalidade e valor econômico mais importantes do que um valor cultural associado
  - maior liberdade de concorrência

# Direito Autoral e a Lei do Software

---

- Formas de expressão vs. idéias
  - direito de exclusividade do autor é exercido sobre o código
  - algoritmo e estruturas de dados não são protegidos
- Direito autoral coibe a pirataria
- Não se aplicam as disposições relativas aos direitos morais

# Elementos Não Literais

---

- Proteção do *Look & Feel*
  - *Apple vs. Microsoft*
  - *Lotus vs. Borland*
- Estrutura e organização do programa
  - Base de dados
- Hipótese de integração de um programa a um outro sistema

# Elementos Não Literais

---

- Semelhança
  - em características funcionais da aplicação não constitui infração
- Cópia
  - implementa os procedimentos lógicos de maneira substancialmente similar
- Originalidade
  - mais de uma alternativa

# Contratos

---

- Lei 9.609 Art. 9º: “o uso de programa de computador no País será objeto de contrato de licença”
  - autorização de uso
  - não há cessão dos direitos patrimoniais

# Contratos

---

- Acordo de vontades
  - na conformidade da lei
  - finalidade de adquirir, resguardar, transferir, conservar, modificar ou extinguir direitos
- Liberdade de contratação é regra
  - o que a lei não proíbe, o que não afronta a ordem pública, é permitido



# Tipos de Contratos

---

- Típico vs. Atípico
  - interpretado de acordo com o contrato típico mais próximo e aos contratos em geral
- Contrato de adesão
  - cláusulas ambíguas interpretadas favoravelmente à parte que não teve o poder de influenciar no conteúdo

# Tipos de Contratos

---

- **Contrato benéfico**
  - apenas uma das partes auferir os benefícios, enquanto a outra sofre os encargos
  - estipulações em favor de terceiro
    - não é pré-determinado quem serão as demais pessoas que se beneficiarão da licença
- *Contrato de licenciamento em rede*
  - contrato aberto a qualquer membro

# Contratos e Licenças Recíprocas

---

- Licenciado de hoje seja *ipso facto* o licenciante de amanhã
- Usuário exerce a priori as decisões que a ele caberia pelo direito de autor
- Partes são interdependentes, agindo ao mesmo tempo como contratantes e contratados

# Contratos e Licenças Recíprocas

---

- Oferta *erga omnes*, constituindo uma rede aberta
- Não cria escassez legal como no contrato liberal clássico
- Uniformidade contratual
  - qualquer aperfeiçoamento pode beneficiar igualmente todas as partes

# Responsabilidade Civil

---

- Código Civil, artigo 392º
  - nos contratos benéficos, responde por simples culpa o contratante, a quem o contrato proveite, e por dolo aquele a quem não favoreça. Nos contratos onerosos, responde cada uma das partes por culpa, salvo exceções previstas em lei

# Responsabilidade Civil

---

- Não existe “fraqueza do consumidor”, ninguém é obrigado ou compelido de qualquer forma a aceitar a licença e utilizar o software livre
- Relação não é bipolar (consumidor e o fornecedor), envolve interesses de toda a coletividade

# Responsabilidade Civil

---

- Outras relações jurídicas podem ser estabelecidas a partir do software livre, que podem incorrer em outras obrigações e responsabilidades

---

# Licenças de Software Livre



# Licenças de Software Livre

---

- Licença é distribuída junto com o fonte
- Detentor escolhe quais direitos estará transferindo e quais as condições
- Inúmeras possibilidades para redigir o texto

# Licenças de Software Livre

---

- Importante evitar proliferação de licenças
  - Melhor reaproveitar licenças já consolidadas
- Portais apresentam grande número de licenças
  - Sourceforge: 74 licenças
  - OSI: lista com 65 licenças aprovadas
    - Vários novos pedidos todo mês

# Open Source Initiative

---

- Redistribuição Livre
- Código Fonte
- Trabalhos Derivados
- Integridade do Código Fonte do Autor
- Sem Discriminação a Pessoas ou Grupos
- Sem Discriminação a Áreas de Empreendimento
- Distribuição da Licença
- A Licença Não Deve Ser Específica a um Produto
- A Licença Não Deve Restringir Outro Software
- A Licença Deve Ser Neutra às Tecnologias

