

Data Web Wizard

Andrea Marín Alarcón
Luis Felipe Landa Lizarralde

May 2019

1 Introducción

La aplicación de Data Web Wizard tiene como propósito ser un manejador de bases de datos en línea. Los usuarios son capaces de crear una cuenta y, con dicha cuenta, una base de datos personal. Data Web Wizard se encarga de generar la base de datos en Apache Derby de manera transparente para el usuario. Esta base de datos solo se puede acceder por el usuario ya que se utiliza un manejo de sesión con HTTP.

Una vez que el usuario cuenta con una base de datos, es capaz de modificarla agregando tablas. Las tablas no tienen límite en cuanto al número de campos que pueden tener y pueden ser de tipo *VARCHAR*, *INT*, y *boolean*. Los campos que sean de tipo *VARCHAR* también pueden determinar la longitud de las cadenas que aceptan.

Las tablas también se pueden alterar: agregando, eliminando y modificando tuplas. En una sola interfaz, el usuario puede agregar entradas a una tabla de su base de datos, actualizar los valores ingresados y por último eliminar la entrada. Para agregar la tupla a la tabla solo se deben llenar los campos de la tabla; para actualizar un dato se debe ingresar la llave primaria y el nuevo valor del campo; para eliminar una entrada solo es necesario ingresar la llave primaria.

Data Web Wizard también le otorga al usuario la capacidad de consultar las tablas creadas. En una interfaz separada de la de creación y modificación sea pueden observar los datos de una tabla. Se puede determinar cuántos elementos se quieren visualizar de forma simultánea y el sistema de paginación se actualiza de manera automática y asíncrona.

Una vez que el usuario haya hecho los cambios necesarios puede cerrar su sesión por medio del botón que dice "cerrar sesión"; con esto el usuario es redirigido a la página de registro. Una vez que se haya cerrado la sesión, el usuario es incapaz de volver a acceder a las páginas que consultan las tablas o las modifican.

2 Obstáculos

Como la funcionalidad de Data Web Wizard fue implementada con servicios web, la interacción con estos fue la dificultad principal. Los servicios web deben ser independientes de quien los consume, por ende no pueden regresar objetos específicos a una plataforma e.g. Java. Inicialmente buscamos que nuestros servicios SOAP regresaran objetos, lo cual llevó a problemas de ejecución y compatibilidad con los servicios RESTful. La problemática se resolvió haciendo que los servicios SOAP únicamente regresaran tipos de datos primitivos e.g. *boolean* y *string*.

Otro problema se presentó cuando fue necesario cargar una página con datos precargados. Al principio intentamos implementar una función en JavaScript que llamara al servicio web que otorgaba la información y ejecutarla en un evento onload de HTML. Sin embargo, vimos que esto no daba resultados y se volvió necesario incluir servlets en la implementación de la funcionalidad de Data Web Wizard. El servlet se volvió el encargado de contactar al servicio web y cargar la página siguiente con los datos necesarios.

El uso de servlets trajo consigo otro problema. Cómo lograr que la información que tenía Java se compartiera con JavaScript. Puesto que información como el usuario se manejaba por medio de sesiones HTTP con Java, la interacción de ambas tecnologías era necesario. Para lograrlo fue necesario que la información conocida por Java se inyectara en la página por medio de elementos de input ocultos (hidden). Una vez que la información se encontraba en estos elementos del DOM, JavaScript era capaz de obtener los datos y utilizarlos.

3 Bugs

La aplicación solo cuenta con un solo bug conocido. En la página donde se agrega una nueva tabla a la base de datos, se le permite al usuario agregar campos a su tabla. Si el campo anterior era una llave primaria, al presionar el botón de "agrega campo", el nuevo campo agregado se toma como llave primaria y el usuario debe volver a presionar el botón que hace al campo anterior llave primaria.

