Instalación de Apache, PHP y MySQL:

Se instala el servidor web WampServer que nos brinda los servicios de Apache, PHP y MySQL con los que estaremos trabajando en este proyecto.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamentePHP 7.1  
APACHE 2.4.41  
MySQL 5



Instalación de Composer:

* Ejecutamos el instalador y seguimos sus pasos.
* Una vez terminado, podemos abrir una consola de Windows y ejecutar “composer”

Texto

Descripción generada automáticamente

Creación del proyecto laravel utilizando composer:

* Utilizando composer nos dirigimos en la dirección donde se instaló wamp y creamos el proyecto.
* Escribimos el comando composer create-project --prefer-dist laravel/laravel curso\_laravel "5.5.\*“

Texto

Descripción generada automáticamente

* Nos dirigimos al navegador y ya se nos creó el proyecto laravel.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Mostrar el listado de comandos del proyecto que creamos:

* Para conocer el listado completo de los comandos disponibles para nuestro proyecto creado ejecutamos en la consola, en el directorio raíz de nuestro proyecto de Laravel:
* php artisan list

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Configuración del Virtual Host:

* Entramos en la opción que nos brinda wamp para añadir un virtual host.
* Escribimos la URL que queramos que tenga nuestro virtual host “curso-laravel.com”.
* Añadimos la dirección donde tenemos el proyecto que creamos con laravel “c:\wamp64\www\curso\_laravel\public” y presionamos el botón Start the creation of the virtual host, este botón nos creara el virtual host con la dirección URL que le asignamos.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Estructura del proyecto:

* **app**: Contiene todo el código de la aplicación.
* **app/Console:** Contiene todos los comandos artisan para usar en la aplicación.
* **app/Exceptions:** Contiene las clases para manejar excepciones.
* **app/Http:** Almacena los controladores creados en la aplicación.
* **bootstrap:** Contiene los archivos y código que usa boostrap.
* **config:** Contiene los archivos de configuración.
* **database:** Contiene las migraciones y archivos de creación de registros para pruebas.
* **public:** Contiene archivos como imágenes, JavaScript CSS, etc.
* **resources:** Ahí se encuentran todos los archivos y vistas de nuestra aplicación.
* **routes:** Contiene las rutas para nuestra aplicación, (comunicación vista controlador).
* **.env:** Este fichero se utiliza para almacenar los valores de configuración que son propios de la máquina o instalación actual.
* **composer.json:** Este fichero es el utilizado por Composer para realizar la instalación de Laravel.

Creación de la BD:

Se crea la base de datos en MySQL con las siguientes características:Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Configuración de conexión a la base de datos, database y .env:

Se configura la conexión a la BD con el proyecto:

* Primer documento que se configura es el archivo database.php
* Charset = utf8
* Collation = utf8\_spanish\_ci

Texto

Descripción generada automáticamente

* Segundo documento que se configura con la BD es .env:

Texto

Descripción generada automáticamente

Creación del Login y Register:

Desde ventana de comandos CMD dentro del proyecto:

* Ejecutar el comando make:auth
* php artisan make:auth

Texto

Descripción generada automáticamente

Se nos crea dos carpetas en una están las vistas donde tenemos el login y register y en otra los layouts.

Interfaz de usuario gráfica, Aplicación, Correo electrónico

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ejecutar las migraciones:

Se debe ejecutar las migraciones para que nuestra BD exista, desde ventana de comandos CMD dentro del proyecto:

* Ejecutar el comando php artisan migrate

Texto

Descripción generada automáticamente

En la BD ya se agregó las migraciones que se ejecutaron.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Mostrar todas las rutas que tiene el proyecto:

Desde la ventana de comando CMD dentro del proyecto:

* Ejecutar el comando php artisan route:list

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

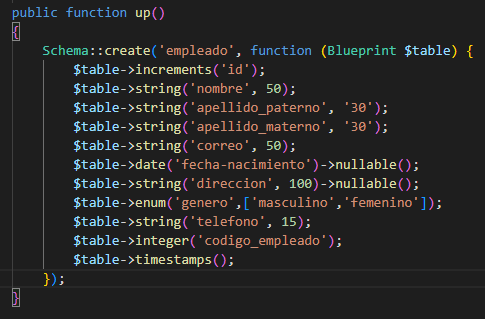
Creación del CRUD:

* Se realiza el CRUD de empleados.
* Creamos migración para tabla de BD
  + php artisan make:migration create\_empleado\_table –create=empleado

Texto

Descripción generada automáticamente

* Agregamos las columnas de BD dentro de la migración que se creó PHP



* Ejecutamos la migración
  + php artisan migrate

Texto

Descripción generada automáticamente

* Creamos el modelo junto con el controlador para la tabla creada
  + php artisan make:model Empleado -c -r
  + El modelo creado se genera en la carpeta **app\Empleado.php**
  + El controlador creado se genera en la carpeta **app\Http\Controller\EmpleadoController.php**

Texto

Descripción generada automáticamente

Agregar las funciones para CRUD de empleado:

* **Index**

**Texto

Descripción generada automáticamente**

* **Create**

**Texto

Descripción generada automáticamente**

* **Store**

**Texto

Descripción generada automáticamente**

* **Edit**

**Texto

Descripción generada automáticamente**

* **Update**

**Texto

Descripción generada automáticamente**

* **Destroy**

**Imagen que contiene Texto

Descripción generada automáticamente**

Agregar ruta para las funciones del controlador en el archivo de rutas routes\web.php:

**Texto

Descripción generada automáticamente**

Creación de la vista para mostrar Empleados:

* Creamos una nueva carpeta llamada Empleado en la ruta /resources/view, después de crear la carpeta creamos un archivo php llamado index.blade.php que va a contener las vistas (index, create, delete, edit).



Creación de la vista créate para agregar Empleados:

* Creamos un nuevo archivo php para agregar empleados con el nombre de create.blade.php donde tendrá un formulario para agregar los datos de los empleados.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Creación de la vista editar Empleados:

* Creamos un nuevo archivo php para editar los datos de los empleados con el nombre de edit.blade.php donde mandamos a llamar id del empleado seleccionado y nos mostrara un formulario con los datos que se agregaron anteriormente para poder editarlos.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Creación del button eliminar Empleado con un modal:

* En el archivo index.blade.php se agregó un button para eliminar un empleado junto con un modal que nos saldrá una ventana para preguntar si queremos eliminar el empleado.

Tabla

Descripción generada automáticamente con confianza media

Uso de Middleware por ruta:

Desde ventana de comandos CMD dentro del proyecto:

* Ejecutar el comando php artisan make:middleware
* Junto con el nombre de nuestro middleware que deseemos EditCreateDelete
* El middleware que se creo se guardo en la ruta **App\Http\Middleware**

Texto

Descripción generada automáticamente

Dentro del middleware que creamos se agregara una función para que un usuario que este autenticado en específico pueda hacer alguna de las funciones que se crearon anteriormente (créate,edit,delete).

Texto

Descripción generada automáticamente

Abrimos otro archivo con el nombre Kernel.php que esta en la ruta **App\Http**, agregaremos la ruta de nuestro middleware que creamos.

Texto

Descripción generada automáticamente

En el archivo donde están todas las rutas **route\web.php,** agregaremos el middleware que solo puede acceder el usuario agregamos en la ruta de las funciones del controlador.

Route::get('/empleado/create', 'EmpleadoController@create')->name('empleado.create')->middleware("editDeleteCreate.admin");

Texto

Descripción generada automáticamente

En este caso el middleware con la anulación de todos los empleados se la agregamos a la función Create para agregar usuarios solo el usuario que se especificó en el middleware puede agregar nuevos usuarios.

**Proyecto Evaluación**

**Texto

Descripción generada automáticamente**

Texto

Descripción generada automáticamente

Texto

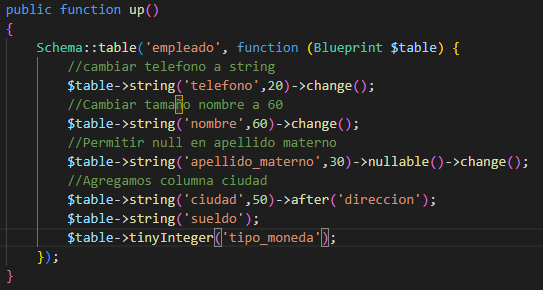
Descripción generada automáticamente

Texto

Descripción generada automáticamente

Texto

Descripción generada automáticamente



Tabla

Descripción generada automáticamente con confianza media

