

ENACO

WEB INVENTARIO TI [2021]

La presente documentación tiene la finalidad de explicar la estructura y desarrollo de la página web, así como el funcionamiento de ella.

INDICE GENERAL

Contenido

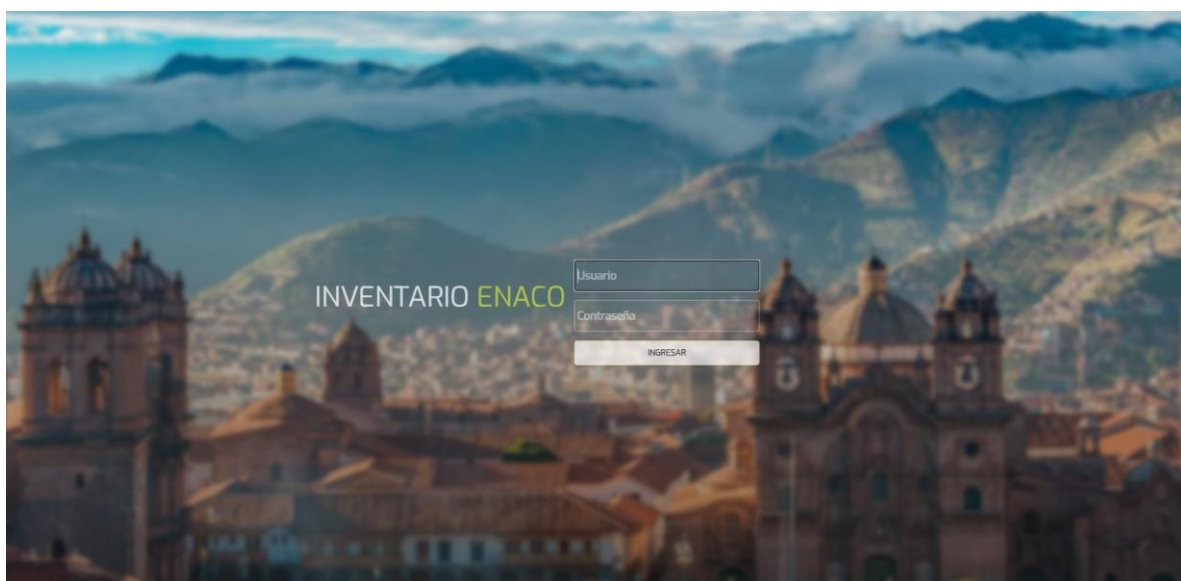
Introducción	1
Diccionario de Datos	2
Programación WEB	6
Manual de Usuario	10

Introducción

La página web INVENTARIO ENACO, tiene como propósito gestionar información importante acerca de los activos TI que se encuentran a disposición de la empresa.

El presente documento busca explicar a detalle cómo está estructurado y desarrollado la web. De igual manera se busca explicar el funcionamiento para que tanto administradores y usuarios de la web puedan darle un correcto uso a esta.

URL WEB: inventario.enaco.com.pe



Diccionario de Datos

La presente sección tiene el objetivo de explicar cómo se encuentra estructurada la base de datos, indicando los tipos de datos utilizados para cada campo y una pequeña descripción de cada una de las tablas.

Base de datos: Gestionada por MySQL

Nombre de DB: enaco_inventario

Diagrama de DB:

Tablas:

- **roles:** dicha tabla guardará información con respecto a los roles que maneja la web, contando con Administradores y Usuarios normales. En esta se encuentra el campo permiso donde se indica los privilegios según rol.

Column	Type	Default Value	Nullable
◇ id	int(10) unsigned		NO
◇ name	varchar(191)		NO
◇ slug	varchar(191)		NO
◇ code	int(11)	NULL	YES
◇ created_at	timestamp	NULL	YES
◇ updated_at	timestamp	NULL	YES
◇ permiso	text	NULL	YES

- **tm_colaborador:** dicha tabla guardará información con respecto a los colaboradores que laboran en ENACO, en esta se hace referencia a la tabla de Roles mediante el campo **tipo_usuario** a **roles** donde se distinguirá a los colaboradores administradores de usuarios normales.

