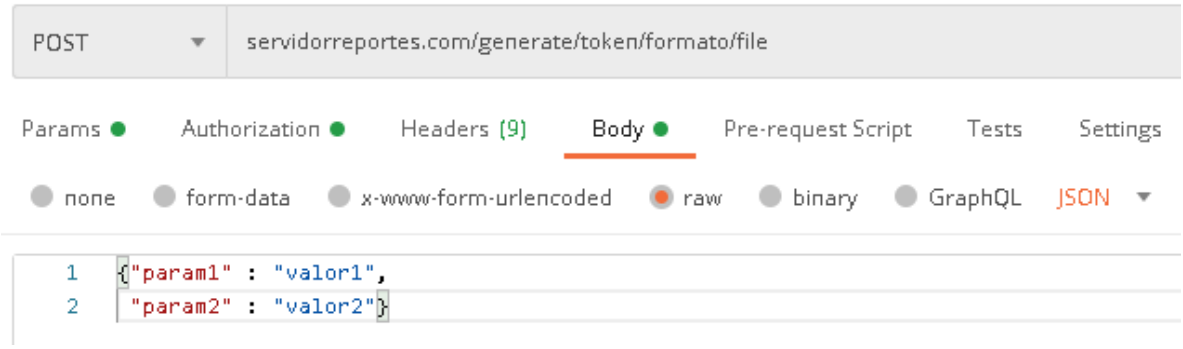
Sistema de Información para la Gestión de Reportes JasperReports

La extensión .JASPER se utiliza para los archivos que se crean con JasperReports . Estos archivos son también conocidos como JasperReports. Estos archivos contienen el contenido de un informe guardado en un formato binario y compilado a partir de un archivo .jrxml. Cuando se pretende realizar reportes en las aplicaciones se utilizan estos archivos JASPER sobre una implementación en Java, a través de la cual se envía una fuente especifica de datos (MySQL, PostgreSQL, etc). El reporte realiza la lectura de la fuente de datos y ejecuta las consultas definidas para generar reportes en archivos TXT ,HTML , PDF ,XLS , CSV o XML .

Este tipo de tecnología es fácil de implementar sobre aplicaciones cuyo nucleo es Java, como desarrollos con Swing, JSF, Java FX, Spring y otras más, pero para desarrollos en tecnologías como PHP se recurre a la creación de Bridges que permitan a través de máquinas virtuales poder generar estos reportes.

Se ha implementado un servicio REST que permite realizar la generación de reportes a través de peticiones, donde se define cierta dirección con los parámetros y se genera el archivo en PDF o XLS, que son los dos formatos soportados inicialmente. A continuación, se muestra la petición realizada al servidor utilizando POSTMAN



En la petición se pueden observar los atributos

- token: corresponde al token que identifica la conexión de la base de datos en el sistema.
- formato: define el formato en el cual se desea exportar el reporte PDF o XLS.
- file: corresponde al archivo del reporte a generar.

En este servicio los usuarios se pueden registrar gratuitamente y deben proceder a activar su cuenta a través de un correo electrónico. En este registro gratuito los usuarios quedan automáticamente con el rol Usuario e inicia en estado 0 o inactivos.

Después de que un usuario se encuentra registrado puede acceder al portal con sus credenciales, donde podrá ver información de los token de conexión configurados, así como poder editar y gestionar un nuevo token con la información necesaria en la tabla. Al registrar un nuevo token se debe asociar al usuario y se debe poder validar los datos de conexión. Es decir debe contar con un botón que permita realizar el testeo de estos datos. El campo state de la tabla connectiontoken define los siguientes estados:

- 0: creado sin validar
- 1: conexión validada
- 2: sin conexión

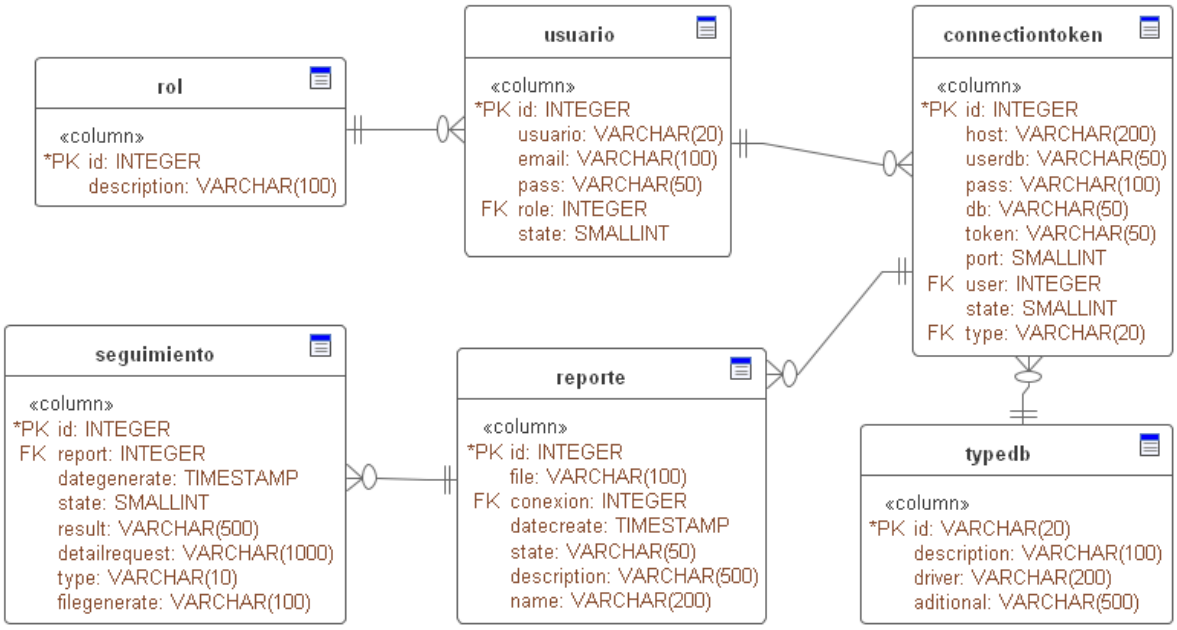
Para validar la conexión en la tabla typedb se tienen algunos datos de conexión, como el driver y el protocolo, el resto de datos se tienen en la tabla connectiontoken. De esta forma se crea la cadena de conexión:

```
String cadena = "jdbc:" + typedb.getId() + "://" + host + ":"+port+"/" + db + (typedb.getAditional()!=null?typedb.getAditional():"");
```

A esta conexión se deben asociar los distintos reportes que se van a cargar, para esto se debe realizar una interfaz donde el usuario pueda registrar el reporte seleccionando los tokens de conexión que tiene configurados su usuario.

Hay un rol adicional al rol de Usuario, el cual es el rol Administrador. En caso de ingresar como administrador, debes poder ver la relación de todos los usuarios, con el número de tokens y el número de reportes.

A continuación, se observa el modelo de datos que soporta este sistema de información.



A continuación, se observan el listado de requerimientos disponibles para el ejercicio:

- R1 - Registro de usuarios: se debe contar con un formulario que permita realizar el registro de usuarios. El registro de un usuario debe tener el rol Usuario. Inician con estado 0 y no se podrán loguear.

