



# Manual tecnico

Sistema interactivo de  
aperndizaje a distancia

SIAD



Integrantes de equipo:

- Luis Abdul Chávez Bustamante – 256910
- Omar Fernando Garcia Nápoles – 256944
- Darío Andrés Rosales Pérez Rodríguez –251961
- Oscar Ernesto Flores Jasso - 256935

## Contenido

Introducción .....	5
Descripción.....	5
Antecedentes .....	5
Misión y visión.....	5
Objetivos generales.....	6
Descripción y Delimitación del sistema.....	6
Objetivos generales.....	6
Descripción de usuarios .....	6
Diagrama de clases.....	7
Diagrama de casos de uso.....	8
Diagrama de actividades .....	8
Diagramas de secuencia .....	9
Modelado de la base de datos con Erwin .....	10
Esquema de la base de datos .....	11
Diseño de la interfaz de la aplicación.....	19
Inicio .....	19
Registros.....	19
Inicio de sesión.....	19
Menú Principal .....	20
Curso .....	20
Perfil del usuario .....	20
Editar usuario .....	21
Cursos.....	21
Curso que imparto.....	22
Kardex.....	22
Examen .....	23
Agregar pregunta a un examen.....	24
Publicar un examen.....	24
Calificaciones.....	25
Arquitectura del sistema.....	26

Especificación de ayuda en el sistema .....	26
Conexión de la aplicación cliente con el servidor .....	26
Acceso a la base de datos desde la aplicación .....	26
Filtros.....	27
Rutas.....	32
Controladores.....	40
Control de autenticación.....	40
Controlador base .....	42
Controlador de curso .....	43
Pagina principal .....	50
Controlador de perfiles .....	51
Controlador de preguntas.....	52
Controlador de recursos (tareas, exámenes, etc.) .....	57
Controlador de pruebas .....	73
Modelos.....	80
Respuestas .....	80
Cursos.....	80
Recurso “archivo” .....	81
Kardex.....	82
Preguntas .....	82
Recursos .....	83
Examen.....	84
Recurso “URL” .....	90
Usuario .....	90
Recurso “Video” .....	93
Vistas .....	94
Autenticacion .....	94
Inicar sesión.....	94
Registro .....	95
Bienvenida.....	95
Cursos.....	96

Añadir .....	96
Editar .....	97
Listas .....	97
Administrador .....	100
Calificaciones .....	108
Registros .....	109
Vista .....	111
Exámenes .....	113
Añadir .....	113
Editar .....	114
Información .....	115
Administrador .....	116
Publicar .....	119
Calificaciones .....	121
Vista .....	122
Macros .....	129
Depuración .....	129
Campo .....	130
Barra lateral .....	131
Perfiles .....	132
Editar .....	132
Vista .....	133
Plantillas .....	134
Por defecto .....	134
Invitado .....	139
Preguntas .....	142
Añadir .....	142
Editar .....	145
Recursos .....	148
Añadir .....	148
Editar .....	149

Vista.....	151
Principal.....	152
Kardex.....	154
Conclusión .....	155
Bibliografía .....	155

## Introducción

Este documento contiene toda la información sobre los recursos de la plataforma "SIAD" (Sistema Interactivo de Aprendizaje a Distancia) ya que llevan una descripción muy específica sobre las características físicas y técnicas de cada elemento: ya sea el manejo de cursos, de estudiantes, calificaciones, etc.

El SIAD surge de la gran plusvalía del internet además de que es más sencillo que un estudiante pueda tener acceso a este por lo tanto se decidió hacer esta plataforma para que pueda haber aprendizaje sea donde sea

# Organización

---

## Descripción

El sistema funciona mediante la alta de alumnos y maestros ya sean uno u otro podrán tener diferentes accesos a los cursos es decir si se es un maestro este podrá agregar un recurso al mismo o por ejemplo de igual manera un examen. Pero en dado caso que sea un estudiante este podrá ver mas no editar los recursos además de hacer los exámenes y no editarlos como lo haría el profesor.

## Antecedentes

Nuestro sistema está basado en la plataforma "moodle" ya que de igual manera nos sirve para la educación en línea, nuestro sistema puede contener tantos alumnos y profesores es decir un usuario puede agregar cursos o participar en ellos.

## Misión y visión

Nuestra empresa es para poder conectar gracias a internet todos los cursos posibles para que podamos realizar una serie de cursos y que todo el mundo pueda aprender más acerca de carreras tecnológicas o más si es así como lo desea. Nuestra empresa esta para todo el mundo que le guste conocer más allá de los rasgos avanzados, intermedios o más básicos de algún tema en específico o bien si esta persona es muy buena él/ella misma puede iniciar un curso en la página del "SIAD" para poder compartir sus conocimientos con todo aquella persona que le interese.

## Objetivos generales

Nuestro principal objetivo es la unión de los diferentes tipos de cursos en un solo lugar para esto gracias a todos los usuarios que se vayan registrando pueden ir abriendo algún curso de algún tema que deseen enseñar.

# Sistema

---

## Descripción y Delimitación del sistema

La plataforma SIAD se encargara de cursos en línea para que el estudiante pueda aprender sin tener que salir de casa además esta maneja a los usuarios tanto como alumnos como maestros mostrando un kardex con los diferentes cursos que han tomado hasta la fecha junto con sus respectivas calificaciones del mismo además de que se encargara en dado caso de querer agregar un curso, se podrán agregar recursos para el completo entendimiento del tema y por supuesto se podrán agregar exámenes a los alumnos para que se pueda comprobar si realmente han quedado claros todos los temas, en este se pueden agregar preguntas del tipo selección, selección múltiple, pregunta abierta y una vez publicado el examen aparecerán las calificaciones de los alumnos. En dado caso que sea el estudiante podrá revisar todos los recursos impartidos por el maestro y hacer el examen debido al curso una vez finalizado el examen nos dará las calificaciones que hemos tenido en los exámenes de ese curso para poder tener un control de las mismas.

## Objetivos generales

Brindar un servicio de cursos en línea el cual sean definidos por los mismos usuarios es decir pueden ser tanto alumnos como maestros si les gustaría compartir el conocimiento que ellos tienen.

## Descripción de usuarios

La plataforma SIAD dispondrá de dos tipos de usuarios: los alumnos y los maestros, los alumnos se convertirán en maestros una vez abran un curso

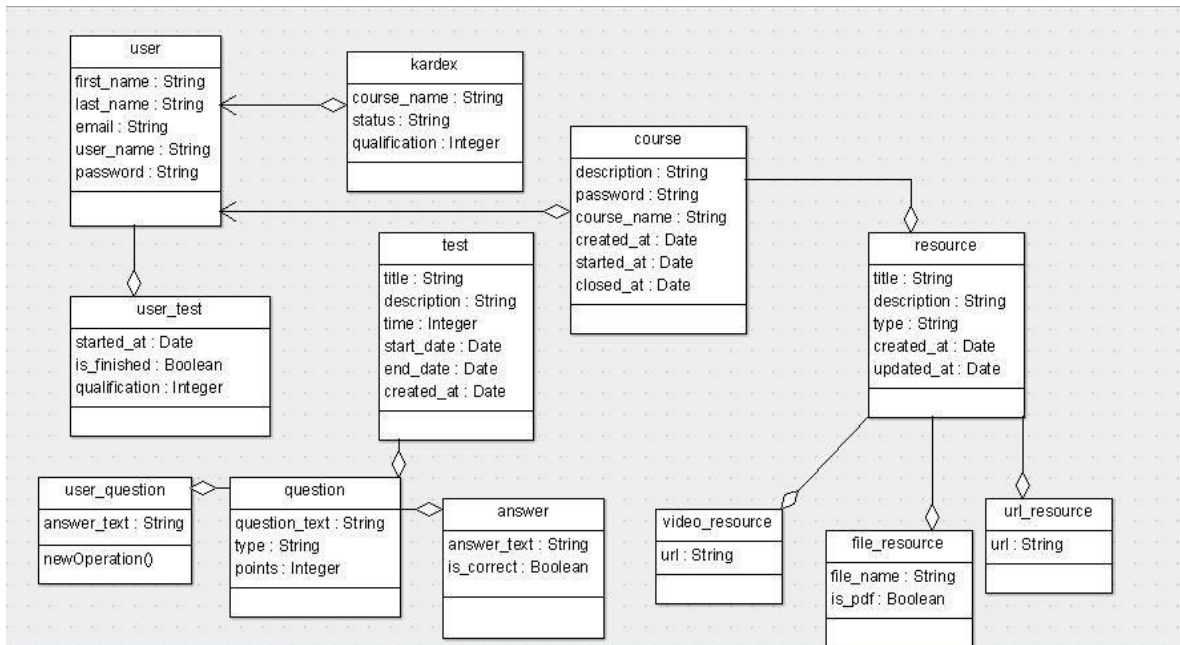
**Maestros.-** Podrán hacer manejo de los diferentes tipos de recursos que irán publicando durante el transcurso del curso. Además tendrán un control para llevar a cabo algún examen es decir podrán crear sus propios exámenes ya sea con preguntas de selección o de selección múltiple o bien preguntas abiertas puede publicarlo a la fecha y hora que el desee además de que el maestro podrá finalizar el curso cuando este lo desee una vez hecho esto los alumnos recibirán sus calificaciones en el kardex.

**Alumnos.-** Los alumnos inscritos en algún curso podrán ver pero no modificar los recursos disponibles en los mismos y a la hora de poder hacer un examen aparecerá un reloj el cual será el

límite de tiempo que tienen para hacerlo, una vez finalizado el examen podrán ver su calificación y una vez finalizado un curso los mismos alumnos podrán entrar al kardex y podrán ver sus calificaciones.

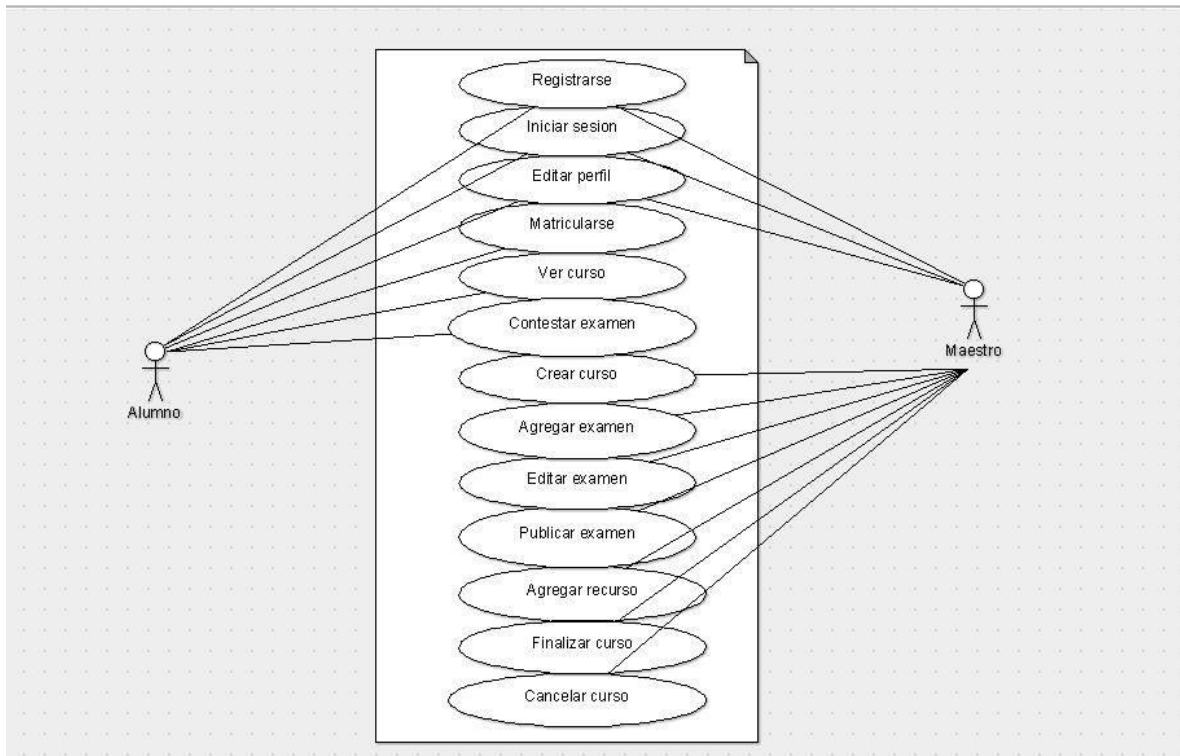
# Especificación de requerimientos

## Diagrama de clases

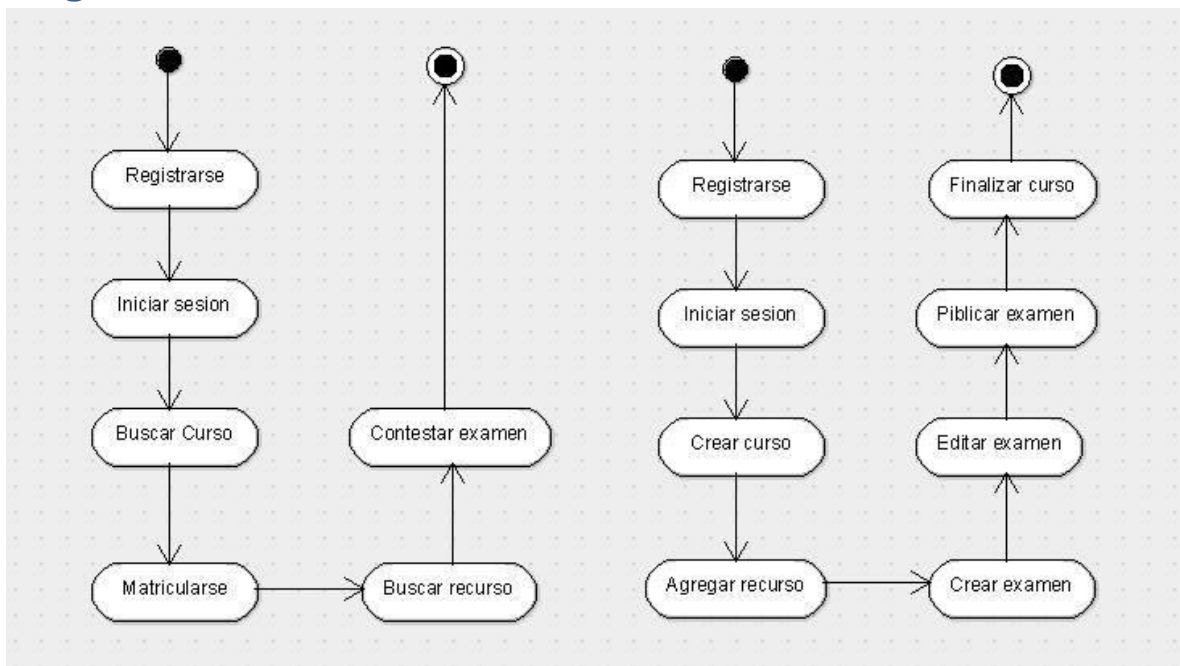




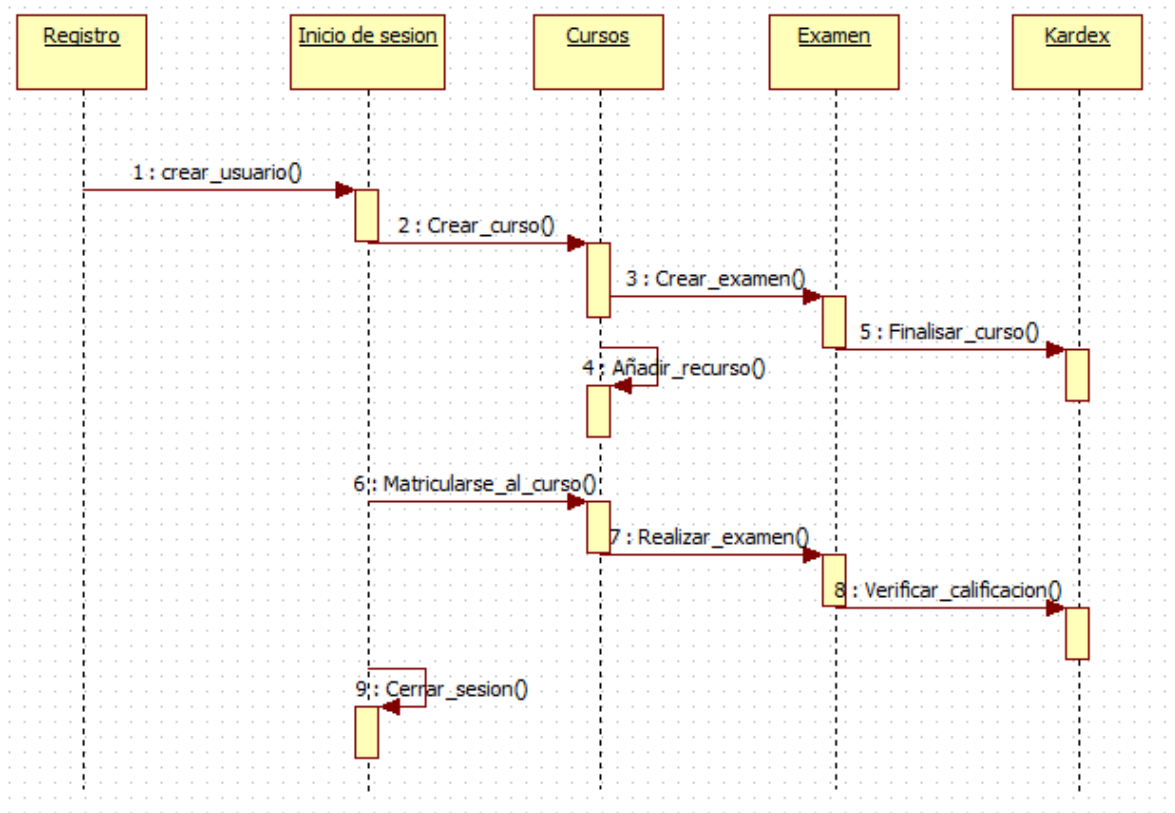
## Diagrama de casos de uso



## Diagrama de actividades

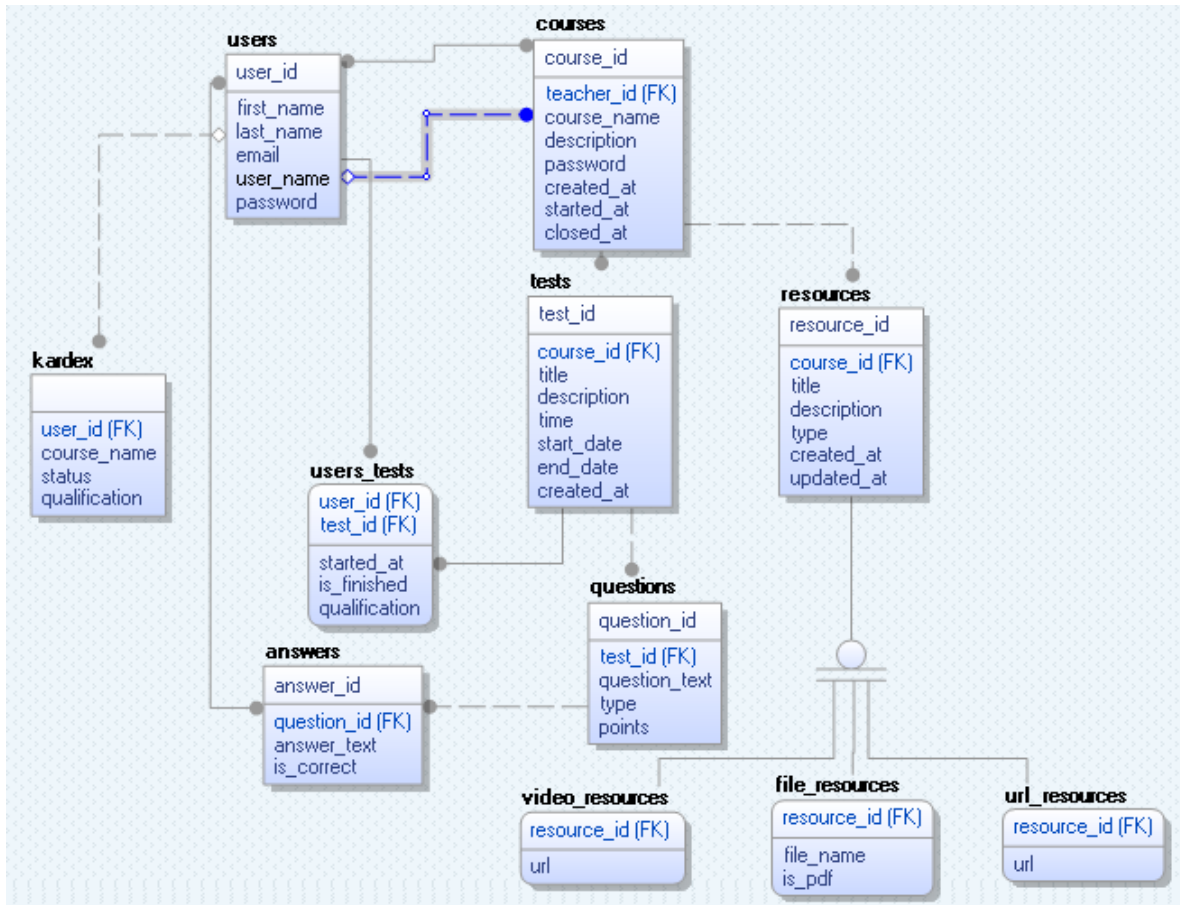


## Diagramas de secuencia



# Diseño de la base de datos

## Modelado de la base de datos con Erwin



## Esquema de la base de datos

```
-- phpMyAdmin SQL Dump
-- version 4.0.4.1
-- http://www.phpmyadmin.net
--
-- Servidor: 127.0.0.1
-- Tiempo de generaci3n: 26-11-2013 a las 03:40:36
-- Versi3n del servidor: 5.5.32
-- Versi3n de PHP: 5.4.19

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8 */;

--
-- Base de datos: `siad`
--
CREATE DATABASE IF NOT EXISTS `siad` DEFAULT CHARACTER SET utf8
COLLATE utf8_unicode_ci;
USE `siad`;

--
-- Estructura de tabla para la tabla `answers`
--

DROP TABLE IF EXISTS `answers`;
CREATE TABLE IF NOT EXISTS `answers` (
  `answer_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `question_id` int(10) unsigned NOT NULL,
  `answer_text` text COLLATE utf8_unicode_ci NOT NULL,
  `is_correct` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`answer_id`),
  KEY `answers_question_id_foreign` (`question_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=46 ;

--
-- Estructura de tabla para la tabla `courses`
--
```

```

DROP TABLE IF EXISTS `courses`;
CREATE TABLE IF NOT EXISTS `courses` (
  `course_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `teacher_id` int(10) unsigned NOT NULL,
  `course_name` varchar(120) COLLATE utf8_unicode_ci NOT NULL,
  `description` text COLLATE utf8_unicode_ci NOT NULL,
  `password` varchar(20) COLLATE utf8_unicode_ci DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
  `started_at` timestamp NULL DEFAULT NULL,
  `closed_at` timestamp NULL DEFAULT NULL,
  PRIMARY KEY (`course_id`),
  KEY `courses_teacher_id_foreign` (`teacher_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
  AUTO_INCREMENT=9 ;