Column	Type	Default Value	Nullable
◇ id_colaborador	bigint(20) unsigned		NO
◇ co_colaborador	varchar(255)		NO
◇ no_colaborador	varchar(255)	NULL	YES
◇ ap_paterno_colabora...	varchar(255)	NULL	YES
◇ ap_materno_colabor...	varchar(255)	NULL	YES
◇ nu_documento	text	NULL	YES
◇ ti_documento	text	NULL	YES
◇ usuario	text	NULL	YES
◇ password	text	NULL	YES
◇ email	text	NULL	YES
◇ foto	text	NULL	YES
◇ tipo_usuario	int(10) unsigned		NO
◇ estado_usuario	int(11)	NULL	YES

- **tm_sede:** dicha tabla guardará información con respecto a las sedes de ENACO.

Column	Type	Default Value	Nullable
◇ id_sede	bigint(20) unsigned		NO
◇ co_sede	varchar(255)	NULL	YES
◇ no_sede	varchar(255)	NULL	YES
◇ de_sede	varchar(255)	NULL	YES
◇ esta_sede	int(11)	NULL	YES

- **tm_oficina:** dicha tabla guardará información con respecto a las oficinas de ENACO, haciendo referencia a la sede que pertenece mediante el campo **id_sede** con la tabla **tm_sede**.

Column	Type	Default Value	Nullable
◇ id_oficina	bigint(20) unsigned		NO
◇ id_sede	bigint(20) unsigned		NO
◇ co_oficina	text	NULL	YES
◇ no_oficina	varchar(255)	NULL	YES
◇ de_oficina	varchar(255)	NULL	YES

- **tm_cargo_laboral:** dicha tabla guardará información con respecto a los cargos laborales que se manejan en ENACO.

Column	Type	Default Value	Nullable
◇ id_cargo_laboral	bigint(20) unsigned		NO
◇ co_cargo_laboral	varchar(255)	NULL	YES
◇ no_cargo_laboral	varchar(255)	NULL	YES
◇ de_cargo_laboral	varchar(255)	NULL	YES

- **tm_ofi_trabajador:** dicha tabla guardará información con respecto a la asignación de un colaborador con una oficina, sede y el cargo laboral que ocupa.

Column	Type	Default Value	Nullable
◇ id_ofi_trabajador	bigint(20) unsigned		NO
◇ id_oficina	bigint(20) unsigned		NO
◇ id_cargo_laboral	bigint(20) unsigned		NO
◇ id_sede	bigint(20) unsigned		NO
◇ id_colaborador	bigint(20) unsigned		NO
◇ est_trabajador	int(11)	1	NO
◇ created_at	timestamp	NULL	YES
◇ updated_at	timestamp	NULL	YES

- **tm_categoria_equipos:** dicha tabla guardará información con respecto a las categorías que se asignará posteriormente a registros de equipos.

Column	Type	Default Value	Nullable
◇ id_cat_equipos	bigint(20) unsigned		NO
◇ des_cate_equipo	varchar(255)	NULL	YES
◇ esta_cate_equipo	int(11)	NULL	YES
◇ created_at	timestamp	NULL	YES
◇ updated_at	timestamp	NULL	YES

- **tm_equipos:** dicha tabla guardará información con respecto a los equipos de ENACO, haciendo referencia a la categoría de esta mediante el campo **id_cat_equipos** con la tabla **tm_categoria_equipos**.

