

# PROYECTO FINAL

API administrativo de horas para el servicio social.

**Profesor:** Rojano Caceres José Rafael

**EE:** Tecnologías para la integración de soluciones

**Fecha:** Febrero - Julio de 2022

## Equipo 1

- ♦ Omar Alejandro Alonso Lizardi
- ♦ Nadia Itzel Bravo Guevara
- ♦ Karla Fernanda Guevara Flores

# Índice

1. Introducción.....	3
2. Motivación .....	4
3. Problemática.....	4
4. Solución.....	5
5. Costos .....	6
6. Diagrama de despliegue.....	7
7. Diagrama de casos de uso .....	8
8. Diagrama E-R.....	9
9. Documentación del API (SOAP Y REST) .....	10
Semana 1 (REST) .....	10
Semana 2 (REST) .....	12
Semana 3 (SOAP) .....	13
10. Plan de pruebas.....	15
Semana 1 (REST) .....	15
Semana 2 (REST) .....	20
Semana 3 (SOAP) .....	23
11. Ejecución de contenedores .....	24

## 1. Introducción

En este documento se presenta el proyecto desarrollado por estudiantes de la Universidad Veracruzana de la Facultad de Estadística e Informática con los conocimientos que fueron adquiridos a lo largo de la experiencia educativa de Tecnologías para la integración de soluciones. Para este proyecto se desarrollará un API que apoye en la administración de horas que realizan los alumnos durante su servicio social así mismo como apoyo a los coordinadores para poder realizar una correcta administración de alumnos y dependencias.

Para comenzar explicaremos a continuación los protocolos aprendidos y que utilizaremos en este proyecto, los cuales son:

- ◆ **SOAP:** Es un protocolo que hace uso de diferentes procesos para que puedan comunicarse por medio de intercambios de datos en XML. Es un paradigma de mensajería de dirección sin estado, que puede ser utilizado para formar protocolos más completos y complejos según las necesidades de las aplicaciones que lo implementan.
- ◆ **REST:** Interfaz para conectar varios sistemas basados en el protocolo HTTP Y sirve para obtener y generar datos y operaciones, devolviéndolos en formatos como XML y JSON.

REST se apoya en HTTP donde hace uso de:

- ◆ **Post:** Para crear recursos nuevos.
- ◆ **Get:** Para obtener un listado o un recurso en concreto.
- ◆ **Put:** Para modificar.
- ◆ **Patch:** Para modificar un recurso.
- ◆ **Delete:** Para borrar un recurso.

## **2. Motivación**

La motivación para desarrollar este sistema es el poder crear un API que apoye tanto a estudiantes como coordinadores en la agilización del conteo de horas que se realizan durante el servicio social, así como la reducción de uso de papel, ya que creemos que tiene un uso desmedido el cual podría controlarse mediante el uso de una API que permita llevar el control de este.

## **3. Problemática**

Durante la lluvia de ideas que tuvimos para poder elegir un problema al cual darle solución, nos dimos cuenta de uno que la mayoría de los miembros del equipo presentaban es en cuanto al papeleo que manejan en su servicio social, ya que mensualmente se debe entregar al respectivo coordinador un reporte mensual donde se especifiquen las horas (entre otros datos) que el alumno cumplió durante el mes.

En cuanto a la parte de coordinación, encontramos el problema del papeleo ya que un coordinador no solo cuenta con un alumno, sino que se encarga de llevar la administración de todos los estudiantes de su respectiva carrera y las dependencias donde se encuentran dichos estudiantes.

El hecho de tener todos estos registros en papel lo encontramos como un riesgo de seguridad, ya que pueden ocurrir plagios o incidentes que pongan en riesgo la integridad de la información manejada en ellos.

## 4. Solución

Nuestra propuesta de solución para la problemática antes planteada es crear un API con 2 microservicios que haga uso uno de SOAP y el otro de REST que le permita a alumnos y coordinadores realizar esta tarea de una manera mas segura y rápida. Ya que al crear este API se pueden generar diferentes clientes que hagan uso de este servicio.

