

COLEGIO UNIVERSITARIO DE CARTAGO

Estudiantes:

Jeferson Chinchilla Segura

Olman Sánchez Zuniga

Miguel Alvarado Barahona

Curso:

Programación V

Evaluación de:

Entregable proyecto

Profesor:

Gabriel Gonzales Solano

Cartago, junio 2021

Contenido

Introducción	1
Diseño bade de datos	2
Diagrama de clases	3
Explicación diagrama de clases	4
Pruebas técnicas	5
Conclusiones y recomendaciones	12
Conclusiones:	12
Recomendaciones:	12

Introducción

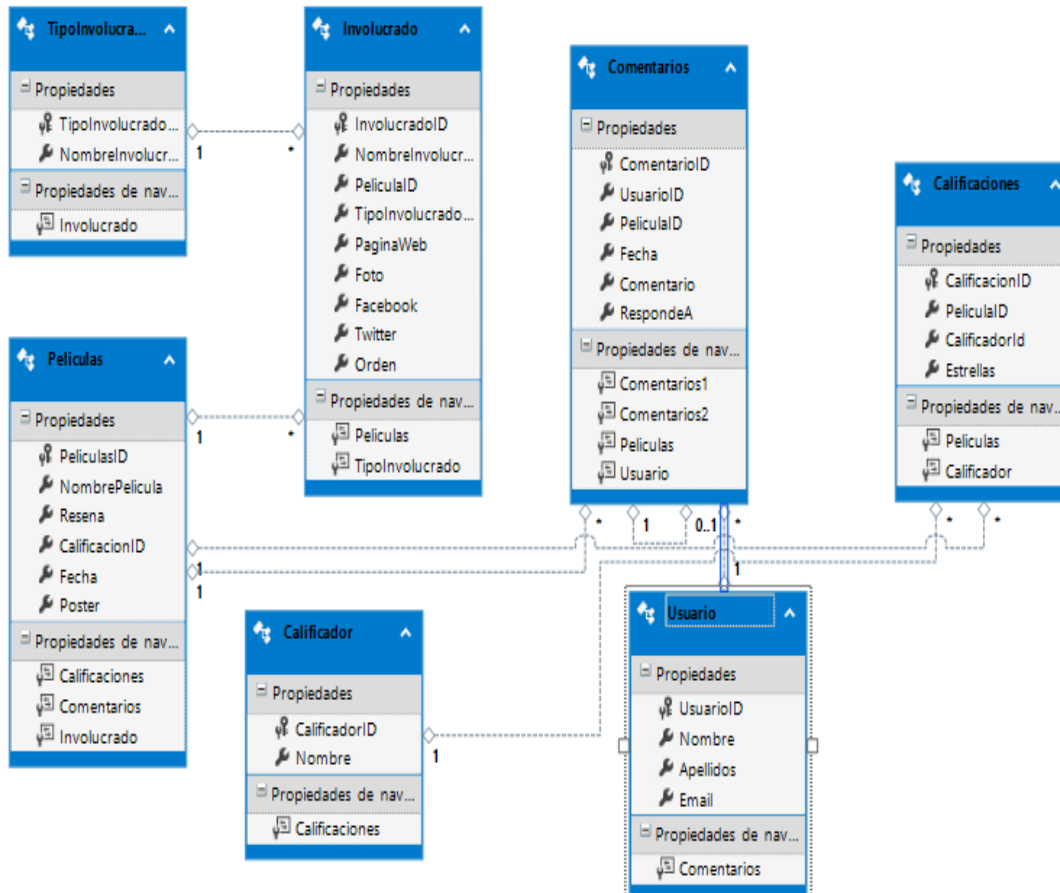
Se realizará un API Rest que permitirá la administración de un sistema omnicanal, se desea realizar un sistema que permita llevar registro respecto de las películas que se producen alrededor del mundo para permitir a los usuarios que opinen sobre ellas y poder posteriormente tener información respecto a los gustos de estos.

