



UNIVERSIDAD AUTÓNOMA DE GUADALAJARA

MAESTRIA EN CIENCIAS COMPUTACIONALES

PROYECTO BASE DE DATOS AVANZADAS

Proyecto visible en: <http://noehdez.info>

José Noé Hernández Vivanco | 3024021

Alumno de la Maestría en

Ciencias Computacionales

Dr. Rogelio Dávila Pérez

Profesor de la Materia

Bases de datos Avanzadas

Objetivo

Desarrollar un sistema en el cual se apliquen técnicas de uso e implementación de Bases de datos avanzadas, utilizando diversas tecnologías y en su caso enfocarlo para la Web. Uno de los objetivos es tener una base diseñada para administrar una biblioteca de registros de películas, en donde se incluyen información tal como: Actores, Fechas, Premios, Directores etc. El sistema deberá funcionar como un Servicio de **API (RESTful / WebService)** que actuara como el centro de datos y de peticiones para obtener información de la misma, retornándole en formato JSON.

Marco Teórico

Para poder desarrollar esta aplicación necesitamos conocer y utilizar las siguientes herramientas basadas para la Web 2.0. Las tecnologías, herramientas y técnicas que se utilizaron durante el desarrollo de esta aplicación son las siguientes:

1. HTML5
2. CSS, CSS3
3. Bootstrap “CSS Framework”
4. JavaScript
5. Angular JS “JavaScript Framework”
6. jQuery (JavaScript Framework)
7. AJAX “Asynchronous JavaScript And XML (JavaScript asíncrono y XML)”
8. Patrón de diseño MVC (Model View Controller)
9. PHP
10. Laravel 5.2 (PHP Framework)
11. MySQL Server
12. JSON
13. Digital Ocean “Cloud Computing, Infraestructure as a Service IaaS”

A continuación se describen cada una de las tecnologías, herramientas y técnicas aplicadas:

- **HTML5**

HTML5 (HyperText Markup Language, versión 5) es la quinta revisión importante del lenguaje básico de la World Wide Web, HTML. HTML5 especifica dos variantes de sintaxis para HTML: una «clásica», HTML (text/html), conocida como HTML5, y una variante XHTML conocida como sintaxis XHTML5 que deberá servirse con sintaxis XML (application/xhtml+xml).^{1 2}

- **CSS**

Las hojas de estilo en cascada (***Cascading Style Sheets, CSS***) son un lenguaje formal usado para definir la presentación de un documento estructurado escrito en HTML o XML (y por extensión en XHTML). El W3C es el encargado de formular la especificación de las hojas de estilo que servirá de estándar para los agentes de usuario o navegadores. La idea que se encuentra detrás del desarrollo de CSS es separar la estructura de un documento de su presentación. La información de estilo puede ser adjuntada tanto como un documento separado o en el mismo documento HTML. En este último podrían definirse estilos generales en la cabecera del documento o en cada etiqueta particular mediante el atributo "style".

Las ventajas de utilizar CSS (u otro lenguaje de estilo) son:

1. Control centralizado de la presentación de un sitio web completo, con lo que se agiliza de forma
2. considerable la actualización del mismo.
3. Los navegadores permiten a los usuarios especificar su propia hoja de estilo local que será aplicada a un sitio web remoto, con lo que aumenta considerablemente la accesibilidad.
4. Una página puede disponer de diferentes hojas de estilo según el dispositivo que la muestre.
5. El documento HTML en sí mismo es más claro de entender y se consigue reducir
6. considerablemente su tamaño

- **JavaScript, jQuery, AJAX y Angular JS.**

JavaScript es un lenguaje interpretado utilizado principalmente en páginas web.

Todos los navegadores interpretan el código JavaScript integrado dentro de las páginas web. Para interactuar con una página web se provee al lenguaje JavaScript de una implementación del DOM (Modelo de Objetos del Documento). JavaScript se ejecuta en el agente de usuario al mismo tiempo que las sentencias van descargándose junto con el código HTML. **jQuery** es una biblioteca o framework de JavaScript que permite simplificar la manera de interactuar con los documentos HTML, manipular el árbol **DOM**, manejar eventos, desarrollar animaciones y agregar interacción con la tecnología AJAX a páginas web.

jQuery, al igual que otras bibliotecas, ofrece una serie de funcionalidades basadas en JavaScript que de otra manera requerirían de mucho más código, es decir, con las funciones propias de esta biblioteca se logran grandes resultados en menos tiempo y espacio.

