

SISTEMAS ORIENTADOS A SERVICIOS

PRÁCTICA DE RESTful

TEMA: API de un calendario

Integrantes:

Angélica Guamán Albarracín, Matricula: t110212.

.

ÍNDICE

1. INTRODUCCIÓN.....	4
2. RECURSOS.....	5
2.1. RECURSOS SIMPLES.....	5
2.2. RECURSOS COMPUESTOS.....	5
3. ESQUEMA ENTIDAD/RELACIÓN	5
4. CREACIÓN DE LA BASE DE DATOS	6
4.1. CREACIÓN DE LA BBDD	6
4.2. TABLA USUARIO	6
4.3. TABLA CALENDARIO	7
4.4. TABLA CITA	7
5. INSERCIÓN DE DATOS.....	8
5.1. TABLA USUARIO	8
5.2. TABLA CALENDARIO	8
5.3. TABLA CITA	9
6. URIS Y VERBOS HTTP	10
6.1. OBTENEMOS TODOS LOS USUARIOS	10
6.2. CREAR UN USUARIO	10
6.3. MODIFICAR UN USUARIOS.....	11
6.4. ELIMINAR UN USUARIO	12
6.5. OBTENER UN USUARIO	12
6.6. CREAR UN CALENDARIO	13
6.7. MODIFICAR CALENDARIO.....	14
6.8. ELIMINAR UN CALENDARIO	14
6.9. OBTENER UN CALENDARIO DE UN USUARIO.....	15
6.10. OBTENER UNA LISTA DE TODAS NUESTRAS CITAS O CITAS PÚBLICOS DE OTROS USUARIOS.....	15
6.11. CREAR UNA CITA.....	17
6.12. MODIFICAR UNA CITA.....	18
6.13. ELIMINAR DE UNA CITA	19
6.14. OBTENER CITAS DE UN CALENDARIO	19
6.15. OBTENER UNA CITA DE UN CALENDARIO	20
6.16. OBTENER EL NÚMERO DE CITAS EN UN DETERMINADO PERIODO	20
7. TABLAS DE RECURSOS.....	21
7.1. OBTENEMOS TODOS LOS USUARIOS	21
7.2. CREAR UN USUARIO	22
7.3. MODIFICAR UN USUARIOS.....	22
7.4. ELIMINAR UN USUARIO	22
7.5. OBTENER UN USUARIO	22
7.6. CREAR UN CALENDARIO	23
7.7. MODIFICAR UN CAMPO DE UN CALENDARIO.....	24
7.8. ELIMINAR DE UN CALENDARIO	24
7.9. OBTENER CALENDARIOS DE UN USUARIO	24
7.10. CREAR UNA CITA.....	26
7.11. MODIFICAR UNA CITA.....	26
7.12. ELIMINAR DE UNA CITA	26
7.13. OBTENER UNA CITA DE UN CALENDARIO	28
7.14. OBTENER CITAS DE UN CALENDARIO	28

7.15.	OBTENER EL NÚMERO DE CITAS EN UN DETERMINADO PERIODO.....	29
7.16.	OBTENER UNA LISTA DE TODAS NUESTRAS CITAS O CITAS PÚBLICOS DE OTROS USUARIOS.....	29
8.	CONCLUSIÓN	31

1. INTRODUCCIÓN

Se trata de diseñar e implementar una API REST en la cual se va a trabajar con una lista de calendarios, donde un usuario puede tener asociado varios calendarios y un calendario puede tener varios dueños, un calendario está compuesta por una o varias citas.

Para la realización de esta API se ha utilizado NetBeans IDE 7.4 junto con MysqlWorkbench 5.2.47 y mysql 5.6.17.

2. RECURSOS

Lo primero que he hecho es identificar los recursos necesarios para la API, incluyendo colecciones, recursos compuestos, contenedores/fábricas, controladores etc.

2.1. RECURSOS SIMPLES

Usuario
Calendario
Cita

2.2. RECURSOS COMPUESTOS

Que estos son una colección de RECURSOS SIMPLES.

