

**UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO**  
**FACULTAD DE INGENIERÍA ELÉCTRICA, ELECTRÓNICA, INFORMÁTICA Y  
MECÁNICA**



**ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS**

---

**“INFORME FINAL DEL SISTEMA DE TUTORÍAS”  
GRUPO 2**

---

**Asignatura:** Desarrollo de Software I

**Docente:** QUINTANILLA PORTUGAL, Roxana Lisette

<b>Estudiantes:</b>	<b>Código</b>
---------------------	---------------

CCUITO QUISPE, Jhon Albert	161757
CERATI CERRILLO, Fiorella	182898
ESPINOZA CHAMPI, Israel Enrique	182908
MASIAS USCAMAYTA, Naomi Isabel	155180
MORA CCARHUARUPAY, Luz Lucero	182923
OLIVARES CAMERO, Aler Sebastian	182925
QUISPE PUMA, Lucero	151780
TINTAYA TACO, Yurema Lisbeth	174454
VILCAHUAMAN CACERES, Miguel	174961

**Cusco - Perú**

**2021**

## **CAPÍTULO I: Proceso del negocio**

### **1. Descripción del negocio.**

#### **1.1 Descripción general del negocio**

##### **Razón Social**

Universidad Nacional de San Antonio Abad del Cusco (Escuela Profesional de Ingeniería Informática y de Sistemas)

##### **Giro del Negocio**

Educación superior universitaria

##### **Descripción General de sus actividades**

Universidad San Antonio Abad del Cusco, corporación universitaria sin fines de lucro. La principal actividad es garantizar calidad de educación, mejorar la formación profesional y promover el desarrollo humano y para el dicho desarrollo del estudiante también aplica el sistema de tutoría para una mejora estadía estudiantil

### **UNIVERSIDAD NACIONAL DE SAN ANTONIO ABAD DEL CUSCO**

### **REGLAMENTO DE TUTORÍA DE LA ESCUELA PROFESIONAL DE INGENIERÍA INFORMÁTICA Y DE SISTEMAS**

#### **Normas generales**

##### **Art. 1º Naturaleza y alcance del reglamento.**

El presente reglamento regula la actividad tutorial académica en la Escuela Profesional de Ingeniería Informática y de Sistemas de la Universidad Nacional de San Antonio Abad del Cusco (UNSAAC) que se brinda a los estudiantes matriculados en esta escuela. Comprende a autoridades, funcionarios académicos, profesores, estudiantes y personal administrativo de apoyo.

##### **Art. 2º Base Legal**

1. Constitución Política del Perú de 1993
2. Ley Universitaria 30220, Art. 87.5
3. Estatuto de la UNSAAC, Art. 195.5
4. Ley de Procedimientos Administrativos
5. Ley de Control Interno de las Entidades del Estado
6. R.S. N^001-2007-ED – Aprueba como política de Estado el Proyecto Educativo Nacional al
7. Resolución C.U.-0220-2017-UNSAAC. Reglamento de Tutoría Académica de la UNSAAC.

Art. 3º De la Ley Universitaria 30220.

El Art. 102º de la citada ley establece los siguiente:

- a) A los estudiantes que desaprueben por segunda vez una materia (asignatura) se les brindará una dedicación tutorial específica, de carácter obligatorio.
- b) La desaprobación de una misma materia (asignatura) por tres veces da lugar a que el estudiante sea separado temporalmente por un año de la universidad. Al término de este plazo, el estudiante solo se podrá matricular en un máximo de dos materias (asignaturas) desaprobadas por tercera vez.
- c) Si un estudiante desaprueba por cuarta vez una misma materia (asignatura) procede su retiro definitivo de la universidad.

#### Del Objetivo

Art. 4º El objetivo del presente reglamento es establecer procedimientos y procesos que regulan el Programa de tutorías para los Estudiantes de la Escuela Profesional de Ingeniería Informática y de Sistemas en Riesgo Académico.

#### Del Alcance

Art. 5º El presente Reglamento es de cumplimiento obligatorio por parte de las autoridades, funcionarios, docentes y estudiantes de la Escuela Profesional en riesgo académico.

Art. 6º El presente Reglamento contempla la aplicación e implementación de un Programa de Tutoría que estará bajo la responsabilidad del Decano de la Facultad de Ingeniería Eléctrica, Electrónica, Informática y Mecánica de la UNSAAC.

Art. 7º La oficina de Bienestar Universitario brindará obligatoriamente a la Escuela Profesional asistencia psicológica y asistencia social para los estudiantes comprendidos en el presente reglamento.

#### Del Programa de Tutoría

Art. 8º El Programa de tutoría tiene como objetivo contribuir a mejorar el rendimiento académico de los estudiantes, para prevenir o tratar de evitar la separación temporal o retiro definitivo de la Escuela Profesional de los estudiantes que se encuentren en riesgo académico.

Art. 9º El Programa de Tutoría es de carácter obligatorio para los estudiantes que han desaprobado por segunda vez una asignatura.

Art. 10° El Programa de Tutoría brinda a los estudiantes en riesgo académico los siguientes servicios:

- a) Tutoría: realizada por un Docente – Tutor.
- b) Talleres de reforzamiento académico: realizada por los Estudiantes Ayudantes de Tutoría solicitud del Docente-Tutor
- c) Asistencia Psicológica: realizada por un(a) licenciado (a) en psicología
- d) Asistencia Social: Realizada por un(a) licenciada en Trabajo Social

Art. 11° La aplicación del Programa de Tutoría tiene los siguientes propósitos:

- a) Brindar al estudiante que se encuentre en riesgo académico un asesoramiento personalizado en el proceso de matrícula, recomendándole las asignaturas que debe llevar en un determinado período académico.
- b) Guiar y orientar al estudiante que se encuentra en riesgo académico, de manera ordenada y sistemática, de conformidad con la malla curricular.
- c) Monitorear el desempeño académico del estudiante que se encuentra en riesgo académico durante el período académico correspondiente.
- d) Brindar al estudiante que se encuentra en riesgo académico talleres de reforzamiento en las asignaturas que lo necesite
- e) Incentivar al estudiante que se encuentra en riesgo académico el interés por los estudios.
- f) Brindar al estudiante que se encuentra en riesgo académico asistencia psicológica y asistencia social.

Del Consejo de Facultad de la Facultad de Ingeniería Eléctrica, Electrónica, Informática y Mecánica. Art. 12° El Consejo de Facultad es el encargado de aprobar el Plan Anual de Trabajo del Programa de Tutoría, elaborado y presentado por el Coordinador General de dicho programa.

Art. 13° El Consejo de Facultad al término de cada período académico evaluará el informe de las actividades realizadas por el Coordinador General del Programa de Tutoría.

Art. 14º El Consejo de Facultad es el encargado de otorgar a los Estudiantes Ayudantes de Tutoría los créditos por la actividad extra curricular por realizar talleres de reforzamiento académico.

Del Decanato de la Facultad de Ingeniería Eléctrica, Electrónica, Informática y Mecánica.

Art. 15º El Decanato, a través del Decano de la Facultad gestionará ante las dependencias correspondientes de la UNSAAC, si lo considera necesario, los recursos humanos, materiales, etc. para la implementación del Programa de Tutoría en la Escuela Profesional de Ingeniería Informática y de Sistemas.

Art. 16º El Decano, designa al Coordinador General del Programa de Tutoría mediante Resolución Decanal por un período de dos (02) años, quien será un docente ordinario principal o asociado con un mínimo de cinco (05) años de experiencia docente en la UNSAAC.

De la Dirección de Escuela Profesional de Ingeniería Informática y de Sistemas.

Art. 17º Son funciones del director de la Escuela Profesional:

- a) Solicitar a la Dirección del Centro de Cómputo el Avance Curricular de los estudiantes que se encuentren en riesgo académico
- b) Determinar y seleccionar en coordinación con el Decano de la Facultad la cantidad necesaria de Docentes Tutores para cada semestre académico, teniendo en cuenta que cada Docente Tutor tendrá a su cargo el asesoramiento de un número proporcional de estudiantes en riesgo académico. Esta información será remitida al Coordinador General del Programa de Tutoría, incluyendo el Avance Curricular de los estudiantes en riesgo académico.
- e) Elaborar el perfil del Estudiante Ayudante de Tutoría.
- c) Seleccionar a los Estudiantes Ayudantes de Tutoría a través de un concurso interno en la Facultad de acuerdo a los requerimientos del Docente Tutor.

De la Oficina del Programa de Tutoría

Art. 18º La Oficina del Programa de Tutoría es el órgano ejecutivo del Programa de Tutoría para estudiantes que se encuentran en riesgo académico; se implementará en cada Facultad y estará a cargo del Coordinador General del Programa de Tutoría.

Art. 19º El Coordinador General del Programa de Tutoría es el responsable de elaborar en coordinación con el Directores de la Escuela Profesional el Plan Anual de Trabajo del Programa de Tutoría,

el mismo que deberá ser presentado al Consejo de Facultad a través del Decano. Asimismo, el Coordinador General del Programa de Tutoría es el responsable de ejecutar dicho Plan.

Art. 20º Son funciones y obligaciones del Coordinador General del Programa de Tutoría:

- a) Proporcionar a cada Docente Tutor el Avance Curricular de los estudiantes en riesgo académico que se encuentran bajo su tutoría. Esta información debe ser confidencial.
- b) Programar para los Docentes Tutores de la Escuela Profesional los horarios y los lugares de atención que brindaran a los estudiantes que se encuentran en riesgo académico bajo su tutoría, de acuerdo a lo señalado en el Art. 22º. Asimismo, publicar en físico y en el Portal Web de la Escuela Profesional la relación de Docentes Tutores indicando sus oficinas, horarios de atención y la relación de códigos de los estudiantes en riesgo académico que estarán bajo la tutoría de cada Docente Tutor.
- c) Programar y publicar en físico y en el Portal Web de la Escuela Profesional el horario y aulas donde se realizarán los talleres de reforzamiento académico para las diferentes asignaturas que sean necesarias, para los estudiantes en riesgo académico, los mismos que serán realizados por los Estudiantes Ayudantes de Tutoría, de acuerdo a lo señalado en el Art. 17º, inciso d).
- e) Dirigir, supervisar y controlar las actividades del Docente Tutor a fin de que se cumpla con el desarrollo del Plan de Trabajo del Programa de Tutoría.
- f) Atender las solicitudes que presenten los Docentes Tutores, los estudiantes ayudantes de tutoría y los estudiantes que se encuentran en riesgo académico en lo que corresponde al Programa de Tutoría.
- g) Solicitar al Decano de la Facultad un profesional especializado en Psicología y en Trabajo Social para dar atención a los estudiantes en riesgo académico.
- h) Remitir mensualmente a los Docentes Tutores los informes de los estudiantes que se encuentran en riesgo académico, que fueron atendidos por los profesionales de Psicología y de Trabajo Social.
- i) Presentar al Consejo de Facultad a través del Decano cada quince (15) días después del cierre programado del período académico correspondiente, el informe de las actividades realizadas del Programa de Tutoría de la Escuela

Profesional, con el visto bueno del Director de Escuela Profesional respectiva.

- j) Presentar mensualmente al Director de la Escuela Profesional correspondiente un informe detallado de las actividades desarrolladas por los Docentes Tutores, por los Estudiantes Ayudantes de Tutoría, por los profesionales de Psicología y Trabajo Social de la Facultad.
- k) Convocar mensualmente a reuniones a los Docentes Tutores, Estudiantes Ayudantes de Tutoría, a los profesionales de Psicología y de Trabajo Social para evaluar el trabajo que vienen desarrollando con los estudiantes que se encuentran en riesgo académico.
- l) Organizar en coordinación con los Directores de Escuelas Profesionales charlas y talleres de capacitación para docentes y estudiantes relacionadas con la temática de tutoría.

#### De los Docentes Tutores

Art. 21° Los Docentes Tutores de la Escuela Profesional desempeñaran la labor de tutoría académica en forma complementaria a sus labores de enseñanza, investigación, capacitación, proyección social y administración. Serán los responsables de asesorar personalmente a cada uno de los estudiantes en riesgo académico que se encuentran bajo su tutoría.

Art. 22° La dedicación de los Docentes Tutores será de seis (06) horas no lectivas por semana; tendrá a su cargo un número proporcional de estudiantes. Estas horas se contarán en su carga no lectiva para efectos de control de permanencia.

El incumplimiento del Docente Tutor a su obligación de tutoría conlleva al descuento económico correspondiente y además significará un demérito en su legajo personal; sin perjuicio de ser sometido a proceso administrativo.

Art. 23° Son funciones y obligaciones de los Docentes Tutores:

- a) Determinar el diagnóstico académico de los estudiantes bajo su tutoría, basándose en el Avance Curricular de los estudiantes que hayan repetido dos veces una asignatura.
- b) Controlar la asistencia de los estudiantes bajo su tutoría en el horario y lugar programado a las sesiones de tutoría e informar semanalmente al Coordinador General del Programa de Tutoría el desempeño académico de los mismos.

- c) Proporcionar a los Estudiantes Ayudantes de Tutoría un formato de control de asistencia de los estudiantes en riesgo académico que están bajo su cargo, que asisten a los talleres de reforzamiento académico.
- d) Orientar y autorizar las asignaturas en las que debe matricularse el estudiante en riesgo académico en el periodo lectivo correspondiente, de acuerdo al reglamento de matrícula.
- e) Brindar personalmente a los estudiantes en riesgo académico que se encuentran bajo su tutoría motivación, orientación, consejos y recomendaciones para su desarrollo académico exitoso.
- f) Monitorear la asistencia y el desarrollo de los talleres de reforzamiento académico realizados por los Estudiantes Ayudantes de Tutoría.
- g) Presentar un informe cada quince (15) días al Coordinador General del Programa de Tutoría del estado situacional (académico, psicológico, etc.) de cada uno de los estudiantes en riesgo académico bajo su tutoría.

#### De los Estudiantes Ayudantes de Tutoría

Art. 23º Los Estudiantes Ayudantes de Tutoría de la Escuela Profesional, son los encargados de brindar talleres de reforzamiento académico en una asignatura que necesite el estudiante en riesgo académico. Deben pertenecer al tercio superior de dicha asignatura. Un Estudiante Ayudante de Tutoría solo podrá desempeñar esta labor un máximo de 2 semestres académicos.

Art. 24º Son deberes de los Estudiantes Ayudantes de Tutoría:

- a) Cumplir con responsabilidad los compromisos asumidos con el Docente Tutor en el Programa de Tutoría.
- b) Mantener una buena relación con sus compañeros, evitando cualquier acto que atente contra la ética, la moral y las buenas costumbres.
- c) Presentar semanalmente al Docente Tutor el reporte de asistencia de los estudiantes a su cargo que se encuentran en riesgo académico.
- d) Cumplir semanalmente con un horario de atención para el taller de reforzamiento de una asignatura como mínimo 3 horas efectivas, para estudiantes que se encuentran en riesgo académico.

Art. 25º Son derechos de los Estudiantes Ayudantes de Tutoría:

- a) a) Recibir un (01) crédito académico extracurricular por cada ciclo de reforzamiento académico en una asignatura a los estudiantes que se encuentran en riesgo académico, realizado en forma satisfactoria. Para obtener el crédito académico extracurricular el Docente Tutor previo informe, solicitará al Director de la Escuela Profesional respectiva se le otorgue al Estudiante Ayudante de Tutoría el crédito correspondiente. El Director de la Escuela Profesional después de evaluar el caso con su visto bueno elevará el pedido al Decano para luego ser visto en Consejo de Facultad.
- b) Recibir un Certificado a nombre de la Facultad por su participación como Estudiante Ayudante de Tutoría.
- c) Reservar la ración alimenticia en el comedor universitario.

Art. 26º Aquellos estudiantes que sean separados por incumplir con sus deberes de Estudiantes Ayudantes de Tutoría perderán inmediatamente los derechos indicados en el Art. 27 y no podrán ser nuevamente designados como Estudiantes Ayudantes de Tutoría.

#### De los estudiantes en riesgo Académico

Art. 27º Los Estudiantes de Pregrado de la Escuela Profesional de Ingeniería Informática y de Sistemas en Riesgo Académico son aquellos estudiantes que desaprueben por segunda vez una asignatura de su Plan de Estudios.

