

Programa de Formação Web Developers PHP

Agosto/2021



CONSTRUSITE BRASIL
O SEU SUCESSO ONLINE

GALAXPAY 
SEU NEGÓCIO RECORRENTE



Aula 11

01

Composer

...

02

Disparo de emails,
consumindo libs

...

03

Exclusão usando
SweetAlert

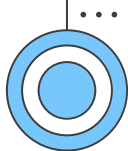
...



GALAXPAY\$
SEU NEGÓCIO RECORRENTE



Composer



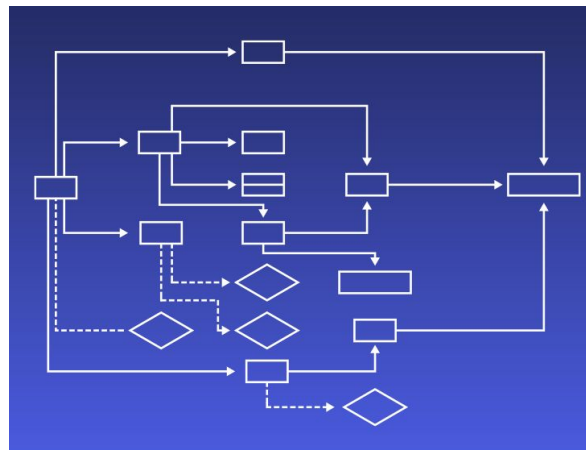
Composer

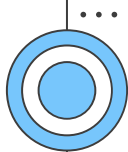
- Composer é um **gerenciador de pacotes** para o PHP.
- Facilita acesso a bibliotecas de código de terceiros, tornando o desenvolvimento do projeto mais simples e permitindo com que o desenvolvedor foque apenas em seu código, não no uso de código de terceiros.



Mas o que é um Gerenciador de Dependências?

- É um gerenciador de dependências =P
- Um gerenciador de dependência nada mais é que um **facilitador** para instalação, remoção e atualização de pacotes externos em projetos.
- Evitando “copiar-colar” de código, gerando um sistema mais fácil de ser “bugado”.

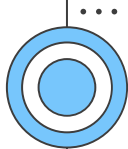




Como funciona o Composer?

- Composer permite que o desenvolvedor determine quais dependências são necessárias para um projeto funcionar.
- A partir daí, é ele quem faz o “gerenciamento” destas dependências, realizando a instalação, atualização e remoção, **removendo** essa responsabilidade do desenvolvedor.





Ao código

- No terminal, acesse a pasta do seu projeto e digite: **composer init**
- Seguir os famosos next -> next

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL

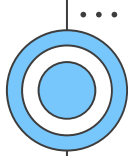
PS C:\xampp\htdocs\aula-11> composer init

Welcome to the Composer config generator

This command will guide you through creating your composer.json config.

Package name (<vendor>/<name>) [gabriel/aula-11]:
```





Ao código

```
Minimum Stability []:  
Package Type (e.g. library, project, metapackage, composer-plugin) []:  
License []:
```

Define your dependencies.

Would you like to define your dependencies (require) interactively [yes]?

Search for a package:

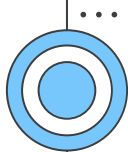
Would you like to define your dev dependencies (require-dev) interactively [yes]?

Search for a package:

Add PSR-4 autoload mapping? Maps namespace "Izabelladev\Aula11" to the entered relative path. [src/, n to skip]:

```
{  
  "name": "izabelladev/aula-11",  
  "autoload": {  
    "psr-4": {  
      "Izabelladev\\Aula11\\": "src/"  
    }  
  },  
  "authors": [  
    {  
      "name": "Izabella Oliveira",  
      "email": "izabella.dev@construsitebrasil.com.br"  
    }  
  ],  
  "require": {}  
}
```

Do you confirm generation [yes]? ☒



Ao código

The screenshot shows the Visual Studio Code editor interface. The Explorer panel on the left shows the project structure with folders like `base-php`, `code`, and `aula-11`. The `composer.json` file is selected. The Outline panel shows the structure of the `autoload` section. The main editor displays the contents of `composer.json`, which is a JSON file defining the project's autoloading configuration. The terminal at the bottom shows the command `composer require` being executed, and the output indicates that the autoloading files have been generated successfully.