En nuestro API se podrá realizar lo siguiente:

Usamos SOAP para:

- Registrar horas (POST)
- Consultar horas (tanto realizadas y por terminar) (GET)
- Generar reporte mensual (GET)

Usamos REST para:

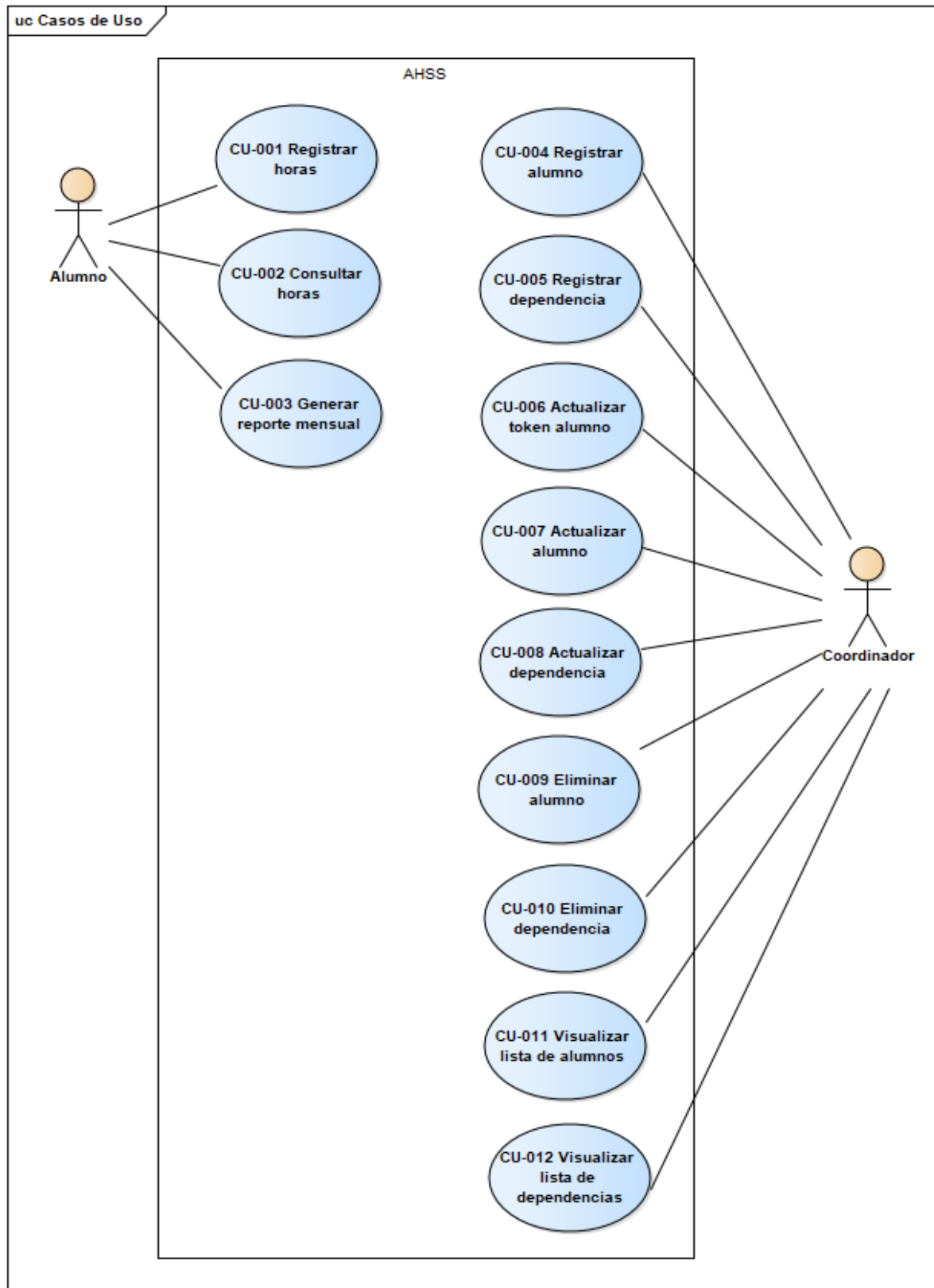
- Registrar alumno (POST)
- Registrar dependencia (POST)
- Actualizar token alumno (PUT)
- Actualizar alumno (PUT)
- Actualizar dependencia (PUT)
- Eliminar alumno (DELETE)
- Eliminar dependencia (DELETE)
- Visualizar lista de alumnos (GET)
- Visualizar lista de dependencias (GET)

