Curso

**ARQUITECTURA DE SOFTWARE**

Laboratorio 1

**APLICACIÓN WEB – CRUD**

Estudiantes

**JOAQUIN DAVID HERNÁNDEZ CÁRDENAS**

**JOHN BAIRON LONDOÑO CANO**

**JAIME LONDOÑO CIRO**

**FARLEY RÚA SUÁREZ**

**UNIVERSIDAD DE ANTIOQUIA**

**FACULTAD DE INGENIERÍA**

**DEPARTAMENTO DE SISTEMAS**

**MEDELLÍN – COLOMBIA**

**2014**

**CONTENIDO**

1. Objetivo del sistema.
2. Alcance del sistema.
3. Arquitectura del sistema.
   1. Capa de persistencia.
   2. Capa de negocio.
      1. Paquete del controlador.
      2. Paquete DAO.
      3. Paquete POJO.
   3. Capa de presentación.
4. Funcionamiento de la aplicación.
5. **Objetivo general.**

El objetivo general del sistema es simular el comportamiento de una Aplicación Web, que permita registrar, editar, eliminar o consultar la información de los estudiantes de una institución educativa universitaria. La idea con este sistema es implementar en el desarrollo del mismo, la estructura de arquitectura tipo CRUD, basada en capas que soporten la base de datos, la persistencia de la información, la lógica del negocio, la lógica de la aplicación y las vistas.

1. **Alcance del sistema.**

El sistema le permite al cliente realizar las siguientes transacciones en la base de datos:

* Registrar un estudiante en la base de datos con los siguientes campos: número de identificación, nombre, apellido y nivel académico.
* Actualizar la información de cualquier estudiante, accediendo a sus datos por medio del número de identificación.
* Eliminar toda la información pertinente de un estudiante, buscándolo por medio de su número de identificación.
* Consultar la información de cualquier estudiante, ingresando su número de documento de identificación.

1. **Arquitectura del sistema.**

Para llevar a cabo el objetivo de este sistema, se realizó la arquitectura de éste, estructurándolo en las siguientes capas:

* 1. **Capa de persistencia:**

Se refiere a la capa que ofrece a las demás, el servicio de persistencia de datos o información, y la recuperación posterior de la misma. En esta capa se utilizó la base de datos extraída que tiene Java, la cual permite tener la persistencia de los datos.

* 1. **Capa del negocio:**

La capa de negocio contiene la lógica principal de procesamiento de datos dentro de nuestra aplicación Web. Se comunica con la capa de presentación para obtener los datos ingresados por el cliente y presentar la información resultante, así como la capa de acceso a la base de datos para almacenar ésta información allí.

Esta capa contiene los siguientes paquetes:

* + 1. **Paquete del controlador:**

Este paquete contiene una clase llamada *StudentServlet.java*, la cual se encarga de extraer las entradas del cliente y comunicárselas al paquete DAO para que este las almacene. Luego de obtener una respuesta de DAO, el controlador avisa a la capa de presentación el resultado del almacenamiento de los datos ingresados por el cliente.

* + 1. **Paquete DAO:**

En el paquete DAO están contenidas clases *StudentDAO.java* y *StudentDAOLocal.java*. Este paquete es el encargado de recibir la información transmitida por el controlador de los datos que se han ingresado, para éste hacer la conexión con la base de datos y hacer la respectiva operación solicitada por el cliente.

