

# Como Enviar Emails Através do PHP e do PHPMailer: Guia Completo

O email é uma parte integral de qualquer projeto ou negócio. Enquanto existem diversas plataformas de email – incluindo [Hostinger](#), [Zoho Mail](#) e [G Suite](#) – você também pode enviar um email PHP. Neste tutorial, você vai aprender como enviar emails usando a função incorporação **PHP mail()** e o **PHPMailer** através do **Protocolo de Transferência de Correio Simples (SMTP)**.

Conteúdo

- [Como Usar a Função PHP mail\(\)](#)
- [Como Usar o PHPMailer para Enviar Emails](#)
- [Como Resolver Erros Comuns do PHP Mail e do PHPMailer](#)
- [Conclusão](#)

## Como Usar a Função PHP mail()

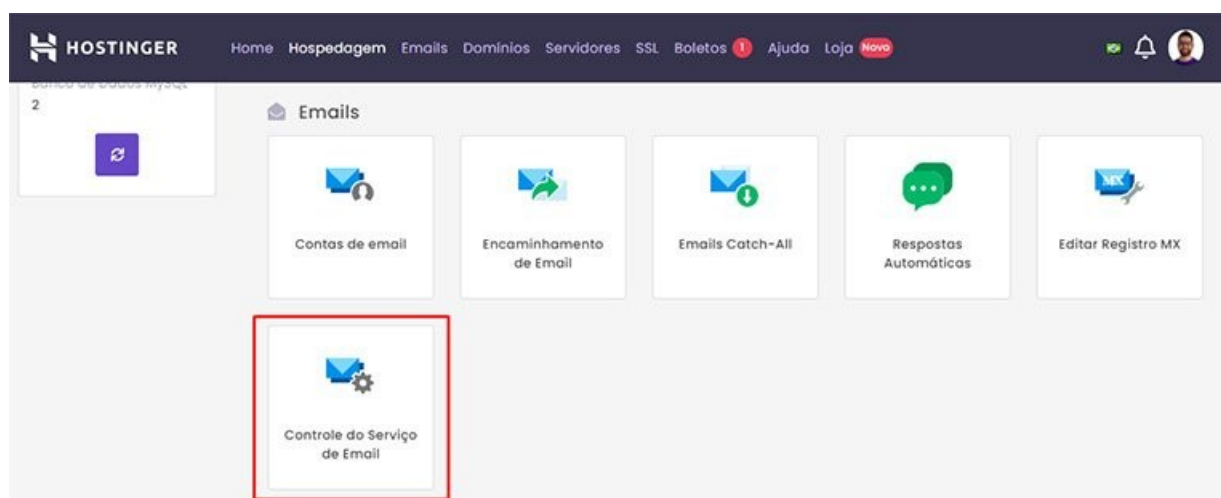
Para usar a função **PHP mail()**, você precisa do programa **Sendmail**, que geralmente é configurado pelo administrador do sistema. É ele que permite o envio de emails.

Na hora de usar essa funcionalidade, garanta que o seu provedor de hospedagem lhe dá a opção de gerenciar o **serviço Sendmail** de modo manual.

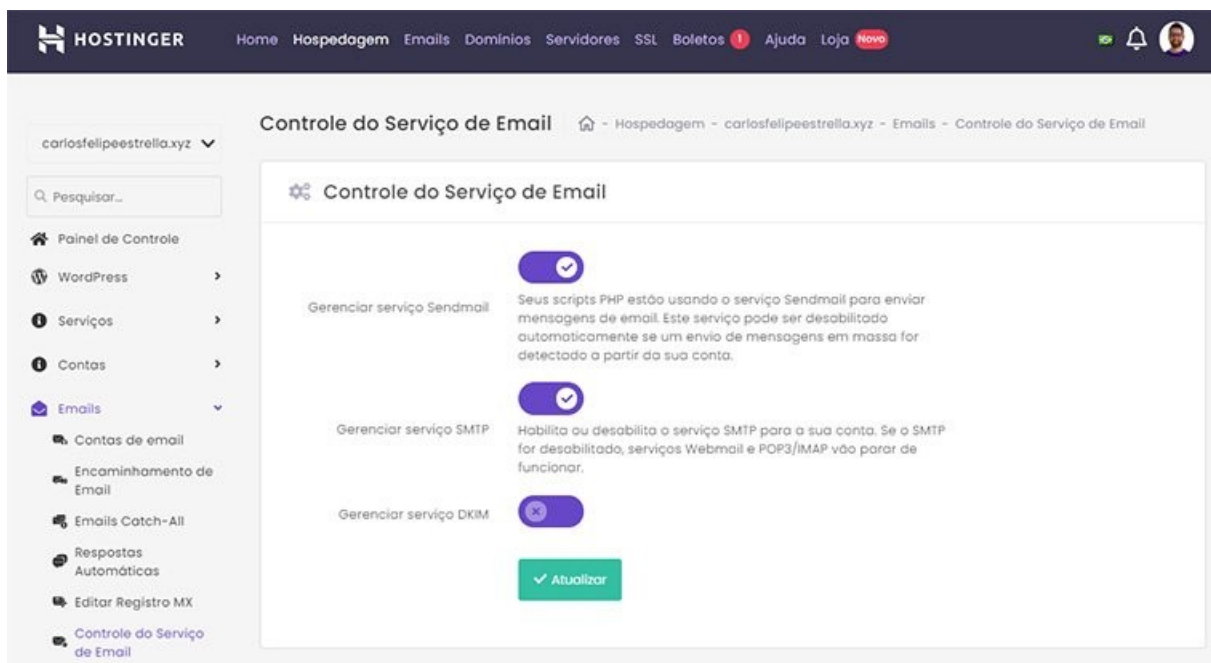
Está com problemas para enviar emails? O **PHP mail()** é ativado por padrão na Hostinger!

[Hospedagem de Site com Desconto](#)

Se você já usa a Hostinger, é possível habilitar ou desabilitar essa função ao acessar o hPanel. Clique em **Emails -> Controle do Serviço de Email**.