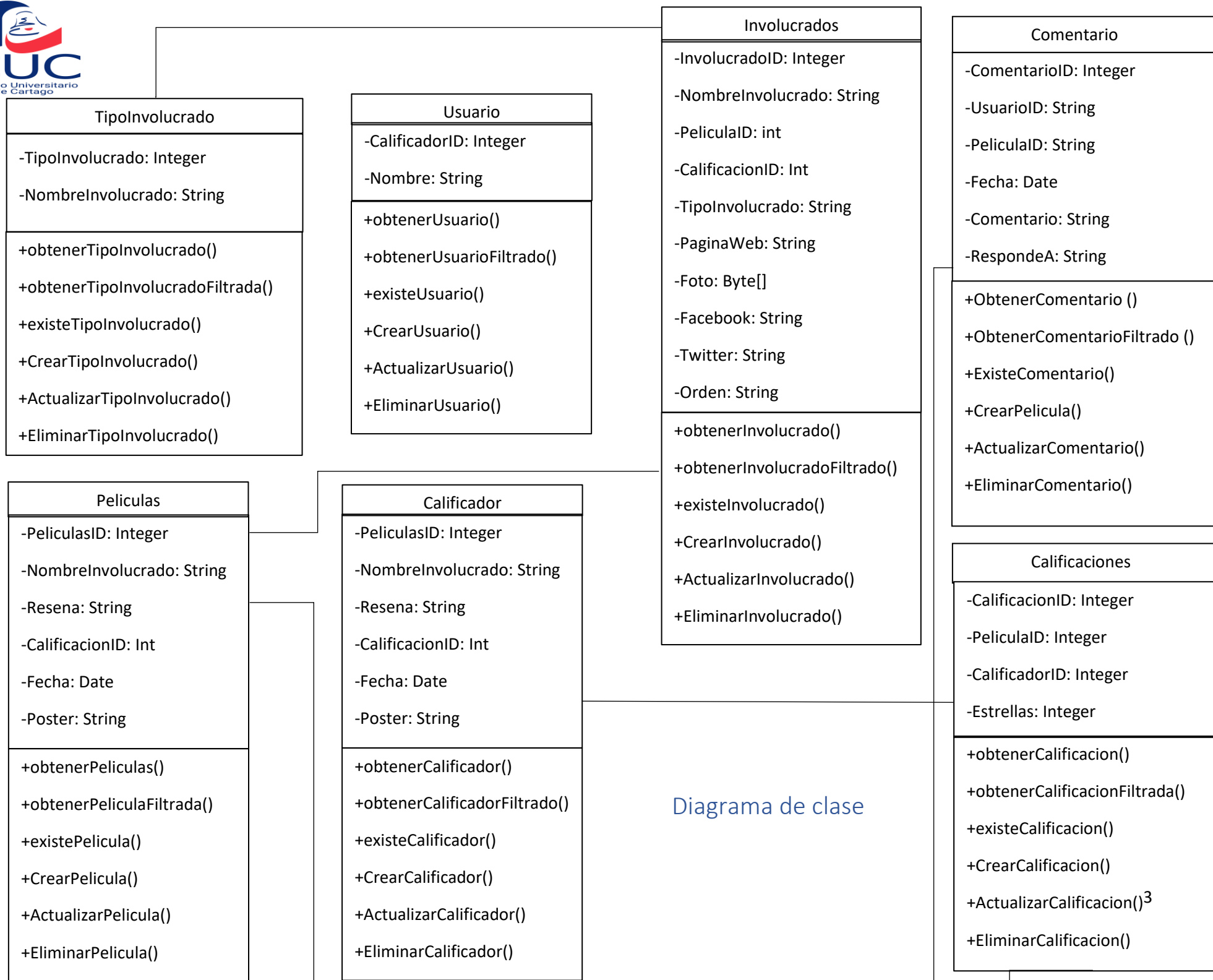
Para un usuario poder acceder a visualizar esta información y realizar comentarios sobre sus películas favoritas, debe primero registrarse en el sistema, para ello debe indicar: nombre de usuario deseado (no se puede repetir), Nombre, Apellidos, Email y Contraseña.

Una vez que el usuario se registra, debe mostrársele una página de bienvenida donde se desplegarán las películas más recientes y una opción para buscar la película que desea y poder realizar comentarios sobre estas.

Ya sea de la página de películas recientes o de la página de resultados de búsqueda, se debe dar la opción al usuario de ver el detalle de la película, mostrando su póster, título, reseña y calificación. En este detalle deben aparecer además los comentarios de los usuarios, donde el usuario podrá crear un hilo de comentario nuevo o bien responder a un hilo de comentarios existentes.

