



DESARROLLO DE SISTEMAS DISTRIBUIDOS

TAREA #14

REPORTES CON JASPER REPORTS

14 de Septiembre del 2018

PROFESOR: ENRIQUE ZÁRATE JOSÉ ASUNCIÓN

ALUMNO: SALDAÑA AGUILAR ANDRÉS

GRUPO: 4CM3

Introducción

RMI (Java Remote Method Invocation) es un mecanismo ofrecido por Java para invocar un método de manera remota. Forma parte del entorno estándar de ejecución de Java y proporciona un mecanismo simple para la comunicación de servidores en aplicaciones distribuidas basadas exclusivamente en Java. Si se requiere comunicación entre otras tecnologías debe utilizarse CORBA o SOAP en lugar de RMI.

RMI se caracteriza por la facilidad de su uso en la programación por estar específicamente diseñado para Java; proporciona paso de objetos por referencia (no permitido por SOAP), recolección de basura distribuida (Garbage Collector distribuido) y paso de tipos arbitrarios (funcionalidad no provista por CORBA).

A través de RMI, un programa Java puede exportar un objeto, con lo que dicho objeto estará accesible a través de la red y el programa permanece a la espera de peticiones en un puerto TCP. A partir de ese momento, un cliente puede conectarse e invocar los métodos proporcionados por el objeto.

JasperReports es una biblioteca de creación de informes que tiene la habilidad de entregar contenido enriquecido al monitor, a la impresora o a ficheros PDF, HTML, XLS, CSV y XML.

Desarrollo

El objetivo de este ejercicio es crear una aplicación cliente y servidor que haga uso de la tecnología RMI para la comunicación y que permita realizar la consulta de las entidades de la base de datos escuelas y carreras, así como el reporte de alumnos por carreras:

A continuación, la base de datos y sus registros,

Escuelas

idEscuela	nombreEscuela	calle	numero
2	ESIA	Otra Calle	2888
3	ESIT	Otra Calle	1888
4	ESIOIE	Juan de dios Batiz	1888

Alumnos

id	nombre	paterno	idCarrera
1	Andres	Saldana	1
2	Gabriela	Saldana	1
3	Irving	Ortiz	1
4	Brendon	Altuzar	1
5	Soeni	Bias	2
6	Ricardo	Culebra	2

Carreras

id	nombreCarrera
1	Sistemas
2	Fisico Matematico

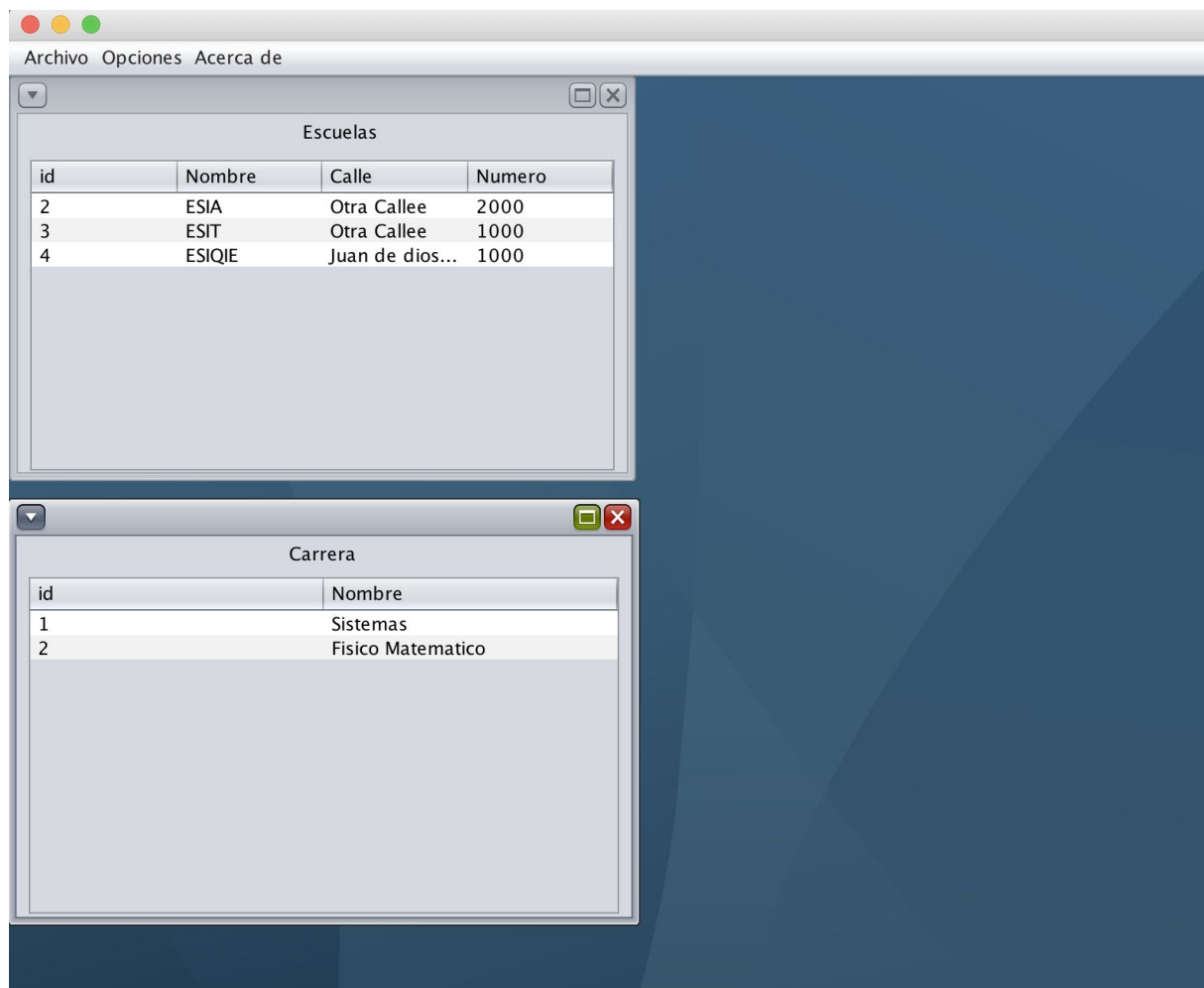
En resumen, cree dos proyectos, uno para el cliente y otro para el servidor, donde:

Servidor: consiste de una interfaz de las acciones que realiza la aplicación y su implementación, la clase que crea el objeto que será accesible por el cliente, clase que se encarga de la persistencia de la base (persistence unit), las entidades necesarias (escuela, carrera, alumno) y clase que tiene la plantilla del reporte.

Cliente: tiene una aplicación gráfica, una interfaz que el servidor se encargará de implementar, y los paneles internos donde mostraremos la lista de registros.

Resultados

En la interfaz, tendremos tres opciones, la primera, es aquella donde visualizamos las escuelas, y la segunda es la visualización de carreras:



Modifique el código de la sección de carreras para incluir un CRUD completo sobre esa entidad, donde se listan las carreras, borramos carreras, agregamos carreras y modificamos carreras:

The 'Carrera' window displays a table with the following data:

id	Nombre
2	Fisico Matematico
3	quimica industrial
4	ingenieria quimica
5	ingenieria textil
6	ingenieria mecanica

Below the table, there is a text input field labeled 'Nombre de la Carrera' and another labeled 'Identificador'. To the right of the 'Identificador' field are two buttons: 'Agregar' and 'Confirmar'.

Si queremos agregar una nueva carrera, solamente debemos poner el nombre y dar en confirmar que dará un mensaje de éxito si todo sale bien y al confirmar este mismo, actualizará la lista que proviene de la base de datos, en el ejemplo agregue Arquitecto:

The left screenshot shows the 'Carrera' window with 'Arquitecto' entered in the 'Nombre de la Carrera' field. A 'Message' dialog box is displayed in the foreground with the text 'Exito al insertar' and an 'OK' button.

The right screenshot shows the 'Carrera' window after the update. The table now includes a new row:

id	Nombre
2	Fisico Matematico
3	quimica industrial
4	ingenieria quimica
5	ingenieria textil
6	ingenieria mecanica
7	Arquitecto

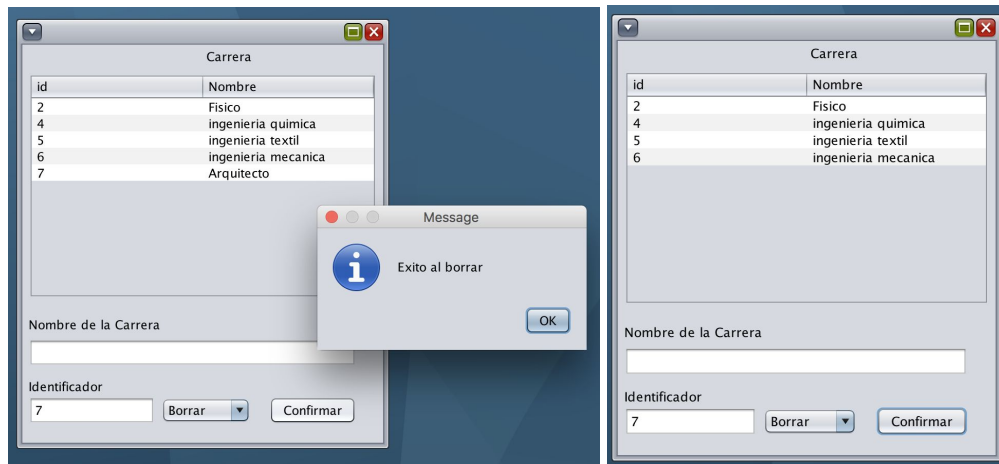
Si queremos actualizar un registro, pondremos el nuevo nombre que se le dará al registro, y el id que tiene relacionado, en el ejemplo modifique química industrial por Quimica:

The left screenshot shows the 'Carrera' window with 'Quimica' entered in the 'Nombre de la Carrera' field and '3' entered in the 'Identificador' field. A 'Message' dialog box is displayed in the foreground with the text 'Exito al actualizar' and an 'OK' button.

The right screenshot shows the 'Carrera' window after the update. The table now reflects the change:

id	Nombre
2	Fisico
3	Quimica
4	ingenieria quimica
5	ingenieria textil
6	ingenieria mecanica
7	Arquitecto

Por último, al borrar un registro de Carrera, solamente debemos proporcionar el id del registro a borrar, en este caso borre el registro número 7, Arquitecto:



También está la sección de alumnos por carrera, que genera un reporte con los alumnos ordenados por carrera que puedes guardar en formato pdf.

The image shows a screenshot of a JasperViewer window. The window title is 'JasperViewer'. The report displayed is titled 'Alumnos por Carrera'. The report contains a table with the following data:

id	nombre	paterno	idCarrera
1	Andres	Saldana	1
2	Gabriela	Saldana	1
3	Irving	Ortiz	1
4	Brandon	Altuzar	1
5	Soemi	Blas	2
6	Ricardo	Culebro	2

Conclusiones

El envío de información con el uso de RMI es muy sencillo y tiene muchas aplicaciones, así como el uso de Jasper Reports, que facilita mucho el diseño y creación de reportes de la base de datos.

Referencias bibliográficas

Instalacion de jasperReports: <https://www.youtube.com/watch?v=eaGLX7UlrnA&t=431s>