# Categorias de Licenças

---

- Permissivas
- Recíprocas Totais
- Recíprocas Parciais

# Licenças Permissivas

---

- Licenças acadêmicas
- Usadas em projetos de pesquisa / provas de conceito
- Não impõem grandes restrições
- Permitem atingir maior número de pessoas
- Exemplo: TCP/IP do **BSD** Unix

# Licenças Permissivas

---

- Empresas podem fechar o software
- Exemplo: Kerberos (**MIT**) e Microsoft
- Contra-exemplo: servidor **Apache**
  - marca forte
  - importância dos padrões

# Licenças Recíprocas Totais

---

- Trabalho derivado precisa ser distribuído sob os mesmos termos
- Permissão para executar, copiar, modificar e distribuir versões modificadas do programa
- Impedem que sejam adicionadas novas restrições

# Licenças Recíprocas Totais

---

- Resultados esperados:
  - aumentar quantidade de software livre
  - preservar a compatibilidade entre versões
- Problema:
  - software isolado



# Licenças Recíprocas Totais

---

- Principal exemplo: GPL
  - 2.0
  - GPLv3
  - AGPL

# O que há de novo na GPLv3?

---

- Facilitar compatibilidade com outras licenças (Apache 2.0)
- Evitar *tivoização* (assinaturas digitais)
- Punir acordos de patentes discriminatórios (Microsoft)

# O que há de novo na GPLv3

---

- Invalidar ilegalidade de quebra de DRM em software livre
- Adequar a licença a compartilhamento *peer-to-peer*
- Evitar interpretações diferentes das previstas decorrentes de legislações locais

# Licenças Recíprocas Parciais

---

- Modificações disponibilizadas sob a mesma licença
- Trabalho pode ser usado em outro projeto, com outra licença
- Equilibra dois fatores:
  - atração de interesse para a comunidade
  - força e longevidade do código fonte disponível

# Licenças Recíprocas Parciais

---

- “Baseadas em arquivo”
- Argumento da FSF: usar apenas quando já existem alternativas
- Exemplos:
  - LGPL
  - Mozilla Public License

---

# Compatibilidade entre Licenças

# Importância da Compatibilidade

---

- Reputação do projeto
- Complicações legais

# Fatores a Serem Considerados

---

- Leis locais
- Cláusulas das licenças
- Formas de integração
- Formas de distribuição



# Compatibilidade por Categoria

---

- Licenças Permissivas
  - BSD License
    - Original
    - Simplificada
    - Duas cláusulas
  - MIT/X11 License
  - Apache Public License

# Compatibilidade por Categoria

---

- Licenças Recíprocas Totais
  - GPL 2.0
  - GPLv3
  - AGPL

# Compatibilidade por Categoria

---

- Licenças Recíprocas Parciais
  - Mozilla Public License
  - LGPL

# Integração entre Sistemas

---

- Link estático
- Link dinâmico
- Plug-in
  - fork
  - exec
  - compartilhamento de estrutura de dados
  - chamada de funções