```
base-php > code > aula-11 > {} composer.json > {} require
1  {
2      "name": "izabelladev/aula-11",
3      "autoload": {
4          "psr-4": {
5              "Izabelladev\\Aula11\\": "src/"
6          }
7      },
8      "authors": [
9          {
10             "name": "Izabella Oliveira",
11             "email": "izabella.dev@construsitebrasil.com.br"
12         }
13     ],
14     "require": {}
15 }
16
```

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

```
bash + v [ ] [ ] ^ x
Generating autoload files
Generated autoload files
PSR-4 autoloading configured. Use "namespace Izabelladev\\Aula11;" in src/
Include the Composer autoloader with: require 'vendor/autoload.php';
izabelladev@inovacao01 ~/base-php/code/aula-11 (main) $
```

Ln 14, Col 18 Spaces: 4 UTF-8 LF JSON

Pacotes

- Maior site de pacotes PHP.
- <https://packagist.org/>
- Acesse o site, e pesquise por **phpmailer**
- Vamos usar o **phpmailer/phpmailer**

```
],  
"require": {  
    "phpmailer/phpmailer": "^6.5"  
}
```

Installation & loading

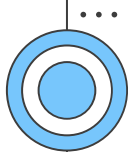
PHPMailer is available on **Packagist** (using seu `composer.json` file:

```
"phpmailer/phpmailer": "^6.5"
```

or run

```
composer require phpmailer/phpmailer
```





Instalando

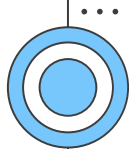
- Rodar o comando **composer install**
- Ele irá baixar, atualizar ou remover seu pacote.
- E que se faça mágica...

```
composer.json
AULA-11
  vendor
    composer.json
    composer.lock
    config.php
    index.php

"authors": [
  {
    "name": "Gabriel Henrique",
    "email": "46637713+gabrielhenriqsp@users.noreply.github.com"
  }
],
"require": {
  "phpmailer/phpmailer": "^6.5"
}
```

Composer 2. See <https://blog.packagist.com/deprecating-composer-1-support/>
Updating dependencies (including require-dev)
Package operations: 1 install, 0 updates, 0 removals
- Installing phpmailer/phpmailer (v6.5.1): Downloading (100%)
phpmailer/phpmailer suggests installing hayageek/oauth2-yahoo (Needed for Yahoo XOAUTH2 au
phpmailer/phpmailer suggests installing league/oauth2-google (Needed for Google XOAUTH2 au
phpmailer/phpmailer suggests installing psr/log (For optional PSR-3 debug logging)
phpmailer/phpmailer suggests installing stevenmaguire/oauth2-microsoft (Needed for Microso
phpmailer/phpmailer suggests installing symfony/polyfill-mbstring (To support UTF-8 if the
Writing lock file
Generating autoload files
1 package you are using is looking for funding.
Use the `composer fund` command to find out more!
PS C:\xampp\htdocs\aula-11> |



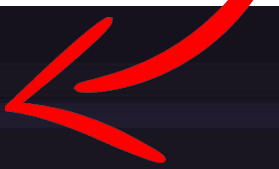


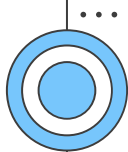
Utilizando

- É necessário incluir o arquivo **vendor/autoload.php**;

```
{ } composer.json  index.php x
base-php > code > aula-11 > index.php
1  <?php
2
3  require 'vendor/autoload.php';
4
```