Art. 28º Los Estudiantes en Riesgo Académico tienen los siguientes derechos:

- a) Tener un Docente Tutor asignado por el Coordinador General del Programa de
- d) Tutoría, quien lo asesorará hasta la culminación de un período académico.
- b) Mantener comunicación permanente con su Docente Tutor en forma presencial, durante el horario programado por el Docente Tutor.
- e) Solicitar al Coordinar General del Programa de Tutoría el cambio del Docente Tutor, sustentando dicho pedido convenientemente. De ser aceptada y atendida la solicitud ya no podrá ser uso del derecho de cambio a otro Docente Tutor durante un período académico.
- c) En caso de que un estudiante en riesgo académico se encuentre en calidad de no regular (matriculado en menos de doce (12) créditos), podrá tramitar a pedido del Docente

Tutor, el apoyo de servicios universitarios que dispone la UNSAAC.

- d) Solicitar al Docente Tutor el cambio del Estudiante Ayudante de Tutoría que le brinda reforzamiento académico, sustentando dicho pedido convenientemente. De ser aceptada y atendida la solicitud ya no podrá hacer uso del derecho de cambio a otro Estudiante Ayudante de Tutoría durante un período académico

Art. 29º Los Estudiantes en Riesgo Académico tienen los siguientes deberes:

- a) Cumplir con responsabilidad y dedicación lo que establece el Programa de Tutoría para los Estudiantes de Pregrado en Riesgo Académico (Tutoría personalizada, talleres de reforzamiento, etc.).
- b) Asistir obligatoriamente a la tutoría en el horario y lugar programado por Docente Tutor, de acuerdo a lo indicado en Art. 24º, inciso b).
- c) Asistir a los talleres de reforzamiento académico programados.
- d) Asistir a la entrevista psicológica de entrada en la Facultad o en la Oficina de Bienestar Universitario antes del inicio de un período académico.
- e) Asistir al Área Psicológica de la Facultad o a la Oficina de Bienestar Universitario de la UNSAAC, programada.
- f) Asistir al Área de Asistencia Social de la Facultad o a la Oficina de Bienestar Universitario de la UNSAAC, programada.
- g) Mantener una buena relación con sus compañeros, evitando cualquier acto que atente contra la ética, la moral y las buenas costumbres.
- h) Mantener informado al Docente Tutor de la dirección domiciliaria, del número actualizado de su teléfono celular y correo electrónico.
- i) Informar a su Docente Tutor sobre sus actividades académicas, poniendo énfasis en las dificultades que se presenten para una pronta y adecuada orientación.

Disposiciones transitorias, complementarias y finales

Art. 30º A la entrada en vigencia del presente Reglamento, los estudiantes que se encuentren repitiendo por tres (3) o más veces una o

más asignaturas, obligatoriamente deben matricularse en las asignaturas desaprobadas y asistir a las sesiones de tutoría.

Art. 31º Los estudiantes que al término del período académico 2017-2 hayan desaprobado una misma asignatura por tres (3) veces, estarán sujetos Art. 3º, inciso b) del presente Reglamento.

Art. 32º Los estudiantes que al término del período académico 2017-2 hayan desaprobado una misma asignatura por cuarta vez o más veces, estarán sujetos al Art. 3º, inciso c) del presente Reglamento.

Art. 33º Para la tutoría de los estudiantes que se encuentran involucrados en el Art. 31º y Art. 32º de este Reglamento, se aplicará todo lo que establece el presente Reglamento.

Art. 34º Los casos no contemplados en este Reglamento deben ser resueltos por la Oficina del Programa de Tutoría, los mismos que se pondrán en conocimiento al Vicerrectorado Académico a través del Decano para la mejora del presente Reglamento.

Art. 31º La Facultad, según disponibilidad podrá hacer extensiva la aplicación del presente Reglamento a otros segmentos de los estudiantes, a solicitud de los interesados, los mismos que podrán beneficiarse del programa de tutoría para mejorar su rendimiento académico.

## **1.2 Estructura Organizacional**

### **Escuela Profesional de Ingeniería Informática y de Sistemas**

El Carrera Profesional de Ingeniería Informática y de Sistemas es un ente, que se encarga del desarrollo de la informática en nuestra región, haciendo que el campo de la computación sea un medio más para el desarrollo de nuestra sociedad, así como fomentar la investigación y el desarrollo de software de la región conformada por:

- Director de la Escuela Profesional

Designado por el Decano entre los docentes principales de la Facultad con doctorado en la especialidad correspondiente a la Escuela de la que será Director. El mandato es de tres (3) años. No hay reelección inmediata.

- Estudiantes

Estudiante regular es aquel que registra matrícula en un mínimo de doce (12) créditos. Salvo que falte el menor número de créditos para concluir la carrera.

## **1.3 Identificación del área**

El área en donde se va desarrollar este proyecto es la Escuela Profesional de Ingeniería Informática y de Sistemas.

### **Facultad de Ingeniería Electrica,Electronica y de sistemas**

- Decano:M.sc. Vladimiro Casildo Canal Bravo

### **Escuela profesional de ingeniería informática y de Sistemas**

- Director:Mgt. Nila Zonia Acurio Usca.

#### **Misión**

La Carrera Profesional de Ingeniería Informática y de Sistemas realiza y fomenta la investigación haciendo cursos y conferencias así como la coordinación permanente con nuestro docentes y alumnos, así es como promovemos la investigación de las tesis y su conclusión.

#### **Visión**

La Carrera Profesional de Ingeniería Informática y de Sistemas es un ente, que se encarga del desarrollo de la informática en nuestra región, haciendo que el campo de la computación sea un medio más para el desarrollo de nuestra sociedad, así como fomentar la investigación y del desarrollo de software de la región Cusco

## **2. Identificación de procesos**

### **2.1. Interfaz tutores**

- Modificar ficha
- Crear ficha
- Cambiar contraseña tutor

### **2.2. Interfaz Administrador**

- Insertar nuevo tutor y alumno
- Modificar tutor y alumno
- Eliminar tutor y alumno

## **3. Identificación del problema**

- Dificultad en la administración de los documentos (Ordenar y almacenar los documentos )
- Pérdida de tiempo del Tutor al momento de buscar a su alumno asignado
- Acumulación de documentos con riesgo a pérdidas.
- Problemas de cruce de horarios del tutor con su estudiante.
- Por errores del Administrador encargado al gestionar los documentos de listas de tutor se salte un documento enviando información incompleta
- Conflictos en la verificación de estudiantes y docentes que ya no pertenecen a la escuela profesional
- Pérdida de tiempo en el envío físico de la nómina de tutores hacia los docentes.
- Demora en la búsqueda de las Fichas de Sesión de forma física
- Problemas en el llenado de datos incorrectos del estudiante en una ficha física por lo cual lleva a crear o imprimir una nueva ficha de Sesión generando pérdida de tiempo
- Mala organización para la designación de tutores debido al traspapeleo
- El papeleo generado por las tutorías, como se estaba realizando hasta antes de la pandemia, resulta un trabajo tedioso.
- Ahora en la actualidad debido a la pandemia realizar ese papeleo es imposible por lo que llevar también un sistema mediante hojas de cálculo o nóminas no resulta algo eficiente para controlar las tutorías.

#### **4. Identificación de requerimientos**

Nro	Requerimiento
R1	Iniciar Sesión
R2	Crear Ficha Tutoría
R3	Modificar Ficha de Tutoría
R4	Cambiar contraseña del tutor
R5	Agregar, Modificar, Eliminar Tutor
R6	Agregar, Modificar, Eliminar Alumno

#### **5. Identificación de actores**

ACTORES	Responsabilidades del actor en el sistema de información
---------	--

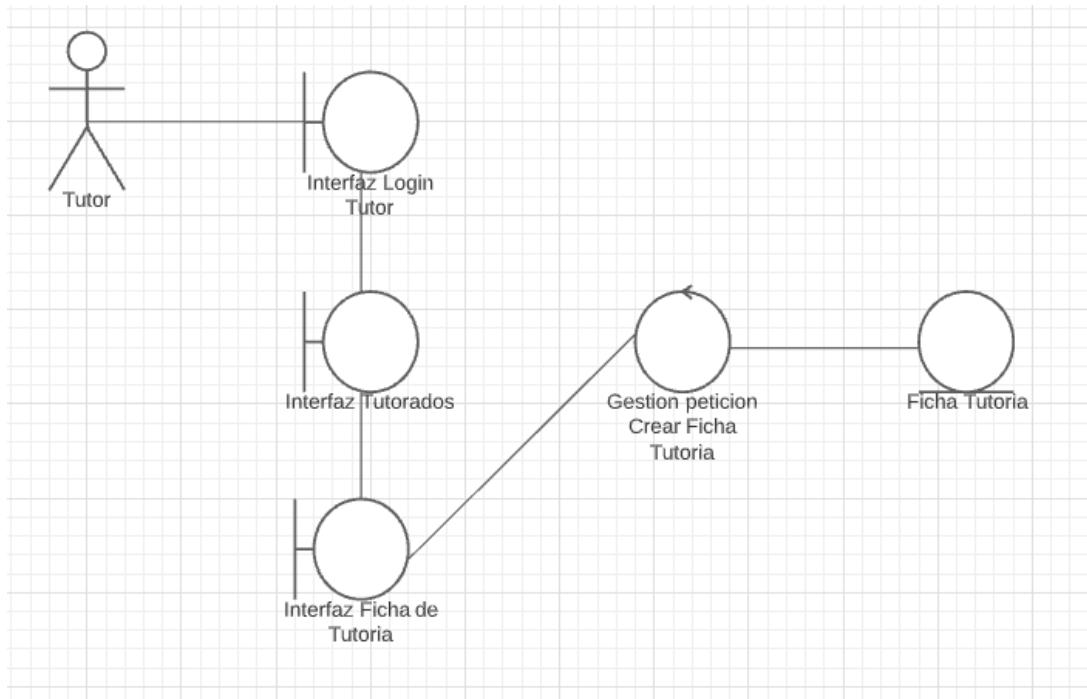
<b>Tutor</b>	Profesor asignado para, principalmente, orientar y guiar a sus alumnos en el proceso de aprendizaje durante uno o varios cursos escolares. Encargado del llenado de ficha de tutoría
<b>Administrador</b>	Responsable de acceder a la interfaz para hacer las operaciones CRUD ya sea sobre docentes o sobre alumnos,
<b>Estudiante</b>	Brinda información a su tutorado para el respectivo llenado de ficha de tutoría

## CAPÍTULO 2:Fase de Elaboración

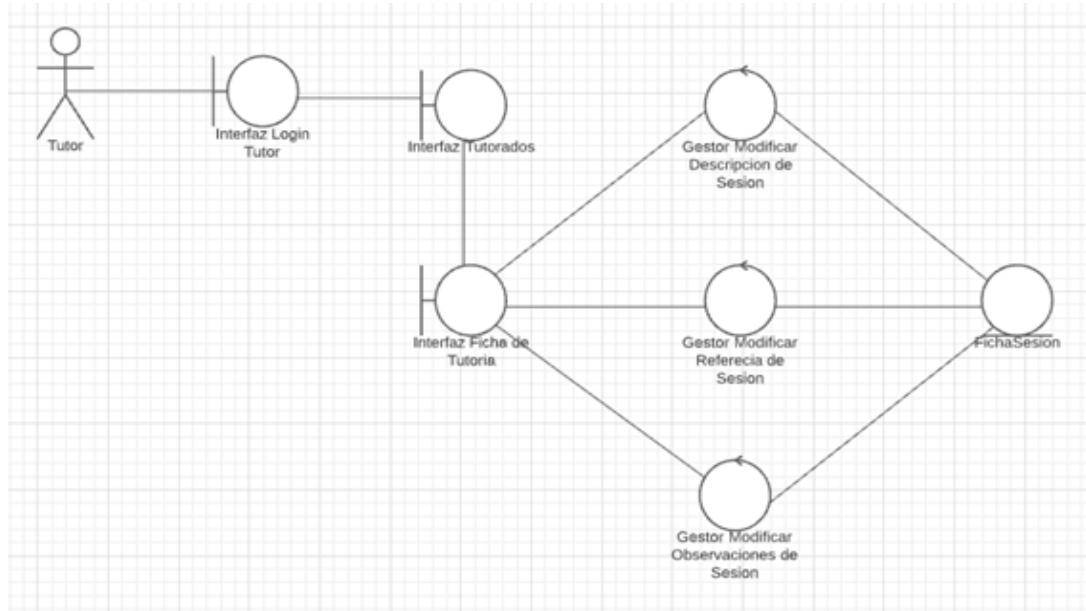
### 1. Casos de uso del sistema

#### 1.1 Diagrama de clases de análisis

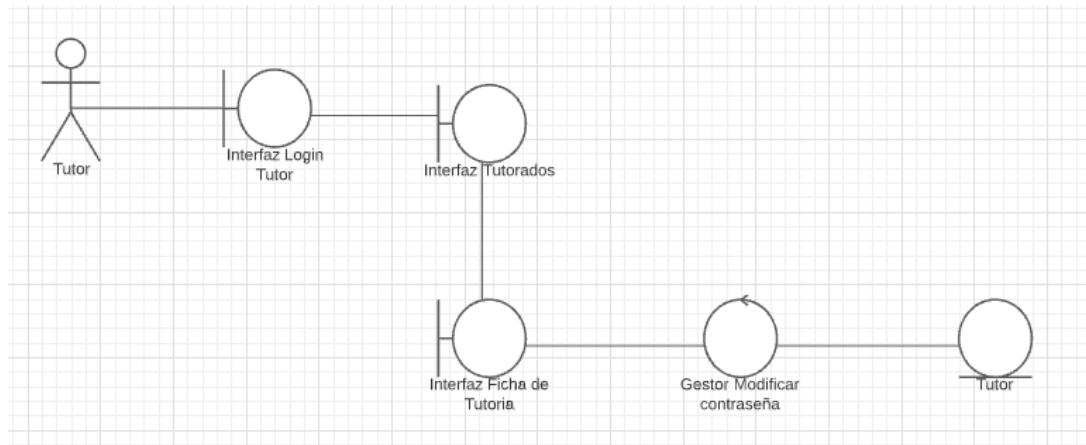
##### Caso de uso: Crear Ficha Tutoría



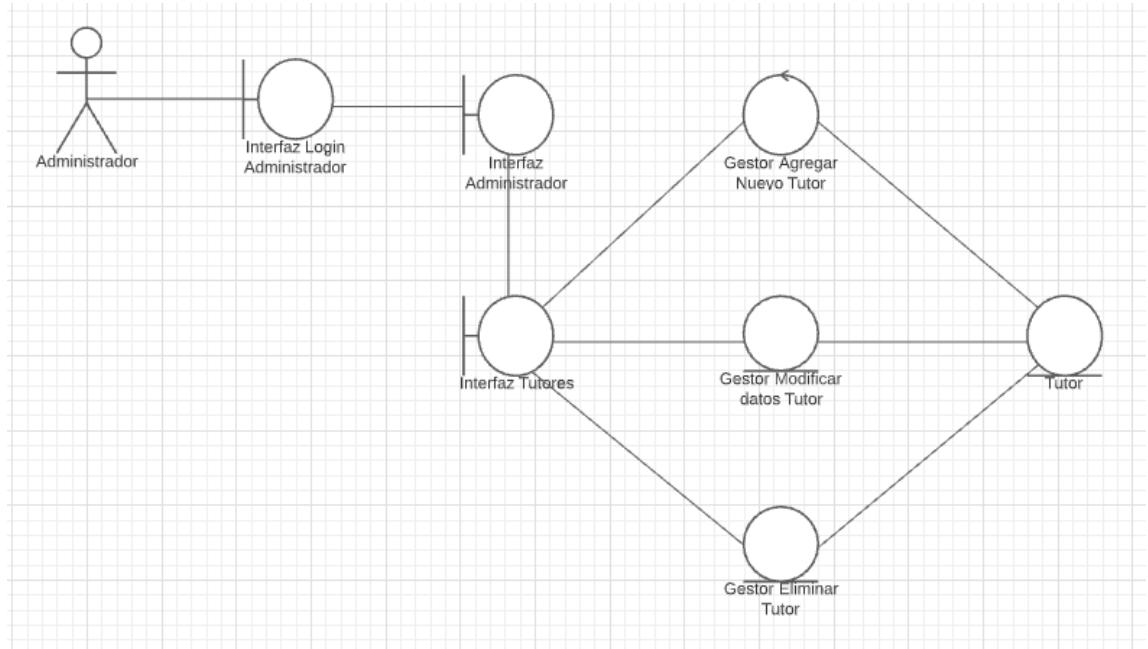
##### Caso de uso: Modificar Ficha Tutoría



