

Informe avances reingeniería

Análisis del código fuente del SistemaZ

Grupo: **G06**

Integrantes:

Nombre	Padrón
<i>Pedro Flynn</i>	<i>105742</i>
<i>Agustina Schmidt</i>	<i>103409</i>
<i>Agustina Fraccaro</i>	<i>103199</i>
<i>Kevin Gadacz</i>	<i>104531</i>
<i>Abraham Osco</i>	<i>102256</i>
<i>Ricardo Luizaga</i>	<i>87528</i>

Obtención del código fuente compilado:

Para obtener el código de la máquina virtual CentOS, primero localizamos dónde están los JAR:

- 1) Nos situamos en la raíz usando **cd ..**
- 2) Buscamos SistemaZ con **find . -name "*Sistema***
- 3) Nos dirigimos a **/var/lib/tomcat5/webapps/** y encontramos los archivos **SistemaZ.war** y **WS-expendiente.war**.
- 4) Luego usamos netcat para poder transferir los archivos, seguiremos la sgt lógica:

En la máquina que va recibir los archivos hacemos por ejemplo:

- a) **nc -l -p 12345 > SistemaZ.war.** (Creamos un server que escucha en el puerto 12345 y la rstp la redirecciona a **SistemaZ.war**).

En la máquina que enviará los archivos hacemos:

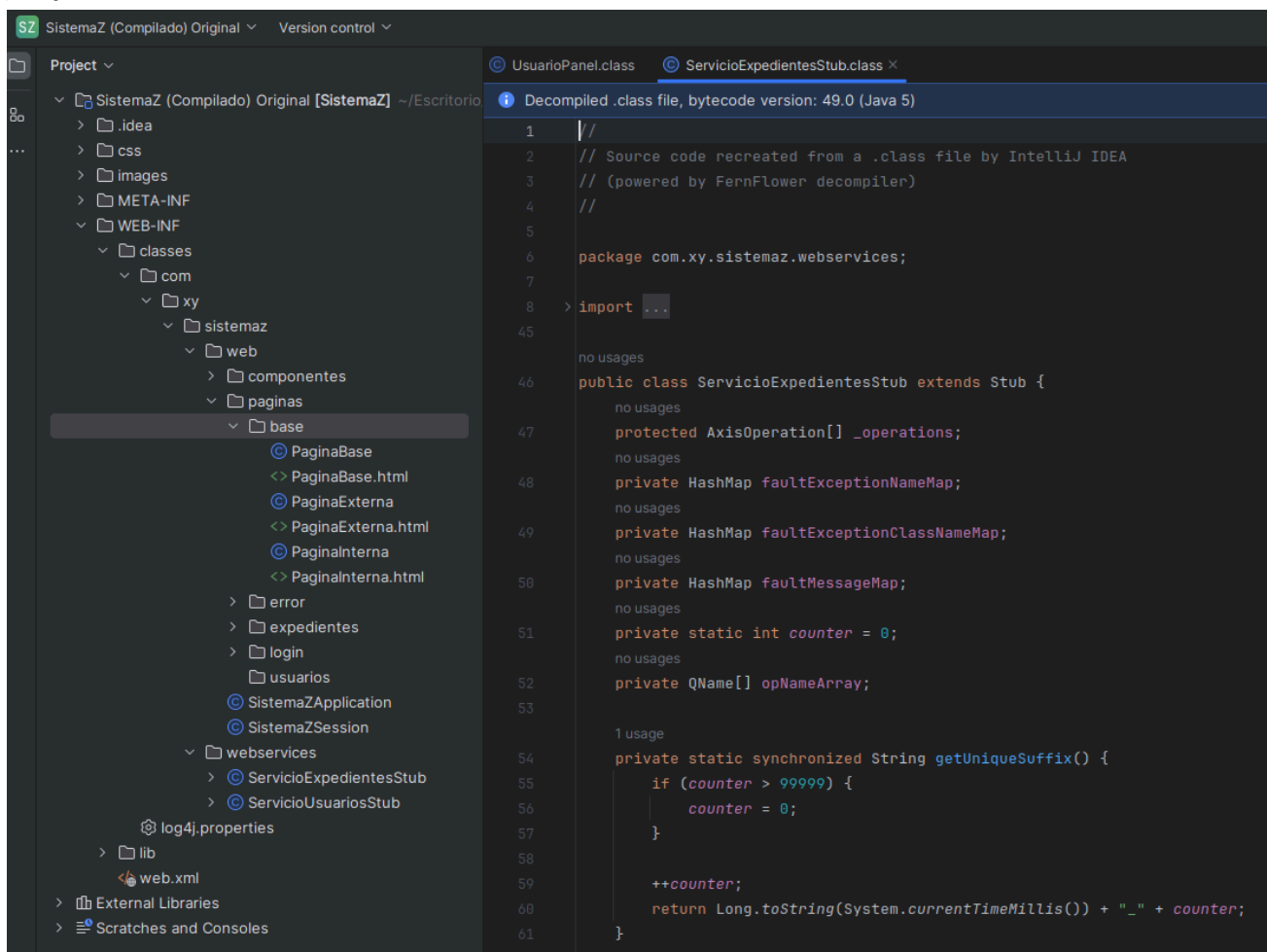
- b) **sudo nc 172.20.0.70 12345 < /var/lib/tomcat5/webapps/SistemaZ.war**

Repetimos este razonamiento con el archivo **WS-expendiente.war**

- 5) Para encontrar la base de datos hacemos: **find . -name "*.sql"** y lo enviamos usando netcat.

