

Servicio de contratos de electricidad

Integrantes:

Bryan Alberto Duran Cuellar

Pablo Lucas Negrete

Luis Armando Villa Ceballos

Iván Antonio García Campos

Equipo no.4



Experiencia educativa: Integración de soluciones

Profesor: José Rafael Rojano Cáceres

ÍNDICE

Introducción.....	2
Motivación.....	3
Problemática.....	4
Solución.....	5
Operaciones por SOAP	5
Operaciones por REST	6
Costos.....	7
Heroku	7
Clever Cloud.....	7
Modelo de despliegue.....	9
Documentación del API SOAP y REST	10
Endpoint SOAP.....	10
Endpoint REST.....	11
Parámetros de recepción	11
SOAP	11
REST.....	12
Parámetros devueltos	13
SOAP	13
Plan de pruebas	15
SOAP	15
REST.....	16
Forma de ejecución de los contenedores.....	16
Repositorio en GitHub.....	16
Documentación en los archivos Docker file y Docker-compose.yml	16
Microservicio 1	16
Microservicio 2	17

Introducción

El diseño del software tiende a ser cada vez más modular. Los Servicios Web nos permitirán distribuir nuestra aplicación a través de Internet, pudiendo una aplicación utilizar los servicios ofrecidos por cualquier servidor conectado a Internet. Y para el desarrollo de estos servicios web se hacen uso de tecnologías como SOAP y REST que son dos enfoques distintos para la transmisión de datos, ambos definen como diseñar interfaces de programación de aplicaciones (API).

Para la realización de este proyecto se hará uso de tecnologías SOAP y REST para realizar los servicios que haremos, también se hará uso del lenguaje de programación Java, y finalmente Docker para contenerizar nuestros servicios.

Motivación

Nuestra motivación es el pensar en el resultado que este proyecto puede darnos en muchos factores, en primer lugar, por la experiencia que estaremos adquiriendo aunada a un montón de conocimiento que ganaremos además del enseñado por nuestro profesor; misma experiencia que nos será de gran ayuda en nuestra vida laboral. También nos dará una idea definida acerca de la manera en que están hechos muchos de los servicios que utilizamos y consumimos cotidianamente como lo son los servicios web a disposición de la población, los cuales facilitan acciones necesarias que antes conllevaban una inversión de tiempo mayor por lo tanto ofrecen una facilidad y ahorro de tiempo entre otros recursos.

Problemática

La globalización, el aumento del número y velocidad de las transacciones y la movilidad, provocados por la rápida evolución de la tecnología han dejado obsoleta la forma de atender los negocios. Para un gran número de empresas es casi un «obligación» ofrecer servicios web y cada vez más frecuentemente las empresas nacen con el objetivo único de que sus servicios se consulten en internet. Las ventajas son múltiples, destacando el aumento de clientes potenciales, la ausencia de horario comercial y la comodidad de poder ofrecer a nuestros clientes administrar su servicio desde cualquier ubicación. Ahora bien, todos estos beneficios se ponen en ejecución para ofrecer a los clientes los medios adecuados para que las contrataciones se realicen lo más rápidas, seguras y sencillas posibles, así como la administración de sus servicios.

Solución

Como solución pensamos en desarrollar un servicio web para la creación de contratos y para la renta de luz que se lleva a cabo bimestralmente, con la finalidad de reducir los tiempos de espera en lugares físicos y hacerlo de forma más rápida y segura en el servicio.

Donde se pretende que los personas cuenten con las siguientes operaciones:

Operaciones por SOAP

- **Agregar contrato:** La persona dará de alta su servicio, con su número de contrato, además proporcionará datos personales como su nombre completo, domicilio, número telefónico, firma electrónica y el número de contrato se genera automáticamente.
- **Cancelar Servicio:** La persona accede al servicio mediante número de contrato y firma electrónica, entonces el servicio será cancelado.
- **Consultar Servicio:** La persona accede al servicio mediante su número de contrato y su firma electrónica.
- **Modificar Servicio:** La persona accede al servicio mediante su número de contrato y firma electrónica así modificara su contrato de luz. Luego el usuario proporcionara los datos a modificar.
- **Mostrar contratos:** Las personas podrán visualizar todos los contratos, podrá visualizar el nombre, el número de contrato y la firmae.