```

```

-- -----

```

```

--

```

```

-- Estructura de tabla para la tabla `file_resources`
--

```

```

DROP TABLE IF EXISTS `file_resources`;
CREATE TABLE IF NOT EXISTS `file_resources` (
  `resource_id` int(10) unsigned NOT NULL,
  `file_name` text COLLATE utf8_unicode_ci NOT NULL,
  `is_pdf` tinyint(1) NOT NULL DEFAULT '0',
  PRIMARY KEY (`resource_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

```

-- -----

```

```

--

```

```

-- Estructura de tabla para la tabla `kardex`
--

```

```

DROP TABLE IF EXISTS `kardex`;
CREATE TABLE IF NOT EXISTS `kardex` (
  `user_id` int(10) unsigned NOT NULL,
  `course_name` varchar(50) COLLATE utf8_unicode_ci NOT NULL,
  `status` varchar(40) COLLATE utf8_unicode_ci NOT NULL,
  `qualification` double DEFAULT NULL,
  KEY `kardex_user_id_foreign` (`user_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

```

-- -----

```

```

--

```

```

-- Estructura de tabla para la tabla `migrations`
--

```

```

--

```

```

DROP TABLE IF EXISTS `migrations`;
CREATE TABLE IF NOT EXISTS `migrations` (
  `migration` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `batch` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

-- -----

--
-- Estructura de tabla para la tabla `questions`
--

DROP TABLE IF EXISTS `questions`;
CREATE TABLE IF NOT EXISTS `questions` (
  `question_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `test_id` int(10) unsigned NOT NULL,
  `question_text` text COLLATE utf8_unicode_ci NOT NULL,
  `type` enum('open','choice','multiple') COLLATE utf8_unicode_ci
NOT NULL DEFAULT 'open',
  `points` int(10) unsigned NOT NULL,
  PRIMARY KEY (`question_id`),
  KEY `questions_test_id_foreign` (`test_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=13 ;

-- -----

--
-- Estructura de tabla para la tabla `resources`
--

DROP TABLE IF EXISTS `resources`;
CREATE TABLE IF NOT EXISTS `resources` (
  `resource_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `course_id` int(10) unsigned NOT NULL,
  `title` varchar(120) COLLATE utf8_unicode_ci NOT NULL,
  `description` text COLLATE utf8_unicode_ci,
  `type` enum('video','file','url') COLLATE utf8_unicode_ci NOT
NULL DEFAULT 'url',
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
  `updated_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
  PRIMARY KEY (`resource_id`),
  KEY `resources_course_id_foreign` (`course_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=11 ;

-- -----

--
-- Estructura de tabla para la tabla `tests`

```

```
--
DROP TABLE IF EXISTS `tests`;
CREATE TABLE IF NOT EXISTS `tests` (
  `test_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `course_id` int(10) unsigned NOT NULL,
  `title` varchar(120) COLLATE utf8_unicode_ci NOT NULL,
  `description` text COLLATE utf8_unicode_ci,
  `time` int(10) unsigned NOT NULL,
  `start_date` timestamp NULL DEFAULT NULL,
  `end_date` timestamp NULL DEFAULT NULL,
  `created_at` timestamp NOT NULL DEFAULT '0000-00-00 00:00:00',
  PRIMARY KEY (`test_id`),
  KEY `tests_course_id_foreign` (`course_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=12 ;
```

```
-- -----
--
-- Estructura de tabla para la tabla `url_resources`
--
```

```
DROP TABLE IF EXISTS `url_resources`;
CREATE TABLE IF NOT EXISTS `url_resources` (
  `resource_id` int(10) unsigned NOT NULL,
  `url` text COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`resource_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;
```

```
-- -----
--
-- Estructura de tabla para la tabla `users`
--
```

```
DROP TABLE IF EXISTS `users`;
CREATE TABLE IF NOT EXISTS `users` (
  `user_id` int(10) unsigned NOT NULL AUTO_INCREMENT,
  `first_name` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `last_name` varchar(20) COLLATE utf8_unicode_ci NOT NULL,
  `email` varchar(255) COLLATE utf8_unicode_ci NOT NULL,
  `user_name` varchar(15) COLLATE utf8_unicode_ci NOT NULL,
  `password` varchar(60) COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`user_id`),
  UNIQUE KEY `users_email_unique` (`email`),
  UNIQUE KEY `users_user_name_unique` (`user_name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci
AUTO_INCREMENT=7 ;
```

```

-- -----
--
-- Estructura de tabla para la tabla `users_courses`
--

DROP TABLE IF EXISTS `users_courses`;
CREATE TABLE IF NOT EXISTS `users_courses` (
  `user_id` int(10) unsigned NOT NULL,
  `course_id` int(10) unsigned NOT NULL,
  PRIMARY KEY (`user_id`,`course_id`),
  KEY `users_courses_course_id_foreign` (`course_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

```

-- -----
--
-- Estructura de tabla para la tabla `users_questions`
--

DROP TABLE IF EXISTS `users_questions`;
CREATE TABLE IF NOT EXISTS `users_questions` (
  `user_id` int(10) unsigned NOT NULL,
  `question_id` int(10) unsigned NOT NULL,
  `answer_text` text COLLATE utf8_unicode_ci NOT NULL,
  KEY `users_questions_user_id_foreign` (`user_id`),
  KEY `users_questions_question_id_foreign` (`question_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

```

```

-- -----
--
-- Estructura de tabla para la tabla `users_tests`
--

DROP TABLE IF EXISTS `users_tests`;
CREATE TABLE IF NOT EXISTS `users_tests` (
  `user_id` int(10) unsigned NOT NULL,
  `test_id` int(10) unsigned NOT NULL,
  `started_at` timestamp NULL DEFAULT NULL,
  `is_finished` tinyint(1) NOT NULL DEFAULT '0',
  `qualification` double NOT NULL DEFAULT '0',
  PRIMARY KEY (`user_id`,`test_id`),
  KEY `users_tests_test_id_foreign` (`test_id`)
)

```



```

) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

-- -----

--
-- Estructura de tabla para la tabla `video_resources`
--

DROP TABLE IF EXISTS `video_resources`;
CREATE TABLE IF NOT EXISTS `video_resources` (
  `resource_id` int(10) unsigned NOT NULL,
  `url` text COLLATE utf8_unicode_ci NOT NULL,
  PRIMARY KEY (`resource_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8 COLLATE=utf8_unicode_ci;

--
-- Restricciones para tablas volcadas
--

--
-- Filtros para la tabla `answers`
--
ALTER TABLE `answers`
  ADD CONSTRAINT `answers_question_id_foreign` FOREIGN KEY
  (`question_id`) REFERENCES `questions` (`question_id`) ON DELETE
  CASCADE ON UPDATE CASCADE;

--
-- Filtros para la tabla `courses`
--
ALTER TABLE `courses`
  ADD CONSTRAINT `courses_teacher_id_foreign` FOREIGN KEY
  (`teacher_id`) REFERENCES `users` (`user_id`) ON DELETE CASCADE ON
  UPDATE CASCADE;

--
-- Filtros para la tabla `file_resources`
--
ALTER TABLE `file_resources`
  ADD CONSTRAINT `file_resources_resource_id_foreign` FOREIGN KEY
  (`resource_id`) REFERENCES `resources` (`resource_id`) ON DELETE
  CASCADE ON UPDATE CASCADE;

--
-- Filtros para la tabla `kardex`
--
ALTER TABLE `kardex`

```

```

    ADD CONSTRAINT `kardex_user_id_foreign` FOREIGN KEY (`user_id`)
REFERENCES `users` (`user_id`) ON DELETE CASCADE ON UPDATE
CASCADE;

--
-- Filtros para la tabla `questions`
--
ALTER TABLE `questions`
    ADD CONSTRAINT `questions_test_id_foreign` FOREIGN KEY
(`test_id`) REFERENCES `tests` (`test_id`) ON DELETE CASCADE ON
UPDATE CASCADE;

--
-- Filtros para la tabla `resources`
--
ALTER TABLE `resources`
    ADD CONSTRAINT `resources_course_id_foreign` FOREIGN KEY
(`course_id`) REFERENCES `courses` (`course_id`) ON DELETE CASCADE
ON UPDATE CASCADE;

--
-- Filtros para la tabla `tests`
--
ALTER TABLE `tests`
    ADD CONSTRAINT `tests_course_id_foreign` FOREIGN KEY
(`course_id`) REFERENCES `courses` (`course_id`) ON DELETE CASCADE
ON UPDATE CASCADE;

--
-- Filtros para la tabla `url_resources`
--
ALTER TABLE `url_resources`
    ADD CONSTRAINT `url_resources_resource_id_foreign` FOREIGN KEY
(`resource_id`) REFERENCES `resources` (`resource_id`) ON DELETE
CASCADE ON UPDATE CASCADE;

--
-- Filtros para la tabla `users_courses`
--
ALTER TABLE `users_courses`
    ADD CONSTRAINT `users_courses_course_id_foreign` FOREIGN KEY
(`course_id`) REFERENCES `courses` (`course_id`) ON DELETE CASCADE
ON UPDATE CASCADE,
    ADD CONSTRAINT `users_courses_user_id_foreign` FOREIGN KEY
(`user_id`) REFERENCES `users` (`user_id`) ON DELETE CASCADE ON
UPDATE CASCADE;

--

```

```

-- Filtros para la tabla `users_questions`
--
ALTER TABLE `users_questions`
  ADD CONSTRAINT `users_questions_question_id_foreign` FOREIGN KEY
    (`question_id`) REFERENCES `questions` (`question_id`) ON DELETE
    CASCADE ON UPDATE CASCADE,
  ADD CONSTRAINT `users_questions_user_id_foreign` FOREIGN KEY
    (`user_id`) REFERENCES `users` (`user_id`) ON DELETE CASCADE ON
    UPDATE CASCADE;

--
-- Filtros para la tabla `users_tests`
--
ALTER TABLE `users_tests`
  ADD CONSTRAINT `users_tests_test_id_foreign` FOREIGN KEY
    (`test_id`) REFERENCES `tests` (`test_id`) ON DELETE CASCADE ON
    UPDATE CASCADE,
  ADD CONSTRAINT `users_tests_user_id_foreign` FOREIGN KEY
    (`user_id`) REFERENCES `users` (`user_id`) ON DELETE CASCADE ON
    UPDATE CASCADE;

--
-- Filtros para la tabla `video_resources`
--
ALTER TABLE `video_resources`
  ADD CONSTRAINT `video_resources_resource_id_foreign` FOREIGN KEY
    (`resource_id`) REFERENCES `resources` (`resource_id`) ON DELETE
    CASCADE ON UPDATE CASCADE;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;

```

## Diseño de la interfaz de la aplicación

### Inicio

#### SIAD

Sistema Interactivo de Aprendizaje a Distancia

¿Ya eres parte de SIAD?

Inicia sesión

¿Quieres ser parte de SIAD?

Regístrate

— SIAD

### Registros

#### SIAD

Sistema Interactivo de Aprendizaje a Distancia

Nombre

Apellido

Correo

Nombre de usuario

Contraseña

Confirmar contraseña

Registrar

### Inicio de sesión

#### SIAD

Sistema Interactivo de Aprendizaje a Distancia

Nombre de usuario


Contraseña

☐ Recordarme?

Iniciar sesión

## Menú Principal

SIAD Sistema Interactivo de Aprendizaje a Distancia

**Fernando Garcia**  
[Salir](#)

[Inicio](#)  
[Perfil](#)  
[Cursos](#)  
[Kardex](#)

# ¡Bienvenido a SIAD!

SIAD es el sistema de Aprendizaje a distancia creado por y para la comunidad de estudiantes de la UACH.

**Ultimos 10 cursos**

Curso	Profesor	
CursoJava01	<b>Javier Maldonado</b> (a256970@uach.mx)	<a href="#">Matricularme</a>
Java (Frames)	<b>Dario Andres Perez Rodriguez</b> (delmosviad@hotmail.com)	<a href="#">Matricularme</a>

## Curso

**Diseño Grafico**

**Profesor** Oscar Flores

**Curso** Diseño Grafico

**Descripción** Este curso se trata de una mira acerca de como lo gráfico en cualquier producto comerciable es igual de importante o mas importante que el producto, y con este curso se le proveerá de herramientas, tips, software y experiencias para que el usuario crea sus propios logos, dibujos, etc, ya sea para una empresa inventado o para una empresa ya existente.

[Unirme](#)


## Perfil del usuario

SIAD Sistema Interactivo de Aprendizaje a Distancia

**Fernando Garcia**  
[Salir](#)

[Inicio](#)  
[Perfil](#)  
[Cursos](#)  
[Kardex](#)

**Perfil**



**Nombre:** Fernando  
**Apellido:** Garcia  
**Nombre de usuario:** miother  
**Correo:** backtrack32@hotmail.es

[Editar](#)

## Editar usuario

Editar Perfil

Nombre

Oscar

Apellido

Flores

Correo

socra5720@live.com.mx

Nueva contraseña

Confirmar contraseña


Contraseña actual

Guardar

Cancelar

## Cursos

SIAD Sistema Interactivo de Aprendizaje a Distancia



Fernando Garcia

Salir

Inicio

Perfil

Cursos

Kardex

Cursos

Cursos que imparto

Crear curso

Nombre del curso	Estudiantes	
Curso de JavaME <div>Aún no inicia</div>	0	<div>Ver</div>

Cursos en los que estoy inscrito

Nombre del curso	Nombre del maestro	
Java (Frames) <div>Aún no inicia</div>	Dario Andres Perez Rodriguez	<div>Ver</div> <div>Calificaciones</div>

## Curso que imparto

**Nuevo curso**

**Nombre**

Base de Datos

**Descripcion**

Este materia tiene como objetivo aprender las facilidades de administrar una base usando diferentes tipos de Software.

**Contraseña**


.....

Crear

Cancelar

## Kardex

SIAD Sistema Interactivo de Aprendizaje a Distancia



**Fernando Garcia**

Salir

Inicio

Perfil

Cursos

**Kardex**

**Kardex**

**Nombre:** Fernando

**Apellido:** Garcia

**Nombre de usuario:** miother

**Correo:** backtrack32@hotmail.es

Nombre del curso	Estado	Calificación
Java (Frames)	qualified	100

# Examen

SIAD Sistema Interactivo de Aprendizaje a Distancia



Dario Andres Perez Rodriguez

Salir

Inicio

Perfil

Cursos

Kardex

## Agregar examen

Titulo

Examen de objetos

Descripcion

Este examen esta basado a los diferentes componentes que tiene el IDE NetBeans con java

Agregar

Cancelar



## Agregar pregunta a un examen

SIAD Sistema Interactivo de Aprendizaje a Distancia



Dario Andres Perez Rodriguez

Salir

Inicio

Perfil

Cursos

Kardex

### Agregar Pregunta de seleccion

Texto En un JTextField obtiene caracteres tipo

Puntos 1

Respuesta

String

Nada

Es correcta



+ Agregar

✕ Eliminar

✕ Eliminar

Terminar

Cancelar

## Publicar un examen

Inicio

Perfil

Cursos

Kardex

### Publicar examen: HTML Examen 1

Fecha de inicio

2013-11-26 15:20

2013 - Nov							
Su	Mo	Tu	We	Th	Fr	Sa	
27	28	29	30	31	1	2	00:00
3	4	5	6	7	8	9	00:30
10	11	12	13	14	15	16	01:00
17	18	19	20	21	22	23	01:30
24	25	26	27	28	29	30	

Fecha de cierre

2013-11-26 16:00

2013 - Nov							
Su	Mo	Tu	We	Th	Fr	Sa	
27	28	29	30	31	1	2	16:00
3	4	5	6	7	8	9	16:30
10	11	12	13	14	15	16	17:00
17	18	19	20	21	22	23	17:30
24	25	26	27	28	29	30	18:00

Tiempo en minutos

20

Publicar

Cancelar

## Calificaciones

SIAD Sistema Interactivo de Aprendizaje a Distancia



Dario Andres Perez Rodriguez

Salir

Inicio

Perfil

Cursos

Kardex

### Calificaciones del examen: HTML Examen 1

Alumno	Calificación
Fernando Garcia	90

Regresar

## Arquitectura del sistema

Nuestro sistema está basado en un “modelo vista controlador” el cual separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario es decir, por un lado define componentes para la representación de la información, y por otro lado para la interacción del usuario, por lo tanto es perfecto para manejar nuestro sistema además de que gracias a este modelo no estaremos limitados a una sola computadora o algún solo servidor sino que dispondremos de muchas más herramientas para el correcto funcionamiento del sistema.

## Especificación de ayuda en el sistema

Nuestro sistema tiene varios objetos y/o características visuales a la hora de estar dentro de la aplicación alguna de estos ejemplos son al momento de ver los colores de los botones

Rojo.- Significa un estado de “peligro” es decir que si se confirma la acción no se puede deshacer.

Azul.- Conlleva a una acción positiva es decir “cambios” de una ventana a otra

Blanco.- es el color neutro es decir dan una acción pero no es ni positiva ni negativa

Además no solamente contamos con aquellos colores sino que también contamos con pequeños iconos justo antes de la palabra para poder conocer visualmente lo que ese botón puede hacer un ejemplo de esto es “agregar” es un botón que a su lado derecho tiene un icono el cual es una “pluma” esto puede indicar al usuario que lo que va a hacer el botón es crear algo nuevo.

## Conexión de la aplicación cliente con el servidor

Nosotros hemos utilizado Laravel es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP 5.

Además estamos utilizando un sistema de mapeo de datos relacional llamado Eloquent ORM que facilita la creación de modelos. Este ORM se funda en patrón active record y su funcionamiento es muy sencillo. Gracias a esto podemos tener una conexión de la aplicación con el servidor

## Acceso a la base de datos desde la aplicación

Nosotros hemos utilizado MySQL como acceso a la base de datos ya que es más sencillo y liviano al momento de poder utilizar el sistema además que gracias a este manejador de base de datos nos da más funciones para poder controlar y además para poder validar la base de datos y no poder tener ningún problema con ella, además es necesaria este para que el sistema funcione ya que MySQL tiene más opciones para poder personalizar fácilmente la base de datos y como se ha dicho antes es más fácil validarlo.

# Código de la aplicación

---

## Filtros

```
1 <?php
2
3 App::before(function($request) {
4 });
5
6
7 App::after(function($request, $response) {
8 });
9
10 Route::filter('auth', function() {
11     if (Auth::guest()) return Redirect::guest('login');
12 });
13
14
15 Route::filter('auth.basic', function() {
16     return Auth::basic();
17 });
18
19 Route::filter('guest', function() {
20     if (Auth::check()) return Redirect::to('/');
21 });
22
23 Route::filter('csrf', function() {
24     if (Session::token() != Input::get('_token')) {
25         throw new
26 Illuminate\Session\TokenMismatchException;
27     }
28 });
29
30 Route::filter('welcome', function() {
31     if (!Auth::check()) {
32         return Redirect::route('welcome');
33     }
34 });
35
36 Route::filter('course.teacher', function($route) {
37     $user = Auth::user();
```

