30 de mayo de 2018

Danny Xie

Instituto tecnologico de costa rica

Documentación fase ii

Proyecto 2

**Índice**

Introducción ……………………………………….……………………………………………………….. 2

Descripción del problema…………………….………………………………………………………… 2

Documentación del Web service……………………………………………………………………… 4

Documentación del app………………………………………………………………………………….. 5

Conclusiones……………………………..………………………………………………………………….. 5

**Descripción del problema**

Actualmente la universidad Tecnológica usa un sistema de base de datos relacional para realizar sus operaciones cotidianas. De la cual este posee registros acerca de los campus, facultades, personal, edificios, grados, entre otros registros de diferentes entidades. La oficina de Servicios de Tecnología de Información (STI), es el encargado de las bases de datos de la universidad, este nos describe el problema de utilizar base de datos modelos relacional-objeto usando el motor de Oracle.

Se describe que la universidad posee ocho campus en todo el país. De la cual la entidad Campus tiene relación con las entidades Edificio y Persona, además no existe relación directa sobre las entidades cursos de diferentes grados y las facultades.

La universidad posee cinco facultades, cada facultad está compuesta de distintos departamentos, escuelas y centros de investigación. Este se implementa con objetos. Usar tablas anidadas en estas entidades descritas anteriormente.

Los datos School\_Prof y Dept\_Prof de las entidades facultad, escuela, departamento y centros de investigación son entidades. Además, el atributo Unit de centro de investigación son varias unidades.

Cada campus posee varios edificios, cada uno están compuestos por distintas habitaciones que esta puede ser oficinas, salas de clase y laboratorios. La facultad puede ocupar varios edificios. Un edificio sólo puede ser asignado a una facultad.

Cada facultad ofrece a los estudiantes una cantidad de grados, cada grado particular sólo puede ser asignado por una facultad particular.

Los datos del personal, este se puede categorizar en dos tipos: estudiantes y staff. El staff se puede categorizar en administrador, técnico, profesor, tutor. Un lecturer puede ser categorizado como associate lecturer o senior lecturer. Además, un tutor puede ser un estudiante.

Cada estudiante tendrá una carrera, este puede llevar más de un degree. Además, el estudiante está enlazado con una materia, además la materia está relacionada con un profesor que es el encargado de la materia.

Para la segunda fase del problema se va a desarrollar una aplicación web y un web service RESTful que será consumida por la aplicación web.

**Documentación de la implementación de la aplicación**

La aplicación se implementó en una plataforma web en el lenguaje C# usando la arquitectura Model View Controller.

En la siguiente sesión de la aplicación está el registro del personal de la universidad. En esta sesión permite el registro del personal tales como el registro de estudiantes, profesores, empleados, estudiantes tutores, empleados tutores, técnicos, administradores y profesores asociados.

A continuación, en la figura 1 se muestra la pantalla principal de registro de personal de la universidad. En la barra superior se muestra el personal que se podrá registrar en el sistema, como se describió anteriormente. Cada sección este compuesto por 4 módulos, para realizar operaciones sobre cada registro, por ejemplo, visualizar, eliminar, actualizar y agregar registros.

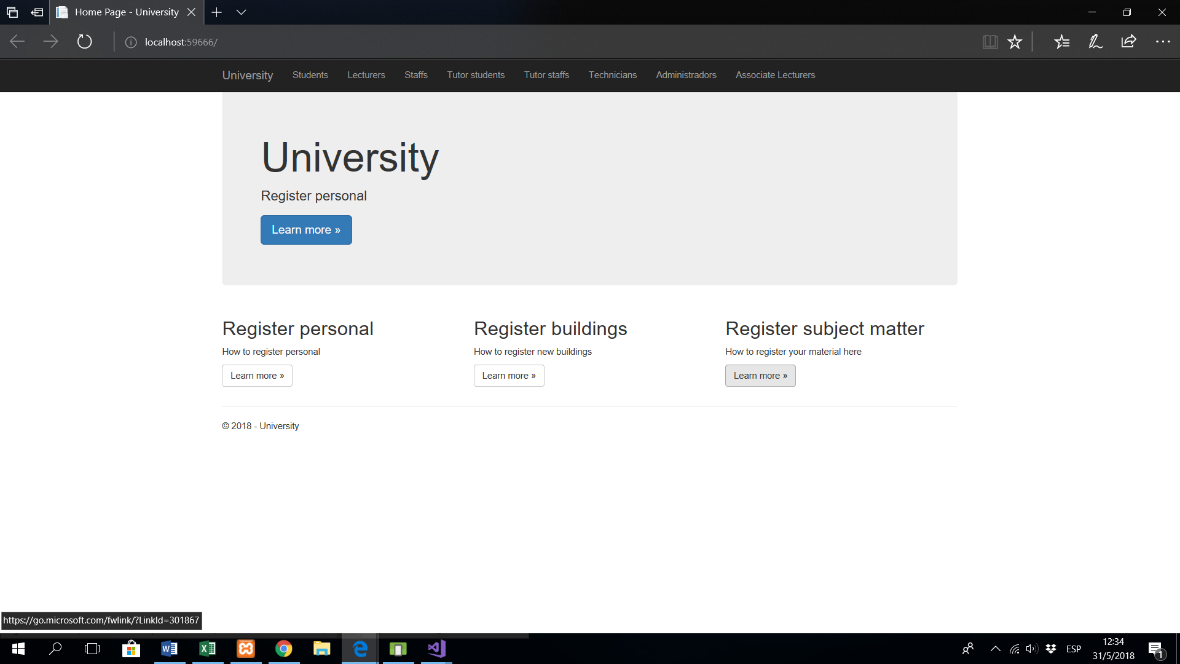


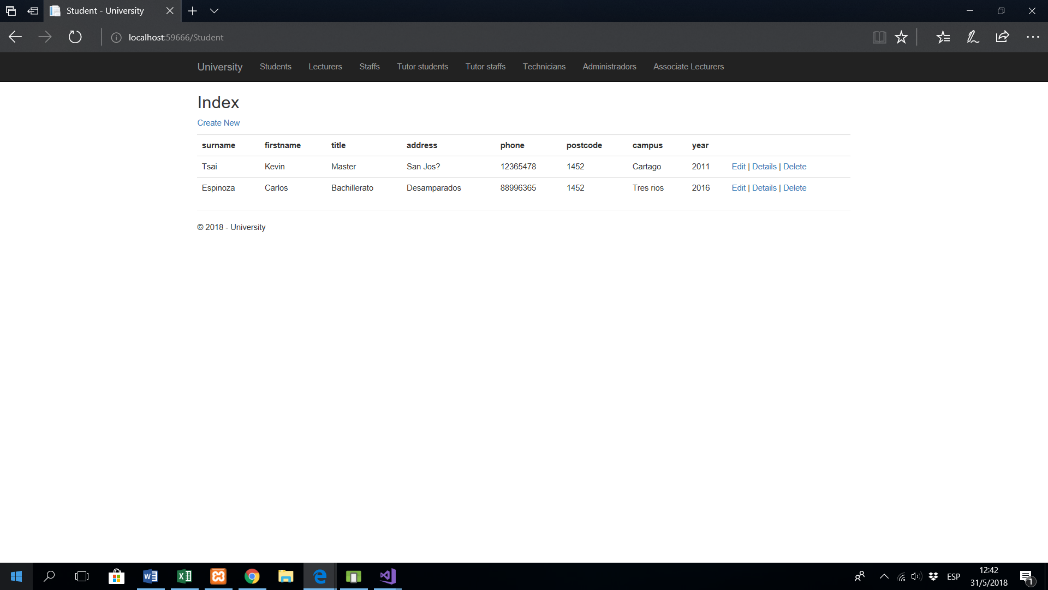
Figura 1

A continuación, se describirá cada sección de registro del personal:

* Sección Student

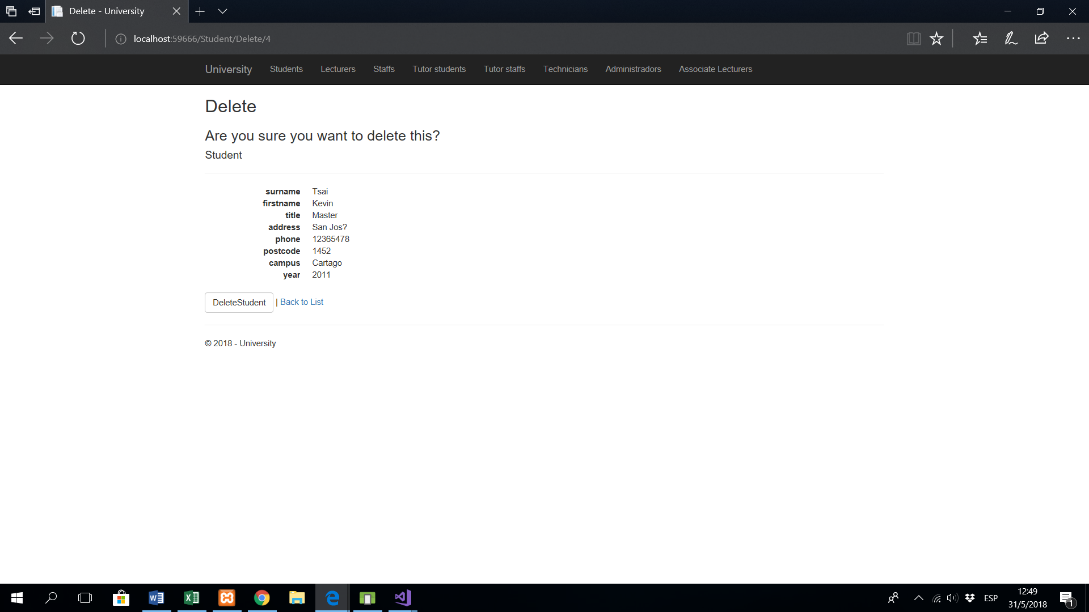
1. **Visualizar registros**

En esta sección se podrá visualizar los registros de los estudiantes guardados en el sistema.



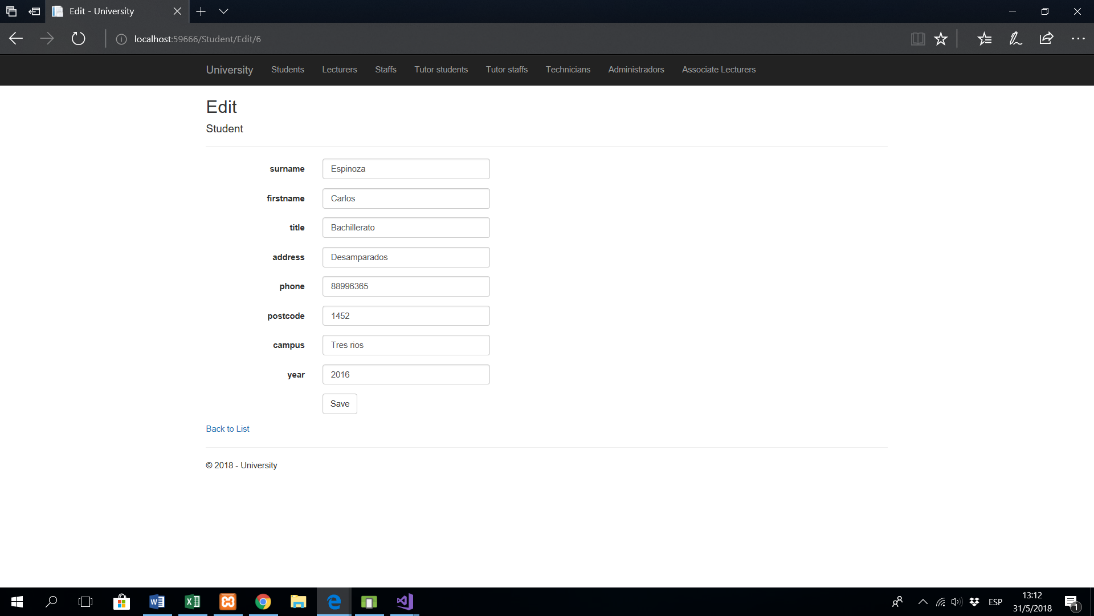
1. **Eliminar registro**

Después de hacer click en un registro en la opción de delete se le mostrrá la siguiente vista para eliminar el registro seleccionado. Si está seguro de eliminar el registro debe tocar el botón *DeleteStudent* o sino *back To List* para regresar a la sección anterior.



