



Universidad Veracruzana

- ❖ Juan Pablo Ortiz Páez
- ❖ Luis Enrique Zapata López
- ❖ Carlos Enrique Gonzales Arcocha



Proyecto Final

Servicio de administración de
préstamos escolares

TECNOLOGÍAS PARA LA
INTEGRACIÓN DE SOLUCIONES

JOSÉ RAFAEL ROJANO CÁCERES

6 -JUNIO - 2022

Índice

1.	Introducción.....	3
2.	Motivación	4
3.	Problemática.....	4
4.	Solución	5
5.	Costos	6
	Heroku.....	6
	Github	6
	Clever Cloud	6
6.	Diagrama de Despliegue	7
7.	Documentación préstamo de cubículos (SOAP)	8
7.1.	EndPoint.....	8
7.2.	Parámetros de recepción y forma de ejecución	8
	RegistrarPrestamo	8
	ReportarCubiculo.....	9
	VerCubiculo	9
	VerReportes	9
7.3.	Parámetros devueltos.....	10
	RegistrarPrestamo	10
	ReportarCubiculo.....	10
	VerCubiculo	11
	VerReportes	12
7.4.	Plan de pruebas	13
8.	Documentación préstamo de artículos (API REST)	14
8.1.	EndPoint.....	14
8.2.	Parámetros de recepción	14
8.3.	Parámetros devueltos	15
8.4.	Plan de prueba	18
9.	- Forma de ejecución de los contenedores.....	19
9.1.	Plan de prueba	21
10.	Liga de GitHub	22

1. Introducción

Como conclusión de la experiencia educativa “Tecnologías para la integración de soluciones” se busca evaluar y aplicar el contenido visto en clase, esto con la realización de un proyecto final, definiendo el proyecto, es requerido realizar un conjunto de servicios con el que se pueda realizar la administración de préstamos en la Facultad de estadística e informática (FEI). En el desarrollo del trabajo se emplearán las tecnologías SOAP y REST, el servicio no contará con la implementación de una interfaz gráfica ya que solo se trabajará sobre el backend, esto debido a que se busca tener un mayor énfasis sobre el uso de los servicios y micro-servicios para la integración de soluciones.

El planteamiento del proyecto busca tener un alcance de desarrollo con el que se resuelva una problemática en nuestro entorno como estudiantes, se indicara en el transcurso de este documento las motivaciones del tema escogido, la problemática que surge en su planteamiento y la solución que proporcionaremos. Además, se describe el planteamiento de los servicios mencionados en esta pequeña documentación, también se especifican los parámetros que el servicio maneja, plan de pruebas donde se demostrará el funcionamiento del sistema y la manera en que se deben ejecutar los contenedores.

2. Motivación

La motivación que sigue el desarrollo de este proyecto es apoyar a la facultad con el proceso de administración de préstamos de los cubículos de estudiantes ubicados en el nuevo edificio de la facultad, el préstamo de diferentes artículos (como controles de los proyectores que las aulas ocupan, balones de juegos, cámaras, proyectores y demás) que puedan ser prestados a docentes y estudiantes dentro de las instalaciones de la FEI, ya que el proceso de administración de préstamos que se usa actualmente se hace a mano y por escrito, o en el caso de los cubículos no se lleva a cabo un proceso de administración, y la implementación de este conjunto de servicios facilitaría dicha administración.

3. Problemática

La problemática reside principalmente en el préstamo de los cubículos de los cuales no existe un proceso encargado de administrar quienes hacen uso de los cubículos para estudiantes, actualmente un alumno o grupo de alumnos busca a su correspondiente jefe de carrera y realizan un préstamo del cubículo correspondiente, por lo que el jefe de carrera procede a usar la llave para abrirla, por lo que el proceso tiende a ser largo y poco eficaz, ya que se debe encontrar al jefe de carrera y si está ocupado se perdería tiempo por parte del jefe de carrera y los alumnos para esperar a que el jefe de carrera se libere para abrir el cubículo, pero la problemática principal dentro del préstamo de cubículos es que la mayoría de alumnos desconocen la posibilidad de que les permitan usarlos, por la experiencia al haber hecho uso de ellos se ha notado que solo se hace uso de 2 de los 4 cubículos disponibles y para algunos estudiantes les ayudaría mucho un espacio adecuado donde puedan trabajar o tomar sus clases en línea y demás actividades.

Por el lado del prestamos de los artículos, existe la problemática de que el proceso administrativo de los préstamos se hace a mano y por escrito, llevando un registro en una libreta, y aunque actualmente este método funciona existe la posibilidad de facilitar y agilizar tal proceso de préstamo.

4. Solución

El planteamiento de realizar un conjunto de servicios para la administración de los préstamos en la facultad es dividirlos en micro-servicios, un servicio para el préstamo de los cubículos estudiantiles y otro servicio para el préstamo de controles o utilería de los salones.

A continuación, se realiza una especificación de las funciones que hará cada servicio (todo esto estará detallado más adelante):

Servicio de préstamo de cubículos:

- Ver cubículo.
- Registrar préstamo.
- Realizar reporte.
- Ver reportes.