Operaciones por REST

- **Generación de firma electrónica:** La persona generara su firma electrónica proporcionando su nombre completo.
- **Historial de pago:** La persona podrá ver su consumo de energía e importe del mes anterior, proporcionando firma electrónica.
- **Pagar:** La persona pagara su servicio proporcionando su número de contrato y la cantidad a pagar.

Costos

Se realizó la investigación de distintos servicios de hosting, que también contarán con compatibilidad con Docker, y encontramos los siguientes:

Heroku



Heroku es una plataforma basada en la nube como servicio (PaaS) diseñada para que los desarrolladores y equipos creen, envíen, supervisen y escalen aplicaciones modernas. Esta plataforma basada en contenedores brinda a los desarrolladores más tiempo para centrarse en el producto principal sin tener que preocuparse de mantener la infraestructura de la aplicación. Heroku ofrece herramientas, servicios y flujos de trabajo integrados para ayudar a las organizaciones de todos los tamaños a maximizar la productividad individual y del equipo para poder entregar aplicaciones en el mercado más rápidamente.

Y cabe destacar que Heroku permite usarse totalmente gratis, aunque es obvio que al usarlo gratuitamente no se contara con las mismas características si se cuenta con algún plan de pago. Un plan gratuito ofrece:

- RAM: 512 MBs.
- Permite el despliegue desde Git.
- 2 procesadores.
- Dominios personalizados.

Fuente: Pricing | Heroku (<https://www.heroku.com/pricing>)

Clever Cloud



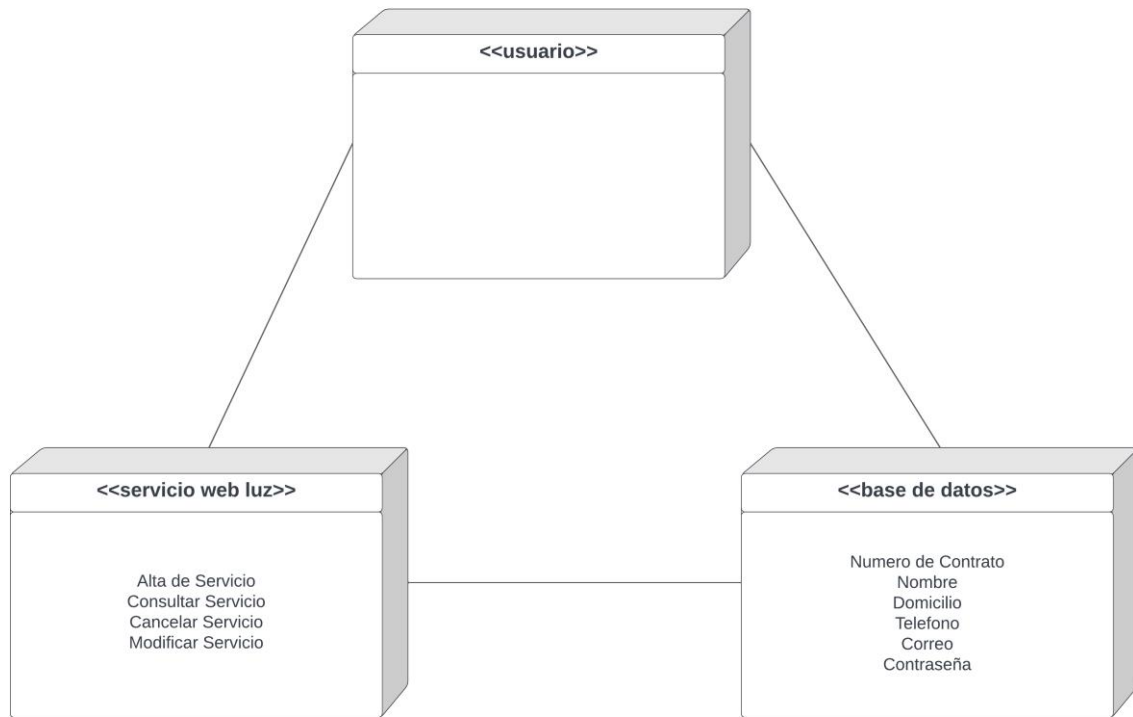
Clever Cloud es una plataforma como servicio que ha sido diseñada para ayudar a las organizaciones a operar, automatizar y ampliar sus negocios con diversos tiempos de ejecución y complementos. Permite a los equipos obtener información general del estado actual de los Scalers y la actividad presente de la memoria RAM, la CPU, el disco y la red. Entre las principales funciones de Clever Cloud, figuran la administración de aplicaciones, redimensionamiento de aplicaciones, migración de bases de datos, nombres de dominio y enlaces de implementación.

Clever Cloud será usado para que los registros que se hagan permanezcan, debido a que en Heroku tras un periodo de inactividad se reinicia, y no los datos no persisten. El plan básico de Clever Cloud de base de datos MySQL es el siguiente:

	PLAN NAME	BACKUPS	LOGS	MAX CONNECTION LIMIT	MAX DB SIZE	MEMORY	METRICS	TYPE	VCPUS	PRICE
<input type="radio"/>	DEV	Daily - 7 Retained	No	5	10 MB	Shared	No	Shared	Shared	0.00 €

Fuente Pricing – Clever Cloud(<https://www.clever-cloud.com/pricing/#databases>)

Modelo de despliegue



Documentación del API SOAP y REST

Los endpoints son las URLs de un API o un backend que responden a una petición. Para la realización de nuestro proyecto proponemos los siguientes endpoints.

Endpoint SOAP

Debido a que son realizados por SOAP estas son mostradas desde el wsdl, con ayuda de la extensión de Wizzler.

- AgregarContrato
- CancelarServicio
- ConsultarContrato
- ModificarServicio
- MostrarContratos



Endpoint REST

Parámetros de recepción

Los parámetros de recepción serán datos que el cliente proporcionará. Además, habrá distintos tipos de datos que se recibirán, para los cuales dicha recepción será estricta y no procederá la acción del servicio si falta algún dato por proporcionar o no cumple con las características requeridas. Los parámetros de recepción serán los siguientes:

SOAP

- **AgregarContrato:** El cliente proporcionará su nombre, domicilio, número telefónico y su firma electrónica.

```
POST https://microservicio-contrato.herokuapp.com:443/ws
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <AgregarContratoRequest xmlns="https://t4is.uv.mx/contratos">
      <nombre>Enrique Lopez Fox</nombre>
      <domicilio>Los pinos #1</domicilio>
      <telefono>28565413</telefono>
      <firmae>a15s25</firmae>
    </AgregarContratoRequest>
  </Body>
</Envelope>
```

- **CancelarServicio:** El cliente proporcionará su número de contrato y firma electrónica.

```
POST https://microservicio-contrato.herokuapp.com:443/ws
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <CancelarServicioRequest xmlns="https://t4is.uv.mx/contratos">
      <ncontrato>4</ncontrato>
      <firmae>2fdgk</firmae>
    </CancelarServicioRequest>
  </Body>
</Envelope>
```

- ConsultarContrato: El cliente proporcionara su número de contrato y firma electrónica.

```
POST https://microservicio-contrato.herokuapp.com:443/ws
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <CancelarServicioRequest xmlns="https://t4is.uv.mx/contratos">
      <ncontrato>1</ncontrato>
      <firmae>aaa</firmae>
    </CancelarServicioRequest>
  </Body>
</Envelope>
```

- ModificarServicio: El cliente proporcionara su número de contrato y firma electrónica, seguido de los datos a actualizar; nombre, domicilio, y teléfono.

```
POST https://microservicio-contrato.herokuapp.com:443/ws
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <ModificarServicioRequest xmlns="https://t4is.uv.mx/contratos">
      <ncontrato>5</ncontrato>
      <firmae>gggg4444</firmae>
      <nombre>Carlos Estrada Estrada</nombre>
      <domicilio>Calle #5 Col Juarez</domicilio>
      <telefono>2896548</telefono>
    </ModificarServicioRequest>
  </Body>
</Envelope>
```

- MostrarContratos: Para esta operación no es necesario proporcionar datos.

```
POST https://microservicio-contrato.herokuapp.com:443/ws
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <MostrarContratosRequest xmlns="https://t4is.uv.mx/contratos">[any]</MostrarContratosRequest>
  </Body>
</Envelope>
```

REST

Parámetros devueltos

Los parámetros que devolverá el servicio serán dependiendo de qué operación esté siendo usada, los parámetros que se retornarán serán los datos en la base de datos en la mayoría de los casos.