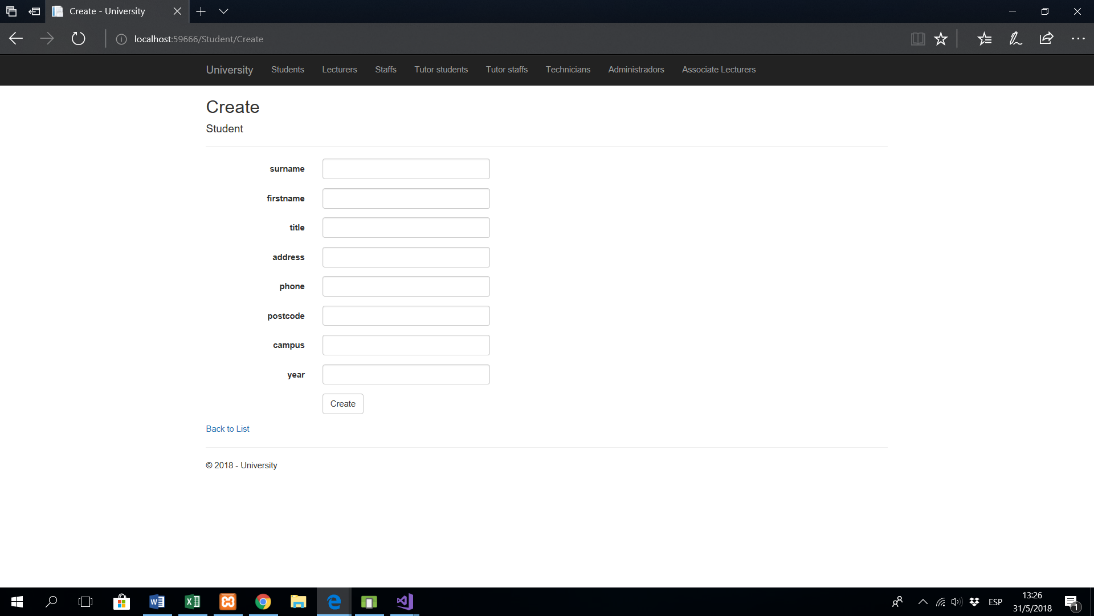
1. **Actualizar registro**

Al tocar el botón de editar este podrá actualizar los datos del registro en el sistema. Este podrá actualizar los registros, después de actualizarlos deberá tocar el botón de *save* para guardar los cambios en el sistema.



1. **Registrar un registro**

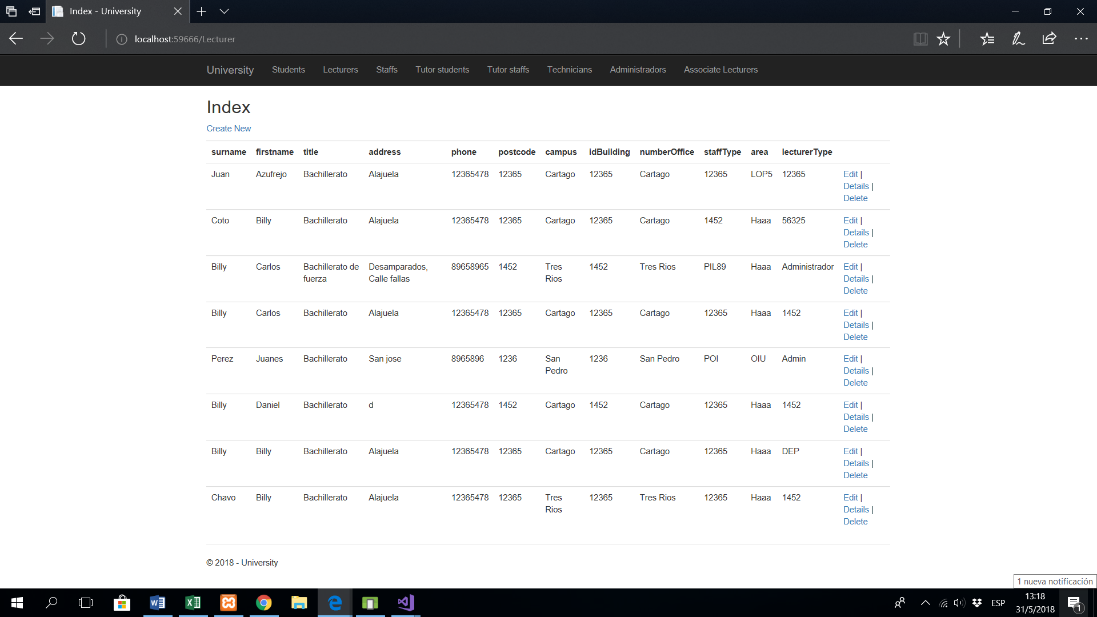
Al tocar *Create New*, se le mostrará los campos que debe llenar para poder registrar el registro en el sistema. AL terminar de llenar los datos deberá tocar el botón de *Create* para poder registrar el nuevo registro en el sistema.



* Sección Lecturer

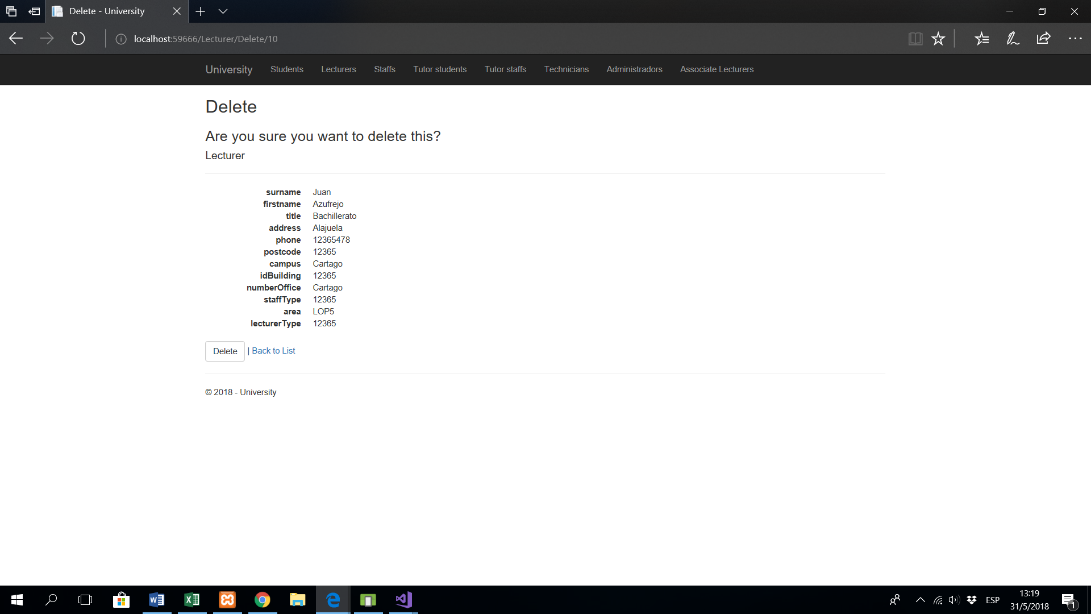
1. **Visualizar registros**

En esta sección se podrá visualizar los registros de los profesores guardados en el sistema.

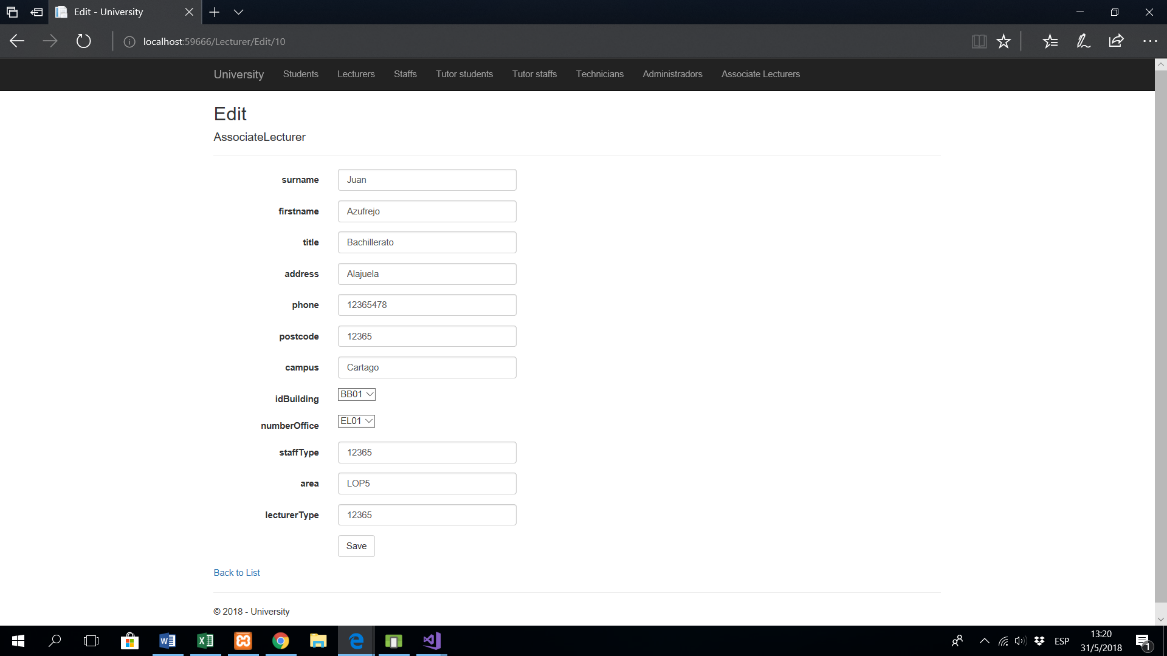


1. **Eliminar registro**

Después de hacer click en un registro en la opción de delete se le mostrrá la siguiente vista para eliminar el registro seleccionado. Si está seguro de eliminar el registro debe tocar el botón *Delete* o sino *back To List* para regresar a la sección anterior.

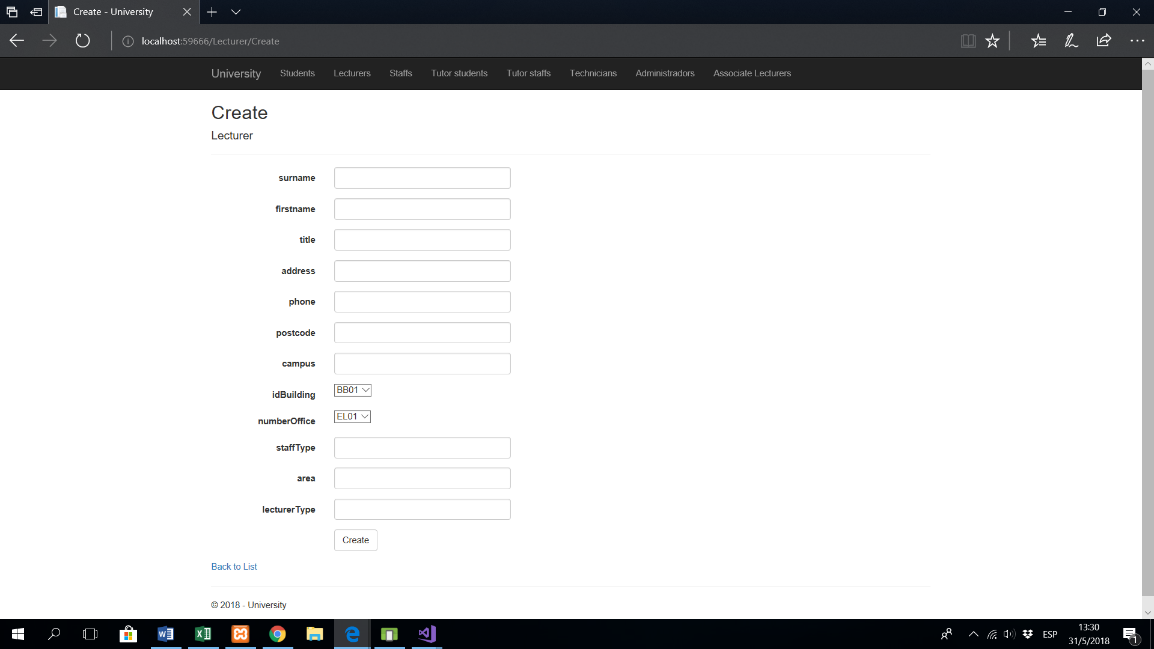


1. **Actualizar registro**

Al tocar el botón de editar este podrá actualizar los datos del registro en el sistema. Este podrá actualizar los registros, después de actualizarlos deberá tocar el botón de *save* para guardar los cambios en el sistema.