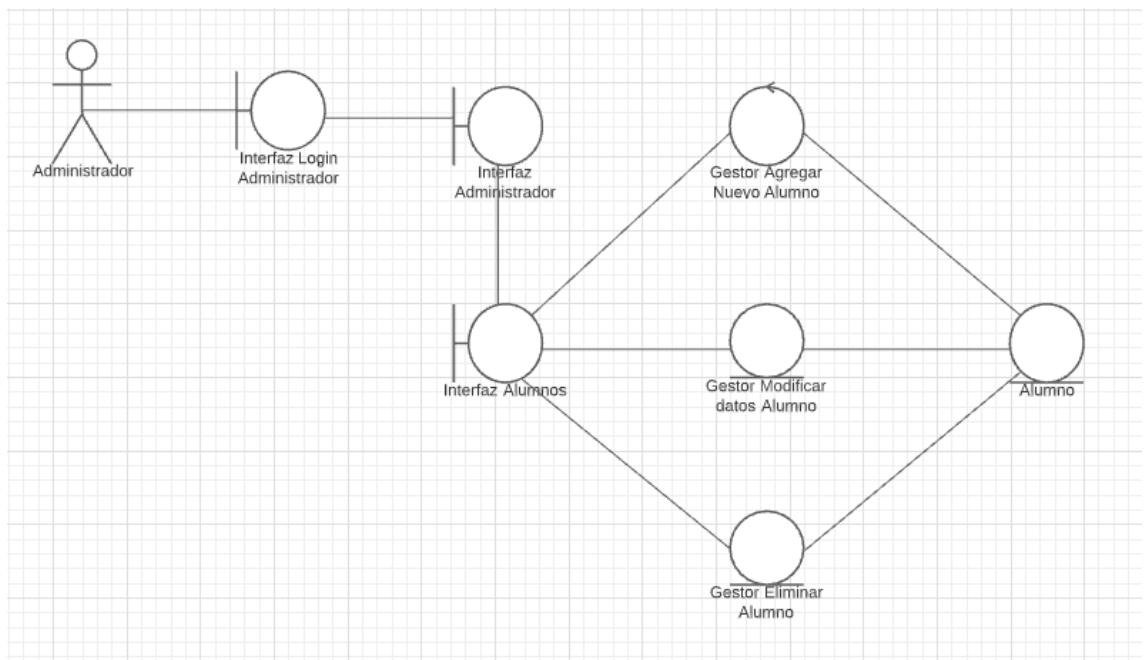
### **caso de uso: Cambiar contraseña de Tutor**



### **caso de uso: Agregar, modificar y eliminar Tutor**

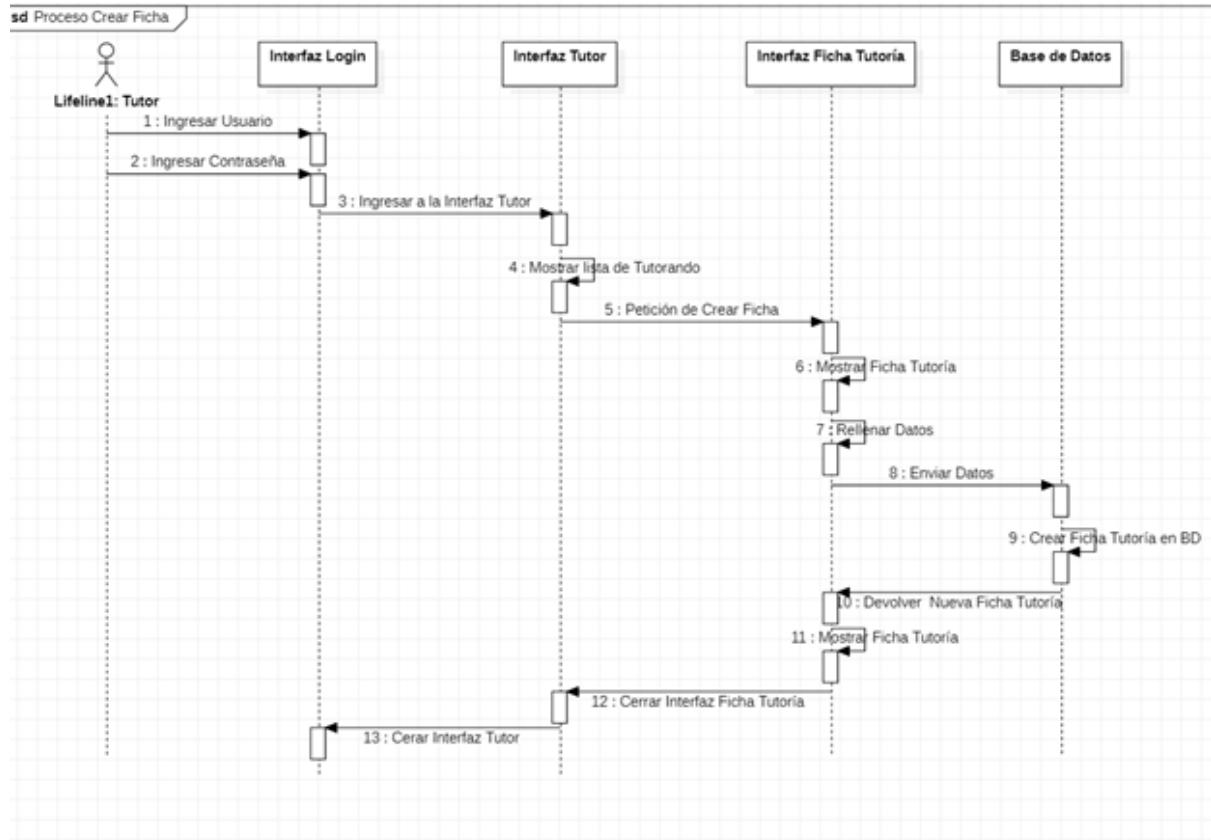


### **caso de uso: Agregar, modificar y eliminar Alumno**

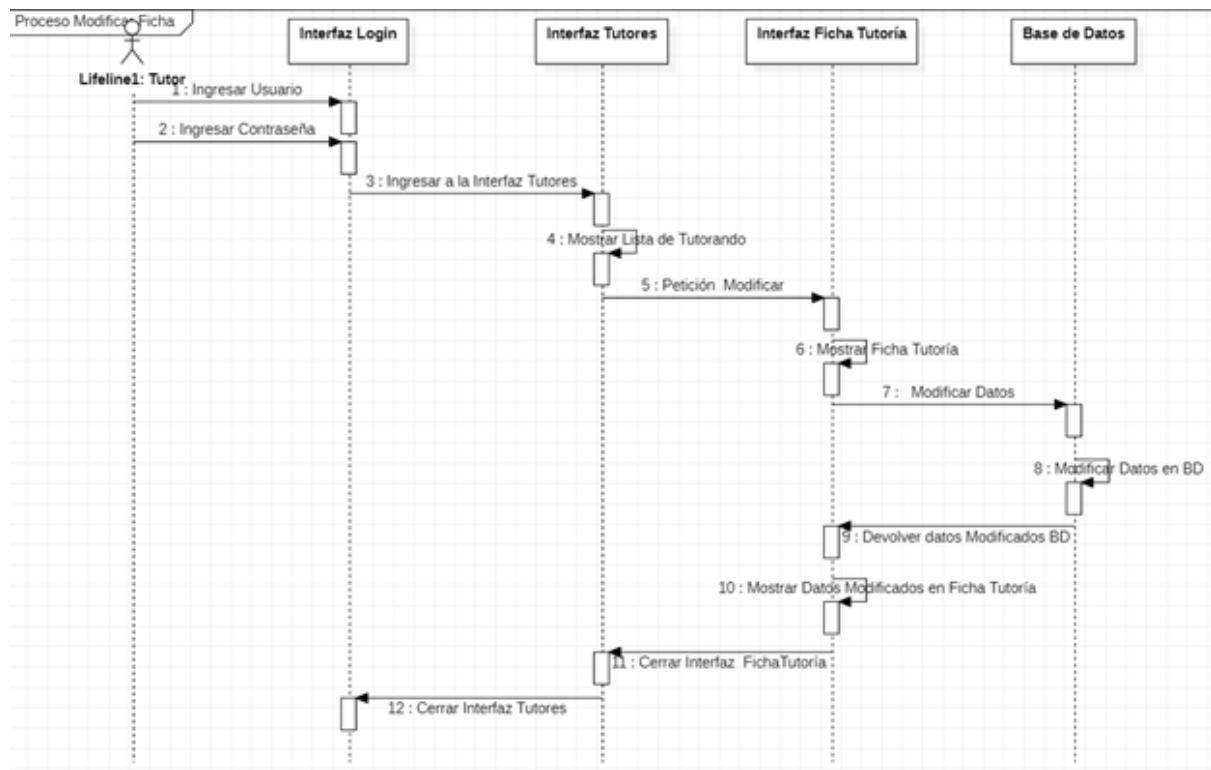


## **1.2 Diagrama de Secuencias**

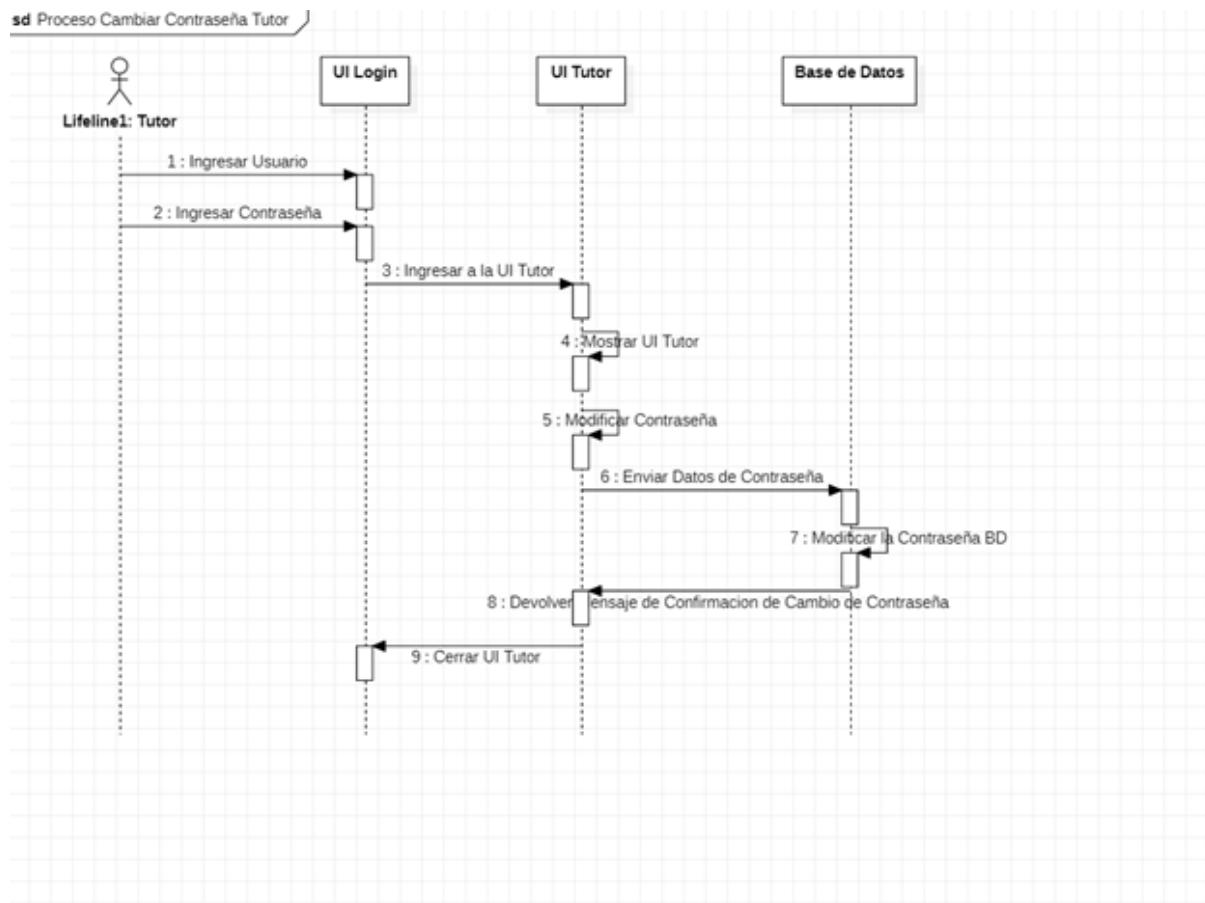
### **caso de uso: Crear Ficha Tutoría**



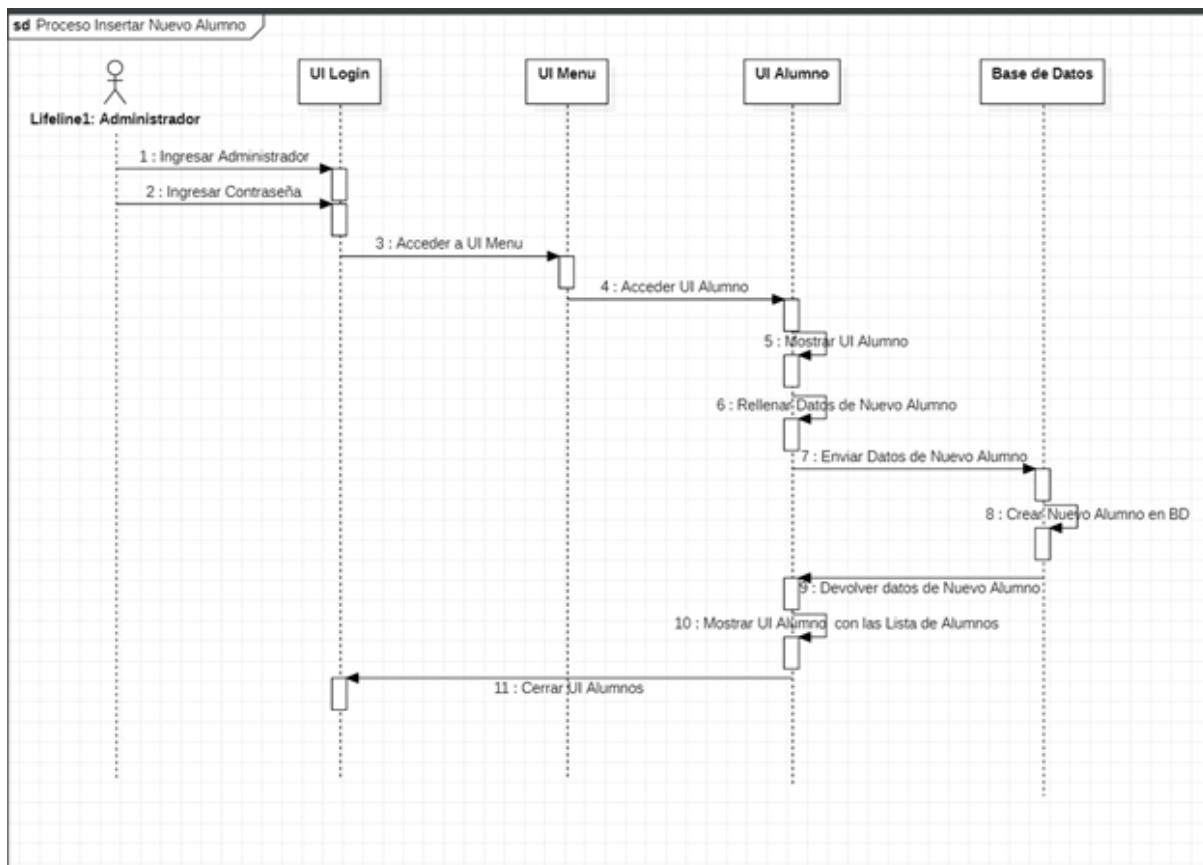
## caso de uso: Modificar Ficha Tutoría



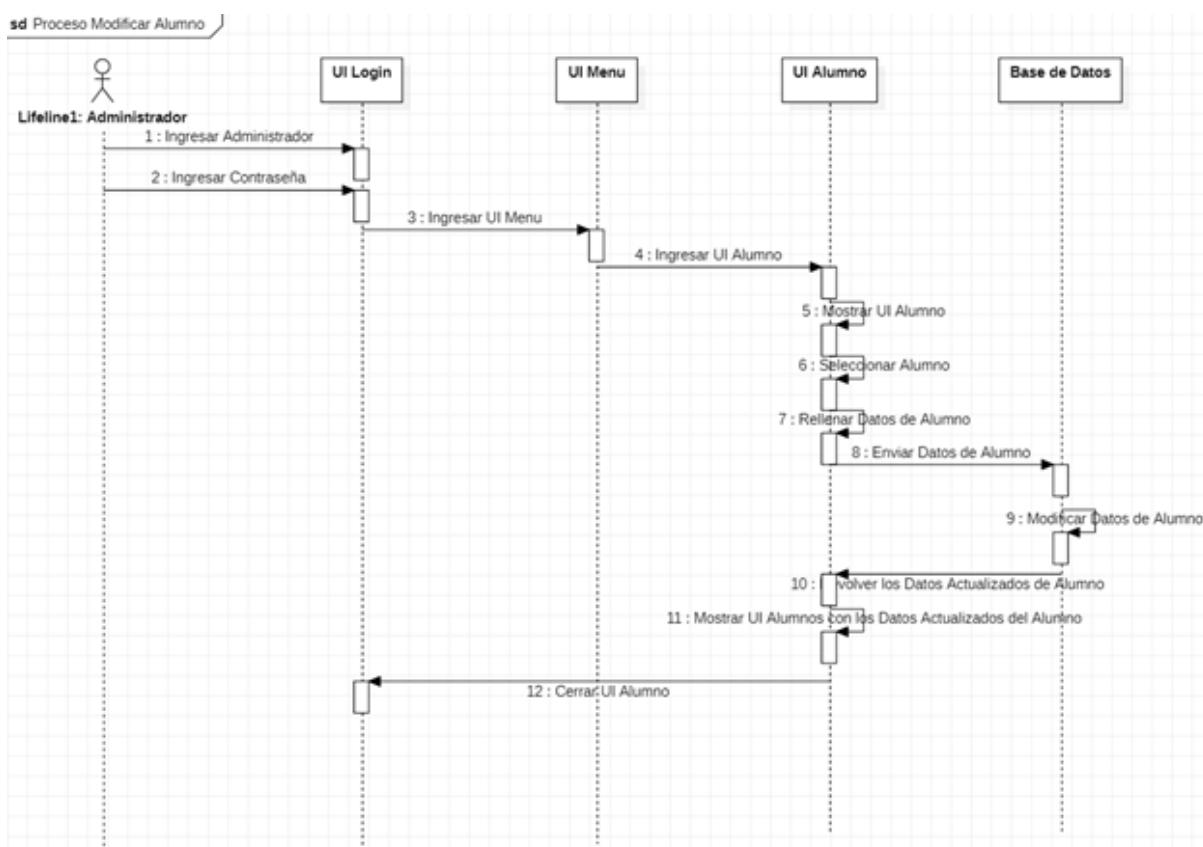
## caso de uso: Cambiar Contraseña de Tutor



