# Construindo Produtos com o FreeBSD

Resumo

O projeto FreeBSD é um projeto voluntário e colaborativo de âmbito mundial, o qual desenvolve um sistema operacional de alta qualidade, capaz de ser utilizado em diferentes arquiteturas computacionais. O projeto FreeBSD distribui o código fonte do seu produto sob uma licença liberal, com a intenção de incentivar o uso de seu código. Colaborar com o projeto FreeBSD pode ajudar sua empresa a reduzir o tempo necessário para colocar um produto no mercado, a reduzir seus custos de engenharia e a melhorar qualidade de seus produtos.

Este artigo analisa as questões envolvidas no uso do código do FreeBSD em appliances e softwares. Ele também destaca as características do FreeBSD, que o tornam uma excelente base para o desenvolvimento de produtos. O artigo conclui sugerindo um conjunto das "melhores práticas" de organizações que colaboram com o projeto FreeBSD.

---

# Índice

1. Introdução . . . . . 1
2. O FreeBSD como um conjunto de blocos de construção . . . . . 3
3. Colaborando com o FreeBSD . . . . . 6
4. Conclusão . . . . . 10
Bibliografia . . . . . 10

# 1. Introdução

Atualmente o FreeBSD é bem conhecido como um sistema operacional de alto desempenho para servidores. Ele está instalado em milhões de servidores web e em outros hosts conectados diretamente a internet em todo o mundo. O código do FreeBSD também é parte integrante de muitos produtos, que vão desde aparelhos como roteadores de rede, firewalls e dispositivos de armazenamento, até computadores pessoais. Partes do FreeBSD também têm sido utilizadas em softwares comerciais (consulte O FreeBSD como um conjunto de blocos de construção).

Neste artigo, vamos olhar para o projeto FreeBSD como um recurso de engenharia de software, como um conjunto de blocos de construção e de processos, os quais você pode utilizar para construir produtos.

Embora o código fonte do FreeBSD seja distribuído gratuitamente ao público, para desfrutar plenamente dos benefícios do trabalho do projeto, as organizações precisam *colaborar* com o mesmo. Nas seções subsequentes do presente artigo discutiremos formas eficazes de colaborar com o projeto, bem como os perigos que precisam ser evitados ao fazê-lo.

**Aviso ao Leitor.** O autor considera que as características do projeto FreeBSD mencionadas neste artigo eram substancialmente verdadeiras no momento em que o artigo foi concebido e escrito (2005). No entanto, o leitor deve ter em mente que as práticas e processos utilizados por comunidades de código aberto podem mudar ao longo do tempo, e que portanto as informações deste artigo devem ser consideradas apenas como indicativas e não como verdades absolutas.

## 1.1. Público Alvo

Este documento tem como público alvo os seguintes grupos de pessoas:

- Tomadores de decisão em empresas que estejam em busca de meios para melhorar a qualidade de seus produtos, de reduzir o tempo necessário para lançá-los no mercado e de reduzir seus custos de engenharia no longo prazo.
- Consultores de tecnologia procurando melhores práticas em alavancar o código aberto.
- Observadores da indústria interessados em compreender a dinâmica dos projetos de código aberto.
- Desenvolvedores de software que utilizam o FreeBSD e que buscam formas de contribuir com o projeto.

## 1.2. Objetivos do artigo

Após a leitura deste artigo, você deve ter:

- Uma melhor compreensão dos objetivos do Projeto FreeBSD e de sua estrutura organizacional.
- Uma visão geral das tecnologias disponíveis no projeto.
- Uma melhor compreensão do modelo de desenvolvimento adotado pelo Projeto FreeBSD e dos processos de engenharia envolvidos no lançamento de uma nova versão do sistema.
- Consciência dos canais de comunicação utilizados pelo projeto e do nível de transparência que você pode esperar.
- Consciência das melhores formas de trabalhar com o projeto - a melhor forma de reduzir os custos de engenharia, de reduzir o tempo necessário para levar seu produto ao mercado, de gerir vulnerabilidades de segurança e preservar a compatibilidade futura com seu produto à medida que o projeto FreeBSD evolui.