```

37
38     $course = $route->getParameter('course');
39
40     if (!$user->teacherCourses->contains($course-
>course_id)) {
41         return Redirect::route('course.lists');
42     }
43 });
44 Route::filter('course.student', function($route) {
45     $user = Auth::user();
46
47     $course = $route->getParameter('course');
48
49     if (!$user->studentCourses->contains($course-
>course_id)) {
50         return Redirect::route('course.lists');
51     }
52 });
53 Route::filter('course.candidat', function($route) {
54     $user = Auth::user();
55
56     $course = $route->getParameter('course');
57
58     if ($user->teacherCourses->contains($course-
>course_id)) {
59         return Redirect::route('course.manager',
$course->course_id);
60     }
61
62     if ($user->studentCourses->contains($course-
>course_id)) {
63         return Redirect::route('course.view', $course-
>course_id);
64     }
65 });
66 Route::filter('course.available', function($route) {
67     $course = $route->getParameter('course');
68
69     if ($course->started_at) {
70         return Redirect::route('course.lists');
71     }
72 });

```

```

73 Route::filter('course.closed', function($route) {
74     $course = $route->getParameter('course');
75
76     if (NULL != $course->closed_at) {
77         Session::push('messages', array(
78             'type' => 'info',
79             'text' => "El curso '$course->course_name'
todavía no termina"
80         ));
81
82         return Redirect::route('course.lists');
83     }
84 });
85 Route::filter('course.started', function($route) {
86     $course = $route->getParameter('course');
87
88     if (!$course->started_at) {
89         Session::push('messages', array(
90             'type' => 'info',
91             'text' => "El curso '$course->course_name'
todavía no inicia"
92         ));
93
94         return Redirect::route('course.lists');
95     }
96 });
97
98 Route::filter('resource.both', function($route) {
99     $user = Auth::user();
100
101     $course = $route->getParameter('resource')->course;
102
103     if (!$user->teacherCourses->contains($course->course_id)) {
104         if (!$user->studentCourses->contains($course->course_id)) {
105             return Redirect::route('course.lists');
106         }
107
108         if (!$course->started_at) {
109             return Redirect::route('course.lists');
110         }

```

```

111     }
112 });
113 Route::filter('resource.teacher', function($route) {
114     $user = Auth::user();
115
116     $course = $route->getParameter('resource')->course;
117
118     if (!$user->teacherCourses->contains($course->course_id)) {
119         return Redirect::route('course.lists');
120     }
121 });
122 Route::filter('resource.student', function($route) {
123     $user = Auth::user();
124
125     $course = $route->getParameter('resource')->course;
126
127     if (!$user->studentCourses->contains($course->course_id)) {
128         return Redirect::route('course.lists');
129     }
130
131     if (!$course->started_at) {
132         return Redirect::route('course.lists');
133     }
134 });
135
136 Route::filter('test.teacher', function($route) {
137     $user = Auth::user();
138
139     $course = $route->getParameter('test')->course;
140
141     if (!$user->teacherCourses->contains($course->course_id)) {
142         return Redirect::route('course.lists');
143     }
144 });
145 Route::filter('test.student', function($route) {
146     $user = Auth::user();
147
148     $course = $route->getParameter('test')->course;
149

```

```

150     if (!$user->studentCourses->contains($course-
>course_id)) {
151         return Redirect::route('course.lists');
152     }
153
154     if (!$course->started_at) {
155         return Redirect::route('course.lists');
156     }
157 });
158 Route::filter('test.valid', function($route) {
159     $test = $route->getParameter('test');
160
161     if (!$test->canStart(Auth::user())) {
162         return Redirect::route('test.info', $test-
>test_id);
163     }
164 });
165 Route::filter('test.editable', function($route) {
166     $test = $route->getParameter('test');
167
168     if (!$test->isEditable()) {
169         return Redirect::route('course.manager', $test-
>course->course_id);
170     }
171 });
172 Route::filter('test.finalized', function($route) {
173     $test = $route->getParameter('test');
174
175     if (!$test->isFinalized()) {
176         return Redirect::route('course.manager', $test-
>course->course_id);
177     }
178 });
179 Route::filter('test.filled', function($route) {
180     $test = $route->getParameter('test');
181
182     if (!$test->hasQuestions()) {
183         return Redirect::route('course.manager', $test-
>course->course_id);
184     }
185 });

```



## Rutas

```
1 <?php
2
3 Route::model('user', 'User');
4 Route::model('course', 'Course');
5 Route::model('resource', 'Resource');
6 Route::model('test', 'Test');
7 Route::model('question', 'Question');
8
9 Route::group(array('prefix' => 'auth'), function() {
10     Route::get('', array(
11         'as' => 'login',
12         'before' => 'guest',
13         'uses' => 'AuthController@login'
14     ));
15     Route::get('welcome', array(
16         'as' => 'welcome',
17         'before' => 'guest',
18         'uses' => 'AuthController@welcome'
19     ));
20     Route::get('register', array(
21         'as' => 'register',
22         'before' => 'guest',
23         'uses' => 'AuthController@register'
24     ));
25     Route::post('create', array(
26         'as' => 'user.create',
27         'before' => 'guest',
28         'uses' => 'AuthController@create'
29     ));
30     Route::post('login', array(
31         'as' => 'do.login',
32         'before' => 'guest',
33         'uses' => 'AuthController@doLogin'
34     ));
35     Route::get('logout', array(
36         'as' => 'do.logout',
37         'before' => 'auth',
38         'uses' => 'AuthController@doLogout'
39     ));
40 });
41
```

```

42 Route::group(array('before' => 'welcome'), function() {
43
44     View::share('user', Auth::user());
45
46     View::composer('layouts.default', function($view) {
47         $links = array(
48             array('name' => 'Inicio', 'route' =>
'dashboard'),
49             array('name' => 'Perfil', 'route' => array(
50                 'profile.view',
51                 'profile.edit',
52             )),
53             array('name' => 'Cursos', 'route' => array(
54                 'course.lists',
55                 'course.register',
56                 'course.add',
57                 'course.edit',
58                 'course.manager',
59                 'course.view',
60                 'course.qualifications',
61                 'resource.add',
62                 'resource.edit',
63                 'resource.view',
64                 'test.add',
65                 'test.edit',
66                 'test.info',
67                 'test.view',
68                 'test.manager',
69                 'test.publish',
70                 'test.qualifications',
71                 'question.add',
72                 'question.edit',
73             )),
74             array('name' => 'Kardex', 'route' =>
'kardex'),
75         );
76
77         $view->with('links', $links);
78     });
79
80     Route::get('', array(
81         'as' => 'dashboard',

```

```

82         'uses' => 'DashboardController@dashboard'
83     ));
84
85     Route::get('kardex', array(
86         'as' => 'kardex',
87         'uses' => 'DashboardController@kardex'
88     ));
89
90     Route::group(array('prefix' => 'profile'),
91 function() {
92         Route::get('', array(
93             'as' => 'profile.view',
94             'uses' => 'ProfileController@view'
95         ));
96         Route::get('edit', array(
97             'as' => 'profile.edit',
98             'uses' => 'ProfileController@edit'
99         ));
100        Route::post('update', array(
101            'as' => 'profile.update',
102            'uses' => 'ProfileController@update'
103        ));
104    });
105
106    Route::group(array('prefix' => 'courses'),
107 function() {
108        Route::get('', array(
109            'as' => 'course.lists',
110            'uses' => 'CourseController@lists'
111        ));
112        Route::get('add', array(
113            'as' => 'course.add',
114            'uses' => 'CourseController@add'
115        ));
116        Route::post('create', array(
117            'as' => 'course.create',
118            'uses' => 'CourseController@create'
119        ));
120        Route::get('edit/{course}', array(
121            'as' => 'course.edit',
122            'before' =>
123            'course.teacher|course.available',

```

```

121         'uses' => 'CourseController@edit'
122     ));
123     Route::post('update/{course}', array(
124         'as' => 'course.update',
125         'before' =>
126         'course.teacher|course.available',
127         'uses' => 'CourseController@update'
128     ));
129     Route::get('register/{course}', array(
130         'as' => 'course.register',
131         'before' =>
132         'course.candidat|course.available',
133         'uses' => 'CourseController@register'
134     ));
135     Route::post('subscribe/{course}', array(
136         'as' => 'course.subscribe',
137         'before' =>
138         'course.candidat|course.available',
139         'uses' => 'CourseController@subscribe'
140     ));
141     Route::get('manager/{course}', array(
142         'as' => 'course.manager',
143         'before' => 'course.teacher',
144         'uses' => 'CourseController@manager'
145     ));
146     Route::get('cancel/{course}', array(
147         'as' => 'course.cancel',
148         'before' => 'course.teacher',
149         'uses' => 'CourseController@cancel'
150     ));
151     Route::get('start/{course}', array(
152         'as' => 'course.start',
153         'before' =>
154         'course.teacher|course.available',
155         'uses' => 'CourseController@start'
156     ));
157     Route::get('close/{course}', array(
158         'as' => 'course.close',
159         'before' => 'course.teacher|course.started',
160         'uses' => 'CourseController@close'
161     ));
162     Route::get('view/{course}', array(

```

```

159         'as' => 'course.view',
160         'before' => 'course.student|course.started',
161         'uses' => 'CourseController@view'
162     ));
163     Route::get('unsubscribe/{course}', array(
164         'as' => 'course.unsubscribe',
165         'before' => 'course.student|course.started',
166         'uses' => 'CourseController@unsubscribe'
167     ));
168     Route::get('qualifications/{course}', array(
169         'as' => 'course.qualifications',
170         'before' => 'course.student|course.started',
171         'uses' => 'CourseController@qualifications'
172     ));
173 });
174
175 Route::group(array('prefix' => 'resources'),
176 function() {
177     Route::get('{course}/add/{type}', array(
178         'as' => 'resource.add',
179         'before' => 'course.teacher',
180         'uses' => 'ResourceController@add'
181     ))->where('type', 'video|file|url');
182     Route::post('{course}/create', array(
183         'as' => 'resource.create',
184         'before' => 'course.teacher',
185         'uses' => 'ResourceController@create'
186     ));
187     Route::get('edit/{resource}', array(
188         'as' => 'resource.edit',
189         'before' => 'resource.teacher',
190         'uses' => 'ResourceController@edit'
191     ));
192     Route::post('update/{resource}', array(
193         'as' => 'resource.update',
194         'before' => 'resource.teacher',
195         'uses' => 'ResourceController@update'
196     ));
197     Route::get('delete/{resource}', array(
198         'as' => 'resource.delete',
199         'before' => 'resource.teacher',
200         'uses' => 'ResourceController@delete'

```

```

200     ));
201     Route::get('view/{resource}', array(
202         'as' => 'resource.view',
203         'before' => 'resource.both',
204         'uses' => 'ResourceController@view'
205     ));
206     Route::get('download/{resource}', array(
207         'as' => 'resource.download',
208         'before' => 'resource.both',
209         'uses' => 'ResourceController@download'
210     ));
211 });
212
213 Route::group(array('prefix' => 'tests'), function()
214 {
215     Route::get('{course}/add', array(
216         'as' => 'test.add',
217         'before' => 'course.teacher',
218         'uses' => 'TestController@add',
219     ));
220     Route::post('{course}/create', array(
221         'as' => 'test.create',
222         'before' => 'course.teacher',
223         'uses' => 'TestController@create',
224     ));
225     Route::get('edit/{test}', array(
226         'as' => 'test.edit',
227         'before' => 'test.teacher|test.editable',
228         'uses' => 'TestController@edit',
229     ));
230     Route::post('update/{test}', array(
231         'as' => 'test.update',
232         'before' => 'test.teacher|test.editable',
233         'uses' => 'TestController@update',
234     ));
235     Route::get('delete/{test}', array(
236         'as' => 'test.delete',
237         'before' => 'test.teacher|test.editable',
238         'uses' => 'TestController@delete',
239     ));
240     Route::get('manager/{test}', array(
241         'as' => 'test.manager',

```

```

241         'before' => 'test.teacher|test.editable',
242         'uses' => 'TestController@manager',
243     ));
244     Route::get('{course}/publish/{test}', array(
245         'as' => 'test.publish',
246         'before' =>
247         'test.teacher|test.editable|test.filled|course.started',
248         'uses' => 'TestController@publish',
249     ));
250     Route::post('{course}/publish/{test}', array(
251         'as' => 'test.do.publish',
252         'before' =>
253         'test.teacher|test.editable|test.filled|course.started',
254         'uses' => 'TestController@doPublish',
255     ));
256     Route::get('info/{test}', array(
257         'as' => 'test.info',
258         'before' => 'test.student',
259         'uses' => 'TestController@info'
260     ));
261     Route::get('view/{test}', array(
262         'as' => 'test.view',
263         'before' => 'test.student|test.valid',
264         'uses' => 'TestController@view'
265     ));
266     Route::post('submit/{test}', array(
267         'as' => 'test.submit',
268         'before' => 'test.student|test.valid',
269         'uses' => 'TestController@submit'
270     ));
271     Route::get('qualifications/{test}', array(
272         'as' => 'test.qualifications',
273         'before' => 'test.teacher|test.finalized',
274         'uses' => 'TestController@qualifications'
275     ));
276     Route::group(array('prefix' => 'questions'),
277     function() {
278         Route::get('{test}/add/{type}', array(
279             'as' => 'question.add',
280             'before' => 'test.teacher|test.editable',

```

```

280         'uses' => 'QuestionController@add'
281     ))->where('type', 'open|choice|multiple');
282     Route::post('{test}/create', array(
283         'as' => 'question.create',
284         'before' => 'test.teacher|test.editable',
285         'uses' => 'QuestionController@create'
286     ));
287     Route::get('{test}/edit/{question}', array(
288         'as' => 'question.edit',
289         'before' => 'test.teacher|test.editable',
290         'uses' => 'QuestionController@edit'
291     ));
292     Route::post('{test}/update/{question}', array(
293         'as' => 'question.update',
294         'before' => 'test.teacher|test.editable',
295         'uses' => 'QuestionController@update'
296     ));
297     Route::get('{test}/delete/{question}', array(
298         'as' => 'question.delete',
299         'before' => 'test.teacher|test.editable',
300         'uses' => 'QuestionController@delete'
301     ));
302 });
303 });

```



## Controladores

### Control de autenticación

```
1 <?php
2
3 class AuthController extends BaseController {
4
5     protected $layout = 'layouts.guest';
6
7     public function welcome() {
8         $this->layout->content =
9         View::make('auth.welcome');
10    }
11
12    public function register() {
13        $this->layout->content =
14        View::make('auth.register');
15    }
16
17    public function login() {
18        $this->layout->content =
19        View::make('auth.login');
20    }
21
22    public function create() {
23        $input = Input::only('first_name', 'last_name',
24        'email', 'user_name', 'password',
25        'password_confirmation');
26
27        $validator = Validator::make($input,
28        array(
29            'first_name' =>
30            'required|alpha|min:4|max:20',
31            'last_name' =>
32            'required|alpha|min:4|max:20',
33            'email' => "required|email|unique:users",
34            'user_name' =>
35            'required|alpha_dash|min:5|max:15|unique:users',
36            'password' =>
37            'required|confirmed|min:5|max:20'
38        )
39    );
40    }
```

```

31
32         if ($validator->fails()) {
33             return Redirect::back()-
34 >withErrors($validator)->withInput($input);
35         }
36
37         if (User::create($input)) {
38             Session::push('messages', array('type' =>
39 'info', 'text' => 'Usuario registrado correctamente!'));
40
41             return Redirect::route('login');
42         }
43
44         Session::push('messages', array('type' =>
45 'danger', 'text' => 'No se puede completar el
46 registro!'));
47
48         return Redirect::back()->withInput($input);
49     }
50
51     public function doLogin() {
52         $input = Input::only('user_name', 'password');
53
54         $validator = Validator::make($input,
55             array(
56                 'user_name' => 'required',
57                 'password' => 'required'
58             )
59         );
60
61         if ($validator->fails()) {
62             return Redirect::back()-
63 >withErrors($validator)->withInput($input);
64         }
65
66         $input['user_name'] =
67 strtolower($input['user_name']);
68
69         if (!Auth::attempt($input,
70 Input::has('remember'))) {
71             Session::push('messages', array('type' =>
72 'danger', 'text' => 'Usuario/Contraseña invalido[s]!'));

```

```

65
66         return Redirect::back()->withInput($input);
67     }
68
69     return Redirect::route('dashboard');
70 }
71
72 public function doLogout() {
73     if (Auth::logout()) {
74         return Redirect::route('welcome');
75     }
76
77     return Redirect::back();
78 }
79 }

```

### Controlador base

```

1 <?php
2
3 class BaseController extends Controller {
4
5     /**
6      * Setup the layout used by the controller.
7      *
8      * @return void
9      */
10    protected function setupLayout()
11    {
12        if ( ! is_null($this->layout))
13        {
14            $this->layout = View::make($this->layout);
15        }
16    }
17
18 }

```

## Controlador de curso

```
1 <?php
2
3 class CourseController extends BaseController {
4
5     protected $layout = 'layouts.default';
6
7     public function lists() {
8         $teacherCourses = Auth::user()->teacherCourses;
9         $studentCourses = Auth::user()->studentCourses;
10
11         $this->layout->content =
12         View::make('courses.lists')
13             ->with('teacherCourses', $teacherCourses)
14             ->with('studentCourses', $studentCourses);
15     }
16
17     public function add() {
18         $this->layout->content =
19         View::make('courses.add');
20     }
21
22     public function create() {
23         $input = Input::only('course_name',
24         'description', 'password');
25
26         $user = Auth::user();
27
28         $validator = Validator::make($input,
29         array(
30             'course_name' =>
31             'required|min:10|max:120',
32             'description' => 'max:500',
33             'password' => 'max:20'
34         )
35         );
36
37         if ($validator->fails()) {
38             return Redirect::back()-
39             >withErrors($validator)->withInput($input);
40         }
41     }
42 }
```

```

37     $input['created_at'] = Carbon::now();
38
39     $course = new Course($input);
40
41     $user->teacherCourses()->save($course);
42
43     return Redirect::route('course.manager',
$course->course_id);
44 }
45
46 public function edit(Course $course) {
47     $this->layout->content =
View::make('courses.edit')
48         ->with('course', $course);
49 }
50
51 public function update(Course $course) {
52     $input = Input::only('course_name',
'description', 'password');
53
54     $validator = Validator::make($input,
55         array(
56             'course_name' =>
'required|min:10|max:120',
57             'description' => 'max:500',
58             'password' => 'max:20'
59         )
60     );
61
62     if ($validator->fails()) {
63         return Redirect::back()-
>withErrors($validator)->withInput($input);
64     }
65
66     $course->update($input);
67
68     return Redirect::route('course.manager',
$course->course_id);
69 }
70
71 public function start(Course $course) {
72     $course->started_at = Carbon::now();

```

```

73         $course->closed_at = NULL;
74         $course->save();
75
76         return Redirect::route('course.manager',
77 $course->course_id);
78     }
79
80     public function close(Course $course) {
81         $course->students->each(function($student) use
82 ($course) {
83             $totalTests = 0;
84             $qualification = 0;
85
86             $course->tests->each(function($test) use
87 ($student, &$totalTests, &$qualification) {
88                 if ($test->hasQualification($student) ||
89 $test->isFinalized()) {
90                     $totalTests++;
91
92                     if (!$test->students-
93 >contains($student->user_id)) {
94                         $test->students()-
95 >attach($student->user_id);
96                     } else {
97                         $qualification += $test-
98 >students->find($student->user_id)->pivot-
99 >qualification;
100                 }
101             }
102         });
103
104         $student->questions-
105 >each(function($question) use ($student, $course) {
106             if ($course->course_id == $question-
107 >test->course->course_id) {
108                 $student->questions()-
109 >detach($question->question_id);
110             }
111         });
112
113         $student->tests->each(function($test) use
114 ($student, $course) {

```

```

103         if ($course->course_id == $test->course-
>course_id) {
104             $student->tests()->detach($test-
>test_id);
105         }
106     });
107
108     if ($totalTests) {
109         $qualification /= $totalTests;
110     }
111
112     Kardex::create(array(
113         'user_id' => $student->user_id,
114         'course_name' => $course->course_name,
115         'status' => $totalTests ? 'qualified' :
'completed',
116         'qualification' => $qualification
117     ));
118 });
119
120 $course->students()->detach();
121
122 $course->tests->each(function($test) {
123     $test->time = 0;
124     $test->start_date = NULL;
125     $test->end_date = NULL;
126
127     $test->save();
128 });
129
130 $course->started_at = NULL;
131 $course->closed_at = Carbon::now();
132 $course->save();
133
134 return Redirect::route('course.manager',
$course->course_id);
135 }
136
137 public function manager(Course $course) {
138     $this->layout->content =
View::make('courses.manager')
139     ->with('course', $course);

```

```

140     }
141
142     public function view(Course $course) {
143         $this->layout->content =
144         View::make('courses.view')
145             ->with('course', $course);
146     }
147
148     public function register(Course $course) {
149         $this->layout->content =
150         View::make('courses.register')
151             ->with('course', $course);
152     }
153
154     public function subscribe(Course $course) {
155         if ($course->password) {
156             $input = Input::only('password');
157
158             $validator = Validator::make($input,
159                 array(
160                     'password' => 'required',
161                 )
162             );
163
164             if ($validator->fails()) {
165                 return Redirect::back()-
166                 >withErrors($validator)->withInput($input);
167             }
168
169             if ($course->password != $input['password'])
170             {
171                 Session::push('messages', array('type'
172                 => 'danger', 'text' => 'Contraseña invalida!'));
173
174                 return Redirect::back()-
175                 >withInput($input);
176             }
177         }
178
179         $course->students()->attach(Auth::user()-
180         >user_id);
181     }

```