```
index.php  EmailSend.php 4  autoload.php x
vendor > autoload.php
1  <?php
2
3  // autoload.php @generated by Composer
4
5  require_once __DIR__ . '/composer/autoload_real.php';
6
7  return ComposerAutoloaderInit0e0286f511612d9a1e63e2b1258bc8f8::getLoader();
8
```





Utilizando

- Geralmente dentro do packagist tem um exemplo da classe que estamos consumindo.

README



A Simple Example

```
<?php
//Import PHPMailer classes into the global namespace
//These must be at the top of your script, not inside a function
use PHPMailer\PHPMailer\PHPMailer;
use PHPMailer\PHPMailer\SMTP;
use PHPMailer\PHPMailer\Exception;

//Load Composer's autoloader
require 'vendor/autoload.php';

//Create an instance; passing 'true' enables exceptions
$mail = new PHPMailer(true);

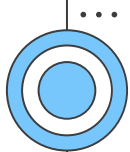
try {
    //Server settings
    $mail->SMTPDebug = SMTP::DEBUG_SERVER;
    $mail->isSMTP();
    $mail->Host      = 'smtp.example.com';
    $mail->SMTPAuth  = true;
    $mail->Username  = 'user@example.com';
    $mail->Password  = 'secret';
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;
    $mail->Port      = 465;

    //Recipients
    $mail->setFrom('from@example.com', 'Mailer');
    $mail->addAddress('joe@example.net', 'Joe User'); //Add
    $mail->addAddress('ellen@example.com'); //Name
    $mail->addReplyTo('info@example.com', 'Information');
    $mail->addCC('cc@example.com');
    $mail->addBCC('bcc@example.com');

    //Attachments
    $mail->addAttachment('/var/tmp/file.tar.gz'); //Add
```



GALAXYPAY
SEU NEGÓCIO RECORRENTE

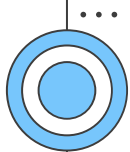


Vamos a implementação

- Criar uma classe EmailSend.php
- Método send, e dentro dele montar seu método.