AJAX es un acrónimo de ***Asynchronous JavaScript And XML (JavaScript asíncrono y XML)***, es una técnica de desarrollo web para crear aplicaciones interactivas o RIA (***Rich Internet Applications***).

Estas aplicaciones se ejecutan en el cliente, es decir, en el navegador de los usuarios mientras se mantiene la comunicación asíncrona con el servidor en segundo plano. De esta forma es posible realizar cambios sobre las páginas sin necesidad de recargarlas, lo que significa aumentar la interactividad, velocidad y usabilidad en las aplicaciones.

AngularJS, o simplemente Angular, es un framework de JavaScript de código abierto, mantenido por Google, que se utiliza para crear y mantener aplicaciones web de una sola página. Su objetivo es aumentar las aplicaciones basadas en navegador con capacidad de Modelo Vista Controlador (**MVC**), en un esfuerzo para hacer que el desarrollo y las pruebas sean más fáciles.

La biblioteca lee el HTML que contiene atributos de las etiquetas personalizadas adicionales, entonces obedece a las directivas de los atributos personalizados, y une las piezas de entrada o salida de la página a un modelo representado por las variables estándar de JavaScript. Los valores de las variables de JavaScript se pueden configurar manualmente, o recuperados de los recursos **JSON** estáticos o dinámicos.



AngularJS se combina con el entorno en tiempo de ejecución Node.js, el framework para servidor Express.js y la base de datos MongoDB para formar el conjunto MEAN.

En nuestro caso utilizaremos PHP con Laravel para crear este Stack.

- **PHP**

PHP es un lenguaje interpretado de propósito general ampliamente usado, diseñado especialmente para desarrollo web y que puede ser incrustado dentro de código HTML. Generalmente se ejecuta en un servidor web, tomando el código en PHP como su entrada y creando páginas web como salida. Puede ser desplegado en la mayoría de los servidores web y en casi todos los sistemas operativos y plataformas sin costo alguno. PHP se encuentra instalado en más de 20 millones de sitios web y en un millón de servidores. Es también el módulo Apache más popular entre las computadoras que utilizan Apache como servidor web.

El gran parecido que posee PHP con los lenguajes más comunes de programación estructurada, como C y Perl, permiten a la mayoría de los programadores crear aplicaciones complejas con una curva de aprendizaje muy corta. También les permite involucrarse con aplicaciones de contenido dinámico sin tener que aprender todo un nuevo grupo de funciones.

- **MySQL Server**

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario con más de seis millones de instalaciones. MySQL AB desarrolla MySQL como software libre en un esquema de licenciamiento dual. Por un lado se ofrece bajo la GNU GPL para cualquier uso compatible con esta licencia, pero para aquellas empresas que quieran incorporarlo en productos privativos deben comprar a la empresa una licencia específica que les permita este uso. Está desarrollado en su mayor parte en ANSI C. MySQL es muy utilizado en aplicaciones web como MediaWiki, Amazon, Yahoo, Flickr o Drupal; en plataformas (Linux/Windows-Apache-MySQL-PHP/Perl/Python), y por herramientas de seguimiento de errores como Bugzilla.

- **DigitalOcean**

DigitalOcean es un proveedor Estadounidense de servidores virtuales privados, basado en la ciudad de Nueva York. La compañía alquila facilidades de centros de computo existentes, incluyendo sitios como Nueva York, Amsterdam, San Francisco, Londres y Singapur.

En este tipo de Servicio actúa como computación en la Nube (Cloud Computing) y es todo lo que puede ofrecer un sistema informático como servicio, de modo que los usuarios puedan acceder a los servicios disponibles "en la nube de Internet" sin conocimientos (o, al menos sin ser expertos) en la gestión de los recursos que usan.

- **JSON**

JSON, acrónimo de **JavaScript Object Notation**, es un formato de texto ligero para el intercambio de datos. JSON es un subconjunto de la notación literal de objetos de JavaScript aunque hoy, debido a su amplia adopción como alternativa a XML, se considera un formato de lenguaje independiente.

Una de las supuestas ventajas de JSON sobre XML como formato de intercambio de datos es que es mucho más sencillo escribir un analizador sintáctico (parser) de JSON. En JavaScript, un texto JSON se puede analizar fácilmente usando la función `eval()`, lo cual ha sido fundamental para que JSON haya sido aceptado por parte de la comunidad de desarrolladores AJAX, debido a la ubicuidad de JavaScript en casi cualquier navegador web.