```

175         return Redirect::route('course.view', $course-
>course_id);
176     }
177
178     public function unsubscribe($course) {
179         $user = Auth::user();
180
181         if ($course->students()->detach($user->user_id))
182     {
183         Session::push('messages', array('type' =>
184         'info', 'text' => "Diste de baja el curso '$course-
185         >course_name'"));
186
187         $user->questions->each(function($question)
188         use ($user, $course) {
189             if ($course->course_id == $question-
190             >test->course->course_id) {
191                 $user->questions()-
192                 >detach($question->question_id);
193             }
194         });
195
196         $user->tests->each(function($test) use
197         ($user, $course) {
198             if ($course->course_id == $test->course-
199             >course_id) {
200                 $user->tests()->detach($test-
201                 >test_id);
202             }
203         });
204
205         Kardex::create(array(
206             'user_id' => $user->user_id,
207             'course_name' => $course->course_name,
208             'status' => 'unsubscribe',
209             'qualification' => 0
210         ));
211     } else {
212         Session::push('messages', array('type' =>
213         'warning', 'text' => "El curso '$course->course_name' no
214         se pudo dar de baja"));
215     }

```

```

205
206     return Redirect::route('course.lists');
207 }
208
209 public function cancel(Course $course) {
210     $resources = $course->resources;
211
212     if ($course->delete()) {
213         Session::push('messages', array('type' =>
214 'info', 'text' => "Cancelaste el curso '$course-
215 >course_name'"));
216
217         $resources->each(function($resource) {
218             if (File::exists('uploads/resources/' .
219 $resource->resource_id)) {
220                 File::delete('uploads/resources/' .
221 $resource->resource_id);
222             }
223         });
224     } else {
225         Session::push('messages', array('type' =>
226 'warning', 'text' => "El curso '$course->course_name' no
227 se pudo cancelar"));
228     }
229
230     return Redirect::route('course.lists');
231 }
232
233 public function qualifications(Course $course) {
234     $user = Auth::user();
235
236     $course->tests->each(function($test) use ($user)
237 {
238         if (!$test->students->contains($user->
239 user_id)) {
240             if ($test->isExpired($user)) {
241                 $test->students()->attach($user->
242 user_id);
243             }
244         }
245     });
246 }

```

```

238         $this->layout->content =
239         View::make('courses.qualifications')
240         ->with('course', $course);
241     }

```

## Pagina principal

```

1 <?php
2
3 class DashboardController extends BaseController {
4
5     protected $layout = 'layouts.default';
6
7     public function dashboard() {
8         $user = Auth::user();
9
10        $courses = Course::where('teacher_id', '!=',
11        $user->user_id)
12        ->whereNull('started_at')
13        ->orderBy('created_at', 'DESC')
14        ->take(10)
15        ->get();
16
17        $this->layout->content = View::make('dashboard')
18        ->with('courses', $courses);
19    }
20
21    public function kardex() {
22        $this->layout->content = View::make('kardex');
23    }
24 }

```

## Controlador de perfiles

```
1 <?php
2
3 class ProfileController extends BaseController {
4
5     protected $layout = 'layouts.default';
6
7     public function view() {
8         $this->layout->content =
9         View::make('profiles.view');
10    }
11
12    public function edit() {
13        $this->layout->content =
14        View::make('profiles.edit');
15    }
16
17    public function update() {
18        $input = Input::only('first_name', 'last_name',
19        'email', 'password', 'new_password',
20        'new_password_confirmation');
21
22        $user = Auth::user();
23
24        $validator = Validator::make($input,
25        array(
26            'first_name' => 'alpha|min:4|max:20',
27            'last_name' => 'alpha|min:4|max:20',
28            'email' =>
29            "email|unique:users,email,$user->user_id,user_id",
30            'password' => 'required',
31            'new_password' =>
32            'confirmed|min:5|max:20'
33        )
34        );
35
36        if ($validator->fails()) {
37            return Redirect::back()-
38            >withErrors($validator)->withInput($input);
39        }
40
41        if (!Hash::check($input['password'], $user-
```

```

35     >password)) {
36         Session::push('messages', array('type' =>
37             'danger', 'text' => 'Contraseña invalida!'));
38     }
39     return Redirect::back()->withInput($input);
40 }
41 if ($input['new_password']) {
42     $user->password =
43     Hash::make($input['new_password']);
44 }
45 $user->update($input);
46 return Redirect::route('profile.view');
47 }
48 }

```

### Controlador de preguntas

```

1 <?php
2
3 class QuestionController extends BaseController {
4
5     protected $layout = 'layouts.default';
6
7     public function add(Test $test, $type) {
8         $this->layout->content =
9         View::make('questions.add')
10             ->with('course', $test->course)
11             ->with('test', $test)
12             ->with('type', $type);
13     }
14
15     public function create(Test $test) {
16         $input = Input::only('question_text', 'type',
17         'points', 'answer_text', 'is_correct');
18
19         $validator = Validator::make($input, array(
20             'question_text' => 'required|min:3|max:200',
21             'points' => 'required|numeric|min:1'
22         ));
23     }
24 }

```

```

21
22         if ($validator->fails()) {
23             return Redirect::back()-
24 >withErrors($validator)->withInput($input);
25         }
26         $min = 'open' == $input['type'] ? 1 : 2;
27
28         $answerText = Input::get('answer_text',
29 array());
30         $isCorrect = Input::get('is_correct', array());
31
32         if (!is_array($answerText) ||
33 !is_array($isCorrect)) {
34             Session::push('messages', array(
35                 'type' => 'danger',
36                 'text' => 'No se puede agregar la
37 pregunta'
38             ));
39
40             return Redirect::back()->withInput($input);
41         }
42
43         if (count($answerText) != count($isCorrect)) {
44             Session::push('messages', array(
45                 'type' => 'danger',
46                 'text' => 'No se puede agregar la
47 pregunta'
48             ));
49
50             return Redirect::back()->withInput($input);
51         }
52
53         if ($min > count($answerText)) {
54             Session::push('messages', array(
55                 'type' => 'danger',
56                 'text' => "Tiene que existir por lo
57 menos $min respuesta(s)"
58             ));
59
60             return Redirect::back()->withInput($input);
61         }

```

```

57
58     $correctCount = 0;
59
60     for ($i = 0; $i < count($answerText); $i++) {
61         if (1 == $isCorrect[$i] && '' !=
62 $answerText[$i]) {
63             $correctCount++;
64         }
65     }
66
67     if (0 === $correctCount) {
68         Session::push('messages', array(
69             'type' => 'danger',
70             'text' => 'Tiene que existir por lo
71 menos una respuesta correcta'
72         ));
73
74         return Redirect::back()->withInput($input);
75     }
76
77     $question = new Question($input);
78
79     $test->questions()->save($question);
80
81     for ($i = 0; $i < count($answerText); $i++) {
82         if ('' != $answerText[$i]) {
83             $answer = new Answer(array(
84                 'answer_text' => $answerText[$i],
85                 'is_correct' => 1 == $isCorrect[$i]
86             ));
87             $question->answers()->save($answer);
88         }
89     }
90
91     return Redirect::route('test.manager', $test-
92 >test_id);
93 }
94
95     public function edit(Test $test, Question $question)
96     {
97         $this->layout->content =

```

```

View::make('questions.edit')
95     ->with('course', $test->course)
96     ->with('test', $test)
97     ->with('question', $question);
98 }
99
100     public function update(Test $test, Question
$question) {
101         $input = Input::only('question_text', 'points',
'answer_text', 'is_correct');
102
103         $validator = Validator::make($input, array(
104             'question_text' => 'required|min:3|max:200',
105             'points' => 'required|numeric|min:1'
106         ));
107
108         if ($validator->fails()) {
109             return Redirect::back()-
>withErrors($validator)->withInput($input);
110         }
111
112         $answerText = Input::get('answer_text',
array());
113         $isCorrect = Input::get('is_correct', array());
114
115         if (!is_array($answerText) ||
!is_array($isCorrect)) {
116             Session::push('messages', array(
117                 'type' => 'danger',
118                 'text' => 'No se puede agregar la
pregunta'
119             ));
120
121             return Redirect::back()->withInput($input);
122         }
123
124         if (count($answerText) !== count($isCorrect)) {
125             Session::push('messages', array(
126                 'type' => 'danger',
127                 'text' => 'No se puede agregar la
pregunta'
128             ));

```



```

129
130         return Redirect::back()->withInput($input);
131     }
132
133     if (0 === count($answerText)) {
134         Session::push('messages', array(
135             'type' => 'danger',
136             'text' => 'Tiene que existir por lo
137 menos una respuesta'
138         ));
139
140         return Redirect::back()->withInput($input);
141     }
142
143     $correctCount = 0;
144
145     for ($i = 0; $i < count($answerText); $i++) {
146         if (1 == $isCorrect[$i] && '' !=
147 $answerText[$i]) {
148             $correctCount++;
149             break;
150         }
151     }
152
153     if (0 === $correctCount) {
154         Session::push('messages', array(
155             'type' => 'danger',
156             'text' => 'Tiene que existir por lo
157 menos una respuesta correcta'
158         ));
159
160         return Redirect::back()->withInput($input);
161     }
162
163     $question->update($input);
164     $question->answers()->delete();
165
166     for ($i = 0; $i < count($answerText); $i++) {
167         $answer = new Answer(array(
168             'answer_text' => $answerText[$i],
169             'is_correct' => 1 == $isCorrect[$i]
170         ));

```

```

168
169         $question->answers()->save($answer);
170     }
171
172     return Redirect::route('test.manager', $test-
173 >test_id);
174 }
175
176 public function delete(Test $test, Question
177 $question) {
178     $question->delete();
179
180     return Redirect::route('test.manager', $test-
181 >test_id);
182 }
183 }

```

### Controlador de recursos (tareas, exámenes, etc.)

```

1 <?php
2
3 class ResourceController extends BaseController {
4
5     protected $layout = 'layouts.default';
6 }
7 <?php
8
9
10 class ResourceController extends BaseController {
11
12     protected $layout = 'layouts.default';
13
14     public function add(Course $course, $type) {
15
16         $this->layout->content = View::make('resources.add')
17             ->with('course', $course)
18             ->with('type', $type);
19     }
20 }

```

```

11     }
12
13     private function createVideoResource(Course $course) {
14         $input = Input::only('title', 'description', 'type', 'url');
15
16         $validator = Validator::make($input, array(
17             'title' => 'required|min:5|max:120',
18             'description' => 'max:500',
19             'url' => array('required', 'url',
20 "regex:/^(?:http(?:s)?://(?:www\.)?(?:youtu\..be\/|youtube\.com\/(
21
22             ));
23
24         if ($validator->fails()) {
25             return Redirect::back()->withErrors($validator)->withInput();
26         }
27
28         $resource = new Resource($input);
29
30         $course->resources()->save($resource);
31
32         $videoResource = new VideoResource($input);
33
34         $resource->toVideo()->save($videoResource);

```

```

33
34     return Redirect::route('course.manager', $course->course_id);
35 }
36
37 private function createFileResource(Course $course) {
38     $input = Input::only('title', 'description', 'type');
39     $input = array_add($input, 'file', Input::file('upload'));
40
41     $validator = Validator::make($input, array(
42         'title' => 'required|min:5|max:120',
43         'description' => 'max:500',
44         'file' => 'required|max:10240'
45     ));
46
47     if ($validator->fails()) {
48         $input = array_forget($input, 'file');
49
50         return Redirect::back()->withErrors($validator)->withInput();
51     }
52
53     $resource = new Resource($input);
54
55     $course->resources()->save($resource);

```

```

56
57     $file = Input::file('upload');
58
59     $fileResource = new FileResource(array(
60         'file_name' => $file->getClientOriginalName(),
61         'is_pdf' => 'application/pdf' === $file->getMimeType()
62     ));
63
64     if ($file->move('uploads/resources', $resource->resource_id))
65         $resource->toFile()->save($fileResource);
66     } else {
67         $resource->delete();
68     }
69
70     return Redirect::route('course.manager', $course->course_id);
71 }
72
73 private function createUrlResource(Course $course) {
74     $input = Input::only('title', 'description', 'type', 'url');
75
76     $validator = Validator::make($input, array(
77         'title' => 'required|min:5|max:120',
78         'description' => 'max:500',

```

```

79         'url' => 'required|url'
80     ));
81
82     if ($validator->fails()) {
83         return Redirect::back()->withErrors($validator)->withInput();
84     }
85
86     $resource = new Resource($input);
87
88     $course->resources()->save($resource);
89
90     $urlResource = new UrlResource($input);
91
92     $resource->toUrl()->save($urlResource);
93
94     return Redirect::route('course.manager', $course->course_id);
95 }
96
97 public function create(Course $course) {
98     if (Input::has('type')) {
99         switch (Input::get('type')) {
100             case 'video':
101                 return $this->createVideoResource($course);

```

```

102         case 'file':
103             return $this->createFileResource($course);
104         case 'url':
105             return $this->createUrlResource($course);
106     }
107 }
108
109     return Redirect::route('course.manager', $course->course_id);
110 }
111
112 public function edit(Resource $resource) {
113     $this->layout->content = View::make('resources.edit')
114         ->with('course', $resource->course)
115         ->with('resource', $resource);
116 }
117
118 private function updateVideoResource(Resource $resource) {
119     $input = Input::only('title', 'description', 'attached');
120
121     $validator = Validator::make($input, array(
122         'title' => 'required|min:5|max:120',
123         'description' => 'max:500',
124         'attached' => array('required', 'url',

```

```

125         ));
126
127         if ($validator->fails()) {
128             return Redirect::back()->withErrors($validator)->withInput();
129         }
130
131         $resource->update($input);
132
133         $resource->toVideo()->update(array(
134             'url' => Input::get('attached')
135         ));
136
137         return Redirect::route('course.manager', $resource->course->course_id);
138     }
139
140     private function updateFileResource(Resource $resource) {
141         $input = Input::only('title', 'description');
142         $input = array_add($input, 'file', Input::file('upload'));
143
144         $validator = Validator::make($input, array(
145             'title' => 'required|min:5|max:120',
146             'description' => 'max:500',

```



```
147         'file' => 'max:10240'
148     ));
149
150     if ($validator->fails()) {
151         $input = array_forget($input, 'file');
152
153         return Redirect::back()->withErrors($validator)->withInput();
154     }
155
156     $resource->update($input);
157
158     if (Input::hasFile('upload')) {
159         $file = Input::file('upload');
160
161         if (File::exists('uploads/resources/' . $resource->resource_name)) {
162             File::delete('uploads/resources/' . $resource->resource_name);
163         }
164
165         $resource->toFile()->update(array(
166             'file_name' => $file->getClientOriginalName(),
167             'is_pdf' => 'application/pdf' === $file->getMimeType() ? true : false,
168         ));
169     }
```

```

170         $file->move('uploads/resources', $resource->resource_id);
171     }
172
173     return Redirect::route('course.manager', $resource->course->course_id);
174 }
175
176 private function updateUrlResource(Resource $resource) {
177     $input = Input::only('title', 'description', 'attached');
178
179     $validator = Validator::make($input, array(
180         'title' => 'required|min:5|max:120',
181         'description' => 'max:500',
182         'attached' => 'required|url'
183     ));
184
185     if ($validator->fails()) {
186         return Redirect::back()->withErrors($validator)->withInput();
187     }
188
189     $resource->update($input);
190
191     $resource->toUrl()->update(array(
192         'url' => Input::get('attached')

```

```
193         ));
194
195         return Redirect::route('course.manager', $resource->course->course_id);
196     }
197
198     public function update(Resource $resource) {
199         switch ($resource->type) {
200             case 'video':
201                 return $this->updateVideoResource($resource);
202             case 'file':
203                 return $this->updateFileResource($resource);
204             case 'url':
205                 return $this->updateUrlResource($resource);
206         }
207     }
208
209     public function delete(Resource $resource) {
210         if (File::exists('uploads/resources/' . $resource->resource_id)) {
211             File::delete('uploads/resources/' . $resource->resource_id);
212         }
213
214         $resource->delete();
215     }
```

```

216         return Redirect::route('course.manager', $resource->course->course_id);
217     }
218
219     public function view(Resource $resource) {
220         $this->layout->content = View::make('resources.view')
221             ->with('course', $resource->course)
222             ->with('resource', $resource);
223     }
224
225     public function download(Resource $resource) {
226         if ($resource->is('file')) {
227             return Response::download('uploads/resources/' . $resource->file_name);
228         }
229
230         return Redirect::route('course.view', $resource->course->course_id);
231     }
232 }

```

```

6
7     public function add(Course $course, $type) {
8         $this->layout->content = View::make('resources.add')
9             ->with('course', $course)
10            ->with('type', $type);
11     }
12
13     private function createVideoResource(Course $course) {
14         $input = Input::only('title', 'description', 'type', 'url');
15
16         $validator = Validator::make($input, array(

```

```

17         'title' => 'required|min:5|max:120',
18         'description' => 'max:500',
19         'url' => array('required', 'url',
"regex:/^(?:http(?:s)?://(?:www\.)?(?:youtu\be\/|youtube\.com\/|
20         ));
21
22         if ($validator->fails()) {
23             return Redirect::back()->withErrors($validator)->withInput();
24         }
25
26         $resource = new Resource($input);
27
28         $course->resources()->save($resource);
29
30         $videoResource = new VideoResource($input);
31
32         $resource->toVideo()->save($videoResource);
33
34         return Redirect::route('course.manager', $course->course_id);
35     }
36
37     private function createFileResource(Course $course) {
38         $input = Input::only('title', 'description', 'type');
39         $input = array_add($input, 'file', Input::file('upload'));
40
41         $validator = Validator::make($input, array(
42             'title' => 'required|min:5|max:120',
43             'description' => 'max:500',
44             'file' => 'required|max:10240'
45         ));
46
47         if ($validator->fails()) {
48             $input = array_forget($input, 'file');
49
50             return Redirect::back()->withErrors($validator)->withInput();
51         }
52
53         $resource = new Resource($input);
54
55         $course->resources()->save($resource);
56
57         $file = Input::file('upload');

```

```

58
59     $fileResource = new FileResource(array(
60         'file_name' => $file->getClientOriginalName(),
61         'is_pdf' => 'application/pdf' === $file->getMimeType()
62     ));
63
64     if ($file->move('uploads/resources', $resource->resource_id))
65         $resource->toFile()->save($fileResource);
66     } else {
67         $resource->delete();
68     }
69
70     return Redirect::route('course.manager', $course->course_id);
71 }
72
73 private function createUrlResource(Course $course) {
74     $input = Input::only('title', 'description', 'type', 'url');
75
76     $validator = Validator::make($input, array(
77         'title' => 'required|min:5|max:120',
78         'description' => 'max:500',
79         'url' => 'required|url'
80     ));
81
82     if ($validator->fails()) {
83         return Redirect::back()->withErrors($validator)->withInput();
84     }
85
86     $resource = new Resource($input);
87
88     $course->resources()->save($resource);
89
90     $urlResource = new UrlResource($input);
91
92     $resource->toUrl()->save($urlResource);
93
94     return Redirect::route('course.manager', $course->course_id);
95 }
96
97 public function create(Course $course) {
98     if (Input::has('type')) {
99         switch (Input::get('type')) {

```

```

100         case 'video':
101             return $this->createVideoResource($course);
102         case 'file':
103             return $this->createFileResource($course);
104         case 'url':
105             return $this->createUrlResource($course);
106     }
107 }
108
109     return Redirect::route('course.manager', $course->course_id);
110 }
111
112 public function edit(Resource $resource) {
113     $this->layout->content = View::make('resources.edit')
114         ->with('course', $resource->course)
115         ->with('resource', $resource);
116 }
117
118 private function updateVideoResource(Resource $resource) {
119     $input = Input::only('title', 'description', 'attached');
120
121     $validator = Validator::make($input, array(
122         'title' => 'required|min:5|max:120',
123         'description' => 'max:500',
124         'attached' => array('required', 'url',
125             "regex:/^(?:http(?:s)?://(?:www\.)?(?:youtu\be\/|youtube\.com\/|
126             ));
127
128     if ($validator->fails()) {
129         return Redirect::back()->withErrors($validator)->withInput();
130     }
131
132     $resource->update($input);
133
134     $resource->toVideo()->update(array(
135         'url' => Input::get('attached')
136     ));
137
138     return Redirect::route('course.manager', $resource->course->course_id);
139 }
140
141 private function updateFileResource(Resource $resource) {

```