Column	Type	Default Value	Nullable
id_equipo	bigint(20) unsigned		NO
serie_equipo	varchar(255)	NULL	YES
cod_opatrimonial	varchar(255)	NULL	YES
des_equipo	text	NULL	YES
tipoBien	varchar(255)	NULL	YES
esta_equipo	int(11)	NULL	YES
fotos_equipo	longtext	NULL	YES
id_cat_equipos	bigint(20) unsigned		NO
created_at	timestamp	NULL	YES
updated_at	timestamp	NULL	YES

- **tm_ofi_traba_equipo:** dicha tabla guardará información con respecto a la asignación de un equipo con un colaborador de una oficina, en esta se dispone información tanto como oficina y sede donde se ubicará el equipo con el colaborador asignado y su cargo laboral.

Column	Type	Default Value	Nullable
id_ofi_traba_equipo	bigint(20) unsigned		NO
no_equipo	text	NULL	YES
sis_operativo	text	NULL	YES
estado_equipo	text	NULL	YES
esta_ofi_traba_equipo	int(11)	NULL	YES
id_equipo	bigint(20) unsigned	NULL	YES
id_ofi_trabajador	bigint(20) unsigned	NULL	YES
created_at	timestamp	NULL	YES
updated_at	timestamp	NULL	YES

- **tm_categoria_componentes:** dicha tabla guardará información con respecto a las categorías que se asignará posteriormente a registros de componentes.

Column	Type	Default Value	Nullable
id_cat_componentes	bigint(20) unsigned		NO
des_cate_componentes	varchar(255)	NULL	YES
esta_cate_componen...	int(11)	NULL	YES
created_at	timestamp	NULL	YES
updated_at	timestamp	NULL	YES

- **tm_componentes:** dicha tabla guardará información con respecto a los componentes de ENACO, haciendo referencia a la categoría de esta mediante el campo **id_cat_componentes** con la tabla **tm_categoria_componentes**.

Column	Type	Default Value	Nullable
id_componente	bigint(20) unsigned		NO
serie_componete	text	NULL	YES
des_componete	text	NULL	YES
esta_componete	int(11)	NULL	YES
fotos_componete	longtext	NULL	YES
id_cat_componentes	bigint(20) unsigned	NULL	YES
created_at	timestamp	NULL	YES
updated_at	timestamp	NULL	YES

- **tm_ofi_traba_equi_compo:** dicha tabla guardará información con respecto a la asignación de un componente con un equipo asignado, en esta se dispone información tanto como oficina y sede donde se ubicará el componente con el equipo ya asignado previamente.

Column	Type	Default Value	Nullable
◇ id_ofi_traba_equi_co...	bigint(20) unsigned		NO
◇ esta_ofi_traba_equi_...	int(11)	NULL	YES
◇ id_ofi_traba_equipo	bigint(20) unsigned	NULL	YES
◇ created_at	timestamp	NULL	YES
◇ updated_at	timestamp	NULL	YES

- **tm_solicitudes:** dicha tabla guardará información con respecto a solicitudes asignadas por el administrador, esta servirá posteriormente en el registro de una solicitud por parte de un usuario en el que tendrá como objetivo identificar el tipo de solicitud.

Column	Type	Default Value	Nullable
◇ id_solicitud	bigint(20) unsigned		NO
◇ cod_solicitud	text	NULL	YES
◇ nom_solicitud	text	NULL	YES
◇ esta_solicitud	int(11)	NULL	YES
◇ created_at	timestamp	NULL	YES
◇ updated_at	timestamp	NULL	YES

- **tm_soli_ofi_equi_traba:** dicha tabla guardará información con respecto a las solicitudes que registraran los usuarios. Haciendo referencia a un tipo de solicitud, esta mediante el campo **id_solicitud** con la tabla **tm_solicitudes**.

