

## Software Livre (Aula 04)

Profa. Dra. Elisa Yumi Nakagawa  
2. Semestre de 2008

## Conteúdo

- Histórico de Software Livre
- Terminologia
- Processo de Desenvolvimento
  - Fases do Processo de Desenvolvimento
  - Infra-estrutura
- Licença de Software Livre
- Ferramentas de Software Livre
  - Linux
  - Apache
  - PHP
  - MySQL
- Utilização de Software Livre

## Histórico de Software Livre

- Quando os computadores chegaram às universidades e eram objeto de pesquisa, o código fonte era livre.
- Na década de 80, o software passou a ser tratado como um produto e a gerar lucros.
- Em 1984, Richard Stallman fundou a *Free Software Foundation (FSF)* e o projeto GNU, dando origem ao movimento de Software Livre.
- Nos últimos anos, este movimento teve um grande impulso, motivado principalmente pelo advento do sistema operacional livre Linux.

## Terminologia

- Alguns termos que devem ser considerados quando se discute software livre:
  - Software livre (*free software*),
  - Código aberto (*open source*),
  - Software proprietário,
  - *Freeware*,
  - *Shareware*, e
  - Domínio Público.

## Terminologia: Software livre

- Software livre pode ter diversos significados:
  - (i) a forma como o software é licenciado,
  - (ii) o conjunto de todos os softwares licenciados desta forma, e
  - (iii) uma comunidade organizada em torno de softwares licenciados desta forma.

## Terminologia: Software livre

- Um software livre deve proporcionar aos seus usuários as seguintes liberdades principais [FSF]:
  - A liberdade de executar o programa para qualquer propósito;
  - A liberdade de estudar o funcionamento do programa e adaptá-lo às suas necessidades.
  - A liberdade de redistribuir cópias do código-fonte; e
  - A liberdade de aperfeiçoar o programa e distribuir o software juntamente com os aperfeiçoamentos de modo que toda a comunidade beneficie-se.

### Terminologia: Código Aberto

- Ultimamente código aberto e software livre têm sido utilizados como tendo o mesmo significado.
- Foi proposto por Raymond preocupado em disseminar entre as empresas a idéia de software livre, uma vez que havia uma forte associação do termos software livre com hostilidade aos direitos de propriedade intelectual, comunismo, entre outras, que poderiam prejudicar o sucesso do software livre.
- Em 1998, parte da comunidade de software livre decidiu adotar o termo software de código aberto.
  - Movimento de Código Aberto.

### Terminologia: Software proprietário e comercial

- Software proprietário é uma categoria de software em que os usuários não podem ter acesso ao código-fonte do programa.
- São a maioria dos softwares em uso atualmente.
- Muitas vezes, confundido com software comercial.
  - Software comercial é desenvolvido com o objetivo de gerar lucros com sua distribuição.

### Terminologia: *Shareware*

- Softwares classificados como shareware são distribuídos como binários, e, muitas vezes, é requisitado o pagamento de uma “pequena” taxa após um determinado tempo de utilização do software.

### Terminologia: *Freeware*

- Freeware foi bastante utilizado na década de 80 para referir-se aos softwares distribuídos gratuitamente, no qual somente o código executável era disponibilizado.

### Terminologia: Domínio Público

- Software de domínio público é uma classe de softwares que não possuem um copyright, ou seja, o autor abandonou os direitos sobre o software e, por isso, pode ser utilizado e modificado sem nenhuma restrição.

### Terminologia

Categoria	Copyright	Código-Fonte Disponível	Distribuído Gratuita	Permissão de Alteração	Número de Usuários Limitado
Proprietário	Sim	Não	Não	Não	Sim
Shareware	Sim	Não	Sim	Não	Sim
Freeware	Sim	Não	Sim	Não	Não
Software Livre	Sim	Sim	Sim	Sim	Não
Domínio Público	Não	Sim	Sim	Sim	Não

## Características de Software Livre

- Alguns fatores que têm impulsionado o software livre:
  - Estabilidade
  - Confiabilidade
  - Portabilidade
  - Baixo Custo
  - Suporte dos desenvolvedores aos usuários
  - Evolução contínua
  - Acesso ao código fonte

## Licença de Software Livre

- *Copyright*
  - determina o proprietário do software.
  - protege a propriedade intelectual
  - assegura ao seu “dono” -- autores, artistas e outros -- o direito exclusivo para reproduzir e derivar lucros de seus esforços.
- Um software que possui um *copyright* não pode ser vendido ou distribuído sem a permissão de seu dono.