Descompilación del código fuente:

Una vez que obtuvimos el código fuente usando **IntelliJ Idea** podemos abrir el proyecto:

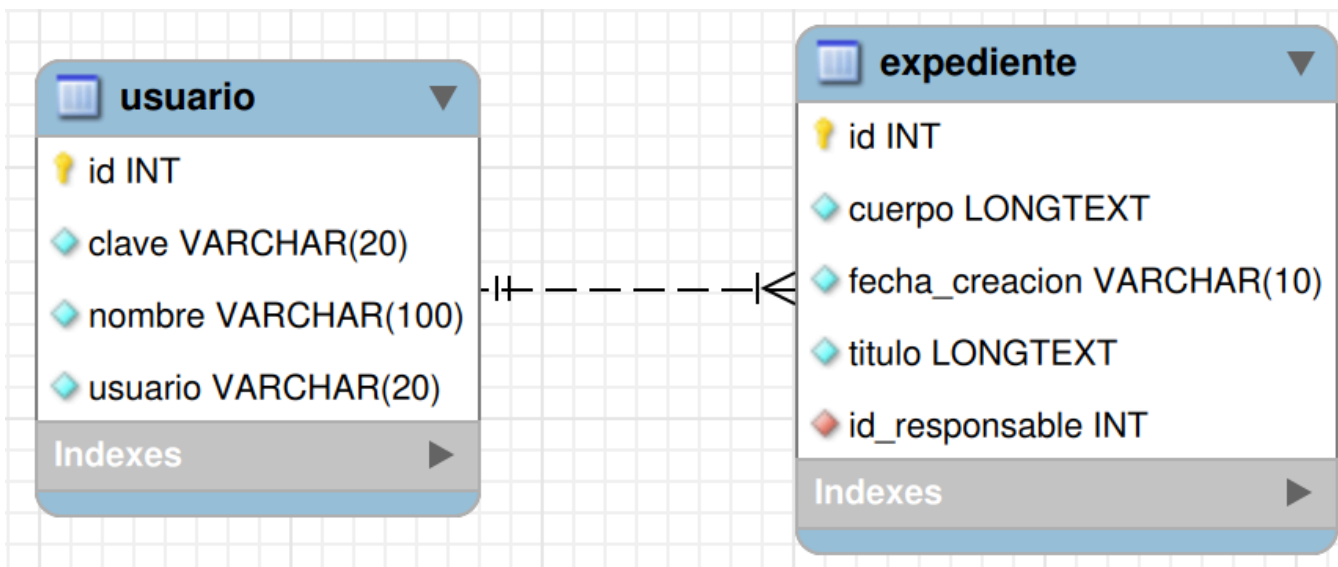


Podemos visualizar el código ya descompilado que nos autogenera el IDE **intellij idea**. Entonces para obtener los .java hicimos el sgt procedimiento:

- 1) Luego seleccionamos un archivo y crear una clase con el mismo nombre pero que finaliza en un .java, pegando aquí el código descompilado.
- 2) Repetimos esta lógica para cada clase, de esta manera obtuvimos el código descompilado.

Base de datos de SistemaZ:

- 1) A priori tenemos esta base de datos que recibimos inicialmente.
- 2) Debido a que ahora incluiremos mayor seguridad se podrían crear nuevas tablas, nuevos atributos e incluso modificar atributos actuales por ejemplo para la clave al encriptar con SHA-512 o SHA-256.
- 3) Del diagrama actualmente podemos leer que:
 - a) Un usuario puede tener muchos expedientes.
 - b) Un expediente corresponde a uno y solo un usuario.



Endpoints :

1) POST Login: axis2-admin/login:

a) Básicamente es para que el usuario se logee.

The image shows the login interface of 'Sistema Z - Expediente' on the left. It has a title bar, a header with the system name, and a login section titled 'Ingreso'. The login section contains two input fields: 'Usuario' with the value 'pedro' and 'Clave' with masked characters '.....'. Below these is an 'Entrar' button. On the right, a browser's developer tools network tab is open, showing a POST request to 'http://172.20.0.140:8080/SistemaZ/login?wicket:interface:NotActive=true&random=0.8515072984353382'. The status is '200 OK'.