* + 1. **Paquete POJO:**

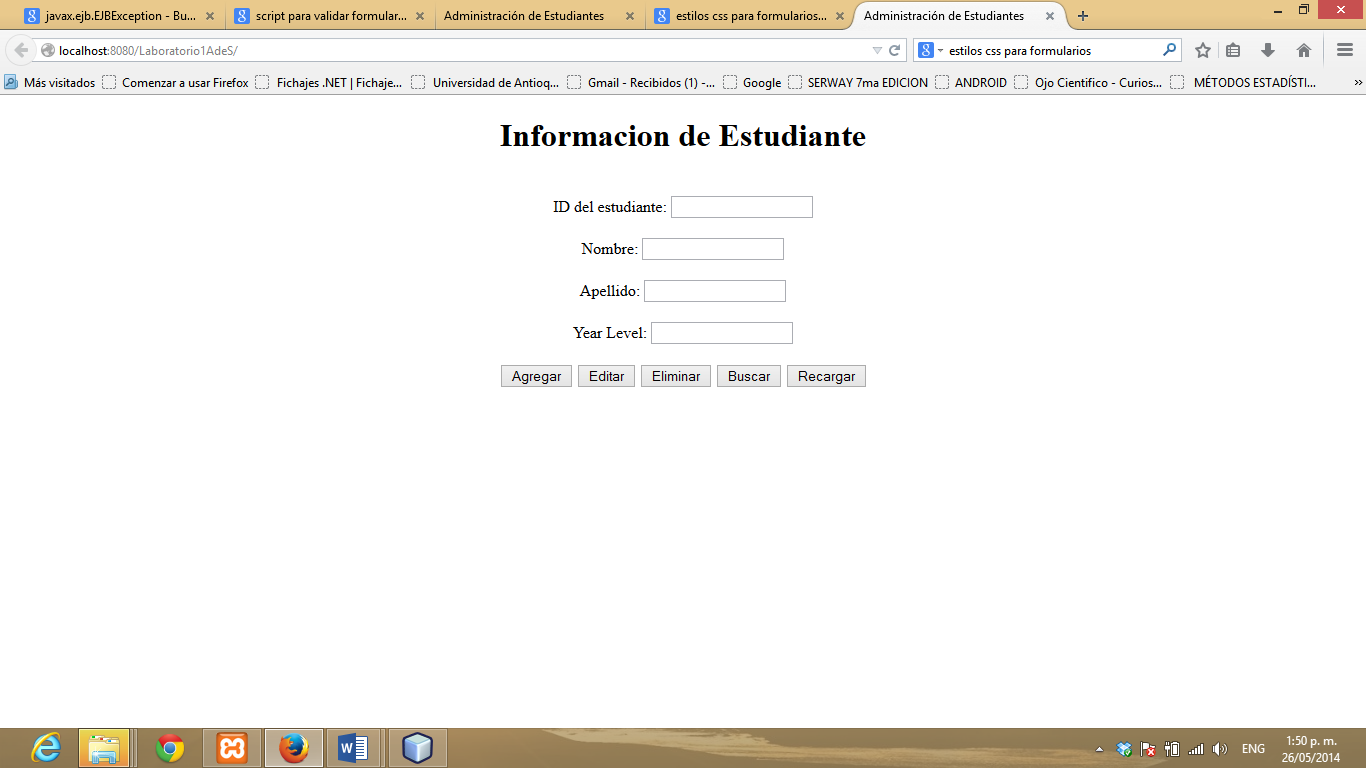
El paquete pojo contiene una clase llamada *Student.java*, ésta clase está encargada de crear el objeto que contiene los datos que están siendo ingresados por el cliente. Además, cada objeto es almacenado en la base de datos como un elemento de la tabla *Student*, manejando cada uno de sus campos en columnas.

* 1. **Capa de presentación:**

La capa de presentación es la que interactúa directamente con el cliente, y le permite a éste el ingreso y la realización de las transacciones de una forma organizada y amigable. En esta capa están contenidos los archivos *.jsp*, los cuales son códigos encargados de elaborar las vistas de la aplicación.