## Licença de Software Livre

- Licença de software:
  - é um documento que estabelece a forma como o dono do *copyright* permite a distribuição e cópia de um software
  - estabelece um *copyright* que pode proteger o proprietário de algum bem intelectual.
- Com relação aos softwares proprietários:
  - usuário não compra o software em si, mas sim as licenças para uso do software
  - Software continua em propriedade do dono (empresa ou autor) do software

## Licença de Software Livre: Copyleft

- *Copyleft*:
  - um trocadilho com o termo *Copyright* e pode ser traduzido como Deixar Copiar
  - maneira de fazer com que programas que são software livre, além de todas as modificações e versões posteriores do programa continuem sendo livres
    - não é permitido adicionar restrições de forma a negar às outras pessoas as liberdades principais.

## Licença de Software Livre: GPL

- Na comunidade de software livre, a GPL é a licença mais conhecida.
  - Foi proposta pela FSF
  - Características Principais:
    - liberdade de distribuir cópias de software (e cobrar por isso, se desejar)
    - ter acesso ao código-fonte
    - poder realizar modificações no software
    - utilizar partes dele em novos programas livres
    - garantir que o software continuará sendo livre e gratuito para os seus usuários

## Licença de Software Livre:

### Categorias

- Licenças de software podem ser categorizadas em:
  - Licença de Software Livre
    - Compatível com a GPL
    - Não-Compatível com a GPL
  - Licença de Software Não Livre

## Licença de Software Livre:

### Categorias

#### 1.1. Licença de Software Livre:

##### Compatível com a GPL:

- Softwares que possuem este tipo de licença podem ter seus módulos combinados com softwares licenciados pela GPL
- Exemplos:
  - GNU Lesser General Public License (GNU LGPL ou simplesmente LGPL)
  - licença BSD
  - licença X11.

## Licença de Software Livre:

### Categorias

#### 1.2. Licença de Software Livre:

##### Não-Compatível com a GPL:

- Software ou módulos de softwares que possuem este tipo de licença não podem ser combinados com softwares licenciados pela GPL resultando em um software maior com licença não compatível com a GPL.
- Exemplos:
  - versão 1.0 e 1.1 da licença do Apache: a MPL (*Mozilla Public License*)
  - versão 2.02 da licença do PHP; e

## Licença de Software Livre:

### Categorias

#### 2. Licença de Software Não Livre

- São licenças pertencentes a softwares considerados não livres
- Inúmeras licenças desta classe; “dono” do software tem a possibilidade de propor uma licença diferente.
- licenças de softwares proprietários, freeware e shareware
- Exemplos:
  - Licença Comunitária de Código-Fonte da Sun: não concede liberdades essenciais como a de publicar uma versão modificada.

## Licença de Software Livre:

### Categorias

#### ■ Mudança de Categoria:

- versão 3 do PHP é regida por uma licença compatível com a GPL
- versão 4 do PHP possui uma licença não-compatível com a GPL

## Ferramentas de Software Livre

- Inúmeros softwares livres para os mais diversos domínios de aplicação e sendo utilizados para os mais diversos fins.
- Softwares proprietários têm sido desenvolvidos para serem utilizados em plataformas de software livre. Ex: Kylix
- Exemplos de Software Livre:
  - Linux
  - Apache
  - PHP
  - MySQL

## Ferramentas de Software Livre

#### ■ Linux

- Surgiu no início da década de 90.
- Baseado no Minix.
- Escrito por um estudante de Ciência da Computação da Finlândia: Linus Torvalds.
- Utilizado em sistemas embutidos, no controle de dispositivos robóticos, entre outros.

## Ferramentas de Software Livre

- Linux
  - Atualmente, o Linux pode ser considerado o sistema operacional mais portátil estando disponível para os diversos computadores pessoais.
  - O kernel implementa elementos comuns às diversas arquiteturas.
  - Código-fontes de hardwares específicos são confinados em módulos e não inseridos no kernel.