Servicio de préstamo de artículos:

- Ver artículos prestados.
- Realizar préstamo.
- Regresar artículo.

5. Costos

Heroku

Heroku es una plataforma como servicio (PaaS) de computación en la nube la cual inicialmente fue desarrollada para soportar aplicaciones en lenguaje Ruby pero posteriormente ha extendido su soporte a numerosos lenguajes y aplicaciones que ofrecen una mayor flexibilidad y agilidad a la hora de construir aplicaciones en WEB.

Github

Es una forja (plataforma de desarrollo colaborativo) para alojar proyectos utilizando el sistema de control de versiones Git. Se utiliza principalmente para la creación de código fuente de programas de ordenador. El software que opera GitHub fue escrito en Ruby on Rails.

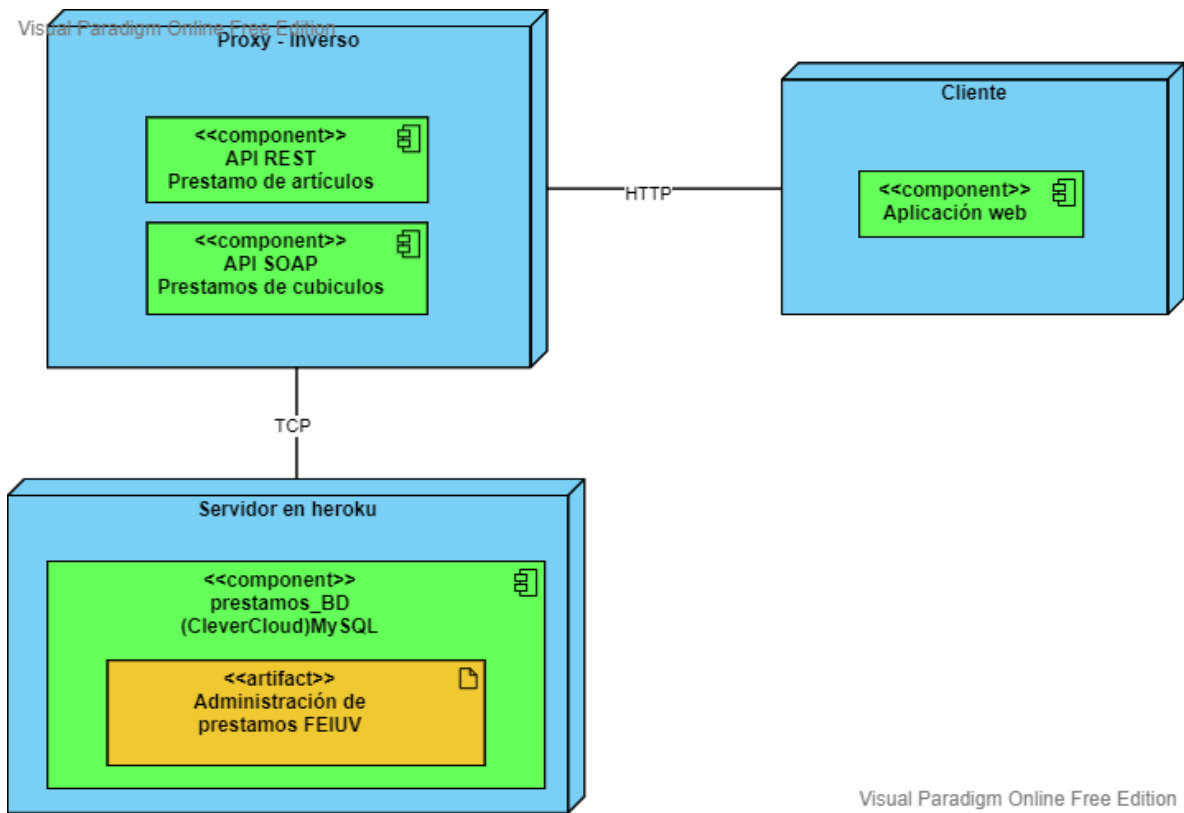
Clever Cloud

Clever Cloud es una plataforma de automatización de TI para administrar todas las operaciones mientras te enfocas en el valor de tu negocio.

Clever Cloud es un proveedor de soluciones de PaaS (plataforma como servicio, por sus siglas en inglés) optimizadas por el tiempo de ejecución de la tecnología (Java, Scala, Node.js, PHP y Docker si tienes necesidades especiales).

Clever Cloud también es un proveedor de soluciones SaaS dedicadas a proyectos de DevOps: instalaciones físicas para la integración y la implementación continuas (Jenkins, GitLab, GitHub Enterprise, Nexus, Rocket.Chat y más).

