



**INSTITUTO FEDERAL**  
Goiano

Campus  
Ceres

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA

GOIANO - CAMPUS CERES

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

PROGRAMAÇÃO DE DISPOSITIVOS MÓVEIS

**LICENÇAS OPEN SOURCE**

CERES

2022

## SUMÁRIO

<b>Licenças Open Source</b>	<b>3</b>
<b>O que é uma licença MIT?</b>	<b>3</b>
<b>O que é a Licença Apache 2.0?</b>	<b>4</b>
<b>O que é a Licença GPL?</b>	<b>5</b>
<b>Conclusão</b>	<b>6</b>

## Licenças Open Source

Grande parte das licenças de código aberto possuem como base os mesmos princípios, sendo eles a garantia de que o software pode ser modificado, usado comercialmente e ser distribuído de forma privada ou não, portanto os autores ficam isentos de prover garantias quanto ao software. Algumas regras e normas também são compartilhados entre as licenças, visto que a licença e os direitos devem ser incluídos no software e que a OSI (Iniciativa pelo código aberto) deve aprovar a concessão.

## O que é uma licença MIT?

A licença MIT representa um acordo comumente usado por desenvolvedores de software sempre que eles lançam software livre e de código aberto. Este contrato é mais precisamente chamado de **Licença Expat**, pois ajuda a diferenciá-lo de outras licenças.

Isso significa que, quando um autor disponibiliza um software sob a licença MIT, ele está concedendo permissão aos desenvolvedores para usar, compartilhar ou modificar o software sem nenhum custo extra. No entanto, isso está sujeito a duas condições:

1. Os desenvolvedores concordam em não culpar o autor se o software não funcionar conforme o esperado.
2. O autor ainda possuirá os direitos autorais de seu trabalho, portanto, o desenvolvedor não poderá remover a licença. Se o desenvolvedor modificar o software de alguma forma, ele precisa ser claro sobre as mudanças que foram feitas e quais mudanças essas licenças estão distribuindo.

Pode acontecer que apenas parte de um pacote de software seja licenciado pelo MIT. Por exemplo, parte do software pode incluir gráficos sob uma licença diferente ou pode vir com algumas adições não gratuitas. É tarefa do autor tornar o escopo da licença o mais claro possível. O código de licença do MIT pode ser usado comercialmente e em software de origem em nuvem, embora o aviso de direitos autorais e o aviso de permissão precisem ser incluídos na documentação do software.

É importante destacar que o código da licença MIT não pode ser redistribuído sob outra licença, pois o código deve permanecer sob a licença MIT, mas o desenvolvedor pode distribuí-lo como parte de outro projeto sob qualquer licença desejada. Nesse caso, quaisquer alterações feitas no código são liberadas sob a nova licença. Por fim, a licença MIT. Dentre todas as que listamos, ela é a mais permissiva de todas, além de ser a mais famosa. Comparada às demais, ela oferece pouquíssima proteção aos autores do software. Um grande projeto com essa licença é o próprio Training Center!, jQuery, Bootstrap e o Rails.

## Pré-requisitos

Existem algumas instruções a serem consideradas ao usar a licença MIT nos projetos de software, aqui está um resumo dos aspectos mais importantes a serem considerados:

- Quando o desenvolvedor libera o código-fonte, ele deve colocar uma cópia do texto da licença MIT no topo de cada arquivo-fonte como um comentário.
- Quando o desenvolvedor lança um pacote de software, ele deve incluir uma cópia da licença MIT no diretório raiz do pacote. O arquivo precisa ser nomeado "COPYING" ou "LICENSE".
- O desenvolvedor deve exibir a licença MIT como parte do contrato de licença de usuário final (EULA) de seu software.
- O desenvolvedor deve exibir a licença MIT em qualquer documentação.
- Quando o desenvolvedor exibe a licença MIT, ele também deve exibir todas as informações de direitos autorais e, se apropriado, deixar claro a que a licença se aplica.