```
index.php  EmailSend.php  composer.json  config.php
Classes > EmailSend.php > EmailSend > send
6
7  class EmailSend{
8      public function send(){
9
10         $mail = new PHPMailer(true);
11
12         try {
13             //Server settings
14             $mail->SMTPDebug = SMTP::DEBUG_SERVER;           //Enable verbose error output
15             $mail->isSMTP();                                   //Send using SMTP
16             $mail->Host       = 'smtp.example.com';           //Set the SMTP host name
17             $mail->SMTPAuth   = true;                         //Enable SMTP authentication
18             $mail->Username   = 'user@example.com';           //SMTP username
19             $mail->Password   = 'secret';                     //SMTP password
20             $mail->SMTPSecure = PHPMailer::ENCRYPTION_SMTPS;   //Enable encryption
21             $mail->Port       = 465;                           //TCP port number
22
23             //Recipients
24             $mail->setFrom('from@example.com', 'Mailer');
25             $mail->addAddress('joe@example.net', 'Joe User'); //Add a recipient
26             $mail->addReplyTo('info@example.com', 'Information');
27             $mail->addCC('cc@example.com');
28
29             //Content
30             $mail->isHTML(true);                               //Set email format to HTML
31             $mail->Subject = 'Here is the subject';
32             $mail->Body    = 'This is the HTML message body <b>in bold!</b>';
33             $mail->AltBody = 'This is the body in plain text for non-HTML mail clients';
34
35             $mail->send();
36             echo 'Message has been sent';
37         } catch (Exception $e) {
38             echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";
39         }
40     }
41 }
42
```





Vamos a implementação

- Fez? Rodou no navegador?
- Deu certo?

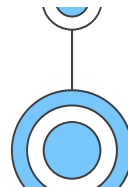
← → ↻ ⚠ Não seguro | http://192.168.2.101/Gabriel/aula-11/

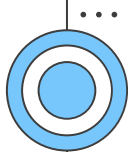
Apps Intranet Const... Informações d... Logins autom... Procedimento... Tickets Autom... w³ Icons from Fo

2021-09-28 13:16:57 SMTP ERROR: Failed to connect to server: php_network_getaddresses: getaddrinfo failed: Name or service not known (0)
SMTP connect() failed. <https://github.com/PHPMailer/PHPMailer/wiki/Troubleshooting>
Message could not be sent. Mailer Error: SMTP connect() failed. <https://github.com/PHPMailer/PHPMailer/wiki/Troubleshooting>NULL



GALAXPAY
SEU NEGÓCIO RECORRENTE

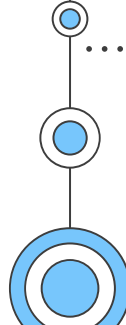
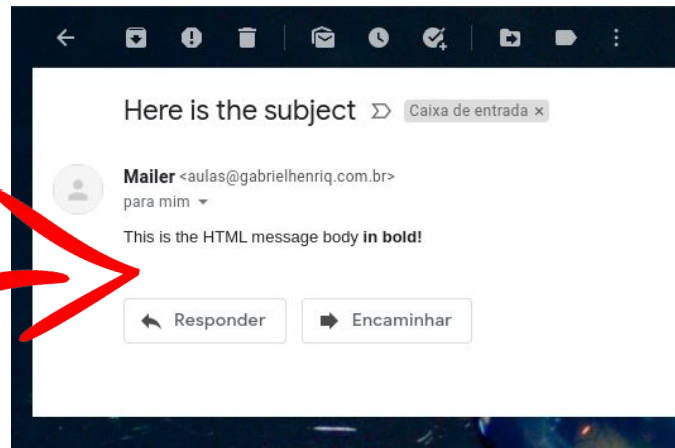


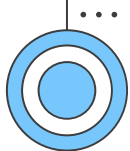


Corrigindo...

- Dados para autenticar:
- Servidor: mail.gabrielhenriq.com.br
- Email: aulas@gabrielhenriq.com.br
- Senha: 01020304@#\$

```
try {  
    //Server settings  
    $mail->SMTPDebug = SMTP::DEBUG_SERVER; //Enable verbose  
    $mail->isSMTP(); //Send using S  
    $mail->Host = 'mail.gabrielhenriq.com.br'; //Set  
    $mail->SMTPAuth = true; //Enable SMTP  
    $mail->Username = 'aulas@gabrielhenriq.com.br'; //SM  
    $mail->Password = '01020304@#$'; //SMTP pa  
    $mail->SMTPSecure = PHPMailer::ENCRYPTION_STARTTLS; //Enable impli  
    $mail->Port = 587; //TCP port to  
  
    //Recipients  
    $mail->setFrom('aulas@gabrielhenriq.com.br', 'Mailer');  
    $mail->addAddress('gabriel.construsite@gmail.com', 'Joe User'); //Add  
  
    //Content  
    $mail->isHTML(true); //Set email format t  
    $mail->Subject = 'Here is the subject';  
    $mail->Body = 'This is the HTML message body <b>in bold!</b>';  
    $mail->AltBody = 'This is the body in plain text for non-HTML mail clients  
  