## 1.3. Estrutura do Artigo

O restante deste artigo está estruturado da seguinte forma:

- O FreeBSD como um conjunto de blocos de construção introduz o projeto FreeBSD, explora a estrutura organizacional, tecnologias principais e processos de engenharia de releases.
- Colaborando com o FreeBSD descreve meios de colaboração para com o projeto FreeBSD. Examina armadilhas comuns encontrado por empresas trabalhando com projetos voluntários como o FreeBSD.
- Conclusão conclui.

# 2. O FreeBSD como um conjunto de blocos de construção

O FreeBSD fornece uma excelente base sobre a qual podemos construir produtos:

- O código fonte do FreeBSD é distribuído sob uma licença BSD liberal, o que facilita sua adoção em produtos comerciais [Mon2005] com um mínimo de preocupações.
- O Projeto FreeBSD possui excelentes práticas de engenharia as quais podem ser aproveitadas.
- O projeto oferece uma transparência excepcional em seu funcionamento, permitindo que as empresas que utilizam o seu código se planejem de forma eficaz para o futuro.
- A cultura do projeto FreeBSD, herdada do Grupo de Pesquisa de Ciências da Computação da Universidade da Califórnia em Berkeley [McKu1999-1], fomenta trabalhos de alta qualidade. Algumas funcionalidades do FreeBSD definem o estado da arte.

O [GoldGab2005] analisa em maior profundidade os motivos comerciais para se utilizar código fonte aberto. Para as organizações, os benefícios do uso de componentes do FreeBSD em seus produtos incluem a redução do tempo necessário para lançar novos produtos no mercado, menores custos e menores riscos de desenvolvimento.

## 2.1. Construindo com o FreeBSD

Aqui estão alguns exemplos de como as empresas estão utilizando o FreeBSD:

- Como um provedor (upstream source) de códigos testados para bibliotecas e utilitários.

  Por estar "downstream" do projeto, as organizações aproveitam os novos recursos, correções de bugs e testes que o código upstream recebe.

- Como sistema operacional integrado (por exemplo, em um roteador OEM e ou em um dispositivo de firewall). Neste modelo, as empresas utilizam uma versão customizada do kernel e do conjunto de aplicativos do FreeBSD, juntamente com uma camada proprietária de gestão para os seus dispositivos. Os fabricantes de equipamentos originais (OEMs) se beneficiam da adição por parte do FreeBSD de suporte a novos componentes de hardware, bem como se beneficia dos testes que o sistema base recebe.

  O FreeBSD é distribuído com um ambiente de desenvolvimento auto-hospedado o qual permite a fácil criação de tais configurações.

- Como um ambiente Unix compatível para as funções de gerenciamento em dispositivos de armazenamento high-end e em dispositivos de rede, executando em uma lâmina separada "blade".

  O FreeBSD fornece ferramentas para a criação de imagens do sistema operacional dedicadas a executar uma função específica. Sua implementação da API unix BSD é madura e testada. O FreeBSD também pode proporcionar um ambiente de desenvolvimento cruzado estável para os outros componentes de dispositivos topo de linha.

- Como um meio de obter testes abrangentes e suporte de uma equipe mundial de desenvolvedores para "propriedade intelectual" não crítica.

  Neste modelo, as organizações contribuem com estruturas infraestruturais úteis para o projeto FreeBSD (por exemplo, veja netgraph(3)). A ampla exposição que o código recebe ajuda a identificar rapidamente problemas de desempenho e bugs. O envolvimento de desenvolvedores de primeira linha também leva a extensões úteis à infraestrutura das quais a organização contribuinte também se beneficia.

- Como um ambiente de desenvolvimento que suporta desenvolvimento cruzado para sistemas operacionais embarcados, como RTEMS e eCOS.

  Há muitos ambientes de desenvolvimento completos na coleção de 37000 aplicativos portados e empacotados com o FreeBSD.

- Como forma de suportar uma API estilo Unix em um sistema operacional que de outro modo seria proprietário, aumentando a sua palatabilidade para os desenvolvedores de aplicativos.