## Ferramentas de Software Livre

- Linux
  - O kernel do Linux está sob o controle de Linux Torvalds
  - Centenas de desenvolvedores estão trabalhando independentemente em diversas partes do sistema.
  - Pode ser executado em máquinas menos potentes do que as necessárias para rodar programas proprietário.
  - Na América Latina, nos últimos dois anos, o uso do Linux teve um crescimento de 915%.
  - Licença: GPL

## Ferramentas de Software Livre

- Apache (<http://www.apache.org>)
  - É um servidor HTTPD desenvolvido a partir do servidor NCSA original.
  - Multiplataforma.
  - Nos últimos anos, tornou-se o servidor de Web mais utilizado: 54% dos servidores (Netcraft, 2002) e 67,2% dos servidores (Netcraft, 2004)
  - O Apache possui uma série de listas de discussões, um repositório CVS e um sistema para Web de informe de defeitos.
  - Não possui nenhum líder, existindo para tanto um núcleo de desenvolvedores, bem como desenvolvedores secundários da comunidade.
  - As alterações no código-fonte são discutidas publicamente através de lista de discussões.
  - Licença: Apache Software License

## Ferramentas de Software Livre

- PHP (<http://www.php.net>)
  - Trata-se de uma linguagem de script para a criação de página dinâmicas para Web.
  - É um projeto da Apache Software Foundation
  - Características:
    - simplicidade
    - semelhante com linguagem como C e Perl.
    - fácil conectividade com diversos tipos de bases de dados, tais como Oracle, Sybase e MySQL.
    - multiplataforma

## Ferramentas de Software Livre

- PHP
  - Iniciou-se em 1984, no entanto, a partir de junho de 1998, com a versão 3.0, tem ganhado popularidade.
  - Atualmente, o PHP é a linguagem para páginas dinâmicas que está em maior crescimento. O uso de PHP tem crescido a uma taxa de 15% ao mês e tem sido utilizada em 36% dos servidores Apache (Netcraft, 04/2002).
  - Licença: PHP License

## Ferramentas de Software Livre

- MySQL (<http://www.mysql.org>)
  - Mantido por uma empresa que vende serviços sobre o MySQL.
  - Considerado um dos mais conhecidos sistemas de gerenciamento de banco de dados relacional.
  - Características:
    - rápido
    - confiável
    - multi-usuário
    - fácil utilização.
    - multiplataforma,
    - possui uma série de funções que podem ser chamadas por diversas linguagens de programação, tais como C, C++, Java, Perl, PHP e Teal

## Utilização de Software Livre

- No Brasil, software livre tem assumido importância cada vez maior em órgãos e autarquias do Governo Federal, Estadual e Municipal.
- SERPRO (Serviço de Processamento de Dados), maior órgão de informática do Governo Federal → utilização do Linux no ambiente Web.
- Estado do Rio Grande do Sul:
  - Projeto Software Livre RS
  - Procempa (Processamento de Dados do Município de Porto Alegre) → escolas municipais baseada em softwares livres.

## Utilização de Software Livre

- Grandes empresas têm investido no setor de software livre :
  - IBM (<http://www-124.ibm.com/developerworks/oss/>)
  - Sun (<http://www.openoffice.org/>)
  - HP
  - Dell
  - Oracle
  - Compaq
  - Intel
  - NEC, entre outras.
- Software livre na educação: ensino em universidades e no ensino fundamental

## Divulgação de Software Livre

- Diversos eventos têm mostrado a relevância do uso de software livre no Brasil e no mundo:
  - Fórum Internacional de Software Livre (<http://www.softwarelivre.rs.gov.br/forum/>)
  - Workshop on Open Source Software Engineering (<http://opensource.ucc.ie/icse2001/> e <http://opensource.ucc.ie/icse2002/>)
- Objetivo:
  - reunir pesquisadores, especialistas e parlamentares
  - disseminar conceitos referentes ao software livre, tais como patentes de software, oferta de emprego, entre outros.

## Processo de Desenvolvimento

- Modelo de desenvolvimento de software livre
  - é bastante diferente dos modelos tradicionais.
  - não está formalmente definido
- Melhor descrição foi escrita por Raymond (1999) e chamada de "*The Cathedral and Bazaar*" (O Catedral e o Bazar)

## Processo de Desenvolvimento

- Modelo Bazar:
  - Parte do princípio de que qualquer desenvolvedor pode contribuir para o desenvolvimento de um software
  - As contribuições serão filtradas de forma "darwiniana", sendo que o melhor código sobreviverá.
  - Projeto de sucesso:
    - Linux
    - Apache
    - Perl

## Processo de Desenvolvimento

- Características do Modelo Bazar:
  - dispersão geográfica dos desenvolvedores
  - desenvolvimento compartilhado e distribuído
  - tempo de desenvolvimento curto
  - alta qualidade tanto na presença de características desejáveis quanto em relação ao número reduzido de defeitos
  - tempo de resposta para a correção dos defeitos identificados no software

## Processo de Desenvolvimento

- Para se obter sucesso no Modelo Bazar:
  - objetivo bem definido do que será desenvolvido
  - motivação fácil de ser compreendida
  - bom líder que mantenha os desenvolvedores motivados e o objetivo em mente
  - comunidade de participantes que trabalhe com entusiasmo e de forma descentralizada
  - tecnologia que possibilita a comunicação de forma eficiente, como a Internet.