4 Conclusiones

Los elementos que demostraron ser de mayor importancia en la implementación del proyecto fueron la validación de parámetros y el diseño de los servicios. Uno afecta al otro de manera directa y constante. Al diseñar un servicio web de cierta manera, se le obliga recibir parámetros con una estructura particular. Esto a su vez se traduce en que el *input* del usuario debe adquirir nuevas formas de forma transparente para el usuario. La validación de los parámetros para los servicios entonces se puede hacer por parte del usuario: prohibiendo que el usuario inserte valores que causen error, o limpiando el *input* para que sea aceptable para el servicio.

A su vez, la manera en que el usuario consume el servicio dictamina la manera que se deben implementar dichos servicios. Si el usuario desea generar una base de datos que contenga números o caracteres poco comunes en su nombre, nosotros debemos aceptar este *input* e implementar un servicio que provea la ejecución esperada. Si esto se vuelve imposible, los límites de la validación son los que imponen las restricciones de la implementación del servicio.

5 Diagramas

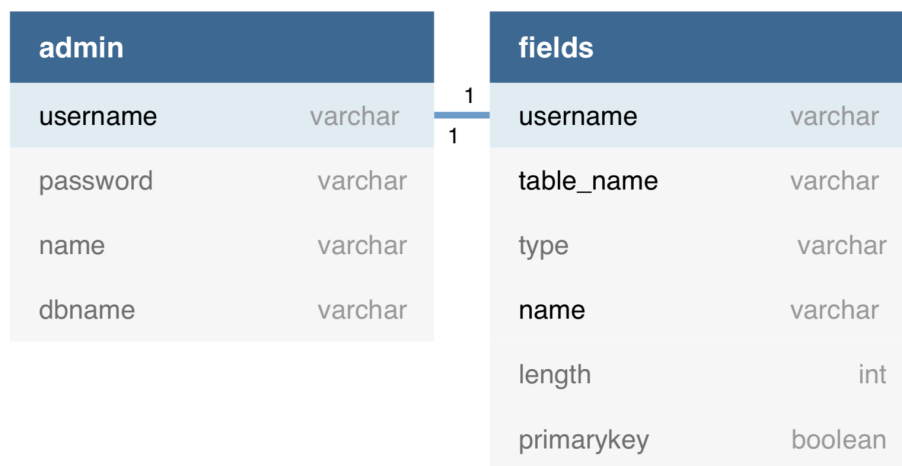


Figure 1: Diagrama de base de datos

La base de datos del administrador se compone de una tabla *admin* que contiene los usuarios con su nombre, contraseña, usuario y nombre de base de datos. Esta tabla de *admin* se relaciona con otra llamada *fields*. *Fields* permite al administrador conocer las tablas que corresponden a la base de datos del usuario y los campos dentro de cada una de estas tablas con su tipo, longitud y si son llave primaria. Esta base principal ayuda al momento de ejecutar queries específicas a cada usuario que suceden de forma transparente.