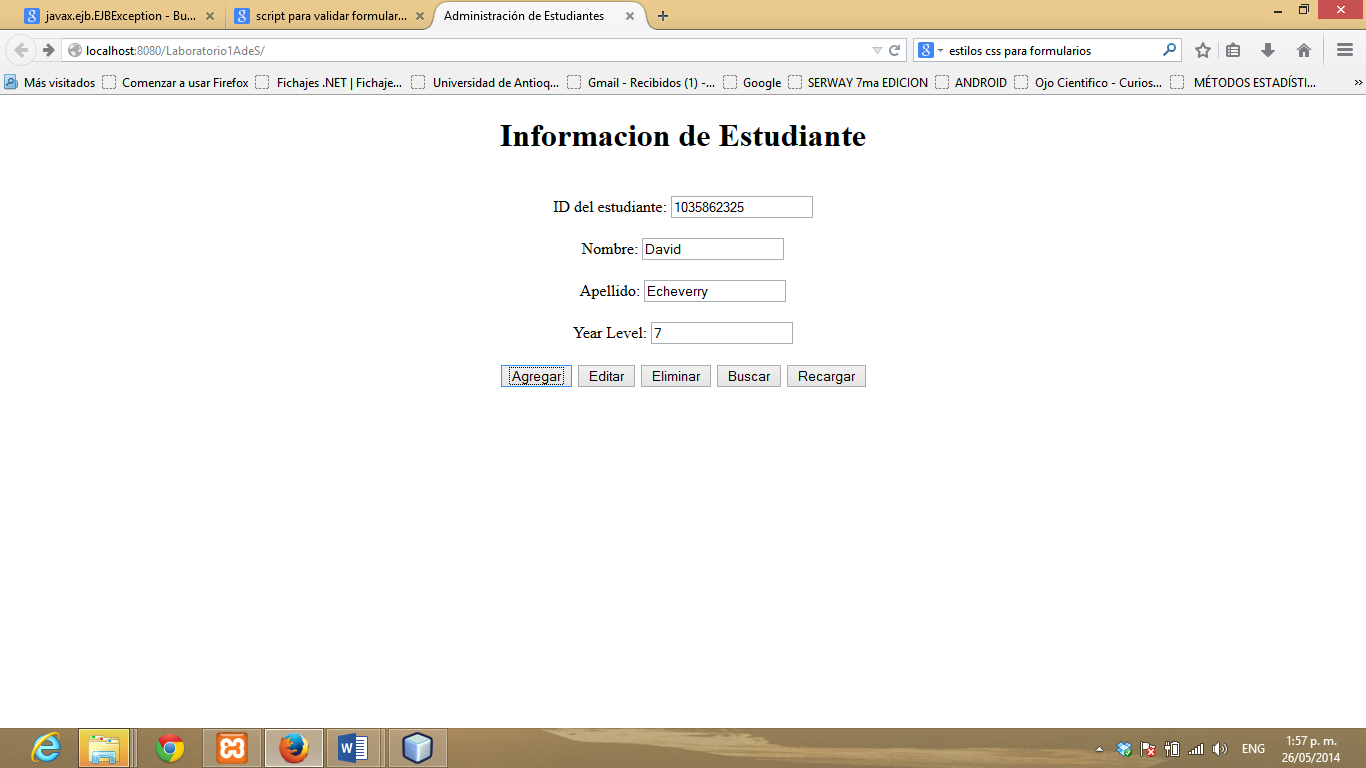
1. **Funcionamiento de la aplicación.**

Inicialmente al ejecutar la aplicación web, en pantalla se visualiza el formulario que se visualiza en la *Figura 1.*



**Figura1.**

Si el cliente desea ingresar un estudiante nuevo, debe llenar completamente cada uno de los campos del formulario y posteriormente clic en el botón **Agregar** para insertar la información en la base de datos. *Ver Figura 2.*