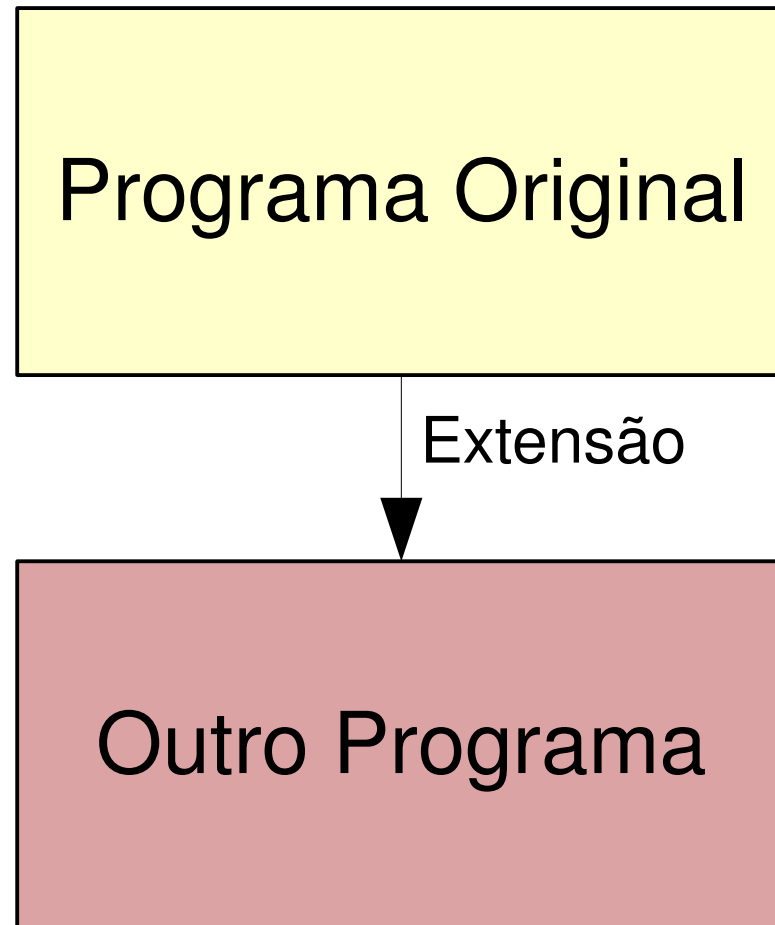
# Integração entre Sistemas

---

- **Agregação / Combinação**
  - mecanismo de comunicação
    - exec, pipes, rpc, chamada de função dentro de um endereço compartilhado...
  - semântica da comunicação
    - tipos de informação que são compartilhadas