SOAP

- **AgregarContrato:** Retornará un mensaje de éxito, con el nombre del que realizó el contrato. Si algún dato se encuentra vacío retornara un mensaje de error.



The screenshot shows a SOAP client interface with two examples of successful responses for the 'AgregarContrato' operation. The URL is 'https://microservicio-contrato.herokuapp.com:443/ws'.

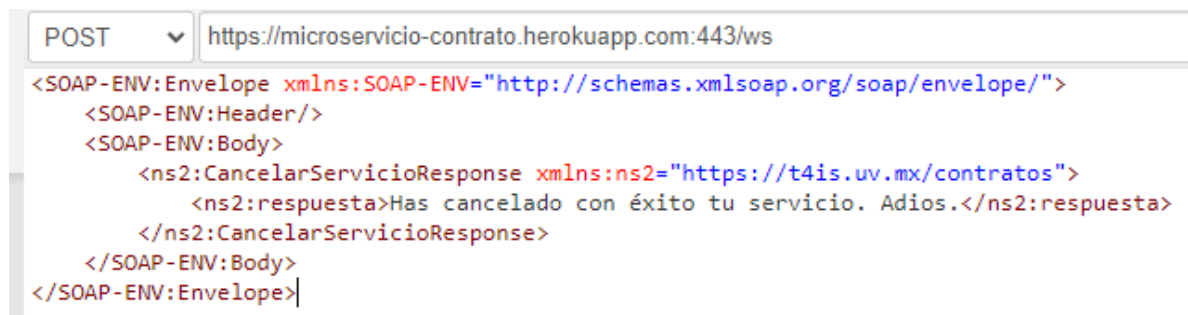
Example 1:

```
POST https://microservicio-contrato.herokuapp.com:443/ws
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:AgregarContratoResponse xmlns:ns2="https://t4is.uv.mx/contratos">
      <ns2:respuesta>Contrato Creado con los datos: Enrique Lopez Fox</ns2:respuesta>
    </ns2:AgregarContratoResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Example 2:

```
POST https://microservicio-contrato.herokuapp.com:443/ws
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:AgregarContratoResponse xmlns:ns2="https://t4is.uv.mx/contratos">
      <ns2:respuesta>Ha ocurrido un error al crear el contrato, por favor vuelve a intentarlo. Al parecer olvidaste un dato.</ns2:respuesta>
    </ns2:AgregarContratoResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- **CancelarServicio:** Retornara un mensaje de éxito, si el número de contrato coincide con la firma electrónica y que ninguno de estos dos parámetros vaya vacío.



The screenshot shows a SOAP client interface with a successful response for the 'CancelarServicio' operation. The URL is 'https://microservicio-contrato.herokuapp.com:443/ws'.

```
POST https://microservicio-contrato.herokuapp.com:443/ws
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:CancelarServicioResponse xmlns:ns2="https://t4is.uv.mx/contratos">
      <ns2:respuesta>Has cancelado con éxito tu servicio. Adios.</ns2:respuesta>
    </ns2:CancelarServicioResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- ConsultarContrato: Retornara los datos del cliente; ncontrato, nombre, domicilio, teléfono, si el número de contrato coincide con la firma electrónica y que ninguno de estos dos parámetros vaya vacío.

```
POST https://microservicio-contrato.herokuapp.com:443/ws
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:ConsultarContratoResponse xmlns:ns2="https://t4is.uv.mx/contratos">
      <ns2:contratos>
        <ns2:ncontrato>2</ns2:ncontrato>
        <ns2:nombre>ablo</ns2:nombre>
        <ns2:domicilio>aaa</ns2:domicilio>
        <ns2:telefono>1111</ns2:telefono>
      </ns2:contratos>
    </ns2:ConsultarContratoResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- ModificarServicio: Retornara un mensaje de éxito, si el número de contrato coincide con la firma electrónica y que ninguno de estos dos parámetros vaya vacío.

```
POST https://microservicio-contrato.herokuapp.com:443/ws
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:ModificarServicioResponse xmlns:ns2="https://t4is.uv.mx/contratos">
      <ns2:respuesta>Tus datos han sido modificados exitosamente.</ns2:respuesta>
    </ns2:ModificarServicioResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

- MostrarContratos: Esta operación retornara el nombre de las personas, su número de contrato y firma electrónica de todas las personas que hayan realizado su contrato.

