Tarefa-3: Em aula invertida, os alunos deverão realizar uma Pesquisa online em grupo (com até 4 pessoas) sobre “Software/Hardware Livre e Proprietário” e elaborar trabalho em Editor de texto ou Apresentação de Slides ou ainda em seu caderno abordando os temas pesquisados. Essa tarefa será avaliada como Pesquisa. Entrega até 31/05.

O QUE É SOFTWARE LIVRE?

Um software é considerado livre quando seu código-fonte está liberado segundo uma licença que dá mais liberdade de uso ao usuário que faz uso do programa.

A licença de código aberto mais comum é a GNU GPL (Licença de Uso Geral do Projeto GNU), que concede quatro liberdades ao usuário:

1. A liberdade de executar o programa para qualquer propósito (liberdade no 0)

2. A liberdade de estudar como o programa funciona e adaptá-lo às suas necessidades (liberdade no 1). O acesso ao código-fonte é um pré-requisito para esta liberdade.

3. A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo (liberdade no 2).

4. A liberdade de aperfeiçoar o programa e liberar os seus aperfeiçoamentos, de modo que toda a comunidade se beneficie deles (liberdade no 3). O acesso ao código-fonte é um pré-requisito para esta liberdade.

Todos os softwares são liberados segundo uma licença. Mesmo os softwares que não estão liberados segundo uma licença de forma explícita, são propriedade intelectual dos seus criadores.

FILOSOFIA DE USO

O software livre se baseia no modelo de SaaS (Software as a Service), ou seja, Software como um Serviço. Ele baseia-se na prestação de serviços fazendo uso daquele software. O lucro não está na venda de licenças do software, mas nos serviços agregados a ele. Entre eles, podemos listar alguns:

• Treinamento para uso do software livre, na forma de documentação escrita, cursos, apresentações, entre outros;

• Duplo licenciamento do código, permitindo que exista variações na versão do software;

• Integração com produtos de hardware, e temos inúmeros exemplos, de projetores de imagem a roteadores sem fio, de equipamentos de rede a celulares;

• Certificações, que tem por objetivo provar a proficiência do profissional em algum software livre em específico;

• Suporte ao software livre. Vez por outra, é necessário ter alguém que ajude nos momentos de dificuldade, assim como é necessário que correções no software sejam feitas;

• Serviços baseados em software livre, como os provedores de hospedagem.

• Redistribuição, que é a venda de mídias contendo software livre. Cobra-se pelo trabalho da cópia e pela mídia, não pelo software em si;

• Criação de extensões não-livres para algum software livre.

VANTAGENS

A maioria dos usuários não tem interesse em ter acesso ao código-fonte do software do qual está fazendo uso. Mas o código aberto traz diversas vantagens para quem o usa, como:

• Clara economia de recursos – não há necessidade de pagar pela licença de algum software que seja livre. Como o software é livre, ele pode ser distribuído sem custo algum. Vemos aí a economia de recursos financeiros;

• Correção de falhas – como o código- fonte é aberto, a correção de falhas é mais rápida, pois existem vários desenvolvedores investigando o código-fonte. Esta característica é enunciada pelo que conhecemos como a Lei de Linus, que diz: “Dados olhos suficientes, todos os erros são óbvios”;

• Compartilhamento do código- fonte – o desenvolvimento de novas aplicações pode ser simplificado, pois não é necessário que as mesmas sejam reescritas do zero. Com isto, é possível reduzir custos e minimizar a duplicação de esforços;

• Aumento no número de desenvolvedores – o número de desenvolvedores que participam do processo de redação do código-fonte pode ser maior, agrupando pessoas que não necessariamente estão no início do projeto. Por exemplo, temos inúmeros projetos de software livre liderados por empresas que recebem colaborações de desenvolvedores que não são funcionários dessa mesma empresa;

• Melhoria na qualidade do código – como o código-fonte é aberto, os desenvolve dores serão mais cautelosos ao redigi-lo, pois várias pessoas examinarão o código escrito. Sendo assim, sua produção será avaliada por outros profissionais e possivelmente terá reflexos em sua carreira profissional;

• Maior segurança – o código-fonte, sendo aberto, terá menos chance de conter falhas de segurança, ou backdoors. O backdoor é um recurso usado por alguns desenvolvedores em seus programas, de forma que eles possam ter acesso remoto ao seu sistema, caso precise. Por exemplo: é sabido que várias empresas, nos seus softwares de código fechado, capturam informações do computador dos clientes e remetem-na para seus servidores, com propósitos diversos. Uma backdoor colocada em um software de código aberto é rapidamente localizada e removida;

