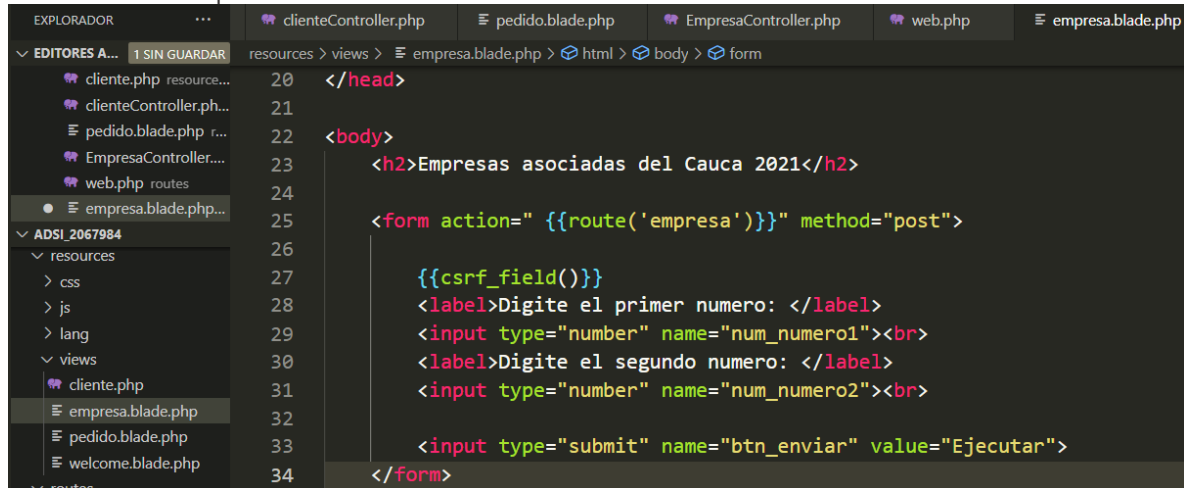


## Manejo del Modelo Vista Controlador en Laravel

**Ejercicio 1.** Diseñar un formulario en HTML el cual solicite 2 números enteros y permita realizar las operaciones aritméticas. Devolviendo el valor de estas en pantalla.

Creemos una vista para crear el formulario HTML



```

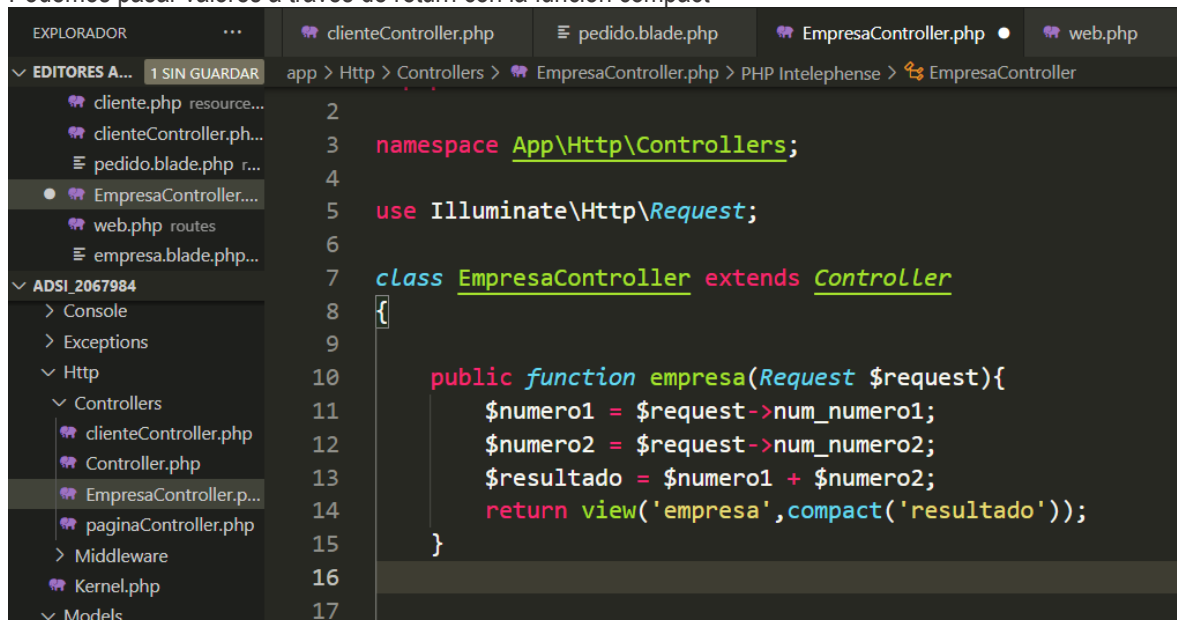
EXPLORADOR  ...  clienteController.php  pedido.blade.php  EmpresaController.php  web.php  empresa.blade.php
EDITORES A... 1 SIN GUARDAR  resources > views > empresa.blade.php > html > body > form
cliente.php resource... 20 </head>
clienteController.ph... 21
pedido.blade.php r... 22 <body>
EmpresaController.... 23 <h2>Empresas asociadas del Cauca 2021</h2>
web.php routes 24
empresa.blade.php... 25 <form action="{{route('empresa')}}" method="post">
26 {{csrf_field()}}
27 <label>Digite el primer numero: </label>
28 <input type="number" name="num_numero1"><br>
29 <label>Digite el segundo numero: </label>
30 <input type="number" name="num_numero2"><br>
31 <input type="submit" name="btn_enviar" value="Ejecutar">
32 </form>
33
34