Diseño de la Solución

El Proyecto se desarrollara con en Lenguaje de programación PHP y el Framework **Laravel 5.2** el cual actuara como Back-End (WebService) y utilizando la tecnologías antes mencionadas.

Este Framework se configurara en un Servidor en la Nube "Cloud Computing" con el proveedor **DigitalOcean** como se mencionó antes, montado sobre el sistema Operativo Ubuntu versión 14.02 y su configuración correspondiente a LAMP (Linux, Apache, MySQL/MariaDB and PHP, Python or Perl).

Para el Front-End se utilizara el Framework **Angular JS**, originalmente desarrollo por Google Inc, y el cual es el Framework de JavaScript mas utilizados en proyectos de Desarrollo Web modernos.

Nuestro Framework Angular se encargara de realizar peticiones al Servidor (API) donde está montado el Backend Laravel, esta tecnologías serán estarán montadas dentro del Servidor previamente configurado como LAMP.

Más adelante se detallaran la configuración de los mismos.

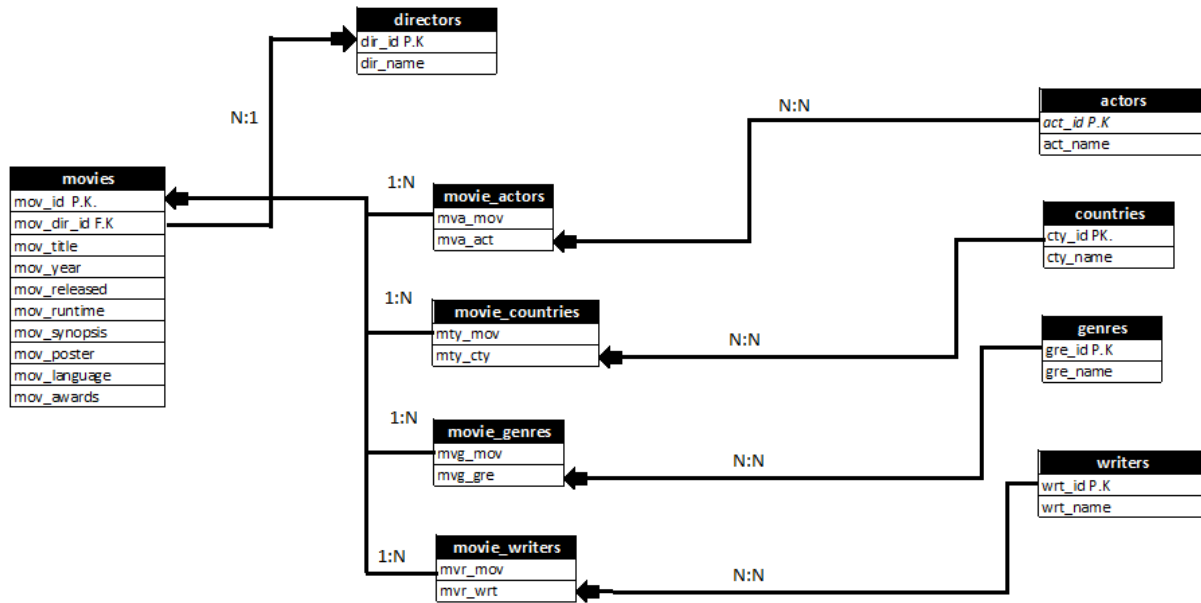
Implementación

La forma en que implementara nuestro proyecto se describe en el siguiente gráfico:



Como se ve en el grafico, tenemos nuestro motor de Base de Datos MySQL, aquí es donde se almacenaran los registros de las películas. Laravel es nuestro intermediaron y es el encargado de manipular todas las peticiones que se hagan hacia el Servidor, mediante Servicios REST, y este a su vez responder las peticiones en formato JSON, que seran las recibidas del lado del cliente.