- R2 - Envío de correo de activación de usuarios: al momento del registro de usuarios se debe enviar un correo con un enlace de activación.
- R3 - Activación de usuarios: a través de un enlace se debe permitir la realización de la activación de un usuario.
- R4 – Ingreso al sistema: se debe permitir el ingreso al sistema con las credenciales registradas, mientras el usuario este activo o en estado 1.
- R5 – Administración de Tipos de BD: el usuario logueado como Administrador debe poder registrar los tipos de BD.
- R6 – Dashboard de datos del usuario: el usuario logueado como Usuario debe poder acceder a un dashboard o panel con los datos del número de tokens registrados, así como el número de reportes registrados
- R6.1 – Datos de seguimiento: el sistema debe mostrar los datos de seguimiento del usuario logueado.
- R7 – Registro de tokens: el sistema debe permitir el registro de los tokens de conexión incluyendo los datos básicos, así como seleccionando el tipo de BD.
- R8 – Validación de la conexión: el sistema debe poder permitir la validación de una conexión a un servidor de base de datos externo.
- R9 – Actualización de datos del token: el sistema debe permitir cargar la información del token y poder actualizar sus datos.
- R10 – Listar los tokens de conexión: el sistema debe permitir listar todos los tokens de conexión con su información básica que incluye estado.
- R11 – Listar los reportes: el sistema debe permitir listar los reportes de un usuario junto a su información básica. Importante el tipo de motor de base de datos a usar.
- R12 – Registrar un nuevo reporte: el sistema debe permitir a un usuario registrar un nuevo reporte, seleccionando la conexión del usuario al cual va a ser asignado.
- R12.1 – Cargar el archivo del reporte: como un valor adicional se debe poder cargar el archivo del reporte a utilizar.
- R13 – Listar reportes de una conexión: el sistema debe poder permitir listar los datos de una conexión y todos sus datos básicos, pero adicional debe permitir listar los reportes asociados a este token de conexión.
- R14 – Información de un reporte: el sistema debe permitir ver la información de un reporte, asi como los datos de conexión.
- R14.1 – Descargar el archivo del reporte: el sistema debe permitir recargar el archivo del reporte.
- R15 – Editar información de un reporte: el sistema debe permitir cargar la información de un reporte para poder editarla.

A continuación, se muestra el listado de requerimientos con su respectivo peso en el desarrollo de sus actividades. Para este ejercicio debe realizar un mínimo de **60** puntos del total disponible.

Requerimiento	Requisito	Puntos
R1		6
R2	R1	6
R3		10
R4		10
R5	R4	10
R6	R4	8
R6.1	R6	4
R7	R4	10
R8	R7	15
R9	R11	6
R10	R4	6
R11	R4	8
R12	R4	10
R12.1	R12	10
R13	R4	10
R14	R11	6
R14.1	R12.1	6
R15	R11	8

Limitantes

- Para el desarrollo de este ejercicio se debe usar una arquitectura sobre Java utilizando MAVEN.
- Se debe una arquitectura MVC identificando claramente las distintas capas del ejercicio.
- Se debe implementar un framework JPA.
- Para el Frontend debes utilizar tu propio diseño.

Procedimiento

Para el desarrollo del examen final deben tener cámara encendida, utilizar un repositorio en Github, el cual debe ser cargado en el enlace del examen final definido en el PLAD.

Deben realizar un commit al terminar cada requerimiento que vayan realizando. Al finalizar la entrega, en el repositorio deben incluir un archivo de texto donde describan los requerimientos realizados.