```

141 $input = Input::only('title', 'description');
142 $input = array_add($input, 'file', Input::file('upload'));
143
144 $validator = Validator::make($input, array(
145     'title' => 'required|min:5|max:120',
146     'description' => 'max:500',
147     'file' => 'max:10240'
148 ));
149
150 if ($validator->fails()) {
151     $input = array_forget($input, 'file');
152
153     return Redirect::back()->withErrors($validator)->withInput();
154 }
155
156 $resource->update($input);
157
158 if (Input::hasFile('upload')) {
159     $file = Input::file('upload');
160
161     if (File::exists('uploads/resources/' . $resource->resource_id)) {
162         File::delete('uploads/resources/' . $resource->resource_id);
163     }
164
165     $resource->toFile()->update(array(
166         'file_name' => $file->getClientOriginalName(),
167         'is_pdf' => 'application/pdf' === $file->getMimeType()
168     ));
169
170     $file->move('uploads/resources', $resource->resource_id);
171 }
172
173 return Redirect::route('course.manager', $resource->course->id);
174 }
175
176 private function updateUrlResource(Resource $resource) {
177     $input = Input::only('title', 'description', 'attached');
178
179     $validator = Validator::make($input, array(
180         'title' => 'required|min:5|max:120',
181         'description' => 'max:500',
182         'attached' => 'required|url'

```



```

183         ));
184
185         if ($validator->fails()) {
186             return Redirect::back()->withErrors($validator)->withInput();
187         }
188
189         $resource->update($input);
190
191         $resource->toUrl()->update(array(
192             'url' => Input::get('attached')
193         ));
194
195         return Redirect::route('course.manager', $resource->course->course_id);
196     }
197
198     public function update(Resource $resource) {
199         switch ($resource->type) {
200             case 'video':
201                 return $this->updateVideoResource($resource);
202             case 'file':
203                 return $this->updateFileResource($resource);
204             case 'url':
205                 return $this->updateUrlResource($resource);
206         }
207     }
208
209     public function delete(Resource $resource) {
210         if (File::exists('uploads/resources/' . $resource->resource_id)) {
211             File::delete('uploads/resources/' . $resource->resource_id);
212         }
213
214         $resource->delete();
215
216         return Redirect::route('course.manager', $resource->course->course_id);
217     }
218
219     public function view(Resource $resource) {
220         $this->layout->content = View::make('resources.view')
221             ->with('course', $resource->course)
222             ->with('resource', $resource);
223     }
224

```

```

225     public function download(Resource $resource) {
226         if ($resource->is('file')) {
227             return Response::download('uploads/resources/' . $resource->getFilename());
228         }
229
230         return Redirect::route('course.view', $resource->course->course_id);
231     }
232 }

```

## Controlador de pruebas

```

1  <?php
2
3  class TestController extends BaseController {
4
5      protected $layout = 'layouts.default';
6
7      public function add(Course $course) {
8          $this->layout->content = View::make('tests.add')
9              ->with('course', $course);
10     }
11
12     public function create(Course $course) {
13         $input = Input::only('title', 'description');
14
15         $validator = Validator::make($input, array(
16             'title' => 'required|min:5|max:120',
17             'description' => 'max:500'
18         ));
19
20         if ($validator->fails()) {
21             return Redirect::back()-
22 >withErrors($validator)->withInput($input);
23         }
24
25         $input = array_add($input, 'time', 0);
26         $input = array_add($input, 'created_at',
27 Carbon::now());
28
29         $test = new Test($input);
30
31         $course->tests()->save($test);

```

```

30
31         return Redirect::route('course.manager',
$course->course_id);
32     }
33
34     public function edit(Test $test) {
35         $this->layout->content =
View::make('tests.edit')
36             ->with('course', $test->course)
37             ->with('test', $test);
38     }
39
40     public function update(Test $test) {
41         $input = Input::only('title', 'description');
42
43         $validator = Validator::make($input, array(
44             'title' => 'required|min:5|max:120',
45             'description' => 'max:500'
46         ));
47
48         if ($validator->fails()) {
49             return Redirect::back()-
>withErrors($validator)->withInput($input);
50         }
51
52         $test->update($input);
53
54         return Redirect::route('course.manager', $test-
>course->course_id);
55     }
56
57     public function delete(Test $test) {
58         $test->delete();
59
60         return Redirect::route('course.manager', $test-
>course->course_id);
61     }
62
63     public function manager(Test $test) {
64         $this->layout->content =
View::make('tests.manager')
65             ->with('course', $test->course)

```

```

66         ->with('test', $test);
67     }
68
69     public function publish(Course $course, Test $test)
70     {
71         $this->layout->content =
72         View::make('tests.publish')
73         ->with('course', $test->course)
74         ->with('test', $test);
75     }
76
77     public function doPublish(Course $course, Test
78     $test) {
79         $input = Input::only('start_date', 'end_date',
80         'time');
81
82         $validator = Validator::make($input, array(
83             'start_date' => 'required|date',
84             'end_date' => 'required|date',
85             'time' => 'required|numeric|min:1',
86         ));
87
88         if ($validator->fails()) {
89             return Redirect::back()-
90             >withErrors($validator)->withInput($input);
91         }
92
93         $now = Carbon::now();
94         $startDate = Carbon::createFromFormat('Y-m-d
95         H:i', $input['start_date']);
96         $endDate = Carbon::createFromFormat('Y-m-d H:i',
97         $input['end_date']);
98
99         if ($now->gt($startDate)) {
100             Session::push('messages', array(
101                 'type' => 'danger',
102                 'text' => "La fecha de inicio no puede
103                 ser menor a la fecha actual."
104             ));
105
106             return Redirect::back()->withInput($input);
107         }
108     }

```

```

100
101     if ($startDate->gt($endDate)) {
102         Session::push('messages', array(
103             'type' => 'danger',
104             'text' => "La fecha de cierre no puede
ser menor a la fecha de inicio."
105         ));
106
107         return Redirect::back()->withInput($input);
108     }
109
110     if ($startDate->eq($endDate)) {
111         Session::push('messages', array(
112             'type' => 'danger',
113             'text' => "Las fechas de inicio y cierre
no pueden ser iguales."
114         ));
115
116         return Redirect::back()->withInput($input);
117     }
118
119     $test->start_date = $startDate;
120     $test->end_date = $endDate;
121     $test->time = $input['time'];
122     $test->save();
123
124     return Redirect::route('course.manager', $test-
>course->course_id);
125 }
126
127 public function info(Test $test) {
128     $user = Auth::user();
129
130     if ($test->isExpired($user)) {
131         $test->students()->attach($user->user_id);
132
133         return Redirect::route('test.info', $test-
>test_id);
134     }
135
136     $this->layout->content =
View::make('tests.info')

```

```

137         ->with('course', $test->course)
138         ->with('test', $test);
139     }
140
141     public function view(Test $test) {
142         $now = Carbon::now();
143
144         if (!$test->students->contains(Auth::user()-
145 >user_id)) {
146             $startedAt = $now->format('Y-m-d H:i:s');
147
148             $test->students()->attach(Auth::user()-
149 >user_id, array(
150                 'started_at' => $startedAt
151             ));
152         } else {
153             $startedAt = $test->students-
154 >find(Auth::user()->user_id)->pivot->started_at;
155         }
156
157         $start = Carbon::createFromFormat('Y-m-d H:i:s',
158 $startedAt);
159         $end = $start->copy()->addMinutes($test->time);
160
161         $seconds = ($test->time * 60) - $now-
162 >diffInSeconds($start);
163
164         $hours = floor($seconds / 3600);
165         $minutes = floor(($seconds - ($hours * 3600)) /
166 60);
167         $seconds = $seconds - ($hours * 3600) -
168 ($minutes * 60);
169
170         if (1 === strlen($hours)) $hours = "0$hours";
171         if (1 === strlen($minutes)) $minutes =
172 "0$minutes";
173         if (1 === strlen($seconds)) $seconds =
174 "0$seconds";
175
176         $time = "$hours:$minutes:$seconds";
177
178         $userQuestions = Auth::user()->questions-

```

```

170 >filter(function($question) use ($test) {
171         if ($test->test_id === $question->test-
172 >test_id) {
173             return $question;
174         }
175     });
176
177     $this->layout->content =
178     View::make('tests.view')
179         ->with('course', $test->course)
180         ->with('test', $test)
181         ->with('time', $time)
182         ->with('userQuestions', $userQuestions);
183     }
184
185     public function submit(Test $test) {
186         $user = Auth::user();
187
188         $user->questions->each(function($question) use
189 ($user, $test) {
190             if ($test->test_id == $question->test-
191 >test_id) {
192                 $user->questions()->detach($question-
193 >question_id);
194             }
195         });
196
197         $input = Input::all();
198
199         foreach ($input as $key => $value) {
200             if (starts_with($key, 'question')) {
201                 $questionId = substr($key,
202 strlen('question'));
203
204                 $question = Question::find($questionId);
205
206                 if ($question) {
207                     if (is_array($value)) {
208                         foreach ($value as $text) {
209                             $user->questions()-
210 >attach($questionId, array('answer_text' => $text));
211                         }
212                     }
213                 }
214             }
215         }
216     }
217 }

```

```

204         } else {
205             if ($value !== '') {
206                 $user->questions()-
>attach($questionId, array('answer_text' => $value));
207             }
208         }
209     }
210 }
211 }
212
213 if (Input::has('save')) {
214     return Redirect::back();
215 }
216
217 $user = User::find($user->user_id);
218
219 $pivot = $user->tests->find($test->test_id)-
>pivot;
220
221 $pivot->is_finished = true;
222
223 $userPoints = $test->userPoints($user);
224 $totalPoints = $test->totalPoints();
225
226 $qualification = ($userPoints / $totalPoints) *
100.0;
227
228 $pivot->qualification = $qualification;
229 $pivot->save();
230
231 return Redirect::route('test.info', $test-
>test_id);
232 }
233
234 public function qualifications(Test $test) {
235     $this->layout->content =
View::make('tests.qualifications')
236         ->with('course', $test->course)
237         ->with('test', $test);
238 }
239 }

```



## Modelos

### Respuestas

```
1 <?php
2
3 class Answer extends Eloquent {
4
5     protected $table = 'answers';
6
7     protected $primaryKey = 'answer_id';
8
9     protected $fillable = array('answer_text',
10 'is_correct');
11
12     public $timestamps = false;
13
14     public function question() {
15         return $this->belongsTo('Question');
16     }
17 }
```

### Cursos

```
1 <?php
2
3 class Course extends Eloquent {
4
5     protected $table = 'courses';
6
7     protected $primaryKey = 'course_id';
8
9     protected $fillable = array('course_name',
10 'description', 'password', 'created_at');
11
12     public $timestamps = false;
13
14     public function getTestsAttribute() {
15         return $this->tests()->getQuery()-
16 >orderBy('start_date', 'ASC')->get();
17     }
18
19     public function getResourcesAttribute() {
```

```

18         return $this->resources()->getQuery()-
>orderBy('created_at', 'ASC')->get();
19     }
20
21     public function teacher() {
22         return $this->belongsTo('User', 'teacher_id');
23     }
24
25     public function students() {
26         return $this->belongsToMany('User',
'users_courses');
27     }
28
29     public function tests() {
30         return $this->hasMany('Test');
31     }
32
33     public function resources() {
34         return $this->hasMany('Resource');
35     }
36 }

```

### Recurso “archivo”

```

1 <?php
2
3 class FileResource extends Eloquent {
4
5     protected $table = 'file_resources';
6
7     protected $primaryKey = 'resource_id';
8
9     protected $fillable = array('file_name', 'is_pdf');
10
11     public $timestamps = false;
12
13     public function resource() {
14         return $this->belongsTo('Resource');
15     }
16 }

```

## Kardex

```
1 <?php
2
3 class Kardex extends Eloquent {
4
5     protected $table = 'kardex';
6
7     protected $fillable = array('user_id', 'course_name',
8     'status', 'qualification');
9
10    public $timestamps = false;
11
12    public function student() {
13        return $this->belongsTo('User');
14    }
15 }
```

## Preguntas

```
1 <?php
2
3 class Question extends Eloquent {
4
5     protected $table = 'questions';
6
7     protected $primaryKey = 'question_id';
8
9     protected $fillable = array('question_text', 'type',
10     'points');
11
12    public $timestamps = false;
13
14    public function is($type) {
15        return $this->type === $type;
16    }
17
18    public function test() {
19        return $this->belongsTo('Test');
20    }
21
22    public function answers() {
23        return $this->hasMany('Answer');
24    }
25 }
```

```

23     }
24
25     public function students() {
26         return $this->belongsToMany('User',
27         'users_questions')->withPivot('answer_text');
28     }
29 }

```

## Recursos

```

1 <?php
2
3 class Resource extends Eloquent {
4
5     protected $table = 'resources';
6
7     protected $primaryKey = 'resource_id';
8
9     protected $fillable = array('title', 'description',
10     'type');
11
12     public $timestamps = true;
13
14     public function getAttachedAttribute() {
15         switch ($this->type) {
16             case 'video': return $this->toVideo->url;
17             case 'file': return $this->toFile->file_name;
18             case 'url': return $this->toUrl->url;
19         }
20     }
21
22     public function is($type) {
23         return $this->type === $type;
24     }
25
26     public function course() {
27         return $this->belongsTo('Course');
28     }
29
30     public function toVideo() {
31         return $this->is('video') ? $this-
32         >hasOne('VideoResource') : NULL;

```

```

31     }
32
33     public function toFile() {
34         return $this->is('file') ? $this-
>hasOne('FileResource') : NULL;
35     }
36
37     public function toUrl() {
38         return $this->is('url') ? $this-
>hasOne('UrlResource') : NULL;
39     }
40 }

```

## Examen

```

1 <?php
2
3 class Test extends Eloquent {
4
5     protected $table = 'tests';
6
7     protected $primaryKey = 'test_id';
8
9     protected $fillable = array('title', 'description',
'time', 'created_at');
10
11     public $timestamps = false;
12
13     public function isStartGreaterThanOrNow() {
14         if ($this->start_date) {
15             $now = Carbon::now();
16             $start = Carbon::createFromFormat('Y-m-d
H:i:s', $this->start_date);
17
18             if ($start->gt($now)) {
19                 return true;
20             }
21         }
22
23         return false;
24     }
25 }

```

```

26     public function isEndGreaterThanOrNow() {
27         if ($this->end_date) {
28             $now = Carbon::now();
29             $end = Carbon::createFromFormat('Y-m-d
H:i:s', $this->end_date);
30
31             if ($end->gt($now)) {
32                 return true;
33             }
34         }
35
36         return false;
37     }
38
39     public function isBetweenNow() {
40         if ($this->start_date && $this->end_date) {
41             $now = Carbon::now();
42             $start = Carbon::createFromFormat('Y-m-d
H:i:s', $this->start_date);
43             $end = Carbon::createFromFormat('Y-m-d
H:i:s', $this->end_date);
44
45             if ($now->between($start, $end)) {
46                 return true;
47             }
48         }
49
50         return false;
51     }
52
53     public function isFinalized() {
54         return $this->isPublished() && !$this-
>isEndGreaterThanOrNow();
55     }
56
57     public function isEditable() {
58         return !$this->start_date && !$this->end_date;
59     }
60
61     public function isPublished() {
62         return $this->start_date && $this->end_date;
63     }

```

```

64
65     public function hasQualification(User $user) {
66         return $user->tests->contains($this->test_id);
67     }
68
69     public function isExpired(User $user) {
70         if ($user->tests->contains($this->test_id)) {
71             return false;
72         }
73
74         if ($this->start_date && $this->end_date) {
75             $now = Carbon::now();
76             $end = Carbon::createFromFormat('Y-m-d
H:i:s', $this->end_date);
77
78             if ($now->gt($end)) {
79                 return true;
80             }
81         }
82
83         return false;
84     }
85
86     public function canStart(User $user) {
87         if ($this->start_date && $this->end_date) {
88             $now = Carbon::now();
89             $start = Carbon::createFromFormat('Y-m-d
H:i:s', $this->start_date);
90             $end = Carbon::createFromFormat('Y-m-d
H:i:s', $this->end_date);
91
92             if ($user->tests->contains($this->test_id))
93         {
94                 $pivot = $user->tests->find($this-
95 >test_id)->pivot;
96
97                 if ($pivot->is_finished) {
98                     return false;
99                 }
100
101                 if ($pivot->started_at) {
102                     $started =

```

```

Carbon::createFromFormat('Y-m-d H:i:s', $pivot-
>started_at);
101
102         if ($now->diffInMinutes($started) <
$this->time) {
103             return true;
104         }
105
106         return false;
107     }
108 }
109
110     if ($now->between($start, $end)) {
111         return true;
112     }
113 }
114
115     return false;
116 }
117
118     public function isFinished(User $user) {
119         if ($user->tests->contains($this->test_id)) {
120             $pivot = $user->tests->find($this->test_id)-
>pivot;
121
122             if ($pivot->is_finished) {
123                 return true;
124             }
125         }
126
127         return false;
128     }
129
130     public function userPoints(User $user) {
131         $points = 0;
132
133         if (!$this->isFinished($user)) {
134             return $points;
135         }
136
137         $user->questions->each(function($question) use
(&$points) {

```



```

138         if ($question->test->test_id === $this-
>test_id) {
139             $question->answers-
>each(function($answer) use ($question, &$amp;points) {
140                 if ($answer->answer_text ==
$question->pivot->answer_text) {
141                     if ($answer->is_correct) {
142                         $points += $question-
>points;
143                     } else {
144                         if ($question-
>is('multiple')) {
145                             $points -= $question-
>points;
146                         }
147                     }
148                 }
149             });
150         }
151     });
152
153     return $points;
154 }
155
156 public function totalPoints() {
157     $points = 0;
158
159     $this->questions->each(function($question) use
(&$points) {
160         if ($question->type === 'open' || $question-
>type === 'choice') {
161             $points += $question->points;
162         } else {
163             $question->answers-
>each(function($answer) use ($question, &$amp;points) {
164                 if ($answer->is_correct) {
165                     $points += $question->points;
166                 }
167             });
168         }
169     });
170 }

```

```

171         return $points;
172     }
173
174     public function hasQuestions() {
175         return 0 !== $this->questions->count();
176     }
177
178     public function getStartDateFormatAttribute() {
179         if ($this->attributes['start_date']) {
180             $date = Carbon::createFromFormat('Y-m-d
H:i:s', $this->attributes['start_date']);
181
182             return $date->formatLocalized('%A %d %B %Y')
. ' - ' . $date->format('h:i:s A');
183         }
184     }
185
186     public function getEndDateFormatAttribute($value) {
187         if ($this->attributes['end_date']) {
188             $date = Carbon::createFromFormat('Y-m-d
H:i:s', $this->attributes['end_date']);
189
190             return $date->formatLocalized('%A %d %B %Y')
. ' - ' . $date->format('h:i:s A');
191         }
192     }
193
194     public function course() {
195         return $this->belongsTo('Course');
196     }
197
198     public function questions() {
199         return $this->hasMany('Question')-
>with('answers');
200     }
201
202     public function students() {
203         return $this->belongsToMany('User',
'users_tests')->withPivot('started_at', 'is_finished',
'qualification');
204     }
205 }

```

## Recurso "URL"

```
1 <?php
2
3 class UrlResource extends Eloquent {
4
5     protected $table = 'url_resources';
6
7     protected $primaryKey = 'resource_id';
8
9     protected $fillable = array('url');
10
11     public $timestamps = false;
12
13     public function resource() {
14         return $this->belongsTo('Resource');
15     }
16 }
```

## Usuario

```
1 <?php
2
3 use Illuminate\Auth\UserInterface;
4 use Illuminate\Auth\Reminders\RemindableInterface;
5
6 class User extends Eloquent implements UserInterface,
7     RemindableInterface {
8
9     protected $table = 'users';
10
11     protected $primaryKey = 'user_id';
12
13     protected $hidden = array('password');
14
15     protected $fillable = array('first_name',
16         'last_name', 'email', 'user_name', 'password');
17
18     public $timestamps = false;
19
20     public function getAvatarAttribute() {
21         return "http://www.gravatar.com/avatar/" .
22             md5(strtolower(trim($this->email))) . "?d=mm";
23     }
24 }
```

```

20     }
21
22     public function getAnswer(Question $question) {
23         if ($this->questions->contains($question-
24 >question_id)) {
25             return $this->questions->find($question-
26 >question_id)->pivot->answer_text;
27         }
28         return '';
29     }
30     public function hasAnswer(Question $question,
31 $answer_text) {
32         foreach ($this->questions as $userQuestion) {
33             if ($question->question_id === $userQuestion-
34 >question_id) {
35                 if ($userQuestion->pivot->answer_text ==
36 $answer_text) {
37                     return true;
38                 }
39             }
40         }
41         return false;
42     }
43     public function setUsernameAttribute($value) {
44         $this->attributes['user_name'] =
45 strtolower($value);
46     }
47     public function setPasswordAttribute($value) {
48         $this->attributes['password'] =
49 Hash::make($value);
50     }
51     public function getFullNameAttribute() {
52         return $this->first_name . ' ' . $this-
53 >last_name;
54     }
55 }

```

```

54     public function getAuthIdentifier() {
55         return $this->getKey();
56     }
57
58     public function getAuthPassword() {
59         return $this->password;
60     }
61
62     public function getReminderEmail() {
63         return $this->email;
64     }
65
66     public function teacherCourses() {
67         return $this->hasMany('Course', 'teacher_id');
68     }
69
70     public function studentCourses() {
71         return $this->belongsToMany('Course',
72 'users_courses');
73     }
74
75     public function kardex() {
76         return $this->hasMany('kardex');
77     }
78
79     public function questions() {
80         return $this->belongsToMany('Question',
81 'users_questions')->withPivot('answer_text');
82     }
83
84     public function tests() {
85         return $this->belongsToMany('Test',
86 'users_tests')->withPivot('started_at', 'is_finished',
87 'qualification');
88     }
89 }

```

## Recurso “Video”

