

# Códigos

Coisas que provavelmente vou precisar depois (ou não)

- [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](#)

# 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)".

# 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';
```

# 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`
```

# 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.

# 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(/[^\d-]/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
```

# 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"
);
```

# 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.

# jQuery On

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

# 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;
```

# 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
```