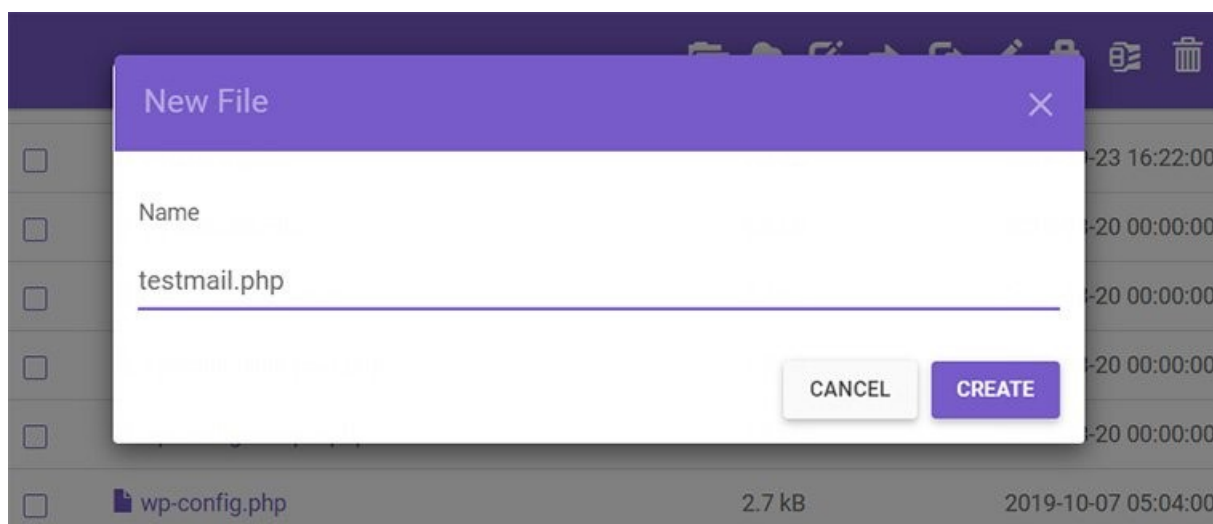
Por padrão, o serviço do Sendmail já fica habilitado. Ainda assim, é recomendável você checar novamente para ter certeza disso.



## Crie um Arquivo de Testes para o PHP Mail

Primeiramente, você precisa criar um arquivo para o PHP mail script e colocá-lo no seu diretório **public\_html** para que ele possa ser acessado facilmente através do seu domínio.

1. Para acessar o hPanel, clique no [Gerenciador de Arquivos](#).
2. Crie um novo arquivo ao clicar em **Add New** (Adicionar Novo). Vamos nomear esse arquivo **testmail.php** e então clicar em **Create** (Criar).



3. Clique duas vezes no **testmail.php**, já que iremos editá-lo usando a função **PHP mail()**. Aqui está um exemplo dos componentes básicos da função que vamos usar neste tutorial. Elas serão explicadas com mais detalhes na próxima seção. Por enquanto, clique em **Save &**

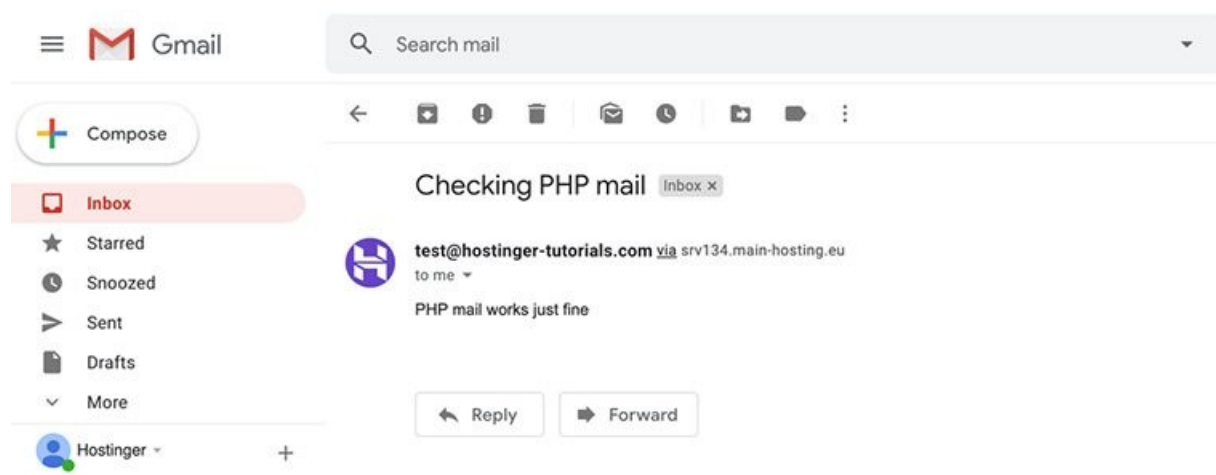
**Close** (Salvar e Fechar) quando tiver terminado suas edições.



The screenshot shows a code editor window titled "Edit file" with a file path of "/public\_html/testmail.php". The code is a PHP script that configures and sends an email. At the bottom right of the editor, there are two buttons: "SAVE & CLOSE" and "SAVE".

```
1 <?php
2 ini_set( 'display_errors', 1 );
3 error_reporting( E_ALL );
4 $from = "test@hostinger-tutorials.com";
5 $to = "test@hostinger.com";
6 $subject = "Checking PHP mail";
7 $message = "PHP mail works just fine";
8 $headers = "From:" . $from;
9 mail($to,$subject,$message, $headers);
10 echo "The email message was sent.";
11 ?>
```

4. Agora você pode enviar um email PHP ao acessar **SeuDominio/testmail.php** a partir do seu navegador. Tenha em mente que você deve mudar **SeuDominio** para o domínio que você usou na criação do testmail.php
5. O endereço para o qual você enviou a mensagem deverá recebê-la sem problemas.