1. **Crear registro**

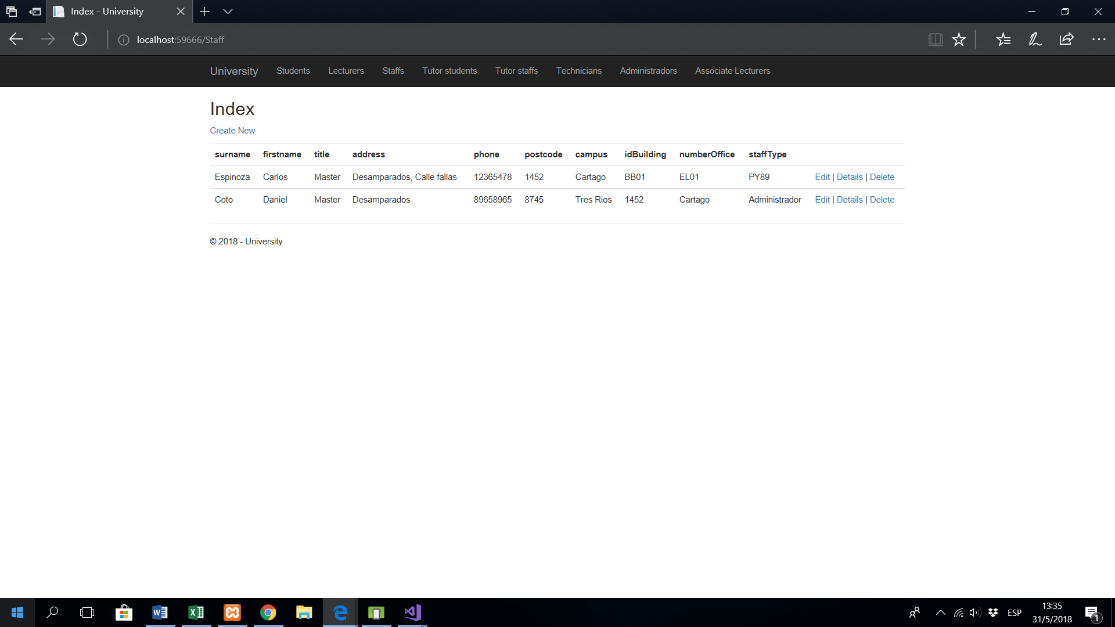
Al tocar *Create New*, se le mostrará los campos que debe llenar para poder registrar el registro en el sistema. AL terminar de llenar los datos deberá tocar el botón de *Create* para poder registrar el nuevo registro en el sistema.



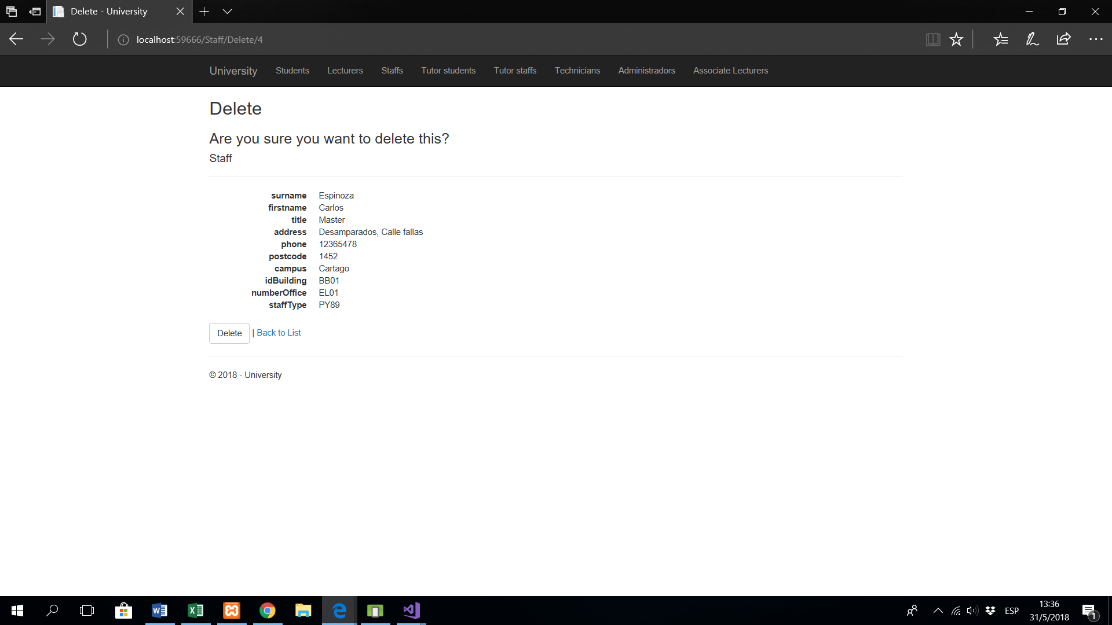
* Sección Staff

1. **Visualizar registros**

En esta sección se podrá visualizar los registros de los personal guardados en el sistema.

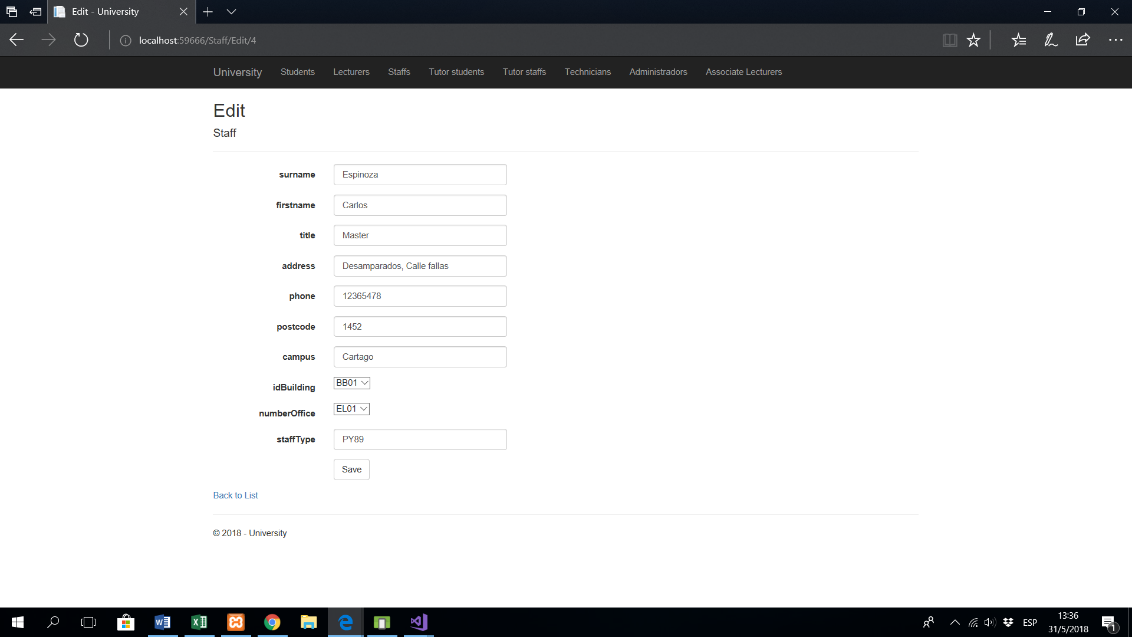


1. **Eliminar registro**

Después de hacer click en un registro en la opción de delete se le mostrrá la siguiente vista para eliminar el registro seleccionado. Si está seguro de eliminar el registro debe tocar el botón *Delete* o sino *back To List* para regresar a la sección anterior.

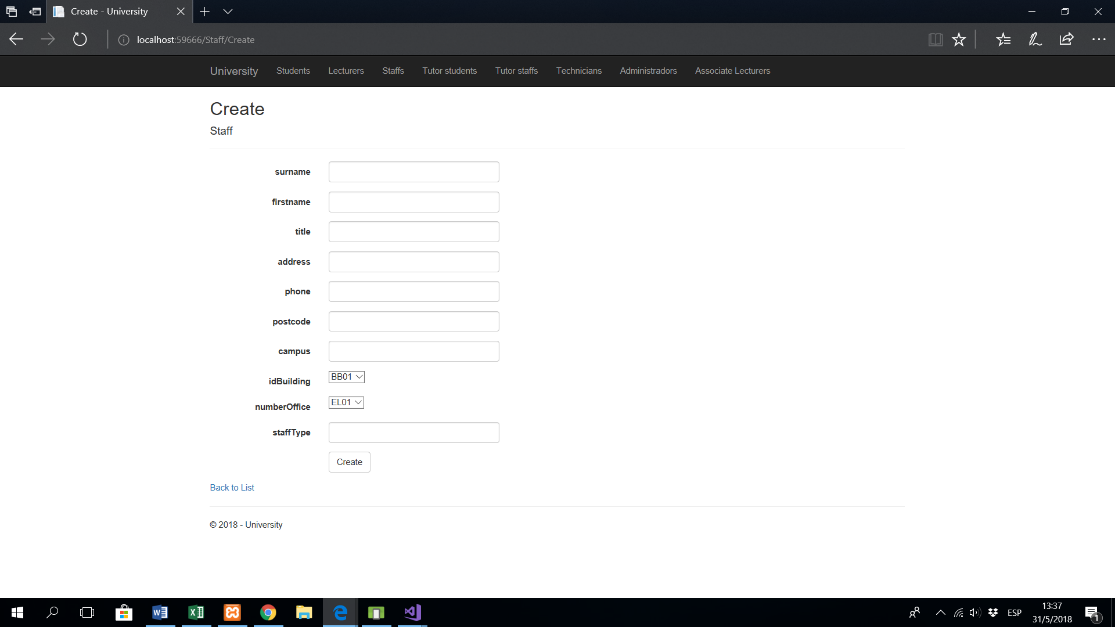
1. **Actualizar registro**

Al tocar el botón de editar este podrá actualizar los datos del registro en el sistema. Este podrá actualizar los registros, después de actualizarlos deberá tocar el botón de *save* para guardar los cambios en el sistema.