Diseño de la Base de Datos



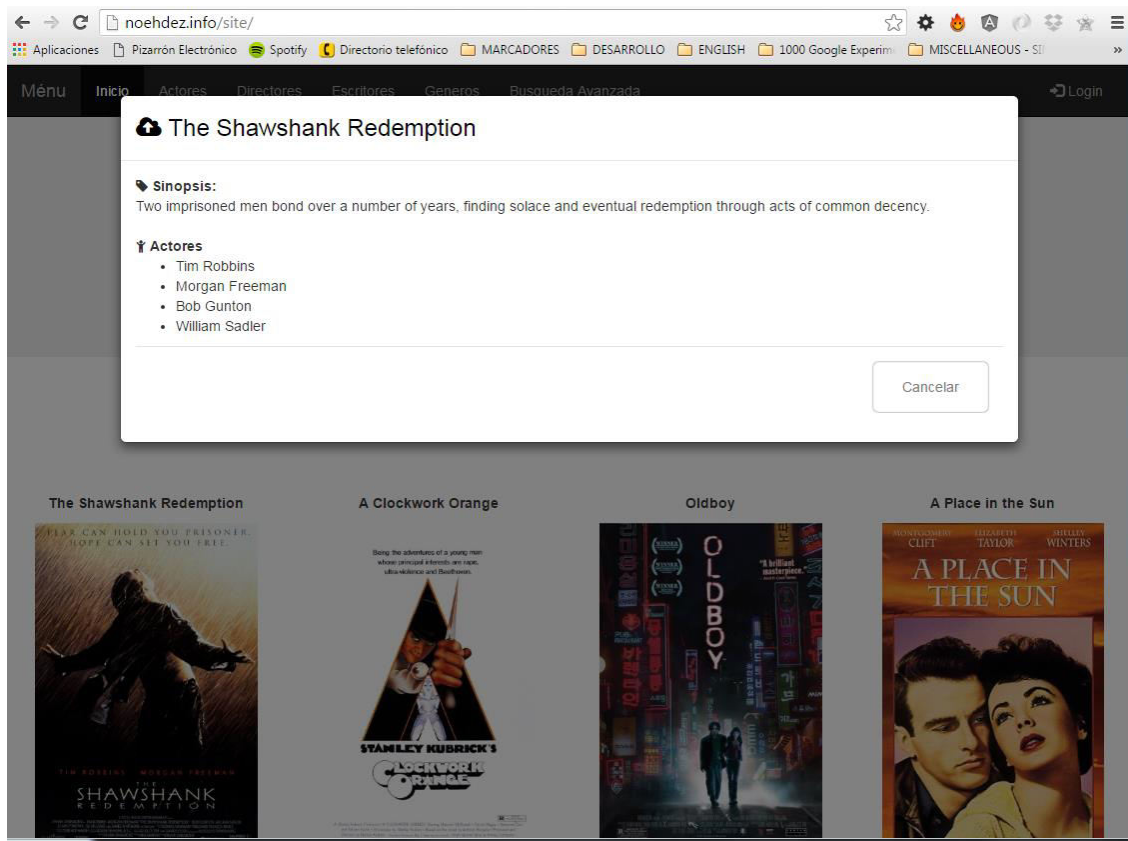
Resultados

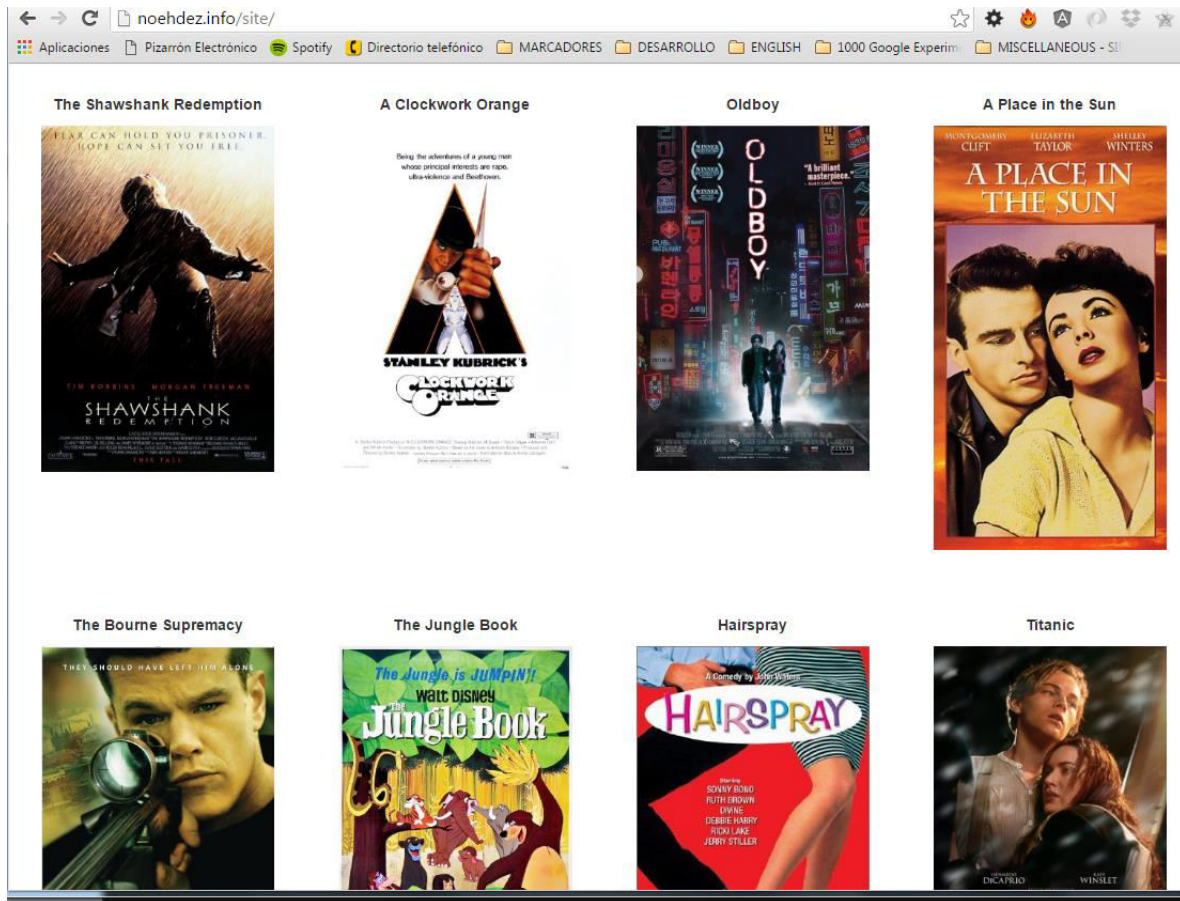
Nuestro Servidor está preparado para recibir peticiones REST y estas a su vez devueltas en formato JSON, un ejemplo de respuesta se muestra a continuación:

```

1  {
2  "movies": [
3  {
4      "mov_id": "1",
5      "mov_dir_id": "1",
6      "mov_title": "The Shawshank Redemption",
7      "mov_year": "1994",
8      "mov_released": "14 Oct 1994",
9      "mov_runtime": "142",
10     "mov_synopsis": "Two imprisoned men bond over a number of years, finding solace and eventual redemption through acts of common decency.",
11     "mov_poster": "http://ia.media-imdb.com/images/H/MV5BODU4MjU4NjIwN15BM15BanBnXkFtZTgwMDU2MjEyMDE@._V1_SX300.jpg",
12     "mov_language": "English",
13     "mov_awards": "Nominated for 7 Oscars. Another 14 wins & 21 nominations."
14 },
15 {
16     "mov_id": "2",
17     "mov_dir_id": "2",
18     "mov_title": "A Clockwork Orange",
19     "mov_year": "1971",
20     "mov_released": "02 Feb 1972",
21     "mov_runtime": "136",