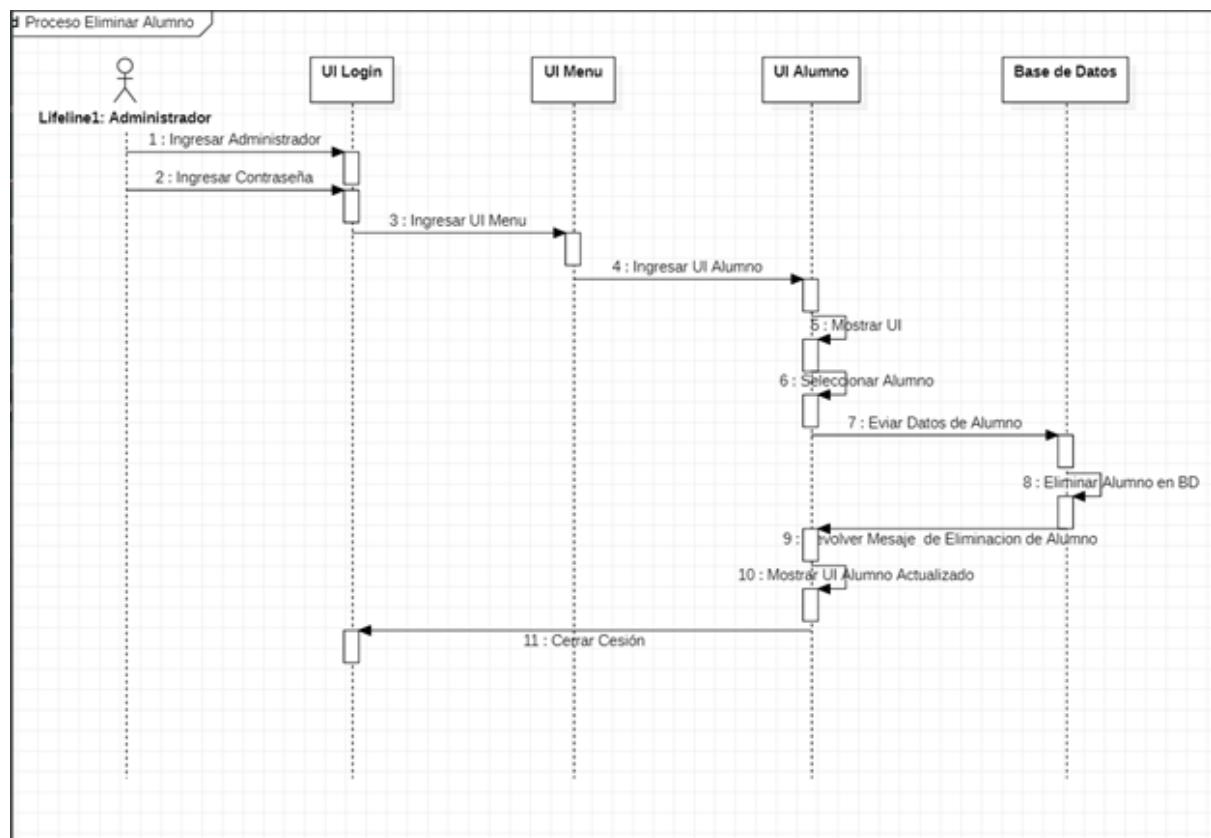
## caso de uso: Insertar Nuevo Alumno



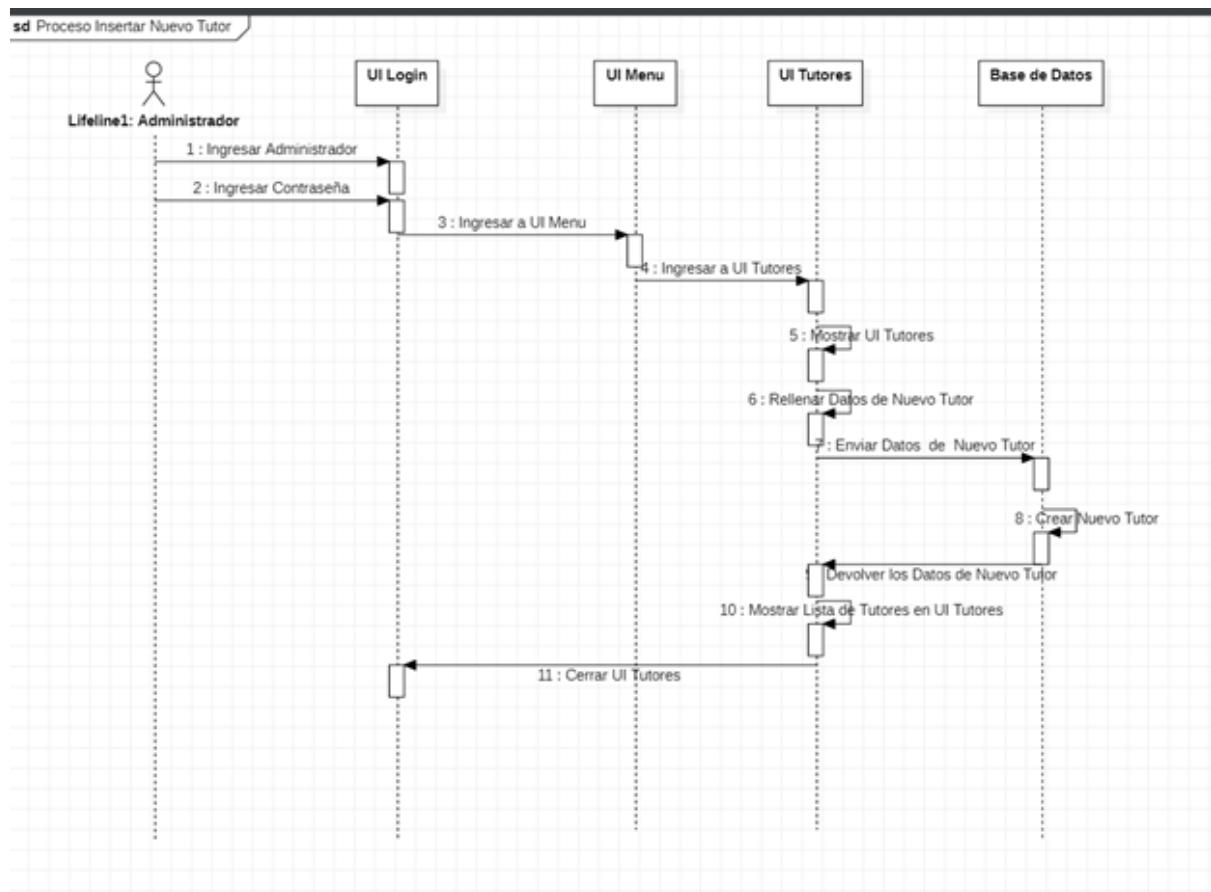
## caso de uso: Modificar Alumno



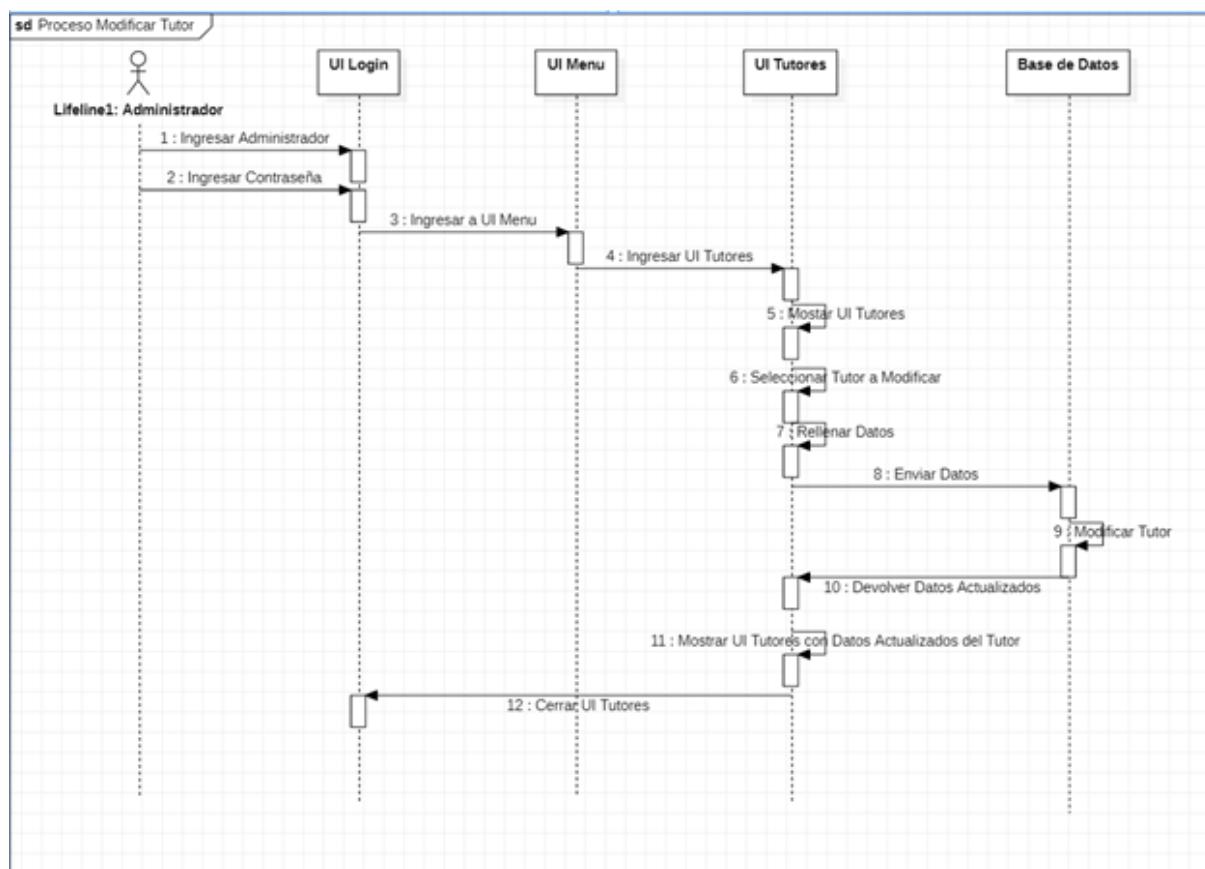
## caso de uso: Eliminar Alumno



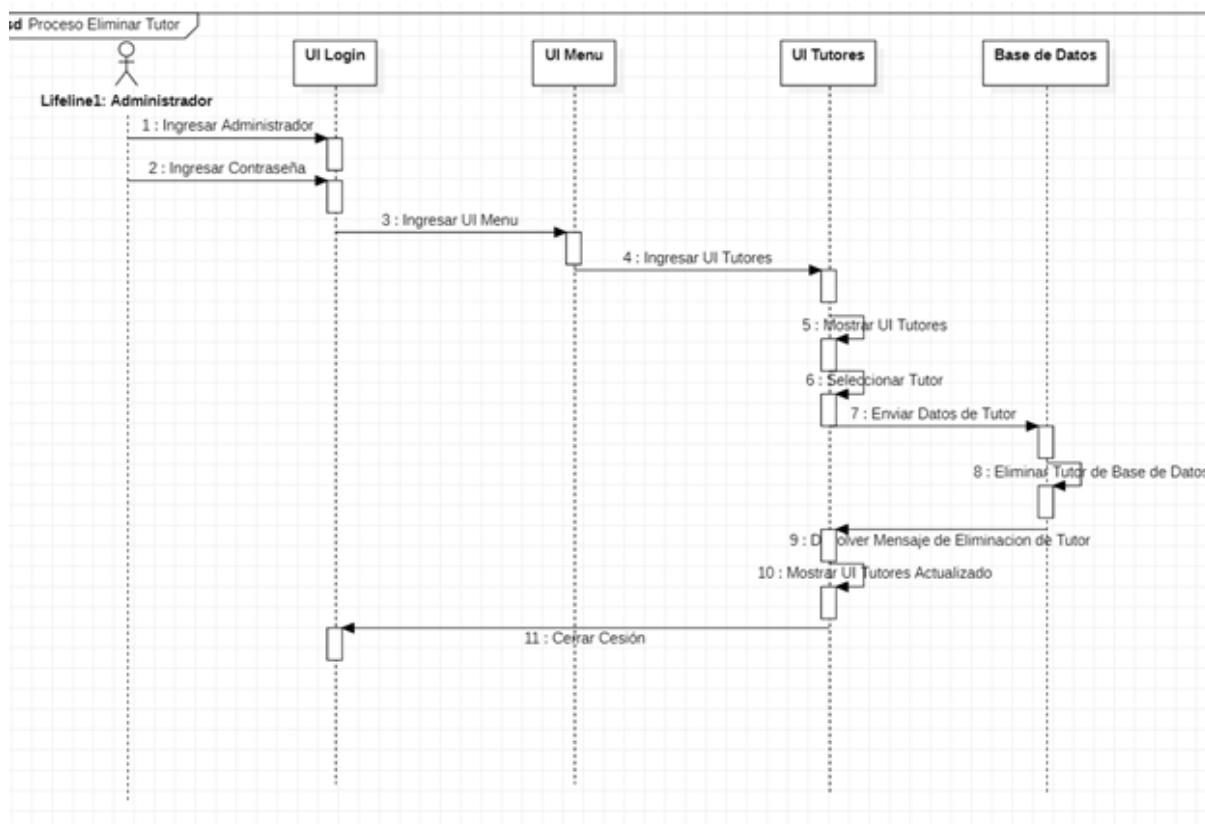
## caso de uso: Insertar Nuevo Tutor



### caso de uso: Modificar Tutor



## caso de uso: Eliminar Tutor



## **2. arquitectura utilizada**

La arquitectura candidata a utilizar será de 3 capas, donde el objetivo primordial es la separación de la capa de presentación, capa de negocio y la capa de datos. Como sabemos esta arquitectura es un modelo de desarrollo de software en el que el objetivo primordial es la separación de las partes que componen un sistema de software o también una arquitectura de cliente servidor.