```

Creemos un controlador para recibir los valores capturados y hacer las operaciones aritméticas  
**php artisan make:controller usuarioController**

Observe que el formulario recibe los datos por el método POST. Los cuales llegan al controlador a través de la variable Request que permite capturar datos del formulario en formato JSON, los cuales podemos hacer referencia o extraerlos con su nombre y →

Podemos pasar valores a través de return con la función compact



```

EXPLORADOR  ...  clienteController.php  pedido.blade.php  EmpresaController.php  web.php
EDITORES A... 1 SIN GUARDAR  app > Http > Controllers > EmpresaController.php > PHP Intelephense > EmpresaController
cliente.php resource... 2
clienteController.ph... 3 namespace App\Http\Controllers;
pedido.blade.php r... 4
EmpresaController.... 5 use Illuminate\Http\Request;
web.php routes 6
empresa.blade.php... 7 class EmpresaController extends Controller
8 {
9
10 public function empresa(Request $request){
11     $numero1 = $request->num_numero1;
12     $numero2 = $request->num_numero2;
13     $resultado = $numero1 + $numero2;
14     return view('empresa',compact('resultado'));
15 }
16
17

```

Para obtener el resultado de las operaciones en pantalla; necesitamos crear un método en el mismo controlador que me lleve a una ruta de tipo GET y visualizar el resultado en pantalla.

```
public function create(){  
    return view('empresa');  
}
```

Creamos las rutas en el archivo web

```
//recibe los datos desde el formulario  
Route::post('/empresa-  
opera',[EmpresaController::class, 'empresa'])-  
>name('empresa');  
//muestra los datos en pantalla resultado de la operacion  
Route::get('/empresa', [EmpresaController::class, 'create']  
);
```

### **Ejercicio 2.**

Leer tres números y determinar si la suma de cualquier pareja de ellos es igual al tercer número. Si se cumple esta condición escribir "Iguales", en caso contrario, escribir "Distintas".

### **Ejercicio 3.**

A un trabajador le pagan según sus horas y una tarifa de pago por horas. Si la cantidad de horas trabajadas es mayor a 50 horas. La tarifa se incrementa en un 30% para las horas extras. Calcular el salario del trabajador dadas las horas trabajadas y la tarifa.

### **Ejercicio 4.**

Sumar los números enteros solicitando al usuario el número hasta el que se desea sumar; validar que el número máximo a ingresar sea 1000, ej: Digite los números a sumar: 5 -> Sumara los números 1,2,3,4,5 = Respuesta=> 15

### **Ejercicio 5.**

Se necesita desarrollar un programa que solicite a un usuario el radio y altura de un cilindro para dar como respuesta el volumen y área del cilindro.

### **Ejercicio 6.**

Realizar un programa el cual reciba de un formulario la siguiente información de un estudiante: Nombre, Calificación del primer parcial y calificación del examen final y en seguida calcule e imprima el valor de la calificación definitiva, teniendo en cuenta que ella se obtiene como el 40% de la calificación del primer Parcial más el 60% de la calificación del examen final. Además, deberá imprimir



un mensaje de aprobó o reprobó según corresponda y concepto valorativo el cual se obtiene de la siguiente manera:

*Malo si la calificación definitiva es inferior a 2.*

*Insuficiente: si la calificación definitiva va desde 2 y es inferior a 3.*

*Aceptable: si la calificación definitiva va desde 3 y es inferior a 4.*

*Bueno: si la calificación definitiva va desde 4 y es inferior a 4.5.*

*Excelente: si la calificación definitiva va desde 4.5 en adelante.*

## Bibliografía

<https://laravel.com/docs/8.x>

<https://www.youtube.com/watch?v=KKpXpWCTlbo&list=PLPI81lqbj-4KHPEGngoy5PSjjxcwnpCdb>

<https://aprendible.com/series/laravel-desde-cero>

<https://imacreste.com/bootstrap-colores-fuentes-y-tipografias/#Small>

<https://bluuweb.github.io/tutorial-laravel/bases-datos/>

[http://www.oscarabadfolgueira.com/crear-una-base-datos-mysql-desde-consola/#Acceder\\_a\\_mysql\\_desde\\_la\\_consola](http://www.oscarabadfolgueira.com/crear-una-base-datos-mysql-desde-consola/#Acceder_a_mysql_desde_la_consola)

BEATI Hernán (2015). PHP - Creación de páginas Web dinámicas 2a edición. Editorial Alfaomega. 2015

**W3C The World Wide Web Consortium (2016).** HTML 5.1 W3C Recommendation.

Recuperado de: <https://www.w3.org/TR/html>

**W3C The World Wide Web Consortium (2017).** CSS Snapshot 2017. Recuperado de:

<https://www.w3.org/TR/CSS>

<https://www.cursosdesarrolloweb.es/cursos-gratuitos/>

## CONTROL DEL DOCUMENTO

	Nombre	Cargo	Dependencia	Fecha
Autor(es)	Franco Reina Argoty	Instructor	CTPI	Junio 2021