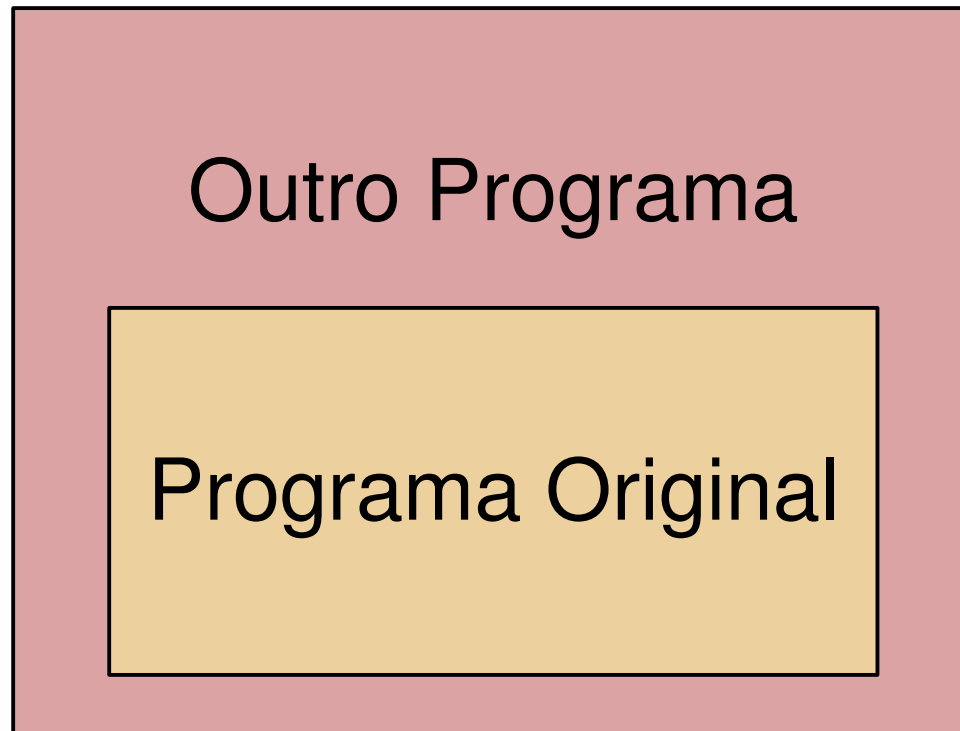
# Integração entre Sistemas

---



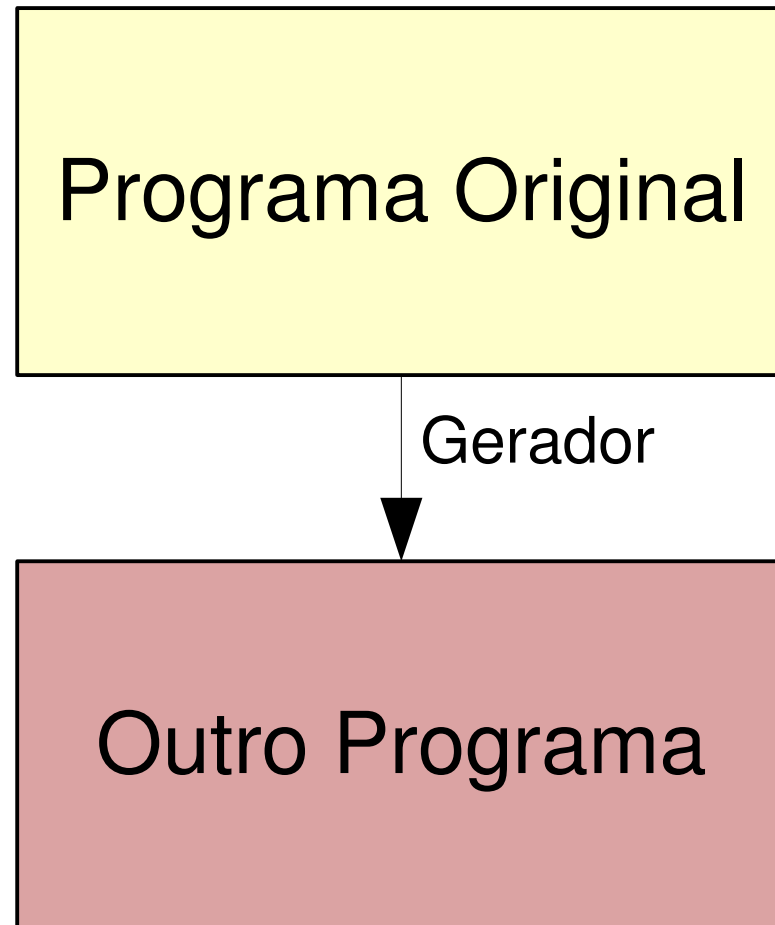
# Integração entre Sistemas

---



# Integração entre Sistemas

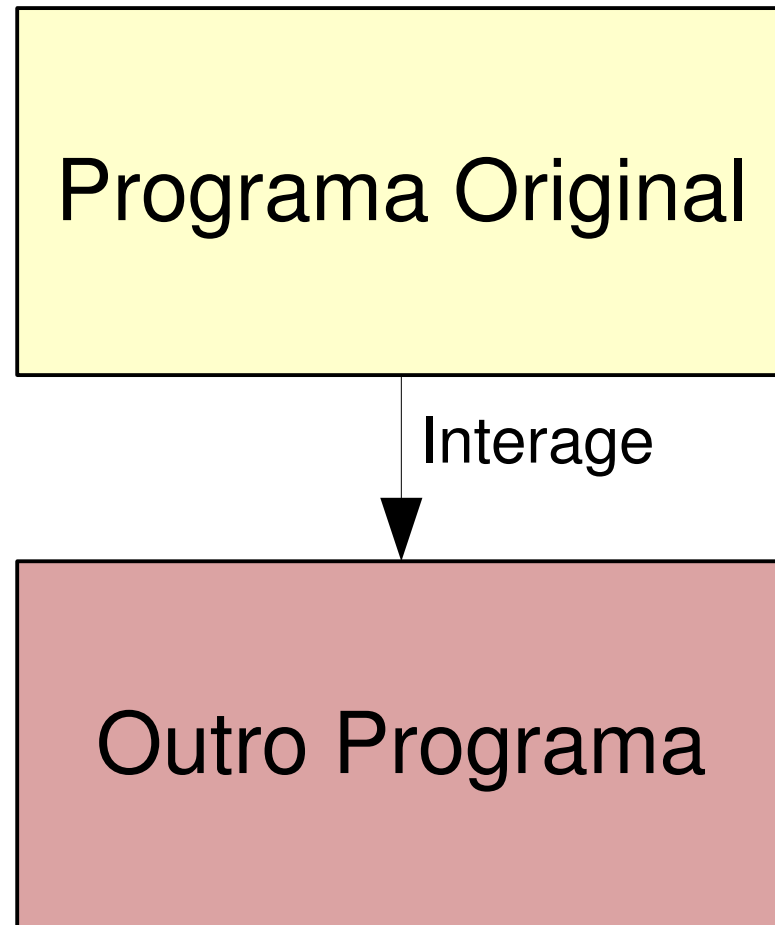
---





# Integração entre Sistemas

---



# Integração entre Sistemas

	Programa	Outro Programa
Extensão	GPL LGPL	GPL LGPL
Combinação	GPL LGPL	GPL LGPL
Geração	GPL LGPL	Depende do programa conter partes do original
Interação	GPL LGPL	Depende da forma de interação