Diseño bade de datos





Explicación diagrama de clases

Tipo Involucrado: Tipo involucrado es un atributo que es una llave principal donde a través de esta llave se ingresa el nombre del involucrado donde propiamente es el actor, director o personaje principal donde con esta llave trae todos los métodos que se desea con esta tabla: obtener el involucrado, registrar, eliminar entre otros.

Involucrados: La llave principal de esta tabla es InvolucradoID que dará acceso a los datos a su vez, cuenta con dos llaves foráneas que son Película ID y Calificación ID donde son el llamado de datos que necesitamos de las otras tablas, concluyendo, esta tabla contiene los métodos obtener, crear, actualizar, eliminar y revisión si existe propiamente de los datos.

Calificador: Contiene un CalificadorID para identificarlo, va a contener el nombre del involucrado, una reseña, una fecha y su un poster, contiene los métodos obtener, crear, actualizar, eliminar y revisión si existe propiamente de los datos.

Calificaciones: Contendrá una CalificacionID para identificar la calificación a su vez cuenta con PeliculaID para obtener el identificador de la película a calificar y también el CalificadorID y sus respectivas estrellas, también cuenta con los métodos de obtener, crear, actualizar, eliminar y revisión si existe propiamente de los datos.

Comentario: Su clave principal es ComentarioID además que cuenta con dos llaves foráneas provenientes de la Tabla Usuario y Tabla Peliculas lo que conlleva a los métodos para verificar que están correcto los datos y seguido hacer la ejecución que se disponga a realizar.

Usuario: En la tabla más principal de todas las clases es la que será los primeros ingresos del programa y será de gran beneficio para la verificación de datos.

Peliculas: Contiene el nombre de la película, su calificación, nombre del involucrado, su respectiva fecha y su poster con los mismos métodos de obtener, crear, actualizar, eliminar y revisión si existe propiamente de los datos.

Pruebas técnicas

# HU	Criterio de aceptación	Evidencia
------	------------------------------	-----------

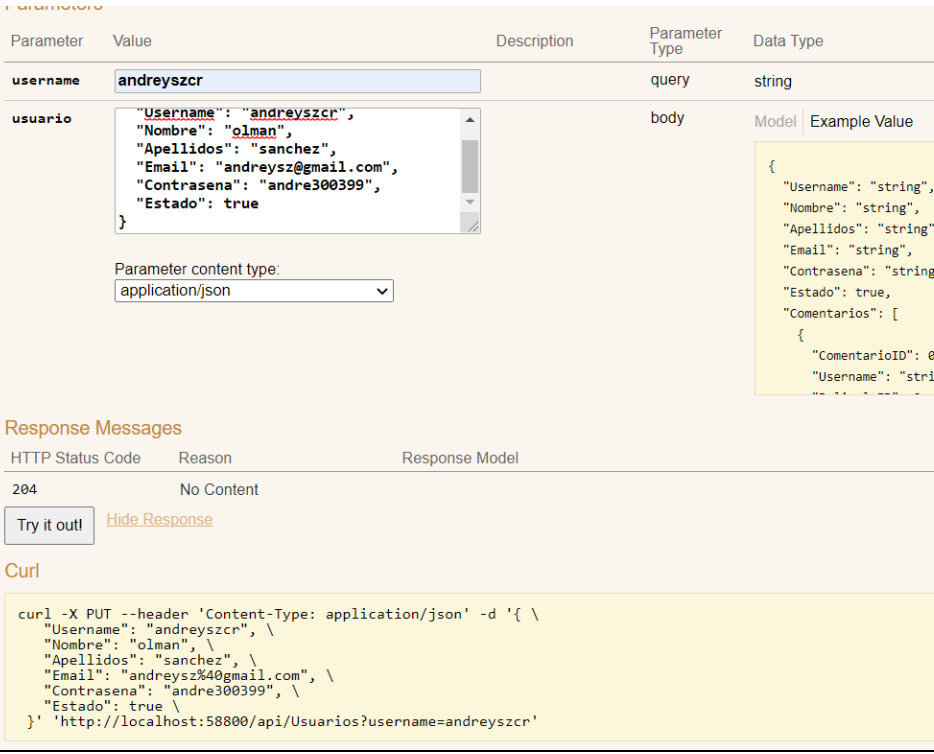
Calificadores POST	Se agrega los calificadores	<h3>Parameters</h3> <table> <thead> <tr> <th>Parameter</th><th>Value</th><th>Description</th></tr> </thead> <tbody> <tr> <td>calificador</td><td> <pre>{ "CalificadorID": 2, "Nombre": "novelas" }</pre> </td><td></td></tr> </tbody> </table> <p>Parameter content type: <input type="text" value="application/json"/></p> <p>Try it out! Hide Response</p> <h3>Curl</h3> <pre>curl -X POST --header 'Content-Type: application/json' --header 'Acc "CalificadorID": 2, \ "Nombre": "novelas" \ }' 'http://localhost:58800/api/Calificadors'</pre> <h3>Request URL</h3> <pre>http://localhost:58800/api/Calificadors</pre> <h3>Response Body</h3> <pre>"Calificador creado"</pre>	Parameter	Value	Description	calificador	<pre>{ "CalificadorID": 2, "Nombre": "novelas" }</pre>	
Parameter	Value	Description						
calificador	<pre>{ "CalificadorID": 2, "Nombre": "novelas" }</pre>							