1. **Crear registro**

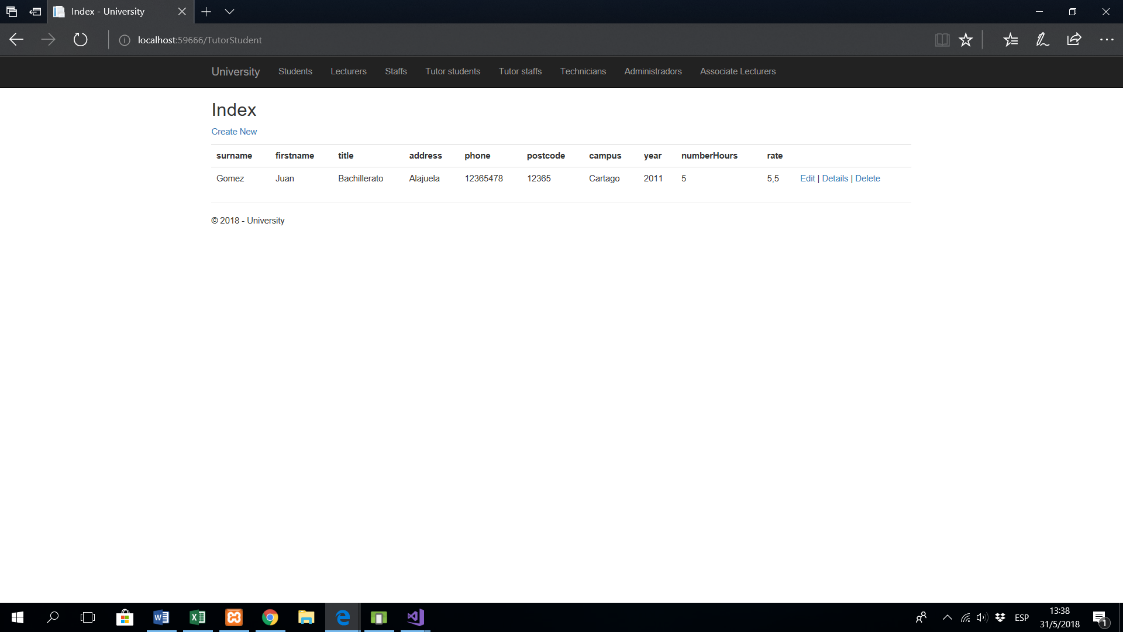
Al tocar *Create New*, se le mostrará los campos que debe llenar para poder registrar el registro en el sistema. AL terminar de llenar los datos deberá tocar el botón de *Create* para poder registrar el nuevo registro en el sistema.



* Sección Tutor Student

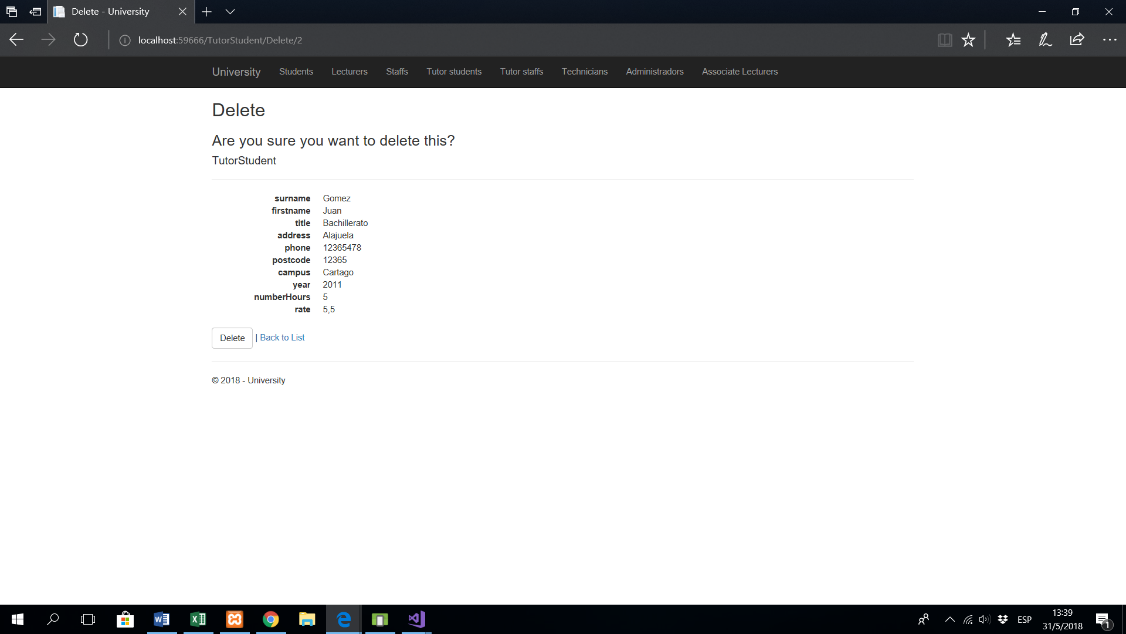
1. **Visualizar registros**

En esta sección se podrá visualizar los registros de los tutor student guardados en el sistema.



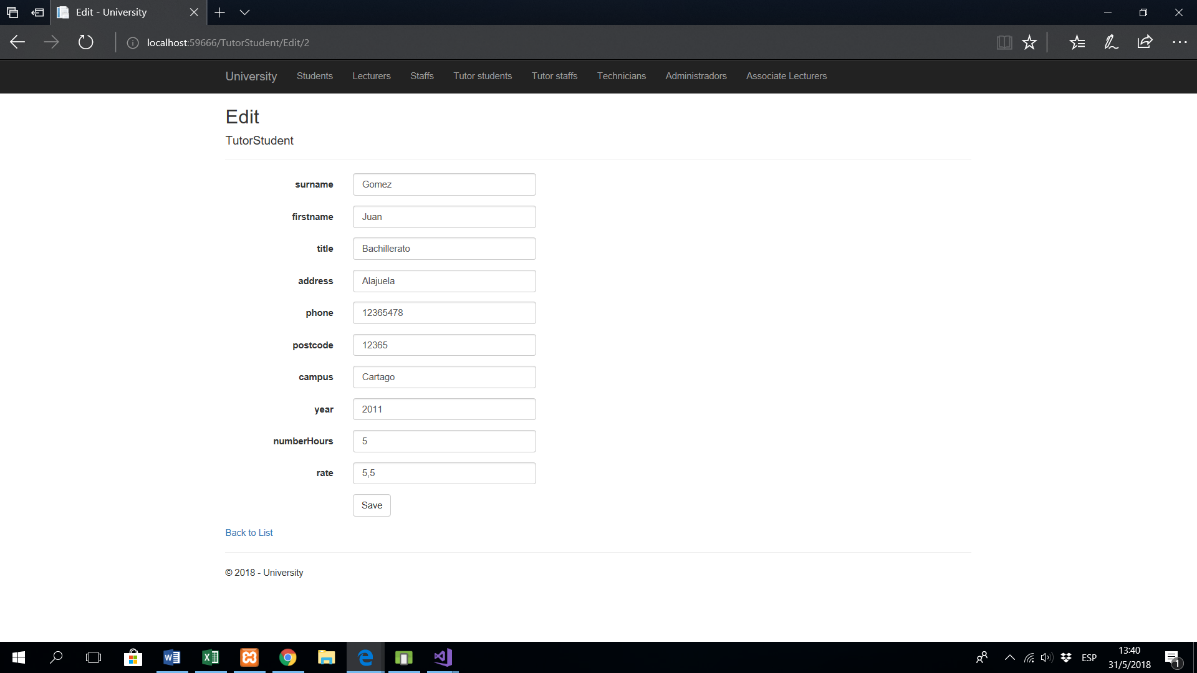
1. **Eliminar registro**

Después de hacer click en un registro en la opción de delete se le mostrará la siguiente vista para eliminar el registro seleccionado. Si está seguro de eliminar el registro debe tocar el botón *Delete* o sino *back To List* para regresar a la sección anterior.



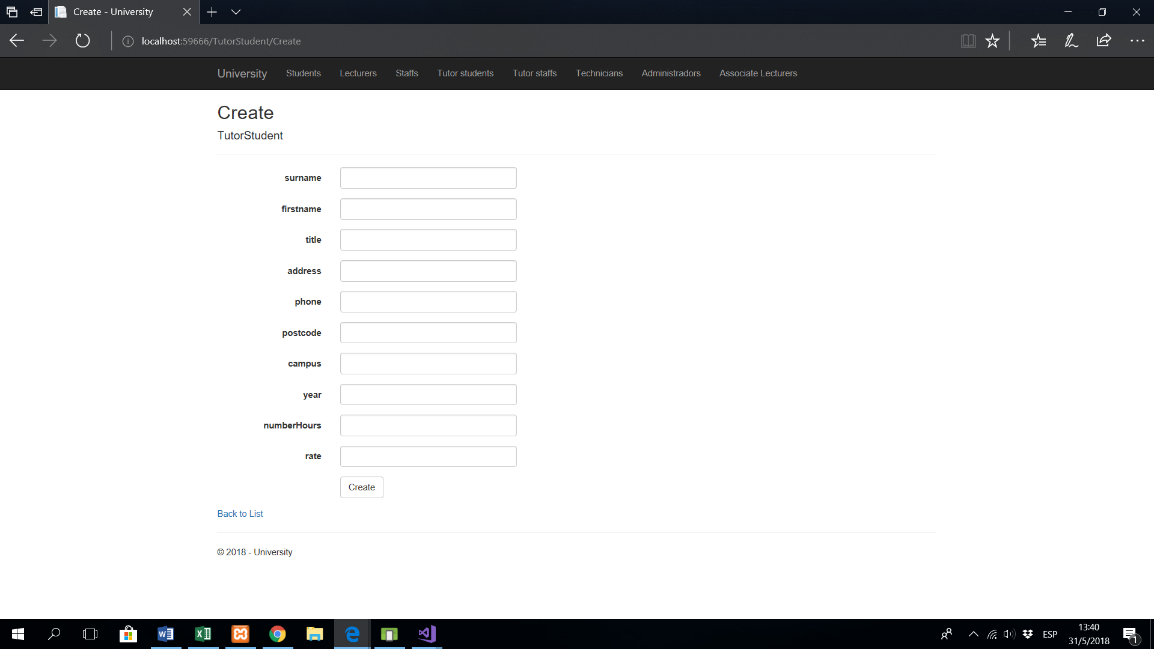
1. **Actualizar registro**

Al tocar el botón de editar este podrá actualizar los datos del registro en el sistema. Este podrá actualizar los registros, después de actualizarlos deberá tocar el botón de *save* para guardar los cambios en el sistema.



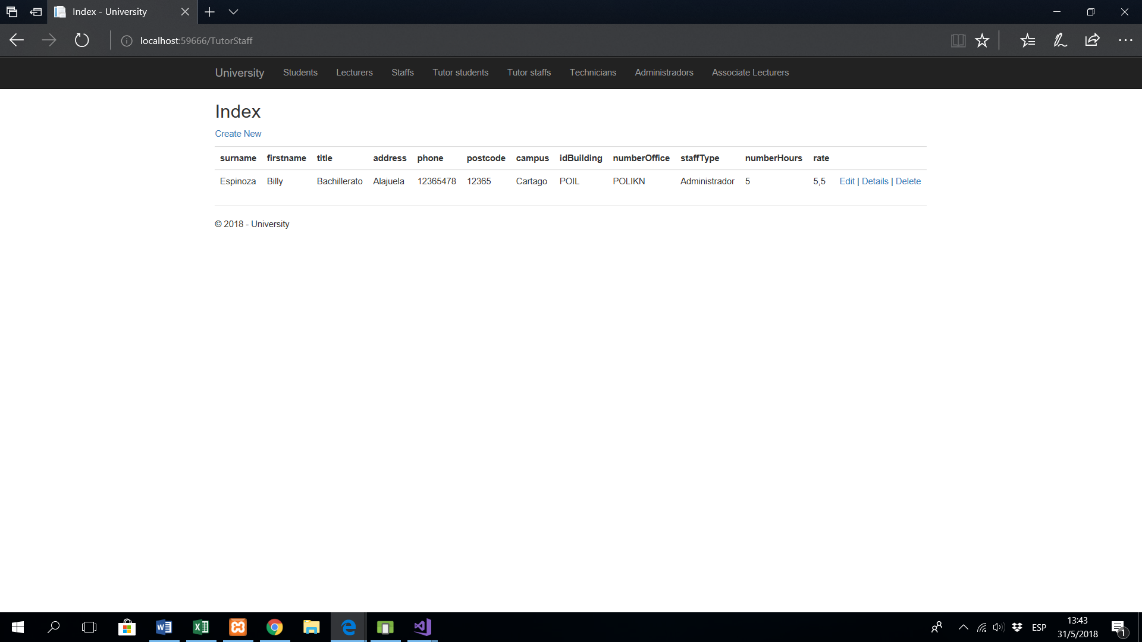
1. **Crear registro**

Al tocar *Create New*, se le mostrará los campos que debe llenar para poder registrar el registro en el sistema. AL terminar de llenar los datos deberá tocar el botón de *Create* para poder registrar el nuevo registro en el sistema.