```
1 <?php
2
3 class VideoResource extends Eloquent {
4
5     protected $table = 'video_resources';
6
7     protected $primaryKey = 'resource_id';
8
9     protected $fillable = array('url');
10
11     public $timestamps = false;
12
13     public function getVideoIdAttribute() {
14         if (preg_match("/^(?:http(?:s)?:\/\/\/?)(?:www\.)?(?:youtu\.be\
15             return $id[1];
16     }
17
18     return '';
19 }
20
21 public function resource() {
22     return $this->belongsTo('Resource');
```

```
23     }
24 }
```

## Vistas

### Autenticacion

#### Inicar sesión

```
1 @section('content')
2     <form class="form-horizontal" role="form" action="{
route('do.login') }}" method="POST">
3         @include('includes.field', array('name' =>
'user_name', 'label' => 'Nombre de usuario', 'type' =>
'text'))
4         @include('includes.field', array('name' =>
'password', 'label' => 'Contraseña', 'type' =>
'password'))
5         <div class="form-group">
6             <div class="col-sm-offset-2 col-sm-10">
7                 <div class="checkbox">
8                     <label>
9                         <input type="checkbox"
name="remember"> Recordarme?
10                     </label>
11                 </div>
12             </div>
13         </div>
14         <div class="form-group">
15             <div class="col-sm-offset-2 col-sm-10">
16                 <button type="submit" class="btn btn-
primary">Iniciar sesión</button>
17             </div>
18         </div>
19     </form>
20 @stop
```

## Registro

```
1 @section('content')
2     <form class="form-horizontal" role="form" action="{{
route('do.login') }}" method="POST">
3         @include('includes.field', array('name' =>
'user_name', 'label' => 'Nombre de usuario', 'type' =>
'text'))
4         @include('includes.field', array('name' =>
'password', 'label' => 'Contraseña', 'type' =>
'password'))
5         <div class="form-group">
6             <div class="col-sm-offset-2 col-sm-10">
7                 <div class="checkbox">
8                     <label>
9                         <input type="checkbox"
name="remember"> Recordarme?
10                     </label>
11                 </div>
12             </div>
13         </div>
14         <div class="form-group">
15             <div class="col-sm-offset-2 col-sm-10">
16                 <button type="submit" class="btn btn-
primary">Iniciar sesión</button>
17             </div>
18         </div>
19     </form>
20 @stop
```

## Bienvenida

```
1 @section('content')
2     <div class="jumbotron">
3         <div class="text-center">
4             <p>¿Ya eres parte de SIAD?</p>
5             <p>
6                 <a class="btn btn-primary btn-lg btn-
block" href="{{ route('login') }}">
7                     Inicia sesión
8                 </a>
9             </p>
10            <p>¿Quieres ser parte de SIAD?</p>
```



```

11         <p>
12             <a class="btn btn-primary btn-lg btn-
block" href="{{ route('register') }}">
13                 Registrare
14             </a>
15         </p>
16     </div>
17     <blockquote align="right">
18         <small>SIAD</small>
19     </blockquote>
20 </div>
21 @stop

```

## Cursos

### Añadir

```

1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h3 class="panel-title">Nuevo curso</h3>
5         </div>
6         <form role="form" class="form-horizontal"
action="{{ route('course.create') }}" method="POST">
7             <div class="panel-body">
8                 @include('includes.field', array('name' =>
'course_name', 'label' => 'Nombre', 'type' => 'text'))
9                 @include('includes.field', array('name' =>
'description', 'label' => 'Descripcion', 'type' =>
'textarea'))
10                @include('includes.field', array('name' =>
'password', 'label' => 'Contraseña', 'type' =>
'password'))
11            </div>
12            <div class="panel-footer">
13                <button type="submit" class="btn btn-primary
btn-lg btn-block">Crear</button>
14                <a class="btn btn-default btn-lg btn-block"
href="{{ route('course.lists') }}">Cancelar</a>
15            </div>
16        </form>
17    </div>
18 @stop

```

## Editor

```
1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h3 class="panel-title">Editor curso: {{
5 $course->course_name }}</h3>
6         </div>
7         {{ Form::model($course, array('route' =>
8 array('course.update', $course->course_id))) }}
9         <div class="panel-body">
10             @include('includes.field', array('name' =>
11 'course_name', 'label' => 'Nombre', 'type' => 'text'))
12             @include('includes.field', array('name' =>
13 'description', 'label' => 'Descripcion', 'type' =>
14 'textarea'))
15             @include('includes.field', array('name' =>
16 'password', 'label' => 'Contraseña', 'type' =>
17 'password'))
18         </div>
19         <div class="panel-footer">
20             <button type="submit" class="btn btn-primary
21 btn-lg btn-block">Guardar</button>
22             <a class="btn btn-default btn-lg btn-block"
23 href="{{ route('course.manager', $course->course_id)
24 }}">Cancelar</a>
25         </div>
26         {{ Form::close() }}
27     </div>
28 @stop
```

## Listas

```
1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h1 class="panel-title">Cursos</h1>
5         </div>
6         <div class="panel-body">
7             <h2>Cursos que imparto</h2>
8             <a class="btn btn-primary" href="{{
9 route('course.add') }}">Crear curso</a>
10         <div class="table-responsive">
```

```

10         <table class="table table-striped table-
11         hover table-condensed table-bordered">
12             <thead>
13                 <tr>
14                     <th>Nombre del curso</th>
15                     <th>Estudiantes</th>
16                     <th></th>
17                 </tr>
18             </thead>
19             <tbody>
20                 @foreach ($teacherCourses as
21                 $course)
22                     <tr>
23                         <td>
24                             {{ $course->
25                             course_name }}
26                             @if (!$course->
27                             started_at)
28                                 <span
29                                 class="label label-info">Aún no inicia</span>
30                                 @endif
31                             </td>
32                             <td>{{ $course->students-
33                             count() }}</td>
34                             <td>
35                                 <a class="btn btn-
36                                 primary btn-xs" href="{{ route('course.manager', $course-
37                                 >course_id) }}">
38                                     <span
39                                     class="glyphicon glyphicon-eye-open"></span> Ver
40                                     </a>
41                             </td>
42                         </tr>
43                     @endforeach
44                 </tbody>
45             </table>
46         </div>
47         <h2>Cursos en los que estoy inscrito</h2>
48         <div class="table-responsive">
49             <table class="table table-striped table-
50             hover table-condensed table-bordered">
51                 <thead>

```

```

42         <tr>
43             <th>Nombre del curso</th>
44             <th>Nombre del maestro</th>
45             <th></th>
46         </tr>
47     </thead>
48     <tbody>
49         @foreach ($studentCourses as
$course)
50             <tr>
51                 <td>
52                     {{ $course-
>course_name }}
53                     @if (!$course-
>started_at)
54                         <span
class="label label-info">Aún no inicia</span>
55                         @endif
56                 </td>
57                 <td>{{ $course->teacher-
>full_name }}</td>
58                 <td>
59                     <a class="btn btn-
primary btn-xs" href="{{ route('course.view', $course-
>course_id) }}">
60                         <span
class="glyphicon glyphicon-eye-open"></span> Ver
61                         </a>
62                         <a class="btn btn-
info btn-xs" href="{{ route('course.qualifications',
$course->course_id) }}">
63                             <span
class="glyphicon glyphicon-list-alt"></span>
Calificaciones
64                             </a>
65                     </td>
66                 </tr>
67             @endforeach
68         </tbody>
69     </table>
70 </div>
71 </div>

```

```

72         <div class="panel-footer">
73         </div>
74     </div>
75 @stop

```

#### Administrador

```

1  @section('content')
2      <h1 class="text-center">{{ $course->course_name
3  }}</h1>
4      <div class="panel panel-primary">
5          <div class="panel-heading">
6              <h3 class="panel-title">
7                  Información
8                  <span class="pull-right">
9                      @if (!$course->started_at)
10                         <a class="btn btn-default btn-
11 xs" href="{{ route('course.edit', $course->course_id)
12 }}">
13                         <span class="glyphicon
14 glyphicon-edit"></span>
15                         Editar
16                     </a>
17                 @endif
18             </span>
19         </h3>
20     </div>
21     <div class="panel-body">
22         @if ($course->started_at)
23             @if (0 === $course->students->count())
24                 <div class="alert alert-info">
25                     Todos los estudiantes se dieron
26 de baja
27                 </div>
28             @endif
29         @else
30             <p class="lead">
31                 <small class="text-muted text-
32 primary">
33                     Comparte este enlace para
34 permitir la entrada al curso
35                 </small>

```

```

29         <input type="text" class="form-
control" value="{{ route('course.register', $course-
>course_id) }}">
30         @if ($course->description)
31         <p><b>Descripción:</b> {{
$course->description }}</p>
32         @endif
33         @if ($course->password)
34         <p><b>Contraseña:</b> {{
$course->password }}</p>
35         @endif
36         </p>
37         @if (0 === $course->students->count())
38         <p class="text-info">Actualmente el
curso no tiene estudiantes, necesitas tener por lo menos
un estudiante para comenzar el curso</p>
39         @endif
40         @endif
41         @if (0 < $course->students->count())
42         <p>
43         <button type="button" class="btn
btn-primary btn-xs" data-toggle="collapse" data-
target="#students">
44         <span class="glyphicon
glyphicon-user"></span>
45         <b>Estudiantes:</b> {{ $course-
>students->count() }}
46         </button>
47         </p>
48         <div id="students" class="table-
responsive collapse">
49         <table class="table table-striped
table-bordered table-hover table-condensed">
50             <thead>
51             <tr>
52             <th>Nombre del
estudiante</th>
53             <th>Correo</th>
54             </tr>
55             </thead>
56             <tbody>
57             @foreach ($course->students

```

```

as $student)
58                                     <tr>
59                                     <td>{{ $student-
>full_name }}</td>
60                                     <td>{{ $student-
>email }}</td>
61                                     </tr>
62                                     @endforeach
63                                 </tbody>
64                             </table>
65                         </div>
66                     @endif
67                     <div class="text-center">
68                         @if (!$course->started_at && 0 <
$course->students->count())
69                             <a class="btn btn-primary btn-lg
btn-block confirm tip" href="{{ route('course.start',
$course->course_id) }}" data-toggle="tooltip" title=""
data-original-title="Una vez iniciado el curso no se
podran inscribir mas estudiantes">
70                                 Iniciar curso
71                             </a>
72                         @endif
73                         @if ($course->started_at && !$course-
>closed_at)
74                             <a class="btn btn-primary btn-lg
btn-block confirm tip" href="{{ route('course.close',
$course->course_id) }}" data-toggle="tooltip" title=""
data-original-title="Al finalizar el curso se
calificaran a los estudiantes, posteriormente podras
reabrir el curso">
75                                 Finalizar curso
76                             </a>
77                         @endif
78                             <a class="btn btn-default btn-lg btn-
block confirm tip" href="{{ route('course.cancel',
$course->course_id) }}" data-toggle="tooltip" title=""
data-original-title="Esto eliminara totalmente el curso
del sistema">
79                                 Cancelar curso
80                             </a>
81                         </div>

```

```

82         </div>
83     </div>
84     <div class="panel panel-primary">
85         <div class="panel-heading">
86             <h3 class="panel-title">
87                 Exámenes
88                 <span class="pull-right">
89                     <a class="btn btn-default btn-xs"
href="{{ route('test.add', $course->course_id) }}">
90                         <span class="glyphicon
glyphicon-pencil"></span>
91                         Agregar
92                     </a>
93                 </span>
94             </h3>
95         </div>
96         <table class="table table-striped table-bordered
table-hover table-condensed">
97             <thead>
98                 <tr>
99                     <th>Titulo</th>
100                    <th>Estado</th>
101                    <th></th>
102                </tr>
103            </thead>
104            <tbody>
105                @foreach ($course->tests as $test)
106                    <tr>
107                        <td>{{ $test->title }}</td>
108                        <td>
109                            @if ($test->isPublished())
110                                Publicado
111                                <p><b>Inicio:</b> {{
$test->start_date_format }}</p>
112                                <p><b>Cierre:</b> {{
$test->end_date_format }}</p>
113                            @endif
114                            @if ($test->isEditable())
115                                Editable
116                            @endif
117                            @if ($test->isFinalized())
118                                Finalizado

```



```

119         @endif
120     </td>
121     <td>
122         @if ($test->isEditable())
123             <a class="btn btn-
primary btn-xs" href="{{ route('test.edit', $test-
>test_id) }}">
124                 <span
class="glyphicon glyphicon-edit"></span> Editar
125             </a>
126             <a class="btn btn-
primary btn-xs" href="{{ route('test.manager', $test-
>test_id) }}">
127                 <span
class="glyphicon glyphicon-edit"></span> Administrar
128             </a>
129             @if ($test-
>hasQuestions() && $course->started_at)
130                 <a class="btn btn-
primary btn-xs" href="{{ route('test.publish',
array($course->course_id, $test->test_id) }}">
131                     <span
class="glyphicon glyphicon-send"></span> Publicar
132                 </a>
133             @endif
134             <a class="btn btn-danger
btn-xs confirm" href="{{ route('test.delete', $test-
>test_id) }}">
135                 <span
class="glyphicon glyphicon-remove"></span> Eliminar
136             </a>
137         @endif
138         @if ($test->isFinalized())
139             <a class="btn btn-
default btn-xs" href="{{ route('test.qualifications',
$test->test_id) }}">
140                 <span
class="glyphicon glyphicon-list-alt"></span>
Calificaciones
141             </a>
142         @endif
143     </td>

```

```

144         </tr>
145         @endforeach
146     </tbody>
147 </table>
148 <div class="panel-footer">
149 </div>
150 </div>
151 <div class="panel panel-primary">
152     <div class="panel-heading">
153         <h3 class="panel-title">
154             Recursos
155             <span class="pull-right">
156                 <div class="btn-group">
157                     <button type="button" class="btn
158 btn-default btn-xs dropdown-toggle" data-
159 toggle="dropdown">
160                         <span class="glyphicon
161 glyphicon-pencil"></span>
162                         Agregar <span
163 class="caret"></span>
164                     </button>
165                     <ul class="dropdown-menu pull-
166 right" role="menu">
167                         <li>
168                             <a href="{
169 route('resource.add', array($course->course_id,
170 'video')) }}">
171                                 <span
172 class="glyphicon glyphicon-play"></span>
173                                 Video
174                                 </a>
175                         </li>
176                         <li>
177                             <a href="{
178 route('resource.add', array($course->course_id, 'file'))
179 }}">
180                                 <span
181 class="glyphicon glyphicon-file"></span>
182                                 Archivo
183                                 </a>
184                         </li>
185                     </ul>
186                 </div>
187             </span>
188         </h3>
189     </div>
190 </div>

```

```

175         <a href="{{
route('resource.add', array($course->course_id, 'url'))
}}">
176         <span
class="glyphicon glyphicon-link"></span>
177         Enlace
178     </a>
179 </li>
180 </ul>
181 </div>
182 </span>
183 </h3>
184 </div>
185 <table class="table table-striped table-bordered
table-hover table-condensed">
186     <thead>
187     <tr>
188         <th>Titulo</th>
189         <th>Tipo</th>
190         <th></th>
191     </tr>
192 </thead>
193 <tbody>
194     @foreach ($course->resources as
$resource)
195         <tr>
196             <td>{{ $resource->title }}</td>
197             <td>
198                 @if ($resource->type ===
'video')
199                     <span class="glyphicon
glyphicon-play"></span>
200                     Video
201                 @endif
202                 @if ($resource->type ===
'file')
203                     <span class="glyphicon
glyphicon-file"></span> Archivo
204                 @endif
205                 @if ($resource->type ===
'url')
206                     <span class="glyphicon

```

```

207 glyphicon-link"></span> Enlace
208                               @endif
209                               </td>
210                               <td>
211                                   <a class="btn btn-primary
btn-xs" href="{{ route('resource.edit', $resource-
>resource_id) }}">
212                                       <span class="glyphicon
glyphicon-edit"></span> Editar
213                                       </a>
214                                       <a class="btn btn-default
btn-xs" href="{{ route('resource.view', $resource-
>resource_id) }}">
215                                           <span class="glyphicon
glyphicon-eye-open"></span> Ver
216                                           </a>
217                                           <a class="btn btn-danger
btn-xs confirm" href="{{ route('resource.delete',
$resource->resource_id) }}">
218                                               <span class="glyphicon
glyphicon-remove"></span> Eliminar
219                                               </a>
220                                           </td>
221                                       </tr>
222                                   @endforeach
223                               </tbody>
224                               </table>
225                               <div class="panel-footer">
226                                   <div>
227 @stop

```

## Calificaciones

```
1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h1 class="panel-title">Calificaciones</h1>
5         </div>
6         <div class="panel-body">
7             <dl class="dl-horizontal">
8                 <dt>Profesor</dt>
9                 <dd>{{ $course->teacher->full_name
10 }}</dd>
11                 <dt>Curso</dt>
12                 <dd>{{ $course->course_name }}</dd>
13                 <dt>Descripción</dt>
14                 <dd>{{ $course->description }}</dd>
15             </dl>
16             <div class="table-responsive">
17                 <table class="table table-striped table-
18 hover table-condensed table-bordered">
19                     <thead>
20                         <tr>
21                             <th>Examen</th>
22                             <th>Fecha</th>
23                             <th>Calificación</th>
24                         </tr>
25                     </thead>
26                     <tbody>
27                         @foreach ($course->tests as
28 $test)
29                             @if ($test-
30 >hasQualification($user) || $test->isFinalized())
31                                 <tr>
32                                     <td>{{ $test->title
33 }}</td>
34                                     <td>
35                                         <p><b>Inicio</b>
36                                         {{ $test->start_date_format }}</p>
37                                         <p><b>Fin</b> {{
38 $test->end_date_format }}</p>
39                                     </td>
40                                 </tr>
41                             @endif ($test-
```

```

35 >students->contains($user->user_id))
round($test->students->find($user->user_id)->pivot-
>qualification, 2) }}
36 @else
37 0
38 @endif
39 </td>
40 </tr>
41 @endif
42 @endforeach
43 </tbody>
44 </table>
45 </div>
46 </div>
47 <div class="panel-footer">
48 </div>
49 </div>
50 @stop

```

## Registros

```

1 @section('content')
2 <div class="panel panel-primary">
3 <div class="panel-heading">
4 <h1 class="panel-title">{{ $course-
>course_name }}</h1>
5 </div>
6 <div class="panel-body">
7 <dl class="dl-horizontal">
8 <dt>Profesor</dt>
9 <dd>{{ $course->teacher->full_name
}}</dd>
10 <dt>Curso</dt>
11 <dd>{{ $course->course_name }}</dd>
12 <dt>Descripción</dt>
13 <dd>{{ $course->description }}</dd>
14 </dl>
15 <form class="form form-inline" action="{{
route('course.subscribe', $course->course_id) }}"
method="POST">
16 @if ($course->password)

```

```

17         @if ($errors->has('password'))
18             <div class="form-group col-xs-8
19 has-error">
20                 <div class="input-group
21 input-group-sm">
22                     <span class="input-group-
23 addon">
24                         <span
25 class="glyphicon glyphicon-lock"></span>
26                         </span>
27                         <input type="password"
28 class="form-control" name="password"
29 placeholder="Contraseña">
30                     </div>
31                     <span class="help-block alert
32 alert-danger">
33                         {{ $errors-
34 >first('password') }}
35                     </span>
36                 </div>
37             @else
38                 <div class="form-group col-xs-8">
39                     <div class="input-group
40 input-group-sm">
41                         <span class="input-group-
42 addon">
43                             <span
44 class="glyphicon glyphicon-lock"></span>
45                             </span>
46                             <input type="password"
47 class="form-control" name="password"
48 placeholder="Contraseña">
49                         </div>
50                     </div>
51                 @endif
52                 <button type="submit" class="btn btn-
53 primary btn-sm">Unirme</button>
54             @else
55                 <div class="text-center">
56                     <button type="submit" class="btn
57 btn-primary btn-sm">Unirme</button>
58                 </div>

```

```

44         @endif
45     </form>
46 </div>
47 </div>
48 @stop

```

#### Vista

```

1  @section('content')
2      <div class="text-center">
3          <h1>
4              {{ $course->course_name }}
5              <a class="btn btn-warning btn-xs confirm"
href="{{ route('course.unsubscribe', $course->course_id)
6              }}">Dar de baja</a>
7          </h1>
8      </div>
9      <div class="panel panel-primary">
10         <div class="panel-heading">
11             <h3 class="panel-title">Exámenes</h3>
12         </div>
13         <div class="panel-body">
14             <div class="list-group">
15                 @foreach ($course->tests as $test)
16                     @if ($test->isPublished())
17                         <a href="{{ route('test.info', $test-
18                         >test_id) }}" class="list-group-item">
19                             <h4 class="list-group-item-
20                             heading">
21                                 {{ $test->title }}
22                                 @if ($test-
23                                 >hasQualification($user))
24                                     <span class="glyphicon
25                                     glyphicon-ok-circle"></span>
26                                 @endif
27                             </h4>
28                             <p class="list-group-item-
29                             text">{{ $test->description }}</p>
30                             </a>
31                         @endif
32                     @endforeach
33                 </div>

```