### **2.1 Capa de presentación**

Es la que ve el usuario (también se le denomina “capa de usuario”), presenta al sistema al usuario, le comunica la información y captura la información del usuario en un mínimo de proceso. También es conocida como interfaz gráfica y debe tener la característica de ser amigable para el usuario. Esta capa se comunica únicamente con la capa de negocio. En la primera capa se encontrarán todas las interfaces como son interfaces para Administrador y Tutor así como el login.

### **2.2 Capa de negocios**

Es donde residen los programas que se ejecutan, se reciben las peticiones del usuario y se envían las respuestas tras el proceso. Denominamos capa de negocio porque es aquí donde se establecen todas las reglas que deben cumplir y se comunica con la capa de presentación, para recibir solicitudes y presentar los resultados, y con la capa de datos para solicitar al gestor de base de datos almacenar o recuperar datos de él. En la segunda capa se encontrarán las clases necesarias como Agregar , Editar, Eliminar Alumno ,etc.

### **2.3 Capa de datos**

Es dónde residen los datos y está encargado de acceder a los mismos. Formada por uno o más gestores de base de datos que realizan todo el almacenamiento o recuperación de información desde la capa de negocio. En la tercera capa se encuentra la base de datos con las tablas necesarias como Tutor, Alumnos, Ficha Tutoría, EscuelaProfesional,Administrador, etc.

## **3. Hardware y software**

### **Hardware**

En el proyecto se utilizó laptop personales cada integrante el cual se encargará de todas las acciones que necesita el software.

## **Software**

En el proyecto se utilizarán diferentes tipos de herramientas, a continuación haremos una mención con su detalle.

-Visual Studio: Entorno de desarrollo integrado que permite crear aplicaciones. Encargada del diseño de las interfaces y de la construcción del código correspondiente. Exactamente se utilizó C#.NET para la creación de formularios

-Microsoft SQL Server: Sistema de gestión de base de datos relacional. Encargado crear tablas, almacenar y recuperar la base de datos según lo solicitado del Visual Studio. Se uso SQL Server conectado al .NET ya que nos permitirá efectuar consultas con el fin de recuperar de forma sencilla la información de la base de datos así como mostrar cambios en ella.

-Star UML: Encargada de poder realizar los diagramas de casos de uso, diagrama de clases de análisis y diagrama de colaboración. Herramienta de gran utilidad.

Servicios en línea:

-Google Docs: Permite crear documentos y compartirlos con los otros usuarios. Es de gran utilidad para poder trabajar en equipo y poder documentar nuestro informe del proyecto a presentar.

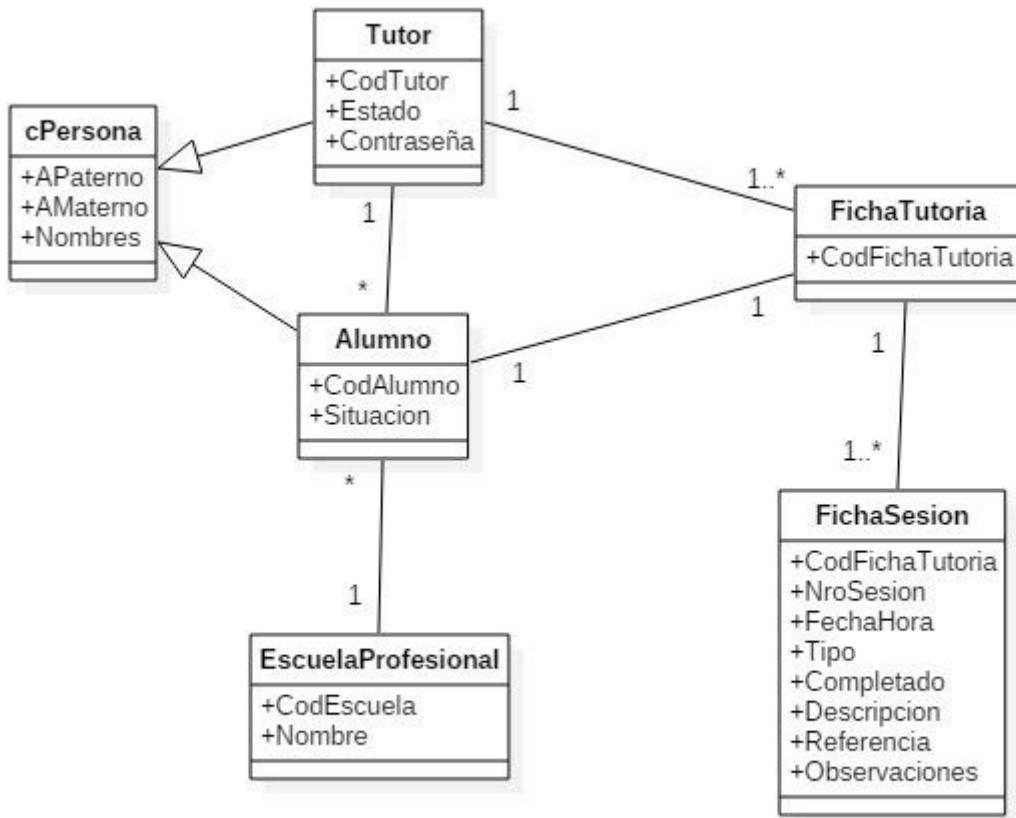
-Github: Repositorio online gratuito que permite gestionar proyectos y controlar versiones de código en la nube , este repositorio nos ayudo hacer modificaciones mediante ramas y generar nuevos códigos que sean visibles que nos permite tanto el control como la colaboración de todos los integrantes.

## **CAPÍTULO 3: Fase de construcción**

### **1. Implementación de la base de datos**

#### **1.1 Modelo conceptual de la base de datos**

Diagrama de clases en StarUML.



## 1.2 Script de la base de datos

Dividido en 5 archivos de propósito diferente:

- Archivo “Creation.sql”; este archivo nos servirá para crear las tablas de la base de datos. (Vista parcial)

```
--Tabla de tutor
create table Tutor
(
    CodTutor      char(3),
    APaterno     varchar(50),
    AMaterno     varchar(50),
    Nombres       varchar(30),
    Estado        varchar(40) check (Estado in('Contratado', 'Nombrado')),
    Contraseña   varbinary(20),
    primary key (CodTutor)
)
select CodTutor, APaterno, AMaterno, Nombres, Estado from Tutor
```

- Archivo “Triggers.sql”, servirá para tener un historial de cambios realizados en la base de datos. (Vista Parcial)

```
--delete alumno
create trigger trHistorialAlumnDelete
on Alumno for delete
as
declare @cod char(6)
select @cod = CodAlumno from deleted
insert into Historial values(system_user, getdate(), 'Alumno', 'delete', @cod)
```

3. Archivo “insert\_Docentes\_Alumnos.sql”, nos servirá para insertar todos los datos de Docentes Tutores y Alumnos en sus respectivas tablas como ejemplo para el correcto funcionamiento de la aplicación. (Vista Parcial)

```
insert into Tutor (CodTutor,Nombres,APaterno,AMaterno,Estado,Contraseña)
values
    ('001','Boris','Chullo','Llave','Nombrado',convert (varbinary,'1234')),
    ('002','Carlos Ramon','Quispe','Onofre','Nombrado',convert (varbinary,'1234')),
    ('003','Dennis Ivan','Candia','Oviedo','Nombrado',convert (varbinary,'1234')),
    ('004','Edwin','Carrasco','Poblete','Nombrado',convert (varbinary,'1234')),
    ('005','Emilio','Palomino','Olivera','Nombrado',convert (varbinary,'1234')),
    ('006','Enrique','Gamarra','Saldivar','Nombrado',convert (varbinary,'1234')),
```

```
insert into Alumno values
    ('112207','IBARRA','CASTILLO','WALDO ERICK','Riesgo','001','IN'),
    ('141660','AROSTEGUI','CERNA','JAIR FREDERICK','Riesgo','001','IN'),
    ('154856','HUAHUATICO','SORIA','RONALD','Riesgo','001','IN'),
    ('164563','CARPIO','HERMOZA','HAIDER ALEX','Riesgo','001','IN'),
    ('175101','QUISPE','ESCALANTE','CARLA','Riesgo','001','IN'),
    ('184212','VALENCIA','CUSIPUMA','LUIS MAO','Riesgo','001','IN'),
```

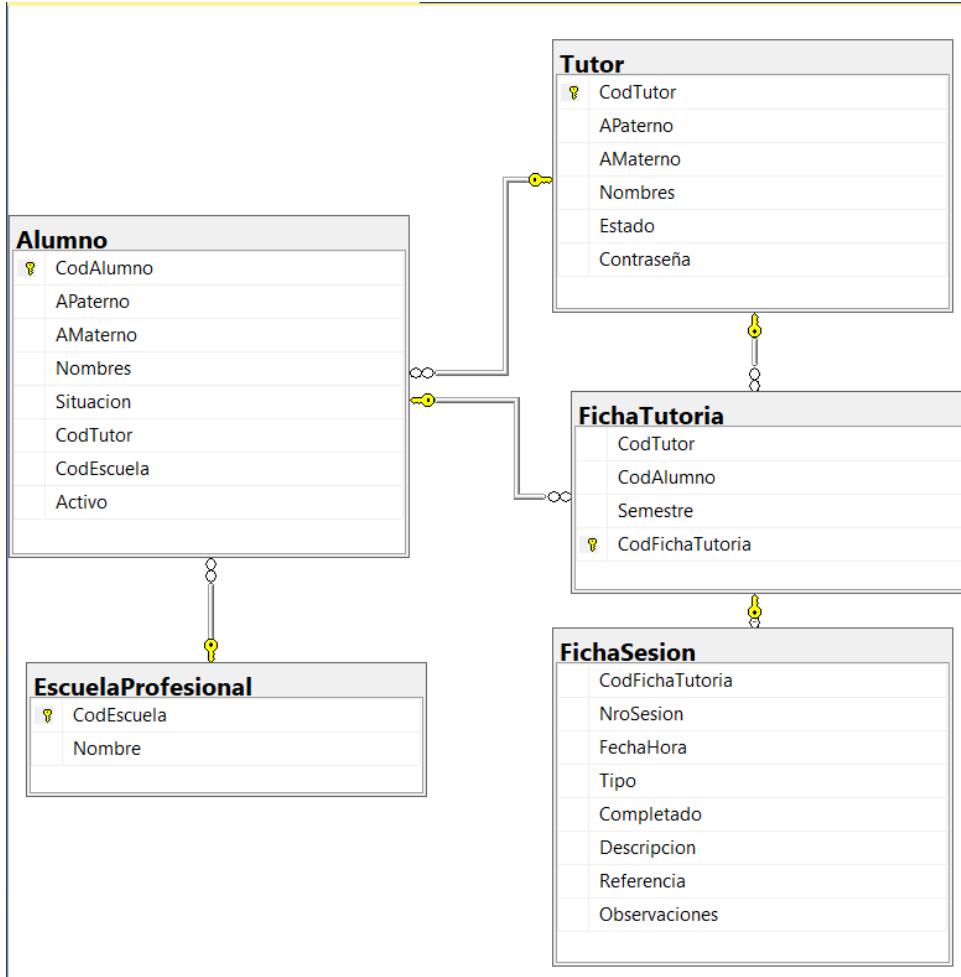
4. Archivo “InsertarAdminYFichaTutoria.sql”, administradores y ejemplo de fichas de tutoría.

5. Archivo “Procedimientos Almacenados.sql”. (Vista Parcial)

```
create proc ModificarAlumno
@CodAlumno varchar(15),
@APaterno varchar(50),
@AMaterno varchar(50),
@Nombres varchar(30),
@Situacion varchar(50),
@CodTutor char(3),
@CodEscuela varchar(5)
as
update Alumno set CodAlumno = @CodAlumno,
                    APaterno = @APaterno,
                    AMaterno = @AMaterno,
                    Nombres = @Nombres,
                    Situacion = @Situacion,
                    CodTutor = @CodTutor,
                    CodEscuela = @CodEscuela
where CodAlumno = @CodAlumno
```

### 1.3 Diagrama de la base de datos

- Tablas de propósito general:



- Tabla Administración:



- Tabla Historial:

Historial	
NroOperacion	
Editor	
Fecha	
Tabla	
Operacion	
TuplaAfectada	

## 1.4 Base de Datos en la Nube

### 1.4.1 Creación de BD en azure:

Para crear la base de datos seguiremos los siguientes pasos:

1. Iniciamos sesión en Azure
2. En el buscador, escribimos SQL Database y seleccionamos esa opción.

The screenshot shows the Azure portal search interface. The search bar at the top contains the text "sql database". Below the search bar, there are three main sections: "Servicios", "Marketplace", and "Documentación".

- Servicios:** This section lists several Azure services related to databases:
  - SQL Database** (highlighted with a gray background)
  - Grupos de SQL dedicados (antes SQL DW)
  - Bases de datos de extensión de SQL Server
  - Servidores de Azure Database for MySQL
  - Reservas
  - Azure Database for MySQL flexible servers
  - Azure Database for PostgreSQL server groups
  - Servidores de Azure Database for PostgreSQL
  - Servidores flexibles de Azure Database for PostgreSQL
  - Azure Database Migration Projects
- Marketplace:** This section lists marketplace offerings related to SQL Database:
  - Striim for Real-Time Integration to SQL Database
  - SQL Database Reserved vCores
  - EDISPHERE Private Instance using MS-SQL Database
  - CockroachDB
- Documentación:** This section links to documentation pages:
  - Documentación de Azure SQL Database - Azure SQL Database ...
  - ¿Qué es el servicio Azure SQL Database? - Azure SQL ...
  - SQL Server a Azure SQL Database: guía de migración - Azure ...
  - Replicación de Azure SQL Server en Azure SQL Database ...

At the bottom of the search results, there are two sections: "Recursos" and "Grupos de recursos", both of which indicate "No se han encontrado resultados." (No results found).

Below the search results, there is a message: "No ha encontrado lo que estaba buscando?" (Did you find what you were looking for?). It includes two options: "Intentar buscar en el registro de actividades" (Try searching in the activity log) and "Intentar la búsqueda en Azure Active Directory" (Try searching in Azure Active Directory). A note at the bottom says "Buscando en todas las suscripciones. Cambiar" (Searching across all subscriptions. Change).

3. Damos click en crear

Inicio >

## SQL Database

Directorio predeterminado

+ Crear Reservas Administrar vista Actualizar Exportar a CSV Abrir consulta Asignar etiquetas Eliminar Comentarios

Filtrar por cualquier campo... Suscripción == todo Grupo de recursos == todo Ubicación == todo Agregar filtro

Mostrando de 1 a 1 de 1 registros.

Nombre ↑↓	Servidor ↑↓	Tipo de réplica ↑↓	Plan de tarifa ↑↓	Ubicación ↑↓
<input type="checkbox"/> BD_Tutoria (serverproyectotutoria/BD_Tutoria)	serverproye...	--	General Purpose: G...	Este de EE. UU.

4. Escogemos la opción de Básico y llenamos el campo Grupo de recursos con el nombre de nuestra base de datos.

### Crear base de datos SQL

Microsoft

Básico Redes Seguridad Configuración adicional Etiquetas Revisar y crear

Cree una base de datos SQL con la configuración que prefiera. Complete la pestaña de configuración básica y, a continuación, vaya a Revisar y crear para efectuar el aprovisionamiento con valores predeterminados automáticos, o bien visite cada pestaña para personalizarlos. [Más información](#)

#### Detalles del proyecto

Seleccione la suscripción para administrar recursos implementados y los costes. Use los grupos de recursos como carpetas para organizar y administrar todos los recursos.

Suscripción \*  Azure para estudiantes

Grupo de recursos \*  Proyecto\_Tutoria\_DS

[Crear nuevo](#)

#### Detalles de la base de datos

Indique la configuración necesaria para esta de los recursos de proceso y almacenamiento.

Nombre de la base de datos \*

Servidor \*

Un grupo de recursos es un contenedor que tiene los recursos relacionados de una solución de Azure.

Nombre \*

Aceptar

Cancelar

5. En la sección de Detalles de la base de datos, creamos un nuevo servidor con los parámetros deseados.

## Nuevo servidor

Microsoft

X

Nombre del servidor \*

Escriba un nombre de servidor

.database.windows.net

Inicio de sesión del administrador del servidor \*

Escribir inicio de sesión del administrador ...

Contraseña \*

Confirmar contraseña \*

Ubicación \*

(US) Centro-Sur de EE. UU.

▼

6. Dar click al botón de revisar y crear, luego click en el botón de crear.
7. Esperar hasta que la implementación esté completa.
8. Dar click en el botón Ir al recurso.
9. Se muestra la información general del proyecto.