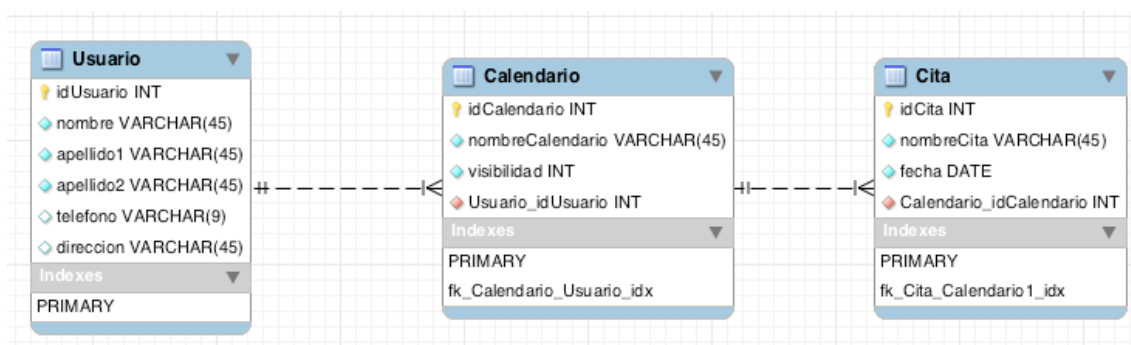
Usuarios
Calendarios
Citas

3. ESQUEMA ENTIDAD/RELACIÓN

Para implementar esta API partimos de un esquema E/R que lo hacemos con la aplicación de MysqlWorkbench cabe destacar que es una base de datos, donde cada tabla tendrá un identificador para hacer referencia a esa fila también habrá propagación de clave ajena. Lo primero que se debe hacer es añadir usuarios a nuestra base de datos, luego calendarios y por último las citas.

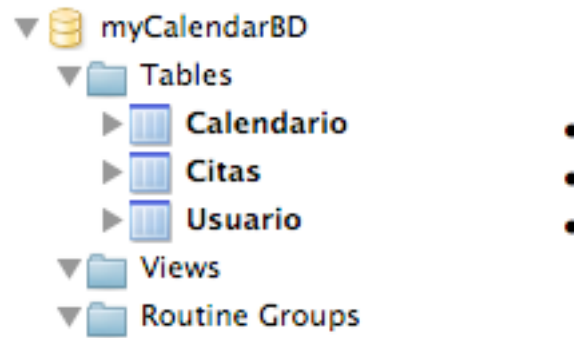
- Un usuario puede tener asociados varios calendarios.
- Un calendario puede tener varias citas.

La siguiente figura muestra la estructura de la base de datos:



4. CREACIÓN DE LA BASE DE DATOS

Como hemos utilizado la aplicación MysqlWorkbench que mediante su interfaz gráfica nos ha facilitado la creación de la BBDD, también nos genera un script con las sentencias sql de acuerdo a nuestros datos.



4.1. CREACIÓN DE LA BBDD

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS, UNIQUE_CHECKS=0;
SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,
FOREIGN_KEY_CHECKS=0;
SET @OLD_SQL_MODE=@@SQL_MODE,
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';

DROP SCHEMA IF EXISTS `calendarBBDD` ;
CREATE SCHEMA IF NOT EXISTS `calendarBBDD` DEFAULT CHARACTER SET utf8
COLLATE utf8_general_ci ;
USE `calendarBBDD` ;
```

4.2. TABLA USUARIO

```
-----
-- Table `calendarBBDD`.`Usuario`
-----
DROP TABLE IF EXISTS `calendarBBDD`.`Usuario` ;

CREATE TABLE IF NOT EXISTS `calendarBBDD`.`Usuario` (
  `idUsuario` INT NOT NULL ,
  `nombre` VARCHAR(45) NOT NULL ,
  `apellido1` VARCHAR(45) NOT NULL ,
  `apellido2` VARCHAR(45) NOT NULL ,
  `telefono` VARCHAR(9) NULL ,
  `direccion` VARCHAR(45) NULL ,
  PRIMARY KEY (`idUsuario`) )
ENGINE = InnoDB;
```

4.3. TABLA CALENDARIO

```
-----  
-- Table `calendarBBDD`.`Calendario`  
-----  
DROP TABLE IF EXISTS `calendarBBDD`.`Calendario` ;  
  
CREATE TABLE IF NOT EXISTS `calendarBBDD`.`Calendario` (  
  `idCalendario` INT NOT NULL ,  
  `nombreCalendario` VARCHAR(45) NOT NULL ,  
  `visibilidad` INT NOT NULL ,  
  `Usuario_idUsuario` INT NOT NULL ,  
  PRIMARY KEY (`idCalendario`),  
  INDEX `fk_Calendario_Usuario_idx` (`Usuario_idUsuario` ASC),  
  CONSTRAINT `fk_Calendario_Usuario`  
    FOREIGN KEY (`Usuario_idUsuario` )  
    REFERENCES `calendarBBDD`.`Usuario` (`idUsuario` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;
```

4.4. TABLA CITA

```
-----  
-- Table `calendarBBDD`.`Cita`  
-----  
DROP TABLE IF EXISTS `calendarBBDD`.`Cita` ;  
  
CREATE TABLE IF NOT EXISTS `calendarBBDD`.`Cita` (  
  `idCita` INT NOT NULL ,  
  `nombreCita` VARCHAR(45) NOT NULL ,  
  `fecha` DATE NOT NULL ,  
  `Calendario_idCalendario` INT NOT NULL ,  
  PRIMARY KEY (`idCita`),  
  INDEX `fk_Cita_Calendario1_idx` (`Calendario_idCalendario` ASC),  
  CONSTRAINT `fk_Cita_Calendario1`  
    FOREIGN KEY (`Calendario_idCalendario` )  
    REFERENCES `calendarBBDD`.`Calendario` (`idCalendario` )  
    ON DELETE NO ACTION  
    ON UPDATE NO ACTION)  
ENGINE = InnoDB;  
  