    $mail->send();  
    echo 'Message has been sent';  
} catch (Exception $e) {  
    echo "Message could not be sent. Mailer Error: {$mail->ErrorInfo}";  
}
```





Atividade:

- 1) Usando nossa calculadora de frete de ontem.
Sempre que um frete for calculado disparar um email com o seguinte corpo:

"Olá administrador,

O usuario **NOME_USUARIO** acaba de calcular o frete: **NOME_FRETE**

Para o produto : **NOME_PRODUTO**

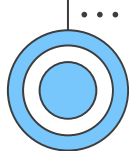
Totalizando: **VALOR_TOTAL**

Att: Equipe de atendimento **NOME_SISTEMA.**"



GALAXPAY\$
SEU NEGÓCIO RECORRENTE





Enviando Form com Anexo.

- O PHPMailer, fornece nativamente o envio de arquivos.
- Para isso, vamos criar um novo arquivo.
disparar-anexo.php
- Neste arquivo, vamos ter um <form></form>
Com um input de Name, e outro FILE para anexarmos a img.

Apps Intranet Const... Informações d... Logins aut

Informe seu nome:

Selecione seu arquivo: Nenhum arquivo selecionado

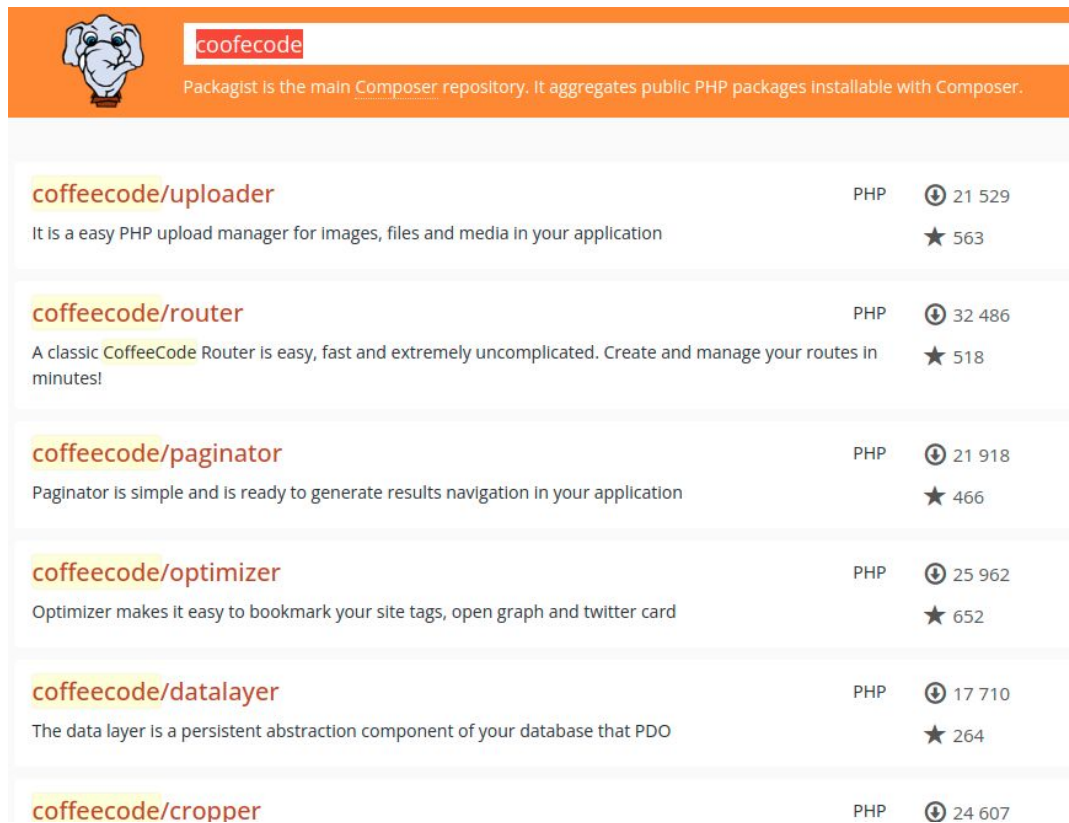
```
//Attachments
$mail->addAttachment('/var/tmp/file.tar.gz'); //Add attachments
$mail->addAttachment('/tmp/image.jpg', 'new.jpg'); //Optional name
```



GALAXPAY
SEU NEGÓCIO RECORRENTE

Dicas de Pacotes

- coofecode/