6. Diagrama de Despliegue



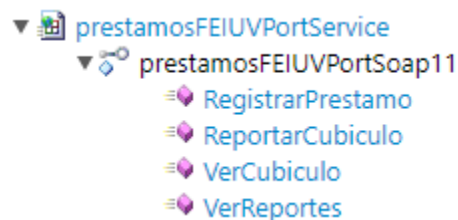
7. Documentación préstamo de cubículos (SOAP)

7.1. EndPoint

Dentro de este servicio se tienen 4 métodos para el control de los cubículos, cada uno de ellos cuenta con un método de recepción y envío de distintos tipos de datos, cada uno de ellos se explica más detalladamente a continuación:

7.2. Parámetros de recepción y forma de ejecución

Cada uno de los métodos del EndPoint recibe algún dato o valor correspondiente a lo que se quiere hacer, a continuación, se muestra el qué recibe y el porqué de ello:



RegistrarPrestamo

Registra un nuevo préstamo para un cubículo, este recibirá:

- **Matrícula** de quien solicitó el préstamo ("zS18014524").
- **Carrera** de quien solicitó el préstamo ("Tecnologías computacionales").
- El **tiempo** que se solicitó (especificado con formato de 24 hrs y agregado como un rango de horas, ejemplo: "14-15" que sería un préstamo de 02:00 pm a 03:00 pm).
- **Día de la semana** requerido ("Viernes").
- **Número del día** requerido ("20").
- **Mes** que quiere el préstamo efectivo ("Mayo").

Ejemplo del formato XML que recibe este método:

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <RegistrarPrestamoRequest xmlns="https://t4is.uv.mx/prestamosFEIUV">
      <matricula>zS18014524</matricula>
      <carrera>Tecnologías computacionales</carrera>
      <tiempo>14-16</tiempo>
      <diaS>Lunes</diaS>
      <diaN>15</diaN>
      <mes>Junio</mes>
    </RegistrarPrestamoRequest>
  </Body>
</Envelope>
```


ReportarCubiculo

Crearé un nuevo reporte, esto en caso de que algo dentro de los cubículos esté fuera de lo permitido, este recibirá:

- **Fecha** de cuando se encontró el cubículo en esa condición (2022-05-20 (AAAA-MM-DD)).
- **Hora** del momento que se encontró el cubículo en esa condición (12:39:29 (hh:mm:ss)).
- **Carrera** al que pertenece el cubículo ("Tecnologías computacionales").
- **Descripción** del estado en el que se encontró el cubículo (Dejaron sucia la mesa).

Ejemplo del formato XML que recibe este método:

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <ReportarCubiculoRequest xmlns="https://t4is.uv.mx/prestamosFEIUV">
      <fecha>2022-05-20</fecha>
      <hora>12:39:29</hora>
      <carrera>Tecnologías computacionales</carrera>
      <descripcion>Dejaron sucia la mesa</descripcion>
    </ReportarCubiculoRequest>
  </Body>
</Envelope>
```

VerCubiculo

Mostraré un listado de todos los préstamos del cubículo solicitado, este recibirá:

- **Carrera** al que pertenece el cubículo ("Tecnologías computacionales").

Ejemplo del formato XML que recibe este método:

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <VerCubiculoRequest xmlns="https://t4is.uv.mx/prestamosFEIUV">
      <nomCubiculo>Tecnologías computacionales</nomCubiculo>
    </VerCubiculoRequest>
  </Body>
</Envelope>
```

VerReportes

Mostraré un listado de todos los reportes del cubículo solicitado, este recibirá:

- **Carrera** al que pertenece el cubículo ("Tecnologías computacionales").

Ejemplo del formato XML que recibe este método:

```
<Envelope xmlns="http://schemas.xmlsoap.org/soap/envelope/">
  <Body>
    <VerReportesRequest xmlns="https://t4is.uv.mx/prestamosFEIUV">
      <cubicNom>Tecnologías computacionales</cubicNom>
    </VerReportesRequest>
  </Body>
</Envelope>
```

7.3. Parámetros devueltos

RegistrarPrestamo

Devuelve:

- Mensaje de confirmación

Ejemplo del formato XML que recibe este método:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:RegistrarPrestamoResponse xmlns:ns2="https://t4is.uv.mx/prestamosFEIUV">
      <ns2:confirmacion>Se registro el prestamo a la matrícula zS18014531 para el cubículo de
Tecnologías computacionales</ns2:confirmacion>
    </ns2:RegistrarPrestamoResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

ReportarCubiculo

Devuelve:

- Mensaje de confirmación

Ejemplo del formato XML que recibe este método:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:ReportarCubiculoResponse xmlns:ns2="https://t4is.uv.mx/prestamosFEIUV">
      <ns2:verificacion>Se realizó el reporte al cubiculo de Tecnologías
computacionales</ns2:verificacion>
    </ns2:ReportarCubiculoResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

VerCubiculo

Devuelve:

- Mostrará un listado de todos los préstamos del cubículo solicitado

Ejemplo del formato XML que recibe este método:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:VerCubiculoResponse xmlns:ns2="https://t4is.uv.mx/prestamosFEIUV">
      <ns2:horario>
        <ns2:matricula>zS18014524</ns2:matricula>
        <ns2:carrera>Tecnologías computacionales</ns2:carrera>
        <ns2:tiempo>12-13</ns2:tiempo>
        <ns2:diaS>Lunes</ns2:diaS>
        <ns2:diaN>15</ns2:diaN>
        <ns2:mes>Mayo</ns2:mes>
      </ns2:horario>
      <ns2:horario>
        <ns2:matricula>zS18014525</ns2:matricula>
        <ns2:carrera>Tecnologías computacionales</ns2:carrera>
        <ns2:tiempo>12-13</ns2:tiempo>
        <ns2:diaS>Lunes</ns2:diaS>
        <ns2:diaN>15</ns2:diaN>
        <ns2:mes>Mayo</ns2:mes>
      </ns2:horario>
      <ns2:horario>
        <ns2:matricula>zS18014527</ns2:matricula>
        <ns2:carrera>Tecnologías computacionales</ns2:carrera>
        <ns2:tiempo>15-21</ns2:tiempo>
        <ns2:diaS>Jueves</ns2:diaS>
        <ns2:diaN>25</ns2:diaN>
        <ns2:mes>Abril</ns2:mes>
      </ns2:horario>
      <ns2:horario>
        <ns2:matricula>zS18014531</ns2:matricula>
        <ns2:carrera>Tecnologías computacionales</ns2:carrera>
        <ns2:tiempo>14-16</ns2:tiempo>
        <ns2:diaS>Lunes</ns2:diaS>
        <ns2:diaN>15</ns2:diaN>
        <ns2:mes>Junio</ns2:mes>
      </ns2:horario>
    </ns2:VerCubiculoResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

VerReportes

Devuelve:

- Mostrará un listado de todos los reportes del cubículo solicitado

Ejemplo del formato XML que recibe este método:

```
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header/>
  <SOAP-ENV:Body>
    <ns2:VerReportesResponse xmlns:ns2="https://t4is.uv.mx/prestamosFEIUV">
      <ns2:listaReportes>
        <ns2:fecha>2022-07-24</ns2:fecha>
        <ns2:hora>23:59:59</ns2:hora>
        <ns2:carrera>Tecnologías computacionales</ns2:carrera>
        <ns2:descripcion>Pablo se robó un gato y lo está criando en el
cubículo</ns2:descripcion>
      </ns2:listaReportes>
      <ns2:listaReportes>
        <ns2:fecha>2022-05-20</ns2:fecha>
        <ns2:hora>12:19:29</ns2:hora>
        <ns2:carrera>Tecnologías computacionales</ns2:carrera>
        <ns2:descripcion>Explotó el frigobar</ns2:descripcion>
      </ns2:listaReportes>
      <ns2:listaReportes>
        <ns2:fecha>2022-05-20</ns2:fecha>
        <ns2:hora>12:39:29</ns2:hora>
        <ns2:carrera>Tecnologías computacionales</ns2:carrera>
        <ns2:descripcion>Consumo de luz excesivo se detubo de golpe</ns2:descripcion>
      </ns2:listaReportes>
    </ns2:VerReportesResponse>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

7.4. Plan de pruebas

Parámetro	Enlace	Entrada	Salida	Condición	Observación
registrarPrestamo	http://prestamos-feiuv.herokuapp.com/ws/prestamosFEIUV.wsdl	<pre><matricula>zS18053103</matricula> <carrera>Estadística</carrera> <tiempo>7-12</tiempo> <diaS>Martes</diaS> <diaN>24</diaN> <mes>Mayo</mes></pre>	<pre><ns2:confirmacion>Se registró el préstamo a la matrícula zS18053103 para el cubículo de Estadística</ns2:confirmacion></pre>	Ninguna	Funciona con normalidad
VerCubiculos	http://prestamos-feiuv.herokuapp.com/ws/prestamosFEIUV.wsdl	<pre><nomCubiculo>Estadística </nomCubiculo></pre>	<pre><ns2:horario> <ns2:matricula>zS18053103</ns2:matr icula> <ns2:carrera>Estadística</ns2:carrera> <ns2:tiempo>7-12</ns2:tiempo> <ns2:diaS>Martes</ns2:diaS> <ns2:diaN>24</ns2:diaN> <ns2:mes>Mayo</ns2:mes> </ns2:horario></pre>	Escribir la carrera justo como se insertó (Mayúsculas y minúsculas)	Funciona con normalidad
ReportarCubiculo	http://prestamos-feiuv.herokuapp.com/ws/prestamosFEIUV.wsdl	<pre><fecha>2022-05-20</fecha> <hora>12:39:29</hora> <carrera>Tecnologías computacionales</carrera> <descripcion>Dejaron sucia la mesa</descripcion></pre>	<pre><ns2:verificacion>Se realizó el reporte al cubiculo de Tecnologías computacionales</ns2:verificacion></pre>	Ninguna	Funciona con normalidad
verReportes	http://prestamos-feiuv.herokuapp.com/ws/prestamosFEIUV.wsdl	<pre><cubicNom>Tecnologías computacionales</cubicNom></pre>	<pre><ns2:listaReportes> <ns2:fecha>2022-07-24</ns2:fecha> <ns2:hora>23:59:59</ns2:hora> <ns2:carrera>Tecnologías computacionales</ns2:carrera> <ns2:descripcion>Pablo se robó un gato y lo está criando en el cubículo</ns2:descripcion> </ns2:listaReportes></pre>	Escribir la carrera justo como se insertó (Mayúsculas y minúsculas)	Funciona con normalidad