USE `calendarBBDD` ;  
  
SET SQL_MODE=@OLD_SQL_MODE;  
SET FOREIGN_KEY_CHECKS=@OLD_FOREIGN_KEY_CHECKS;  
SET UNIQUE_CHECKS=@OLD_UNIQUE_CHECKS;
```

5. INSERCIÓN DE DATOS

Código SQL para hacer la inserción de datos, son necesarios para poder realizar las pruebas en nuestra API.

5.1. TABLA USUARIO

```
INSERT INTO `calendarBBDD`.`Usuario` (`idUsuario`, `nombre`, `apellido1`, `apellido2`, `telefono`, `direccion`) VALUES ('101', 'Angelica', 'Guaman', 'Albarracin', '611223344', 'C/ Hortensia');
INSERT INTO `calendarBBDD`.`Usuario` (`idUsuario`, `nombre`, `apellido1`, `apellido2`, `telefono`, `direccion`) VALUES ('102', 'Ruth', 'Guaman', 'Albarracin', '622334455', 'C/ Blasco Jimeno');
INSERT INTO `calendarBBDD`.`Usuario` (`idUsuario`, `nombre`, `apellido1`, `apellido2`, `telefono`, `direccion`) VALUES ('103', 'Mayra', 'Rivas', 'Suasnavas', '633445566', 'C/ Vera');
INSERT INTO `calendarBBDD`.`Usuario` (`idUsuario`, `nombre`, `apellido1`, `apellido2`, `telefono`, `direccion`) VALUES ('104', 'Christian', 'Guaman', 'Cocha', '644556677', 'C/ Atocha');
INSERT INTO `calendarBBDD`.`Usuario` (`idUsuario`, `nombre`, `apellido1`, `apellido2`, `telefono`, `direccion`) VALUES ('105', 'Jose', 'Rincon', 'Quintero', '655667788', 'C/ Lechuga');
INSERT INTO `calendarBBDD`.`Usuario` (`idUsuario`, `nombre`, `apellido1`, `apellido2`, `telefono`, `direccion`) VALUES ('106', 'Jose', 'Rincon', 'Rojas', '666778899', 'C/ Miranda');
INSERT INTO `calendarBBDD`.`Usuario` (`idUsuario`, `nombre`, `apellido1`, `apellido2`, `telefono`, `direccion`) VALUES ('107', 'Ernesto', 'Rincon', 'Rojas', '677889900', 'C/ Virgen de la Vega');
```

5.2. TABLA CALENDARIO

```
INSERT INTO `calendarBBDD`.`Calendario` (`idCalendario`, `nombreCalendario`, `visibilidad`, `Usuario_idUsuario`) VALUES ('201', 'Calendario1', '1', '101');
INSERT INTO `calendarBBDD`.`Calendario` (`idCalendario`, `nombreCalendario`, `visibilidad`, `Usuario_idUsuario`) VALUES ('202', 'Calendario2', '1', '101');
INSERT INTO `calendarBBDD`.`Calendario` (`idCalendario`, `nombreCalendario`, `visibilidad`, `Usuario_idUsuario`) VALUES ('203', 'Calendario3', '0', '101');
INSERT INTO `calendarBBDD`.`Calendario` (`idCalendario`, `nombreCalendario`, `visibilidad`, `Usuario_idUsuario`) VALUES ('204', 'Calendario4', '1', '102');
INSERT INTO `calendarBBDD`.`Calendario` (`idCalendario`, `nombreCalendario`, `visibilidad`, `Usuario_idUsuario`) VALUES ('205', 'Calendario5', '1', '102');
INSERT INTO `calendarBBDD`.`Calendario` (`idCalendario`, `nombreCalendario`, `visibilidad`, `Usuario_idUsuario`) VALUES ('206', 'Calendario6', '1', '102');
INSERT INTO `calendarBBDD`.`Calendario` (`idCalendario`, `nombreCalendario`, `visibilidad`, `Usuario_idUsuario`) VALUES ('207', 'Calendario7', '0', '103');
INSERT INTO `calendarBBDD`.`Calendario` (`idCalendario`, `nombreCalendario`, `visibilidad`, `Usuario_idUsuario`) VALUES ('208', 'Calendario8', '0', '103');
INSERT INTO `calendarBBDD`.`Calendario` (`idCalendario`, `nombreCalendario`, `visibilidad`, `Usuario_idUsuario`) VALUES ('209', 'Calendario9', '0', '103');
INSERT INTO `calendarBBDD`.`Calendario` (`idCalendario`, `nombreCalendario`, `visibilidad`, `Usuario_idUsuario`) VALUES ('210', 'Calendario10', '1', '104');
INSERT INTO `calendarBBDD`.`Calendario` (`idCalendario`, `nombreCalendario`, `visibilidad`, `Usuario_idUsuario`) VALUES ('211', 'Calendario11', '1', '104');
```


5.3. TABLA CITA