  Aqui, partes do kernel do FreeBSD e dos programas de aplicação são "portadas" para serem executadas junto com outras tarefas no sistema operacional proprietário. A disponibilidade de uma implementação estável e bem testada da API Unix™ pode reduzir o esforço necessário para portar aplicativos populares para o sistema operacional proprietário. Como o FreeBSD é fornecido com documentação de alta qualidade para seus componentes internos e possui processos eficazes de gerenciamento de vulnerabilidades e engenharia de release, os custos para se manter atualizado são mantidos baixos.

## 2.2. Tecnologias

Existe um grande número de tecnologias suportadas pelo projeto FreeBSD. Abaixo você encontra uma lista com alguma delas:

- Um sistema completo que pode hospedar-se cruzadamente para muitas arquiteturas:
- Um kernel modular capaz de multiprocessamento simétrico, com módulos de kernel carregáveis e um sistema de configuração flexível e fácil de usar.
- Suporte para emulação de binários Linux™ e SVR4 em velocidades próximas às da máquina. Suporte para drivers de rede binários do Windows™ (NDIS).
- Bibliotecas para muitas tarefas de programação: arquivamento, suporte a FTP e HTTP, suporte a threads, além de um ambiente de programação completo semelhante ao POSIX™.
- Recursos de segurança: Controle de Acesso Obrigatório (mac(9)), jails (jail(2)), ACLs e suporte a dispositivos criptográficos no kernel.
- Funcionalidades avançadas de rede: firewalls, gerenciamento de QoS, rede TCP/IP de alta performance com suporte a muitos recursos avançados.

  O framework Netgraph (netgraph(4)) no kernel do FreeBSD permite que módulos de rede do kernel sejam conectados de maneiras flexíveis.

- Suporte para tecnologias de armazenamento: Fibre Channel, SCSI, RAID de software e

hardware, ATA e SATA.

O FreeBSD suporta vários sistemas de arquivos, e seu sistema de arquivos nativo UFS2 suporta "soft updates", snapshots e tamanhos de sistema de arquivos muito grandes (16 TB por sistema de arquivos) [McKu1999].

O framework GEOM (geom(4)) no kernel do FreeBSD permite que módulos de armazenamento do kernel sejam compostos de maneiras flexíveis.

- Mais de 37000 aplicativos portados, tanto comerciais quanto de código aberto, gerenciados por meio da coleção de ports do FreeBSD.

## 2.3. Estrutura Organizacional

A estrutura organizacional do FreeBSD não é hierárquica.

Existem basicamente dois tipos de contribuidores para o FreeBSD: usuários gerais do FreeBSD e desenvolvedores com acesso de escrita (conhecidos como *committers* na gíria) à base de código-fonte.

Existem muitos milhares de colaboradores no primeiro grupo, a grande maioria das contribuições para o FreeBSD vêm de indivíduos desse grupo; A permissão de commit (acesso de escrita) no repositório é concedida a pessoas que contribuem de forma consistente para o projeto. O direito de commit vem acompanhado de responsabilidades adicionais, e para facilitar o aprendizado das mesmas, um mentor é atribuído a todos os novos committers.

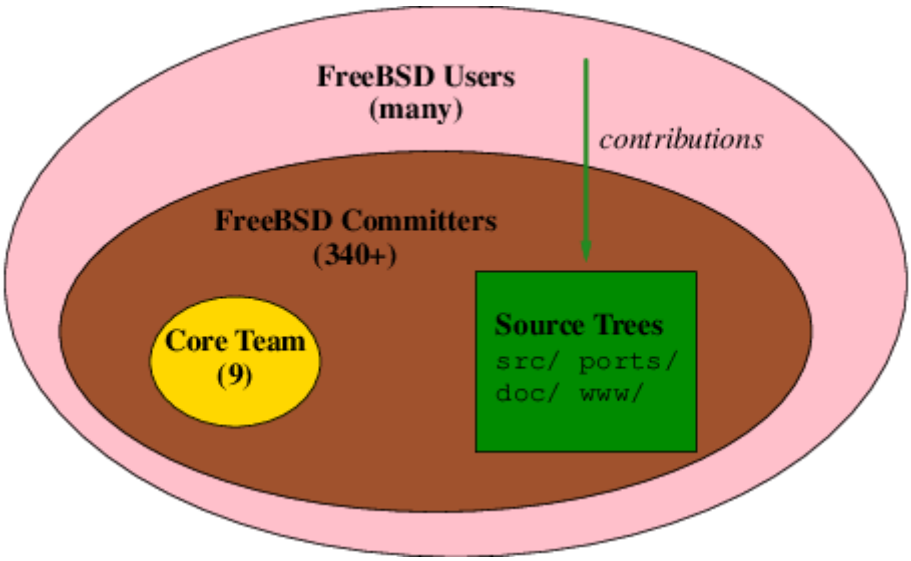



*Figura 1. Organização do FreeBSD*

A resolução de conflitos é realizada por uma equipe central ("Core Team") de nove membros que é eleita a partir do grupo de committers.