## 5. Costos

Pago a desarrolladores

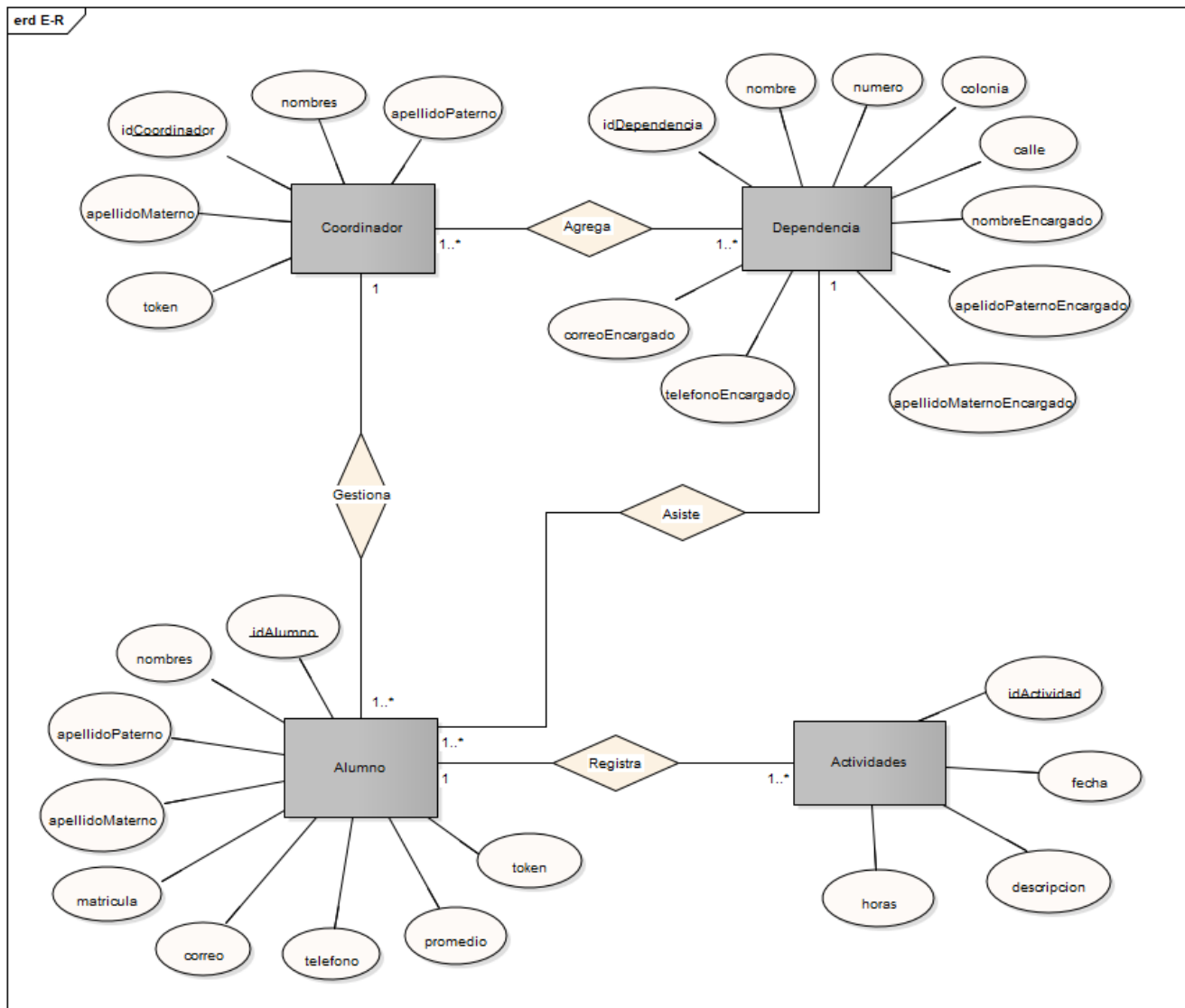
## 6. Diagrama de despliegue

## 7. Diagrama de casos de uso





## 8. Diagrama E-R



9. Documentación del API (SOAP Y REST)

Semana 1 (REST)

	EndPoint	PARAMS	BODY	PARAM DEVUELTOS
Actualizar token Alumno	mysterious-eyrie-42583.herokuapp.com/alumnos/{idAlumno}/actualizarToken (PUT)  Requiere autorización header Authorization [token]	IdAlumno = El id del alumno al cual se le actualizara el token	En este caso no lleva	El endpoint regresa un String, el cual contiene un mensaje de acuerdo con los resultados del proceso.
Actualizar alumno	mysterious-eyrie-42583.herokuapp.com/alumnos/{idAlumno}/actualizar (PUT)  Requiere autorización header Authorization [token]	idAlumno = el id del alumno al cual se le actualizaran los datos	<ul style="list-style-type: none"><li>• <b>idAlumno:</b> de tipo int</li><li>• <b>nombres:</b> de tipo String</li><li>• <b>apellidoPaterno:</b> de tipo String</li><li>• <b>apellidoMaterno:</b> de tipo String</li><li>• <b>matricula:</b> de tipo String</li><li>• <b>correo:</b> de tipo String</li><li>• <b>token:</b> de tipo String</li><li>• <b>promedio:</b> de tipo Double</li><li>• <b>idCoordinador:</b> de tipo int</li><li>• <b>idDependencia:</b> de tipo int</li></ul>	El endpoint regresa un String, el cual contiene un mensaje de acuerdo con los resultados del proceso.