## O Que Significa Cada Componente do PHP Mail

Como mencionamos antes, vamos apresentar os componentes de um script básico de PHP mail. Para este exemplo, nós providenciamos um email de sintaxe básica para ajudar você a compreender essa funcionalidade com maiores detalhes.

Contudo, se você precisa de mais informações sobre a função **Sendmail** e seus componentes, você pode conferir a [documentação PHP oficial](#).

Aqui está a sintaxe do PHP que usamos na seção anterior:

```
<?php
    ini_set( 'display_errors', 1 );
    error_reporting( E_ALL );
    $from = "test@hostinger-tutorials.com";
    $to = "test@hostinger.com";
    $subject = "Checking PHP mail";
    $message = "PHP mail works just fine";
    $headers = "From:" . $from;
    mail($to,$subject,$message, $headers);
    echo "The email message was sent.";
?>
```

Vamos explicar cada passo do script abaixo:

```
ini_set( 'display_errors', 1 );
error_reporting( E_ALL );
```

Essas duas primeiras linhas permitem a criação de relatórios de erros – elas vão contar para você se a execução do script falhar.

```
$from = "test@hostinger-tutorials.com";
```

Essa linha deverá conter o endereço de email que está enviado a mensagem.

A maioria dos provedores não permite que você use um endereço aleatório aqui, já que isso é algo que poderia servir para ataques de [\*spoofing\*](#).

Use um email que foi criado para o seu domínio ou marca para executar o PHP mail de maneira bem-sucedida.

```
$to = "test@gmail.com";
```

Aqui está onde você insere o endereço de email de quem vai receber a mensagem.

```
$subject = "Checking PHP mail";
```

Insira o assunto do email aqui.

```
$message = "PHP mail works just fine";
```

Aqui é onde você pode escrever o corpo da sua mensagem.

```
$headers = "From:" . $from;
```

Especifica informações cruciais, como o endereço de quem enviou, a local para aonde irá a resposta, etc.

```
mail ($to,$subject,$message,$headers);
```

Essa linha é usada para executar a função.

```
echo "The email message was sent.";
```

Essa mensagem vai aparecer caso o script tenha sido executado com sucesso.

# Como Usar o PHPMailer para Enviar Emails

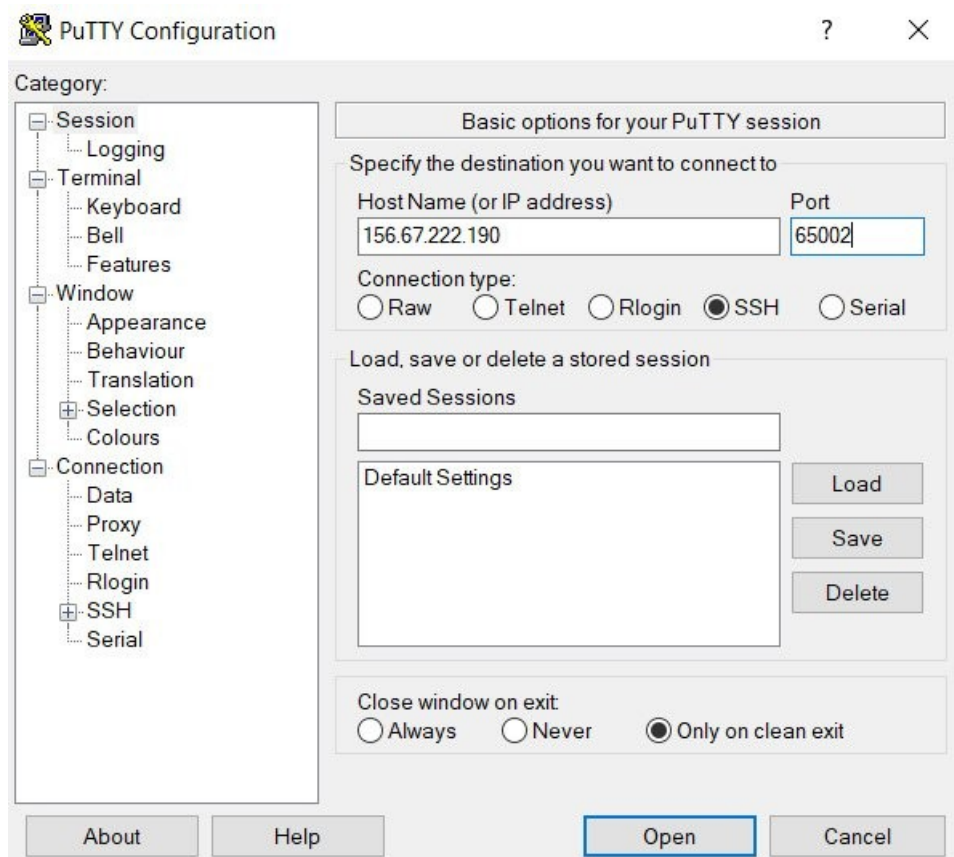
O [PHPMailer](#) é uma popular biblioteca de envio de email PHP. Ela suporta o envio de emails através da função mail() ou do **Protocolo de Transferência de Correio Simples (SMTP)**.

Essa biblioteca simplifica o complicado processo de criar um email PHP ao proporcionar um conjunto de funções para criar e enviar um email.

A instalação do PHPMailer é bem simples, especialmente se você já possui o [Composer](#) instalado. Se você está usando a [Hostinger](#), não é necessário se preocupar com isso, já que a nossa hospedagem já vem com ele pré-instalado em todos os planos.

De qualquer modo, se você precisar instalar o PHPMailer de maneira manual, é necessário se conectar com a sua conta de hospedagem através de um [terminal SSH](#). Siga esses passos:

1. Baixe e instale o [cliente PuTTY SSH](#).
2. Vá até o painel de controle do hPanel, encontre e clique em **Acesso SSH** na seção **Avançado**.
3. Anote suas **Informações de Acesso SSH**. Você vai precisar do seu **SSH IP**, da porta, do nome de usuário e da senha SSH.
4. Abra o PuTTY e então defina o seu SSH IP e sua porta de acordo.