Column	Type	Default Value	Nullable
◇ id_soli_ofi_equi_tra	bigint(20) unsigned		NO
◇ descripcion_solidtud	text	NULL	YES
◇ fotos	longtext	NULL	YES
◇ doc_diagnostico	longtext	NULL	YES
◇ esta_soli_soli_ofi_eq...	int(11)	NULL	YES
◇ id_solicitud	bigint(20) unsigned		NO
◇ id_ofi_traba_equipo	bigint(20) unsigned		NO
◇ created_at	timestamp	current_timestamp()	NO
◇ updated_at	timestamp	current_timestamp()	NO

- **tm_imagenes** y **tm_documentos:** En ambas tablas se guardaran registros de imágenes y documentos subidos por los usuarios en los casos de registros de tablas anteriores.

Column	Type	Default Value	Nullable
◇ id_documento	bigint(20) unsigned		NO
◇ nom_documento	text	NULL	YES
◇ est_documento	text	NULL	YES
◇ id_equipo	bigint(20) unsigned	NULL	YES
◇ id_ofi_traba_equipo	bigint(20)	NULL	YES
◇ id_componente	bigint(20)	NULL	YES
◇ created_at	timestamp	NULL	YES
◇ updated_at	timestamp	NULL	YES
◇ url	text	NULL	YES

tm_documentos

Column	Type	Default Value	Nullable
◇ id	bigint(20) unsigned		NO
◇ nombre	text	NULL	YES
◇ url	text	NULL	YES
◇ created_at	timestamp	NULL	YES
◇ updated_at	timestamp	NULL	YES
◇ id_equipo	bigint(20) unsigned	NULL	YES
◇ id_ofi_traba_equipo	bigint(20)	NULL	YES
◇ id_componente	bigint(20)	NULL	YES
◇ esta_imagen	int(11)	NULL	YES

tm_imagenes

Programación WEB

LARAVEL

El presente proyecto está programado en el lenguaje de PHP bajo LARAVEL, a continuación se explica el funcionamiento de dicho framework.

Laravel es uno de los frameworks de código abierto más fáciles de asimilar para PHP.

El objetivo de Laravel es el de ser un framework que permita el uso de una sintaxis refinada y expresiva para crear código de forma sencilla, evitando el «código espagueti» y permitiendo multitud de funcionalidades. Aprovecha todo lo bueno de otros frameworks y utiliza las características de las últimas versiones de PHP.

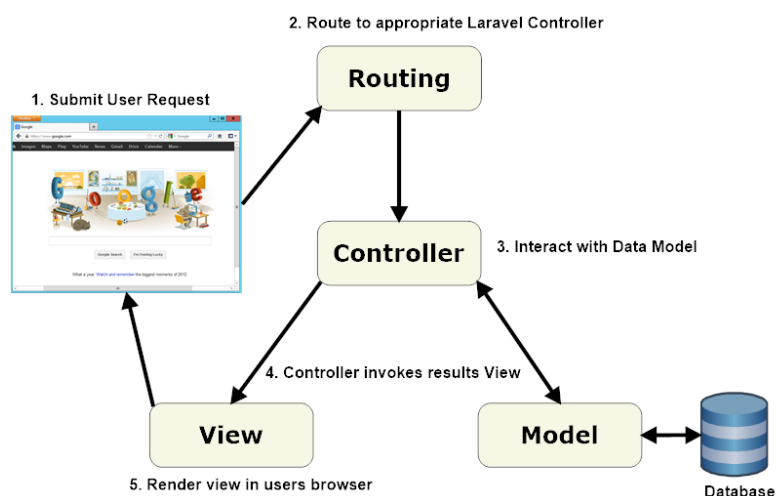


Características Generales

- Sistema de ruteo
- Blade, Motor de plantillas
- Peticiones Fluent y Eloquent ORM a través de la consola de Laravel Tinker
- Soporte para el caché
- Soporte para MVC

Programación de la web

La web ha sido diseñada y programada con ayuda del framework Laravel el cual tuvo el estilo de arquitectura de software MVC.