<b>Registrar dependencias</b>	<p>mysterious-eyrie-42583.herokuapp.com/dependencias <b>(POST)</b></p> <p>Requiere autorización header Authorization [token]</p>	<pre>{   "idDependencia": 1,   "nombre": "Dependencia 1",   "colonia": "Colonia 1",   "calle": "Calle 1",   "numero": 1,   "nombreEncargado": "Nadia",   "apellidoPaternoEncargado": "Bravo",   "apellidoMaternoEncargado": "Guevara",   "correoEncargado": "nadia@correo.com",   "telefono": "2288776655",   "idCoordinador": 1 }</pre>	<ul style="list-style-type: none"> <li>• <b>idDependencia:</b> de tipo int</li> <li>• <b>nombre:</b> de tipo String</li> <li>• <b>colonia:</b> de tipo String</li> <li>• <b>calle:</b> de tipo String</li> <li>• <b>numero:</b> de tipo int</li> <li>• <b>nombreEncargado:</b> de tipo String</li> <li>• <b>apellidoPaternoEncargado:</b> de tipo String</li> <li>• <b>apellidoMaternoEncargado:</b> de tipo String</li> <li>• <b>correoEncargado:</b> de tipo String</li> <li>• <b>teléfono:</b> de tipo String</li> <li>• <b>idCoordinador:</b> de tipo int</li> </ul>	<p>El endpoint regresa un String, el cual contiene un mensaje de acuerdo con los resultados del proceso.</p>
<b>Registrar alumno</b>	<p>mysterious-eyrie-42583.herokuapp.com/alumnos <b>(POST)</b></p> <p>Requiere autorización header Authorization [token]</p>	<pre>{   "idAlumno": 0,   "nombres": "omar",   "apellidoPaterno": "Alonso",   "apellidoMaterno": "Lizardi",   "matricula": "1234567890",   "correo": "correo@correo.com",   "token": "123456",   "promedio": 6.5,   "idCoordinador": 1,   "idDependencia": 1,   "telefono": "2288990088" }</pre>	<ul style="list-style-type: none"> <li>• <b>Id Alumno</b> de tipo int.</li> <li>• <b>Nombres</b> de tipo String.</li> <li>• <b>Apellido Paterno</b> de tipo String.</li> <li>• <b>Apellido Materno</b> de tipo String.</li> <li>• <b>Matricula</b> de tipo String.</li> <li>• <b>Correo</b> de tipo String.</li> <li>• <b>Token</b> de tipo String.</li> <li>• <b>Promedio</b> de tipo Double.</li> <li>• <b>Id Coordinador</b> de tipo int.</li> <li>• <b>Id Dependencia</b> de tipo int.</li> </ul>	<p>El endpoint regresa un String, el cual contiene un mensaje de acuerdo con los resultados del proceso.</p>

			<ul style="list-style-type: none"> <li>• <b>Teléfono</b> de tipo String.</li> </ul>	
--	--	--	---	--

## Semana 2 (REST)

	EndPoint	PARAMS	BODY	PARAM DEVUELTOS
<b>Actualizar dependencia</b>	mysterious-eyrie-42583.herokuapp.com/ Requiere autorización header Authorization [token]	idDependencia: el id de la dependencia que se actualizará	<ul style="list-style-type: none"> <li>• <b>idDependencia:</b> de tipo int</li> <li>• <b>nombre:</b> de tipo String</li> <li>• <b>colonia:</b> de tipo String</li> <li>• <b>calle:</b> de tipo String</li> <li>• <b>numero:</b> de tipo int</li> <li>• <b>nombreEncargado:</b> de tipo String</li> <li>• <b>apellidoPaternoEncargado:</b> de tipo String</li> <li>• <b>apellidoMaternoEncargado:</b> de tipo String</li> <li>• <b>correoEncargado:</b> de tipo String</li> <li>• <b>teléfono:</b> de tipo String</li> <li>• <b>idCoordinador:</b> de tipo int</li> </ul>	El endpoint regresa un String, el cual contiene un mensaje de acuerdo con los resultados del proceso.
<b>Eliminar alumno</b>	mysterious-eyrie-42583.herokuapp.com/ alumnos/{idAlumno}/eliminarAlumno <b>(DELETE)</b>	idAlumno = el id del alumno el cual se eliminará	<ul style="list-style-type: none"> <li>• <b>IdAlumno:</b> de tipo int</li> </ul>	El endpoint regresa un String, el cual contiene un mensaje de acuerdo con los