5. Pressione **Abrir** e uma janela de comando vai aparecer. Digite seu **nome de usuário SSH** e sua **senha**. Assim que tiver feito isso, aperte enter
  1. **NOTA:** o PuTTY **NÃO** vai exibir sua senha. Não fique confuso se a sua senha não aparecer na tela.


6. Execute o seguinte comando:

```
cd public_html
```

7. Aperte **Enter**, e então execute este comando:

```
composer require phpmailer/phpmailer
```

8. Aguarde um momento até que o processo de instalação seja finalizado.

 185.224.138.196 - PuTTY

```
login as: u530700449
u530700449@185.224.138.196's password:
-bash-4.2$ cd public_html
-bash-4.2$ composer require phpmailer/phpmailer
Using version ^6.1 for phpmailer/phpmailer
./composer.json has been created
Loading composer repositories with package information
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
 - Installing phpmailer/phpmailer (v6.1.2): Downloading (100%)
phpmailer/phpmailer suggests installing psr/log (For optional PSR-3 debug logging)
phpmailer/phpmailer suggests installing league/oauth2-google (Needed for Google XOAUTH2 authentication)
phpmailer/phpmailer suggests installing hayageek/oauth2-yahoo (Needed for Yahoo XOAUTH2 authentication)
phpmailer/phpmailer suggests installing stevenmaguire/oauth2-microsoft (Needed for Microsoft XOAUTH2 authentication)
phpmailer/phpmailer suggests installing symfony/polyfill-mbstring (To support UTF-8 if the Mbstring PHP extension is not enabled (^1.2))
Writing lock file
Generating autoload files
-bash-4.2$
```

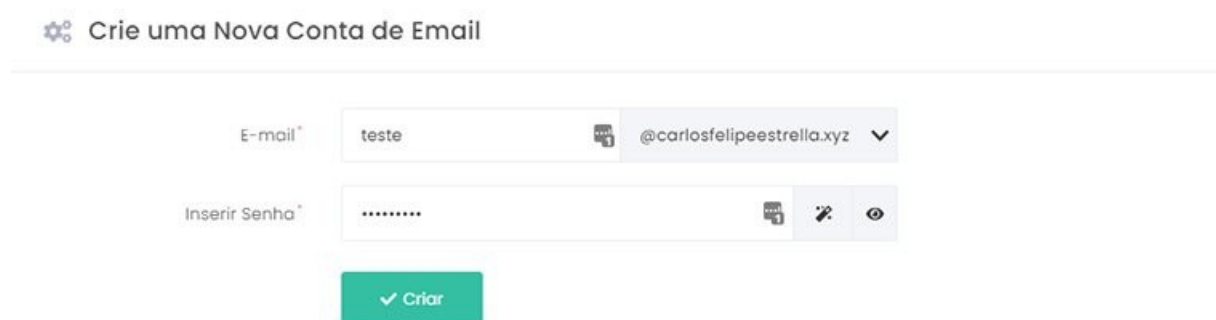
## Como Usar o PHPMailer com o SMTP da Hostinger

Uma vez que o PHPMailer estiver pronto, você pode usá-lo para enviar emails PHP através do SMTP da Hostinger.

1. Crie uma conta de email ao acessar o hPanel e então vá para **Contas de Email -> Criar uma Nova Conta de Email**. Preencha um novo endereço de email e defina uma senha antes



de clicar em **Create**.



- Assim que tiver feito isso, preste atenção nos seus detalhes de SMTP, que estão localizados na mesma página:

i Detalhes do email			
	Hostname	Porta	SSL/TLS, STARTTLS (seguro)
Servidor de entrada (acesso IMAP)	imap.hostinger.com.br	993	✓
Servidor de Saída de Email (SMTP)	smtp.hostinger.com.br	587	✓
Registro MX DNS	5 mx1.hostinger.com.br 10 mx2.hostinger.com.br	-	-
Endereço de email	https://webmail.hostinger.com.br	-	-
Servidor de Entrada (Acesso POP3)	pop.hostinger.com.br	995	✓

**NOTA:** Você deve se lembrar do **nome de usuário** da sua conta de email, da **senha da conta**, do **host SMTP**, e a **porta SMTP** para enviar um email através do PHPMailer.

- Crie um arquivo **testphpmailer.php** na **Public\_html**. Acesse o painel de controle do hPanel e clique em **Gerenciador de Arquivos**.
- Pressione **Add New** (Adicionar Novo). Dê o nome de **testphpmailer.php** para o seu arquivo e clique em **Create** (Criar).
- Clique duas vezes no novo arquivo **testphpmailer.php** e então copie e cole o seguinte:

```
<?php
use PHPMailer\PHPMailer\PHPMailer;
require 'vendor/autoload.php';
$mail = new PHPMailer;
$mail->isSMTP();
$mail->SMTPDebug = 2;
$mail->Host = 'smtp.hostinger.com';
$mail->Port = 587;
$mail->SMTPAuth = true;
$mail->Username = 'test@hostinger-tutorials.com';
$mail->Password = 'SUA SENHA AQUI';
$mail->setFrom('test@hostinger-tutorials.com', 'Your Name');
$mail->addReplyTo('test@hostinger-tutorials.com', 'Your Name');
$mail->addAddress('exemplo@email.com', 'Receiver Name');
$mail->Subject = 'Testing PHPMailer';
$mail->msgHTML(file_get_contents('message.html'), __DIR__);
```

```
$mail->Body = 'This is a plain text message body';
// $mail->addAttachment('test.txt');
if (!$mail->send()) {
    echo 'Mailer Error: ' . $mail->ErrorInfo;
} else {
    echo 'The email message was sent.';
}
?>
```

6. Modifique o código acima do jeito que preferir. Por exemplo, você precisa substituir **SUA SENHA AQUI** com a **senha do seu email** ou **test@hostinger-tutorials.com** com o seu **nome de usuário**, **exemplo@gmail.com** com o endereço de email do destinatário, etc. Assim que tiver feito isso, clique em **Save & Close** (Salvar e Fechar).
7. Seu PHPMailer agora está pronto para ser usado. Execute o script ao inserir **SeuDominio.com/testphpmailer.php** no seu navegador.