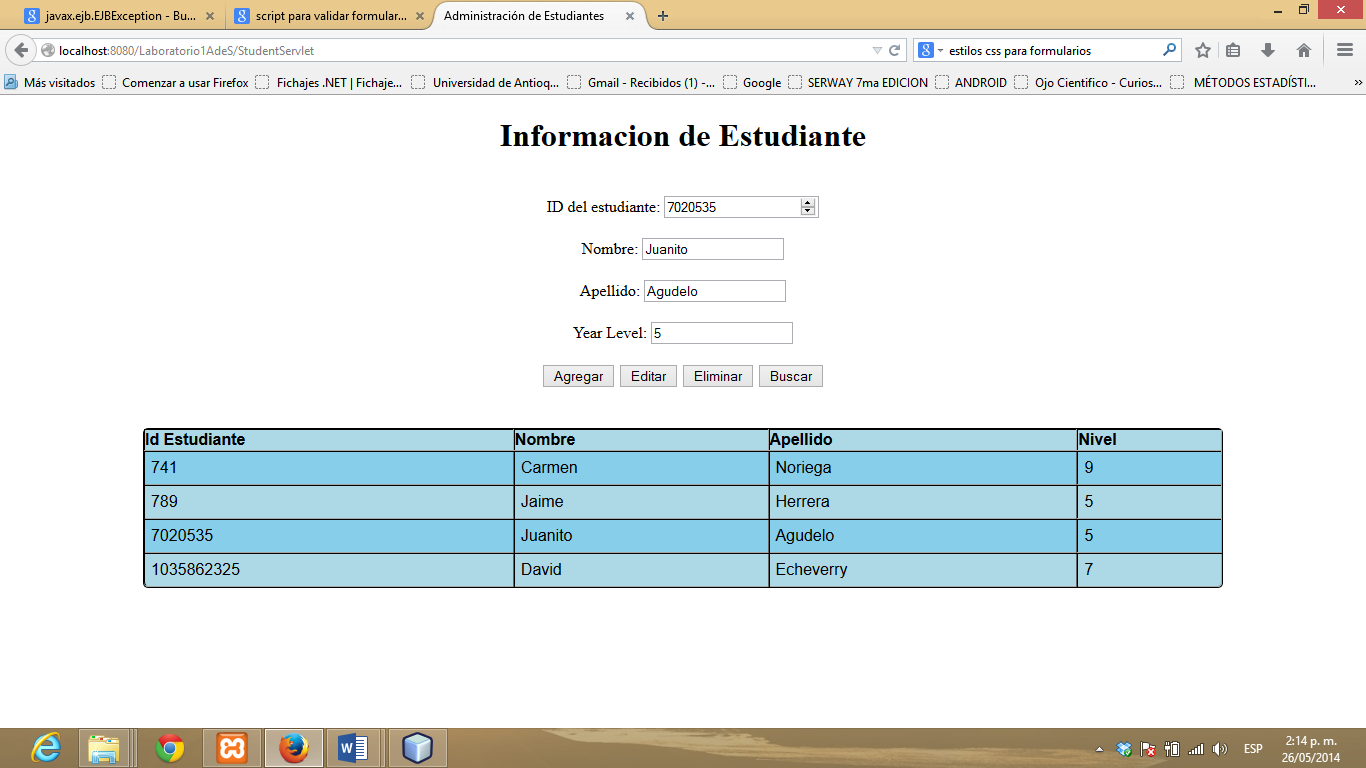
**Figura 2.**

Al ingresar toda la información correspondiente a un estudiante que no ha sido registrado en la base de datos, inmediatamente, la aplicación mostrará en pantalla toda la información contenida en la base de datos por medio de una tabla. *Ver Figura 3.*