The screenshot shows the Microsoft Azure portal interface for a SQL Database named 'BD\_Tutoria'. The left sidebar lists navigation options like Inicio, SQL Database, Power Platform, and Configuración. The main content area displays the database's general information, including its name, connection status, location (Este de EE. UU.), and subscription details. A chart titled 'Uso de procesos' (Process usage) shows activity over the last hour, with a significant spike around 21:15 UTC. On the right, there's a summary of storage usage: 0,07% of the assigned 32 MB is used, resulting in 22 MB of space used.

10. En la opción Cadenas de conexión de la barra izquierda se puede visualizar el comando para realizar la conexión de la base de datos.

BD\_Tutoria (serverproyectotutoria/BD\_Tutoria) | Cadenas de conexión

ADO.NET JDBC ODBC PHP Go

ADO.NET (autenticación de SQL)

```
Server=tcp:serverproyectotutoria.database.windows.net,1433;Initial Catalog=BD_Tutoria;Persist Security Info=False;User ID=admin1;Password=(your_password);MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;
```

Descargar controlador ADO.NET para SQL Server

11. Para que los integrantes del proyecto tengan acceso a la base de datos, es necesario agregar sus respectivas direcciones IP en el área de Establecer Firewall de servidores, dentro de la sección de información general.

BD\_Tutoria (serverproyectotutoria/BD\_Tutoria)

Información general Registro de actividad Etiquetas Diagnóstico y solución de problemas Inicio rápido Editor de consultas (versión preliminar)

Información esencial

Grupo de recur... (cambiar) : Proyecto\_Tutoria\_DS Estado : Conectado Ubicación : Este de EE. UU. Suscripción (cambiar) : Azure para estudiantes Id. de suscripción : 4f666176-fd69-461a-9848-3ce33594f6cf Etiquetas (cambiar) : Haga clic aquí para agregar etiquetas.

Nombre del servidor : serverpr... Grupo elástico : Ningún... Cadenas de conexión : Mostrar Plan de tarifa : Uso gen... Punto de restauración m... : 2021-08-01 00:00:00

Mostrar datos del último período de: 1 hora 24 horas 7 días

Uso de procesos

Configuración de firewall

serverproyectotutoria (SQL server)

Guardar Descartar + Agregar IP de cliente

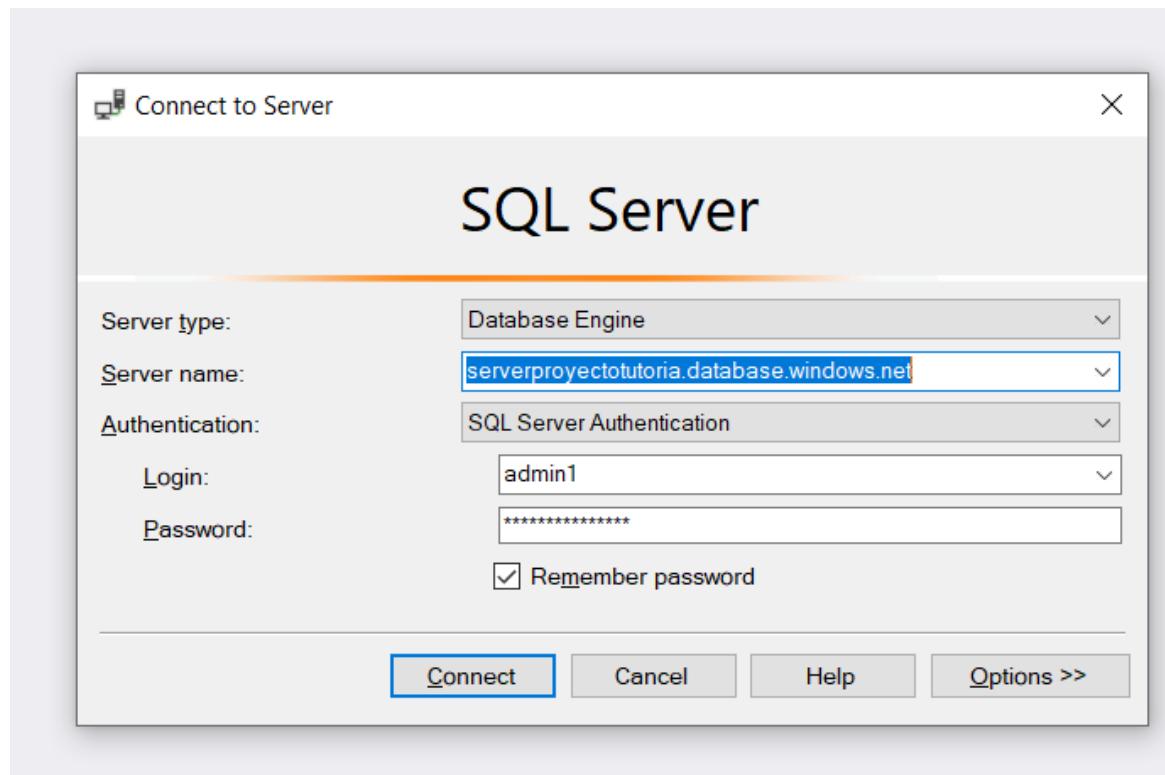
Direciva de conexión: Predeterminado Proxy Redirigir

Permitir que los servicios y recursos de Azure accedan a este servidor: Sí

Dirección IP de cliente: 179.6.34.232

Nombre de regla	IP inicial	IP final	
laptop_Aler	179.6.34.232	179.6.34.232	...
laptop_Enrique	181.67.104.206	181.67.104.206	...
laptop_Fiorella	179.6.34.5	179.6.34.5	...
laptop_Jhon	190.108.87.245	190.108.87.245	...
laptop_Luz_Lucero	179.6.34.172	179.6.34.172	...
laptop_yurema	181.67.72.231	181.67.72.231	...

12. Para administrar la base de datos desde SQL server, se debe colocar el nombre del servidor, la autenticación SQL server Authentication , usuario y contraseña. Si la dirección IP del ordenador es alguna de las direcciones previamente registradas, se permite la conexión.



Connect ▾

- serverproyectotutoria.database.windows.net (SQL Server 12.0.2000.8 - admin1)
  - Databases
    - System Databases
    - BD\_Tutoria
      - Database Diagrams
      - Tables
      - Views
      - External Resources
      - Synonyms
      - Programmability
      - Query Store
      - Extended Events
      - Storage
      - Security
    - Security
    - Integration Services Catalogs

13. Se ejecutan las consultas necesarias para la implementación de las tablas, usando la nueva base de datos.
14. En el proyecto reemplazar la línea de conexión con la siguiente línea: “`SqlConnection conn = new SqlConnection("Data Source=tcp:serverproyectotutoria.database.windows.net;Initial Catalog=BD_Tutoria;UserID=admin1;Password=Grupo2DesarrolloSoftware");`”

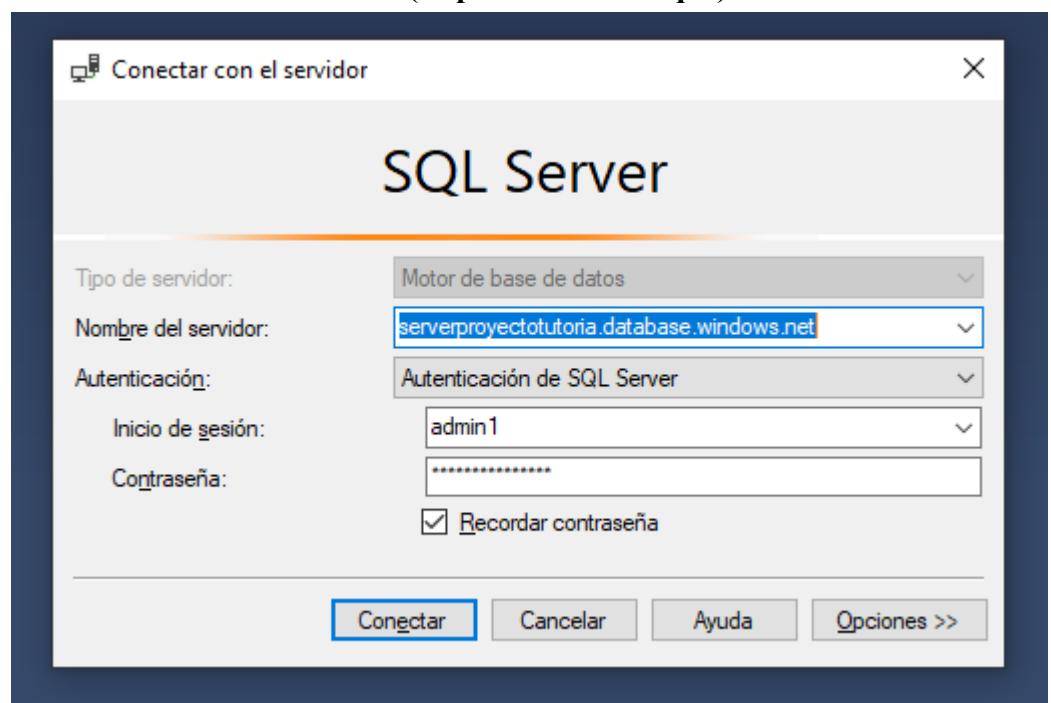
```
namespace Datos
{
    1 referencia
    public class ConexionSQL
    {
        SqlConnection conn = new SqlConnection("Data Source=tcp:serverproyectotutoria.database.windows.net;
//SqlConnection conn = new SqlConnection("Data Source=LAPTOP-IUT020T4;Initial Catalog=Tutoria;Integ
```

#### 1.4.2 Pruebas:

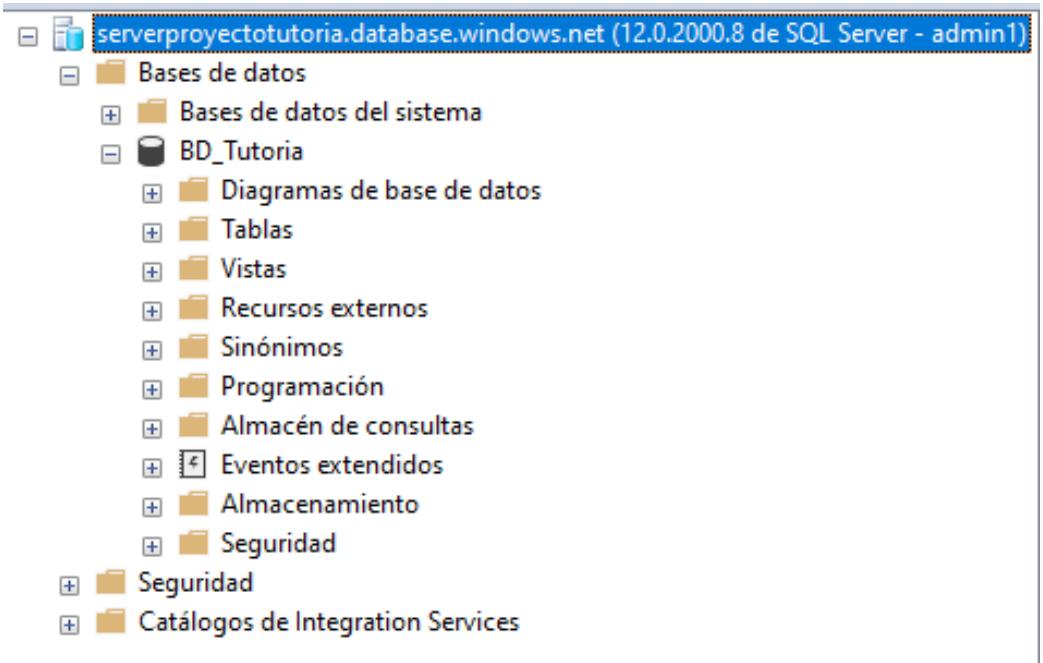
Para comprobar el correcto funcionamiento de la base de datos en la nube, realizaremos algunas pruebas en dos máquinas cuyas direcciones IP públicas fueron previamente agregadas en la configuración del firewall:

##### 1. Conectar a servidor en la nube:

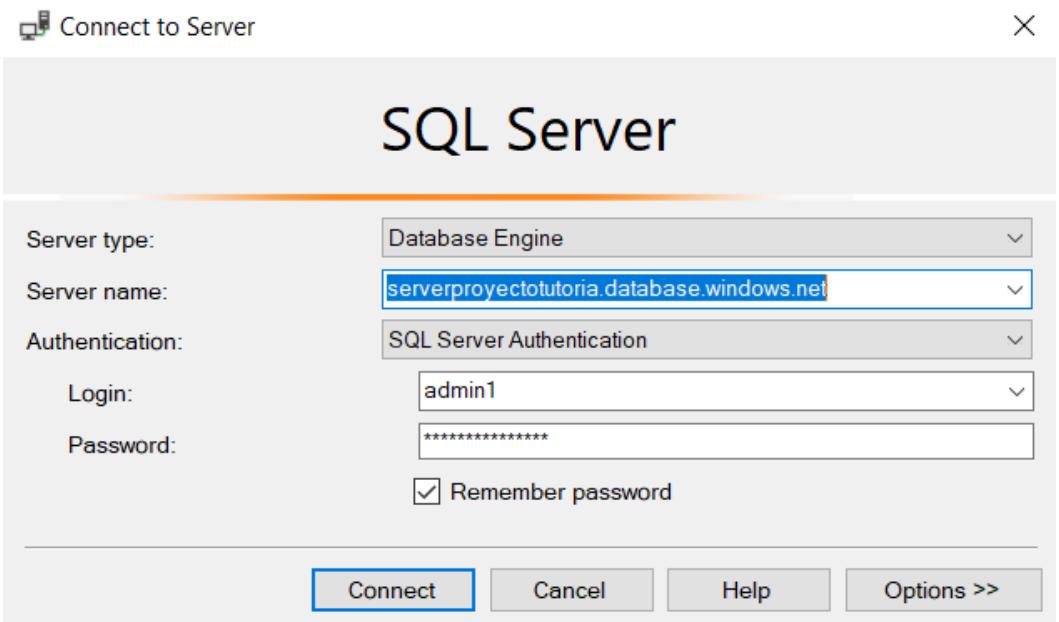
- 181.67.104.206 (IP pública de Enrique)



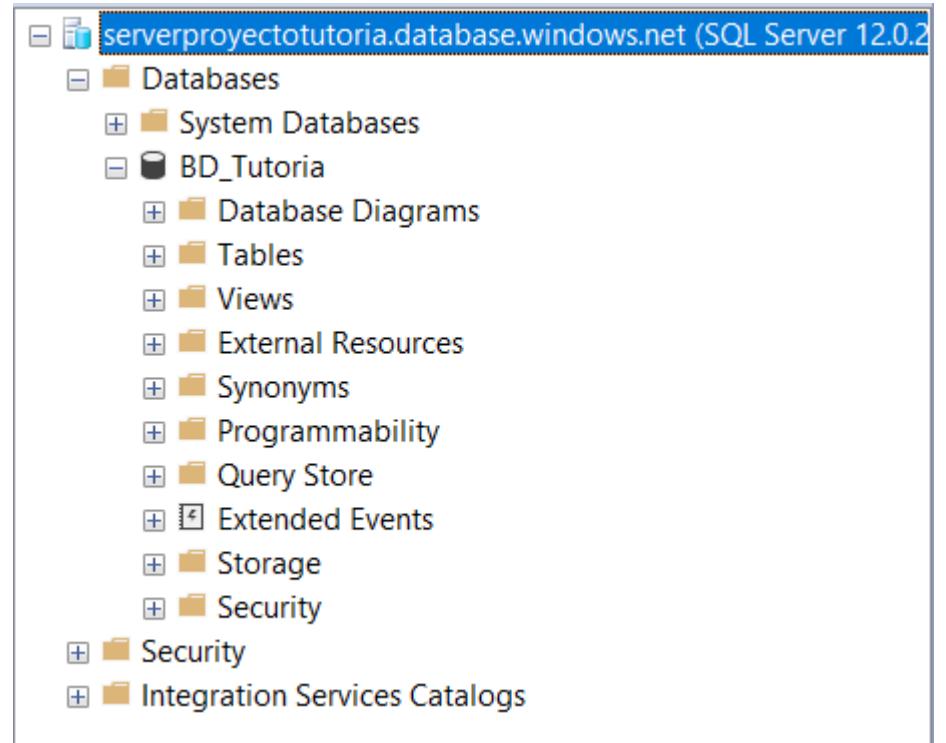
Como esta IP fue agregada se conectará de forma exitosa



- 179.6.35.45(IP pública de Miguel)



Como esta IP fue agregada se conectará de forma exitosa



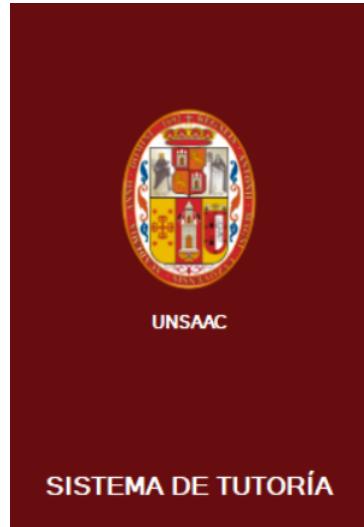
La base de datos ya está conectada, ahora realizaremos algunas pruebas dentro del programa.