```

28         </div>
29     </div>
30     <div class="panel panel-info">
31         <div class="panel-heading">
32             <h3 class="panel-title">Recursos</h3>
33         </div>
34         <div class="panel-body">
35             <div class="list-group">
36                 @foreach ($course->resources as $resource)
37                     @if ($resource->is('url'))
38                         <a href="{{ $resource->toUrl->url }}"
39 target="_blank" class="list-group-item">
40                             <h4 class="list-group-item-
41 heading">
42                                 {{ $resource->title }}
43                             <span class="glyphicon
44 glyphicon-link"></span>
45                             </h4>
46                             <p class="list-group-item-
47 text">{{ $resource->description }}</p>
48                             </a>
49                             @endif
50                             @if ($resource->is('file'))
51                                 <a href="{{ route('resource.view',
52 $resource->resource_id) }}" target="_blank" class="list-
53 group-item">
54                                     <h4 class="list-group-item-
55 heading">
56                                         {{ $resource->title }}
57                                         @if ($resource->toFile-
58 >is_pdf)
59                                             <span class="glyphicon
60 glyphicon-file"></span>
61                                         @else
62                                             <span class="glyphicon
63 glyphicon-download"></span>
64                                         @endif
65                                     </h4>
66                                     <p class="list-group-item-
67 text">{{ $resource->description }}</p>
68                                     </a>
69                             @endif

```

```

59         @if ($resource->is('video'))
60             <a href="{ route('resource.view',
$resource->resource_id) }}" target="_blank" class="list-
group-item">
61                 <h4 class="list-group-item-
heading">
62                     {{ $resource->title }}
63                     <span class="glyphicon
glyphicon-play"></span>
64                 </h4>
65                 <p class="list-group-item-
text">{{ $resource->description }}</p>
66             </a>
67         @endif
68     @endforeach
69 </div>
70 </div>
71 </div>
72 @stop

```

## Exámenes

### Añadir

```

1  @section('content')
2      <div class="panel panel-primary">
3          <div class="panel-heading">
4              <h3 class="panel-title">Agregar examen</h3>
5          </div>
6          <form role="form" class="form-horizontal"
action="{ route('test.create', $course->course_id) }}"
method="POST">
7              <div class="panel-body">
8                  @include('includes.field', array('name' =>
'title', 'label' => 'Titulo', 'type' => 'text'))
9                  @include('includes.field', array('name' =>
'description', 'label' => 'Descripcion', 'type' =>
'textarea'))
10             </div>
11             <div class="panel-footer">
12                 <button type="submit" class="btn btn-primary
btn-lg btn-block">Agregar</button>

```

```

13         <a class="btn btn-default btn-lg btn-block"
href="{{ route('course.manager', $course->course_id)
}}">Cancelar</a>
14     </div>
15 </form>
16 </div>
17 @stop

```

#### Editor

```

1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h3 class="panel-title">Editor examen</h3>
5         </div>
6         {{ Form::model($test, array('route' =>
array('test.update', $test->test_id))) }}
7         <div class="panel-body">
8             @include('includes.field', array('name' =>
'title', 'label' => 'Titulo', 'type' => 'text'))
9             @include('includes.field', array('name' =>
'description', 'label' => 'Decripcion', 'type' =>
'textarea'))
10        </div>
11        <div class="panel-footer">
12            <button type="submit" class="btn btn-primary
btn-lg btn-block">Guardar</button>
13            <a class="btn btn-default btn-lg btn-block"
href="{{ route('course.manager', $course->course_id)
}}">Cancelar</a>
14        </div>
15        {{ Form::close() }}
16    </div>
17 @stop

```

## Informacion

```
1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h3 class="panel-title">
5                 {{ $test->title }}
6                 @if ($test->hasQualification($user))
7                     <span class="glyphicon glyphicon-ok-
circle"></span>
8                 @endif
9             </h3>
10        </div>
11        <div class="panel-body">
12            <dl class="dl-horizontal">
13                <dt>Descripción</dt>
14                <dd>{{ $test->description }}</dd>
15                <dt>Fecha de inicio</dt>
16                <dd>{{ $test->start_date_format }}</dd>
17                <dt>Fecha de cierre</dt>
18                <dd>{{ $test->end_date_format }}</dd>
19                <dt>Tiempo</dt>
20                <dd>{{ $test->time }} Minutos</dd>
21                <dt>Numero de preguntas</dt>
22                <dd>{{ $test->questions->count() }}</dd>
23            </dl>
24            @if ($test->canStart($user))
25                <p class="text-center">
26                    <a class="btn btn-lg btn-warning"
href="{{ route('test.view', $test->test_id) }}">
27                        Comenzar examen
28                    </a>
29                </p>
30            @else
31                @if ($test->isFinished($user))
32                    <table class="table table-striped
table-bordered table-hover table-condensed">
33                        <thead>
34                            <tr>
35                                <th>Puntos</th>
36                                <th>Calificacion</th>
37                            </tr>
38                        </thead>
```

```

39         <tbody>
40         <tr>
41             <td>{{ $test-
>userPoints($user) }}/{{ $test->totalPoints() }}</td>
42             <td>{{ round($user-
>tests->find($test->test_id)->pivot->qualification, 2)
}}</td>
43         </tr>
44     </tbody>
45 </table>
46 @endif
47 @endif
48 </div>
49 </div>
50 @stop

```

#### Administrador

```

1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h3 class="panel-title">
5                 Preguntas: {{ $test->questions->count()
}}
6                 <span class="pull-right">
7                     <div class="btn-group">
8                         <button type="button" class="btn
btn-default btn-xs dropdown-toggle" data-
toggle="dropdown">
9                             <span class="glyphicon
glyphicon-pencil"></span>
10                             Agregar <span
class="caret"></span>
11                             </button>
12                             <ul class="dropdown-menu pull-
right" role="menu">
13                                 <li>
14                                     <a href="{{
route('question.add', array($test->test_id, 'open')) }}">
15                                         <span
class="glyphicon glyphicon-pencil"></span>
16                                         Abierta

```

```

17         </a>
18     </li>
19     <li>
20         <a href="{
route('question.add', array($test->test_id, 'choice'))
}">
21             <span
class="glyphicon glyphicon-check"></span>
22             Seleccion
23         </a>
24     </li>
25     <li>
26         <a href="{
route('question.add', array($test->test_id, 'multiple'))
}">
27             <span
class="glyphicon glyphicon-th-list"></span>
28             Multiple
29         </a>
30     </li>
31 </ul>
32 </div>
33 </span>
34 </h3>
35 </div>
36 <table class="table table-striped table-bordered
table-hover table-condensed">
37     <thead>
38     <tr>
39         <th>Pregunta</th>
40         <th>Tipo</th>
41         <th>Puntos</th>
42         <th>Respuestas</th>
43         <th></th>
44     </tr>
45 </thead>
46 <tbody>
47     @foreach ($test->questions as $question)
48     <tr>
49         <td>{{ $question->question_text
50         <td>

```

```

51         @if ('open' === $question-
>type)
52             Abierta
53         @endif
54         @if ('choice' === $question-
>type)
55             Seleccion
56         @endif
57         @if ('multiple' ===
$question->type)
58             Multiple
59         @endif
60     </td>
61     <td>{{ $question->points }}</td>
62     <td>
63         @foreach ($question->answers
as $answer)
64             <p>
65                 {{ $answer-
>answer_text }}
66                 @if ($answer-
>is_correct)
67                     <span
class="glyphicon glyphicon-check"></span>
68                 @endif
69             </p>
70         @endforeach
71     </td>
72     <td>
73         <a class="btn btn-primary
btn-xs" href="{{ route('question.edit', array($test-
>test_id, $question->question_id)) }}">
74             <span class="glyphicon
glyphicon-edit"></span> Editar
75         </a>
76         <a class="btn btn-danger btn-
xs confirm" href="{{ route('question.delete',
array($test->test_id, $question->question_id)) }}">
77             <span class="glyphicon
glyphicon-remove"></span> Eliminar
78         </a>
79     </td>

```

```

80         </tr>
81     @endforeach
82 </tbody>
83 </table>
84 <div class="panel-body"></div>
85 <div class="panel-footer">
86     <a class="btn btn-primary btn-lg btn-block"
href="{{ route('course.manager', $course->course_id)
87     }}">Finalizar</a>
88 </div>
89 @stop

```

## Publicar

```

1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h3 class="panel-title">Publicar examen: {{
5 $test->title }}</h3>
6         </div>
7         <form role="form" class="form-horizontal"
8 action="{{ route('test.do.publish', array($course-
9 >course_id, $test->test_id)) }}" method="POST">
10         <div class="panel-body">
11             @include('includes.field', array('name' =>
12 'start_date', 'label' => 'Fecha de inicio', 'type' =>
13 'text'))
14             @include('includes.field', array('name' =>
15 'end_date', 'label' => 'Fecha de cierre', 'type' =>
16 'text'))
17             @include('includes.field', array('name' =>
18 'time', 'label' => 'Tiempo en minutos', 'type' =>
19 'number'))
20         </div>
21     <div class="panel-footer">

```



```

13         <button type="submit" class="btn btn-primary
btn-lg btn-block">Publicar</button>
14         <a class="btn btn-default btn-lg btn-block"
href="{ route('course.manager', $course->course_id)
}">Cancelar</a>
15     </div>
16 </form>
17 </div>
18 @stop
19
20 @section('css')
21     @parent
22     {{ HTML::style('css/jquery.simple-dtpicker.css') }}
23 @stop
24
25 @section('js')
26     @parent
27     {{ HTML::script('js/jquery.simple-dtpicker.js') }}
28     <script>
29         (function($) {
30             $('*[name=start_date]').appendDtpicker({"inline
e": true});
31             $('*[name=end_date]').appendDtpicker({"inline"
: true});
32         })(jQuery);

```

```
33     </script>
```

```
34 @stop
```

### Calificaciones

```
1  @section('content')
2      <div class="panel panel-primary">
3          <div class="panel-heading">
4              <h1 class="panel-title">
5                  Calificaciones del examen: {{ $test-
6              >title }}
7              </h1>
8          </div>
9          <table class="table table-striped table-hover
10             table-condensed table-bordered">
11              <thead>
12                  <tr>
13                      <th>Alumno</th>
14                      <th>Calificación</th>
15                  </tr>
16              </thead>
17              <tbody>
18                  @foreach ($course->students as $student)
19                      <tr>
20                          <td>{{ $student->full_name
21                          >}}</td>
22                          <td>
23                              @if ($student->tests-
24                              >contains($test->test_id))
25                                  {{ $student->tests-
26                                  >find($test->test_id)->pivot->qualification }}
27                              @else
28                                  0
29                              @endif
30                          </td>
31                      </tr>
32                  @endforeach
33              </tbody>
34          </table>
35          <div class="panel-footer">
```

```

32         <a class="btn btn-primary btn-lg btn-block"
href="{{ route('course.manager', $course->course_id)
}}">Regresar</a>
33     </div>
34 </div>
35 @stop

```

#### Vista

```

1 @section('content')
2     <div class="row">
3         <div class="col-lg-6 col-lg-offset-3">
4             <div id="counter"></div>
5             <div class="desc">
6                 <div>Horas</div>
7                 <div>Minutos</div>
8                 <div>Segundos</div>
9             </div>
10        </div>
11    </div>
12    <div class="panel panel-default">
13        <form role="form" class="form form-horizontal"
action="{{ route('test.submit', $test->test_id) }}"
method="POST">
14        <div class="panel-body">
15            @foreach ($test->questions as $key =>
$question)
16                <div class="row">

```

```

17         <div class="col-lg-8 col-lg-offset-2">
18             <div id="item-{{ $key }}"
19 class="item">
20                 <label>{{ $key + 1 }}.- {{
21 $question->question_text }}</label>
22                 @foreach ($question->answers as
23 $answer)
24                     @if ($question->is('open'))
25                         <div class="input-group
26 input-group-sm">
27                             <span class="input-
28 group-addon">
29                                 <span
30 class="glyphicon glyphicon-pencil"></span>
31                                 </span>
32                                 <input type="text"
33 name="question{{ $question->question_id }}" value="{{
34 $user->getAnswer($question) }}" class="form-control">
35                                 </div>
36
37                             @endif
38
39                             @if ($question->is('choice'))
40                                 <div class="input-group
41 input-group-sm">
42                                     <span class="input-
43 group-addon">
44                                         @if ($user-
45 >hasAnswer($question, $answer->answer_text))
46                                             <input
47 type="radio" name="question{{ $question->question_id }}"
48 value="{{ $answer->answer_text }}" checked>
49                                         @else
50                                             <input
51 type="radio" name="question{{ $question->question_id }}"
52 value="{{ $answer->answer_text }}">
53                                         @endif

```

```

37                                     </span>
38                                     <b class="form-
control">{{ $answer->answer_text }}</b>
39                                     </div>
40
41                                     @endif
42
43                                     @if ($question-
>is('multiple'))
44                                     <div class="input-group
input-group-sm">
45                                     <span class="input-
group-addon">
46                                     @if ($user-
>hasAnswer($question, $answer->answer_text))
47                                     <input
type="checkbox" name="question{{ $question->question_id
}}[]" value="{{ $answer->answer_text }}" checked>
48                                     @else
49                                     <input
type="checkbox" name="question{{ $question->question_id
}}[]" value="{{ $answer->answer_text }}">
50                                     @endif
51                                     </span>
52                                     <b class="form-
control">{{ $answer->answer_text }}</b>
53                                     </div>
54
55                                     @endif
56                                     @endforeach
57
58                                     </div>
59
60                                     </div>
61
62                                     </div>

```

```

57         @endforeach
58     </div>
59     <div class="panel-footer">
60         <div class="row">
61             <div class="col-xs-4-col col-xs-offset-
62 4">
63                 <div id="paginator"></div>
64             </div>
65             <button type="submit" class="btn btn-primary
66 btn-lg btn-block" name="save" value="1">Guardar sin
67 enviar</button>
68             <button type="submit" class="btn btn-default
69 btn-lg btn-block">Enviar</button>
70         </div>
71     </form>
72 </div>
73 @stop
74 @section('css')
75     @parent
76     <style type="text/css">
77         .cntSeparator {
78             font-size: 54px;

```

```
77         margin: 10px 7px;
78         color: #000;
79     }
80     .desc div {
81         float: left;
82         font-family: Arial;
83         width: 70px;
84         margin-right: 65px;
85         font-size: 13px;
86         font-weight: bold;
87         color: #000;
88     }
89 </style>
90 @stop
91
92 @section('js')
93     @parent
94     {{ HTML::script('js/jquery.countdown.min.js') }}
95     <script>
96         (function($) {
97             $('#counter').countdown({
```

```

98         stepTime: 1,
99         startTime: "{{ $time }}",
100         timerEnd: function() {
101             window.location.replace('{{
route('test.info', $test->test_id) }}');
102         },
103         image: "{{ URL::to('img/digits.png') }}"
104     });
105
106     var itemsOnPage = 5;
107
108     $('#paginator').pagination({
109         prevText: 'Anterior',
110         nextText: 'Siguiente',
111         items: {{ $test->questions->count() }},
112         itemsOnPage: itemsOnPage,
113         cssStyle: 'light-theme',
114         onInit: function() {
115             var pages =
116             $('#paginator').pagination('getPagesCount');
117
118             if (1 == pages) {
119                 $('#paginator').pagination('disab
le');

```



```

119         $('#paginator').hide();
120     } else {
121         $('.item').hide();
122
123         var current =
124     $('#paginator').pagination('getCurrentPage');
125         var first = (current - 1) *
itemsOnPage;
126         var end = current * itemsOnPage;
127
128         for (var i = first; i < end; i++)
129     {
130         $('#item-' + i).show();
131     }
132
133     },
134
135     onPageClick: function(page, event) {
136
137         event.preventDefault();
138
139         $('.item').hide();
140
141         var current =
142     $('#paginator').pagination('getCurrentPage');
143         var first = (current - 1) *
itemsOnPage;
144         var end = current * itemsOnPage;

```

```

140
141         for (var i = first; i < end; i++) {
142             $('#item-' + i).show();
143         }
144     }
145     });
146     })(jQuery);
147 </script>
148 @stop

```

## Macros

### Depuración

```

1 @if (Config::get('app.debug', false))
2     <div class="list-group">
3         @foreach (DB::getQueryLog() as $log)
4             <a href="#" class="list-group-item">
5                 <h4 class="list-group-item-
heading"><b>Query:</b> {{ $log['query'] }}</h4>
6                 <p class="list-group-item-text">
7                     <b>Bindings:</b>
8                     @foreach ($log['bindings'] as
$binding)
9                         {{ $binding }}
10                     @endforeach
11                 </p>
12                 <b>Time: <small>{{ $log['time'] }}
s</small></b>
13             </a>
14         @endforeach
15     </div>
16 @endif

```

## Campo

```
1 @if ($errors->has($name))
2     <div class="form-group has-error">
3         {{ Form::label($name, $label, array('class' =>
4 'col-sm-2 control-label')) }}
5         <div class="col-sm-10">
6             @if ('textarea' === $type)
7                 {{ Form::textarea($name, null,
8 array('class' => 'form-control')) }}
9             @else
10                 {{ Form::input($type, $name, null,
11 array('class' => 'form-control')) }}
12             @endif
13             <span class="help-block alert alert-
14 danger">{{ $errors->first($name) }}</span>
15         </div>
16     </div>
17 @else
18     <div class="form-group">
19         {{ Form::label($name, $label, array('class' =>
20 'col-sm-2 control-label')) }}
21         <div class="col-sm-10">
22             @if ('textarea' === $type)
23                 {{ Form::textarea($name, null,
24 array('class' => 'form-control')) }}
25             @else
26                 {{ Form::input($type, $name, null,
27 array('class' => 'form-control')) }}
28             @endif
29         </div>
30     </div>
31 @endif
```

### Barra lateral

```
1 <ul class="nav nav-pills nav-stacked">
2     @foreach ($links as $link)
3         @if (!isset($link['route']))
4             <li><a href="#">{{ $link['name'] }}</a></li>
5         @else
6             @if (is_array($link['route']))
7                 @if (in_array(Route::currentRouteName(),
8 $link['route']))
9                     <li class="active" ><a href="{{
10 route($link['route'][0]) }}">{{ $link['name'] }}</a></li>
11                 @else
12                     <li><a href="{{
13 route($link['route'][0]) }}">{{ $link['name'] }}</a></li>
14                 @endif
15             @else
16                 @if
17 (strcasecmp(Route::currentRouteName(), $link['route']) ==
18 0)
19                     <li class="active"><a href="{{
20 route($link['route']) }}">{{ $link['name'] }}</a></li>
21                 @else
22                     <li><a href="{{ route($link['route'])
23 }}">{{ $link['name'] }}</a></li>
24                 @endif
25             @endif
26         @endif
27     @endforeach
28 </ul>
```

## Perfiles

### Editar

```
1 @section('content')
2     {{ Form::model($user, array('route' =>
3     'profile.update', 'class' => 'form-horizontal', 'role' =>
4     'form')) }}
5     <div class="panel panel-warning">
6         <div class="panel-heading">
7             <h1 class="panel-title">Editar
8             Perfil</h1>
9         </div>
10        <div class="panel-body">
11            @include('includes.field', array('name'
12            => 'first_name', 'label' => 'Nombre', 'type' => 'text'))
13            @include('includes.field', array('name'
14            => 'last_name', 'label' => 'Apellido', 'type' => 'text'))
15            @include('includes.field', array('name'
16            => 'email', 'label' => 'Correo', 'type' => 'email'))
17            @include('includes.field', array('name'
18            => 'new_password', 'label' => 'Nueva contraseña', 'type'
19            => 'password'))
20            @include('includes.field', array('name'
21            => 'new_password_confirmation', 'label' => 'Confirmar
22            contraseña', 'type' => 'password'))
23            @include('includes.field', array('name'
24            => 'password', 'label' => 'Contraseña actual', 'type' =>
25            'password'))
26        </div>
27        <div class="panel-footer">
28            <button type="submit" class="btn btn-
29            warning">Guardar</button>
30            <a type="button" class="btn btn-default"
31            href="{{ route('profile.view') }}">Cancelar</a>
32        </div>
33    </div>
34    {{ Form::close() }}
35 @stop
```

## Vista

```
1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h1 class="panel-title">Perfil</h1>
5         </div>
6         <div class="panel-body">
7             <div class="media">
8                 <div class="pull-left">
9                     
11                 </div>
12                 <div class="media-body">
13                     <dl class="dl-horizontal">
14                         <dt>Nombre:</dt>
15                         <dd>{{ $user->first_name }}</dd>
16                         <dt>Apellido:</dt>
17                         <dd>{{ $user->last_name }}</dd>
18                         <dt>Nombre de usuario:</dt>
19                         <dd>{{ $user->user_name }}</dd>
20                         <dt>Correo:</dt>
21                         <dd>{{ $user->email }}</dd>
22                     </dl>
23                 </div>
24             </div>
25             <div class="panel-footer">
26                 @if (Auth::user() === $user)
27                     <a type="button" class="btn btn-primary"
28 href="{{ route('profile.edit') }}">Editar</a>
29                 @endif
30             </div>
31 @stop
```

## Plantillas

### Por defecto