Name	Headers	Payload	Preview	Response	Initiator	Timing	Cookies
lo...	General						
Request URL:		http://172.20.0.140:8080/SistemaZ/login?wicket:interface:NotActive=true&random=0.8515072984353382					
Request Method:		POST					
Status Code:		200 OK					
Remote Address:		172.20.0.140:8080					
Referrer Policy:		strict-origin-when-cross-origin					

2) Logout (al presionar sobre el logout en el sistema):


The image displays two screenshots of network requests in a browser's developer tools. The top screenshot shows a GET request to 'http://192.168.1.47:8080/SistemaZ/logout' with a status of '302 Moved Temporarily'. The bottom screenshot shows a GET request to 'http://192.168.1.47:8080/SistemaZ/login' with a status of '200 OK'.

Request URL	Request Method	Status Code	Remote Address	Referrer Policy
http://192.168.1.47:8080/SistemaZ/logout	GET	302 Moved Temporarily	192.168.1.47:8080	strict-origin-when-cross-origin
http://192.168.1.47:8080/SistemaZ/login	GET	200 OK	192.168.1.47:8080	strict-origin-when-cross-origin

3) POST busqueda expediente: Axis2upload

a) Buscamos el expediente usando por ej: **EXP-12**

The screenshot shows a web application interface on the left and a network packet capture tool (Wireshark) on the right. The web application has a search bar with the text 'EXP-12' and a 'Buscar' button. The network tool shows a list of packets, with the selected packet being an HTTP 200 OK response from 172.20.0.140:8080. The response headers are visible, showing a 200 OK status and a strict-origin-when-cross-origin referrer policy.



Sistema Z - Expedientes

Buscar expediente

Registrar expediente

¡Puede buscar expedientes por su código, por ejemplo *EXP-12*!

Código de expediente

EXP-01

Buscar

Número	EXP-1
Fecha creación	28-09-2010
Responsable	Pedro Gomez
Título	Resolución Provincia de Mendoza sobre caso Fernandez
Cuerpo	Se resuelve enviar la información pedida a la Camara de la Provincia de Mendoza

4) POST registrar expediente:

a) Registramos un nuevo expediente con título y cuerpo.

The screenshot shows a web application with a form titled "Registrar expediente". The form has two main sections: "Título*" and "Cuerpo*". The "Título*" field contains the text "nuevo expediente". The "Cuerpo*" field contains the text "un nuevo expendient". Below the form is a "Registrar" button. To the right of the form, a green text label reads "expediente. El codigo asignado es null".

Below the form, the browser's developer tools are open, showing the "Overview" tab. The "registro_e..." request is selected. The "General" section shows the following details:

- Request URL: http://172.20.0.140:8080/SistemaZ/registro_expediente?wicketignoreifNotActive=true&random=0.401075928815853
- Request Method: POST
- Status Code: 200 OK
- Remote Address: 172.20.0.140:8080
- Referrer Policy: strict-origin-when-cross-origin

The "Response Headers" section is also visible, showing a "Raw" view.



Sistema Z - Expedientes

Buscar expediente

Registrar expediente

Título*

Cuerpo*

Registrar

Se ha registrado el expediente. El código asignado es EXP-15

Cuentas:

(1,'clave01','Juan Perez','juanperez'),
(2,'clave','Sebastian Gonzalez','sgonzalez'),
(3,'pedro','Pedro Gomez','pedro')

Nuevo diseño:

Se les cambio la contraseña a los usuarios ahora es su contraseña anterior añadida un: **XYZ@**. Tenerlo en cuenta para volver a entrar.

Cuenta pedro:

username: **pedro**

password: **pedroXYZ@**

Definición de nuevos Endpoints usando REST:

1) Login:

- a) Método: **POST**
- b) Ruta: **auth/login**
- c) En el header:
 - i) authorization: Basic (username:password in base64).

2) Search file:

- a) Método: **GET**
- b) Ruta: **files/:id**

3) Create A file:

- a) Método: **POST**
- b) Ruta: **files**
- c) Body:

```
{  
  "title": "another title PROBANDO",  
}
```

```

"body": "example of body 1123 PRUEBA "
}

```

4) Refresh token:

a) Método: GET

b) Ruta: auth/refresh

c) header: en el header se ingresará el refresh Token (si está vencido te debería redirigir a login).

i) authorization: Bearer

```

eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJsInVzZXJOYXl1IjoicGVkcm8iLCJpYXQiOiE3MzAwOTA1NDksImV4cCI6MTczMDA5MDY2OX0.OwsiW0E-0hgpQJlvz0n3Z5VPqcRpNzkZaphI2AYMaPA

```

d) Caso de éxito:

```

{
  "accessToken":
  "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiJsInVzZXJOYXl1IjoicGVkcm8iLCJpYXQiOiE3MzAwOTcwOTYsImV4cCI6MTczMDA5MTE1Nn0.sv412Sj8sQve_L0ebRrczUhmNCv87Q3LMwBG_b-pDu0"
}

```

e) caso de no éxito: Redirigir a login

```

{
  "message": "Unauthorized",
  "statusCode": 401
}

```

Nueva Arquitectura:

