

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
10. Plan de pruebas	¡Error! Marcador no definido.
11. Ejecución de contenedores.....	12

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

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:

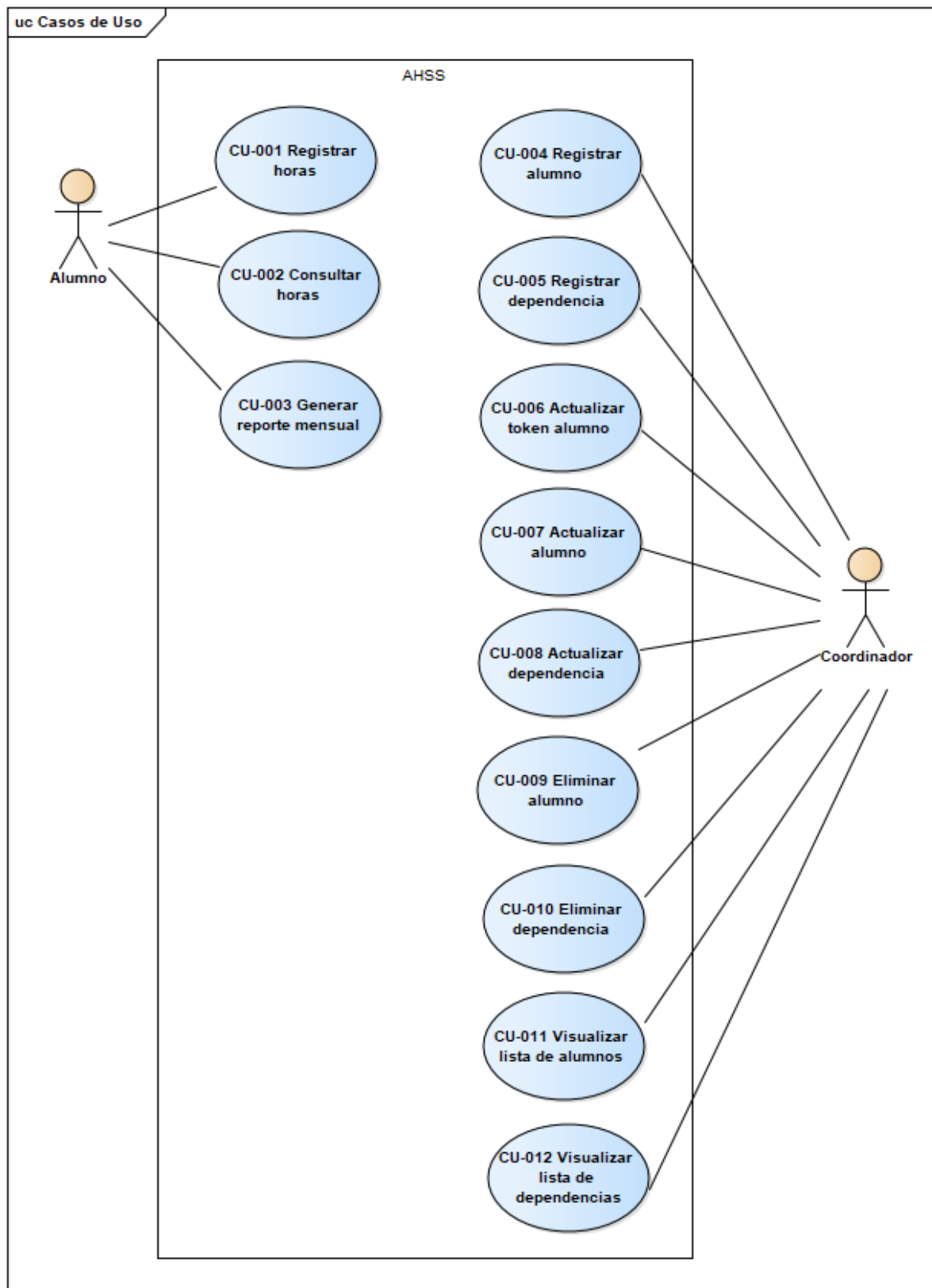
- Registrar alumno (POST)
- Registrar dependencia (POST)
- Actualizar token alumno (PUT)
- Actualizar alumno (PUT)
- Actualizar dependencia (PUT)
- Eliminar alumno (DELETE)
- Eliminar dependencia (DELETE)
- Visualizar lista de alumnos (GET)
- Visualizar lista de dependencias (GET)