## O Que Significa Cada Componente do PHPMailer

Para compreender como o PHPMailer funciona, vamos investigar o exemplo de script acima, que usa SMTP para entrega de emails. Aqui está uma explicação de cada componente:

```
use PHPMailer\PHPMailer\PHPMailer;
```

Essa linha importa a classe do PHPMailer para o *namespace* global.

```
require '../vendor/autoload.php';
```

Então são incluídas as várias bibliotecas que o PHPMailer necessita.

```
$mail->
```

Todas as variáveis similares contêm informações vitais, como detalhes do servidor, cabeçalhos de mensagens, anexos e mais. Em suma, eles garantem que quem está enviando a mensagem está protegido com autenticação SMTP.

```
if (!$mail->send()) {
```

Define o que acontece quando os scripts são executados.

```
echo 'Mailer Error: ' . $mail->ErrorInfo;
```

Vai exibir uma mensagem de erro com uma explicação quando o envio do script falhar.

```
} else {
```

Especifica o que acontece se o script é executado.

```
echo 'The email message was sent!';
```

Se o email foi enviado com sucesso, esta mensagem vai aparecer.

**DICA:** a linha **SMTPDebug = 2;** só é útil quando você testa um script e quer ver como ele funciona. Você precisa mudá-la para **SMTPDebug=0;** se você já terminou de fazer os seus testes. Isso é feito para evitar que o usuário final veja o relatório de entrega do SMTP.



Se você prestou atenção, vai perceber que estamos fazendo algo diferente em comparação com o primeiro exemplo – estamos enviando uma [mensagem HTML](#) ao invés de um texto simples.

Portanto, a sua mensagem vai carregar o seu conteúdo a partir do arquivo **message.html**, que está localizado no mesmo diretório – **public\_html**.

Este formato traz uma maior funcionalidade, se comparado com mensagens simples de texto, já que o HTML é altamente customizável. Você pode mudar a cor, o estilo, a imagem ou até incluir arquivos multimídia que geralmente são obsoletos num email de texto simples.

## Formulário de Contato via PHPMailer

Você pode usar o PHPMailer para mais do que apenas enviar um simples email PHP. Uma maneira que você pode utilizá-lo é criando um formulário de contato através do qual o seu público-alvo pode entrar em contato com você.

Aqui está um exemplo do script:

```
<?php
use PHPMailer\PHPMailer\PHPMailer;
require 'vendor/autoload.php';
$mail = new PHPMailer;
$mail->isSMTP();
$mail->Host = 'smtp.hostinger.com';
$mail->Port = 587;
$mail->SMTPAuth = true;
$mail->Username = 'test@hostinger-tutorials.com';
$mail->Password = 'EMAIL_ACCOUNT_PASSWORD';
$mail->setFrom('test@hostinger-tutorials.com', 'Mr. Drago');
$mail->addAddress('exemplo@gmail.com', 'Nome do Destinatário');
if ($mail->addReplyTo($_POST['email'], $_POST['name'])) {
    $mail->Subject = 'PHPMailer contact form';
    $mail->isHTML(false);
    $mail->Body = <<<EOT
Email: {$_POST['email']}
Name: {$_POST['name']}
Message: {$_POST['message']}
EOT;
    if (!$mail->send()) {
        $msg = 'Sorry, something went wrong. Please try again later.';
    } else {
        $msg = 'Message sent! Thanks for contacting us.';
    }
} else {
    $msg = 'Share it with us!';
}
?>
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <title>Contact form</title>
</head>
<body>
<h1>Do You Have Anything in Mind?</h1>
<?php if (!empty($msg)) {
    echo "<h2>$msg</h2>";
} ?>
<form method="POST">
    <label for="name">Name: <input type="text" name="name"
id="name"></label><br><br>
```

```

<label for="email">Email: <input type="email" name="email"
id="email"></label><br><br>
<label for="message">Message: <textarea name="message" id="message" rows="8"
cols="20"></textarea></label><br><br>
<input type="submit" value="Send">
</form>
</body>
</html>

```

Assim como os scripts anteriores, você precisa criar um novo arquivo na pasta **public\_html**. Neste caso, nós demos o nome de **formscript.php** ao arquivo.

Edite a informação dentro do script de acordo com as suas necessidades. Depois disso, você só precisa rodar o script a partir do seu navegador.

Aqui está como o fica o resultado final:

The screenshot shows a web browser window with the title 'Contact form'. The address bar indicates the URL is 'hostinger-dev-11.xyz/formscript.php'. The main heading of the page is 'Você tem algo em mente?' (Do you have something in mind?). Below the heading is the subheading 'Compartilhe Conosco!' (Share with us!). The form consists of three input fields: 'Nome:' (Name), 'Email:', and 'Mensagem:' (Message). A 'Send' button is located at the bottom left of the form area.

Assim que o cliente enviar um email PHP, ele vai receber um aviso de confirmação, e o conteúdo será enviado para a caixa de entrada do email que você inseriu aqui:

```
$mail->addAddress('exemplo@gmail.com', 'Nome do Destinatário');
```

**DICA:** Caso o formulário de contato do PHPMailer não funcione, adicione a linha **\$mail->SMTPDebug = 2;** para ver o que está causando esse problema. Não se esqueça de apagá-la ou de mudar o **2** para **0** quando tiver terminado.

O PHPMailer oferece [mais exemplos](#) que você pode experimentar no repositório oficial deles no GitHub. Além disso, se você está usando o WordPress, pode facilmente [criar um formulário de contato](#) para lhe ajudar com a ajuda de plugins como [WP Forms](#), [Formidable Forms](#) ou [Gravity Forms](#).



## Como Resolver Erros Comuns do PHP Mail e do PHPMailer

Você pode encontrar erros ocasionalmente quando usa o PHP Mail ou o PHPMailer. Aqui está uma lista com os erros mais comuns e os caminhos para corrigi-los.

### Endereço de Envio Rejeitado: Não é de Propriedade do Usuário

Este erro significa que o servidor não conseguiu promover a autenticação usando as informações que você providenciou. Ele pode aparecer em inglês, como “Sender Address Rejected: Not Owned by the User”.

Para corrigi-lo, confira o endereço que você usou para enviar um email PHP e garanta que ele corresponde a uma caixa de email já existente.

Se ele está apontando para a caixa de entrada errada, altere-a de acordo com as informações corretas. Também cheque se você habilitou os [registros SPF](#).

### Gmail Não Conseguiu Verificar que SeuDominio.com Enviou Esta Mensagem

Se você ver esse aviso quando está testando um script de PHP Mail, ele pode significar uma dessas duas coisas:

- Seu registro SPF não está habilitado. Confira os passos para como habilitá-lo [aqui](#).
- Você enviou um email a partir de um endereço que não existe ou não pertence a você. Nesse caso, garanta que você vai usar informações de autenticação SMTP válidas.

Lembrando que esse erro aparecerá como “Gmail Couldn’t Verify That YourDomain.com Sent This Message” em páginas em inglês.

### Email vai para a Pasta de Spam

Existem diversos motivos que podem levar um PHP mail a aparecer como spam. Aqui estão os mais comuns:

- Assunto enganoso ou que lembra spam. Alguns exemplos disso seriam palavras como “teste” “testando”, “alô”, “urgente” ou algo do tipo. Garanta que você colocou uma intenção clara no assunto da sua mensagem.

- Você está usando o endereço de envio incorreto, o que faz com que as medidas de segurança filtrem o seu email. Isso é feito para evitar *spoofing* e golpes via email.
- Você está usando palavras de gatilho de spam. Essa categoria inclui frases como “**ótima oferta**“, “**clique aqui**“, “**promoção especial**“, “**isto não é spam**” e assim em diante. Tente modificar o conteúdo da sua mensagem para ver se esse é o caso.
- Sua lista de *mailing* não possui um botão de **cancelar inscrição**. Quando muitas pessoas reportam o seu email como spam, vai ser difícil de evitar o filtro de spam. Ter um botão para cancelar a inscrição é uma ótima maneira de evitar isso fazer com que os seus leitores confiem mais em você.

## Conclusão

Parabéns, agora você se familiarizou com o recurso de **enviar email por PHP** e aprendeu a usar o **PHPMailer** para enviar emails com autenticação SMTP. Apesar deste tutorial proporcionar exemplos básicos, a mesma sintaxe pode ser usada para desenvolver um formulário de contato ou outras extensões para o seu site.

Para uma informação mais aprofundada, não se esqueça de conferir a página do [projeto PHPMailer](#). Se você tem dicas, truques ou ideias para compartilhar, estamos ansiosos para ouvi-las na seção de comentários abaixo.

<https://www.hostinger.com.br/tutoriais/como-enviar-emails-usando-php>

# Enviando email com PHP

**Neste artigo iremos ver como criar um formulário para contato, e como enviar as informações do usuário por e-mail através do PHP.**

- 
- Suporte ao aluno
- Anotações
- Favoritar
- Concluído
- 29 Gostei
- 
- 
- 29
- 

[Artigos PHP](#) Enviando email com PHP

Olá pessoal, nesse tutorial estarei ensinando como criar um formulário para contato, e como enviar as informações do usuário por e-mail através do próprio PHP.

Vamos começar fazendo o código HTML do formulário, nele vamos colocar os campos de formulário para depois vermos como estilizamos ele com CSS.

 Saiba mais sobre o [comandos básicos do HTML](#)

## Criando código de envio de e-mail

```
<html>
<head>
  <title>Enviando E-mail com PHP - DevMedia</title>
  <link rel="stylesheet" type="text/css" href="estilo.css">
</head>
<body>
  <div id="contato_form">
    <form action="enviar.php" name="form_contato" method="post" >
      <p class="titulo">Formulário <small class="asteristico">*Campos
Obrigatorios</small></p>
      <table align="center">
        <tr>
          <td>Nome:<sup class="asteristico">*</sup></td>
          <td>
            <input type="text" name="nome" maxlength="40" />
          </td>
        </tr>
        <tr>
          <td>E-mail:<sup class="asteristico">*</sup></td>
          <td>
            <input type="email" name="email" maxlength="30" />
          </td>
        </tr>
      </table>
    </form>
  </div>
</body>
</html>
```

```

<tr>
  <td>Telefone:<sup class="asteristico">*</sup></td>
  <td>
    <input type="text" name="telefone" maxlength="14" />
  </td>
</tr>
<tr>
  <td>Opções:<sup class="asteristico">*</sup></td>
  <td>
    <select name="escolhas" class="campo_input">
      <option value="Opção 1">Opção 1</option>
      <option value="Opção 2">Opção 2</option>
    </select>
  </td>
</tr>
<tr>
  <td>Mensagem:<sup class="asteristico">*</sup></td>
  <td>
    <textarea name="msg" cols="16" rows="5"></textarea>
  </td>
</tr>
<tr align="right">
  <td colspan="2">
    <input type="reset" class="campo_submit" value="Limpar" />
    <input type="submit" class="campo_submit" value="Enviar" />
  </td>
</tr>
<tr>
  <td colspan="2" align="right"><small class="asteristico">* Campos
obrigatorios</small></td>
</tr>
</table>
</form>
</div>
</body>
</html>

```

### Listagem 1. Código do formulário de contato

## Estilizando formulário

Agora vamos criar o código css que será responsável por estilizar o formulário. Vamos salvar

```

<style type="text/css">
body{
  font-size:12px;
  font-family:Verdana, Geneva, sans-serif;
}
#contato_form{
  width:500px;
  min-height:175px;
  color:#999;
  margin:auto;
}
.asteristico{
  color:#F00;
}
</style>

```

### Listagem 2. : Código CSS de estilização do formulário

Vamos entender um pouco o código acima?

