

# Tecnologia

Trechos de códigos, comandos & outras chatices

- [Códigos](#)
  - [How to change the shared url on the ShareThis plugin](#)
  - [Xdebug na linha de comando](#)
  - [Mysql on duplicate key](#)
  - [Alterando a aparência do show processlist](#)
  - [Limite de conexões no mysql](#)
  - [Executando métodos de webservice SOAP usando cURL](#)
  - [Mysql - encontrando registros com um ou mais caracteres maiúsculos](#)
  - [Erro code 1271 - Illegal mix of collations for operation union](#)
  - [Importando vários arquivos SQL de uma vez só](#)
  - [Usando migrations no symfony para desfazer migração](#)
  - [Lendo arquivos de uma pasta com php](#)
  - [Jquery Ajax](#)
  - [Copiando para a área de transferência com Javascript](#)
  - [Avisar se o usuário esqueceu de gravar o formulário](#)
  - [Executando e acompanhando tarefas demoradas na linha de comando do MySQL](#)
  - [Buscar ignorando acentos no Mysql](#)
  - [Ordenando elementos HTML com o Bootstrap](#)
  - [Inserindo HTML em um iframe com jQuery](#)
  - [Máscara para telefones com 8 ou 9 dígitos](#)
  - [Exportando tabelas para excel e acertando o formato dos campos](#)
  - [Utilizando jquery-ui tabs + jquery validation plugin](#)
  - [Alguns comandos do MySQL](#)
  - [jQuery hasAttr](#)
  - [Validando datas em pt-br com jQuery Validate](#)
  - [Acessando conteúdo do CKEDITOR](#)

- [Contar dias ou horas entre datas em Mysql](#)
- [jQuery On](#)
- [D.R no Mysql](#)
- [Fazendo caso](#)
- [Executando um comando como Root via php](#)
- [Exportando um ou mais registros no MySql com mysqldump](#)
- [Lendo arquivos de uma pasta com php](#)
- [Utilizando plugin jquery validation em formulário com abas](#)
- [Executando comandos no client do MySQL sem informar a senha](#)
- [Desativando chaves estrangeiras no Mysql](#)
- [Git](#)

- [Linux](#)

- [Copiar e mesclar duas pastas no terminal](#)
- [dig dig dig dig](#)
- [Erro ao atualizar certificados no let's encrypt](#)
- [Comprimindo pastas](#)
- [Alterando dependências de um pacote deb](#)
- [Encontrando e substituindo em arquivos no shell](#)
- [Removendo um ip do Fail2ban](#)
- [Mini-curso mega-rápido de Torrents e Piratebay](#)
- [Operações com arquivos no shell](#)
- [Exibir qual processo está usando uma determinada porta](#)
- [Agendando desligamento no Debian](#)
- [Vídeos de Timelapse](#)
- [Criando e distribuindo um torrent usando o Transmission](#)
- [Convertendo de MP4 para GIF no Debian](#)
- [Montando drives usb na linha de comando](#)
- [Executando um comando repetidas vezes no terminal](#)
- [Acesso ssh via tor](#)
- [Removendo arquivos com interrogação \(?\) no nome](#)
- [Trabalhando com arquivos no terminal](#)
- [Virtualbox - faltando headers](#)
- [Mexendo com arquivos wbfes para usar no Nintendo Wii](#)

- [Gravando um arquivo ISO em um pendrive](#)
- [Erro NO\\_PUBKEY no Ubuntu/Debian](#)
- [Backup com rsync](#)
- [Listando portas X serviços](#)
- [Um pouco de windows](#)
- [Baixando arquivos recursivamente via FTP](#)
- [nohup](#)
- [Renomeando arquivos sequencialmente](#)
- [Mexendo com o Swap](#)
- [DDNS com a Digital Ocean](#)
- [Atualizando o Debian/Ubuntu](#)
- [Criando link simbólico para uma pasta](#)
- [Trocando o editor do crontab para o nano](#)
- [Configurando o Percona Monitoring and Management usando Docker](#)
- [Habilitando APM no log do Nginx](#)
- [Gerando chaves ssh](#)
- [Docker](#)
- [Usando ffmpeg](#)
- [Bloqueando ip no iptables](#)
- [Comprimir imagens no terminal](#)

# Códigos

Coisas que provavelmente vou precisar depois (ou não)

# How to change the shared url on the ShareThis plugin

Now that the AddThis plugin will be terminated, it's time to switch to yet another sharing platform.

The closest one that I found is ShareThis. Unfortunately, ShareThis does not have an API to change the shared info after the page has been loaded. So, we need to tinker with it a little.

What I needed was to change the url, change the description, and to open on a new page, instead of a small pop-up.

The first thing is that we need to add the data-url, data-title and data-description tags to the html element. Supposedly, this instructs the plugin to get the info from there, instead of searching somewhere else.

```
<div class="sharethis-inline-share-buttons" data-url="" data-title=""></div>
```

Next, include the js file:

```
<script type='text/javascript'  
src='https://platform-api.sharethis.com/js/sharethis.js#property=abc123456'  
async='async'></script>
```

Then, when we need to change the shared url, we must change the data attribute values from the div:

```
$('.sharethis-inline-share-buttons').attr({  
  'data-url' : someurl,  
  'data-title' : 'some title',  
  'data-description' : 'something else'  
});
```

Lastly, we need to reload the plugin with the new data:

```
function shareThis() {  
  if ( __sharethis__ && __sharethis__.config ) {  
    st.open = function (url) {  
      var h, w, wh, ww;  
      if (!url) {  
        return;  
      }  
    }  
  }  
}
```

```
}  
  if (st.mobile) {  
    return window.open(url, '_blank');  
  } else if (url.indexOf('mailto:') > -1 || url.indexOf('viber') > -1) {  
    return document.location = url;  
  } else {  
    return window.open(url, '_blank');  
  }  
};  
__sharethis__.init(__sharethis__.config);  
}  
}
```

This function must be called just after we change the data-attributes from the div element. I'm not sure why, but I found that I need to wait some time before calling the function. On my tests, I found that waiting 200ms is better.

Also, this function redefines the "open" method. On the original version, the popup is too little - it uses about 60% of the actual screen size. With the changes above, the pop-up opens on another page. I think that is way better.

# Xdebug na linha de comando

Primeiro, inicie a depuração normalmente na IDE

Depois, inclua os parâmetros abaixo no comando:

```
-dxdebug.mode=debug -dxdebug.client_host=127.0.0.1 -dxdebug.client_port=9003 -  
dxdebug.start_with_request=yes
```

Exemplo:

```
php -dxdebug.mode=debug -dxdebug.client_host=127.0.0.1 -dxdebug.client_port=9003 -  
dxdebug.start_with_request=yes script.php
```

Obs.: Se estiver usando o Netbeans, é importante configurar o projeto para rodar como "Local Web Site", ao invés de "Script (running on command line)".