O FreeBSD não tem committers "corporativos". Cada committer individual é responsável pelas alterações que introduz no código. O guia do committer do FreeBSD (FreeBSD Committer's guide) [ComGuide] documenta as regras e responsabilidades dos committers.

O modelo de projeto do FreeBSD é examinado em detalhes em [Nik2005].

## 2.4. Processos de Engenharia para liberação de novas versões do FreeBSD

Os processos de engenharia de release do FreeBSD desempenham um papel importante na garantia de que as versões lançadas sejam de alta qualidade. Em qualquer momento, os voluntários do FreeBSD suportam várias linhas de código (FreeBSD Release Branches):

- Novos recursos e código disruptivo entram na branch de desenvolvimento, também conhecida como branch *-CURRENT*.
- As branch *-STABLE* são linhas de código que são ramificadas do HEAD em intervalos regulares. Somente o código testado é permitido em uma branch *-STABLE*. Novos recursos são permitidos apenas depois de terem sido testados e estabilizados na branch *-CURRENT*.
- As branchs *-RELEASE* são mantidas pela equipe de segurança do FreeBSD. Somente correções de bugs para problemas críticos são permitidas nas branchs *-RELEASE*.

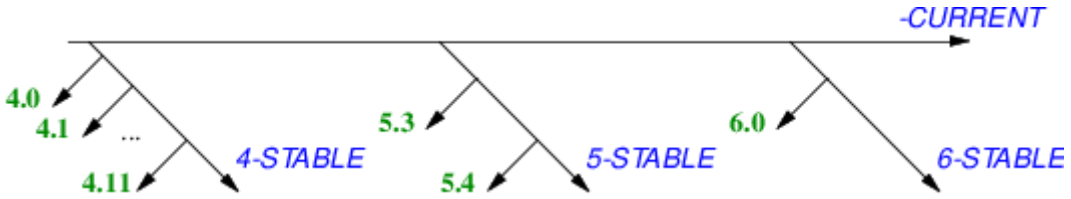



*Figura 2. Release Branches do FreeBSD*

As linhas de código são mantidas vivas enquanto houver interesse dos usuários e dos desenvolvedores nelas.

As arquiteturas de máquinas são agrupadas em "níveis"; as arquiteturas *Tier 1* são totalmente suportadas pelas equipes de engenharia de release e segurança do projeto, as arquiteturas *Tier 2* são suportadas com base no melhor esforço, e as arquiteturas experimentais compõem o *Tier 3*. A lista de arquiteturas suportadas faz parte da coleção de documentação do FreeBSD.

A equipe de engenharia de release publica um road map para futuros lançamentos do FreeBSD no site do projeto. As datas estabelecidas no roteiro não são prazos finais; o FreeBSD é lançado quando seu código e documentação estão prontos.

Os processos de engenharia de release do FreeBSD são descritos em [RelEngDoc].

# 3. Colaborando com o FreeBSD

Projetos open-source como o FreeBSD oferecem código final de alta qualidade.

Embora o acesso a um código fonte de qualidade possa reduzir o custo inicial de desenvolvimento, a longo prazo, os custos com o gerenciamento de mudanças começam a dominar. A medida que os ambientes computacionais mudam ao longo dos anos e novas vulnerabilidades de segurança são descobertas, o seu produto também precisará mudar e se adaptar. O uso de código open-source não deve ser encarado como uma atividade pontual, mas sim como um *processo contínuo*. Os melhores projetos para se colaborar são os que estão *vivos*, ou seja, aqueles com uma comunidade ativa, que tenha objetivos claros e que possua um estilo de trabalho transparente.