```
INSERT INTO `calendarBBDD`.`Cita` (`idCita`, `nombreCita`, `fecha`,  
`Calendario_idCalendario`) VALUES ('301', 'Cita1', '2014-01-10', '201');  
INSERT INTO `calendarBBDD`.`Cita` (`idCita`, `nombreCita`, `fecha`,  
`Calendario_idCalendario`) VALUES ('302', 'Cita2', '2014-05-07', '201');  
INSERT INTO `calendarBBDD`.`Cita` (`idCita`, `nombreCita`, `fecha`,  
`Calendario_idCalendario`) VALUES ('303', 'Cita3', '2014-03-12', '201');  
INSERT INTO `calendarBBDD`.`Cita` (`idCita`, `nombreCita`, `fecha`,  
`Calendario_idCalendario`) VALUES ('304', 'Cita4', '2014-04-04', '202');  
INSERT INTO `calendarBBDD`.`Cita` (`idCita`, `nombreCita`, `fecha`,  
`Calendario_idCalendario`) VALUES ('305', 'Cita5', '2014-06-01', '202');  
INSERT INTO `calendarBBDD`.`Cita` (`idCita`, `nombreCita`, `fecha`,  
`Calendario_idCalendario`) VALUES ('306', 'Cita6', '2015-07-08', '202');
```

6. URIS Y VERBOS HTTP

Para probar el funcionamiento de la API he hecho uso de una aplicación REST client.

6.1. OBTENEMOS TODOS LOS USUARIOS

Para ello utilizamos el verbo GET de la siguiente URI:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/>

El resultado que obtenemos es el siguiente:

```
▼<usuario>
  <apellido1>aman</apellido1>
  <apellido2>Albarracin</apellido2>
  <direccion>C/ Hortensia</direccion>
  <idUsuario>101</idUsuario>
  <nombre>Angelica</nombre>
  <telefono>611223344</telefono>
</usuario>
▼<usuario>
  <apellido1>Guaman</apellido1>
  <apellido2>Albarracin</apellido2>
  <direccion>C/ Blasco Jimeno</direccion>
  <idUsuario>102</idUsuario>
  <nombre>Ruth</nombre>
  <telefono>622334455</telefono>
</usuario>
▼<usuario>
  <apellido1>Rivas</apellido1>
  <apellido2>Suasnavas</apellido2>
  <direccion>C/ Vera</direccion>
  <idUsuario>103</idUsuario>
  <nombre>Mayra</nombre>
  <telefono>633445566</telefono>
</usuario>
```

Podemos observar que se trata de una colección de usuario.

6.2. CREAR UN USUARIO

Para crear un usuario utilizamos el verbo POST con el @Path("{id}").

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/112>

El siguiente código es necesario para poder crear el usuario:

```
<?xml version="1.0" encoding="UTF-8"?>
<usuario>
  <apellido1>Alvarez</apellido1>
  <apellido2>Albarracin</apellido2>
  <direccion>C/ Huerta</direccion>
  <idUsuario>112</idUsuario>
  <nombre>Angie</nombre>
  <telefono>611223300</telefono>
</usuario>
```

Resultado de la petición:

```
<usuario>
  <apellido1>Alvarez</apellido1>
  <apellido2>Albarracin</apellido2>
  <direccion>C/ Huerta</direccion>
  <idUsuario>112</idUsuario>
  <nombre>Angie</nombre>
  <telefono>611223300</telefono>
</usuario>
```

6.3. MODIFICAR UN USUARIOS

Para modificar un usuario utilizamos el verbo PUT con el @Path("{id}").

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/101>

El siguiente código es necesario para poder modificar el usuario:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<usuario>
  <apellido1>SOS</apellido1>
  <apellido2>Albarracin</apellido2>
  <direccion>C/ Hortensia</direccion>
  <idUsuario>101</idUsuario>
  <nombre>Angelica</nombre>
  <telefono>611223344</telefono>
</usuario>
```

Resultado de la petición:

```
<usuario>
  <apellido1>SOS</apellido1>
  <apellido2>Albarracin</apellido2>
  <direccion>C/ Hortensia</direccion>
  <idUsuario>101</idUsuario>
  <nombre>Angelica</nombre>
  <telefono>611223344</telefono>
</usuario>
```

6.4. ELIMINAR UN USUARIO

Para realizar esta petición utilizamos el verbo DELETE que nos devuelve un OK indicando que lo ha eliminado.

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/112>

Ahora hacemos un GET de la URI anterior y nos devolverá un NOT_CONTENT.

Status **204 No Content** ? Loading time: 14 ms

6.5. OBTENER UN USUARIO

Para realizar esta petición utilizamos el verbo GET con el @Path("{id}") indicando en la URI el identificador de usuario.