22     "mov_synopsis": "In future Britain, charismatic delinquent Alex DeLarge is jailed and volunteers for an experimental aversion therapy developed all goes according to plan.",
23     "mov_poster": "http://ia.media-imdb.com/images/H/MV5BMTY3MTQ1MjU4NjIwN15BM15BanBnXkFtZTgwMDU2MjEyMDE@._V1_SX300.jpg",
24     "mov_language": "English",
25     "mov_awards": "Nominated for 4 Oscars. Another 8 wins & 17 nominations."
26 },
27 {
28     "mov_id": "3",
29     "mov_dir_id": "3",
30     "mov_title": "Oldboy",
31     "mov_year": "2003",
32     "mov_released": "21 Nov 2003",
33     "mov_runtime": "120",
34     "mov_synopsis": "After being kidnapped and imprisoned for 15 years, Oh Dae-Su is released, only to find that he must find his captor in 5 days.",
35     "mov_poster": "http://ia.media-imdb.com/images/H/MV5BMTI3NTQyMjU4NjIwN15BM15BanBnXkFtZTcwOTM2MjEyMDE@._V1_SX300.jpg",
36     "mov_language": "Korean",
37     "mov_awards": "26 wins & 16 nominations."
38 },
39 ]
40 }
    
```


Ejemplo del Proyecto funcional:





Descripción Técnica.