```
1 <!doctype html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <title>@yield('title', 'SIAD')</title>
6     <meta name="viewport" content="width=device-width,
7     initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
8     @section('css')
9         {{ HTML::style('css/bootstrap.min.css') }}
10        {{ HTML::style('css/simplePagination.css') }}
11    @show
12</head>
13<body>
14    <div class="container">
15        <div class="row">
16            <div class="col-xs-12">
17                <div class="row">
18                    <div class="col-xs-12">
19                        <nav class="navbar navbar-default"
20                        role="navigation">
21                            <div class="navbar-header">
22                                <button type="button">
```

```

class="navbar-toggle" data-toggle="collapse" data-
target="#user-navbar">
21         <span class="sr-
only">Toggle navigation</span>
22         <span class="icon-
bar"></span>
23         <span class="icon-
bar"></span>
24         <span class="icon-
bar"></span>
25     </button>
26     <a class="navbar-brand"
href="#">SIAD <small>Sistema Interactivo de Aprendizaje a
Distancia</small></a>
27 </div>
28 <div class="collapse navbar-
collapse" id="user-navbar">
29     <!-- Menu -->
30 </div>
31 </nav>
32 </div>
33 </div>
34 <div class="media page-header">
35     <div class="pull-left">
36         
37     </div>
38     <div class="media-body">
39         <h2 class="media-heading">{{
$user->first_name . ' ' . $user->last_name }}</h2>
40         <a type="button" class="btn btn-

```



```

warning btn-xs" href="{{ route('do.logout') }}">Salir</a>
41         </div>
42     </div>
43 </div>
44 </div>
45 <div class="row">
46     <div class="col-xs-3">
47         @section('sidebar')
48             @include('includes.sidebar')
49             @show
50     </div>
51     <div class="col-xs-9">
52         @foreach (Session::get('messages',
array()) as $message)
53             <div class="alert alert-{{
$message['type'] }}">
54                 <button type="button"
class="close" data-dismiss="alert" aria-
hidden="true">&times;</button>
55                 {{ $message['text'] }}
56             </div>
57             {{ Session::forget('messages') }}
58         @endforeach
59         @yield('content')
60     </div>

```

```

61     </div>
62     <div class="row">
63         <div class="col-xs-12">
64             @include('includes.debug')
65         </div>
66     </div>
67 </div>
68
69 @section('js')
70     {{ HTML::script('js/jquery.min.js') }}
71     {{ HTML::script('js/bootstrap.min.js') }}
72     {{ HTML::script('js/holder.js') }}
73     {{ HTML::script('js/jquery.simplePagination.js')
74 }}
75     {{ HTML::script('js/bootbox.min.js') }}
76     <script>
77         (function($) {
78             function(e) {
79                 $(document).on("click", ".confirm",
80                     e.preventDefault();
81
82                     var location = $(this).attr('href');

```

```

82         bootbox.confirm("¿Esta seguro?",
function(confirm) {
83             if (confirm) {
84                 window.location.replace(locati
on);
85             }
86         });
87     });
88     })(jQuery);
89     </script>
90     <script type="text/javascript">
91         $('.tip').tooltip();
92     </script>
93     @show
94 </body>
95 </html>

```

## Invitado

```
1 <!doctype html>
2 <html lang="es">
3 <head>
4     <meta charset="UTF-8">
5     <title>@yield('title', 'SIAD')</title>
6     <meta name="viewport" content="width=device-width,
7     initial-scale=1.0, maximum-scale=1.0, user-scalable=no">
8     @section('css')
9         {{ HTML::style('css/bootstrap.min.css') }}
10    @show
11 </head>
12 <body>
13     <div class="container">
14         <div class="row">
15             <div class="col-xs-12">
16                 <div class="page-header">
17                     <h1>SIAD</h1>
18                     <h2><small>Sistema Interactivo de
19 Aprendizaje a Distancia</small></h2>
20                 </div>
21             </div>
22         </div>
23     </div>
24     <div class="row">
```

```

22         <div class="col-xs-12">
23             @foreach (Session::get('messages',
array()) as $message)
24                 <div class="alert alert-{{
$message['type'] }}">
25                     <button type="button"
class="close" data-dismiss="alert" aria-
hidden="true">&times;</button>
26                     {{ $message['text'] }}
27                 </div>
28                 {{ Session::forget('messages') }}
29             @endforeach
30             @yield('content')
31         </div>
32     </div>
33     <div class="row">
34         <div class="col-xs-12">
35             @include('includes.debug')
36         </div>
37     </div>
38 </div>
39 @section('js')
40     {{ HTML::script('js/jquery.min.js') }}
41     {{ HTML::script('js/bootstrap.min.js') }}

```

```

42     {{ HTML::script('js/holder.js') }}
43     {{ HTML::script('js/bootbox.min.js') }}
44     <script>
45         (function($) {
46             $(document).on("click", ".confirm",
function(e) {
47                 e.preventDefault();
48
49                 var location = $(this).attr('href');
50
51                 bootbox.confirm("¿Esta seguro?",
function(confirm) {
52                     if (confirm) {
53                         window.location.replace(locati
on);
54                     }
55                 });
56             });
57         })(jQuery);
58     </script>
59     @show
60 </body>
61 </html>

```

## Preguntas

### Añadir

```
1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h3 class="panel-title">
5                 Agregar Pregunta
6                 @if ('open' === $type)
7                     abierta
8                 @endif
9                 @if ('choice' === $type)
10                    de seleccion
11                @endif
12                @if ('multiple' === $type)
13                    multiple
14                @endif
15            </h3>
16        </div>
17        <form role="form" class="form-horizontal"
18        action="{{ route('question.create', $test->test_id) }}"
19        method="POST">
20            <div class="panel-body">
21                <input type="hidden" name="type" value="{{
22                $type }}">
23                @include('includes.field', array('name' =>
24                'question_text', 'label' => 'Texto', 'type' => 'text'))
25                @include('includes.field', array('name' =>
26                'points', 'label' => 'Puntos', 'type' => 'number'))
27                @if ('multiple' === $type)
28                    <p><b>Cada respuesta valdra los mismos
29                    puntos</b></p>
30                @endif
31                <table class="table table-striped table-
32                bordered table-hover table-condensed">
33                    <thead>
34                        <tr>
35                            <th>Respuesta</th>
36                            <th>Es correcta</th>
37                            <th>
38                                <button type="button"
39                                class="btn btn-primary btn-xs" id="add-question">
```

```

32                                     <span class="glyphicon
glyphicon-plus"></span> Agregar
33                                     </button>
34                                 </th>
35                            </tr>
36                    </thead>
37                    <tbody id="answers">
38                        @if (($answerText =
Input::old('answer_text')) && ($isCorrect =
Input::old('is_correct')))
39                            @for ($i = 0; $i <
count($answerText); $i++)
40                                <tr>
41                                    <td>
42                                        <input type="text"
name="answer_text[]" value="{{ $answerText[$i] }}"
class="form-control"/>
43                                    </td>
44                                    <td>
45                                        @if (1 ==
$isCorrect[$i])
46                                            <input
type="checkbox" class="form-control checkable" checked/>
47                                            @else
48                                            <input
type="checkbox" class="form-control checkable"/>
49                                            @endif
50                                            <input type="hidden"
name="is_correct[]" value="{{ $isCorrect[$i] }}" />
51                                    </td>
52                                    <td>
53                                        <button type="button"
class="btn btn-default btn-xs removable"><span
class="glyphicon glyphicon-remove"></span>
Eliminar</button>
54                                    </td>
55                                </tr>
56                            @endfor
57                        @endif
58                    </tbody>
59                </table>
60            </div>

```



```

61         <div class="panel-footer">
62             <button type="submit" class="btn btn-primary
btn-lg btn-block">Terminar</button>
63             <a class="btn btn-default btn-lg btn-block"
href="{{ route('test.manager', $test->test_id)
}}">Cancelar</a>
64         </div>
65     </form>
66 </div>
67 @stop
68
69 @section('js')
70     @parent
71     <script>
72         (function($) {
73             $('#answers').on('click', '.removable',
function() {
74                 $(this).closest('tr').remove();
75             });
76             $('#answers').on('click', '.checkable',
function() {
77                 var val = $(this).next().val();
78
79                 val = 0 == val ? 1 : 0;
80
81                 $(this).next().val(val);
82             });
83             $('#add-question').bind('click',
function(event) {
84                 $('#answers').append("<tr><td><input
type=\"text\" name=\"answer_text[]\" class=\"form-
control\"/></td><td><input type=\"checkbox\"
class=\"form-control checkable\"/><input type=\"hidden\"
name=\"is_correct[]\" value=\"0\"/></td><td><button
type=\"button\" class=\"btn btn-default btn-xs
removable\"><span class=\"glyphicon glyphicon-
remove\"></span> Eliminar</button></td></tr>");
85             });
86         })(jQuery);
87     </script>
88 @stop

```

## Editar

```
1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h3 class="panel-title">
5                 Editar Pregunta
6                 @if ('open' === $question->type)
7                     abierta
8                 @endif
9                 @if ('choice' === $question->type)
10                    de seleccion
11                @endif
12                @if ('multiple' === $question->type)
13                    multiple
14                @endif
15            </h3>
16        </div>
17        {{ Form::model($question, array('route' =>
18array('question.update', $test->test_id, $question-
19>question_id))) }}
20        <div class="panel-body">
21            @include('includes.field', array('name' =>
22'question_text', 'label' => 'Texto', 'type' => 'text'))
23            @include('includes.field', array('name' =>
24'points', 'label' => 'Puntos', 'type' => 'number'))
25            @if ('multiple' === $question->type)
26                <p><b>Cada respuesta valdra los mismos
27puntos</b></p>
28            @endif
29            <table class="table table-striped table-
30bordered table-hover table-condensed">
31                <thead>
32                    <tr>
33                        <th>Respuesta</th>
34                        <th>Es correcta</th>
35                        <th>
36                            <button type="button"
37class="btn btn-primary btn-xs" id="add-question">
38                                <span class="glyphicon
39glyphicon-plus"></span> Agregar
40                            </button>
41                        </th>
```

```

34         </tr>
35     </thead>
36     <tbody id="answers">
37         @foreach ($question->answers as
$answer)
38             <tr>
39                 <td>
40                     <input type="text"
name="answer_text[]" value="{{ $answer->answer_text }}"
class="form-control"/>
41                 </td>
42                 <td>
43                     @if ($answer->is_correct)
44                         <input
type="checkbox" class="form-control checkable" checked/>
45                         <input type="hidden"
name="is_correct[]" value="1"/>
46                     @else
47                         <input
type="checkbox" class="form-control checkable"/>
48                         <input type="hidden"
name="is_correct[]" value="0"/>
49                     @endif
50                 </td>
51                 <td>
52                     <button type="button"
class="btn btn-default btn-xs removable"><span
class="glyphicon glyphicon-remove"></span>
Eliminar</button>
53                 </td>
54             </tr>
55         @endforeach
56     </tbody>
57 </table>
58 </div>
59 <div class="panel-footer">
60     <button type="submit" class="btn btn-primary
btn-lg btn-block">Guardar</button>
61     <a class="btn btn-default btn-lg btn-block"
href="{{ route('test.manager', $test->test_id)
}}">Cancelar</a>
62 </div>

```

```

63         {{ Form::close() }}
64     </div>
65 @stop
66
67 @section('js')
68     @parent
69     <script>
70         (function($) {
71             $('#answers').on('click', '.removable',
function() {
72                 $(this).closest('tr').remove();
73             });
74             $('#answers').on('click', '.checkable',
function() {
75                 var val = $(this).next().val();
76
77                 val = 0 == val ? 1 : 0;
78
79                 $(this).next().val(val);
80             });
81             $('#add-question').bind('click',
function(event) {
82                 $('#answers').append("<tr><td><input
type=\"text\" name=\"answer_text[]\" class=\"form-
control\"/></td><td><input type=\"checkbox\"
class=\"form-control checkable\"/><input type=\"hidden\"
name=\"is_correct[]\" value=\"0\"/></td><td><button
type=\"button\" class=\"btn btn-default btn-xs
removable\"><span class=\"glyphicon glyphicon-
remove\"></span> Eliminar</button></td></tr>");
83             });
84         })(jQuery);
85     </script>
86 @stop

```

## Recursos

### Añadir

```
1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h3 class="panel-title">
5                 Agregar
6                 @if ($type === 'video')
7                     Video <span class="glyphicon
8 glyphicon-play"></span>
9                 @endif
10                @if ($type === 'file')
11                    Archivo <span class="glyphicon
12 glyphicon-file"></span>
13                @endif
14                @if ($type === 'url')
15                    enlace <span class="glyphicon
16 glyphicon-link"></span>
17                @endif
18            </h3>
19        </div>
20        <form role="form" class="form-horizontal"
21 enctype="multipart/form-data" action="{
22 route('resource.create', $course->course_id) }"
23 method="POST">
24            <div class="panel-body">
25                <input type="hidden" name="type" value="{
26 $type }" >
27                @include('includes.field', array('name' =>
28 'title', 'label' => 'Titulo', 'type' => 'text'))
29                @include('includes.field', array('name' =>
30 'description', 'label' => 'Descripcion', 'type' =>
31 'textarea'))
32                @if ($type === 'video')
33                    @include('includes.field', array('name'
34 => 'url', 'label' => 'Enlace de youtube', 'type' =>
35 'text'))
36                @endif
37                @if ($type === 'file')
38                    @include('includes.field', array('name'
39 => 'upload', 'label' => 'Archivo', 'type' => 'file'))
```

```

27         <span>
28             <b>Limite: 10mb</b>
29         </span>
30     @endif
31     @if ($type === 'url')
32         @include('includes.field', array('name'
=> 'url', 'label' => 'Enlace', 'type' => 'text'))
33     @endif
34 </div>
35 <div class="panel-footer">
36     <button type="submit" class="btn btn-primary
btn-lg btn-block">Agregar</button>
37     <a class="btn btn-default btn-lg btn-block"
href="{{ route('course.manager', $course->course_id)
}}">Cancelar</a>
38 </div>
39 </form>
40 </div>
41 @stop

```

#### Editar

```

1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h3 class="panel-title">
5                 Editar
6                 @if ($resource->type === 'video')
7                     Video <span class="glyphicon
glyphicon-play"></span>
8                 @endif
9                 @if ($resource->type === 'file')
10                    Archivo <span class="glyphicon
glyphicon-file"></span>
11                @endif
12                @if ($resource->type === 'url')
13                    Enlace <span class="glyphicon
glyphicon-link"></span>
14                @endif
15            </h3>
16        </div>
17        {{ Form::model($resource, array('route' =>

```

```

array('resource.update', $resource->resource_id), 'files'
=> true)) }}
18     <div class="panel-body">
19         @include('includes.field', array('name' =>
'title', 'label' => 'Titulo', 'type' => 'text'))
20         @include('includes.field', array('name' =>
'description', 'label' => 'Descripcion', 'type' =>
'textarea'))
21         @if ($resource->type === 'video')
22             @include('includes.field', array('name'
=> 'attached', 'label' => 'Enlace de youtube', 'type' =>
'text'))
23         @endif
24         @if ($resource->type === 'file')
25             @include('includes.field', array('name'
=> 'upload', 'label' => 'Archivo', 'type' => 'file'))
26             <p>Limite: 10mb</p>
27             <p>{{ $resource->attached }}</p>
28         @endif
29         @if ($resource->type === 'url')
30             @include('includes.field', array('name'
=> 'attached', 'label' => 'Enlace', 'type' => 'text'))
31         @endif
32     </div>
33     <div class="panel-footer">
34         <button type="submit" class="btn btn-primary
btn-lg btn-block">Guardar</button>
35         <a class="btn btn-default btn-lg btn-block"
href="{{ route('course.manager', $course->course_id)
}}">Cancelar</a>
36     </div>
37     {{ Form::close() }}
38 </div>
39 @stop

```

## Vista

```
1 @section('content')
2     <div class="panel panel-default">
3         <div class="panel-heading">
4             <h3 class="panel-title">{{ $resource->title
5 }}</h3>
6         </div>
7         <div class="panel-body">
8             <p class="lead text-muted">{{ $resource-
9 >description }}</p>
10             @if ($resource->is('file'))
11                 @if ($resource->toFile->is_pdf)
12                     <embed src="{{
13 URL::to('uploads/resources/' . $resource->resource_id) }}"
14 class="col-xs-12" height="500" alt="pdf"
15 pluginspage="http://www.adobe.com/products/acrobat/readste
16 p2.html">
17                 @endif
18             </div class="text-center">
19                 <a class="btn btn-primary btn-lg"
20 href="{{ route('resource.download', $resource-
21 >resource_id) }}">Descargar</a>
22             </div>
23         @endif
24         @if ($resource->is('video'))
25             <iframe type="text/html" class="col-xs-12"
26 height="385" src="http://www.youtube.com/embed/{{
27 $resource->toVideo->video_id }}" frameborder="0">
28             </iframe>
```



```

19         @endif
20         @if ($resource->is('url'))
21             <div class="text-center">
22                 <a class="btn btn-primary btn-lg"
href="{{ $resource->toUrl->url }}">Ir</a>
23             </div>
24         @endif
25     </div>
26 </div>
27 @stop

```

## Principal

```

1 @section('content')
2     <div class="jumbotron">
3         <h1 class="text-center">¡Bienvenido a SIAD!</h1>
4         <p>
5             <a class="tip" href="#" data-toggle="tooltip"
title="" data-original-title="Sistema Interactivo de
Aprendizaje a Distancia">SIAD</a>
6             es el sistema de Aprendizaje a distancia
creado por y para la comunidad de estudiantes de la <a
class="tip" href="http://uach.mx" target="_blank" data-
toggle="tooltip" title="" data-original-
title="Universidad Autonoma de Chihuahua">UACH</a>.
7         </p>
8     </div>
9     <div class="panel panel-primary">
10         <div class="panel-heading">
11             <h3 class="panel-title">Ultimos 10
cursos</h3>
12         </div>
13         <table class="table table-striped table-bordered
table-hover table-condensed">

```

```

14         <thead>
15             <tr>
16                 <th>Curso</th>
17                 <th>Profesor</th>
18                 <th></th>
19             </tr>
20         </thead>
21         <tbody>
22             @foreach ($courses as $course)
23                 <tr>
24                     <td>
25                         {{ $course->course_name }}
26                     </td>
27                     <td><b>{{ $course->teacher-
28 >full_name }}</b>
29                         ({{ $course->teacher->email
30 }})</td>
31                     <td>
32                         <a class="btn btn-primary
33 btn-xs" href="{{ route('course.register', $course-
34 >course_id) }}">
35                             @if ($course->password)
36                                 <span
37 class="glyphicon glyphicon-lock"></span>
38                             @endif
39                             Matricularme
40                         </a>
41                     </td>
42                 </tr>
43             @endforeach
44         </tbody>
45     </table>
46 </div>
47 @stop

```

## Kardex

```
1 @section('content')
2     <div class="panel panel-primary">
3         <div class="panel-heading">
4             <h1 class="panel-title">Kardex</h1>
5         </div>
6         <div class="panel-body">
7             <dl class="dl-horizontal">
8                 <dt>Nombre:</dt>
9                 <dd>{{ $user->first_name }}</dd>
10                <dt>Apellido:</dt>
11                <dd>{{ $user->last_name }}</dd>
12                <dt>Nombre de usuario:</dt>
13                <dd>{{ $user->user_name }}</dd>
14                <dt>Correo:</dt>
15                <dd>{{ $user->email }}</dd>
16            </dl>
17            <table class="table table-striped table-hover
table-condensed table-bordered">
18                <thead>
19                    <tr>
20                        <th>Nombre del curso</th>
21                        <th>Estado</th>
22                        <th>Calificación</th>
23                    </tr>
24                </thead>
25                <tbody>
26                    @foreach ($user->kardex as $kardex)
27                        <tr>
28                            <td>{{ $kardex->course_name
29                                }}</td>
30                            <td>{{ $kardex->status
31                                }}</td>
32                            <td>{{ $kardex->qualification
33                                }}</td>
34                        </tr>
35                    @endforeach
36                </tbody>
37            </table>
38        </div>
39        <div class="panel-footer">
40        </div>
```

```
38 </div>
39 @stop
```

## Conclusión

Nuestro sistema o “plataforma” es un excelente método de estudio online para aquellas personas que deseen compartir de sus conocimientos además de que gracias a este tipo de sistemas será más fácil dar a conocer todos los conocimientos que tenemos y nos gustaría compartirlos es muy importante no confundirlo con una escuela es simplemente una serie de cursos en los cuales los alumnos pueden ser tanto maestros como alumnos.

## Bibliografía

Modelo vista controlador

- [http://es.wikipedia.org/wiki/Modelo\\_Vista\\_Controlador](http://es.wikipedia.org/wiki/Modelo_Vista_Controlador)

Laravel

- <http://es.wikipedia.org/wiki/Laravel>

HTTP

- [http://es.wikipedia.org/wiki/Hypertext\\_Transfer\\_Protocol](http://es.wikipedia.org/wiki/Hypertext_Transfer_Protocol)