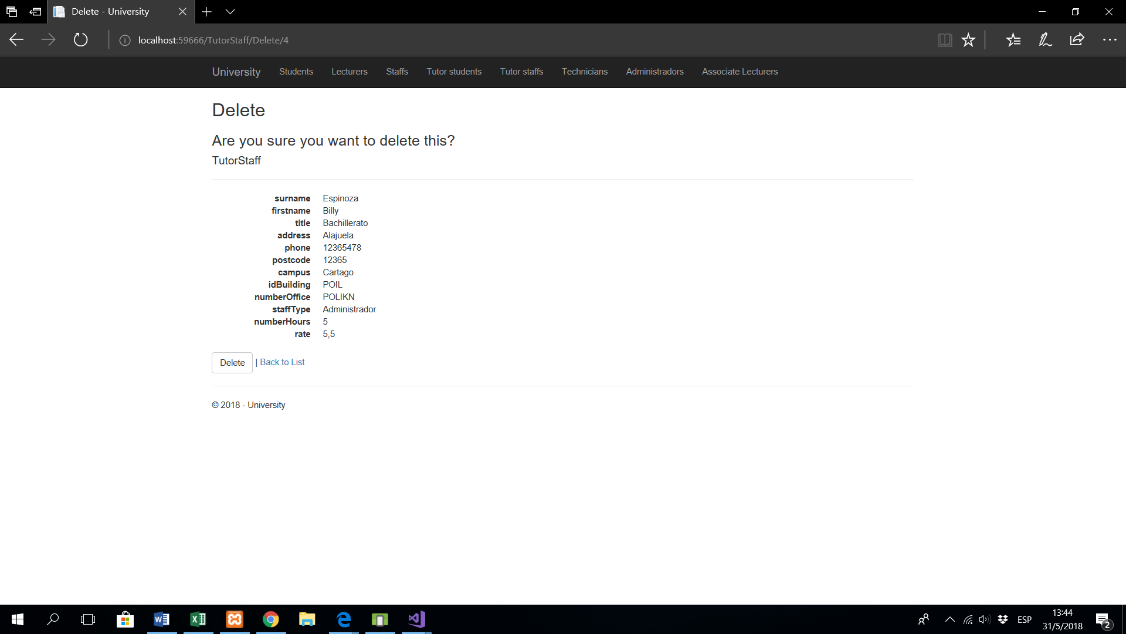
* Sección Tutor Staff

1. **Visualizar registros**

En esta sección se podrá visualizar los registros de los tutor staff guardados en el sistema.

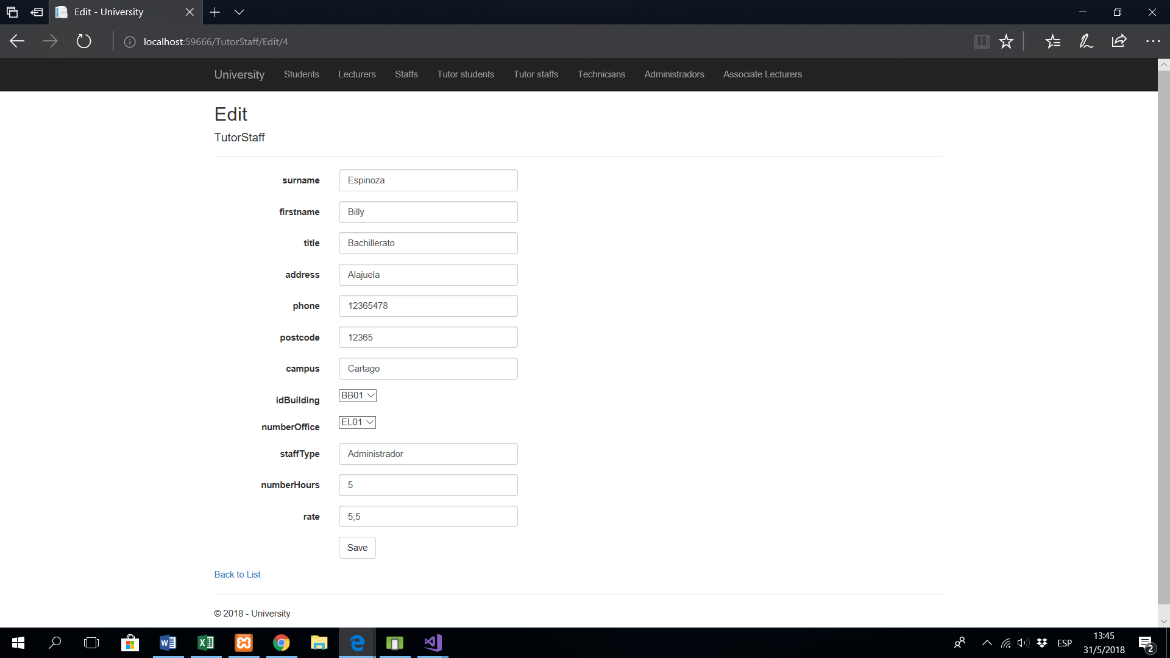
1. **Eliminar registro**

Después de hacer click en un registro en la opción de delete se le mostrará la siguiente vista para eliminar el registro seleccionado. Si está seguro de eliminar el registro debe tocar el botón *Delete* o sino *back To List* para regresar a la sección anterior.



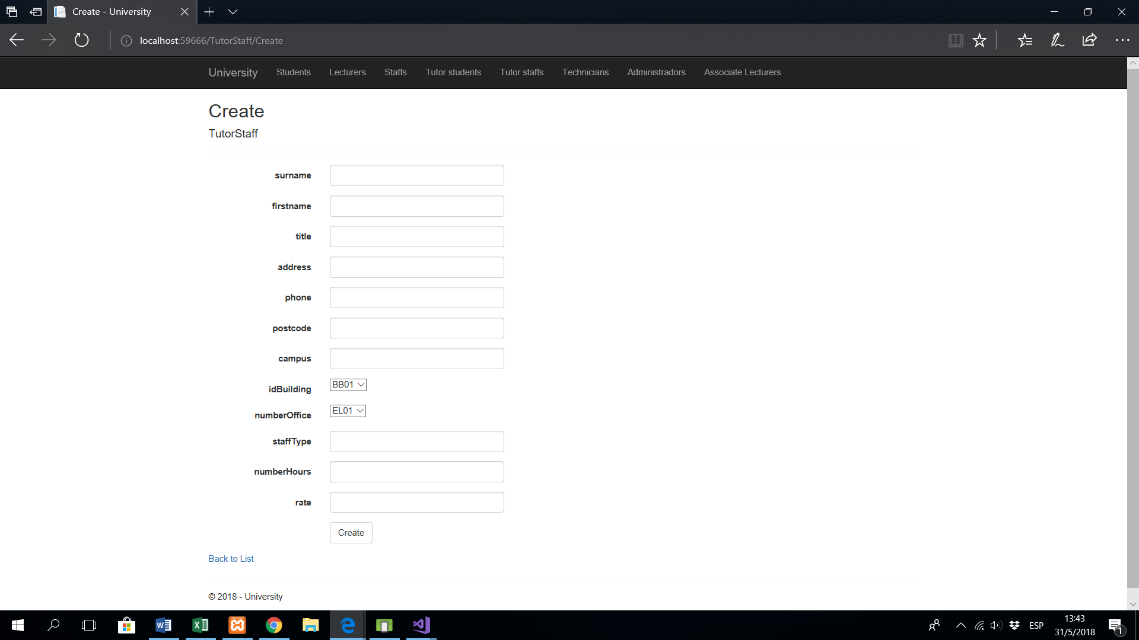
1. **Actualizar registro**

Al tocar el botón de editar este podrá actualizar los datos del registro en el sistema. Este podrá actualizar los registros, después de actualizarlos deberá tocar el botón de *save* para guardar los cambios en el sistema.



1. **Crear registro**

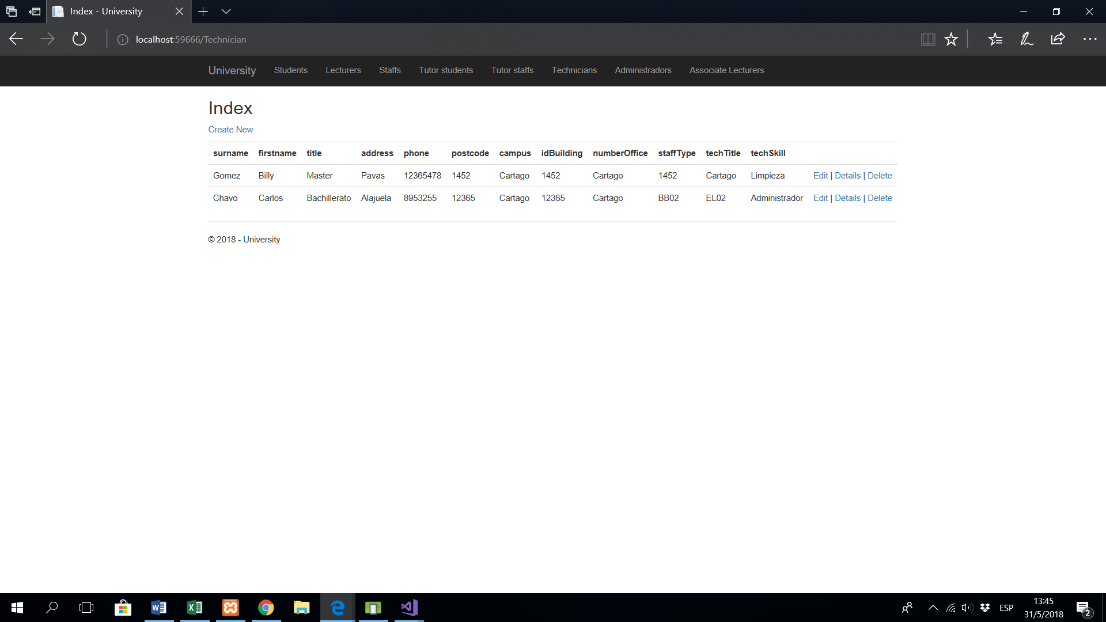
Al tocar *Create New*, se le mostrará los campos que debe llenar para poder registrar el registro en el sistema. AL terminar de llenar los datos deberá tocar el botón de *Create* para poder registrar el nuevo registro en el sistema.



