

SOFTURA SOLUTIONS

Capacitación PHP con Laravel

Entregable

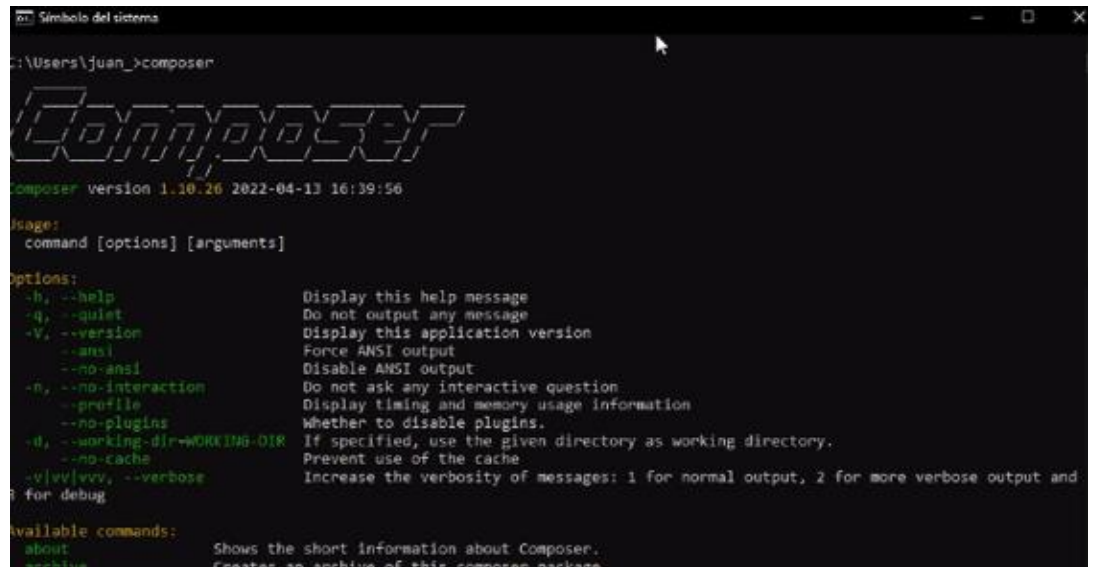
Omar Andrés Juárez Flores



22

Herramientas necesarias:

- Composer
- Wampserver
- Laravel 5.5
- PHP 7.2
- Mysql
- Apache 2.4.4.1



```
C:\Users\juan>composer

Composer version 1.10.26 2022-04-13 16:39:56

Usage:
  command [options] [arguments]

Options:
  -h, --help                Display this help message
  -q, --quiet               Do not output any message
  -V, --version             Display this application version
      --ansi                Force ANSI output
      --no-ansi            Disable ANSI output
  -n, --no-interaction     Do not ask any interactive question
      --profile            Display timing and memory usage information
      --no-plugins         Whether to disable plugins.
  -d, --working-dir=WORKING-DIR If specified, use the given directory as working directory.
      --no-cache           Prevent use of the cache
  -v|vv|vvv, --verbose     Increase the verbosity of messages: 1 for normal output, 2 for more verbose output and 3 for debug

Available commands:
  about          Shows the short information about Composer.
  archive        Creates an archive of this composer package
```

Crear el usuario para el login con este comando desde el cmd

Creemos usuario para nuestro login con el siguiente comando desde el cmd:

php artisan make:auth

Modificar el archivo .env

Crearemos una base de datos con el nombre “curso_laravel_0822”

Se modifica el archivo .env colocando los datos de nuestra base de datos

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3308
DB_DATABASE=curso_laravel_0822
DB_USERNAME=root
DB_PASSWORD=
```

Crear migraciones (Base de datos)

Se crea la tabla “empleado” y posterior

A eso se hizo un “php artisan migrate” desde

El cmd y así el framework nos crea el archivo de

```
public function up()
{
    Schema::create('empleado', function (Blueprint $table) {
        $table->increments('id');
        $table->string('nombre', 50);
        $table->string('apellido_paterno', '30');
        $table->string('apellido_materno', '30');
        $table->string('correo', 50);
        $table->date('fecha_nacimiento')->nullable();
        $table->string('direccion', 100)->nullable();
        $table->enum('genero', ['masculino', 'femenino']);
        $table->char('telefono', 15);
        $table->integer('codigo_empleado');
        $table->timestamps();
    });
}
```

Migraciones de la tabla empleado en el proyecto,

Y despues en dicho archivo incluir los campos de la tabla.

Y nuevamente hacer una migracion en el cmd para que se modifique la tabla en la base de datos y este paso se hara tambien la tabla de datos de contactos.

Creamos nuestros modelos

```
class Empleado extends Model
```

```
{
    protected $table = 'empleado';

    protected $fillable = ['id',
        'nombre',
        'apellido_paterno',
        'apellido_materno',
        'correo',
        'fecha_nacimiento',
        'direccion',
        'genero',
        'telefono',
        'salario',
        'tipo_moneda',
        'codigo_empleado'
    ];
}
```