## **O que é a Licença Apache 2.0?**

A licença de software Apache dá aos usuários permissão para reutilizar o código para praticamente qualquer finalidade, incluindo o uso do código como parte de um software proprietário. Assim como outras licenças de código aberto, a Apache rege como os usuários finais podem utilizar o software em seus próprios projetos.

Portanto é uma licença de código aberto amplamente utilizada e, como outras licenças permissivas, continua a crescer em popularidade porque incentiva o uso de software de código aberto em projetos proprietários. Isso significa que, quando um autor disponibiliza um software sob a licença Apache 2.0, ele está concedendo permissão aos desenvolvedores para qualquer finalidade, incluindo o uso do código como parte de um software proprietário. No entanto, isso está sujeito a algumas condições:

1. Concessão de Licença de Direitos Autorais.
2. Concessão de Licença de Patente.
3. Redistribuição.
4. Envio de Contribuições.
5. Marcas Registradas.
6. Isenção de Garantia.
7. Limitação de responsabilidade
8. Aceitação de Garantia ou Responsabilidade Adicional

Apache é uma licença que oferece bem mais flexibilidade aos seus usuários e os entre eles os mais famosos são Android, Apache e o Swift.

## **Pré-requisitos**

Para aplicar a Licença Apache 2.0 a qualquer projeto, o desenvolvedor deve incluir uma cópia da Licença Apache, geralmente em um arquivo chamado "LICENSE",

junto com um arquivo "NOTICE" que faz referência à licença. O código-fonte não precisa ser necessariamente público quando a distribuição do software é feita. Modificações podem ser lançadas sob qualquer licença. Mudanças feitas no código-fonte precisam ser documentadas. É proibido explicitamente o uso de nomes de marcas registradas encontradas no projeto.

## **O que é a Licença GPL?**

A GNU General Public License (doravante referida como 'a GPL') foi criada pelo programador de computadores Richard Stallman como uma reação rebelde ao enorme crescimento do software proprietário. Ela foi nomeada a 'GNU' GPL, pois foi inicialmente usada por Stallman para licenciar o sistema operacional GNU.

A GPL é uma licença de software livre. Ao contrário do entendimento popular, o uso da palavra "livre" dentro de "software livre" não se refere ao preço, mas (por Stallman's definition) à liberdade do usuário de utilizar, modificar e distribuir o software (e quaisquer obras derivadas). Alguns projetos famosos com esta licença são o Bash, GIMP e o Linux (GPLv2).

Em termos gerais, a GPL baseia-se em 4 liberdades:

- A liberdade de executar o programa, para qualquer propósito.
- A liberdade de estudar como o programa funciona e adaptá-lo às suas necessidades. O acesso ao código-fonte é um pré-requisito para esta liberdade.
- A liberdade de redistribuir cópias de modo que você possa ajudar ao seu próximo.
- A liberdade de aperfeiçoar o programa e liberar os seus aperfeiçoamentos, de modo que toda a comunidade beneficie deles. O acesso ao código-fonte é um pré-requisito para esta liberdade.

Com a garantia destas liberdades, a GPL permite que os programas sejam distribuídos e reaproveitados, mantendo, porém, os direitos do autor por forma a não permitir que essa informação seja usada de uma maneira que limite as liberdades originais. A licença não permite, por exemplo, que o código seja apropriado por outra pessoa, ou que sejam impostos sobre ele restrições que impeçam que seja distribuído da mesma maneira que foi adquirido.

## **Pré-requisitos**

Apenas colocar uma cópia da GNU GPL em um arquivo em seu repositório não declara explicitamente que o código no mesmo repositório pode ser usado sob a GNU GPL. Sem essa declaração, não fica totalmente claro se as permissões na licença realmente se aplicam a qualquer arquivo de origem específico. Uma declaração explícita dizendo que elimina todas as dúvidas.

Deve-se colocar um aviso no início de cada arquivo de origem, informando qual licença ele carrega, a fim de evitar o risco de o código ser desconectado de sua

licença. Se o README do seu repositório diz que o arquivo de origem está sob a GNU GPL.