- O FreeBSD tem uma comunidade de desenvolvimento ativa em torno dele. No momento em que este artigo foi escrito, existiam milhares de colaboradores com representantes de praticamente todos os continentes povoados do mundo, e mais de 300 indivíduos com acesso de escrita aos repositórios do projeto.
- Os objetivos do projeto FreeBSD são [Hub1994]:
  - Desenvolver um sistema operacional de alta qualidade para o hardware de computadores populares, e,
  - Tornar o nosso trabalho disponível para todos sob uma licença liberal.
- O FreeBSD possui uma cultura de trabalho aberta e transparente. Quase todas as discussões no projeto ocorrem por e-mail, em listas de discussão públicas que também são arquivadas para a posteridade. As políticas do projeto são documentadas e mantidas sob controle de revisão. A participação no projeto está aberta a todos.

# 3.1. Compreendendo a cultura do FreeBSD

Para ser capaz de trabalhar de forma eficaz com o projeto FreeBSD, você precisa entender a cultura do projeto.

As regras que regem a operação de um projeto voluntário são diferentes das que regem a operação de uma empresa com fins lucrativos. Um erro comum que as empresas cometem ao se aventurar no mundo open-source é o de subestimar essas diferenças.

**Motivação.** A maioria das contribuições para o FreeBSD é feita voluntariamente, sem a entrada de recompensas monetárias. Os fatores que motivam os indivíduos são complexos, variando desde o altruísmo até o interesse em resolver os tipos de problemas que o FreeBSD tenta resolver. Nesse ambiente, "a elegância nunca é opcional" [Nor1993].

**A Visão de Longo Prazo.** O FreeBSD remonta quase vinte anos ao trabalho do Computer Science Research Group da Universidade da Califórnia em Berkeley. [1] Vários dos desenvolvedores originais do CSRG ainda estão associados ao projeto.

O projeto valoriza perspectivas de longo prazo [Nor2001]. Um acrônimo frequente encontrado no projeto é DTRT, que significa "Faça a Coisa Certa".

Os programas de computador são ferramentas de comunicação: em um nível, os programadores comunicam suas intenções usando uma notação precisa para uma ferramenta (um compilador) que traduz suas instruções para código executável. Em outro nível, a mesma notação é usada para comunicação de intenção entre dois programadores.

Especificações formais e documentos de design raramente são utilizados no projeto. Em seu lugar, é usado um código claro e bem escrito e registros de alterações bem escritos (Um exemplo de entrada de registro de alterações). O desenvolvimento do FreeBSD acontece por meio de "consenso geral e código em execução" (tradução livre de "rough consensus and running code"[Carp1996]).

```
r151864 | bde | 2005-10-29 09:34:50 -0700 (Sat, 29 Oct 2005) | 13 lines
Changed paths:
   M /head/lib/msun/src/e_rem_pio2f.c
```

```
Use double precision to simplify and optimize arg reduction for small
and medium size args too: instead of conditionally subtracting a float
17+24, 17+17+24 or 17+17+17+24 bit approximation to pi/2, always
subtract a double 33+53 bit one.  The float version is now closer to
the double version than to old versions of itself -- it uses the same
33+53 bit approximation as the simplest cases in the double version,
and where the float version had to switch to the slow general case at
|x| == 2^7*pi/2, it now switches at |x| == 2^19*pi/2 the same as the
double version.

This speeds up arg reduction by a factor of 2 for |x| between 3*pi/4 and
2^7*pi/4, and by a factor of 7 for |x| between 2^7*pi/4 and 2^19*pi/4.
```

*Um exemplo de entrada de log de mudanças*

Comunicação entre programadores é melhorada pelo uso de um padrão de codificação comum (style(9)).

**Canais de Comunicação.** Os colaboradores do FreeBSD estão espalhados pelo mundo. O e-mail (e, em menor medida, o IRC) é o meio de comunicação preferido no projeto.

# 3.2. Melhores práticas para colaborar com o projeto FreeBSD

Agora iremos examinar algumas das melhores práticas para se fazer um melhor uso do FreeBSD no desenvolvimento de produtos.