```
class datoContacto extends Model{
```

```
    protected $table = 'datocontacto';

    protected $fillable = ['id',
        'empleado_id',
        'nombre_contacto',
        'email', 'telefono',
        'direccion', 'ciudad',
        'estado',
        'cp',
        'principal',
        'eliminado'];
}
```

Creamos nuestros formularios en nuestras vistas(create, edit) para empleado y datos de contacto(create)

```
<form method="POST" action="{{route('datoContacto.store')}}">
    {{csrf_field()}}

    <input type="hidden" name="empleado_id" value="{{ $empleadoId }}">

    <div class="row">
        <div class="col-md-4">
            <div class="form-group">
                <label class="form-label">Nombre de contacto</label>
                <input type="text" name="nombre_contacto" id="nombre_contacto" class="form-control">
            </div>
        </div>
        <div class="col-md-4">
            <div class="form-group">
                <label class="form-label">Email</label>
                <input type="email" name="email" class="form-control" id="email" placeholder="name@>
            </div>
        </div>
        <div class="col-md-4">
            <div class="form-group">
                <label class="form-label">Telefono</label>
                <input type="text" name="telefono" id="telefono" class="form-control input-sm" plac>
            </div>
        </div>
    </div>

    <div class="row">
```

Modificamos el archivo routes/web.php y agregamos nuestras rutas que utilizaremos para peticiones

```
Route::get('/', function () {  
    return view('welcome');  
});  
  
Auth::routes();  
  
Route::get('/home', 'HomeController@index')->name('home');  
  
Route::get('/empleado/index', 'EmpleadoController@index')->name('empleado.index');  
  
Route::get('/empleado/{empleado}/show', 'EmpleadoController@show')->name('empleado.show');  
  
Route::get('/empleado/create', 'EmpleadoController@create')->name('empleado.create');  
Route::post('/empleado/store', 'EmpleadoController@store')->name('empleado.store');  
  
Route::delete('/empleado/{empleado}', 'EmpleadoController@destroy')->name('empleado.destroy');  
  
Route::get('/empleado/{empleado}/edit', 'EmpleadoController@edit')->name('empleado.edit');  
  
Route::put('/empleado/{empleado}', 'EmpleadoController@update')->name('empleado.update');  
  
Route::post('/changeLang', 'HomeController@changeLang')->name('changeLang');
```

Posteriormente se agregara la librería Guzzle desde el cmd con composer

El comando a utilizar es el siguiente:

composer require guzzlehttp/guzzle

y agregamos la librería a nuestro controlador :

use GuzzleHttp\Client as HttpClient;

Para despues hacer una peticion a un web service

```
public function create() {  
  
    $currencyWS = $this->obtenerCurrencyWS();  
    $listMonedas = explode(";", $currencyWS);  
    return view('Empleado.create', compact('listMonedas'));  
  
}
```

Capturas del sistema

Listado empleados

Agregar empleado

Id empleado	Nombre	Apellidos	Correo	Acciones
10	Alejandro	Hernandez Blanco	alejandro@gmail.com	 Eliminar Editar
9	Ignacio	Varga Sanchez	nacho@gmail.com	 Eliminar Editar
8	Eduardo	Silva Lopez	ricardo@gmail.com	 Eliminar Editar
7	Carlos	Lopez Lopez	omar@gmail.com	 Eliminar Editar
5	Andrea	Hernandez Saldaña	raul@gmail.com	 Eliminar Editar
4	Francisca	Samayoa Perez	ale20_@gmail.com	 Eliminar Editar
1	Omar	Juarez Flores	omar@gmail.com	 Eliminar Editar

Agregar Empleado

Nombre	<input type="text" value="nombre"/>	Apellido Paterno	<input type="text" value="apellido_paterno"/>	Apellido Materno	<input type="text" value="apellido_materno"/>
Fecha nacimiento	<input type="text" value="fecha_nacimiento"/>	Genero	<input type="radio"/> Masculino <input type="radio"/> Femenino		
Email	<input type="text" value="correo"/>	Telefono	<input type="text" value="telefono"/>	Dirección	<input type="text" value="direccion"/>
Salario	<input type="text" value="Salario"/>	Tipo de moneda	<input data-bbox="678 1371 1019 1402" type="text" value=" >--Seleccione Tipo de Moneda--< "/>		
<input type="button" value="Guardar"/> <input type="button" value="Atras"/>					

Datos de empleado

Código empleado: 123

Nombre: Ignacio

Apellido Paterno: Varga

Apellido Materno: Sanchez

Fecha nacimiento: 2000-08-08

Genero: masculino

Dirección: Mexico

Email: nacho@gmail.com

Telefono: 4567891234

Salario: 167.00 MXN

Tipo de moneda: MXN

Datos de contacto

Agregar contacto

Nombre contacto	Email	Telefono	Dirección	Ciudad	Estado	CP	Acciones
Contacto personal	nachito@gmail.com	2414198011	Apizaquito	Apizaco	Tlaxcala	90300	Eliminar ;

Atras