8. Documentación préstamo de artículos (API REST)

8.1. EndPoint

Dentro de este servicio se tienen 4 métodos para el control de los prestamos dentro la Facultad de estadística e informática, cada uno de ellos cuenta con un método de recepción y envío de distintos tipos de datos, cada uno de ellos se explica más detalladamente a continuación:

8.2. Parámetros de recepción

VerInventario | Verbo necesario: **GET**

Muestra el Inventario de los artículos, este muestra:

- **Id** del artículo que se solicita ver ("1")
- **Nombre** del artículo que se solita ver ("Balón de Futbol")
- **Tipo** de artículo que se quiere ver ("Deporte")
- **Lugar** donde se encuentra el artículo que se quiere ver ("Administración")

Nota: No recibe ningún tipo de dato.

GuardarArticulos | Verbo necesario: **POST**

Guarda un nuevo artículo para el inventario, este recibe:

- **Id** del artículo que se desea guardar ("2")
- **Nombre** del artículo que se guarda ("Mesa de Ping Pong")
- **Tipo** de artículo que se guarda ("Deporte")
- **Lugar** en el que se pondrá el articulo a guardar ("Administración")

Ejemplo de parámetros recibidos en formato JSON:

```
{
  "nombre": "Proyector",
  "tipo": "Audiovisual",
  "lugar": "Centro de cómputo"
}
```

VerPrestamos | Verbo necesario: **GET**

Muestra los Prestamos que se han realizado, muestra:

- **Id** del objeto prestado ("1")
- **Articulo prestado** nombre del artículo que se presta ("Balón de Futbol")
- **Prestatario** nombre de la persona a que se le presto el artículo ("S18014503")
- **Lugar de préstamo** lugar donde se hizo el préstamo ("Administración")
- **Hora del préstamo** la hora cuando se prestó el artículo ("12:40")

Nota: No recibe ningún tipo de dato.

GuardarPrestamo | Verbo necesario: **POST**

Guarda un nuevo préstamo, este recibe:

- **Id** (Este se genera automáticamente, no es necesario introducirlo)
- **Artículo prestado** nombre del artículo a guardar ("Balón de Futbol")
- **Prestatario** nombre de la persona a que se le prestara el artículo ("S18014503")
- **Lugar de préstamo** lugar donde se presta el artículo ("Administración")
- **Hora del préstamo** hora en que se presta el artículo ("12:40")

Ejemplo de parámetros recibidos en formato JSON:

```
{
  "articuloPrestado":"Proyector",
  "prestatario":"zS18015986",
  "lugarP":"Aula TC",
  "horaPrestamo":"11:02"
}
```

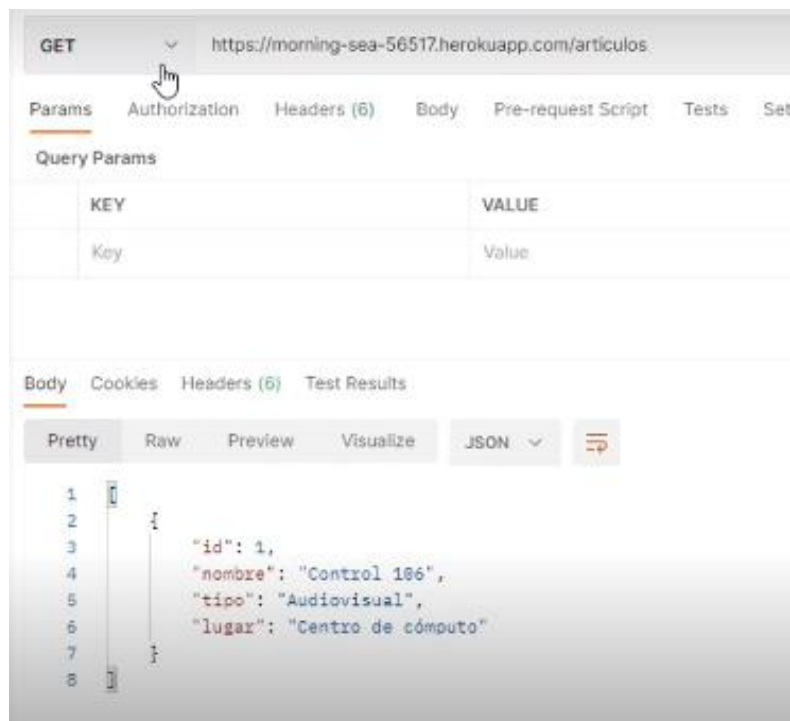
8.3. Parámetros devueltos

VerInventario | Verbo necesario: **GET**

Esta muestra una lista de todos los artículos del inventario.