## Processo de Desenvolvimento

- Características do Modelo Catedral:
  - bastante utilizado tanto na indústria quanto pela comunidade acadêmica
  - desenvolvimento centralizado e controlado
  - necessidade de uma pessoa ou grupo de pessoas que centraliza o processo de desenvolvimento
  - demora no lançamento de novas versões

## Processo de Desenvolvimento:

### Fases do Processo de Desenvolvimento

- Apesar de não estar bem definido, o processo de desenvolvimento costuma apresentar:
  - Especificação de Requisitos
  - Projeto
  - Implementação

## Processo de Desenvolvimento:

### Fases do Processo de Desenvolvimento

#### 1. Especificação de Requisitos

- não é dado muita ênfase na especificação de requisitos
- muito raramente existe um documento formal de especificação
- maior parte dos softwares livres replica de alguma forma a funcionalidade de um ou mais softwares já existentes
- é difícil determinar se a inexistência de requisitos formalizados é um fator negativo para os projetos de softwares livres, uma vez que se observa a existência de inúmeros projetos de software livre de sucesso

## Processo de Desenvolvimento:

### Fases do Processo de Desenvolvimento

#### 2. Projeto

- maioria dos softwares livres apresenta modelos de projeto de mais alto nível
- não se pode identificar uma arquitetura bem definida do sistema a ser desenvolvido

## Processo de Desenvolvimento:

### Fases do Processo de Desenvolvimento

#### 3. Implementação

- implementação colaborativa e distribuída
- codificação inicia-se imediatamente após a sua idealização
- prototipação é uma prática bastante comum
- utilização de serviços disponibilizados pela Internet -- lista de discussões através do correio eletrônico, repositórios de versões, entre outros

## Processo de Desenvolvimento:

### Infra-Estrutura

- Geralmente, os requisitos mínimos para projeto de software livre:
  1. Endereço de Correio Eletrônico
  2. Lista de Discussão
  3. Repositório de Software
  4. Interface para a Solicitação de Mudanças e Apresentação de Defeitos
  5. Site na Web
- Requisito fundamental e básico: Internet

## Processo de Desenvolvimento:

### Infra-Estrutura

#### 1. Endereço de Correio Eletrônico:

- contato com o autor ou os responsáveis pela integração
- para envio de mudanças e solicitações

## Processo de Desenvolvimento:

### Infra-Estrutura

#### 2. Lista de Discussão:

- diversas listas para públicos-alvo diferentes
- Lista de Desenvolvedores:
  - mensagens referentes ao uso e operação do software
  - facilita a comunicação de pedidos e problemas com o software
  - participam usuários avançados que respondem às perguntas mais básicas
  - participam desenvolvedores que captam mensagens que tem impacto no desenvolvimento
- Lista de Usuários:
  - são discutidas as decisões de projeto
  - tem alto tráfego
  - apresentam discussões abordando questões mais elaboradas

## Processo de Desenvolvimento:

### Infra-Estrutura

#### 3. Repositório de Software:

- requisito fundamental que possibilita que o desenvolvimento distribuído
- agiliza o controle e análise das diversas versões do software
- um dos softwares mais utilizados para implantação de repositórios é o CVS (*Concurrent Versions System*)

## Processo de Desenvolvimento:

### Infra-Estrutura

#### 4. Interface para a Solicitação de Mudanças e Apresentação de Defeitos:

- É conectada aos endereços eletrônicos dos responsáveis, bem como com o dos desenvolvedores do projeto.
- Exemplos de interfaces: Gnats e Bugzilla.

## Processo de Desenvolvimento:

### Infra-Estrutura

#### 5. Site na Web:

- tem a finalidade de agrupar pessoas interessadas no software
- registrar as novidades associadas ao software
- fornecer informações das últimas versões





### Para Mais Informações:

- <http://www.fsf.org>
- <http://opensource.ucc.ie/>
- <http://www.openoffice.org/>
- <http://www.cipsga.org.br/>
- <http://www.softwarelivre.unicamp.br>
- <http://www.opensource.org/>