5. Costos

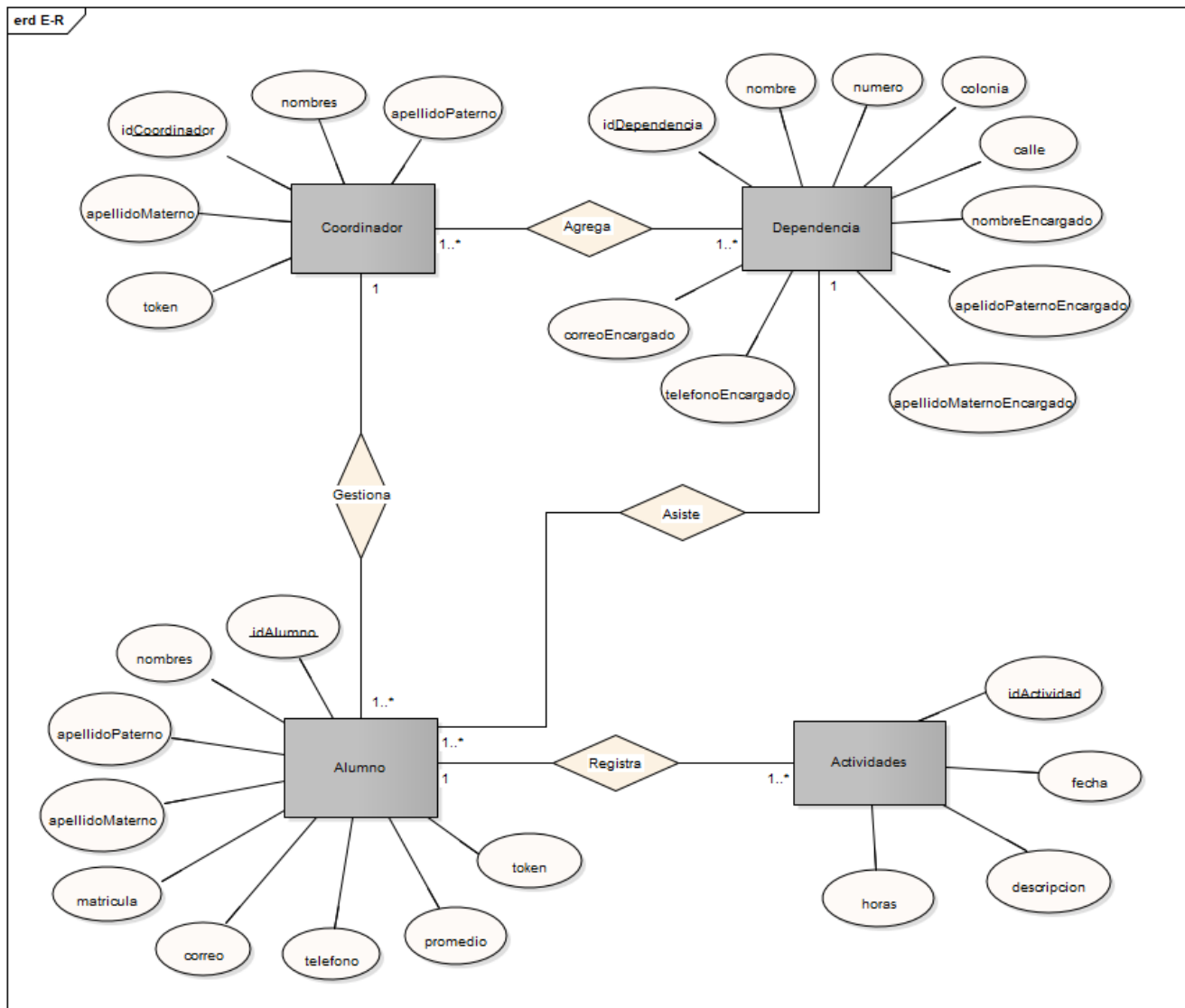
Pago a desarrolladores

6. Diagrama de despliegue

7. Diagrama de casos de uso



8. Diagrama E-R



9. Documentación del API (SOAP Y REST)

	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">• idAlumno: de tipo int• nombres: de tipo String• apellidoPaterno: de tipo String• apellidoMaterno: de tipo String• matricula: de tipo String• correo: de tipo String• token: de tipo String• promedio: de tipo Double• idCoordinador: de tipo int• idDependencia: de tipo int	El endpoint regresa un String, el cual contiene un mensaje de acuerdo con los resultados del proceso.
Registrar dependencias	mysterious-eyrie-42583.herokuapp.com/dependencias (POST) Requiere autorización header Authorization [token]	{ "idDependencia": 1, "nombre": "Dependencia 1",	<ul style="list-style-type: none">• idAlumno: de tipo int int• nombre: de tipo String• colonia: de tipo String• calle: de tipo String	El endpoint regresa un String, el cual

		<pre> "colonia": "Colonia 1", "calle": "Calle 1", "numero": 1, "nombreEncargado": "Nadia", "apellidoPaterno": "Bravo", "apellidoMaterno": "Guevara", "correoEncargado": "nadia@correo.com", "telefono": "2288776655", "idCoordinador": 1 } </pre>	<ul style="list-style-type: none"> • numero: de tipo int • nombreEncargado: de tipo String • apellidoPaterno: de tipo String • apellidoMaterno: de tipo String • correoEncargado: de tipo String • teléfono: de tipo String • idCoordinador: de tipo int 	<p>contiene un mensaje de acuerdo con los resultados del proceso.</p>
Registrar alumno	<p>mysterious-eyrie-42583.herokuapp.com/alumnos (POST)</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, "teléfono": "2288990088" } </pre>	<ul style="list-style-type: none"> • Id Alumno de tipo int. • Nombres de tipo String. • Apellido Paterno de tipo String. • Apellido Materno de tipo String. • Matricula de tipo String. • Correo de tipo String. • Token de tipo String. • Promedio de tipo Double. • Id Coordinador de tipo int. • Id Dependencia de tipo int. • Teléfono de tipo String. 	<p>El endpoint regresa un String, el cual contiene un mensaje de acuerdo con los resultados del proceso.</p>

1. Plan de pruebas

2. Ejecución de contenedores