Ejemplo:

<https://morning-sea-56517.herokuapp.com/articulos>

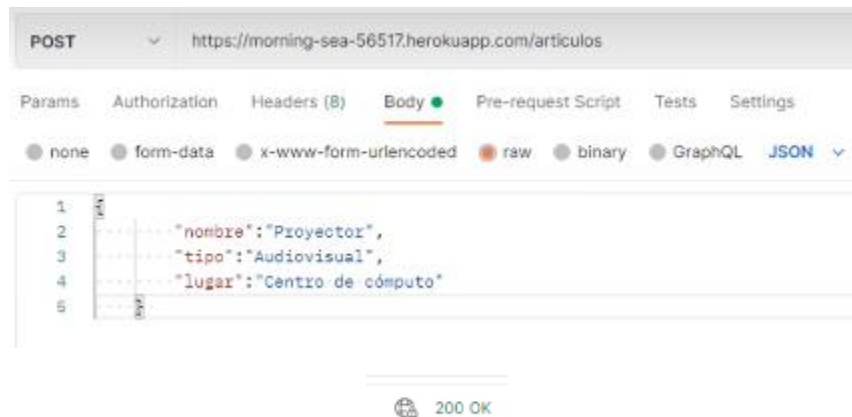


GuardarArticulo | Verbo necesario: **POST**

Guarda un nuevo artículo en la base de datos, con la confirmación del código de respuesta 200 por parte del servidor (este demuestra que funciona correctamente).

Ejemplo:

<https://morning-sea-56517.herokuapp.com/articulos>



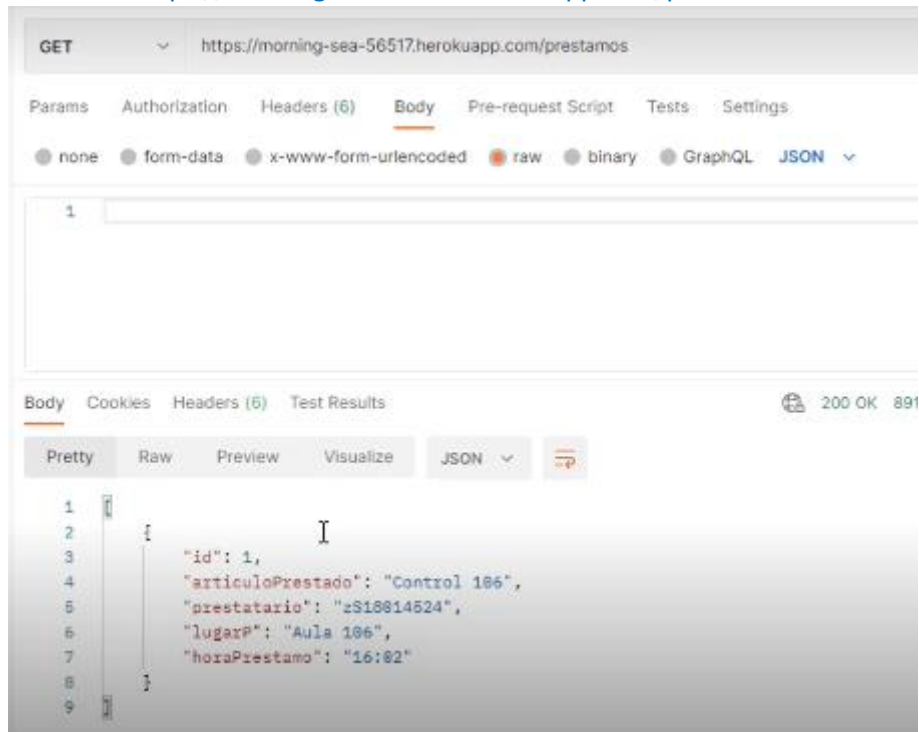
```
MySQL [bt6ynw1qhk1jckogkjqx]> select * from inventario;
+----+-----+-----+-----+
| id | lugar          | nombre      | tipo      |
+----+-----+-----+-----+
|  1 | Centro de cómputo | Control 106 | Audiovisual |
|  2 | Centro de cómputo | Proyector   | Audiovisual |
+----+-----+-----+-----+
2 rows in set (0.164 sec)
```


VerPrestamo | Verbo necesario: GET

Muestra una lista de los préstamos realizados.

Ejemplo:

<https://morning-sea-56517.herokuapp.com/prestamos>

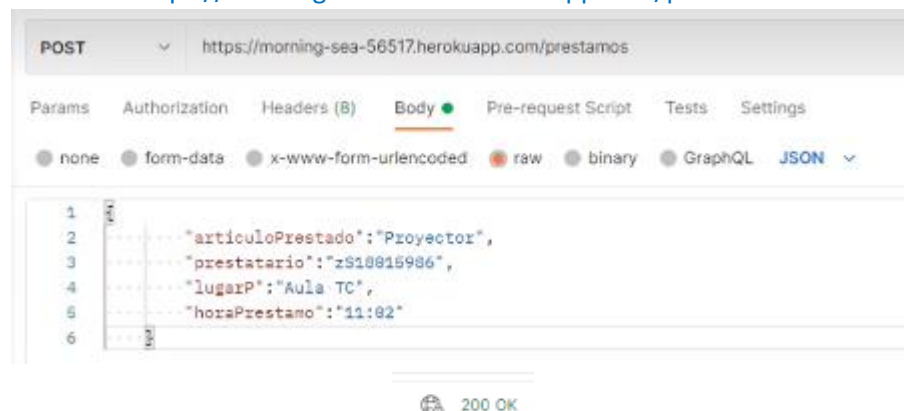


GuardarPrestamo | Verbo necesario: POST

Ingresa nuevos préstamos con la confirmación del código de respuesta 200.

Ejemplo:

<https://morning-sea-56517.herokuapp.com/prestamos>



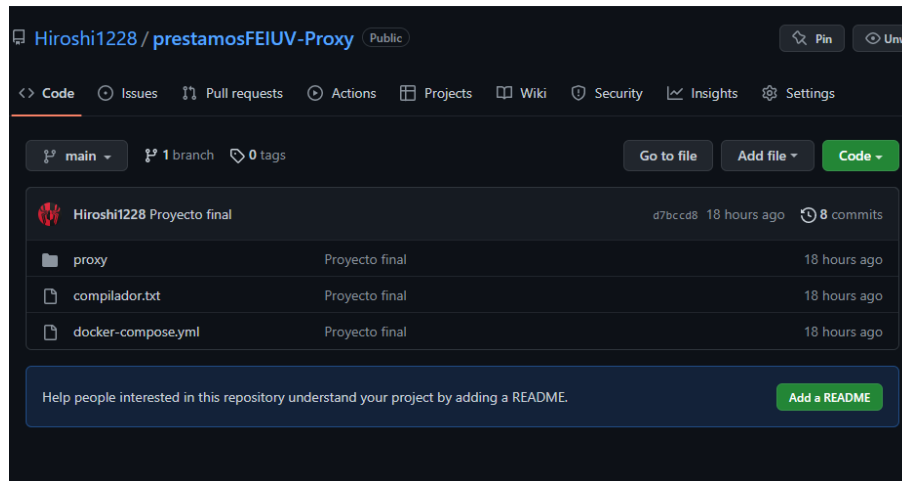
```
MySQL [bt6ynw1qhk1jckogkjx]> select * from prestamos;
+----+-----+-----+-----+-----+
| id | articulo_prestado | hora_prestamo | lugarp | prestatario |
+----+-----+-----+-----+-----+
| 1  | Control 106      | 16:02        | Aula 106 | zS18814524 |
| 2  | Proyector        | 11:02        | Aula TC  | zS18815986 |
+----+-----+-----+-----+-----+
2 rows in set (0.157 sec)
```

8.4. Plan de prueba

Parámetro	Enlace	Entrada	Salida	Condición	Observaciones
VerInventario	https://morning-sea-56517.herokuapp.com/articulos	-	{ "id" : 1, "nombre" : "Control 136", "tipo" : "Audiovisual", "lugar": "Centro de computo" }	Ninguna	Funciona con normalidad
GuardarArticulos	https://morning-sea-56517.herokuapp.com/articulos	{ "nombre":"Proyector", "tipo":"Audiovisual", "lugar":"Centro de cómputo" }	Se guarda el artículo en la base de datos	El artículo no debe tener la misma Id	Funciona con normalidad
VerPrestamo	https://morning-sea-56517.herokuapp.com/prestamos	-	{ "id" : 1, "articuloPrestado" : "Control 136", "Prestatario" : "S18014503", "lugarP": "Centro de computo" "HoraPrestamo" : "12:40" }	Ninguna	Funciona con normalidad
GuardarPrestamo	https://morning-sea-56517.herokuapp.com/prestamos	{ "articuloPrestado":"Proyector", "prestatario":"zS18015986", "lugarP":"Aula TC", "horaPrestamo":"11:02" }	Se guarda el préstamo en la base de datos	El "id" del artículo no debe repetirse	Funciona con normalidad

9. - Forma de ejecución de los contenedores

Para el funcionamiento del proxy inverso nosotros usamos la herramienta “Play with Docker” para poder compilarlo. Para esto lo primero que hicimos fue guardar nuestros archivos en un repositorio alojado en Github.