The screenshot displays the coofecode repository on Packagist. The header features the coofecode logo (an elephant) and the text "Packagist is the main Composer repository. It aggregates public PHP packages installable with Composer." Below this, a list of packages is shown, each with its name, description, PHP version, download count, and star count.

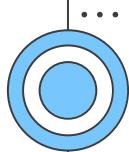
Package Name	Description	PHP	Downloads	Stars
coofecode/uploader	It is a easy PHP upload manager for Images, files and media in your application	PHP	21 529	563
coofecode/router	A classic CoffeeCode Router is easy, fast and extremely uncomplicated. Create and manage your routes in minutes!	PHP	32 486	518
coofecode/paginator	Paginator is simple and is ready to generate results navigation in your application	PHP	21 918	466
coofecode/optimizer	Optimizer makes it easy to bookmark your site tags, open graph and twitter card	PHP	25 962	652
coofecode/datalayer	The data layer is a persistent abstraction component of your database that PDO	PHP	17 710	264
coofecode/cropper		PHP	24 607	





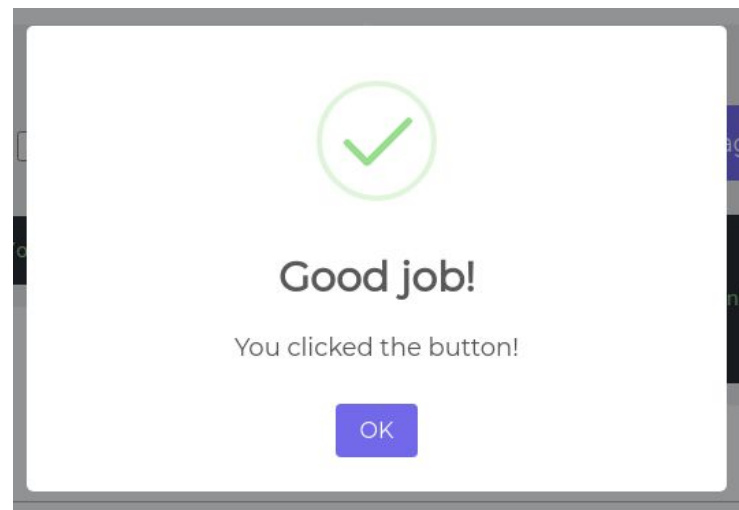
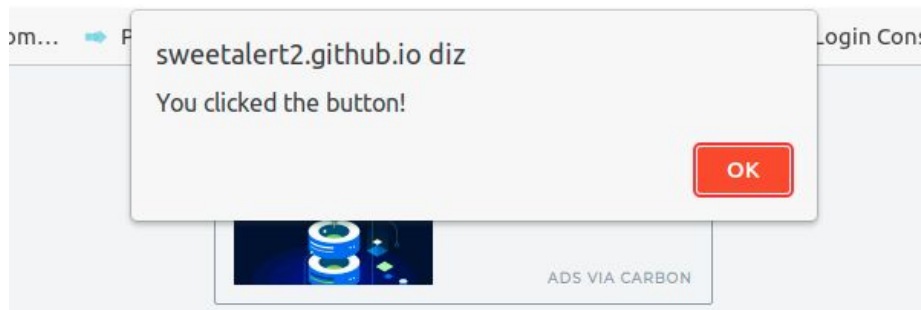
SweetAlert





Conhecendo o SweetAlert

- Substituto para os alertas js padrão
- Uma opção, simples, bonita, elegante, fácil de usar e repleta de funcionalidades.



Como adicionar em seu projeto.

- Necessário o JQuery*

DOWNLOAD & INSTALL