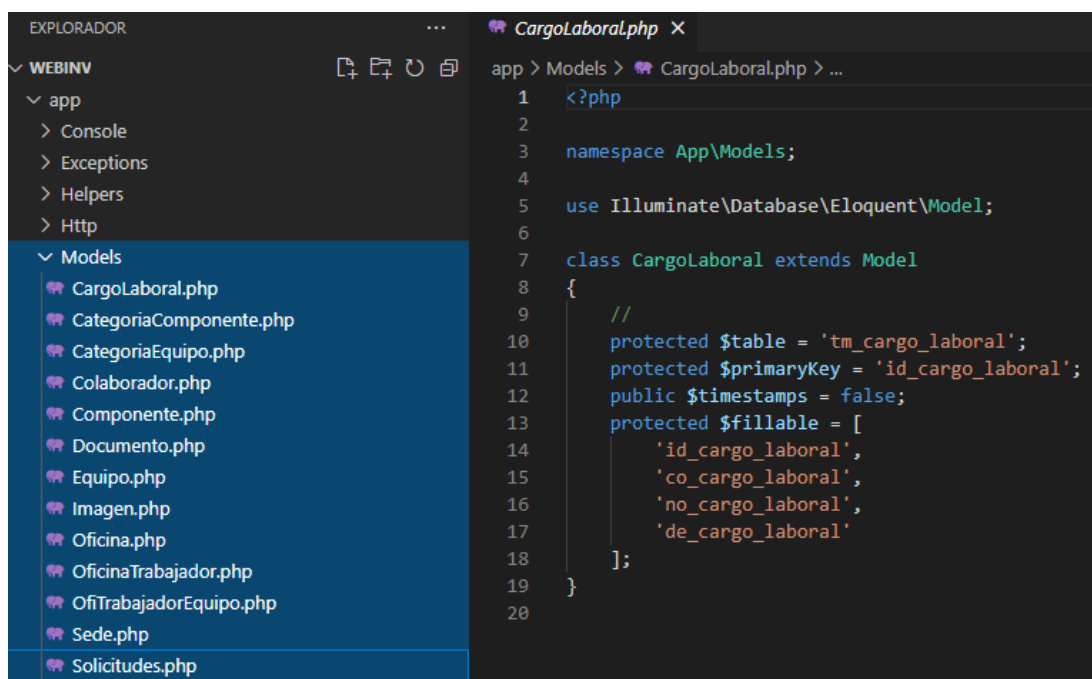
Patrón MVC

En este punto se explicará las secciones del patrón MVC y las direcciones de los archivos para esta.

Modelos de WEB

Las clases están representadas como modelos de la web, en ellos se encuentran especificados los atributos así como el nombre de la tabla; estas se encuentran ubicadas en la siguiente dirección:

.../app/Models/



The screenshot shows a code editor with a dark theme. On the left, the 'EXPLORADOR' (Explorer) pane shows a file tree for 'WEBINVENT'. Under 'app', there are folders for 'Console', 'Exceptions', 'Helpers', and 'Http'. The 'Models' folder is expanded, showing a list of PHP files: 'CargoLaboral.php', 'CategoriaComponente.php', 'CategoriaEquipo.php', 'Colaborador.php', 'Componente.php', 'Documento.php', 'Equipo.php', 'Imagen.php', 'Oficina.php', 'OficinaTrabajador.php', 'OfiTrabajadorEquipo.php', 'Sede.php', and 'Solicitudes.php'. The 'CargoLaboral.php' file is selected. The main editor area shows the code for 'CargoLaboral.php'. The code starts with a PHP tag, a namespace declaration 'App\Models', and a use statement for 'Illuminate\Database\Eloquent\Model'. It then defines a class 'CargoLaboral' that extends 'Model'. Inside the class, there are several protected and public properties: '\$table' (protected, value 'tm_cargo_laboral'), '\$primaryKey' (protected, value 'id_cargo_laboral'), '\$timestamps' (public, value false), and '\$fillable' (protected, an array of 'id_cargo_laboral', 'co_cargo_laboral', 'no_cargo_laboral', and 'de_cargo_laboral'). The class ends with a closing brace and a semicolon.

```
1 <?php
2
3 namespace App\Models;
4
5 use Illuminate\Database\Eloquent\Model;
6
7 class CargoLaboral extends Model
8 {
9     //
10    protected $table = 'tm_cargo_laboral';
11    protected $primaryKey = 'id_cargo_laboral';
12    public $timestamps = false;
13    protected $fillable = [
14        'id_cargo_laboral',
15        'co_cargo_laboral',
16        'no_cargo_laboral',
17        'de_cargo_laboral'
18    ];
19 }
20
```