**Se planeje para o longo prazo**

Criar processos que auxiliem no acompanhamento do desenvolvimento do FreeBSD. Por exemplo:

**Acompanhar o código-fonte do FreeBSD.** O projeto facilita a criação de um espelho do seu repositório SVN usando o svnsync. Ter todo o histórico do código-fonte é útil para depurar problemas complexos e oferece informações valiosas sobre as intenções dos desenvolvedores originais. Use um sistema de controle de código capaz que permita mesclar facilmente as alterações entre o código-base upstream do FreeBSD e o seu próprio código interno.

Um arquivo de origem anotado gerado usando `svn blame` mostra uma parte de um arquivo de listagem anotado do arquivo referenciado pelo registro de alterações em Um exemplo de registro de alterações. A ancestralidade de cada linha do código-fonte é claramente visível. Listagens anotadas mostrando a história de cada arquivo que faz parte do FreeBSD estão disponíveis no website.

```
#REV         #WHO #DATE                                      #TEXT

176410        bde 2008-02-19 07:42:46 -0800 (Tue, 19 Feb 2008) #include
<sys/cdefs.h>
176410        bde 2008-02-19 07:42:46 -0800 (Tue, 19 Feb 2008)
```

```
__FBSDID("$FreeBSD$");
  2116        jkh 1994-08-19 02:40:01 -0700 (Fri, 19 Aug 1994)
  2116        jkh 1994-08-19 02:40:01 -0700 (Fri, 19 Aug 1994) /*
__ieee754_rem_pio2f(x,y)
  8870    rgrimes 1995-05-29 22:51:47 -0700 (Mon, 29 May 1995)  *
176552        bde 2008-02-25 05:33:20 -0800 (Mon, 25 Feb 2008)  * return the
remainder of x rem pi/2 in *y
176552        bde 2008-02-25 05:33:20 -0800 (Mon, 25 Feb 2008)  * use double
precision for everything except passing x
152535        bde 2005-11-16 18:20:04 -0800 (Wed, 16 Nov 2005)  * use
__kernel_rem_pio2() for large x
  2116        jkh 1994-08-19 02:40:01 -0700 (Fri, 19 Aug 1994)  */
  2116        jkh 1994-08-19 02:40:01 -0700 (Fri, 19 Aug 1994)
176465        bde 2008-02-22 07:55:14 -0800 (Fri, 22 Feb 2008) #include <float.h>
176465        bde 2008-02-22 07:55:14 -0800 (Fri, 22 Feb 2008)
  2116        jkh 1994-08-19 02:40:01 -0700 (Fri, 19 Aug 1994) #include "math.h"
```

*Uma listagem de origem anotada gerada usando* `svn blame`

**Use um gatekeeper.** Nomeie um *gatekeeper* para monitorar o desenvolvimento do FreeBSD, para ficar de olho em mudanças que possam potencialmente impactar seus produtos.

**Relate bugs upstream.** Se você notar algum bug no código do FreeBSD que está usando, envie um relatório de bug. Esse passo ajuda a garantir que você não precise corrigir o bug na próxima vez que fizer uma atualização de código do upstream.

**Se alavanque nos esforços de engenharia do FreeBSD para lançamento de novas versões**

Utilize código da branch de desenvolvimento -STABLE do FreeBSD. Este branch de desenvolvimento é formalmente suportado pelas equipes de engenharia de release e de segurança, e é formada apenas por código testado.

**Doe código para reduzir seus custos**

Uma parte significativa dos custos relacionados ao desenvolvimento de um produto é o de realizar a sua manutenção. Ao doar partes não criticas do seu código para o projeto, você se beneficia por ter o seu código exposto de uma forma ampla, exposição que ele não teria de outra forma. Esta exposição por sua vez leva eliminação de um maior numero de bugs e de vulnerabilidades de segurança, e permite que anomalias de desempenho sejam identificadas e corrigidas.

**Obtenha suporte efetivo**

Para produtos com prazos apertados, é recomendado contratar ou entrar em um acordo de consultoria com um desenvolvedor ou empresa com experiência em FreeBSD. O lista de discussão sobre empregos relacionados ao FreeBSD é um canal de comunicação útil para encontrar talentos. O projeto FreeBSD mantém uma galeria de consultores e empresas de consultoria que realizam trabalhos em FreeBSD. O Grupo de Certificação BSD oferece certificação para todos os principais sistemas operacionais derivados do BSD.