Get de calificadores	Se muestran los calificadores	<pre>[{ "CalificadorID": 0, "Nombre": "string" }]</pre> <p>Response Content Type <input type="text" value="application/json"/></p> <p>Try it out! Hide Response</p> <p>Curl</p> <pre>curl -X GET --header 'Accept: application/json' 'http://localhost:</pre> <p>Request URL</p> <pre>http://localhost:58800/api/Calificadors</pre> <p>Response Body</p> <pre>[{ "CalificadorID": 1, "Nombre": "prueba" }, { "CalificadorID": 2, "Nombre": "novelas" }]</pre>
-------------------------	----------------------------------	---

Delete calificador	Se borran los datos del calificador	<div><div>Parameters</div><table><thead><tr><th>Parameter</th><th>Value</th><th>Description</th><th>Parameter</th></tr></thead><tbody><tr><td>id</td><td>2</td><td></td><td>path</td></tr></tbody></table><div><div>Try it out!</div>Hide Response</div><div>Curl</div><pre>curl -X DELETE --header 'Accept: application/json' 'http://localhost:58800/api/Calificadors/2'</pre><div>Request URL</div><pre>http://localhost:58800/api/Calificadors/2</pre><div>Response Body</div><pre>no content</pre></div>	Parameter	Value	Description	Parameter	id	2		path
Parameter	Value	Description	Parameter							
id	2		path							
Get por parámetro calificador	Muestra el calificador por parámetro	<div><div>Response Content Type</div><div>application/json</div><div>Parameters</div><table><thead><tr><th>Parameter</th><th>Value</th></tr></thead><tbody><tr><td>id</td><td>1</td></tr></tbody></table><div><div>Try it out!</div>Hide Response</div><div>Curl</div><pre>curl -X GET --header 'Accept: application/json' 'http://localhost:58800/api/Calificadors/1'</pre><div>Request URL</div><pre>http://localhost:58800/api/Calificadors/1</pre><div>Response Body</div><pre>[{ "CalificadorID": 1, "Nombre": "prueba" }]</pre></div>	Parameter	Value	id	1				
Parameter	Value									
id	1									