Servicios de WEB

Desde aquí se tienen servicios de listar, registrar, editar, actualizar y eliminar. Estos serán solicitados por los CONTROLLERS, a su vez interactúan con los MODELS ya establecidos.

Los services se encuentran ubicados en la siguiente dirección:

```
.../app/Services/

Explorador (Ctrl+Mayús+E)
WEBINV
└─ app
   └─ Services
      └─ CargoLaboralService.php
         CargoLaboralService.php
         CategoriaComponenteService.php
         CategoriaEquipoService.php
         ColaboradorService.php
         ComponenteService.php
         DocumentoService.php
         EquipoService.php
         ImagenService.php
         OficinaService.php
         OficinaTrabajadorService.php
         OfiTrabajadorEquipoService.php
         SedeService.php
         SolicitudesService.php

CargoLaboralService.php X
app > Services > CargoLaboralService.php > ...
1  <?php
2
3  namespace App\Services;
4
5  use App\Models\CargoLaboral;
6
7  class CargoLaboralService
8  {
9
10     public function listar()
11     {
12         $selement = CargoLaboral::orderBy('id_cargo_laboral', 'ASC')->get();
13         return $selement;
14     }
15
16     public function registrar($request)
17     {
18         $selement= new CargoLaboral();
19
20         $selement->co_cargo_laboral=$request->get('co_cargo_laboral');
21         $selement->no_cargo_laboral=$request->get('no_cargo_laboral');
22         $selement->de_cargo_laboral=$request->get('de_cargo_laboral');
23
24         $selement->save();
25     }
26 }
```

Controller de WEB

Desde esta carpeta se visualizan los controladores de cada vista o sección. Donde desde cada archivo se tiene control de listar, registrar, editar, actualizar y eliminar.