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/101>

Resultado de la petición:

```
<usuario>
  <apellido1>SOS</apellido1>
  <apellido2>Albarracin</apellido2>
  <direccion>C/ Hortensia</direccion>
  <idUsuario>101</idUsuario>
  <nombre>Angelica</nombre>
  <telefono>611223344</telefono>
</usuario>
```

6.6. CREAR UN CALENDARIO

Para esta petición utilizamos el verbo POST especificando el id de usuario, y pasándole el código XML del nuevo calendario.

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/101/calendarios/>

Código XML que he utilizado para crear el nuevo calendario son un usuario existente en la base de datos.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<calendario>
  <idCalendario>230</idCalendario>
  <nombreCalendario>CalendarioSOS</nombreCalendario>
  <usuarioidUsuario>
    <apellido1>SOS</apellido1>
    <apellido2>Albarracin</apellido2>
    <direccion>C/ Hortensia</direccion>
    <idUsuario>101</idUsuario>
    <nombre>Angelica</nombre>
    <telefono>611223344</telefono>
  </usuarioidUsuario>
  <visibilidad>0</visibilidad>
</calendario>
```

Resultado de la petición:

```
<calendario>
  <idCalendario>230</idCalendario>
  <nombreCalendario>CalendarioSOS</nombreCalendario>
  <usuarioidUsuario>
    <apellido1>SOS</apellido1>
    <apellido2>Albarracin</apellido2>
    <direccion>C/ Hortensia</direccion>
    <idUsuario>101</idUsuario>
    <nombre>Angelica</nombre>
    <telefono>611223344</telefono>
  </usuarioidUsuario>
  <visibilidad>0</visibilidad>
</calendario>
```

6.7. MODIFICAR CALENDARIO

Utilizamos el verbo PUT especificando el id de usuario, y de queryParams indicando el id de calendario y el nombre nuevo del calendario.

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/101/calendarios?idCalendario=230&nombreCalendario=EXAMEN DE SOS>

Respuesta que nos indica al hacer el cambio:

Status **201 Created** ? Loading time: 243 ms

Calendario modificado:

```
<calendario>
  <idCalendario>230</idCalendario>
  <nombreCalendario>"EXAMEN DE SOS"</nombreCalendario>
  <usuarioidUsuario>
    <apellido1>SOS</apellido1>
    <apellido2>Albarracin</apellido2>
    <direccion>C/ Hortensia</direccion>
    <idUsuario>101</idUsuario>
    <nombre>Angelica</nombre>
    <telefono>611223344</telefono>
  </usuarioidUsuario>
  <visibilidad>0</visibilidad>
</calendario>
```

6.8. ELIMINAR UN CALENDARIO

Utilizamos el verbo DELETE especificando el id del calendario a eliminar.

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.calendario/204/>

Respuesta que devuelve al eliminar el calendario:

Status **404 Not Found** ? Loading time: 12 ms

6.9. OBTENER UN CALENDARIO DE UN USUARIO

Para realizar esta petición utilizamos el verbo GET indicando el id de usuario y el id del calendario del que queremos ver sus datos.

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/101/calendarios/230>

Resultado de la petición:

```
<calendario>
  <idCalendario>230</idCalendario>
  <nombreCalendario>CalendarioSOS</nombreCalendario>
  <usuarioidUsuario>
    <apellido1>SOS</apellido1>
    <apellido2>Albarracin</apellido2>
    <direccion>C/ Hortensia</direccion>
    <idUsuario>101</idUsuario>
    <nombre>Angelica</nombre>
    <telefono>611223344</telefono>
  </usuarioidUsuario>
  <visibilidad>0</visibilidad>
</calendario>
```

6.10. OBTENER UNA LISTA DE TODAS NUESTRAS CITAS O CITAS PÚBLICAS DE OTROS USUARIOS

Para realizar esta petición utilizamos el verbo GET indicando el id de usuario y de queryParam el idUsuario el identificador de usuario donde si estos son iguales nos muestra todas las citas y si son distintos nos muestra las citas de los calendarios públicos de idUsuario.

Podemos filtrar por fecha de inicio (from), final (to) o que las citas se encuentren entre las dos fechas.

URI a utilizar para mostrar la fecha antes de la fecha Final:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/101/citas?idUsuario=101&to=2015-07-08>

URI a utilizar para mostrar la fecha desde o después de la fecha Inicio:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/101/citas?idUsuario=101&from=2015-07-08>

URI a utilizar especificando desde que fechas queremos mostrar las citas:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/101/citas?idUsuario=101&from=2015-05-07&to=2018-07-07>

Respuesta que devuelve al mostrar las citas:

Status 200 OK ? Loading time: 53 ms

Resultado de la petición:

```
<Citas>
  <cita>
    <calendarioidCalendario>
      <idCalendario>260</idCalendario>
      <nombreCalendario>CalendarioSOS
      </nombreCalendario>
      <usuarioidUsuario>
        <apellidol>SOS</apellidol>
        <apellido2>Albarracin</apellido2>
        <direccion>C/ Hortensia</direccion>
        <idUsuario>101</idUsuario>
        <nombre>Angelica</nombre>
        <telefono>611223344</telefono>
      </usuarioidUsuario>
      <visibilidad>0</visibilidad>
    </calendarioidCalendario>
    <fecha>2015-07-08T00:00:00+02:00</fecha>
    <idCita>306</idCita>
    <nombreCita>Cita6</nombreCita>
  </cita>
</cita>
```

También podemos indicar que la cantidad de citas que queremos mostrar:

El formato de la fecha es yyyy-MM-dd, tal y como se muestran en la BBDD.

URI a utilizar para esta petición es:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/101/citas?idUsuario=101&from=2015-05-07&to=2018-07-07&max=1>

6.11. CREAR UNA CITA

Para esta petición utilizamos el verbo POST especificando el id del calendario, y pasándole el código XML de la nueva cita.

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.calendario/260/citas>

Respuesta que devuelve al crear la cita:

Status **201 Created** Loading time: 113 ms

Resultado de la petición:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<cita>
  <calendarioidCalendario>
    <idCalendario>260</idCalendario>
    <nombreCalendario>CalendarioSOS</nombreCalendario>
    <usuarioidUsuario>
      <apellido1>SOS</apellido1>
      <apellido2>Albarracin</apellido2>
      <direccion>C/ Hortensia</direccion>
      <idUsuario>101</idUsuario>
      <nombre>Angelica</nombre>
      <telefono>611223344</telefono>
    </usuarioidUsuario>
    <visibilidad>0</visibilidad>
  </calendarioidCalendario>
  <fecha>2015-07-08T00:00:00+02:00</fecha>
  <idCita>388886</idCita>
  <nombreCita>Cita6</nombreCita>
</cita>
```

Código XML indicando la nueva cita:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<cita>
  <calendarioidCalendario>
    <idCalendario>260</idCalendario>
    <nombreCalendario>CalendarioSOS</nombreCalendario>
    <usuarioidUsuario>
      <apellido1>SOS</apellido1>
      <apellido2>Albarracin</apellido2>
      <direccion>C/ Hortensia</direccion>
      <idUsuario>101</idUsuario>
      <nombre>Angelica</nombre>
      <telefono>611223344</telefono>
    </usuarioidUsuario>
    <visibilidad>0</visibilidad>
  </calendarioidCalendario>
  <fecha>2015-07-08T00:00:00+02:00</fecha>
  <idCita>388886</idCita>
  <nombreCita>Cita6</nombreCita>
</cita>
```

6.12. MODIFICAR UNA CITA

Utilizamos el verbo PUT especificando el id de calendario y el código XML

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.calendario/201/citas>

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<cita>
  <calendarioidCalendario>
    <idCalendario>201</idCalendario>
    <nombreCalendario>Calendario1</nombreCalendario>
    <usuarioidUsuario>
      <apellido1>SOS</apellido1>
      <apellido2>Albarracin</apellido2>
      <direccion>C/ Hortensia</direccion>
      <idUsuario>101</idUsuario>
      <nombre>Angelica</nombre>
      <telefono>611223344</telefono>
    </usuarioidUsuario>
    <visibilidad>1</visibilidad>
  </calendarioidCalendario>
  <fecha>2014-01-10T00:00:00+01:00</fecha>
  <idCita>301</idCita>
  <nombreCita>HACER LAS PRACTICAS</nombreCita>
</cita>
```

Respuesta que devuelve al hacer la modificación:

Status **201 Created** ? Loading time: 147 ms

Resultado de la petición:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes" ?>
<cita>
  <calendarioidCalendario>
    <idCalendario>201</idCalendario>
    <nombreCalendario>Calendario1</nombreCalendario>
    <usuarioidUsuario>
      <apellido1>SOS</apellido1>
      <apellido2>Albarracin</apellido2>
      <direccion>C/ Hortensia</direccion>
      <idUsuario>101</idUsuario>
      <nombre>Angelica</nombre>
      <telefono>611223344</telefono>
    </usuarioidUsuario>
    <visibilidad>1</visibilidad>
  </calendarioidCalendario>
  <fecha>2014-01-10T00:00:00+01:00</fecha>
  <idCita>301</idCita>
  <nombreCita>HACER LAS PRACTICAS</nombreCita>
</cita>
```

6.13. ELIMINAR DE UNA CITA

Devolviendo un OK, indicando que ha hecho la petición correctamente, luego hacemos un GET de la Citas de ese Calendario y nos damos cuenta que ya no aparece en nuestros citas

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.calendario/260/citas?idCita=305>

Resultado de la petición:

```
<Citas>
  <cita>
    <calendarioidCalendario>
      <idCalendario>260</idCalendario>
      <nombreCalendario>CalendarioSOS</nombreCalendario>
      <usuarioidUsuario>
        <apellido1>SOS</apellido1>
        <apellido2>Albarracin</apellido2>
        <direccion>C/ Hortensia</direccion>
        <idUsuario>101</idUsuario>
        <nombre>Angelica</nombre>
        <telefono>611223344</telefono>
      </usuarioidUsuario>
      <visibilidad>0</visibilidad>
    </calendarioidCalendario>
    <fecha>2015-07-08T00:00:00+02:00</fecha>
    <idCita>306</idCita>
    <nombreCita>Cita6</nombreCita>
  </cita>
</Citas>
```

6.14. OBTENER CITAS DE UN CALENDARIO

Para realizar esta petición utilizamos el verbo GET indicando el id de calendario.

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.calendario/201/citas>

Resultado de la petición:

```
<Citas>
  <cita>
    <calendarioidCalendario>
      <idCalendario>201</idCalendario>
      <nombreCalendario>Calendario1</nombreCalendario>
      <usuarioidUsuario>
        <apellido1>SOS</apellido1>
        <apellido2>Albarracin</apellido2>
        <direccion>C/ Hortensia</direccion>
        <idUsuario>101</idUsuario>
        <nombre>Angelica</nombre>
        <telefono>611223344</telefono>
      </usuarioidUsuario>
      <visibilidad>1</visibilidad>
    </calendarioidCalendario>
    <fecha>2014-01-10T00:00:00+01:00</fecha>
    <idCita>301</idCita>
    <nombreCita>Cita1</nombreCita>
  </cita>
</Citas>
```

6.15. OBTENER UNA CITA DE UN CALENDARIO

Para realizar esta petición utilizamos el verbo GET indicando el id de calendario y el id de la cita.

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.calendario/204/citas/301>

Resultado de la petición:

```
<?xml version="1.0" encoding="UTF-8"?>
<cita>
  <calendarioidCalendario>
    <idCalendario>204</idCalendario>
    <nombreCalendario>Calendario4</nombreCalendario>
    <usuarioidUsuario>
      <apellido1>Guaman</apellido1>
      <apellido2>Albarracin</apellido2>
      <direccion>C/ Blasco Jimeno</direccion>
      <idUsuario>102</idUsuario>
      <nombre>Ruth</nombre>
      <telefono>622334455</telefono>
    </usuarioidUsuario>
    <visibilidad>1</visibilidad>
  </calendarioidCalendario>
  <fecha>2014-01-10T00:00:00+01:00</fecha>
  <idCita>301</idCita>
  <nombreCita>Cital</nombreCita>
</cita>
```

6.16. OBTENER EL NÚMERO DE CITAS EN UN DETERMINADO PERIODO

Con esta petición mostramos el número de citas que hay en un periodo de fechas que se la pasamos por queryParam, utilizamos el verbo GET.

URI a utilizar:

<http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.cita/countCitas?from=2015-07-10>

Respuesta que devuelve:

Status **200 OK** ? Loading time: 12 ms

Resultado de la petición:

Raw	Parsed	Response
Open output in new window Copy to clipboard Save as file Open in JSON tab		
0		
Code highlighting thanks to Code Mirror		

7. TABLAS DE RECURSOS

Tablas en la que muestran los identificadores que se utilizan para las peticiones, las queryParam y el tipo de petición que es y los valores que devuelven la petición que cualquier caso(éxito, fallo, etc).

7.1. OBTENEMOS TODOS LOS USUARIOS

URI		
http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/		
Método	GET	
Cadena de consulta (GET)	No necesita ningún id, muestra todos	
Devuelve	200	OK
	401	Unauthorized
	204	No Content

7.2. CREAR UN USUARIO

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/{id}		
Método	POST	
Cadena de consulta (POST)	id = Identificador de usuario, es único.	
	Código	Utilizamos un código XML con los datos del usuario
Cuerpo de la petición (POST)	POX (apiCalendario/entidades.usuario/{id} +XML)	
Devuelve	201	Created y Cabecera Location
	401	Unauthorized
	415	Conflict

7.3. MODIFICAR UN USUARIOS

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/{id}		
Método	PUT	
Cadena de consulta (PUT)	id = Identificador de usuario, es único.	
	Código	Utilizamos código XML indicando los campos a modificar

Cuerpo de la petición (PUT)	POX (apiCalendario/entidades.usuario +XML)	
Devuelve	201	Created y Cabecera Location
	404	NotFound
	401	Unauthorized
	415	Unsuported Media Types.

7.4. ELIMINAR UN USUARIO

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/{id}		
Método	DELETE	
Cadena de consulta (DELETE)	id = Identificador de usuario, es único.	
Devuelve	200	OK
	401	Unauthorized
	204	No Content

7.5. OBTENER UN USUARIO

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/{id}		
Método	GET	

Cadena de consulta (GET)	id = Identificador de usuario, es único.	
Devuelve	200	OK
	401	Unauthorized
	204	No Content

7.6. CREAR UN CALENDARIO

URI		
http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/{id}/calendarios/		
Método	POST	
Cadena de consulta (POST)	Id = Identificador de usuario, es único.	
Código	Necesitamos código XML para la creación del calendario	
Cuerpo de la petición (POST)	POX (apiCalendario/entidades.usuario +XML)	
Devuelve	201	Created y Cabecera Location
	401	Unauthorized
	415	Conflict

7.7. MODIFICAR UN CAMPO DE UN CALENDARIO

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/{idUserio}/calendarios?idCalendario=230&nombreCalendario="EXAMEN DE SOS"		
Método	PUT	
Cadena de consulta (PUT)	idUserio = Identificador de usuario, es único.	
Query Param	idCalendario = Identificador del calendario, único nombreCalendario = nombre del calendario	
Cuerpo de la petición (PUT)	POX (apiCalendario/entidades.usuario +XML)	
Devuelve	201	Created y Cabecera Location
	404	NotFound
	401	Unauthorized
	415	Unsuported Media Types.

7.8. ELIMINAR DE UN CALENDARIO

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.calendario{id}/		
Método	DELETE	
Cadena de consulta (DELETE)	id = Identificador de calendario, es único.	

Devuelve	200	OK
	401	Unauthorized
	204	No Content

7.9. OBTENER CALENDARIOS DE UN USUARIO

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/{id}/calendarios/{idCalendario}		
Método	GET	
Cadena de consulta (GET)	id = Identificador de usuario, es único. idCalendario = Identificador de calendario, es único.	
Devuelve	200	OK
	401	Unauthorized
	204	No Content

7.10. CREAR UNA CITA

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.calendario/{id}/citas		
Método	POST	
Cadena de consulta (POST)	id = Identificador de calendario, es único.	
Código	Para esta petición hay que añadir código XML	
Cuerpo de la petición (POST)	POX (apiCalendario/entidades.calendario +XML)	
Devuelve	201	Created y Cabecera Location
	401	Unauthorized
	415	Conflict

7.11. MODIFICAR UNA CITA

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.calendario/{idCalendario}/citas		
Método	PUT	
Cadena de consulta (PUT)	idCalendario = Identificador de calendario, es único.	
Código	Código XML para modificar la cita	

Cuerpo de la petición (PUT)	POX (apiCalendario/entidades.calendario +XML)	
Devuelve	201	Created y Cabecera Location
	404	NotFound
	401	Unauthorized
	415	Unsuported Media Types.

7.12. ELIMINAR DE UNA CITA

URI		
http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.calendario/{idCalendario}/citas?idCita=305		
Método	DELETE	
Cadena de consulta (DELETE)	idCalendario = Identificador de calendario, es único.	
	idCita=identificador de la cita, único	
Query Param		
Devuelve	200	OK
	401	Unauthorized
	204	No Content

7.13. OBTENER UNA CITA DE UN CALENDARIO

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.calendario/{idCalendario}/citas/{idCita}		
Método	GET	
Cadena de consulta (GET)	idCalendario = Identificador de calendario, es único. idCita = Identificador de cita, es único.	
Cuerpo de la petición (POST)	POX (apiCalendario/entidades.calendario +XML)	
Devuelve	200	OK
	401	Unauthorized
	204	No Content

7.14. OBTENER CITAS DE UN CALENDARIO

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.calendario/{idCalendario}/citas		
Método	GET	
Cadena de consulta (GET)	idCalendario = Identificador de calendario, es único.	

Devuelve	200	OK
	401	Unauthorized
	204	No Content

7.15. OBTENER EL NÚMERO DE CITAS EN UN DETERMINADO PERIODO

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.cita/countCitas?from=2015-07-10		
Método	GET	
Cadena de consulta (GET)		
QueryParam	Tenemos QueryParam como from, to, para saber los intervalos de las fechas, para poder mostrar el número de citas que tenemos.	
Devuelve	200	OK
	401	Unauthorized
	204	No Content

7.16. OBTENER UNA LISTA DE TODAS NUESTRAS CITAS O CITAS PÚBLICAS DE OTROS USUARIOS

URI http://localhost:8080/CalendarioRESTful/apiCalendario/entidades.usuario/{idUser}/citas?idUsuario=101&from=d1&to=d2&max=n		
Método	GET	

Cadena de consulta (GET)	IdUsuario= identificador de usuario	
QueryParam	Tenemos QueryParam como from, to, para saber los intervalos de las fechas a mostrar, con idCalendario el identificador de calendario, y max el máximo numero de citas a mostrar	
Devuelve	200	OK
	401	Unauthorized
	204	No Content

8. CONCLUSIÓN

Me ha parecido una práctica muy interesante ya que he aprendido como funcionan las peticiones http, las operaciones que se pueden hacer y como se utilizan los parámetros (queryParams) para poder filtrar información, también como hacer consultas de los datos.

Se podía haber mejorado mi práctica en la inserción o modificación de una cita, ya que solo bastaba con el idCalendario, en mi caso hay que introducir todos los datos del calendario para introducir la cita. Así mismo se podía haber mejorado en la inserción o modificación de un calendario.