Códigos

# Mysql on duplicate key

Para não precisar verificar se o registro já existe:

```
INSERT INTO tabela SET campo1 = 'valor1', campo2 = 'valor2' ON DUPLICATE KEY UPDATE campo1 = 'valor1';
```



Códigos

# Alterando a aparência do show processlist

Com esse comando:

```
SELECT * FROM INFORMATION_SCHEMA.PROCESSLIST WHERE USER = 'usuario' AND state <> '' ORDER BY TIME  
DESC;
```

Isso no Mysql, obviamente.

# Limite de conexões no mysql

Só uns comandos aleatórios para ver se está na hora de aumentar o limite de conexões ao banco de dados:

```
SHOW STATUS LIKE 'max_used_connections';  
show variables like '%conn%';
```

Se precisar alterar temporariamente o limite:

```
set global max_connections = 1000;
```

Lembrando que ao reiniciar o banco, esse limite vai voltar ao que estiver especificado no arquivo de configurações do Mysql.

# Executando métodos de webservice SOAP usando cURL

..e ativando o xdebug:

```
curl -X POST http://webservices.localhost/webservice.php?XDEBUG_SESSION_START=netbeans-xdebug -d 'request.xml content'
```

Para montar o conteúdo em xml que será enviado para o webservice, dá pra fazer assim (no arquivo que recebe a requisição):

```
$requisicao = file_get_contents("php://input");  
file_put_contents('/request.xml', $requisicao);
```

# Mysql - encontrando registros com um ou mais caracteres maiúsculos

```
SELECT * FROM tabela WHERE campo REGEXP BINARY '[A-Z]';
```

# Erro code 1271 - Illegal mix of collations for operation union

Esse erro acontece quando a gente tenta usar a cláusula UNION pra juntar duas consultas de tabelas diferentes, e o collation das tabelas é diferente.

Nesse caso, precisa descobrir o "character set" e o "collation" das duas tabelas, e alterar de alguma delas, pra que ambas fiquem iguais. De preferência alterar o da tabela que tiver menos dados, pra que a alteração seja mais rápida:

```
SHOW FULL COLUMNS FROM tabela;
```

```
SHOW CREATE TABLE tabela;
```

E o comando para alterar:

```
ALTER TABLE outratabela MODIFY coluna VARCHAR(150) CHARACTER SET latin1 COLLATE latin1_swedish_ci;
```

Também pode acontecer um erro parecido, quando se consulta em uma view:

```
Illegal mix of collations (utf8mb4_general_ci,COERCIBLE) and (utf8mb4_unicode_ci,COERCIBLE) for operation '='
```

Neste caso, o campo que causava o erro era um campo "calculado" - ou seja, o valor dele não vinha do banco de dados, mas de uma cláusula IF. Isso porque o campo calculado herda o conjunto de caracteres (vulgo collation) do campo da tabela. Daí para evitar o erro precisamos forçar o collate na hora de criar os campos da view:

```
IF(('produto`.`categoria` <> 3), CONVERT('tipo_1' USING utf8), CONVERT('tipo_2' USING utf8)) AS  
`produto_categoria`
```

Códigos

# Importando vários arquivos SQL de uma vez só

Assim:

```
for SQL in *.sql; do DB=${SQL%.sql/}; echo importando $SQL; mysql -u login -psenha nomedobanco < $SQL; done
```

# Usando migrations no symfony para desfazer migração

Primeiro, precisa da versão da migração.

Basta pegar a data e hora que estão no nome da classe. Por exemplo, se o nome for **Version20190912112303**:

```
php bin/console doctrine:migrations:execute 20190912112303 --down
```

# Lendo arquivos de uma pasta com php

Sempre esqueço.

```
$pasta = __DIR__.'./pasta/processar';
$iterator = new DirectoryIterator($pasta);

foreach ( $iterator as $entry ) {
    if($entry->isFile ()) {
        $file_extension = pathinfo($entry->getFilename(), PATHINFO_EXTENSION);
        if($file_extension == 'log') {
            $arquivos[] = $entry->getPathname();
        }
    }
}
```



# Jquery Ajax

Um exemplo, porque vivo esquecendo.

```
$.ajax({  
  dataType: 'json'  
  ,url: url  
  ,data: form.serialize()  
  ,method:'POST'  
  ,success: function(data){  
    sucesso(data);  
  }  
  ,error: function(jqXHR, textStatus, errorThrown){  
  
  }  
  ,complete: function(){  
    $('#carregando').slideUp();  
  }  
});
```

# Copiando para a área de transferência com Javascript

Ou como simular ctrl+c.

Novo método:

```
navigator.clipboard.writeText("This is the text to be copied").then(() => {  
  console.log('Content copied to clipboard');  
}, () => {  
  console.error('Failed to copy');  
});
```

Fonte: <https://www.freecodecamp.org/news/copy-text-to-clipboard-javascript/>

~~Método antigo:~~

```
function copy(el) {  
  var copyText = el;  
  copyText.select();  
  document.execCommand("copy");  
}
```

~~A função copy recebe como parâmetro o elemento do qual se deseja copiar o valor:~~

```
$('#input')  
  .click(function(){  
    copy(this);  
  })  
);
```

# Avisar se o usuário esqueceu de gravar o formulário

Primeiro, criamos uma variável:

```
var alguem_mexeu_no_meu_queijo = false;
```

No evento change de qualquer elemento, alteramos essa variável, para sabermos que o usuário mexeu no formulário:

```
$('#select, input, textarea').change(function(){  
    alguem_mexeu_no_meu_queijo = true;  
});
```

Depois, no evento beforeunload da janela, precisamos ver se a variável foi alterada. Se sim, damos o alerta, perguntando se ele quer mesmo sair.

```
window.onbeforeunload = function(){  
    if(alguem_mexeu_no_meu_queijo==true)  
    {  
        return 'Alguém mexeu no meu queijo.';  
    }  
}
```

Ah, quando o usuário envia o formulário, precisamos setar a variável como true, para que não apareça o alerta.

Se tu tiver usando o jquery validation, fica assim:

```
$('#form').validate({  
    submitHandler: function(form) {  
        alguem_mexeu_no_meu_queijo = false;  
        form.submit();  
    }  
});
```



# Executando e acompanhando tarefas demoradas na linha de comando do MySQL

Primeiro, precisamos do comando sql que será executado.

Execute o comando no seu programa preferido (Workbench, direto na linha de comando, etc) e finalize o programa.

Execute o comando abaixo para ver o status do comando (oi?)

```
mysql -v -h ipdobanco -u login -psenha nomedabase -e "show processlist;"
```

O retorno será de pelo menos 2 linhas, uma detalhando o próprio comando SHOW PROCESSLIST, e outra com o comando que foi executado antes:

```
-----
show processlist
-----
+-----+
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| Id | User | Host | db | Command | Time | State | Info | Rows_sent | Rows_examined |
+-----+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
| 19241044 | login | 127.0.0.1:39922 | nomedabase | Query | 529 | updating | delete from tabela | 0 | 416450 |
| 19258640 | login | 127.0.0.1:54285 | nomedabase | Query | 0 | init | show processlist | 0 | 0 |
+-----+-----+-----+-----+-----+-----+-----+-----+
+-----+
```

Agora, precisamos executar esse comando de tempos em tempos, para verificarmos quando finalizou.

```
while sleep 60; do COMANDO; done
```

O exemplo acima fica assim:

```
while sleep 60; do mysql -v -h ipdobanco -u login -psenha nomedabase -e "show processlist;"; done
```

O 60 significa que o comando será executado a cada 60 segundos. Basta alterar pra executar numa frequência diferente.