```

POST https://microservicio-contrato.herokuapp.com:443/ws
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:MostrarContratosResponse xmlns:ns2="https://t4is.uv.mx/contratos">
      <ns2:contratos>
        <ns2:nombre>ablo</ns2:nombre>
        <ns2:ncontrato>2</ns2:ncontrato>
        <ns2:firmae>aaa</ns2:firmae>
      </ns2:contratos>
      <ns2:contratos>
        <ns2:nombre>Bryan duran</ns2:nombre>
        <ns2:ncontrato>4</ns2:ncontrato>
        <ns2:firmae>2fdg</ns2:firmae>
      </ns2:contratos>
      <ns2:contratos>
        <ns2:nombre>Carlos Estrada Estrada</ns2:nombre>
        <ns2:ncontrato>5</ns2:ncontrato>
        <ns2:firmae>gggg4444</ns2:firmae>
      </ns2:contratos>
      <ns2:contratos>
        <ns2:nombre>pablo</ns2:nombre>
        <ns2:ncontrato>6</ns2:ncontrato>
        <ns2:firmae>bbb</ns2:firmae>
      </ns2:contratos>
      <ns2:contratos>
        <ns2:nombre>Enrique Lopez Fox</ns2:nombre>
        <ns2:ncontrato>7</ns2:ncontrato>
        <ns2:firmae>a15s25</ns2:firmae>
      </ns2:contratos>
    </ns2:MostrarContratosResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Plan de pruebas

El plan de prueba describe el ámbito del esfuerzo de prueba general y proporciona un registro del proceso de planificación de prueba.

SOAP

Para probar las 5 operaciones hechas en SOAP se usará la extensión para los navegadores [Wizdler](#). Wizdler es una herramienta, desarrollada con el fin de realizar pruebas a aplicaciones con arquitectura orientada a servicio y transferencia de estado representacional (SOAP y REST).



Y ya usando la extensión, se da click sobre las operaciones disponibles.

REST

Forma de ejecución de los contenedores

Repositorio en GitHub

El repositorio donde se alojará el proyecto será el siguiente:

<https://github.com/BryanDuran/CFE>

Documentación en los archivos Docker file y Docker-compose.yml

Para nuestro proyecto se optó por usar Docker files.

Microservicio 1

El microservicio 1 se encuentra disponible: <https://microservicio-contrato.herokuapp.com/ws/contratos.wsd/>

Explicación del Docker File,

1. Se inicializa la imagen base para las instrucciones subsecuentes en este caso se usará la imagen de JDK8.
2. Se establece el directorio de trabajo; /app.
3. CMD se encarga de pasar el comando al contenedor, en este caso ejecuta el archivo *ejecutar.sh*.
4. ADD se encarga de copiar el archivo JAR de la carpeta app local a la carpeta app de la imagen.
5. Lo mismo pasa con ejecutar, es copiada a la carpeta app de la imagen.

6. Finalmente ejecuta el archivo ejecutar.sh, dentro del archivo ejecutar.sh, se establece el puerto donde se ejecutara, debido a que expose no funciona al hacer el deploy con heroku.

El microservicio 1 contiene todas las operaciones del SOAP, previamente mencionadas.

```
1 From rrojano/jdk8
2 workdir /app
3 cmd ["/app/ejecutar.sh"]
4 add app/ContratosCFE-0.0.1-SNAPSHOT.jar /app/ContratosCFE-0.0.1-SNAPSHOT.jar
5 add ejecutar.sh /app/ejecutar.sh
6 run chmod 755 /app/ejecutar.sh
```

Ejecutar.sh

```
/usr/bin/java -jar -Dserver.port=$PORT ContratosCFE-0.0.1-SNAPSHOT.jar
```

Microservicio 2

El microservicio 2 se encuentra disponible: <https://microservicio-pago.herokuapp.com/>