* Sección Technicians

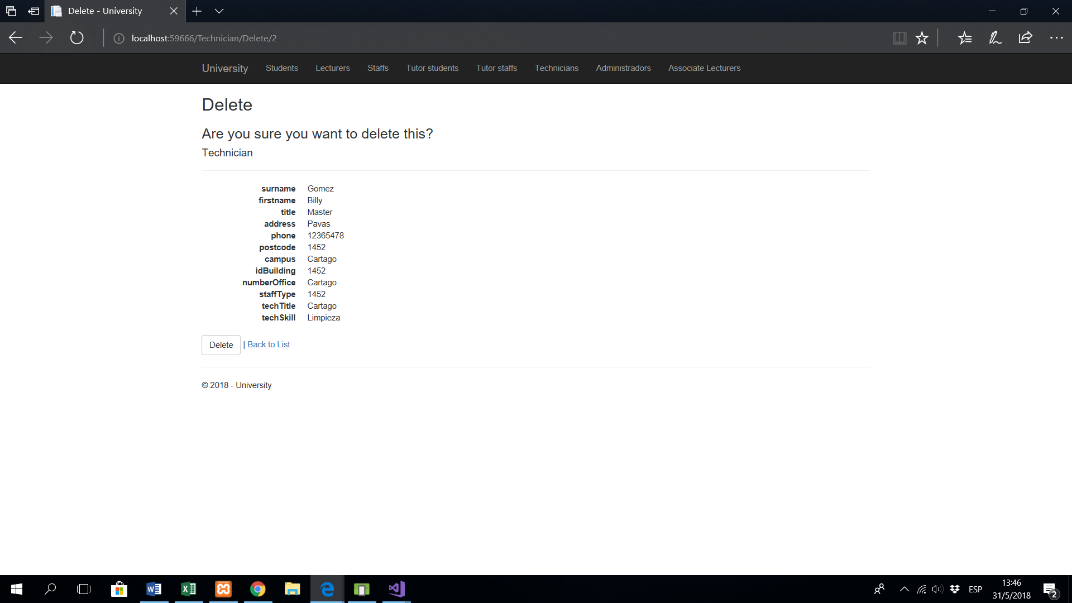
1. **Visualizar registros**

En esta sección se podrá visualizar los registros de los técnicos guardados en el sistema.



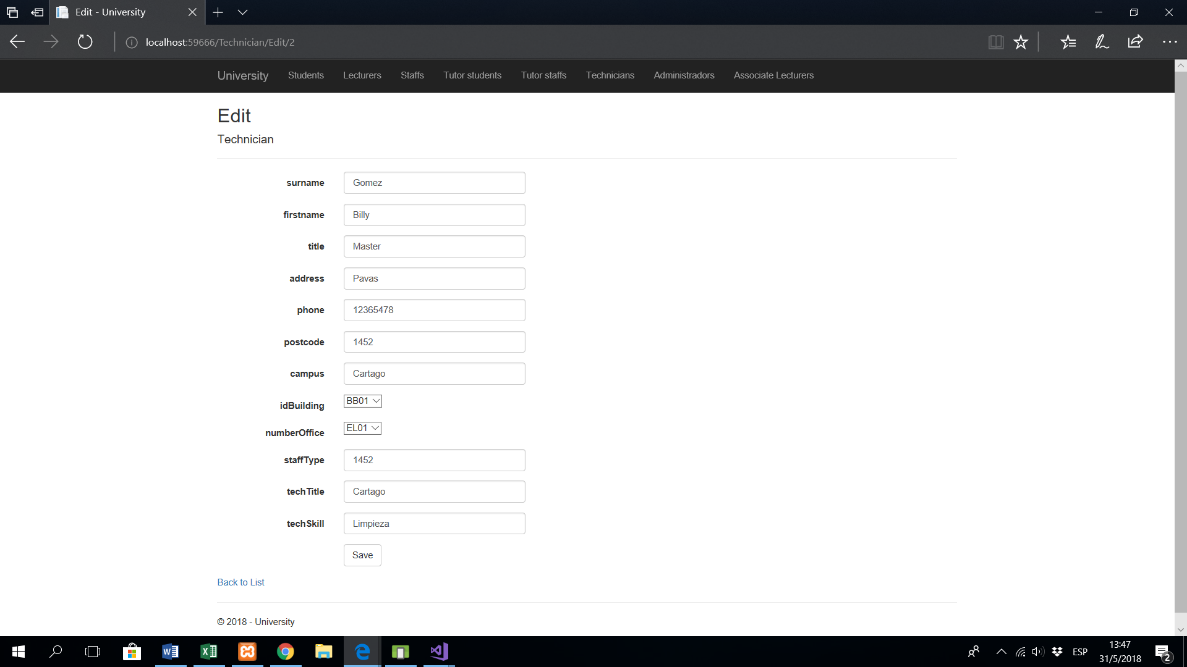
1. **Eliminar registro**

Después de hacer click en un registro en la opción de delete se le mostrará la siguiente vista para eliminar el registro seleccionado. Si está seguro de eliminar el registro debe tocar el botón *Delete* o sino *back To List* para regresar a la sección anterior.



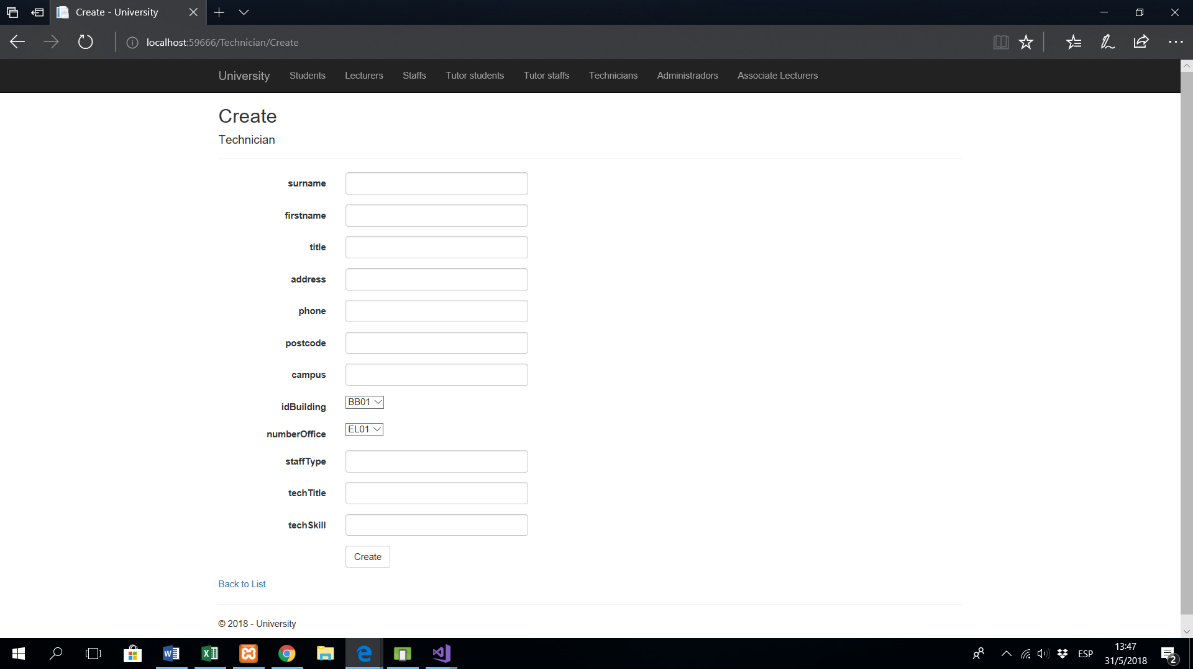
1. **Actualizar registro**

Al tocar el botón de editar este podrá actualizar los datos del registro en el sistema. Este podrá actualizar los registros, después de actualizarlos deberá tocar el botón de *save* para guardar los cambios en el sistema.



1. **Crear registro**

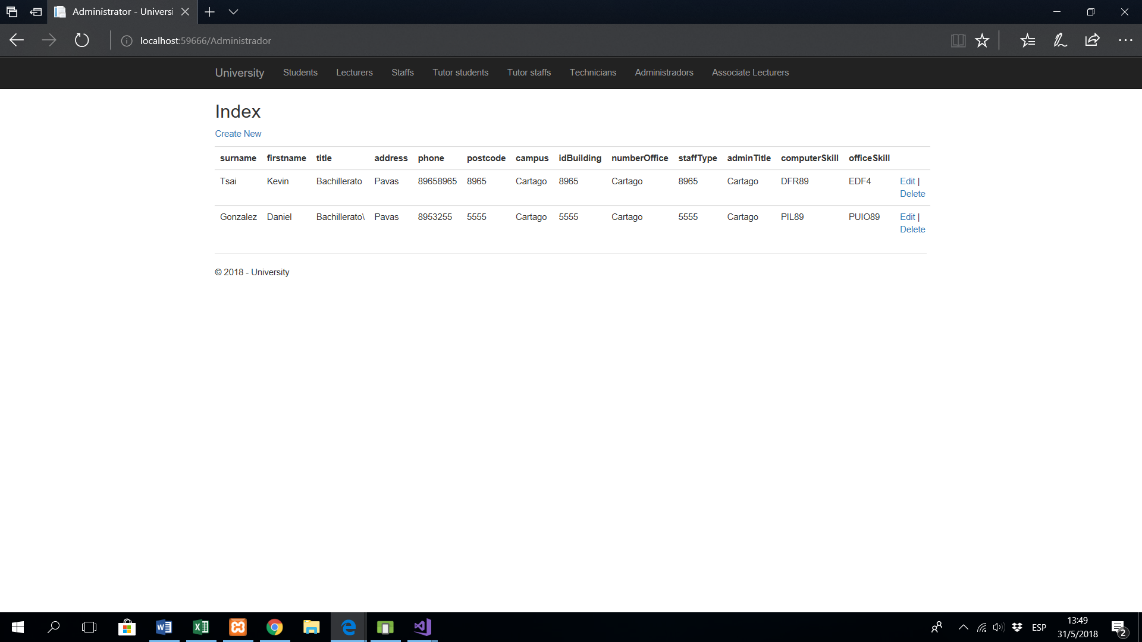
Al tocar *Create New*, se le mostrará los campos que debe llenar para poder registrar el registro en el sistema. AL terminar de llenar los datos deberá tocar el botón de *Create* para poder registrar el nuevo registro en el sistema.



* Sección Administrator

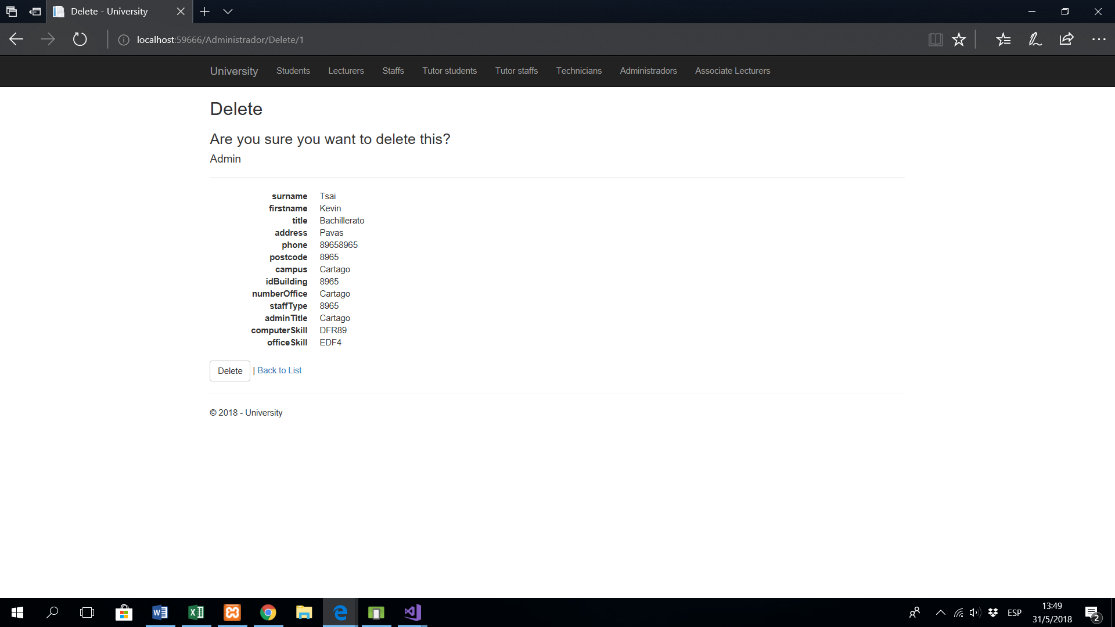
1. **Visualizar registros**

En esta sección se podrá visualizar los registros de los administradores guardados en el sistema.



