Instituto Tecnológico Autónomo de México

Sistemas Distribuidos

DataWebWizard

Proyecto Omega

Amanda Velasco Gallardo

154415

Eduardo Woldenberg Esperón

158279

17 de mayo de 2019

Introducción

El proyecto DataWeb Wizard tenía como objetivo la construcción de un programa que generara conexiones a una base de datos con base en varios servicios web vistos en clase y tuviera la habilidad de ejecutar operaciones de usuario sobre la base de datos dinámicamente.

Específicamente, el proyecto le permite al usuario el uso de un sitio web para ejecutar todas las operaciones necesarias para manejar sus propias tablas en una base de datos general. Es decir, puede generar una tabla de cantidad de entradas dinámica, editar, borrar e insertar registros a la misma y ver las columnas presentes dentro de la tabla y ver los valores dentro de ésta. Aparte de esto, los usuarios nuevos deben registrarse con un nombre de usuario y contraseña para poder hacer uso de la base de datos. Todas las conexiones del usuario a la base de datos se hacen ejecutando servicios web RESTful que aparte utilizan servicios web SOAP para ejecutar las solicitudes hechas a la base. También vale la pena notar que la visualización de datos se hace utilizando tecnologías AJAX y tiene una forma de desplazarse a través de los resultados.

Obstáculos encontrados

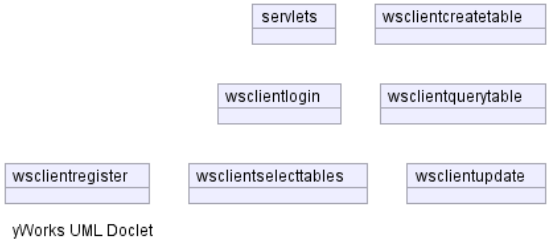
A través del curso del desarrollo surgieron tres obstáculos especialmente notables al progreso del programa; el primero en aparecer fue en el uso de git para controlar las versiones entre miembros del equipo. Aparte de problemas con la utilización del software por falta de experiencia, hubo problemas al intentar ejecutar código escrito en una computadora en otra. Esto es porque NetBeans al crear el proyecto genera muchas cosas escondidas en segundo plano las cuales normalmente no es recomendable cambiar y causó problemas sustanciales en que segmentos de código funcionaban en una computadora y no en otra y viceversa. Para resolver esto se juntaron todos los proyectos en una misma computadora y ésta se utilizó para pruebas. La otra se utilizó aún para programar, pero en segmentos de código pequeños y característica por característica, en lugar de manejar toda la codebase simultáneamente.

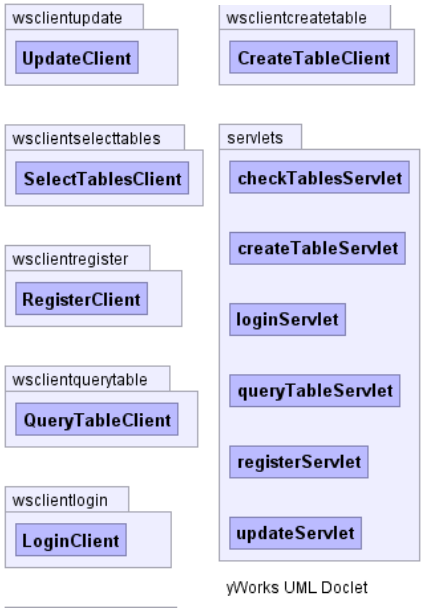
El segundo gran obstáculo surgió al intentar implementar muchos servicios web al mismo tiempo en el proyecto; esto causaba que dejara de funcionar el WSDL al intentar crear varias clases u objetos con el mismo nombre dentro del mismo folder de Windows. Para arreglar esto se separaron todos los servicios SOAP en su propio paquete de servicio y todos los servicios RESTful, en otro proyecto por separado, también fueron especialmente delineados.

El tercer y último problema ocurrió al utilizar JSONObjects para mandar mensajes obtenidos del manejador de bases de datos; lo que sucedía era que el JSONObject no es una estructura ordenada y al llenarlo con los resultados de una consulta, éstos se almacenaban en desorden. Esto causaba problemas al intentar interactuar con los registros individuales, ya sea para borrar, editar o insertarlos. Se resolvió utilizando un JSONObject conformado por varios JSONArrays, ya que estos si están específicamente ordenados.