Put Tipo calificadores	Se actualiza los calificadores	<table><tr><th>Parameter</th><th>Value</th><th>Description</th><th>Parameter Type</th><th>Data Type</th></tr><tr><td>id</td><td>1</td><td></td><td>path</td><td>integer</td></tr><tr><td>calificador</td><td><pre>{ "CalificadorID": 1, "Nombre": "poema" }</pre></td><td></td><td>body</td><td>Model</td></tr><tr><td colspan="4">Parameter content type: application/json</td><td>Example Value <pre>{ "CalificadorID": 0, "Nombre": "string", "Calificaciones": [{ "CalificacionID": "PeliculaID": 0, "CalificadorID": "Estrellas": 0, "Peliculas": { "PeliculasID":</pre></td></tr></table>	Parameter	Value	Description	Parameter Type	Data Type	id	1		path	integer	calificador	<pre>{ "CalificadorID": 1, "Nombre": "poema" }</pre>		body	Model	Parameter content type: application/json				Example Value <pre>{ "CalificadorID": 0, "Nombre": "string", "Calificaciones": [{ "CalificacionID": "PeliculaID": 0, "CalificadorID": "Estrellas": 0, "Peliculas": { "PeliculasID":</pre>
Parameter	Value	Description	Parameter Type	Data Type																		
id	1		path	integer																		
calificador	<pre>{ "CalificadorID": 1, "Nombre": "poema" }</pre>		body	Model																		
Parameter content type: application/json				Example Value <pre>{ "CalificadorID": 0, "Nombre": "string", "Calificaciones": [{ "CalificacionID": "PeliculaID": 0, "CalificadorID": "Estrellas": 0, "Peliculas": { "PeliculasID":</pre>																		
Get Tipo Involucrados	Muestra el tipo de Involucrados	<pre>curl -X GET --header 'Accept: application/json' 'http://localhost:58800/api/TipoInvolucrados'</pre> <p>Request URL</p> <pre>http://localhost:58800/api/TipoInvolucrados</pre> <p>Response Body</p> <pre>[{ "TipoInvolucradoID": 1, "NombreInvolucrado": "pruebaTipo" }]</pre>																				
Post Tipo Involucrados	Agrega involucrados	<table><tr><th>Parameter</th><th>Value</th><th>Description</th><th>Parameter Type</th><th>Data Type</th></tr><tr><td>tipoInvolucrado</td><td><pre>{ "TipoInvolucradoID": , "NombreInvolucrado": "luis" }</pre></td><td></td><td>body</td><td>Model</td></tr><tr><td colspan="4">Parameter content type: application/json</td><td>Example Value <pre>{ "TipoInvolucradoID": "NombreInvolucrado": "Involucrado": [{ "InvolucradoID": "NombreInvolucrado": "PeliculaID": "TipoInvolucradoID": "PaginaWeb": "Foto": "str"</pre></td></tr></table> <p>Try it out! Hide Response</p>	Parameter	Value	Description	Parameter Type	Data Type	tipoInvolucrado	<pre>{ "TipoInvolucradoID": , "NombreInvolucrado": "luis" }</pre>		body	Model	Parameter content type: application/json				Example Value <pre>{ "TipoInvolucradoID": "NombreInvolucrado": "Involucrado": [{ "InvolucradoID": "NombreInvolucrado": "PeliculaID": "TipoInvolucradoID": "PaginaWeb": "Foto": "str"</pre>					
Parameter	Value	Description	Parameter Type	Data Type																		
tipoInvolucrado	<pre>{ "TipoInvolucradoID": , "NombreInvolucrado": "luis" }</pre>		body	Model																		
Parameter content type: application/json				Example Value <pre>{ "TipoInvolucradoID": "NombreInvolucrado": "Involucrado": [{ "InvolucradoID": "NombreInvolucrado": "PeliculaID": "TipoInvolucradoID": "PaginaWeb": "Foto": "str"</pre>																		
Delete Tipo Involucrados	Borra involucrados	<table><tr><td>id</td><td>0</td><td></td><td>path</td><td>integer</td></tr></table> <p>Try it out! Hide Response</p> <p>Curl</p> <pre>curl -X DELETE --header 'Accept: application/json' 'http://localhost:58800/api/TipoInvolucrados/0'</pre> <p>Request URL</p> <pre>http://localhost:58800/api/TipoInvolucrados/0</pre> <p>Response Body</p> <pre>no content</pre>	id	0		path	integer															
id	0		path	integer																		