Na tag table não podemos nos esquecer de colocar o action=""(que é onde colocamos o arquivo em php com o código de envio do e-mail) e o method="" que no nosso caso iremos usar o POST, pois não queremos passar os parâmetros por URL(GET).

## Criando o código PHP de envio de email

Logo a seguir seguem as variáveis que irão armazenar os dados passados anteriormente pelo usuário.

```
<?php
//Variáveis

$nome = $_POST['nome'];
$email = $_POST['email'];
$telefone = $_POST['telefone'];
$sopcoes = $_POST['escolhas'];
$mensagem = $_POST['msg'];
$data_envio = date('d/m/Y');
$hora_envio = date('H:i:s');
```

### Listagem 3. Variáveis passadas pelo formulário

A seguir veremos como fazer a configuração de exibição do corpo do e-mail.

```
// Compo E-mail
$arquivo = "
<style type='text/css'>
body {
margin:0px;
font-family:Verdane;
font-size:12px;
color: #666666;
}
a{
color: #666666;
text-decoration: none;
}
a:hover {
color: #FF0000;
text-decoration: none;
}
</style>
<html>
<table width='510' border='1' cellpadding='1' cellspacing='1'
bgcolor='#CCCCCC'>
<tr>
<td>
<tr>
<td width='500'>Nome:$nome</td>
</tr>
<tr>
<td width='320'>E-mail:<b>$email</b></td>
</tr>
<tr>
<td width='320'>Telefone:<b>$telefone</b></td>
</tr>
<tr>
<td width='320'>Opções:$escolhas</td>
</tr>
<tr>
<td width='320'>Mensagem:$nome</td>
</tr>
```



```

        </td>
    </tr>
    <tr>
        <td>Este e-mail foi enviado em <b>$data_envio</b> às
<b>$hora_envio</b></td>
    </tr>
</table>
</html>
";

```

#### Listagem 4. Configurando exibição do corpo do e-mail

A seguir estarão os comandos para que ele interprete as variáveis e com isso possamos enviar o e-mail.

```

//enviar

// emails para quem será enviado o formulário
$emailenviar = "seuemail@seudominio.com.br";
$destino = $emailenviar;
$assunto = "Contato pelo Site";

// É necessário indicar que o formato do e-mail é html
$headers = 'MIME-Version: 1.0' . "\r\n";
    $headers .= 'Content-type: text/html; charset=iso-8859-1' . "\r\n";
    $headers .= 'From: $nome <$email>';
//$headers .= "Bcc: $EmailPadrao\r\n";

$enviaremail = mail($destino, $assunto, $arquivo, $headers);
if($enviaremail){
    $msgm = "E-MAIL ENVIADO COM SUCESSO! <br> O link será enviado para o e-mail
fornecido no formulário";
    echo " <meta http-equiv='refresh' content='10;URL=contato.php'>";
} else {
    $msgm = "ERRO AO ENVIAR E-MAIL!";
    echo "";
}
?>

```

#### Listagem 5. Interpretando as variáveis e enviando o e-mail

Dessa forma podemos criar qualquer formulário e enviar os seus dados por e-mail para quem quisermos.

Nesse exemplo, ele estará enviando para o e-mail que você colocar na variável ‘\$emailenviar’.

Fico por aqui nesse tutorial, espero que tenham gostado e até o próximo.

<https://www.devmedia.com.br/enviando-email-com-php/37216>

# Como enviar e-mails com PHP Mailer



## Como enviar e-mails com PHP Mailer

→ <https://github.com/PHPMailer/PHPMailer>

Conheça a classe PHP Mailer, conhecida por ser a classe mais popular de envio de e-mails feita em PHP. Neste tutorial aprenda como instalar e enviar e-mails usando uma conta [@gmail.com](#).

---

## Instalação

Você pode instalar a classe PHP Mailer de duas maneiras:

- Via composer
- Baixando e incorporando os arquivos em seu projeto.

### → Composer

A maneira de instalação recomendada pelos desenvolvedores é via composer. Se você já tem um projeto que utiliza composer você deve adicionar a seguinte linha em seu `composer.json`:

```
"phpmailer/phpmailer": "~6.0"
```

Caso você ainda não utilize composer no seu projeto ou esteja começando um do zero, você pode instalar com o seguinte código:

```
composer require phpmailer/phpmailer
```

### → Copiando arquivos

Como alternativa, se você não estiver usando o Composer, copie o conteúdo da pasta PHPMailer em um dos diretórios do seu projeto e carregue cada arquivo de classe manualmente:

```
<?php use PHPMailer\PHPMailer\PHPMailer; use PHPMailer\PHPMailer\Exception; require
'path/to/PHPMailer/src/Exception.php'; require
'path/to/PHPMailer/src/PHPMailer.php'; require 'path/to/PHPMailer/src/SMTP.php';
```

---

## Utilização

### Autorizar conta Google

Este tutorial irá utilizar uma conta [@gmail.com](#) para demonstrar o envio de e-mails, se você possuir um servidor `smtp` configurado pule este passo.

Para enviar e-mails utilizando uma conta `@gmail.com` você deve autorizar sua conta para ser acessada por apps menos seguras, através desse link: <https://myaccount.google.com/lesssecureapps>.