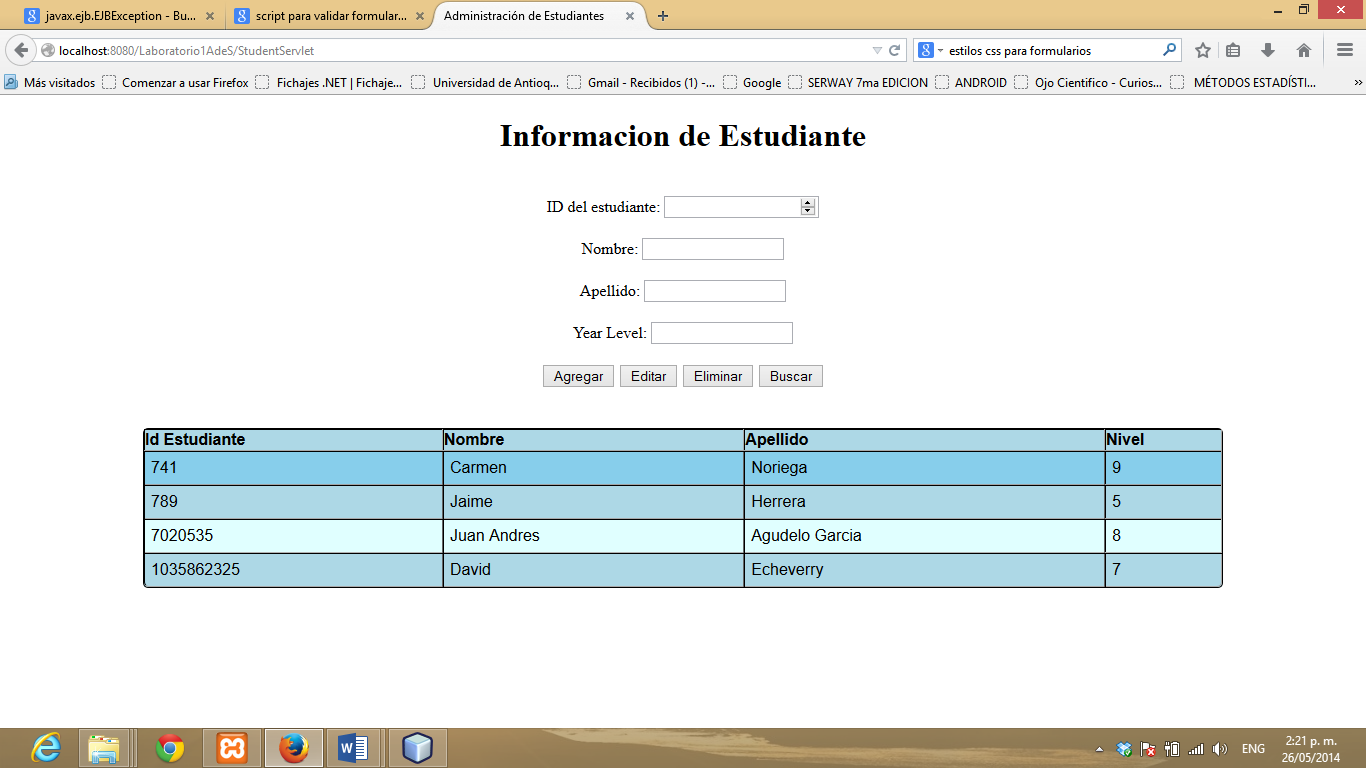
**Figura 3.**

Ahora, si el cliente desea consultar toda la información de algún estudiante que ha sido registrado anteriormente, tan solo debe ingresar el número de identificación en su respectivo campo y dar clic en el botón **Buscar**. *Ver Figura 4*.



**Figura 4.**

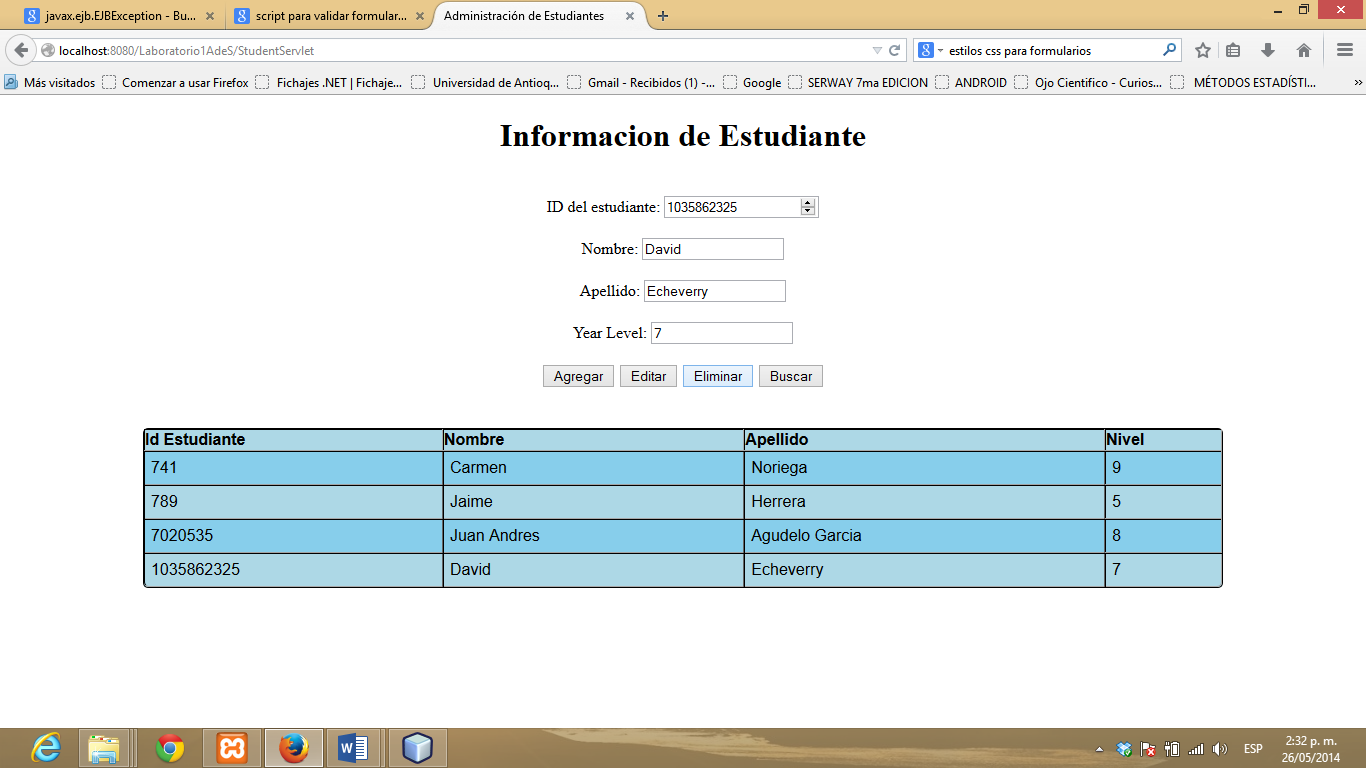
Luego de haber hecho exitosamente la búsqueda del estudiante, el cliente puede editar cada uno de los datos del mismo por medio del botón **Editar**. Como ejemplo práctico, editaremos la información del estudiante ‘*Juanito Agudelo’*. *Ver Figura 5*.