Los controllers se encuentran ubicados en la siguiente dirección:

```
.../app/http/Controllers/Admin

EXPLORADOR
WEBINV
└─ app
   └─ Http
      └─ Controllers
         └─ Admin
            CargoLaboralController.php
            CategoriaComponenteController.php
            CategoriaEquipoController.php
            ColaboradorController.php
            ComponenteController.php
            DocumentoController.php
            EquipoController.php
            ImagenController.php
            OficinaController.php
            OficinaTrabajadorController.php
            OfinicaTrabajadorAjaxController.php
            OfiTrabajadorEquipoAjax.php
            OfiTrabajadorEquipoController.php
            SedeController.php
            SolicitudesController.php

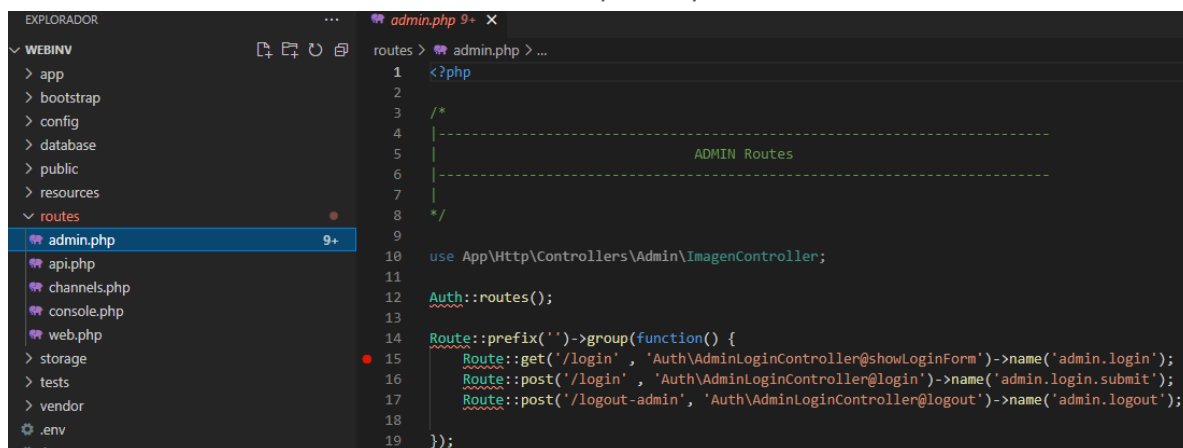
CargoLaboralController.php X
app > Http > Controllers > Admin > CargoLaboralController.php > ...
1  <?php
2
3  namespace App\Http\Controllers\Admin;
4
5  use App\Http\Requests\CargoLaboralRequest;
6  use App\Http\Controllers\Controller;
7  use App\Services\CargoLaboralService;
8
9  class CargoLaboralController extends Controller
10 {
11     //
12     private $service;
13
14     public function __construct(CargoLaboralService $service)
15     {
16         $this->service = $service;
17     }
18
19     public function index()
20     {
21         $selements = $this->service->listar();
22         return view('admin.cargolaboral-adm.index', compact('elements'));
23     }
24
25     public function create()
26 }
```

Rutas de WEB

En el administrador de contenido, routes se encarga de redirigir las vistas pues este interactúa con el Controller.

Las rutas se encuentran ubicados en la siguiente dirección:

.../routes/



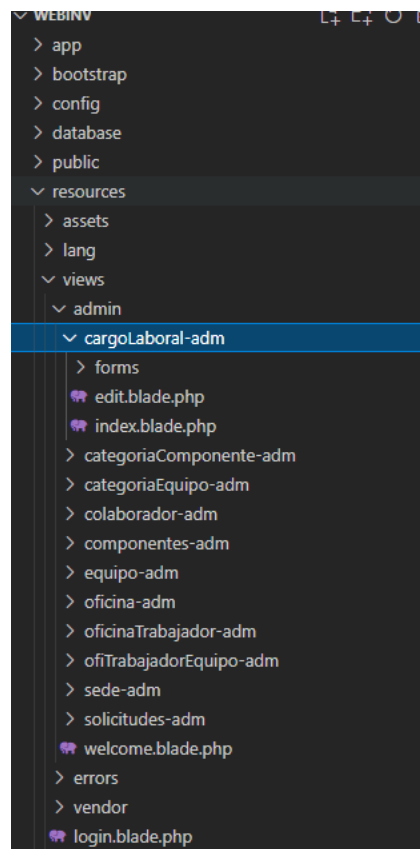
```
1 <?php
2
3 /*
4 |-----
5 |----- ADMIN Routes
6 |-----
7 |
8 */
9
10 use App\Http\Controllers\Admin\ImagenController;
11
12 Auth::routes();
13
14 Route::prefix('')->group(function() {
15     Route::get('/login', 'Auth\AdminLoginController@showLoginForm')->name('admin.login');
16     Route::post('/login', 'Auth\AdminLoginController@login')->name('admin.login.submit');
17     Route::post('/logout-admin', 'Auth\AdminLoginController@logout')->name('admin.logout');
18
19 });
```

Vistas de WEB

Esta carpeta contiene subcarpetas divididas por secciones de la página. En cada una de ellas se encuentra su archivo **index.php** como vista principal así como sus vistas de formularios y editables.

Las rutas se encuentran ubicados en la siguiente dirección:

.../resources/views/Admin/





Manual de Usuario