

```
#####  
#  
# Fácil MVC - Arquivo de filtro geral do framework  
#  
# Opções padrão:  
#  
# -> Indexes gera um arquivo com a lista do conteúdo  
# de um diretório sempre que a URL chamada apontar  
# pra um diretório que não tenha arquivo index  
# (como index.html ou index.php)  
#  
# -> - Indexes desliga essa opção e gera Forbidden nesse caso  
#  
# -> MultiViews em URLs que apontem para um diretório que não exista  
# faz o Apache procurar por arquivos com o mesmo nome e qualquer extensão  
# Assim: /caminho/pra/diretorio pode achar /caminho/pra/diretorio.php  
#  
# -> - MultiViews no nosso caso evita index (ação) ser considerado index.php  
# (index.php é o nosso arquivo Front-Controller)  
#  
# -> + FollowSymlinks informa ao servidor que percorra o links simbólicos,  
# Atalhos criados no sistema de arquivos para simplificar e reaproveitar  
# os endereçamentos de recursos,  
# Esta diretiva deve estar ligada uma vez que precisamos do mod_rewrite  
#
```

```
#####
```

```
options -Indexes -MultiViews +FollowSymlinks
```

```
#####  
#  
# Configurações Gerais PHP_INI_PER_DIR  
#  
# -> Desligando opções depreciadas,  
# -> Ligando short_open_tag para poder permitir <?= nas templates,  
# A partir do PHP 5.4+ as expressões <?= não precisam desta diretiva  
# para funcionarem, portanto você não precisa ligá-la  
# apenas por este motivo,  
# Leia mais: https://bugs.php.net/bug.php?id=50662  
# -> Restringindo uso de cookies a envios HTTP, indicando ao browser  
# que não tente obter cookies via GET ou JavaScript, etc...
```

```
#     que nao tente abrir esses cookies se algum javascript solicitar.
#     Isso previne certos tipos de ataques XSS.
#     Leia mais: http://php.net/function.setcookie.php#79488
#
#####

php_flag magic_quotes_gpc Off
php_flag magic_quotes_runtime Off
php_flag magic_quotes_sybase Off
php_flag register_globals Off
php_flag short_open_tag On
php_flag session.cookie_httponly On

#####
#
#     Ligando o ModRewrite para as principais regras de navegação
#
#####

<IfModule mod_rewrite.c>
    RewriteEngine On

    #####
    # Bloqueando acesso (Forbidden) a qualquer arquivo ou
    # diretório oculto. Iniciado pelo ponto
    #####
    #RewriteCond %{REQUEST_FILENAME} -d
    #RewriteCond %{REQUEST_FILENAME} -f
    RewriteRule "(^/)\." - [F,L]

    #####
    # Capturando chamadas a js, img, css e recursos
    # Ligar essa diretiva pode aumentar a carga no servidor uma vez
    # que toda requisição para arquivo estático vai passar pelo PHP.
    # Mas haverá a vantagem dele poder processar internacionalização
    # também nos arquivos .js e .css além de poder reduzir o tamanho
    # destes (minify) o que pode melhorar o desempenho se o gargalo
    # estiver no tráfego e não no processamento da máquina.
    #####
    # RewriteRule .*((js/)|(css/))(.*) index.php?_url=$0 [L]
```

```

#####
# Direcionando toda URL que não apontar para recurso existente para
# index.php passando o caminho para a variavel $url
#####
RewriteCond %(REQUEST_FILENAME) !-d
RewriteCond %(REQUEST_FILENAME) !-f
RewriteRule ^(.*)$ index.php?url=$1 [QSA,L]

</IfModule>

#####
#
# Compactação Gzip para diminuição de tráfego
#
#####

<IfModule mod_deflate.c>
  # Force deflate for mangled headers developer.yahoo.com/blogs/ydn/posts/2010/12/pushing-
  beyond-gzipping/
  <IfModule mod_setenvif.c>
    <IfModule mod_headers.c>
      SetEnvIfNoCase ^(\Accept-Encoding|X-cept-Encoding|X(15)|^(15)|-(15))$
      ^((gzip|deflate)\s*,\s*\s*\s*)+([X"]-){4,13}$ HAVE_Accept-Encoding
      RequestHeader append Accept-Encoding "gzip,deflate" env=HAVE_Accept-Encoding
    </IfModule>
  </IfModule>
</IfModule>

#####
#
# Removendo cabeçalho de controle de cache ETag
# Não funciona em servidores clusterizados e não traz nenhuma
# vantagem considerável com relação ao controle via
# Last-Modified. Portanto, retirá-lo reduz um cabelo de sapo
# o tamanho da resposta do servidor
#
#####

<IfModule mod_headers.c>

```

Header unset ETag

</IfModule>

FileETag None

```
#####  
#  
#   Redirecionando requisições que forneceram www para a opção sem www.  
#   Colocamos as duas técnicas conforme sugerido no HTML5 Boilerplate.  
#   No entanto não habilitamos nenhuma para que você possa escolher  
#   ou mesmo nunca ligar em situações onde a máquina www for diferente  
#   da máquina padrão de autoridade pelo domínio.  
#   O que motiva esse redirecionamento são as técnicas de SEO que  
#   consideram esse conteúdo duplicado (em duas URLs) algo ruim  
#  
#####  
  
# Opção 1 (a mais indicada):  
# Rewrite "www.example.com -> example.com"  
  
# <IfModule mod_rewrite.c>  
#   RewriteCond %{HTTPS} !=on  
#   RewriteCond %{HTTP_HOST} ^www\. (.+)$ [NC]  
#   RewriteRule ^ http://%1%{REQUEST_URI} [R=301,L]  
# </IfModule>  
  
# -----  
  
# Option 2 (só libere aqui se comentar a primeira opção):  
# Rewrite "example.com -> www.example.com"  
  
# <IfModule mod_rewrite.c>  
#   RewriteCond %{HTTPS} !=on  
#   RewriteCond %{HTTP_HOST} !^www\. .+$ [NC]  
#   RewriteRule ^ http://www.%{HTTP_HOST}%{REQUEST_URI} [R=301,L]  
# </IfModule>
```