• Maior diversidade – do ponto de vista econômico, o software livre promove o estabelecimento de vários fornecedores com base no mesmo software. A competição entre fornecedores traz vantagens aos usuários, como melhorias na qualidade e redução nos preços de serviços agregados;

• Redução da dependência de empresas estrangeiras – uma vantagem do modelo é a menor dependência de serviços prestados por empresas fora do país. Como o código é aberto, qualquer empresa capacitada pode prestar serviços, não necessariamente aquela que desenvolveu o código. Isto permite que empresas locais façam o serviço que antes era repassado a uma empresa estrangeira. Isto desenvolve o mercado local e cria empregos.

O modelo de negócios ainda é novo e as empresas têm pouca experiência em lidar com o software livre. Logo, isto pode causar confusão sobre a viabilidade econômica do modelo, ou a qualidade do software. Estes conceitos estão sendo revertidos aos poucos, pois algumas empresas defendem a opinião de que é mais vantajoso aprimorar/contribuir com o software livre do que investir na construção de um novo software similar e proprietário.

O que é Hardware Livre?

é um [equipamento eletrônico](https://pt.wikipedia.org/wiki/Hardware) projetado e oferecido nos mesmos termos que um software de [código livre](https://pt.wikipedia.org/wiki/C%C3%B3digo_livre). O termo foi primeiramente empregado para refletir o lançamento irrestrito de [informação](https://pt.wikipedia.org/wiki/Informa%C3%A7%C3%A3o) sobre o projeto de hardware, tal como um diagrama, [estrutura de produtos](https://pt.wikipedia.org/wiki/Estrutura_de_produtos) e dados de layout de uma [placa de circuito impresso](https://pt.wikipedia.org/wiki/Placa_de_circuito_impresso).[[1]](https://pt.wikipedia.org/wiki/Hardware_livre#cite_note-info-1)

Com o crescimento dos [dispositivos lógicos reprogramáveis](https://pt.wikipedia.org/wiki/Dispositivo_l%C3%B3gico_program%C3%A1vel), o compartilhamento dos esquemas lógicos de forma aberta também se espalhou. Ao invés de compartilhar as esquemáticas, é o código da [linguagem de descrição de hardware](https://pt.wikipedia.org/wiki/Linguagem_de_descri%C3%A7%C3%A3o_de_hardware) que normalmente se divide. Suas descrições popularmente são usadas para configurar um [system-on-a-chip](https://pt.wikipedia.org/wiki/System-on-a-chip) ou através de [FPGA](https://pt.wikipedia.org/wiki/FPGA) ou diretamente via [ASIC](https://pt.wikipedia.org/wiki/ASIC). Estes módulos, quando distribuídos, são chamados de núcleos semicondutores de propriedade intelectual (Em inglês, [Semiconductor intellectual property core](https://en.wikipedia.org/wiki/Semiconductor_intellectual_property_core" \o "en:Semiconductor intellectual property core) - IP cores) ou Núcleos IP.

é um [equipamento eletrônico](https://pt.wikipedia.org/wiki/Hardware) projetado e oferecido nos mesmos termos que um software de [código livre](https://pt.wikipedia.org/wiki/C%C3%B3digo_livre). O termo foi primeiramente empregado para refletir o lançamento irrestrito de [informação](https://pt.wikipedia.org/wiki/Informa%C3%A7%C3%A3o) sobre o projeto de hardware, tal como um diagrama, [estrutura de produtos](https://pt.wikipedia.org/wiki/Estrutura_de_produtos) e dados de layout de uma [placa de circuito impresso](https://pt.wikipedia.org/wiki/Placa_de_circuito_impresso).[[1]](https://pt.wikipedia.org/wiki/Hardware_livre#cite_note-info-1)

Com o crescimento dos [dispositivos lógicos reprogramáveis](https://pt.wikipedia.org/wiki/Dispositivo_l%C3%B3gico_program%C3%A1vel), o compartilhamento dos esquemas lógicos de forma aberta também se espalhou. Ao invés de compartilhar as esquemáticas, é o código da [linguagem de descrição de hardware](https://pt.wikipedia.org/wiki/Linguagem_de_descri%C3%A7%C3%A3o_de_hardware) que normalmente se divide. Suas descrições popularmente são usadas para configurar um [system-on-a-chip](https://pt.wikipedia.org/wiki/System-on-a-chip) ou através de [FPGA](https://pt.wikipedia.org/wiki/FPGA) ou diretamente via [ASIC](https://pt.wikipedia.org/wiki/ASIC). Estes módulos, quando distribuídos, são chamados de núcleos semicondutores de propriedade intelectual (Em inglês, [Semiconductor intellectual property core](https://en.wikipedia.org/wiki/Semiconductor_intellectual_property_core" \o "en:Semiconductor intellectual property core) - IP cores) ou Núcleos IP.