Script de la base de datos

```
CREATE TABLE `connectiontoken`
(
	`id` INT NOT NULL AUTO_INCREMENT,
	`host` VARCHAR(200) NULL COMMENT 'Corresponde a la dirección del servidor de base de datos',
	`userdb` VARCHAR(50) NULL COMMENT 'Corresponde al nombre del usuario de conexión a la base de datos',
	`pass` VARCHAR(100) NULL,
	`db` VARCHAR(50) NULL COMMENT 'Corresponde al nombre de la base de datos',
	`token` VARCHAR(50) NULL COMMENT 'Define el token que sera utilizado para realizar la busqueda de los datos de conexión a la base de datos.  Este token sera único y utilizado para la conexión',
	`port` SMALLINT NULL COMMENT 'Define el puerto utilizado para la conexión al servidor de base de datos',
	`user` INT NULL,
	`state` SMALLINT NULL,
	`type` VARCHAR(20) NULL COMMENT 'Define el tipo de driver a utilizar en la base de datos',
	CONSTRAINT `PK_connectiontoken` PRIMARY KEY (`id` ASC)
)
COMMENT = 'Almacena la información del Token de Conexión a al base de datos.  Este proceso se realiza por cada conexión que se realice a cada base de datos para ser utilizada y enviarla al reporte.';

CREATE TABLE `reporte`
(
	`id` INT NOT NULL AUTO_INCREMENT,
	`file` VARCHAR(100) NULL COMMENT 'Almacena el nombre del archivo jasper a utilizar',
	`conexion` INT NULL,
	`datecreate` TIMESTAMP NULL,
	`state` VARCHAR(50) NULL COMMENT 'Define el estado del reporte',
	`description` VARCHAR(500) NULL,
	`name` VARCHAR(200) NULL,
	CONSTRAINT `PK_reporte` PRIMARY KEY (`id` ASC)
)
COMMENT = 'Almacena la información de los hosteados en el sistema';

CREATE TABLE `rol`
(
	`id` INT NOT NULL AUTO_INCREMENT,
	`description` VARCHAR(100) NULL COMMENT 'Almacena la información del rol',
	CONSTRAINT `PK_rol` PRIMARY KEY (`id` ASC)
)
COMMENT = 'Almacena la información de los roles del sistema';
```

```
CREATE TABLE `seguimiento`
(
	`id` INT NOT NULL AUTO_INCREMENT,
	`report` INT NULL,
	`dategenerate` TIMESTAMP NULL,
	`state` SMALLINT NULL,
	`result` VARCHAR(500) NULL COMMENT 'Almacena el resultado de la consulta realizada al servidor',
	`detailrequest` VARCHAR(1000) NULL COMMENT 'Almacena los datos de las variables ingresadas',
	`type` VARCHAR(10) NULL COMMENT 'Define el tipo de generación, si fue xls o pdf.  ',
	`filegenerate` VARCHAR(100) NULL COMMENT 'Define el nombre del archivo generado',
	CONSTRAINT `PK_seguimiento` PRIMARY KEY (`id` ASC)
)
COMMENT = 'Almacena información acerca del uso de los reportes';

CREATE TABLE `typedb`
(
	`id` VARCHAR(20) NOT NULL,
	`description` VARCHAR(100) NULL,
	`driver` VARCHAR(200) NULL COMMENT 'Define los datos del driver',
	`aditional` VARCHAR(500) NULL COMMENT 'Define los valores adicionales del Driver',
	CONSTRAINT `PK_typedb` PRIMARY KEY (`id` ASC)
)
COMMENT = 'Almacena la información de los distintos tipos de conexión que soporta el sistema';

CREATE TABLE `usuario`
(
	`id` INT NOT NULL AUTO_INCREMENT,
	`usuario` VARCHAR(20) NULL COMMENT 'Define el usuario utilizado en el sistema, el cual no puede ser cambiado porque genera estructuras de datos y demas.',
	`email` VARCHAR(100) NULL,
	`pass` VARCHAR(50) NULL,
	`role` INT NULL,
	`state` SMALLINT NULL COMMENT 'Define el estado del usuario, que puede ser sin activar, bloqueado u otro estado disponible.',
	CONSTRAINT `PK_usuario` PRIMARY KEY (`id` ASC)
)
COMMENT = 'Almacena la información de los usuarios del sistema';

/* Create Foreign Key Constraints */

ALTER TABLE `connectiontoken`
 ADD CONSTRAINT `FK_connectiontoken_typedb`
```

```
        FOREIGN KEY (`type`) REFERENCES `typedb` (`id`) ON DELETE Restrict ON UPDATE Restrict;

ALTER TABLE `connectiontoken`
 ADD CONSTRAINT `FK_connectiontoken_usuario`
        FOREIGN KEY (`user`) REFERENCES `usuario` (`id`) ON DELETE Restrict ON UPDATE Restrict;

ALTER TABLE `reporte`
 ADD CONSTRAINT `FK_reporte_connectiontoken`
        FOREIGN KEY (`conexion`) REFERENCES `connectiontoken` (`id`) ON DELETE Restrict ON
UPDATE Restrict;

ALTER TABLE `seguimiento`
 ADD CONSTRAINT `FK_seguimiento_reporte`
        FOREIGN KEY (`report`) REFERENCES `reporte` (`id`) ON DELETE Restrict ON UPDATE
Restrict;

ALTER TABLE `usuario`
 ADD CONSTRAINT `FK_usuario_rol`
        FOREIGN KEY (`role`) REFERENCES `rol` (`id`) ON DELETE Restrict ON UPDATE Cascade;
```