Códigos

# Buscar ignorando acentos no Mysql

A query tem que ficar assim:

```
campo LIKE _utf8 '%$variavel%' COLLATE utf8_unicode_ci
```

Não esquecer de escapar a \$variavel.

# Ordenando elementos HTML com o Bootstrap

É só aplicar as classes **order-1** até **order-12** nos elementos, que devem estar dentro de um **.row**.

O detalhe aqui é que o bootstrap só permite até o 12, se tiver mais elementos que isso, toda a ordenação falha.

Nesse caso, precisa criar as classes que faltam, para que funcione:

```
.order-99 {  
  -ms-flex-order: 99;  
  order: 99;  
}
```



# Inserindo HTML em um iframe com jQuery

Às vezes precisamos gerar uma pré-visualização de um bloco de HTML, mas o CSS da página atual pode (e provavelmente vai) conflitar com o da prévia.

Nesse caso, podemos jogar o conteúdo em um iframe:

```
$("#preview_01").contents().find('html').html($('#editor_html').val());
```

# Máscara para telefones com 8 ou 9 dígitos

Esse é um exemplo de uso do plugin masked-input do jQuery para formatação de telefones com 8 ou 9 dígitos.

Quando o campo possui 8 ou menos dígitos, a formatação normal é aplicada, quando possui mais, a formatação é alterada (apesar de eu não concordar com essa formatação, mas enfim).

Link: <https://github.com/digitalBush/jquery.maskedinput>

```
$('#input[name="fone_cartao"]')
    .mask(
        "(99) 9999-9999?9"
    , {
        placeholder: "xxx.xxx.xxx-xx"
    }
    )
    .change(function(){
        var valor = $(this).val().replace(/[^\0-9]/g, '');
        if(valor.length > 10)
        {
            $(this).mask("(99) 99999-999?9");
        } else {
            $(this).mask("(99) 9999-9999?9");
        }
    }
    );
```

# Exportando tabelas para excel e acertando o formato dos campos

Como 'gerar' um arquivo excel com php, todo mundo já sabe:

```
$arquivo = "export".date("dmY_His").".xls";  
header("Content-Type: text/plain");  
header("Content-Disposition: attachment; filename=\"$arquivo\";");
```

Mas como fazer com que as colunas fiquem com os tipos de dados certos, como por exemplo, os campos de data?

Simples: Criando o seguinte css:

```
.date {  
    mso-number-format:"Short Date";  
}
```

E aplicando este estilo na célula da tabela que contém o valor que deve ser formatado.

Aqui tem os formatos aceitos: <http://cosicimiento.blogspot.com.br/2008/11/styling-excel-cells-with-mso-number.html>

Fonte:

<http://cosicimiento.blogspot.com.br/2008/11/styling-excel-cells-with-mso-number.html>

<http://stackoverflow.com/questions/354476/html-to-excel-how-can-tell-excel-to-treat-columns-as-numbers>

# Utilizando jquery-ui tabs + jquery validation plugin

Quando o plugin encontra um elemento com erro, o elemento automaticamente ganha foco. Mas quando o elemento está em uma aba que não está habilitada no momento, não acontece nada.

Pra que o elemento ganhe foco e a aba passe a ser a ativa, segue a função:

```
$('#formulario').validate({  
  ignore: "", //para validar os elementos que estão nas abas desabilitadas  
  highlight: function(element, errorClass, validClass) {  
    var div = $(element).parents('.ui-tabs-panel').attr('id'); //encontra a div em que está o elemento inválido  
    var index = $('#tabs ul li a').index($('a[href="'+div+'"]')); //encontra o indice nos links  
    $('#tabs').tabs( "option", "active", index ); //seta a aba como ativa  
  }  
});
```

# Alguns comandos do MySQL

Medindo os tamanhos dos bancos:

```
SELECT  
TABLE_NAME,  
SUM(data_length + index_length) / 1024 / 1024 AS 'Data Base Size in MB',  
SUM(data_free) / 1024 / 1024 AS 'Free Space in MB'  
FROM  
information_schema.TABLES  
WHERE table_schema = 'nome_do_banco'  
GROUP BY TABLE_NAME;
```

Medindo o tamanho das tabelas:

```
SELECT  
table_schema,  
table_name,  
table_rows,  
(data_length + index_length) / 1024 / 1024 AS 'Data Base Size in MB',  
(data_free) / 1024 / 1024 AS 'Free Space in MB'  
FROM  
information_schema.TABLES  
where table_schema = 'nome_do_banco'  
ORDER BY data_length + index_length DESC
```

Códigos

# jQuery hasAttr

Um plugin simples para utilizar a função hasAttr com jQuery, que funciona do mesmo jeito que hasClass.

```
$.fn.hasAttr = function(name) { return this.attr(name) !== undefined; };
```

# Validando datas em pt-br com jQuery Validate

Assim:

```
$.validator.addMethod(
    "date",
    function(value, element) {
        var check = false;
        var re = /^\\d{1,2}\\\\d{1,2}\\\\d{4}$/;
        if( re.test(value)){
            var adata = value.split('/');
            var gg = parseInt(adata[0],10);
            var mm = parseInt(adata[1],10);
            var aaaa = parseInt(adata[2],10);
            var xdata = new Date(aaaa,mm-1,gg);
            if ( ( xdata.getFullYear() == aaaa ) && ( xdata.getMonth () == mm - 1 ) && ( xdata.getDate() == gg ) )
                check = true;
            else
                check = false;
        } else
            check = false;
        return this.optional(element) || check;
    },
    "Insira uma data válida"
);
```

Códigos

# Acessando conteúdo do CKEDITOR

Assim:

```
CKEDITOR.instances.IdDoElemento.getData()
```

Lembrando que no lugar de `IdDoElemento`, vai, obviamente, o Id do elemento.

Lembrando também que é preciso ter cuidado com o óbvio, já que ele nem sempre é tão óbvio assim.

(a propósito: toda vez que vejo o nome **CKEDITOR**, eu penso em **FUCKEDITOR**, principalmente quando ele não colabora comigo)



# Contar dias ou horas entre datas em Mysql

Para contar quantas horas existem entre duas datas:

```
SELECT TIMEDIFF(STR_TO_DATE('19/01/1985 00:00:00', '%d/%m/%Y %H:%i:%s'),now());
```

Trocando TIMEDIFF por 'DATEDIFF' dá pra contar quantos dias tem entre essas datas.

Códigos

# jQuery On

```
$('.qualquerItemPai').on("click", ".algumItemFilho", function(){  
    //algumacoisa  
});
```

Códigos

# D.R no Mysql

ou criando relações entre tabelas

```
ALTER TABLE tabela  
ADD CONSTRAINT nome_da_relacao  
FOREIGN KEY (coluna_da_tabela)  
REFERENCES tabela_pai(id_da_tabela_pai)  
ON DELETE CASCADE  
ON UPDATE NO ACTION;
```

Caso precise remover a relação:

```
ALTER TABLE tabela DROP FOREIGN KEY nome_da_relacao
```

# Fazendo caso

ou usando SELECT CASE no MySQL:

```
SELECT id, nome, descricao,  
CASE status  
  WHEN 1 THEN 'Aprovado'  
  WHEN 2 THEN 'Vencido'  
  WHEN 3 THEN 'Pendente'  
  WHEN 0 THEN 'Reprovado'  
  ELSE 'Indefinido'  
END AS status  
FROM tabela
```

outro exemplo:

```
CASE  
  WHEN campo1 > 0 THEN campo1  
  WHEN campo2 > 0 THEN campo2  
  WHEN campo3 > 0 THEN campo3  
  ELSE campo0  
END AS campo
```

# Executando um comando como Root via php

Assim:

```
<?php
```

```
$outPut = shell_exec("echo senha | sudo -S apt-get -y install openssh-server");
```

```
echo "<pre>$outPut</pre>";
```

```
?>
```

O detalhe é o 'echo' antes, pra enviar a senha pro sudo, e o -y, pra não pedir confirmação sobre a instalação.

# Exportando um ou mais registros no MySql com mysqldump

Às vezes a gente precisa exportar só um registro de uma tabela. Dá pra fazer assim:

```
mysqldump -t -u usuario -psenha banco tabela --where=id_tabela=666 > ~/insert.sql
```

Só pra lembrar: o parâmetro -p recebe a senha 'colada' nele, por exemplo, se a senha for 'estrogonofe', fica assim:

```
mysqldump -t -u usuario -pestrogonofe banco tabela --where=id_tabela=666 > ~/insert.sql
```

E no final, onde tem '~/insert.sql', é onde vai ser gerado o arquivo com o comando sql pra inserir de volta no banco.

# Lendo arquivos de uma pasta com php

```
$pasta = __DIR__.'./pasta/processar';  
$iterator = new DirectoryIterator($pasta);  
  
foreach ( $iterator as $entry )  
{  
    if($entry->isFile ())  
    {  
        $file_extension = pathinfo($entry->getFilename(), PATHINFO_EXTENSION);  
        if($file_extension=='log')  
        {  
            $arquivos[] = $entry->getPathname();  
        }  
    }  
}
```

# Utilizando plugin jquery validation em formulário com abas

Quando o plugin encontra um elemento com erro, esse elemento automaticamente ganha foco. Mas quando o elemento está em uma aba que não está habilitada no momento, não acontece nada.

Pra que o elemento ganhe foco e a aba passe a ser a ativa, segue a função:

```
$('#formulario').validate({  
    ignore: "", //para validar os elementos que estão nas abas desabilitadas  
    highlight: function(element, errorClass, validClass) {  
        var div = $(element).parents('.ui-tabs-panel').attr('id'); //encontra a div em que está o elemento  
        inválido  
        var index = $('#tabs ul li a').index($('a[href="#'+div+'"]')); //encontra o indice nos links  
        $("#tabs").tabs( "option", "active", index ); //seta a aba como ativa  
    }  
});
```



# Executando comandos no client do MySQL sem informar a senha

Primeiro, crie um arquivo com o seguinte padrão:

```
[client]
host=localhost
user=usuario
password='senha'
port=3306
[mysqldump]
column-statistics=0
```

Depois, ao executar o programa, informe o caminho para esse arquivo:

```
mysql --defaults-extra-file=/caminho/arquivo nomedobanco ...
```

# Desativando chaves estrangeiras no Mysql

Às vezes a gente precisa excluir uma tabela que tem relacionamento com outras.

```
SET FOREIGN_KEY_CHECKS=0; DROP TABLE IF EXISTS nomedatabela;
```

Códigos

# Git

Várias coisas de git:

## Voltar um commit temporariamente:

```
git checkout X (go back in time)
```

```
<make changes and build>
```

```
git checkout . (throw away changes)
```

```
git checkout master (go forward to "now")
```

(fonte: <https://www.reddit.com/r/git/comments/gok5g1/comment/frg9o0s/>)

## Listando arquivos modificados em um commit:

```
git diff-tree --no-commit-id --name-only -r bd61ad98
```

# Linux

Comandos, dicas e outras cositas más que uso de vez em quando

# Copiar e mesclar duas pastas no terminal

Usando rsync:

```
rsync -a /origem/ /destino/
```

No caso acima, vai copiar o conteúdo de origem para destino.

O detalhe é que o destino precisa terminar com /, senão a pasta vai ser copiada dentro da outra, e não mesclada.

Fonte: <https://unix.stackexchange.com/questions/149965/how-to-copy-merge-two-directories>

Linux

# dig dig dig dig

..ou brincando com dns:

Verificando se uma entrada CNAME existe:

```
dig subdominio.dominio.com cname
```

Forçando a verificação a passar por um servidor DNS (pra ver se já propagou por ele):

```
dig @1.0.0.1 subdominio.dominio.com cname
```

(no caso, 1.0.0.1 é o ip do servidor DNS)

Alterando os servidores DNS do servidor:

```
nano /etc/resolv.conf
```

alterar (ou apenas comentar) as linhas que possuem "nameserver", apontando para o ip desejado

# Erro ao atualizar certificados no let's encrypt

Isso no servidor Debian.

Se aparecer a mensagem:

```
("bad handshake: Error([('SSL routines', 'tls_process_server_certificate', 'certificate verify failed')],)".). Skipping
```

Primeiro, precisa tentar instalar os certificados raiz (ou algo assim):

```
apt install ca-certificates
```

Se aparecer a mensagem "ca-certificates is already the newest version", precisa instalar manualmente:

1) Baixar o arquivo de instalação dos certificados:

[https://security.debian.org/debian-security/pool/main/c/ca-certificates/ca-certificates\\_20200601~deb9u2\\_all.deb](https://security.debian.org/debian-security/pool/main/c/ca-certificates/ca-certificates_20200601~deb9u2_all.deb)

(novo link: [https://archive.debian.org/debian-security/pool/updates/main/c/ca-certificates/ca-certificates\\_20200601~deb9u2\\_all.deb](https://archive.debian.org/debian-security/pool/updates/main/c/ca-certificates/ca-certificates_20200601~deb9u2_all.deb))

2) instalar os certificados:

```
dpkg -i ca-certificates_20200601~deb9u2_all.deb
```

3) Pedir para o let's encrypt atualizar os certificados dos sites:

```
certbot renew
```

Se não aparecer a mensagem de que os certificados já estão na última versão, basta pedir para o let's encrypt atualizar os certificados. Eu acho.

Linux

# Comprimindo pastas

```
tar -zcvf archive-name.tar.gz directory-name
```

extrair:

```
tar -zxvf archive-name.tar.gz
```

Se o formato utilizado for **xz**, precisa trocar o `z` por `J`:

```
tar -Jxvf arquivo.tar.xz
```



# Alterando dependências de um pacote deb

1. Descompactar deb: `ar x nomedoarquivo.deb` (serão criados três arquivos: debian-binary control.tar.gz data.tar.gz)
2. Descompactar arquivo control: `tar xzf control.tar.gz` (vai criar alguns arquivos como: postinst postrm preinst prerm md5sums control)
3. Corrigir dependências no arquivo control (exemplo, alterando de **Depends** para **Recommends**)
4. Recompactar control.tar.gz: `tar c postinst postrm md5sums control | gzip -c > control.tar.gz` (verificar todos os arquivos gerados no passo 2)
5. Recompactar arquivo deb: `ar rcs novopacote.deb debian-binary control.tar.gz data.tar.gz` (a ordem é importante!)

Fonte: <https://coderwall.com/p/hes3ha/change-the-dependencies-of-a-deb-package>

# Encontrando e substituindo em arquivos no shell

Procurar um determinado texto em vários arquivos:

```
find . -type f -exec grep -l 'gerar_boletos' {} \;
```

Procurar um determinado texto em um arquivo e salvar o resultado em outro arquivo:

```
grep '177.141.123.210' access_log20140604 > encontrou.log
```

Para apenas listar os nomes dos arquivos, informar o parâmetro -l para o grep:

```
grep -l procurar *
```

Procurando e substituindo um texto em vários arquivos:

```
find . -type f -name "*.php" -exec sed -i 's/procurar/substituir/g' {} +
```

Não esquecer de escapar as barras!

# Removendo um ip do Fail2ban

1) Listar os ips bloqueados:

```
iptables -L -n
```

2) Listar os nomes das regras do fail2ban:

```
fail2ban-client status | grep "Jail list" | sed -E 's/^[^:]+:[ \t]+//' | sed 's/,//g'
```

3) Desbloquear o ip, informando também o nome da regra:

```
fail2ban-client set REGRA unbanip ENDERECO
```

exemplo:

```
fail2ban-client set sshd unbanip 186.204.26.11
```

# Mini-curso mega-rápido de Torrents e Piratebay

Aqui vai um tutorial resumido de como baixar filmes no piratebay.

Primeiro, precisa de um client de bit torrent. Eu sempre uso o Transmission, porque é bem simples:

<https://transmissionbt.com/download/>

Depois de instalar, abra o navegador e acesse o <https://thepiratebay.org/>. Não esqueça de ativar bloqueador de anúncios, pra não ser bombardeado com anúncios chatos.

Pesquise pelo conteúdo que quiser baixar. Na tela do resultado da busca, preste atenção na coluna "**SE**" - essa coluna indica a quantidade de Seeds, ou seja, pessoas que tem o arquivo completo e estão distribuindo. Se não tiver nenhum, a chance de conseguir baixar o arquivo completo é menor. A coluna "**LE**" indica a quantidade de Leeches, que são as pessoas que estão transferindo o arquivo mas não possuem ele completo. Quanto mais Seeds e mais Leeches, mais rápido vai ser o download.

Também é bom prestar atenção no tamanho do arquivo. Para filmes, o tamanho ideal fica entre 800Mb e 4 ou 5 Gb. Menos do que isso pode ser algo com a qualidade muito baixa, e mais do que isso pode ser qualidade muito alta - não que seja um problema, mas nem todos os tocadores conseguem reproduzir vídeos com definição alta demais, sem contar que o tempo de download vai ficar muito maior.

E fuja de torrents que tem as palavras **TS** ou **CAM** no título - estes são aqueles filmes em que alguém grava dentro do cinema, e geralmente a qualidade é terrível.

Após decidir qual torrent vai ser baixado, ainda na tela de resultado da busca, clique com o botão direito no ícone de ímã (🧲) e copie o link magnético - é um link que começa com **magnet:?**.

No programa **Transmission**, clique em **File - Open URL** e cole o link magnético, e é isso. Lembrando que a velocidade vai depender de quantos Seeds e quantos Leeches estão distribuindo o torrent.

Alguns tocadores de vídeo baixam automaticamente as legendas. Caso o seu tocador não faça isso, ou não encontre a legenda, eu costumo copiar o nome completo do arquivo de vídeo e pesquiso no google, adicionando a palavra "legendas".

# Operações com arquivos no shell

Excluindo arquivos com mais de 90 dias:

```
find pasta -mtime +90 -type f -exec rm -f {} \;
```

(altere o +90 para -90 para pegar arquivos com menos de 90 dias)

Excluindo arquivos com uma determinada extensão:

```
find pasta -type f -name *.sql -exec rm -f {} \;
```

Excluindo pastas vazias:

```
find pasta -type d -empty -exec rm -Rf {} \;
```

Excluindo arquivos vazios:

```
find pasta -type f -empty -exec rm -f {} \;
```

Para listar os arquivos ao invés de excluir, troque o `rm -f` por `ls -l`.

Extraindo arquivos existentes em sub-pastas:

```
for file in `find *`; do  
gunzip "${file}" ; done
```

Mesclando vários arquivos em um só:

```
find . -type f -name 'access-*.log' -exec cat {} + >> output.file
```

Gerando log com o goaccess:

```
goaccess output.log -a -o log.html
```

Substituindo texto em um arquivo

```
sed -i 's/procurar/substituir/g' arquivo.txt
```

Dividindo um arquivo em partes:

Dividir o arquivo lista.txt em partes de 1000 linhas cada.  
Cada parte do arquivo será chamada parte1.txt, parte2.txt etc.

```
split -l 1000 lista.txt parte
```

Extrair um arquivo tgz:

```
tar -zxvf arquivo.tgz
```

Extrair um arquivo específico que está dentro do tgz:

```
tar -zxvf arquivo.tgz arquivo_interno.txt
```

Comprimindo uma pasta/arquivo no formato tgz:

```
tar -cvpzf arquivo.tgz pasta
```

Caso o arquivo seja .tar, basta tirar o z do comando:

```
tar -xvf arquivo.tar
```

Linux

# Exibir qual processo está usando uma determinada porta

Assim:

```
lsof -w -n -i tcp:9000
```

# Agendando desligamento no Debian

Editar o arquivo `/etc/rc.local`:

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

# shutdown everyday at 6 am
shutdown -h 06:30 &

exit 0
```



# Vídeos de Timelapse

Itens necessários:

- Celular para capturar as imagens;
- Algum programa para tirar fotos de minuto a minuto (<http://survey7.com/cameraupload/>)
- FFmpeg;

Girando as imagens (caso elas tenham sido capturadas com o celular na horizontal):

```
for file in *.jpg; do convert "$file" -rotate -90 "${file%.jpg}"_rotated.jpg; done
```

As imagens convertidas terão o nome "\_rotated.jpg".

Agora, precisamos criar o vídeo:

```
ffmpeg -r 5 -pattern_type glob -i '*_rotated.jpg' -vcodec libx264 timelapse_5fps.mp4
```

Altere o parâmetro `-r 5` para definir a quantidade de quadros por segundo. Quando for um vídeo longo, é melhor usar uma taxa maior, tipo 15.

# Criando e distribuindo um torrent usando o Transmission

Usando o terminal, é óbvio.

Primeiro, instale o transmission. Junto com ele, vai vir o daemon e alguns programas. Os que precisamos são o transmission-create e o transmission-remote.

Depois, precisamos de uma lista de trackers. Neste link tem alguns:

<https://github.com/ngosang/trackerslist>

Navegue até um nível acima da pasta/arquivo que será compartilhada, e execute este comando, para criar o arquivo .torrent:

```
transmission-create pasta_a_ser_compartilhada/ -o arquivo.torrent -t http://umtracker:1337/announce -t  
udp://outrotracker:1337/announce
```

Informe quantos trackers puder, para facilitar a distribuição. Neste caso tem só dois, mas no teste que fiz, coloquei 57.

Agora é preciso começar a semear o torrent, para que as outras pessoas possam baixar.

```
transmission-remote --add arquivo.torrent --download-dir  
/caminho_completo_ate_a_pasta_compartilhada/ -n  
usuario_do_transmission:senha_do_transmission
```

Aqui tem um detalhe: É preciso informar o caminho para chegar na pasta/arquivo compartilhado na opção `--download-dir`. Com isso, o Transmission vai perceber que tu já tem todos os arquivos do torrent, e vai começar a semear. Isso é importante caso tu esteja semeando um arquivo/pasta que não esteja na pasta padrão de downloads do Transmission.

Por exemplo, se tu vai compartilhar a pasta `/home/usuario/Downloads/blariblabla`, os comandos ficam assim:

```
cd ~/Downloads
```

```
transmission-create blariblabla/ -o arquivo.torrent -t http://umtracker:1337/announce -t  
udp://outrotracker:1337/announce
```

```
transmission-remote --add arquivo.torrent --download-dir  
/home/usuario/Downloads/ -n  
usuario_do_transmission:senha_do_transmission
```

Depois disso, é só distribuir o arquivo .torrent - seja através do Piratebay, ou de qualquer outro modo.

Um jeito bom de testar se o torrent vai funcionar é baixando ele através do <https://seedr.cc/>.

Linux

# Convertendo de MP4 para GIF no Debian

Com o ffmpeg:

```
ffmpeg -i video.mp4 imagem.gif
```

Provavelmente também funciona com outros formatos de vídeo.

# Montando drives usb na linha de comando

Listando os dispositivos e partições encontrados:

```
sudo fdisk -l
```

Montando a partição do dispositivo em uma pasta:

```
sudo mount /dev/sdb1 /media/usb/
```

Obs: A pasta /media/usb precisa ter sido criada antes.

Listando o espaço em disco dos dispositivos:

```
df
```

Desmontando a partição para remover o drive:

```
sudo umount /media/usb
```

Linux

# Executando um comando repetidas vezes no terminal

Assim:

```
while sleep TEMPO; do COMANDO; done
```

Exemplo: exibir a data, a cada segundo:

```
while sleep 1; do date; done
```

# Acesso ssh via tor

Primeiro, instalamos tor e o connect-proxy:

```
sudo apt install tor connect-proxy
```

Depois, configuramos o arquivo `~/.ssh/config` para usar o proxy:

```
Host *
```

```
CheckHostIP no
```

```
Compression yes
```

```
Protocol 2
```

```
ProxyCommand connect -4 -S localhost:9050 $(tor-resolve %h localhost:9050) %p
```

Por fim, conectamos no ssh:

```
ssh user@server
```

Se der a mensagem de erro `ERROR: You must specify the destination port correctly.`, então precisa usar o ip do servidor, ao invés do hostname.

# Removendo arquivos com interrogação (?) no nome

Às vezes aparecem uns arquivos chamados `?.pub`, ou apenas `?` na pasta home.

Na verdade, esse não é o nome dos arquivos. O que acontece é que o nome deles é formado por caracteres unicode que não podem ser exibidos no console.

O jeito é descobrir o inode dos arquivos, e excluir através dele.

Listando o inode dos arquivos:

```
ls -il
```

Excluindo através do inode:

```
find . -inum 15681478197 -exec rm -i {} \;
```

Fonte: <https://unix.stackexchange.com/a/230030>



# Trabalhando com arquivos no terminal

Listando:

```
ls -l
```

Ordenando pela data:

```
ls -l --sort=time
```

Ordenando inversamente:

```
ls -l --sort=time -r
```

Exibindo data e hora completos do arquivo:

```
ls -l --time-style=full-iso
```

Contando quantos arquivos tem na pasta:

```
ls | wc -l
```

Alterando data de modificação de arquivos:

```
touch -d "2 hours ago" filename
```

Para remover arquivos com uma determinada extensão em várias sub-pastas, recursivamente:

```
find pasta -name '*.extensao' -exec rm -f {} \;
```

Obviamente, pode ser usado na pasta atual:

```
find . -name '*.extensao' -exec rm -f {} \;
```

Encontrando uma linha específica em um arquivo:

```
sed -n '12345p' < arquivo.txt
```

Procurar arquivos que contenham um texto:

```
find . -exec grep -l '_0xaae8' {} \;
```

(tenho a impressão de que já tenho um comando parecido com esse em algum lugar, mas não tenho certeza)

Fontes: <https://unix.stackexchange.com/questions/1125/how-can-i-get-a-count-of-files-in-a-directory-using-the-command-line>

<https://askubuntu.com/questions/62492/how-can-i-change-the-date-modified-created-of-a-file>

# Virtualbox - faltando headers

Ao iniciar o Virtualbox, aparece uma mensagem dizendo para executar o `/sbin/vboxconfig` como root.

Ao executar o comando indicado, aparece a seguinte mensagem de erro:

```
This system is currently not set up to build kernel modules.  
Please install the Linux kernel "header" files matching the current kernel  
for adding new hardware support to the system.  
The distribution packages containing the headers are probably:  
linux-headers-amd64 linux-headers-4.9.0-6-amd64  
This system is currently not set up to build kernel modules.  
Please install the Linux kernel "header" files matching the current kernel  
for adding new hardware support to the system.  
The distribution packages containing the headers are probably:  
linux-headers-amd64 linux-headers-4.9.0-6-amd64  
There were problems setting up VirtualBox. To re-start the set-up process, run  
/sbin/vboxconfig  
as root.
```

Para resolver, basta instalar os headers indicados na mensagem de erro.