Get Tipo Calificador	Se muestra los tipos de calificador por parametro	<div><div>Parameters</div><table><thead><tr><th>Parameter</th><th>Value</th><th>Description</th><th>Parameter Type</th><th>Data Type</th></tr></thead><tbody><tr><td>id</td><td>1</td><td></td><td>path</td><td>integer</td></tr><tr><td>nombre</td><td>pruebaTipo</td><td></td><td>query</td><td>string</td></tr></tbody></table><div><div>Try it out!</div><div>Hide Response</div></div><div>Curl</div><pre>curl -X GET --header 'Accept: application/json' 'http://localhost:58800/api/TipoInvolucrados/1?nombre=pruebaTipo'</pre><div>Request URL</div><pre>http://localhost:58800/api/TipoInvolucrados/1?nombre=pruebaTipo</pre><div>Response Body</div><pre>[{ "TipoInvolucradoID": 1, "NombreInvolucrado": "pruebaTipo" }</pre></div>	Parameter	Value	Description	Parameter Type	Data Type	id	1		path	integer	nombre	pruebaTipo		query	string
Parameter	Value	Description	Parameter Type	Data Type													
id	1		path	integer													
nombre	pruebaTipo		query	string													
PUT Tipo involucrados	Actualiza los datos	<div><div><div>id</div><div>1</div><div>path</div><div>integer</div></div><div><div>tipoInvolucrado</div><div><pre>{ "TipoInvolucradoID": 1, "NombreInvolucrado": "paul walker" }</pre></div><div>body</div><div>Model</div><div>Example Value</div><pre>{ "TipoInvolucradoID": 1, "NombreInvolucrado": "paul walker", "Involucrado": [{ "InvolucradoID": 1, "NombreInvolucrado": "paul walker", "PeliculaID": 0, "TipoInvolucradoID": 1, "PaginaWeb": "string", "Foto": "string", "Estado": true }] }</pre></div><div>Parameter content type:</div><div>application/json</div></div> <div>Response Messages</div> <table><thead><tr><th>HTTP Status Code</th><th>Reason</th><th>Response Model</th></tr></thead></table>	HTTP Status Code	Reason	Response Model												
HTTP Status Code	Reason	Response Model															
Post de Usuario	Se ingresa al sistema y la contraseña se encripta	<div><div>Response Content Type</div><div>application/json</div><div>Parameters</div><table><thead><tr><th>Parameter</th><th>Value</th><th>Description</th><th>Parameter Type</th><th>Data Type</th></tr></thead><tbody><tr><td>usuario</td><td><pre>{ "Username": "andreyszcr", "Nombre": "andrey", "Apellidos": "Sanchez Zuñiga", "Email": "andreyszcr@gmail.com", "Contraseña": "andre3093sa", "Estado": true, }</pre></td><td></td><td>body</td><td>Model</td></tr></tbody></table><div>Parameter content type:</div><div>application/json</div><div><div>Try it out!</div><div>Hide Response</div></div><div>Curl</div><pre>curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' -d '{ "Username": "andreyszcr", "Nombre": "andrey", "Apellidos": "Sanchez Zuñiga", "Email": "andreyszcr40@gmail.com", "Contraseña": "andre3093sa", "Estado": true, }' 'http://localhost:58800/api/Usuarios'</pre></div>	Parameter	Value	Description	Parameter Type	Data Type	usuario	<pre>{ "Username": "andreyszcr", "Nombre": "andrey", "Apellidos": "Sanchez Zuñiga", "Email": "andreyszcr@gmail.com", "Contraseña": "andre3093sa", "Estado": true, }</pre>		body	Model					
Parameter	Value	Description	Parameter Type	Data Type													
usuario	<pre>{ "Username": "andreyszcr", "Nombre": "andrey", "Apellidos": "Sanchez Zuñiga", "Email": "andreyszcr@gmail.com", "Contraseña": "andre3093sa", "Estado": true, }</pre>		body	Model													

<p>PUT Usuarios</p>	<p>Actualiza los datos y la contraseña se vuelve a encriptar</p>	

Conclusiones y recomendaciones

Conclusiones:

Se llega a la conclusión que trabajar con API REST se obtienen grandes ventajas, por ejemplo:

- Separación entre el cliente y el servidor: el protocolo REST separa totalmente la interfaz de usuario del servidor y el almacenamiento de datos
- Visibilidad, fiabilidad y escalabilidad: Se puede migrar a otros servidores o realizar todo tipo de cambios en la base de datos, siempre y cuando los datos de cada una de las peticiones se envíen de forma correcta.
- La API REST siempre es independiente del tipo de plataformas o lenguajes: la API REST siempre se adapta al tipo de sintaxis o plataformas con las que se estén trabajando, lo que ofrece una gran libertad a la hora de cambiar o probar nuevos entornos dentro del desarrollo.

Recomendaciones:

- Recordemos que ¡Una API es una interfaz de usuario (UI) para un desarrollador – al igual que cualquier UI, es importante asegurar que la experiencia del usuario esté pensada cuidadosamente!
- Preparar y verificar que tengamos una buena documentación, recordemos que API es tan buena como lo es su documentación – por lo tanto, realiza una buena documentación.
- Usa JSON donde sea posible, XML sólo si tienes la obligación, a la vez debemos considerar usar JSON para cuerpos de peticiones POST, PUT entre otras.