## 2. Inicio de sesión:

En la máquina 181.67.104.206 (IP pública de Enrique), el inicio de sesión fue exitoso:



En la máquina 179.6.35.45(IP pública de Miguel), el inicio de sesión fue exitoso.:



# LOGIN

admin1

contrasenia1



**Entrar**

Ahora haremos algunas modificaciones en las tablas.

### 3. Crear nuevo alumno:

En la Máquina 181.67.104.206 (IP pública de Enrique), agregamos un nuevo alumno con estos datos:

## Alumnos

Codigo	219999	Ap. Paterno	Mendoza	Ap. Materno	Rivas
Nombres	Jose	Escuela P.	IN	<input type="checkbox"/> En riesgo	
Cod. Tutor	017	<b>AGREGAR</b>			

Se agregó exitosamente:

---

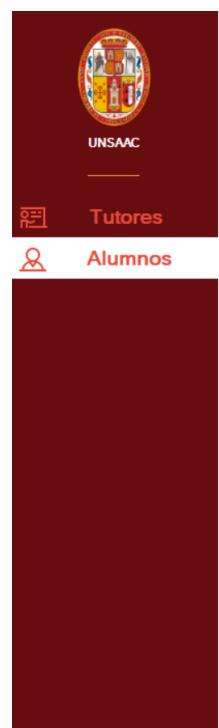
Campo:

APaterno	AMaterno	Nombres	Sit
LAYME		X NELSON	
MAMANI		AI ISAAC	
NAHUAMEL		NIS ANGEL	
DURAND		A SHIRLEY	
ARONI		REN STEP...	
GUTIERREZ	AMACHI	JUAN CARLOS	
CASTILLO	LOPEZ	RICARDO JORGE	
ORUE	QUISPE	ALVARO AMERI...	
CHOCUE	CCOA	DENNIS ALDIO	

Agregado Correctamente

**Aceptar**

En la máquina 179.6.35.45(IP pública de Miguel), se comprueba que se hizo el nuevo registro exitosamente.



The screenshot shows the UNSAAC Alumnos application interface. At the top left is the UNSAAC logo. Below it are tabs for 'Tutores' and 'Alumnos'. The main area has fields for 'Codigo', 'Ap. Paterno', 'Ap. Materno', 'Nombres', 'Escuela P.', 'Cod. Tutor', and 'En riesgo'. A red 'AGREGAR' button is at the bottom right. Below these fields is a table with columns: CodAlumno, APaterno, AMaterno, Nombres, Situacion, CodTutor, and CodEscuela. The first row of the table contains the values: 219999, Mendoza, Rivas, Jose, No Riesgo, 017, and IN.

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
219999	Mendoza	Rivas	Jose	No Riesgo	017	IN

#### 4. Editar un alumno:

En la máquina 181.67.104.206 (IP pública de Enrique), editaremos los datos del alumno agregado anteriormente:



The screenshot shows the UNSAAC Alumnos application interface. It displays a table with columns: CodAlumno, APaterno, AMaterno, Nombres, Situacion, CodTutor, and CodEscuela. The first row of the table contains the values: 219999, Mendoza, Rivas, Jose, No Riesgo, 017, and IN. Above the table, there is a search bar with 'CodAlumno' set to '219999' and a magnifying glass icon.

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
219999	Mendoza	Rivas	Jose	No Riesgo	017	IN

Modificaremos sus nombre Jose por Jose Luis y guardamos:

## Alumnos



The screenshot shows the UNSAAC Alumnos application interface. It displays a form with fields: Codigo (219999), Ap. Paterno (Mendoza), Ap. Materno (Rivas), Nombres (Jose Luis), Escuela P. (IN), and Cod. Tutor (017). Below the form are 'CANCELAR' and 'EDITAR' buttons.

En la máquina con IP 179.6.35.45(IP pública de Miguel), se comprueba que se hizo la modificación exitosamente.



## Alumnos

Código:  Ap. Paterno:  Ap. Materno:   
 Nombres:  Escuela P.:   En riesgo  
 Cod. Tutor:  **AGREGAR**

**Alumnos**

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEsc
219999	Mendoza	Rivas	Jose Luis	No Riesgo	017	IN

### 5. Eliminar alumno:

En la máquina 181.67.104.206 (IP pública de Enrique), Por último eliminaremos el alumno editado anteriormente:

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEsc
219999	Mendoza	Rivas	Jose Luis	No Riesgo	017	IN

Campo:  2199

**Eliminar Alumno**

Seguro que desea Eliminar Alumno de código: 219999?

Campo:  2199

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEsc
219999	Mendoza	Rivas	Jose Luis	No Riesgo	017	IN

Eliminado

En la máquina con IP 179.6.35.45(IP pública de Miguel), se comprueba que se eliminó el registro exitosamente exitosamente.

**Alumnos**

Código	<input type="text"/>	Ap. Paterno	<input type="text"/>	Ap. Materno	<input type="text"/>
Nombres	<input type="text"/>	Escuela P.	<input type="text"/>	<input type="checkbox"/> En riesgo	
Cod. Tutor	<input type="text"/>				

**AGREGAR**

Campo:

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	Cod

**Alumnos**

## 2. Implementación y funcionalidad

### 2.1 Login:

El programa iniciará con una ventana la cual invita al usuario a ingresar su nombre de usuario y contraseña. El usuario que use el programa puede ser un docente tutor de la carrera profesional de Ingeniería Informática y de Sistemas o un Administrador el cual podrá realizar operaciones CRUD.

### 2.2 Como docente:

El docente tutor tendrá asignada una contraseña única la cual le permitirá ingresar a la interfaz en la que se muestran sus tutorados y sesiones de tutoría respectivas.

**LOGIN**

Ingresando como Docente

Usuario

Contraseña

**Entrar**

### 2.3 Como administrador:

El administrador tendrá una contraseña única la cual le permitirá ingresar a la interfaz que le permita hacer operaciones CRUD ya sea sobre tutores o alumnos.

## 2.4 Interfaz de Administrador

Una vez se hayan validado el nombre de usuario y contraseña del administrador, este accede a la interfaz en la que puede realizar operaciones CRUD ya sea sobre docentes o sobre alumnos, en la cual tiene las siguientes opciones:

### 2.4.1 Control de Alumnos

CodNummo	APaterno	AMaterno	Nombres	Situación	CodTutor	CodEscuela
012345	PINAES	BUSTAMANTE	SAMMY YASSER	Riesgo	003	IN
012346	LIMPE	ZEVALLOS	LUIS ANDRES	Riesgo	004	IN
012347	PIUMA	HEJILCA	RICHARD JESUS	Riesgo	005	IN
012348	YARÉZ	SANZ	ANTHONY JOS...	No riesgo	003	IN
012349	OCAHUANTICO	MENDOZA	JULIO CESAR	No riesgo	004	IN
012350	BALLON	SEGOVIA	HENRY LUIS	Riesgo	006	IN
012351	MOREANO	BRICERO	GROVER	No riesgo	005	IN
012352	HUACANI	DÉ LA CRUZ	DANY DARWIN	No riesgo	006	IN
012353	TRIJULLO	TORBISCO	LUIS ANDERSON	Riesgo	007	IN
012354	VARGAS	QUISPE	CHARLY	Riesgo	009	IN
012355	RAMIREZ	APAZA	EBERT	Riesgo	012	IN
012356	VALDEZ	CGUNO	JULIO CESAR	Riesgo	013	IN
012357	CHARA	TACURI	CESAR	Riesgo	014	IN
012358	HUMANAS	FLOREZ	RONALD	Riesgo	016	IN
012359	AVILA	MAURICIO	NEI SPAI	Riesgo	017	IN

#### Características:

- Se puede agregar un nuevo alumno siempre y cuando este tenga un código distinto a los que ya se encuentran en la base de datos, además, el código del tutor también debe estar en la base de datos.

- Se puede hacer una búsqueda por CodAlumno, APaterno, AMaterno, Nombres, Situacion, Cod Tutor y CodEscuela haciendo click en cualquiera de los ítems del ComboBox.
- Se puede modificar un alumno haciendo click en el lápiz verde que se encuentra en la misma fila de sus datos; se mostrarán los datos en los textbox ubicados en la parte superior, pero no se podrá modificar el código por ningún motivo.
- Se puede eliminar cualquier alumno haciendo click en el tacho rojo que se encuentra en la misma fila de sus datos; para esta función, siempre aparecerá un mensaje de alerta en el cual el administrador debe confirmar su acción.

#### 2.4.2 Control de Tutores

Código	Ap. Paterno	Ap. Materno	Nombre	Estado
001	Quique	Oviedo	Carlos Ramon	Nombrado
002	Candia	Oviedo	Dennis Ivan	Nombrado
003	Carrasco	Poblete	Edwin	Nombrado
004	Palomino	Olivera	Emilio	Nombrado
005	Gamana	Saldívar	Enrique	Nombrado
006	Pacheco	Vasquez	Ester	Nombrado
007	Cutipa	Anapa	Gladys Ehtana	Nombrado
008	Ticona	Pani	Guaman	Nombrado
009	Ugarte	Rojas	Hector Eduardo	Nombrado
010	Nina	Hancio	Horman	Nombrado
011	Cesar	Medrano	Ivan	Nombrado
012	Rosas	Huscho	Javier Arturo	Nombrado
013	Chavez	Centeno	Javier David	Nombrado
014	Rivas	Brenes	Inna I. rive	Nombrado
015				

#### Características:

- Se puede agregar un nuevo tutor siempre y cuando este tenga un código distinto a los que ya se encuentran en la base de datos.
- Se puede hacer una búsqueda por CodTutor, APaterno, AMaterno, Nombres y Estado haciendo click en cualquiera de los ítems del ComboBox.
- Se puede modificar un tutor haciendo click en el lápiz verde que se encuentra en la misma fila de sus datos; se mostrarán los datos en los textbox ubicados en

la parte superior, pero no se podrá modificar el código por ningún motivo.

- Se puede eliminar cualquier tutor haciendo click en el tacho rojo que se encuentra en la misma fila de sus datos; para esta función, siempre aparecerá un mensaje de alerta en el cual el administrador debe confirmar su acción.

### 3. Interfaz de Tutor

La interfaz del Tutor tiene la siguiente estructura:

The screenshot shows a user interface for a tutor. On the left, there is a sidebar with the UNSAAC logo at the top, followed by 'Informacion Personal:' and three fields: 'Nombre: Boris', 'Apellidos: Chullo Llave', and 'Codigo: 001'. At the bottom of this sidebar is a 'Cambiar Contraseña' button. The main area is titled 'Tutorados' and contains a table of student data. The table has columns: CodAlumno, APaterno, AMaterno, Nombres, Situacion, CodTutor, and CodEscuela. The data is as follows:

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
112207	IBARRA	CASTILLO	WALDO ERICK	Riesgo	001	IN
125156	CONDORI	FLORES	WILLIAMS DEN...	No riesgo	001	IN
141660	AROSTEGUI	CERNA	JAIR FREDERICK	Riesgo	001	IN
150388	CHACON	VARGAS	JUAN ANTONIO	No riesgo	001	IN
154856	HUAHUATICO	SORIA	RONALD	Riesgo	001	IN
160886	CACERES	LOAYZA	MARIA LUISA	No riesgo	001	IN
164240	ESCOBEDO	DURAN	ROBERTO CAR...	No riesgo	001	IN
164563	CARPIO	HERMOZA	HAIDER ALEX	Riesgo	001	IN
164817	HUAYHUA	JURADO	TANY ARISTID...	Riesgo	001	IN
171571	ROA	LIMACHI	JHON NILSON	No riesgo	001	IN
175101	QUISPE	ESCALANTE	CARLA	Riesgo	001	IN
182901	COLLANTE	CARRASCO	ALBERTO	No riesgo	001	IN
184212	VALENCIA	CUSIPUMA	LUIS MAO	Riesgo	001	IN
184655	SANCA	ZEVALLOS	JERY	No riesgo	001	IN
194525	PAREDES	CURASCO	ARELI SHALON	No riesgo	001	IN
201228	ACHAHUI	CRUZ	MILTON AMED	No riesgo	001	IN

Inicialmente tenemos un panel de información donde se muestran los datos personales del Tutor.

This screenshot shows a panel titled 'Informacion Personal:' containing the following data:  
Nombre: Boris  
Apellidos: Chullo Llave  
Codigo: 001

Seguidamente se encuentra una sección para cambiar la visibilidad de los alumnos, si quiere ver solo la lista de los tutorados o la lista de todos los alumnos.

## Tutorados

Ver solo tutorados

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
112207	IBARRA	CASTILLO	WALDO ERICK	Riesgo	001	IN
125156	CONDORI	FLORES	WILLIAMS DEN...	No riesgo	001	IN
141660	AROSTEGUI	CERNA	JAIR FREDERICK	Riesgo	001	IN
150388	CHACON	VARGAS	JUAN ANTONIO	No riesgo	001	IN
154856	HUAHUATICO	SORIA	RONALD	Riesgo	001	IN
160886	CACERES	LOAYZA	MARIA LUISA	No riesgo	001	IN
164240	ESCOBEDO	DURAN	ROBERTO CAR...	No riesgo	001	IN
164563	CARPIO	HERMOZA	HAIDER ALEX	Riesgo	001	IN
164817	HUAYHUA	JURADO	TANY ARISTID...	Riesgo	001	IN
171571	ROA	LIMACHI	JHON NILSON	No riesgo	001	IN

## Todos los Alumnos

Ver solo tutorados

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
012117	PINARES	BUSTAMANTE	SAMMY YASSER	Riesgo	003	IN
032648	LIMPE	ZEVALLOS	LUIS ANDRES	Riesgo	004	IN
052113	PUMA	HUILCA	RICHARD JESUS	Riesgo	005	IN
070730	YAÑEZ	SANZ	ANTHONY JOS...	No riesgo	003	IN
071106	CCAHUANTICO	MENDOZA	JULIO CESAR	No riesgo	004	IN
073289	BALLON	SEGOVIA	HENRY LUIS	Riesgo	006	IN
080219	MOREANO	BRICEÑO	GROVER	No riesgo	005	IN
081561	HUACANI	DE LA CRUZ	DANY DARWIN	No riesgo	006	IN
083221	TRUJILLO	TORBISCO	LUIS ANDERSO...	Riesgo	007	IN
083222	VARGAS	QUISPE	CHARLY	Riesgo	009	IN
091613	RAMIREZ	APAZA	EBERT	Riesgo	012	IN
091622	VALDEZ	CGUNO	JULIO CESAR	Riesgo	013	IN
093160	CHARA	TACURI	CESAR	Riesgo	014	IN
093169	HUAMANI	FLOREZ	RONALD	Riesgo	016	IN
093175	LAYME	MAMANI	NELSON	Riesgo	017	IN
093178	MAMANI	CRISPIN	ISAI ISAAC	Riesgo	018	IN

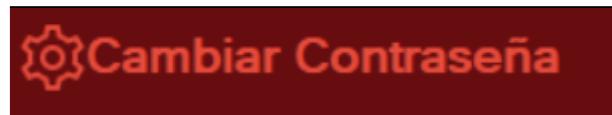
También existe una sección para realizar la búsqueda por la categoría que se deseé.

Campo:	<input type="text" value="CodAlumno"/>	<input type="button" value="▼"/>	<input type="text"/>
CodAlumno			
APaterno		CodTutor	CodEscuela
AMaterno		001	IN
Nombres		001	IN
Situacion		001	IN
CodEscuela		001	IN

Dentro de la tabla de alumnos se puede apreciar un botón para ingresar a la Ficha de tutoría del alumno en esa fila.

CodAlumno	APaterno	AMaterno	Nombres
112207	IBARRA	CASTILLO	WALDO ERICK

En la esquina inferior izquierda se agregó un botón para que el tutor pueda cambiar su contraseña.