## Conclusão

Antes de falarmos de decidir qual licença usar é necessário entender as principais diferenças entre licenças de código aberto:

- Software que usa uma licença permissiva (permissive) coloca restrições mínimas sobre como o software pode ser distribuído.
- Bibliotecas que usam uma licença fracamente (weak) protetora desejam permanecer open source (para os seus clientes), mas permitir a integração com o software proprietário.
- Componentes que usam uma licença fortemente (strong) protetora são obrigados a manter o tipo da licença quando alterações ou adições são feitas.

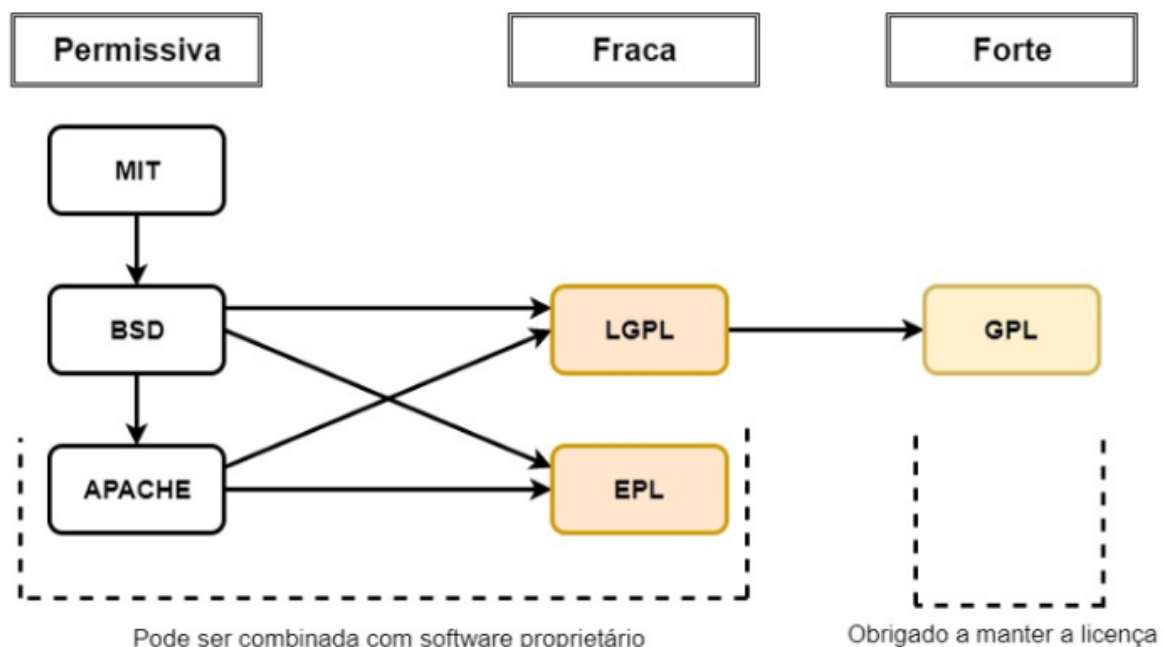


Diagrama das principais licenças open source. Fonte: Site Medium.

Portanto essas diferenças de licenças permissivas, fracas e fortes ajuda a compreender de onde o software fechado pode ser integrado com o software open source. Estas restrições são todas expressas em termos de distribuição de software para o seu cliente, não necessariamente para o público em geral. Vale destacar que estas licenças não impõem nenhuma obrigação em contribuir com as modificações para a comunidade. É importante durante a escolha de qual licença usar, analisar quais as finalidades do software requerente e comparar as obrigações e proteções de cada uma separadamente, procurando assim um *match* que consiga suprir todas as necessidades requisitadas.

**NOTA:** Para mais informação detalhada e completa sobre a GLP acessar:  
<https://www.gnu.org/licenses/gpl-faq.html#LicenseCopyOnly>

**NOTA:** Você pode navegar e verificar e analisar cada tópico acima no site  
<https://www.apache.org/licenses/LICENSE-2.0>.