Configuración de Rutas en Laravel

```
Route::group(['prefix' => 'movie/', 'middleware' => ['guest']], function() {

    Route::get('obtener/mensaje/bienvenida', 'ControladorPrueba@getMessage');
    Route::get('exportacion/json/movies', 'ImportMoviesJsonController@import');
    Route::get('get/countries', 'CountriesController@getCountries');
    Route::get('get/movies', 'MoviesController@getMovies');

});
```

Ejemplo de obtención de Datos mediante el Framework de Laravel

```

1  <?php
2
3  namespace App\Http\Controllers;
4
5  use Illuminate\Http\Request;
6
7  use App\Http\Requests;
8  use DB;
9
10 class MoviesController extends Controller
11 {
12     public function getMovies(){
13
14         try{
15             $movies = DB::table('movies')
16                 ->select(
17                     'mov_id AS idMovie',
18                     'mov_title AS title',
19                     'mov_dir_id AS idDirector',
20                     'mov_year AS year',
21                     'mov_released AS released',
22                     'mov_runtime AS runtime',
23                     'mov_synopsis AS synopsis',
24                     'mov_poster AS image',
25                     'mov_language AS language',
26                     'mov_awards AS awards'
27                 )
28                 ->where('mov_poster', '<>', 'N/A')
29                 ->where('mov_poster', '<>', '')
30                 ->skip(0)->take(100)
31                 ->get();
32
33             return Response()->json(array('status' => 'success',
34                 'movies' => $movies,
35                 'msg' => 'Películas Obtenidas exitosamente.'));
36
37         }catch(\Illuminate\Database\QueryException $e){
38             return Response()->json(array('status' => 'error',
39                 'msg'=>'Error de query al consultar los datos.',
40                 'error'=>$e));
41         }
42     }
43 }

```

Configuración de Angular para el RestFull

```

1 angular.module('movieApp', ['ui.bootstrap'])
2 .constant('globalServerRoute', {
3   apiRoute : 'http://noehdez.info/movie.mx/public/movie',
4   siteRoute : 'http://noehdez.info/'
5 })
6 .run(function($rootScope, globalServerRoute){
7   $rootScope.api = globalServerRoute.apiRoute;
8   $rootScope.site = globalServerRoute.siteRoute;
9 });

```

Controlador de Angular JS para obtener información de la Películas

```

homePageCtrl.js
1 angular.module("movieApp")
2 .controller("homePageCtrl",
3   ["$rootScope", "$scope", "$http", "$compile", "$q", "$uibModal", "$log", "apiFactoryRest", "growlService",
4   function ( $rootScope, $scope, $http, $compile, $q, $uibModal, $log, apiFactoryRest, growlService ) {
5
6     $scope.countries = {};
7     $scope.movies = {};
8
9     $scope.fn = {
10      init : function(){
11        this.loadMovies();
12      },
13      loadCountries : function(){
14        apiFactoryRest.getCountries()
15          .success(function(response){
16            $scope.countries = response.countries;
17            growlService.notice('Mensaje Sistema', 'Yeahhhhh :')
18          })
19          .error(function(error){
20            //
21          });
22      },
23      loadMovies : function(){
24        apiFactoryRest.getMovies()
25          .success(function(response){
26            if(response.status === 'success'){
27              $scope.movies = response.movies;
28            } else if(response.status === 'error'){
29              growlService.notice('Mensaje Sistema', response.msg);
30            }
31          })
32          .error(function(error){
33            growlService.error('Mensaje Sistema', response.msg);
34          });
35      },
36      openMovieModal : function( datosPelicula ){
37        var modalInstance = $uibModal.open({
38          animation: true,
39          templateUrl : 'displayMovie.html',
40          controller : 'displayMovieModalCtrl',
41          size : 'lg',
42          backdrop : 'static',
43          keyboard : false,
44          resolve : {
45            Data: function() {
46              return { datosPelicula : datosPelicula };
47            }
48          }
49        });
50      },
51    };
52
53    $scope.fn.init();
54  });
55

```



Conclusión

Como se vio durante el desarrollo de proyecto, se utilizaron diversas tecnologías para implementar un sistema básico para administrar y consultar películas, con el Uso de las tecnologías (Laravel, Angular JS, Javascript, Ajax, CSS y HTML5) se creó un pequeño proyecto funcional, además del uso del Cloud Computing como **laaS (Infrastructure As a Service)** con el Proveedor de **DigitalOcean**. En la actualidad muchos sistemas en la Web utilizan estas tecnologías y en el uso del **Internet de la Cosas**, utilizan Cloud Computing (Twitter, NetFlix, Wikipedia Etc)

Hoy en la actualidad el uso del Internet ha ido en aumento, porque cualquier solución basada para la Web es indispensable en estos momentos.