Veja abaixo o código completo de um envio simples de mensagem utilizando o PHP Mailer:

```
<?php // Importar as classes use PHPMailer\PHPMailer\PHPMailer; use PHPMailer\
PHPMailer\Exception; // Carregar o autoloader do composer require
'vendor/autoload.php'; // Instância da classe $mail = new
PHPMailer(true); try { // Configurações do servidor $mail-
>isSMTP(); // Define o uso de SMTP no envio $mail->SMTPAuth = true;
// Habilita a autenticação SMTP $mail->Username = 'nomedaconta@gmail.com';
$mail->Password = 'senha'; // Criptografia do envio SSL também é aceito
$mail->SMTPSecure = 'tls'; // Informações especificadas pelo Google $mail-
>Host = 'smtp.gmail.com'; $mail->Port = 587; // Define o remetente
$mail->setFrom('nomedaconta@gmail.com', 'Nome do Remetente'); // Define o
destinatário $mail->addAddress('destinatario@email.com',
'Destinatário'); // Conteúdo da mensagem $mail->isHTML(true); // Seta o
formato do e-mail para aceitar conteúdo HTML $mail->Subject = 'Assunto';
$mail->Body = 'Este é o corpo da mensagem <b>Olá em negrito!</b>'; $mail-
>AltBody = 'Este é o corpo da mensagem para clientes de e-mail que não
reconhecem HTML'; // Enviar $mail->send(); echo 'A mensagem foi
```

```
enviada!';}catch (Exception $e){    echo "Message could not be sent. Mailer  
Error: {$mail->ErrorInfo}";}
```

O exemplo acima é a demonstração básica de uso, a classe PHP Mailer conta com diversas outras funcionalidades como:

- Anexo de arquivos
- Cópias
- Cópias oculta
- Debug
- Etc

Para estudar e descobri mais funcionalidades de incrível classe PHP acesse o repositório oficial em:  
<https://github.com/PHPMailer/PHPMailer>.

<https://php.com.br/51?como-enviar-e-mails-com-php-mailer>

# Envie emails via PHP e SMTP utilizando o GMail ou o Google Apps

Por: GN10

Uma das maneiras mais seguras de enviar emails é via SMTP e ao utilizar o GMail / Google Apps você minimiza os riscos de um servidor instável não entregar sua mensagem e também assegura velocidade de envio.

Para criar um formulário de contato em seu site siga este pequeno tutorial

## 1. Faça o Download do PHPMailer:

<http://code.google.com/a/apache-extras.org/p/phpmailer/downloads/list>

## 2. Extraia os arquivos do PHPMailer:

Após extrair os arquivos renomeie a pasta para *phpmailer*, assim facilitamos a taxonomia dos caminhos até os arquivos.

## 3. Crie um formulário HTML. Abaixo um exemplo com alguns campos:

```
<form action="email.php" method="post">
    <label for="Nome">Nome:</label>
    <input type="text" name="Nome" size="35" />

    <label for="Email">E-mail:</label>
    <input type="text" name="Email" size="35" />

    <label for="Fone">Telefone:</label>
    <input type="text" name="Fone" size="35" />

    <label for="Mensagem">Mensagem:</label>
    <textarea name="Mensagem" rows="8" cols="40"></textarea>

    <input type="submit" name="Enviar" value="Enviar" />
</form>
```

## 4. Crie um arquivo chamado *email.php* com o código abaixo:

```
<?php
$Nome          = $_POST["Nome"];           // Pega o valor do campo Nome
$Fone          = $_POST["Fone"];           // Pega o valor do campo Telefone
$Email         = $_POST["Email"];          // Pega o valor do campo Email
$Mensagem      = $_POST["Mensagem"];       // Pega os valores do campo Mensagem

// Variável que junta os valores acima e monta o corpo do email

$Vai           = "Nome: $Nome\n\nE-mail: $Email\n\nTelefone: $Fone\n\nMensagem:
$Mensagem\n";

require_once("phpmailer/class.phpmailer.php");

define('GUSER', 'enviador@gmail.com');     // <-- Insira aqui o seu GMail
define('GPWD', 'senha');                   // <-- Insira aqui a senha do seu GMail

function smtpmailer($para, $de, $de_nome, $assunto, $corpo) {
    global $error;
    $mail = new PHPMailer();
    $mail->IsSMTP();                        // Ativar SMTP
```

```

        $mail->SMTPDebug = 0;                // Debugar: 1 = erros e mensagens, 2 =
mensagens apenas
        $mail->SMTPAuth = true;              // Autenticação ativada
        $mail->SMTPSecure = 'ssl';           // SSL REQUERIDO pelo GMail
        $mail->Host = 'smtp.gmail.com';      // SMTP utilizado
        $mail->Port = 587;                   // A porta 587 deverá estar aberta em
seu servidor
        $mail->Username = GUSER;
        $mail->Password = GPWD;
        $mail->SetFrom($de, $de_nome);
        $mail->Subject = $assunto;
        $mail->Body = $corpo;
        $mail->AddAddress($para);
        if(!$mail->Send()) {
            $error = 'Mail error: '.$mail->ErrorInfo;
            return false;
        } else {
            $error = 'Mensagem enviada!';
            return true;
        }
    }

// Insira abaixo o email que irá receber a mensagem, o email que irá enviar (o
mesmo da variável GUSER),
o nome do email que envia a mensagem, o Assunto da mensagem e por último a
variável com o corpo do email.

    if (smtpmailer('recebedor@dominio.com.br', 'enviador@gmail.com', 'Nome do
Enviador', 'Assunto do Email', $Vai)) {

        Header("location:http://www.dominio.com.br/obrigado.html"); //
Redireciona para uma página de obrigado.

    }
if (!empty($error)) echo $error;
?>

```

Pronto, agora salve o arquivo com o formulário HTML e o enviar.php

Lembre-se que estes 2 arquivos deverão estar **fora** da pasta **phpmailer** no exemplo acima, mas caso deseje você pode alterar os nomes dos arquivos e pastas, apenas atente-se para estas duas partes do código:

```
<form action="email.php" method="post">
```

e

```
require_once("phpmailer/class.phpmailer.php");
```

Caso você utilize WordPress ou Magento, utilize os plugins abaixo:

WordPress » [WP Mail SMTP](#)

Magento » [ASchroder.com SMTP Pro Email](#)

<https://www.gn10.com.br/blog/dicas/envie-emails-php-smtp-gmail-google-apps/>