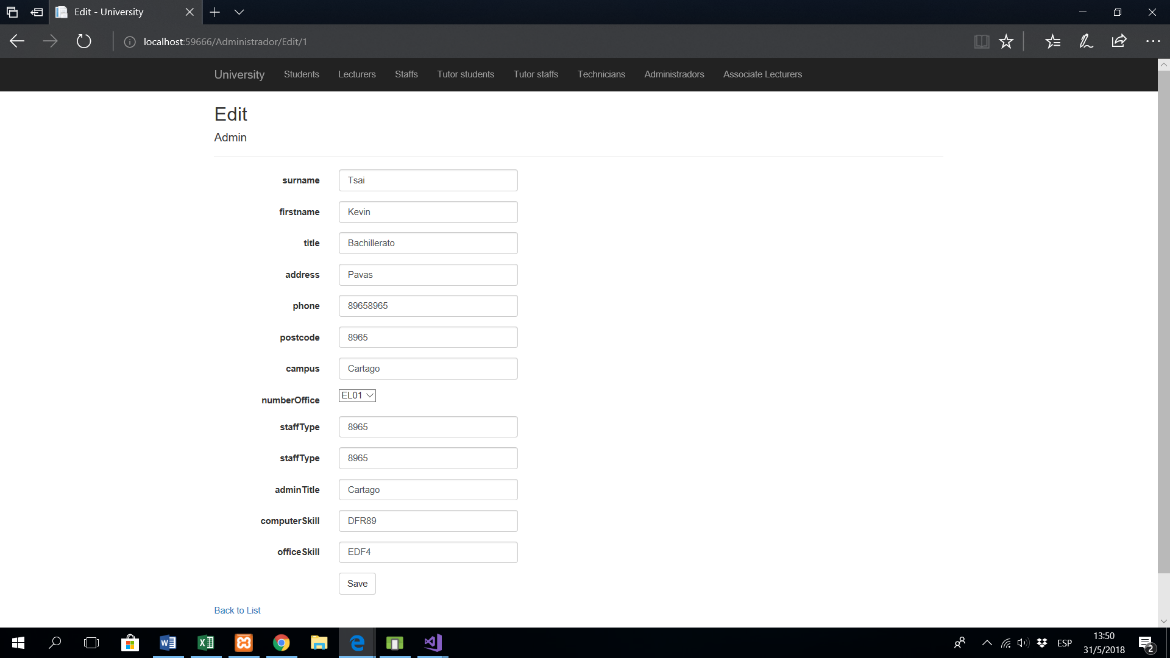
1. **Eliminar registro**

Después de hacer click en un registro en la opción de delete se le mostrará la siguiente vista para eliminar el registro seleccionado. Si está seguro de eliminar el registro debe tocar el botón *Delete* o sino *back To List* para regresar a la sección anterior.

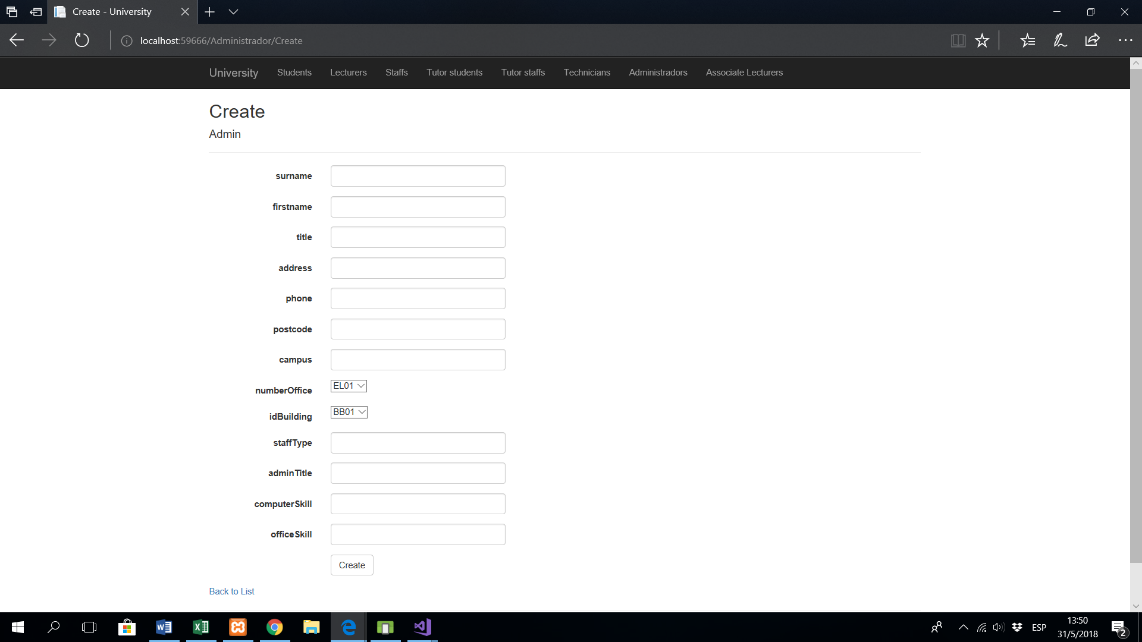


1. **Actualizar registro**

Al tocar el botón de editar este podrá actualizar los datos del registro en el sistema. Este podrá actualizar los registros, después de actualizarlos deberá tocar el botón de *save* para guardar los cambios en el sistema.



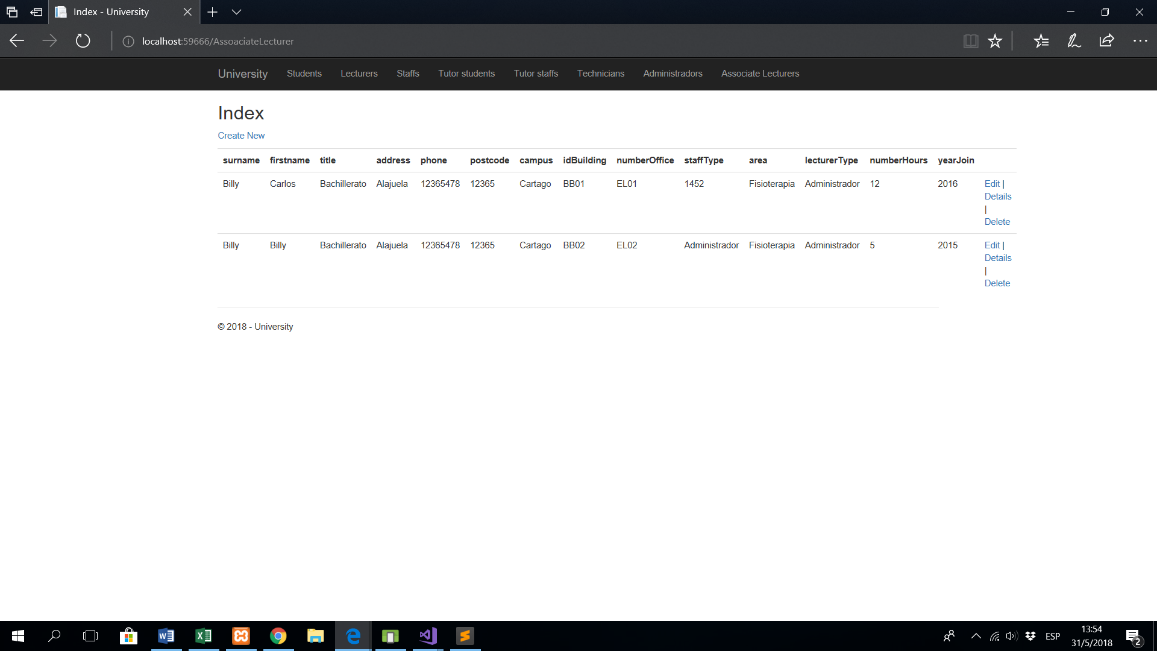
1. **Crear registro**

Al tocar *Create New*, se le mostrará los campos que debe llenar para poder registrar el registro en el sistema. AL terminar de llenar los datos deberá tocar el botón de *Create* para poder registrar el nuevo registro en el sistema.

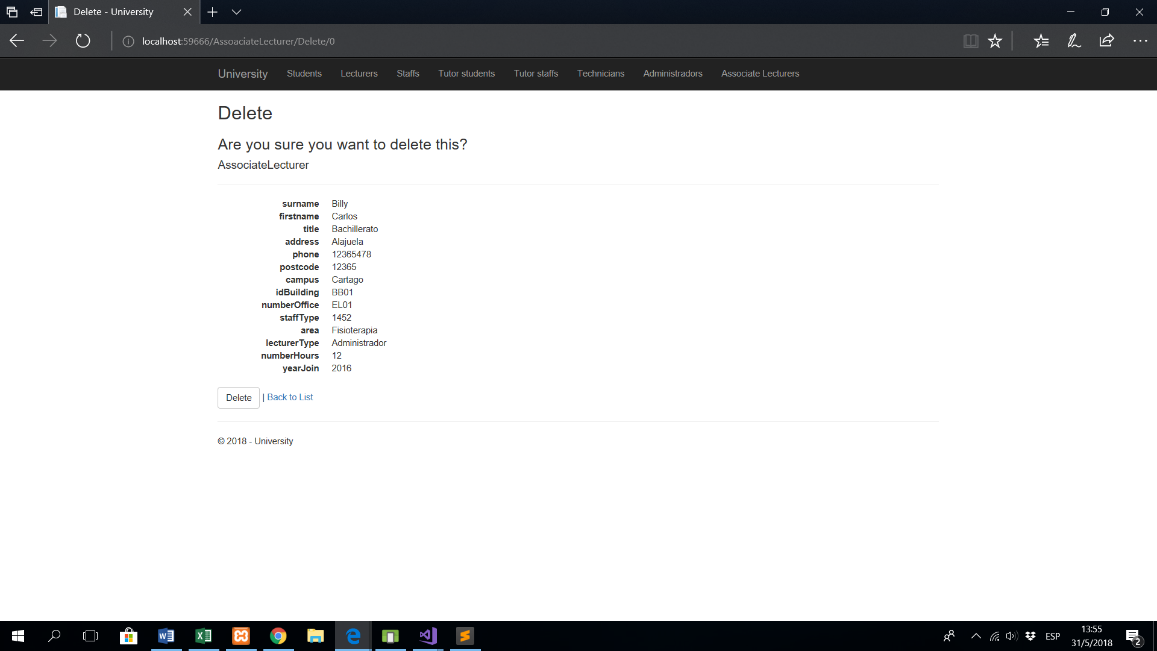
* Sección Profesores asociados

1. **Visualizar registros**

En esta sección se podrá visualizar los registros de los Profesores asociados guardados en el sistema.