```
$ npm install sweetalert2
```

Or grab from [jsdelivr CDN](#)

```
<script src="//cdn.jsdelivr.net/npm/sweetalert2@11"></script>
```

sweetalert2 is the **19th** most popular package on jsDelivr, with **968.007.020** CDN hits in the last month

```
?>  
<script src="https://cdn.jsdelivr.net/npm/sweetalert2@11"></script>  
<script src="https://code.jquery.com/jquery-3.6.0.slim.min.js"></script>
```

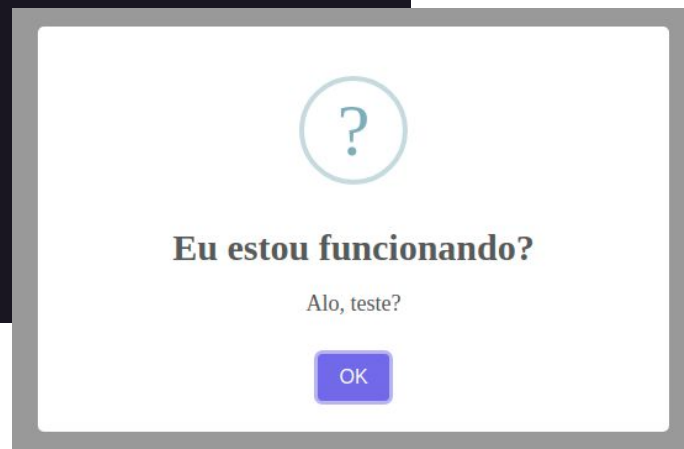


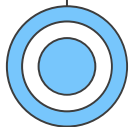
GALAXPAY
SEU NEGÓCIO RECORRENTE

Usando

- Necessário o JQuery*

```
11
12 <script src="//cdn.jsdelivr.net/npm/sweetalert2@11/dist/sweetalert2.min.js"></script>
13 <link rel="stylesheet" href="//cdn.jsdelivr.net/npm/sweetalert2@11/dist/sweetalert2.min.css" id="theme-styles">
14
15
16 <form method="POST" enctype="multipart/form-data" action="/">
17   <label for="name"> Informe seu nome: </label>
18   <input type="text" name="name"></input>
19   <br><br>
20   <label for="file">Selecione seu arquivo: </label>
21   <input type="file" name="file" ></input>
22   <br><br>
23   <input type="submit" name="Enviar Form">
24 </form>
25
26 <script>
27   Swal.fire(
28     'Eu estou funcionando?',
29     'Alo, teste?',
30     'question'
31   )
32 </script>
```





Usando

- Confirmando, ou não ações.

```
25
26 <script>
27 Swal.fire({
28   title: 'Tem certeza que deseja enviar o formulário?',
29   showDenyButton: true,
30   showCancelButton: false,
31   confirmButtonText: 'Sim',
32   denyButtonText: `Cancelar`,
33 }).then((result) => {
34   /* Read more about isConfirmed, isDenied below */
35   if (result.isConfirmed) {
36     Swal.fire('Formulário enviado!', '', 'success')
37   } else if (result.isDenied) {
38     Swal.fire('Formulário não enviado', '', 'info')
39   }
40 })
41 </script>
42
```

Tem certeza que deseja enviar o formulário?

Sim

Cancelar



Formulário enviado!

OK

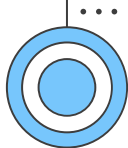


Formulário não enviado

OK



GALAXPAY
SEU NEGÓCIO RECORRENTE



Atividades

- 1) Usando o projeto criado, que envia um anexo.
 - Quando clicar no enviar, quem deve disparar o evento de submit é o JS.
 - De opção de enviar o form ou não, via sweetalert2
 - Caso sim, envie o form com Ajax, e retorne um alert simpático
 - Caso não, volte com um alerta simpático.

Usar os ícones, e funcionalidades do sweetalert2.

<https://sweetalert2.github.io/>



Atividades

- 1) Usando o nosso projeto de frete agora, lembra que ele dispara o email ?
- Usando sweetalert2, e Ajax, vamos fazer:
 - Criar uma opção após o cálculo do frete, do cliente enviar um email.
 - Caso clique neste botão, irá abrir uma caixa input **em sweetalert2** em que solicitará para qual email o sistema irá enviar o orçamento.
 - De posse desse email, dispare as informações necessárias, conforme layout já visto para o email informado.