Extensivas discussões foram feitas, a partir de distintas áreas da [informática](https://pt.wikipedia.org/wiki/Inform%C3%A1tica) e [eletrônica](https://pt.wikipedia.org/wiki/Eletr%C3%B4nica), para que o hardware livre fosse tão acessível quanto o [software livre](https://pt.wikipedia.org/wiki/Software_livre), tanto em nível de custo quanto a nível de utilização técnica.[[10]](https://pt.wikipedia.org/wiki/Hardware_livre#cite_note-10)[[11]](https://pt.wikipedia.org/wiki/Hardware_livre#cite_note-11). Também foram levantados tópicos a respeito de maneiras de colaborar em seu desenvolvimento, para mantê-lo sustentável.[[12]](https://pt.wikipedia.org/wiki/Hardware_livre#cite_note-12)

Uma das maiores diferenças entre os dois tipos de desenvolvimento, físico e lógico, é que os resultados do hardware são tangíveis, onde há custos para o protótipo e fabricação. Através disso, criaram a frase "Liberdade de expressão, não uma cerveja grátis"[[13]](https://pt.wikipedia.org/wiki/Hardware_livre#cite_note-13)[[14]](https://pt.wikipedia.org/wiki/Hardware_livre#cite_note-14), mais normalmente conhecida como [Gratis versus libre](https://pt.wikipedia.org/w/index.php?title=Gratis_versus_libre&action=edit&redlink=1" \o "Gratis versus libre (página não existe)), distinguindo a diferença entre custo zero e a facilidade para usar e modificar uma informação. Embora ele enfrente dificuldades em relação a reduzir os riscos financeiros para projetos de desenvolvedores individuais, alguns membros das comunidades propõem modelos para atender essas necessidades.[[15]](https://pt.wikipedia.org/wiki/Hardware_livre#cite_note-15) Dado a isto, existem iniciativas para desenvolver comunidades sustentáveis e acessíveis para a maioria dos usuários, tais como o [Open Source Hardware Central Bank](https://pt.wikipedia.org/w/index.php?title=Open_Source_Hardware_Central_Bank&action=edit&redlink=1), assim como ferramentas como o [KiCAD](https://pt.wikipedia.org/wiki/KiCAD" \o "KiCAD) para criar esquemáticas mais acessíveis para os usuários.

O hardware aberto podem ter duas vertentes: plataforma e arquitetura. Uma plataforma aberta é aquela que outros fabricantes podem replicá-la e até modificá-la (desde que a compatibilidade com a original seja garantida). Como exemplo de plataforma aberta, temos a plataforma IBM-PC. Atualmente também temos outras como Arduíno.  
Uma plataforma fechada é aquela onde somente um fabricante detém a produção e a venda daquele hardware, como mainframes IBM, supercomputadores CRAY, etc.  
Uma arquitetura aberta envolve design de componentes que podem ser fabricados por outros fabricantes. ARM é o melhor exemplo disso. Vários fabricantes produzem processadores ARM, como Samsung, qualcomm, nVidia (com seu tegra), etc.  
Já uma arquitetura fechada, somente um fabricante detém sua produção, com especial atenção ao x86/x64, que tem AMD e INTEL compartilhando a arquitetura (mas seus processadores são completamente diferentes apesar de serem compatíveis em nível de instruções básicas). Por exemplo temos a IBM com sua arquitetura POWER, a Sun/Oracle com sua arquitetura Sparc, a Intel com a Itanium.

Hardware proprietário é o hardware de computador cuja interface é controlada pelo proprietário, geralmente sob proteção de patente ou segredo comercial.

Historicamente, a maioria dos primeiros hardwares de computador foi projetada como proprietária até a década de 1980, quando o IBM PC mudou esse paradigma. No início da década de 1970, muitos fornecedores tentaram desafiar o monopólio da IBM no mercado de computadores de grande porte, fazendo engenharia reversa e produzindo componentes de hardware eletricamente compatíveis com equipamentos caros e (geralmente) capazes de executar o mesmo software. Esses fornecedores foram apelidados de fabricantes compatíveis com plugues (PCMs).