Para necessidades menos críticas, você pode pedir ajuda nas listas de discussão do projeto. Um guia útil a seguir ao pedir ajuda é dado em [Ray2004].

**Divulgue o seu envolvimento**

Você não é obrigado a divulgar que faz uso do FreeBSD, mas ao fazê-lo você estará ajudando ambos os esforços, o seu e o do projeto.

Dar visibilidade para a comunidade FreeBSD de que a sua empresa utiliza o sistema ajuda a melhorar as suas chances de atrair talentos de alta qualidade. Quanto maior for a lista de organizações que apoiam o FreeBSD maior será a presença do sistema na cabeça (mind share) dos desenvolvedores. Ao contribuir para aumentar o numero de desenvolvedores interessados no FreeBSD, você estará gerando uma base saudável para o seu futuro.

**Suporte os desenvolvedores do FreeBSD**

Às vezes, a maneira mais direta de obter um recurso desejado no FreeBSD é apoiar um desenvolvedor que já está trabalhando em um problema relacionado. A ajuda pode variar desde doações de hardware até assistência financeira direta. Em alguns países, as doações ao projeto FreeBSD são isentas de impostos. O projeto possui uma área de doações dedicada para ajudar doadores. O projeto também mantém uma página da web onde os desenvolvedores listam suas necessidades.

Como política, o projeto FreeBSD reconhece todas as contribuições recebidas em seu site.

# 4. Conclusão

O Objetivo do projeto FreeBSD é criar e distribuir o código fonte de um sistema operacional de alta qualidade. Ao trabalhar com o projeto FreeBSD você pode reduzir os seus custos de desenvolvimento e melhorar o tempo necessário para lançar seus novos produtos no mercado em vários cenários de desenvolvimento de produtos.

Nós examinamos as características do projeto FreeBSD que o tornam uma excelente escolha para fazer parte da estratégia de produtos de uma organização. Em seguida, examinamos a cultura predominante do projeto e examinamos maneiras eficazes de interagir com seus desenvolvedores. O artigo concluiu com uma lista de melhores práticas que podem ajudar as organizações a colaborar com o projeto.

# Bibliografia

[Carp1996] Os Princípios Arquitetônicos da Internet. The Internet Architecture Board. Copyright® 1996.

[ComGuide] Guia do Comitter O Projeto FreeBSD. Direitos Autorais® 2005.

[GoldGab2005] Inovação Acontece em Outros Lugares: Código Aberto como Estratégia de Negócios Ron Goldman. Richard Gabriel. Copyright® 2005. Morgan-Kaufmann.

[Hub1994] Contribuindo para o Projeto FreeBSD Jordan Hubbard. Copyright® 1994-2005. The FreeBSD Project.

[McKu1999] Soft Updates: Uma técnica para eliminar a maioria das gravações síncronas no sistema de arquivos rápido Kirk McKusick. Gregory Ganger. Copyright® 1999.

[McKu1999-1] Vinte anos de Unix Berkeley: de propriedade da AT&T à redistribuição livre Marshall Kirk McKusick. Open Sources: Vozes da Revolução Open Source O'Reilly Inc.. Copyright® 1993.

[Mon2005] Por que você deveria usar uma licença de estilo BSD para o seu projeto de código aberto. Bruce Montague. The FreeBSD Project. Copyright® 2005.

[Nik2005] Um modelo de projeto para o Projeto FreeBSD Niklas Saers. Copyright® 2005. The FreeBSD Project.

[Nor1993] Tutorial sobre Bom Estilo de Programação em Lisp. Peter Norvig. Kent Pitman. Copyright® 1993.

[Nor2001] Ensine a si mesmo a programar em dez anos. Peter Norvig. Copyright® 2001.

[Ray2004] Como fazer perguntas de forma inteligente Eric Steven Raymond. Copyright® 2004.

[RelEngDoc] Engenharia de Releases do FreeBSD Murray Stokely. Copyright® 2001. The FreeBSD Project.

[1] O repositório de código-fonte do FreeBSD contém um histórico do projeto desde o seu início, e há CDs disponíveis que contêm código anterior do CSRG.