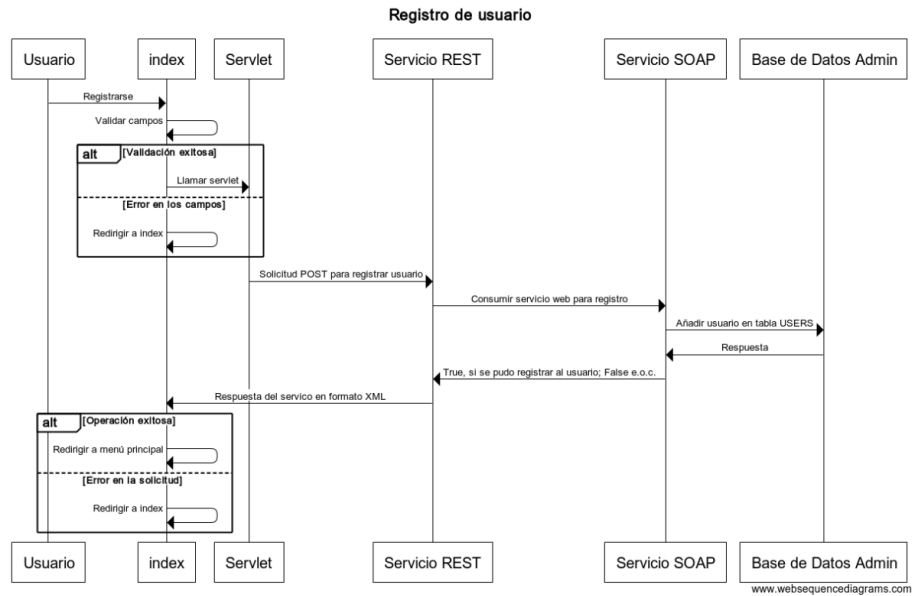


Figure 2: Diagrama de servicio de registro de usuario

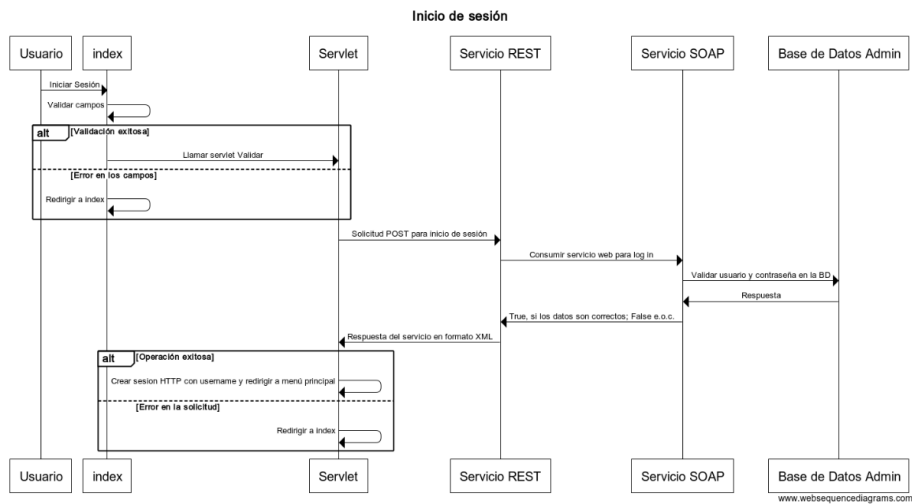


Figure 3: Diagrama de servicio para inicio de sesión

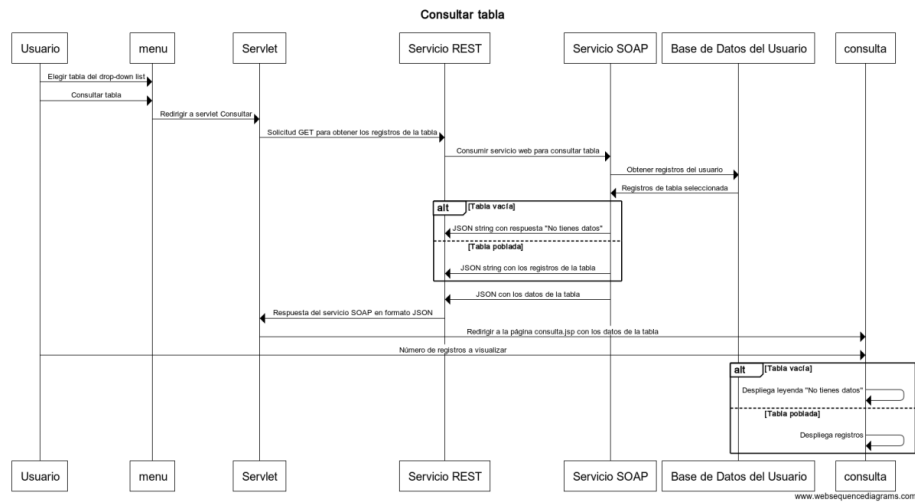


Figure 4: Diagrama de servicio para consultar una tabla