	Requiere autorización header Authorization [token]			resultados del proceso.
<b>Eliminar dependencia</b>	mysterious-eyrie-42583.herokuapp.com/ /dependencias/ {idDependencia}/eliminarDependencia <b>(DELETE)</b> Requiere autorización header Authorization [token]	idDependencia: el id de la dependencia que se eliminará	<ul style="list-style-type: none"><li><b>idDependencia:</b> de tipo int int</li></ul>	El endpoint regresa un String, el cual contiene un mensaje de acuerdo con los resultados del proceso.
<b>Visualizar lista de alumnos</b>	mysterious-eyrie-42583.herokuapp.com/ Requiere autorización header Authorization [token]			El endpoint regresa un String, el cual contiene un mensaje de acuerdo con los resultados del proceso.

Semana 3 (SOAP)


--	--	--	--	--

10. Plan de pruebas

Semana 1 (REST)

Caso de uso: Registrar Alumno					
No.	Condiciones de Entrada	Entrada	Salida Esperada	Condiciones de Salida	Observaciones
01	El Coordinador tiene un token valido	{ "idAlumno": 0, "nombres": "omar", "apellidoPaterno": "Alonso", "apellidoMaterno": "Lizardi", "matricula": "1234567890", "correo": "correo@correo.com", "token": "123456", "promedio": 6.5, "idCoordinador": 1, "idDependencia": 1, "teléfono": "2288990088" }	El sistema arroja el mensaje "Alumno + "nombres" + agregado con éxito"	Un nuevo alumno registrado en la base de datos.	Ninguna, funciona con normalidad
02	El coordinador no tiene un token valido	{ "idAlumno": 0, "nombres": "omar", "apellidoPaterno": "Alonso", "apellidoMaterno": "Lizardi", "matricula": "1234567890", "correo": "correo@correo.com", "token": "123456", "promedio": 6.5, "idCoordinador": 1, "idDependencia": 1, "teléfono": "2288990088" }	El sistema arroja el mensaje "Token no valido"		Ninguna, funciona con normalidad

Caso de uso: Registrar Dependencia					
No.	Condiciones de Entrada	Entrada	Salida Esperada	Condiciones de Salida	Observaciones
03	El Coordinador tiene un token valido	{ "idDependencia": 1, "nombre": "Dependencia 1", "colonia": "Colonia 1", "calle": "Calle 1", "numero": 1, "nombreEncargado": "Nadia", "apellidoPaternoEncargado": "Bravo", "apellidoMaternoEncargado": "Guevara", "correoEncargado": "nadia@correo.com", "telefono": "2288776655", "idCoordinador": 1 }	El sistema arroja el mensaje "Dependencia + "nombre" + agregada con éxito"	Una nueva dependencia registrada en la base de datos.	Ninguna, funciona con normalidad
04	El coordinador no tiene un token valido	{ "idDependencia": 1, "nombre": "Dependencia 1", "colonia": "Colonia 1", "calle": "Calle 1", "numero": 1, "nombreEncargado": "Nadia", "apellidoPaternoEncargado": "Bravo",	El sistema arroja el mensaje "Token no valido"		Ninguna, funciona con normalidad



		<pre>"apellidoMaternoEncargado": "Guevara", "correoEncargado": "nadia@correo.com", "telefono": "2288776655", "idCoordinador": 1 }</pre>			
--	--	---	--	--	--

Caso de uso: Actualizar token alumno					
No.	Condiciones de Entrada	Entrada	Salida Esperada	Condiciones de Salida	Observaciones
05	El Coordinador tiene un token valido	<pre>{   "idAlumno": 0,   "nombres": "omar",   "apellidoPaterno": "Alonso",   "apellidoMaterno": "Lizardi",   "matricula": "1234567890",   "correo": "correo@correo.com",   "token": "123456",   "promedio": 6.5,   "idCoordinador": 1,   "idDependencia": 1,   "teléfono": "2288990088" }</pre>	El sistema arroja el mensaje "Token actualizado"	Los nuevos datos del alumno se han actualizado en el a base de datos.	Ninguna, funciona con normalidad