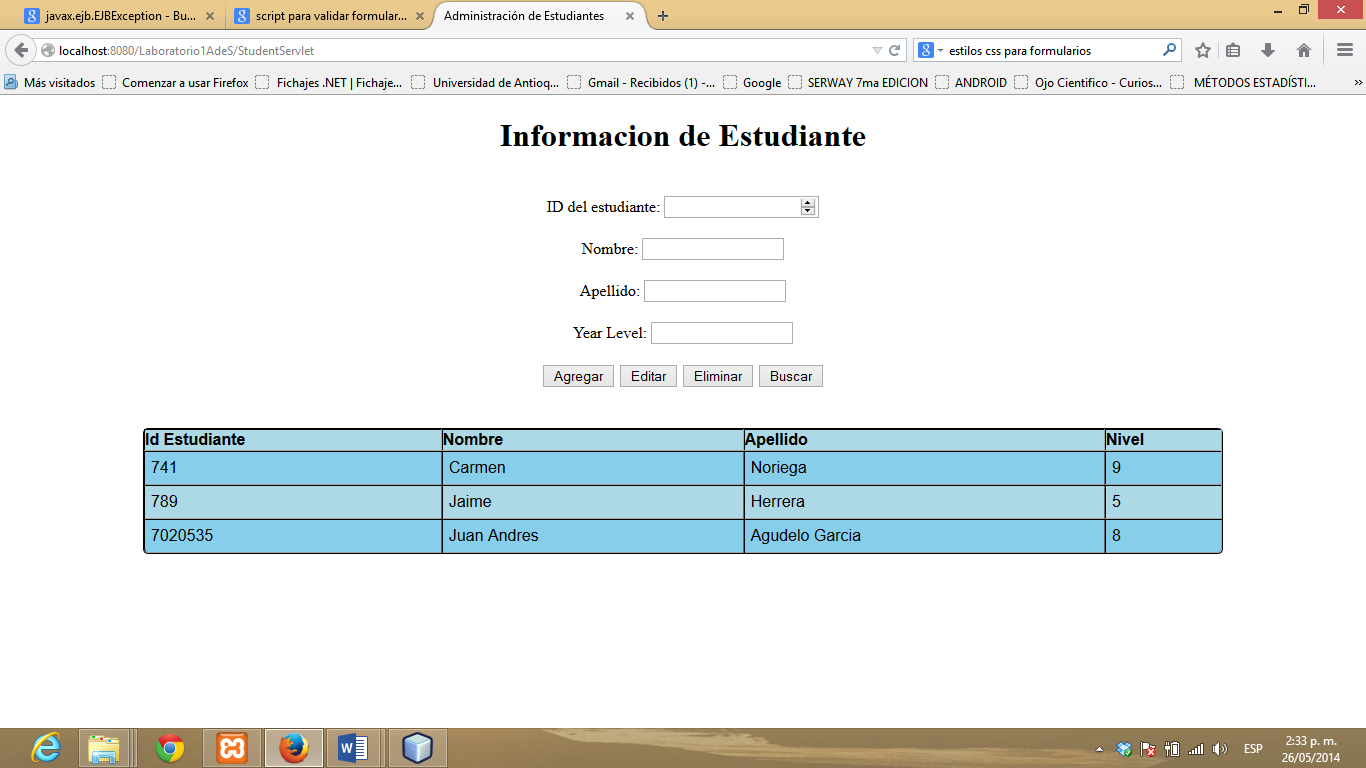
**Figura 5.**

En la *Figura 4*, se observa que la información del estudiante con ID ‘*7020535*’, la cual pertenece a ‘*Juanito Agudelo’*, han sido modificados los campos de su nombre, su apellido y su nivel como se visualiza en la *Figura 5*, en donde su nombre ha sido modificado por *‘Juan Andres’*, su apellido por ‘*Agudelo Garcia’* y su nivel por el ‘7’.

Finalmente, así mismo después de realizar la búsqueda de un estudiante y editarla, también el cliente tiene la opción para eliminar por completo toda la información del estudiante buscado. *Ver Figura 6 y Figura 7*.



**Figura 6.**



**Figura 7.**

Evidentemente si comparamos la *Figura 6* y la *Figura 7*, observamos que toda la información del estudiante con ID ‘*1035862325*’, la cual pertenece al estudiante *‘David Echeverry’*, ha sido eliminada luego de dar clic en el botón **Eliminar**.