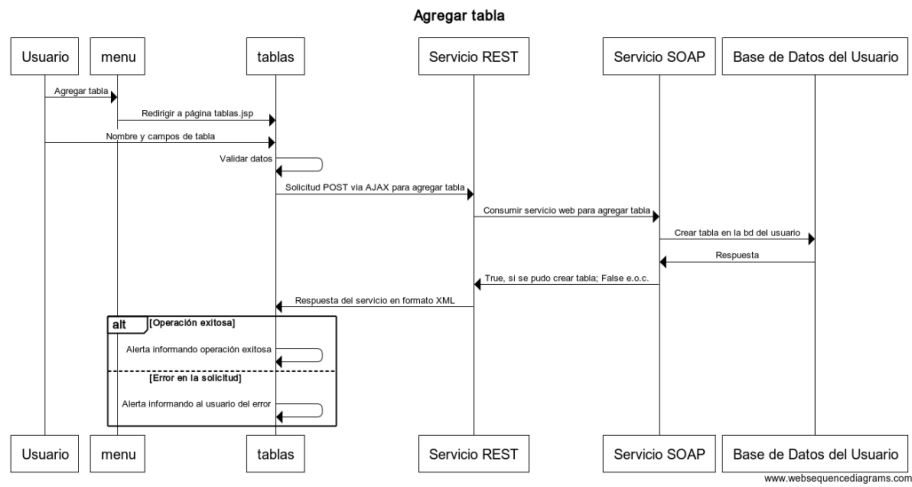


Figure 5: Diagrama de servicio agregar una tabla

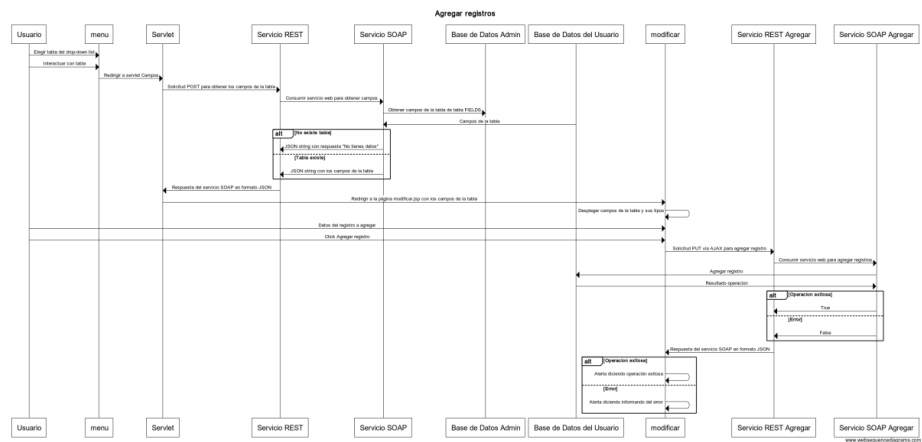


Figure 6: Diagrama de servicio para agregar registro

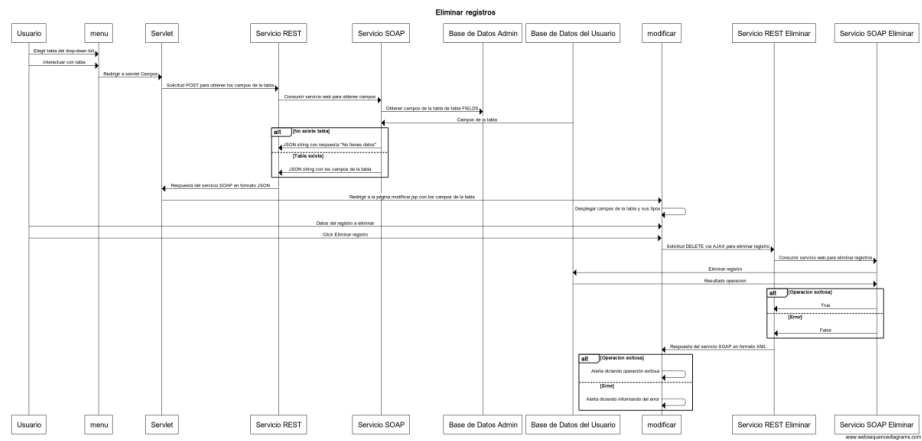


Figure 7: Diagrama de servicio para eliminar registro