06	El coordinador no tiene un token valido	{ "idAlumno": 0, "nombres": "omar", "apellidoPaterno": "Alonso", "apellidoMaterno": "Lizardi", "matricula": "1234567890", "correo": "correo@correo.com", "token": "12345", "promedio": 6.5, "idCoordinador": 1, "idDependencia": 1, "teléfono": "2288990088" }	El sistema arroja el mensaje "Token no valido"		Ninguna, funciona con normalidad
----	---	--	--	--	----------------------------------

Caso de uso: Actualizar alumno					
No.	Condiciones de Entrada	Entrada	Salida Esperada	Condiciones de Salida	Observaciones
07	El Coordinador tiene un token valido	{ "idAlumno": 0, "nombres": "omar alejandro", "apellidoPaterno": "Alonso", "apellidoMaterno": "Lizardi", "matricula": "1234567890", "correo": "correo@correo.com", "token": "123456", "promedio": 6.5, "idCoordinador": 1, "idDependencia": 1, "teléfono": "2288990088" }	El sistema arroja el mensaje "Alumno actualizado"	Los datos del alumno se actualizan en la base de datos.	Ninguna, funciona con normalidad

		}			
08	El coordinador no tiene un token valido	{ "idAlumno": 0, "nombres": "omar alejandro", "apellidoPaterno": "Alonso", "apellidoMaterno": "Lizardi", "matricula": "1234567890", "correo": "correo@correo.com", "token": "123456", "promedio": 6.5, "idCoordinador": 1, "idDependencia": 1, "teléfono": "2288990088" }	El sistema arroja el mensaje "Token no valido"		Ninguna, funciona con normalidad

## Semana 2 (REST)

Caso de uso: Eliminar alumno					
No.	Condiciones de Entrada	Entrada	Salida Esperada	Condiciones de Salida	Observaciones
09	El Coordinador tiene un token valido	idAlumno: 1	El sistema arroja el mensaje "Alumno eliminado"	Los datos del alumno son eliminados de la base de datos	Ninguna, funciona con normalidad
10	El coordinador no tiene un token valido	idAlumno: 1	El sistema arroja el mensaje "Token no valido"		Ninguna, funciona con normalidad

Caso de uso: Eliminar dependencia					
11	El Coordinador tiene un token valido	idDependencia: 1	El sistema arroja el mensaje "Dependencia eliminada"	Los datos de la dependencia son eliminados de la base de datos.	Ninguna, funciona con normalidad
12	El coordinador no tiene un token valido	idDependencia: 1	El sistema arroja el mensaje "Token no valido"		Ninguna, funciona con normalidad

Caso de uso: Actualizar dependencia

13	El Coordinador tiene un token valido	{ "idDependencia": 1, "nombre": "Dependencia 1", "colonia": "Colonia 1", "calle": "Calle 1", "numero": 1, "nombreEncargado": "Nadia Itzel", "apellidoPaternoEncargado": "Bravo", "apellidoMaternoEncargado": "Guevara", "correoEncargado": "nadia@correo.com", "telefono": "2288776655", "idCoordinador": 1 }	El sistema arroja el mensaje "Dependencia +nombre+ actualizada"	Los datos de la dependencia se actualizan en la base de datos.	Ninguna, funciona con normalidad
14	El coordinador no tiene un token valido	{ "idDependencia": 1, "nombre": "Dependencia 1", "colonia": "Colonia 1", "calle": "Calle 1", "numero": 1, "nombreEncargado": "Nadia Itzel", "apellidoPaternoEncargado": "Bravo", "apellidoMaternoEncargado": "Guevara", }	El sistema arroja el mensaje "Token no valido"		Ninguna, funciona con normalidad

		<pre>"correoEncargado": "nadia@correo.com", "telefono": "2288776655", "idCoordinador": 1 }</pre>			
--	--	--	--	--	--

Caso de uso: Visualizar lista de alumnos					
15	El Coordinador tiene un token valido				Ninguna, funciona con normalidad
16	El coordinador no tiene un token valido		El sistema arroja el mensaje "Token no valido"		Ninguna, funciona con normalidad

Semana 3 (SOAP)

Caso de uso:

Caso de uso:					

11. Ejecución de contenedores