```
sudo apt install linux-headers-amd64 linux-headers-4.9.0-6-amd64
```

E depois, rodar de novo o comando para configurar o Virtualbox:

```
sudo /sbin/vboxconfig
```

O detalhe aqui é verificar o header descrito na mensagem de erro.

Linux

# Mexendo com arquivos wbfs para usar no Nintendo Wii

Extraindo de rar para iso:

```
unrar x arquivo.part01.rar
```

Convertendo de ISO, WDF, WIA, CISO, WBFS, GCZ ou FST para wbfs:

```
wit copy --source game.iso --wbfs -d game.wbfs --split -v
```

Encontrando o ID correto no arquivo wbfs:

```
wwt LIST-L game.wbfs
```

Fonte: <https://negativo17.org/wit-wiimms-iso-tools/>

<https://wit.wiimm.de/wit/>

# Gravando um arquivo ISO em um pendrive

Usando o comando dd:

```
sudo dd if=arquivo.iso of=/dev/sdc status=progress
```

Só que isso nem sempre faz com que o pendrive seja "bootável". Por isso, o melhor jeito é instalar o [Ventoy](#) em um pendrive, depois disso, basta copiar os arquivos iso nele que o programa faz o resto. Faz uns dois anos que não preciso mais ficar criando pendrives bootáveis.

Linux

# Erro NO\_PUBKEY no Ubuntu/Debian

Mensagem de erro:

```
The following signatures couldn't be verified because the public key is not available: NO_PUBKEY  
978228591BD3A65C
```

Comando:

```
sudo apt-key adv --keyserver keyserver.ubuntu.com --recv-keys 978228591BD3A65C
```

Fonte: <https://www.vivaolinux.com.br/topico/Linux-Basico/erro-com-chave>

# Backup com rsync

```
rsync -azP --bwlimit=5000 --exclude=git user@host:/origem destino/
```

Quando esquecer para que servem os parâmetros, uma dica é usar o site Explain Shell:

<https://explainshell.com/explain?cmd=rsync+-azP+--bwlimit%3D5000+--exclude%3Dgit+user%40host%3A%2Forigem+destino%2F>

Linux

# Listando portas X serviços

```
sudo lsof -iTCP -sTCP:LISTEN -P
```



Linux

# Um pouco de windows

Alguns comandos úteis (ou não) para o windows 10.

Girar a tela:

Ctrl+Alt+Seta para os lados

Usar vários desktops:

Win + Tab

P.S.: eu sei que essa seção se chama "Linux", mas não vou criar uma seção pra postar dois comandos, né

# Baixando arquivos recursivamente via FTP

Porque nem sempre dá pra usar o sftp.

```
wget -v -r -t 1 --include-directories=public_html,public,httpdocs,httpsdocs,Web --user=usuario --password=senha ftp://dominio.com/
```

no parâmetro include-directories, passar as pastas que serão baixadas. Se for pra pegar tudo, não precisa passar nada.

Linux

# nohup

Para executar um comando e sair sem esperar o retorno.

Pode parecer falta de educação deixar o comando falando sozinho, mas às vezes é necessário.

```
nohup comando > /dev/null 2>&1 &
```

# Renomeando arquivos sequencialmente

Pega vários arquivos chamados "parte1", "parte2" etc. e renomeia para 0001, 0002 e assim por diante.

```
a=1
for i in parte*; do
new=$(printf "%d" "$a") #04 pad to length of 4
mv -- "$i" "$new"
let a=a+1
done
```

# Mexendo com o Swap

```
sudo swapon --show //consultando a utilização do swap
```

```
sudo swapoff -a //desligando swap
```

```
sudo swapon --show //confirmando se desligou
```

```
sudo fallocate -l 4G /swapfile //criando um arquivo de 4Gb para usar como swap
```

```
ls -lh /swapfile //conferindo o tamanho do arquivo
```

```
sudo chmod 600 /swapfile //dando permissão de acesso apenas ao root
```

```
sudo mkswap /swapfile //transformando o arquivo em swap
```

```
sudo swapon /swapfile //habilitando o swap
```

```
sudo swapon --show //conferindo se habilitou
```

```
cat /proc/sys/vm/swappiness //conferindo o swappiness
```

```
sudo sysctl vm.swappiness=30 //alterando o valor do swappiness para 30
```

```
gksudo gedit /etc/sysctl.conf //alterando permanentemente o valor de swappiness
```

```
vm.swappiness=10 //se nao existir esta linha, criar
```

```
echo '/swapfile none swap sw 0 0' | sudo tee -a /etc/fstab //mantendo a alteração permanente
```

Obs: não esquecer de editar o /etc/fstab, comentando a linha da partição de swap, caso já exista uma antes.

# DDNS com a Digital Ocean

Como o DynDNS agora é pago, o jeito é usar alguma outra solução de dns dinâmico, de preferência gratuita.

No caso não é exatamente gratuita - afinal, a Digital Ocean cobra pelo servidor - mas enfim.

Primeiro, siga os passos da instalação:

[https://github.com/digitalm3/DigitalOcean\\_dyndns](https://github.com/digitalm3/DigitalOcean_dyndns)

Depois de seguir os passos, precisa criar um subdomínio no painel da DO.

Não esqueça que este subdomínio precisa ter uma entrada do tipo A apontado para o IP dinâmico, e não para o IP do servidor.

Para atualizar esse IP, instale o ddclient.

Configure o arquivo `/etc/ddclient.conf` da seguinte forma:

```
# /etc/ddclient.conf
script=/nic/update.php
protocol=dyndns2
#use=if, if=enp2s0
use=web, web=checkip.dyndns.org/
server=dyndns.meudominio.com
login=qualquercoisa
password='qualquercoisa'
meusubdominio.meudominio.com
```

na linha **server=dyndns.meudominio.com**, informe a url de onde ficou instalado o script do primeiro passo. No meu caso, eu criei um subdomínio chamado dyndns, mas poderia ter instalado no domínio principal.

O login e senha podem ser qualquer coisa, já que o script não faz nenhuma autenticação (o que pode ser perigoso, caso vá rodar algum serviço sigiloso nesse ddns).

Já a última linha é o subdomínio que foi criado no segundo passo.

Depois, é só rodar um **sudo ddclient update** que o IP vai ser atualizado.

Ah, a linha **use=web, web=checkip.dyndns.org/** é importante para que o ddclient pegue o IP externo da tua rede. Caso contrário, ele pode pegar o IP interno (192.168.0.6, por exemplo).

Linux

# Atualizando o Debian/Ubuntu

Comando pra atualizar o Ubuntu ou Debian.

Eu sempre esqueço...

```
sudo apt update -y && sudo apt upgrade -y && sudo apt dist-upgrade -y && sudo apt-get autoremove -y  
sudo reboot
```

Linux

# Criando link simbólico para uma pasta

Assim:

```
ln -s /var/log/apache2/ ~/logs
```

O detalhe é que a pasta ~/logs não deve existir na hora em que for rodar o comando.



Linux

# Trocando o editor do crontab para o nano

Só vale para a sessão atual do ssh, mas já ajuda:

```
env EDITOR=nano crontab -e
```

# Configurando o Percona Monitoring and Management usando Docker

As instruções são um pouco confusas, então achei melhor criar um guia.

Requisitos:

- 1) Uma máquina rodando Docker com alguma porta exposta (no exemplo usei a **4443**)
- 2) Uma ou mais máquina rodando MariaDB ou Mysql, com acesso root

## 1) Instalar o Servidor PMM

Baixar a imagem do docker:

```
docker pull percona/pmm-server:2
```

Criar o volume para armazenar os dados:

```
docker volume create pmm-data
```

Iniciar o container:

```
docker run --detach --restart always --publish 4443:443 --volume pmm-data:/srv --name pmm-server percona/pmm-server:2
```

## 2) Configurar o servidor

Acesse a administração e crie uma senha. Os dados do primeiro acesso são admin / admin:  
<https://ip-do-servidor:8443>

## 3) Instalar o cliente

Em cada máquina que será monitorada, instale o pmm-client. Por exemplo, no Debian fica assim:

```
wget https://repo.percona.com/apt/percona-release_latest.generic_all.deb
```

```
dpkg -i percona-release_latest.generic_all.deb
```

```
sudo apt update
```

```
sudo apt install -y pmm2-client
```

Nesse link tem mais instruções para outras plataformas: <https://docs.percona.com/percona-monitoring-and-management/setting-up/client/index.html#install>

## 4) Configurar os clientes

Crie um usuário no banco com algumas permissões básicas. O exemplo abaixo é para o MariaDB:

```
CREATE USER 'pmm'@'127.0.0.1' IDENTIFIED BY 'pass';  
UPDATE mysql.user set MAX_USER_CONNECTIONS = 10 WHERE user = 'pmm';  
GRANT SELECT, PROCESS, REPLICATION CLIENT, RELOAD ON *.* TO 'pmm'@'127.0.0.1';  
FLUSH PRIVILEGES;
```

Aqui tem mais exemplos: <https://docs.percona.com/percona-monitoring-and-management/setting-up/client/mysql.html#create-a-database-account-for-pmm>

Conecte o agente com o servidor:

```
pmm-admin config --server-insecure-tls --server-url=https://admin@localhost:senha@ip-do-servidor:4443 --force
```

**admin@localhost** é o usuário que você criou no painel, **senha** é a senha desse usuário.

Conecte o agente com o banco de dados:

```
pmm-admin add mysql --username=pmm --password=senha --server-insecure-tls --server-url=https://admin:senha@ip-do-servidor:4443 --query-source=perfschema hostname localhost:3306
```

**--username** e **--password** são as credenciais de acesso ao banco. As outras credenciais são as mesmas informadas acima (pois é). **perschema** indica que o cliente vai usar a tabela **performance\_schema** para as estatísticas (também pode ser pelo arquivo de slow query, mas não achei exemplo de como funciona). **hostname** é o nome que vai ser exibido no servidor, e é bom que seja diferente para cada instância monitorada.

Fontes:

<https://github.com/percona/pmm>

<https://docs.percona.com/percona-monitoring-and-management/index.html>

# Habilitando APM no log do Nginx

APM é um modo de medir a performance de um sistema.

Por padrão, o log do nginx mostra apenas a hora em que a requisição foi servida, mas não mostra o tempo que ela levou para ser processada.

Essa configuração serve para que o nginx exiba esse tempo, para que seja possível identificar as requisições mais demoradas e que precisam de otimização.

O primeiro passo é encontrar o arquivo nginx.conf (geralmente em /etc/nginx/nginx.conf) e adicionar o seguinte template de log:

```
log_format apm '$remote_addr - $remote_user [$time_local] '
    '$request' $status $body_bytes_sent '
    '$http_referer' '$http_user_agent' ($request_time);
```

A variável [\\$request\\_time](#) possui o tempo que levou para processar a requisição. Coloquei entre parênteses apenas para ficar mais fácil de identificar.

O próximo passo é adicionar a configuração no arquivo vhost do site, para que o log seja gerado:

```
access_log /var/log/nginx/site.com.br/apm.log apm;
```

# Gerando chaves ssh

Eu sempre esqueço.

```
ssh-keygen -t rsa
```

Ler o conteúdo do arquivo:

```
cat ~/.ssh/id_rsa.pub
```

Depois precisa adicionar no arquivo `~/.ssh/authorized_keys` da máquina remota.

Caso apareçam aqueles arquivos chamados **?.pub**, ou apenas **?** na pasta home.

Na verdade, esse não é o nome dos arquivos. O que acontece é que o nome deles é formado por caracteres unicode que não podem ser exibidos no console.

O jeito é descobrir o inode dos arquivos, e excluir através dele.

Listando o inode dos arquivos:

```
ls -li
```

Excluindo através do inode:

```
find . -inum 15681478197 -exec rm -i {} \;
```

Fonte: <https://unix.stackexchange.com/a/230030>

# Docker

Vários comandos do curso de Docker do senai:

acessar o terminal de um container que já está rodando:

```
docker run -it ubuntu
```

iniciar um container e acessar o terminal:

```
docker start -ai 05025384675e
```

iniciar um container sem travar o terminal:

```
docker run -d dockersamples/static-site
```

iniciar um container atribuindo uma porta aleatória:

```
docker run -d -P dockersamples/static-site
```

iniciar um container, mas mapeando uma porta definida com o container. no caso, a porta 8080 é da máquina host, e 80, a do container:

```
docker run -d -p 8080:80 --name site-docker dockersamples/static-site
```

exibindo as portas atreladas ao container:

```
sudo docker port 325b4cbf1b6e
```

atribuindo um nome ao container, para não precisar depender do id:

```
docker run -d -P --name site-docker dockersamples/static-site
```

```
docker stop site-docker
```

parando todos os containers em execução:

```
docker stop $(docker ps -q)
```

examinar um container:

```
docker inspect iddocontainer
```

pode ser usado os primeiros caracteres do id, não precisa ser completo:

```
docker inspect 456b0f275ee1
```

```
docker inspect 456b
```

usando uma pasta específica na hora de montar o volume:

```
docker run -v "/tmp/www:/var/www/" ubuntu
```

neste caso, a pasta **/tmp/www** da máquina host será utilizada como **/var/www** no container

criar uma rede:

```
docker network create --driver bridge minha-rede-docker
```

criar um container nessa rede:

```
docker run -it --name meu-container-1 --network minha-rede-docker ubuntu
```

```
docker run -it --name meu-container-2 --network minha-rede-docker ubuntu
```

agora dá pra acessar um container à partir de outro, usando o hostname ao invés do IP

# Usando ffmpeg

Para comprimir um vídeo:

```
ffmpeg -i input.mp4 -crf 28 output.mp4
```

Esse outro comando especifica o codec de saída como h265:

```
ffmpeg -i input.mp4 -vcodec libx265 -crf 28 output.mp4
```

Para transformar vários arquivos que estão em formato webm (ou qualquer outro formato) para mp4:

```
for file in *.webm; do ffmpeg -i "$file" -vcodec libx265 -crf 28 "${file%.webm}".mp4; done
```

Para apenas converter rapidamente um arquivo em mp4, sem processamento:

```
ffmpeg -i input.mp4 -codec copy output.mp4
```

Para mais opções: <https://askubuntu.com/questions/352920/fastest-way-to-convert-videos-batch-or-single>



Linux

# Bloqueando ip no iptables

Bloqueando uma faixa de ip:

```
iptables -A INPUT -s 35.203.210.0/24 -j DROP
```

Bloqueando apenas um endereço:

```
iptables -A INPUT -s [IP] -j DROP
```

(obs.: ainda não testei esse comando)

Exibindo os ips bloqueados:

```
iptables -L -v
```

Linux

# Comprimir imagens no terminal

Usando o imagemagick:

```
convert imagem.png -resize 80x convertida.png
```

80x é para deixar com 80px de largura

Fonte: <https://opensource.com/article/21/9/resize-image-linux>