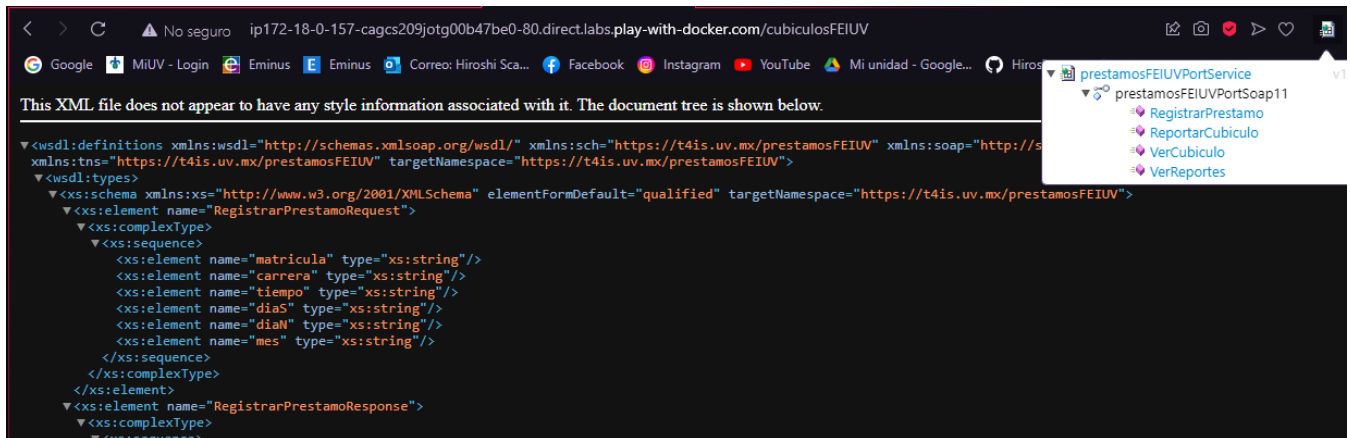


Después en el docker clonamos el repositorio e ingresamos a la carpeta del proxy y con el comando “docker-compose up” podremos compilar nuestro proxy inverso.

A screenshot of the 'Play with Docker' interface. On the left, there's a sidebar with a digital clock showing '03:54:37', a 'CLOSE SESSION' button, and an 'Instances' section with a '+ ADD NEW INSTANCE' button and a list of instances including '192.168.0.13 node1'. The main area shows the IP '192.168.0.13' and an 'OPEN PORT' button. Below this, there are tabs for 'Memory' and 'CPU', and an 'SSH' section with the command 'ssh ip172-18-0-157-cagcs209jotg00b47be0@direct.labs.pla'. At the bottom, there's a terminal window with the following commands and output:

```
#####
[node1] (local) root@192.168.0.13 ~
$ git clone https://github.com/Hiroshi1228/prestamosFEIUV-Proxy.git
Cloning into 'prestamosFEIUV-Proxy'...
remote: Enumerating objects: 36, done.
remote: Counting objects: 100% (36/36), done.
remote: Compressing objects: 100% (27/27), done.
remote: Total 36 (delta 13), reused 31 (delta 8), pack-reused 0
Receiving objects: 100% (36/36), done.
Resolving deltas: 100% (13/13), done.
[node1] (local) root@192.168.0.13 ~
$ cd prestamosFEIUV-Proxy
[node1] (local) root@192.168.0.13 ~/prestamosFEIUV-Proxy
$ docker-compose up
Creating network "prestamosfeiu-proxy_default" with the default driver
Building proxy-service
Step 1/2 : from nginx
latest: Pulling from library/nginx
42c077c10790: Pull complete
```

Y al abrir el puerto 80 podremos acceder a los micro servicios que configuramos.



9.1. Plan de prueba

Parámetro	Enlace	Entrada	Salida	Condición	Observaciones
inventarioFEIUV	http://morning-sea-56517.herokuapp.com/articulos	/inventarioFEIUV	[{"id":1,"nombre":"Control 106","tipo":"Audiovisual","lugar":"Centro de cómputo"}, {"id":2,"nombre":"Proyector","tipo":"Audiovisual","lugar":"Centro de cómputo"}]	Ninguna	Funciona con normalidad
prestamosFEIUV	http://morning-sea-56517.herokuapp.com/prestamos	/prestamosFEIUV	[{"id":1,"articuloPrestado":"Control 106","prestatario":"zS18014524","lugarP":"Aula106","horaPrestamo":"16:02"}, {"id":2,"articuloPrestado":"Proyector","prestatario":"zS18015986","lugarP":"Aula TC","horaPrestamo":"11:02"}]	Ninguna	Funciona con normalidad
cubiculosFEIUV	http://prestamos-feiuv.herokuapp.com/ws/prestamosFEIUV.wsdl	/cubiculosFEIUV	<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/" xmlns:sch="https://t4is.uv.mx/prestamosFEIUV" xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/" xmlns:tns="https://t4is.uv.mx/prestamosFEIUV" targetNamespace="https://t4is.uv.mx/prestamosFEIUV"> </wsdl:definitions>	Ninguna	Funciona con normalidad

10. Liga de GitHub

Carpetas con los micro servicios, proxy inverso y documentación.

<https://github.com/Hiroshi1228/Proyecto-T4IS.git>

Repositorio de servicio SOAP.

<https://github.com/Hiroshi1228/prestamosFEIUV>

Repositorio de servicio REST.

<https://github.com/Hiroshi1228/FEIUVPrestamos.git>

Repositorio de proxy inverso.

<https://github.com/Hiroshi1228/prestamosFEIUV-Proxy.git>