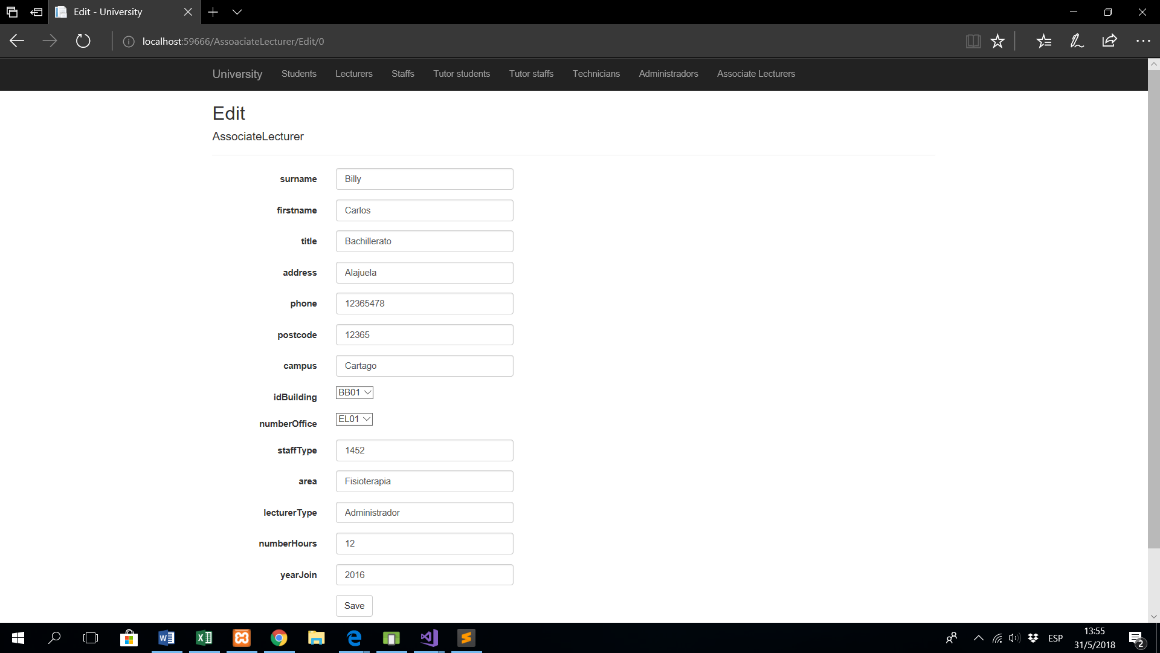


1. **Eliminar registro**

Después de hacer click en un registro en la opción de delete se le mostrará la siguiente vista para eliminar el registro seleccionado. Si está seguro de eliminar el registro debe tocar el botón *Delete* o sino *back To List* para regresar a la sección anterior.

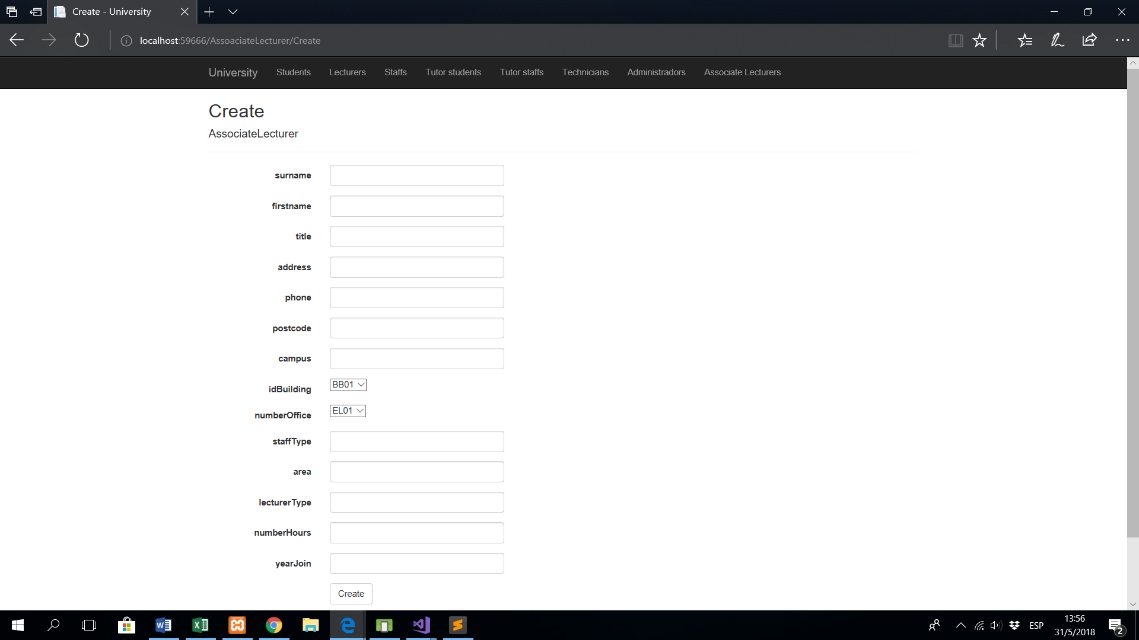
1. **Actualizar registro**

Al tocar el botón de editar este podrá actualizar los datos del registro en el sistema. Este podrá actualizar los registros, después de actualizarlos deberá tocar el botón de *save* para guardar los cambios en el sistema.



1. **Crear registro**

Al tocar *Create New*, se le mostrará los campos que debe llenar para poder registrar el registro en el sistema. AL terminar de llenar los datos deberá tocar el botón de *Create* para poder registrar el nuevo registro en el sistema.



**Documentación del web service**

|  |  |  |
| --- | --- | --- |
| Admin |  |  |
| API | **Description** | **Example** |
| [GET api/Admin](http://localhost:62898/Help/Api/GET-api-Admin) | Método que recupera todos los datos de Admin de la base de datos. | <http://localhost:62898/api/Admin> |
| [POST api/Admin](http://localhost:62898/Help/Api/POST-api-Admin) | Método POST, que crea el Admin en la base de datos. | <http://localhost:62898/api/Admin> |
| [PUT api/Admin/{id}](http://localhost:62898/Help/Api/PUT-api-Admin-id) | Método que actualiza el Admin en la base de datos. | <http://localhost:62898/api/Admin/6> |
| [DELETE api/Admin/{id}](http://localhost:62898/Help/Api/DELETE-api-Admin-id) | Método que elimina el Admin en la base de datos. | <http://localhost:62898/api/Admin/5> |

|  |  |  |
| --- | --- | --- |
| AssociateLecturer |  |  |
| API | **Description** | **Example** |
| [GET api/AssociateLecturer](http://localhost:62898/Help/Api/GET-api-Admin) | Método que recupera todos los datos de AssociateLecturer de la base de datos. | <http://localhost:62898/api/AssociateLecturer> |
| [POST api/AssociateLecturer](http://localhost:62898/Help/Api/POST-api-Admin) | Método POST, que crea el AssociateLecturer en la base de datos. | <http://localhost:62898/api/AssociateLecturer> |
| [PUT api/AssociateLecturer/{id}](http://localhost:62898/Help/Api/PUT-api-Admin-id) | Método que actualiza el AssociateLecturer en la base de datos. | <http://localhost:62898/api/AssociateLecturer/6> |
| [DELETE api/AssociateLecturer/{id}](http://localhost:62898/Help/Api/DELETE-api-Admin-id) | Método que elimina el AssociateLecturer en la base de datos. | <http://localhost:62898/api/AssociateLecturer/5> |

|  |  |  |
| --- | --- | --- |
| TutorStaff |  |  |
| API | **Description** | **Example** |
| [GET api/TutorStaff](http://localhost:62898/Help/Api/GET-api-TutorStaff) | Método que recupera todos los datos de TutorStaff de la base de datos. | <http://localhost:62898/api/TutorStaff> |
| [POST api/TutorStaff](http://localhost:62898/Help/Api/POST-api-TutorStaff) | Método POST, que crea el TutorStaff en la base de datos. | <http://localhost:62898/api/TutorStaff> |
| [PUT api/TutorStaff/{id}](http://localhost:62898/Help/Api/PUT-api-TutorStaff-id) | Método que actualiza el TutorStaff en la base de datos. | <http://localhost:62898/api/TutorStaff/6> |
| [DELETE api/TutorStaff/{id}](http://localhost:62898/Help/Api/DELETE-api-TutorStaff-id) | Método que elimina el TutorStaff en la base de datos. | <http://localhost:62898/api/TutorStaff/5> |

|  |  |  |
| --- | --- | --- |
| TutorStudent |  |  |
| API | **Description** | **Example** |
| [GET api/TutorStudent](http://localhost:62898/Help/Api/GET-api-TutorStudent) | Método que recupera todos los datos de TutorStudent de la base de datos. | <http://localhost:62898/api/TutorStudent> |
| [POST api/TutorStudent](http://localhost:62898/Help/Api/POST-api-TutorStudent) | Método POST, que crea el TutorStudent en la base de datos. | <http://localhost:62898/api/TutorStudent> |
| [PUT api/TutorStudent/{id}](http://localhost:62898/Help/Api/PUT-api-TutorStudent-id) | Método que actualiza el TutorStudent en la base de datos. | <http://localhost:62898/api/TutorStudent/6> |
| [DELETE api/TutorStudent/{id}](http://localhost:62898/Help/Api/DELETE-api-TutorStudent-id) | Método que elimina el TutorStudent en la base de datos. | <http://localhost:62898/api/TutorStudent/5> |

|  |  |  |
| --- | --- | --- |
| Techinician |  |  |
| API | **Description** | **Example** |
| [GET api/Techinician](http://localhost:62898/Help/Api/GET-api-Techinician) | Método que recupera todos los datos de Techinician de la base de datos. | <http://localhost:62898/api/Techinician> |
| [POST api/Techinician](http://localhost:62898/Help/Api/POST-api-Techinician) | Método POST, que crea el Techinician en la base de datos. | <http://localhost:62898/api/Techinician> |
| [PUT api/Techinician/{id}](http://localhost:62898/Help/Api/PUT-api-Techinician-id) | Método que actualiza el Techinician en la base de datos. | <http://localhost:62898/api/Techinician/6> |
| [DELETE api/Techinician/{id}](http://localhost:62898/Help/Api/DELETE-api-Techinician-id) | Método que elimina el Techinician en la base de datos. | <http://localhost:62898/api/Techinician/5> |

|  |  |  |
| --- | --- | --- |
| Student |  |  |
| API | **Description** | **Example** |
| [GET api/Student](http://localhost:62898/Help/Api/GET-api-Student) | Método que recupera todos los datos de Student de la base de datos. | <http://localhost:62898/api/Student> |
| [POST api/Student](http://localhost:62898/Help/Api/POST-api-Student) | Método POST, que crea el Student en la base de datos. | <http://localhost:62898/api/Student> |
| [PUT api/Student/{id}](http://localhost:62898/Help/Api/PUT-api-Student-id) | Método que actualiza el Student en la base de datos. | <http://localhost:62898/api/Student/6> |
| [DELETE api/Student/{id}](http://localhost:62898/Help/Api/DELETE-api-Student-id) | Método que elimina el Student en la base de datos. | <http://localhost:62898/api/Student/5> |

|  |  |  |
| --- | --- | --- |
| Staff |  |  |
| API | **Description** | **Example** |
| [GET api/Staff](http://localhost:62898/Help/Api/GET-api-Staff) | Método que recupera todos los datos de Staff de la base de datos. | <http://localhost:62898/api/Staff> |
| [POST api/Staff](http://localhost:62898/Help/Api/POST-api-Staff) | Método POST, que crea el Staff en la base de datos. | <http://localhost:62898/api/Staff> |
| [PUT api/Staff/{id}](http://localhost:62898/Help/Api/PUT-api-Staff-id) | Método que actualiza el Staff en la base de datos. | <http://localhost:62898/api/Staff/6> |
| [DELETE api/Staff/{id}](http://localhost:62898/Help/Api/DELETE-api-Staff-id) | Método que elimina el Staff en la base de datos. | <http://localhost:62898/api/Staff/5> |

|  |  |  |
| --- | --- | --- |
| Lecturer |  |  |
| API | **Description** | **Example** |
| [GET api/Lecturer](http://localhost:62898/Help/Api/GET-api-Lecturer) | Método que recupera todos los datos de Lecturer de la base de datos. | <http://localhost:62898/api/Lecturer> |
| [POST api/Lecturer](http://localhost:62898/Help/Api/POST-api-Lecturer) | Método POST, que crea el Lecturer en la base de datos. | <http://localhost:62898/api/Lecturer> |
| [PUT api/Lecturer/{id}](http://localhost:62898/Help/Api/PUT-api-Lecturer-id) | Método que actualiza el Lecturer en la base de datos. | <http://localhost:62898/api/Lecturer/6> |
| [DELETE api/Lecturer/{id}](http://localhost:62898/Help/Api/DELETE-api-Lecturer-id) | Método que elimina el Lecturer en la base de datos. | <http://localhost:62898/api/Lecturer/5> |