

Primero, vamos a descargar el composer en cada uno de los diferentes contenedores:

CORREO: Guzzle y PHPMailer

```
PS C:\xampp\htdocs\examendocker\OJALA> CD ..  
PS C:\xampp\htdocs\examendocker> CD CORREO  
PS C:\xampp\htdocs\examendocker\CORREO> composer require guzzlehttp/guzzle
```

```
PS C:\xampp\htdocs\examendocker\OJALA> CD ..  
PS C:\xampp\htdocs\examendocker> CD CORREO  
PS C:\xampp\htdocs\examendocker\CORREO> composer require phpmailer/phpmailer
```

OJALA: Guzzle

```
PS C:\xampp\htdocs\examendocker> cd OJALA  
PS C:\xampp\htdocs\examendocker\OJALA> composer require guzzlehttp/guzzle
```

CESTERO: dompdf


```
C:\xampp\htdocs>cd examendocker  
  
C:\xampp\htdocs\examendocker>cd cestero  
  
C:\xampp\htdocs\examendocker\CESTERO>composer require dompdf/dompdf
```

A continuación vamos a actualizar la versión del guzzle siguiendo los siguientes pasos:

1. Borramos composer.lock
2. Ejecutamos el siguiente comando:

```
PS C:\xampp\htdocs\examendocker\OJALA> cd ..  
PS C:\xampp\htdocs\examendocker> cd CORREO  
PS C:\xampp\htdocs\examendocker\CORREO> composer install
```

Una vez hecho eso, debemos de tener los dockerfile:

CESTERO >  Dockerfile

```
1 from php:8.1.0-apache
2 run apt-get update
3 run docker-php-ext-install pdo pdo_mysql
4 workdir /var/www/html
5 copy . .
```

CORREO >  Dockerfile

```
1 from php:8.1.0-apache
2 run apt-get update
3 run docker-php-ext-install pdo pdo_mysql
4 workdir /var/www/html
5 copy . .
```

DATOS >  Dockerfile

```
1 FROM mariadb:latest
2
3 # Establecer la contraseña de root
4 ENV MARIADB_ROOT_PASSWORD=root
5 run apt-get update
6 run docker-php-ext-install pdo pdo_mysql
7
8 # Copiar el script init.sql al directorio de inicialización
9 COPY examen.sql /docker-entrypoint-initdb.d
10
```

OJALA >  Dockerfile

```
1 from php:8.1.0-apache
```

Una vez hecho esto, haremos el docker compose:

```
version: "3.0"
services:
  ojala:
    build: ./OJALA #'ruta del dockerfile'
    ports:
      - 8085:80
    volumes:
      - "./OJALA:/var/www/html" #'ruta del dockerfile'
    container_name: ojala
    depends_on:
      - datos
  correo:
    build: ./CORREO #'ruta del dockerfile'
    volumes:
      - "./CORREO:/var/www/html" #'ruta del dockerfile'
    container_name: correo
    depends_on:
      - datos
  datos:
    build: ./DATO
  cestero:
    build: ./CESTERO
    volumes:
      - "./CESTERO:/var/www/html"
    container_name: cestero
    depends_on:
      - datos
```

Una vez hecho lo construimos con el siguiente comando:

```
PS C:\xampp\htdocs\examendocker> docker compose build
[+] Building 0.6s (7/7)
=> [datos internal] load build context
[+] Building 5.4s (25/25) FINISHED
=> [datos internal] load build definition from Dockerfile
```

Para arrancarlo, se realiza con el siguiente comando:

```
PS C:\xampp\htdocs\examendocker> docker compose up
[+] Building 0.0s (0/0)
[+] Running 4/0
✓ Container examendocker-datos-1 Created
✓ Container cestero Created
✓ Container correo Created
✓ Container ojala Created
Attaching to cestero, correo, examendocker-datos-1, ojala
```

A continuación, mostraremos los códigos a ejecutar:

```
OJALA > index.php > ...
1  <?php
2      if ($_SERVER["REQUEST_METHOD"]=="POST"){
3          require_once "../vendor/autoload.php";
4          $nombre=$_POST['nombre'];
5          $enviar=isset($_POST['enviar']);
6          if ($enviar){
7              if (strlen($nombre)>0){
8                  try {
9                      $client = new GuzzleHttp\Client();
10
11                      $res = $client->request('POST', 'http://correo', [
12                          'form_params' => [
13                              'nombre' => $nombre,
14                          ]
15                      ]);
16                      echo $res->getBody();
17                  } catch (GuzzleHttp\Exception\RequestException $e) {
18                      // Manejar errores de Guzzle
19                      echo 'Error: ' . $e->getMessage();
20                      if ($e->hasResponse()) {
21                          echo 'Response: ' . $e->getResponse()->getBody()->getContents();
22                      }
23                  }
24              }else{
25                  echo "El nombre no puede estar vacío";
26              }
27          }
28      }
```

CESTERO > index.php > ...

```
1  <?php
2      require_once "../vendor/autoload.php";
3      use Dompdf\Dompdf;
4      if ($_SERVER["REQUEST_METHOD"]=="POST"){
5          $objeto=json_decode($_POST['objeto']);
6          $tipo=$objeto->tipo;
7          $pdfprovisional=new Dompdf();
8          $html='<!DOCTYPE html>
9          <html lang="en">
10         <head>
11             <meta charset="UTF-8">
12             <meta name="viewport" content="width=device-width, initial-scale=1.0">
13             <title>Document</title>
14         </head>
15         <body>';
16         $html.='<p>'.$tipo.'</p>';
17         $html.='</body>
18         </html>';
19         $pdfprovisional->setPaper("A4", "portrait");
20         # Cargamos el contenido HTML.
21         $pdfprovisional->loadHtml($html);
22
23         # Renderizamos el documento PDF.
24         $pdfprovisional->render();
25
26         # Creamos un fichero
27         echo $pdfprovisional->output();
```

CORREO > index.php > ...

```
1  <?php
2      require "vendor/autoload.php";
3      use PHPMailer\PHPMailer\PHPMailer;
4
5      if ($_SERVER["REQUEST_METHOD"] == "POST") {
6
7          $nombre=$_POST['nombre'];
8
9          // Ahora, $datos es un array que contiene los datos enviados en el cuerpo de la solicitud
10         // Puedes acceder a cada elemento así:
11         $conexion=new PDO("mysql:host=datos;dbname=cestero","root","root");
12         $resultado=$conexion->prepare("SELECT * FROM cesta where nombre=:nombre");
13         $resultado->bindParam(":nombre",$nombre,PDO::PARAM_STR);
14         $resultado->execute();
15         $objeto=null;
16
17         while ($tuplas = $resultado->fetch(PDO::FETCH_OBJ)) {
18             $nombre=$tuplas->nombre;
19             $tipo=$tuplas->tipocesta;
20             $objeto=new stdClass();
21             $objeto->nombre=$nombre;
22             $objeto->tipo=$tipo;
23         }
24
25         if ($objeto!=null){
26             $contenido="";
27             try {
```