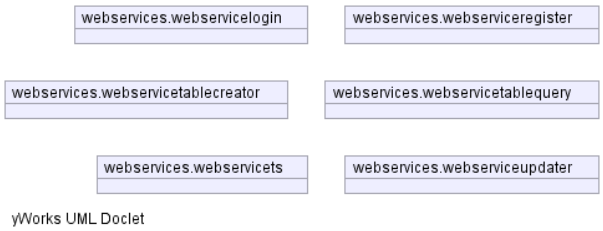
Documentación UML

WEB



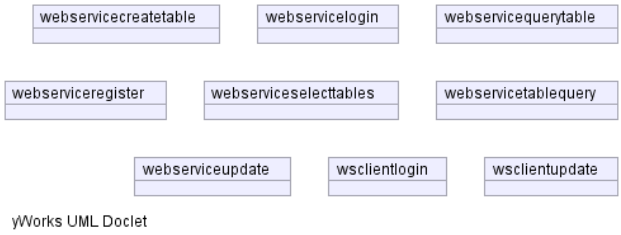


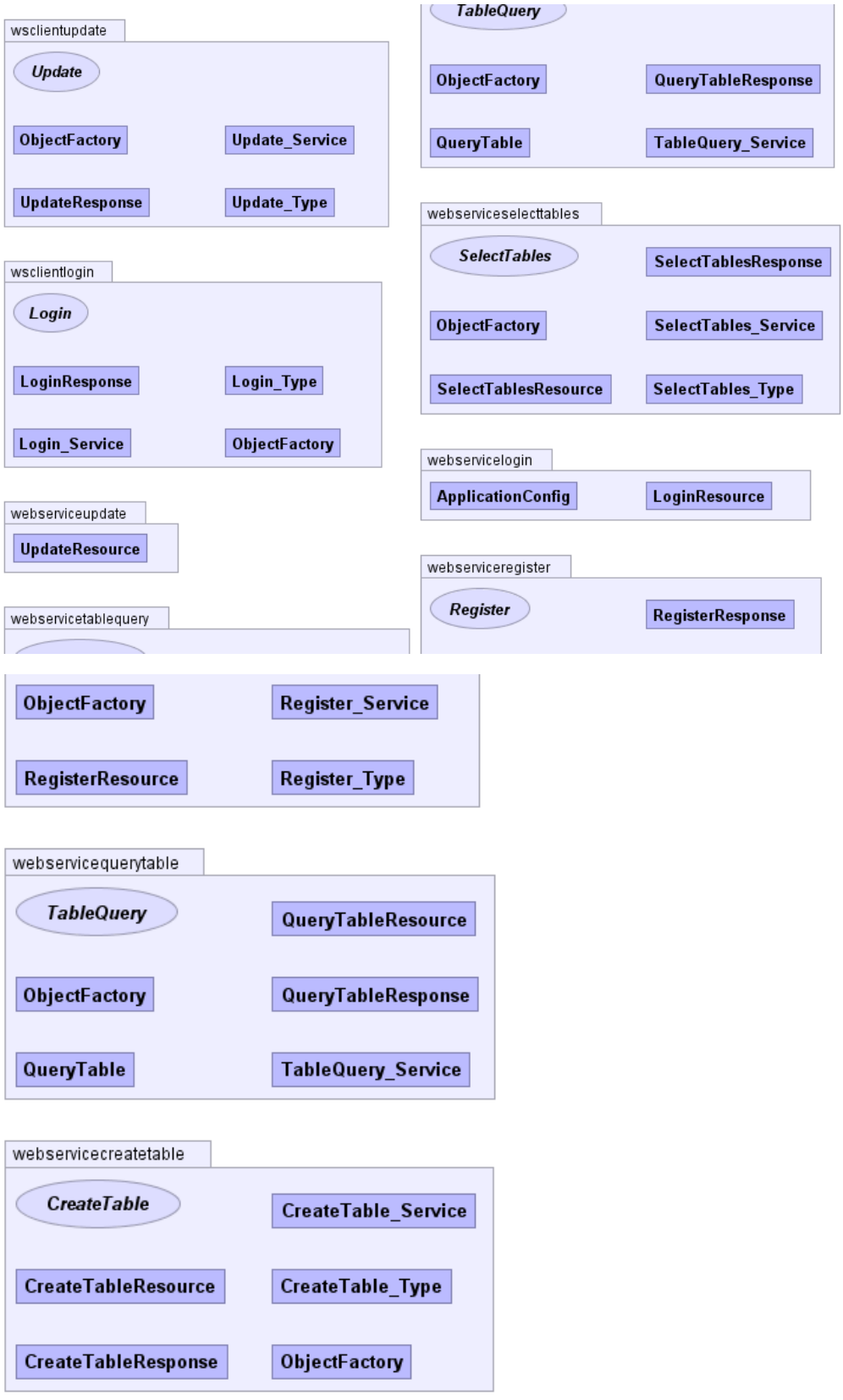
SOAP




webservices.webserviceupdater
Update
webservices.webservicets
SelectTables
webservices.webservicetablequery
TableQuery
webservices.webservicetablecreator
CreateTable
webservices.webserviceregister
Register
webservices.webservicelogin
Login
yWorks UML Doclet

REST


webservicecreatetable
webservicelogin
webservicequerytable
webserviceregister
webserviceselecttables
webservicetablequery
webserviceupdate
wsclientlogin
wsclientupdate
yWorks UML Doclet



yWorks UML Doclet

Lista de bugs conocidos

1. Al momento de querer insertar o borrar entradas, los parámetros no son enviados correctamente al servlet.

Conclusión

En conclusión, el proyecto sirvió como una buena lección del desarrollo de aplicaciones a través de la web, a pesar de que casi todo lo que se programó se puede autogenerar de forma muy simple utilizando APIs populares y de gran utilización en la industria.