Al darle click se despliega el panel de cambio de contraseña.

Confirmar Contraseña Actual:

Contraseña Nueva:

Confirmar Contraseña Nueva:

Confirmar

Cambiar Contraseña

Este panel consta de tres textbox, el primero es para verificar la clave actual del programa, el segundo para crear la nueva contraseña y el tercero para confirmar la contraseña. Al lado de cada textbox existen botones para poder ver momentáneamente la contraseña.

Confirmar Contraseña Actual:

holamundo

○

Y en la zona de confirmación de contraseña aparece una imagen que muestra si la contraseña nueva es igual a la de confirmación.

Finalmente está el botón de confirmación el cual verifica si la contraseña actual es correcta y si las nuevas contraseñas son iguales y diferentes al nulo.

#### 4. Ficha Tutoría

La interfaz Ficha Tutoría tiene la siguientes estructuras:

1. La primera es cuando la ficha es abierta por el tutor del alumno.
  - a. Inicialmente tenemos un panel de información donde se muestran los datos personales del Alumno.

Codigo	150388
Nombres	JUAN ANTONIO
Ap. Paterno	CHACON
Ap. Materno	VARGAS
Escuela P.	IN
Situación	No riesgo

- b. Seguidamente tenemos un panel donde se muestra la información de todas las sesiones realizadas. En el caso de que no se haya realizado determinada sesión, el programa estará listo para agregar una nueva a la base de datos.

The screenshot shows a software window titled "Sesión 1". At the top, it displays the date and time as "2021-09-11 09:43:20" and a dropdown menu for "Tipo de tutoría". Below this, there are three input fields: "Descripción", "Referencia", and "Observaciones", each with a corresponding green edit icon (a pencil inside a square).

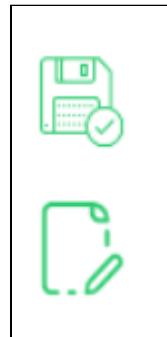
- c. En cada panel de sesión tenemos un elemento para obtener el día y la hora de la sesión, así como el tipo (campo obligatorio) de tutoría que se está realizando.

The screenshot shows the same session management interface as before. In addition to the date and time, there is a dropdown menu labeled "Tipo de tutoría" which is currently set to "Academico". A small preview window shows other options: "Academico", "Profesional", and "Personal".

- d. Luego tenemos los siguientes campos de caja de escritura para la descripción (campo obligatorio), referencia y observaciones.

The screenshot shows the final part of the session management interface where users can enter session details. It features three empty input fields: "Descripción", "Referencia", and "Observaciones", each accompanied by a green edit icon.

- e. Seguidamente tenemos dos botones uno para guardar y el otro para editar.



- f. El botón de guardar, guardará la ficha de sesión si todos los campos obligatorios tienen algún valor. Si agrega la sesión a la base de datos bloqueara la edición de la sesión respectiva.

Sesión 1 Fecha y Hora: 2021-09-11 09:43:20

Tipo de tutoría:

Descripción:	<input type="text" value="El alumno"/>	X	<input type="text" value="Artificial."/>
Referencia:	<input type="text"/>		<input type="button" value="Guardado"/>
Observaciones:	<input type="text" value="El alumno"/>	Aceptar	<input type="button" value="Editar"/>

Sesión 1 Fecha y Hora: 2021-09-11 09:43:20

Tipo de tutoría:

Descripción:	<input type="text"/>	X	<input type="button" value="Guardado"/>
Referencia:	<input type="text"/>		<input type="button" value="Editar"/>
Observaciones:	<input type="text"/>	Aceptar	<input type="button" value="Editar"/>

Escriba en el campo descripción de la sesión 1

Sesión 1 Fecha y Hora: 2021-09-11 09:43:20

Tipo de tutoría:

Descripción:	<input type="text" value="El alumno muestra un gran interes por el campo de la Inteligencia Artificial."/>	X	<input type="button" value="Guardado"/>
Referencia:	<input type="text"/>		<input type="button" value="Editar"/>
Observaciones:	<input type="text" value="El alumno tuvo un per..."/>	Sesión 1 añadida correctamente	<input type="button" value="Aceptar"/>

**Sesión 1**   **Fecha y Hora:** 2021-09-11 09:43:20   **Tipo de tutoría:** Academico

Descripción :	El alumno muestra un gran interés por el campo de la Inteligencia Artificial.	
Referencia :		
Observaciones :	El alumno tuvo un percance con el profesor Angelito.	

- g. El botón editar, se usará para desbloquear los campos de la edición de la sesión requerida y se podrá realizar los cambios correspondientes.

**Sesión 1**   **Fecha y Hora:** 2021-09-11 09:43:20   **Tipo de tutoría:** Academico

Descripción :	El alumno muestra un gran interés por el campo de la Inteligencia Artificial.	
Referencia :		
Observaciones :	El alumno tuvo un percance con el profesor Angelito.	

- h. El botón guardar, guardará los cambios realizados y bloqueará la edición de la sesión.

**Sesión 1**   **Fecha y Hora:** 2021-09-11 09:43:20   **Tipo de tutoría:** Academico

Descripción :	El alumno muestra un gran interés y desempeño por el campo de la Inteligencia Artificial.	
Referencia :	Nota del curso de IA.	
Observaciones :	El alumno tuvo un percance con el profesor Angelito.	

X

Sesión 1 editada correctamente

**Sesión 1** Fecha y Hora: 2021-09-11 09:43:20  Tipo de tutoría: Academico

Descripción:	El alumno muestra un gran interés y desempeño por el campo de la Inteligencia Artificial.	
Referencia:	Nota del curso de IA.	
Observaciones:	El alumno tuvo un percance con el profesor Angelito.	

2. La segunda interfaz es cuando la ficha es abierta por un tutor distinto al tutor del alumno.

**Ficha de Tutoría**

Código	093169	Semestre	<input type="button"/>
Nombres	RONALD		
Ap. Paterno	HUAMANI		
Ap. Materno	FLOREZ		
Escuela P.	IN		
Situación	Riesgo		

**Sesión 1** Fecha y Hora: 2021-09-22 10:16:57  Tipo de tutoría:

Descripción:	
Referencia:	

**Sesión 2** Fecha y Hora: 2021-09-22 10:16:57  Tipo de tutoría:

Descripción:	
Referencia:	

**Sesión 3** Fecha y Hora: 2021-09-22 10:16:57  Tipo de tutoría:

- a. Se muestran los datos personales del alumno, y la información de descripción y referencia. El tutor no puede editar la información de la ficha de tutoría.

## Ficha de Tutoría



Codigo 112207  
Nombres WALDO ERICK  
Ap. Paterno IBARRA  
Ap. Materno CASTILLO  
Escuela P. IN  
Situación Riesgo

Semestre

▼

### NOTA:

En la sección de observaciones se ingresa información confidencial del alumno que solo es visible para su tutor.

**Sesión 1 Fecha y Hora:** 2021-09-10 12:00:00  **Tipo de tutoría:** Academico

Descripción: huola  
Referencia: asbdjkagfia  
Observaciones: asbdjkagfia



**Sesión 2 Fecha y Hora:** 2021-09-17 12:00:00  **Tipo de tutoría:** Profesional

## CAPÍTULO 4:Pruebas

### 1. Pruebas Unitarias

#### 1.1. Formulario Login:

```

private void btEntrar_Click(object sender, EventArgs e)
{
    ConectarSQL conn = new ConectarSQL();
    if (conn.BuscarAdministrador(tbUsuario.Text, tbContra.Text).Rows.Count == 1)
    {
        Crud C = new Crud();
        this.Hide();
        C.Show();
    }
    else
    {
        MessageBox.Show("No se encontro el usuario");
    }
    tbUsuario.Text = " Usuario";
    tbContra.Text = " Contraseña";
    tbContra.UseSystemPasswordChar = false;
}

```

a. **Prueba 1:** inicio de sesión válido

Usuario: cadena de caracteres (válido)

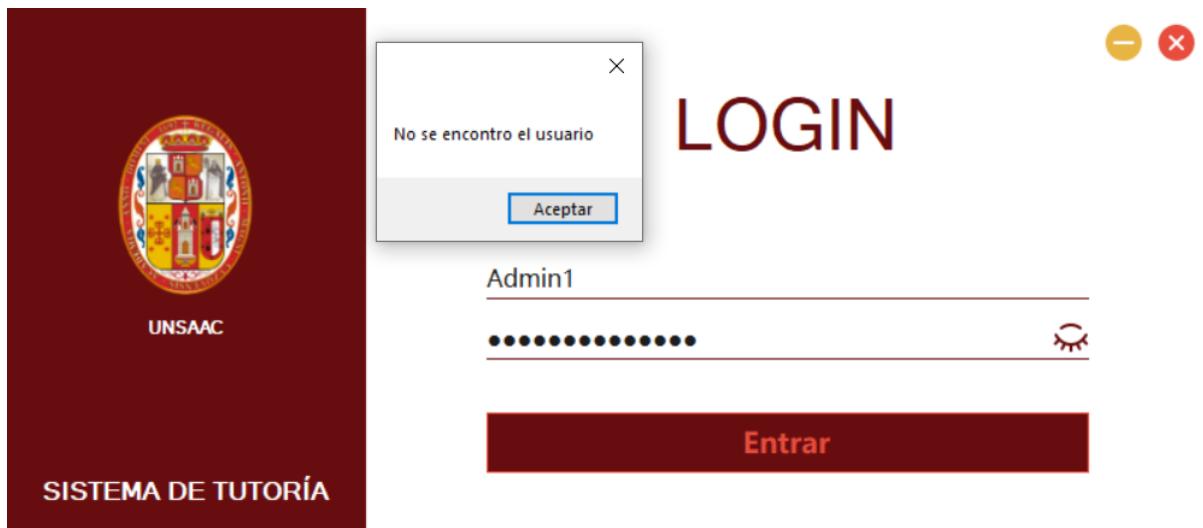
Contraseña: cadena de caracteres (válido)



b. **Prueba 2:** inicio de sesión incorrecto

Usuario: cadena de caracteres (válido)

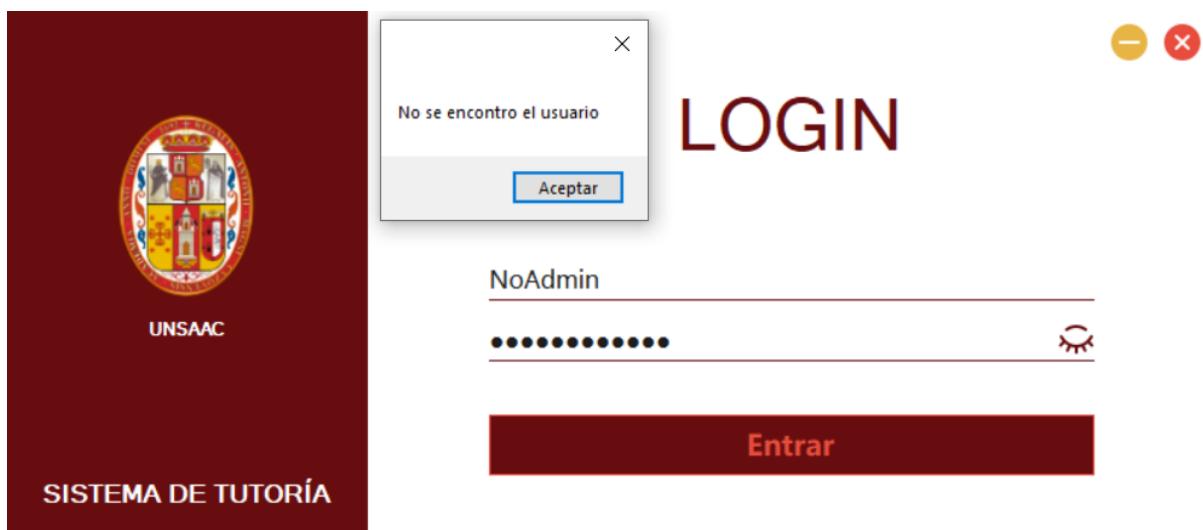
Contraseña: cadena de caracteres (inválido)



c. **Prueba 3:** inicio de sesión incorrecto

Usuario: cadena de caracteres (inválido)

Contraseña: cadena de caracteres



d. **Prueba 4:** inicio de sesión incompleto

Usuario: null

Contraseña: cadena de caracteres

UNSAAC

SISTEMA DE TUTORÍA

No se encontro el usuario

Aceptar

Usuario

Contraseña

Entrar

**e. Prueba 5:** inicio de sesión incompleto

Usuario: cadena de caracteres

Contraseña: null

UNSAAC

SISTEMA DE TUTORÍA

No se encontro el usuario

Aceptar

Admin1

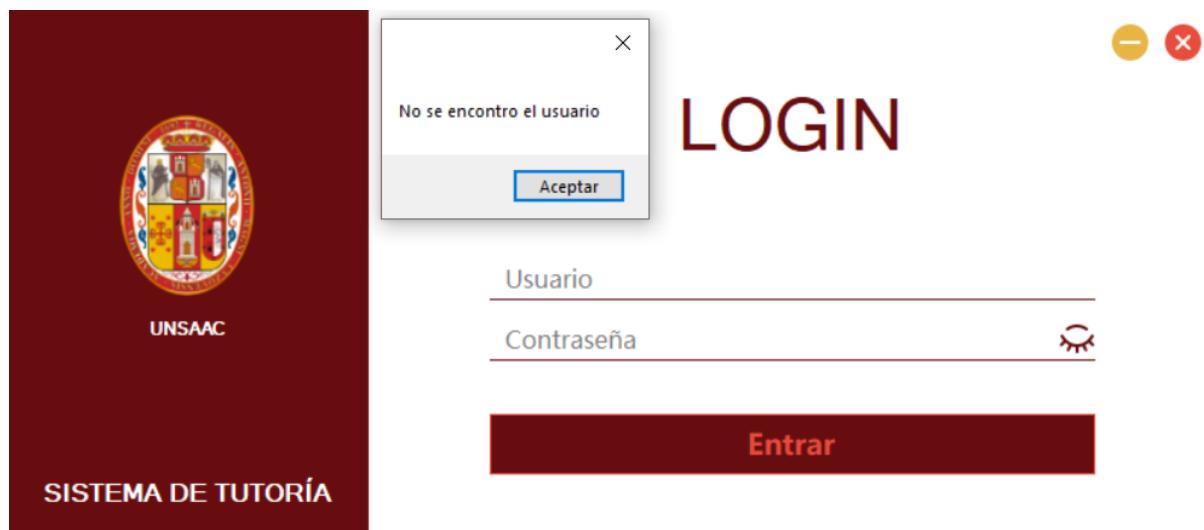
Contraseña

Entrar

**f. Prueba 6:** inicio de sesión inválido

Usuario: null

Contraseña: null



### Cuadro de Pruebas:

N	Test case	Input	Expected output	Actual output	Remarks
1	Inicio de sesión válido	usuario valido y contraseña válida	abrir formulario principal	abrir formulario principal	Pass
2	Inicio de sesión incorrecto	usuario válido y contraseña incorrecta	msgbox: “Contraseña incorrecta”	msgbox: “No se encontró el usuario”	No pass
3	Inicio de sesión incorrecto	usuario invalido y contraseña: cadena de caracteres	msgbox: “No se encontró el usuario”	msgbox: “No se encontró el usuario”	Pass
4	Inicio de sesión incompleto	usuario: null contraseña: cadena de caracteres	msgbox: “Usuario o contraseña incorrectos”	msgbox: “No se encontró usuario”	No pass
5	Inicio de sesión incompleto	usuario: cadena de caracteres contraseña: null	msgbox: “Usuario o contraseña incorrectos”	msgbox: “No se encontró el usuario”	No pass

6	Inicio de sesión inválido	usuario: null contraseña: null	msgbox: “Ingrese un usuario y contraseña”	msgbox: “No se encontró el usuario”	No pass
---	---------------------------	-----------------------------------	--	-------------------------------------	---------

## 1.2 Formulario Buscador:

### 1.2.1 Buscador Tutores

#### Código:

```
private void tbBuscar_TextChanged(object sender, EventArgs e)
{
    if (tbBuscar.Text != "")
    {
        dgvPrincipal.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;
        ConectarSQL conn = new ConectarSQL();
        dgvPrincipal.DataSource = conn.BuscarTutor(cmbCategoria.Text.ToString(), tbBuscar.Text.ToString());
    }
    else if (tbBuscar.Text == "")
    {
        refreshDataGridView();
    }
}
```

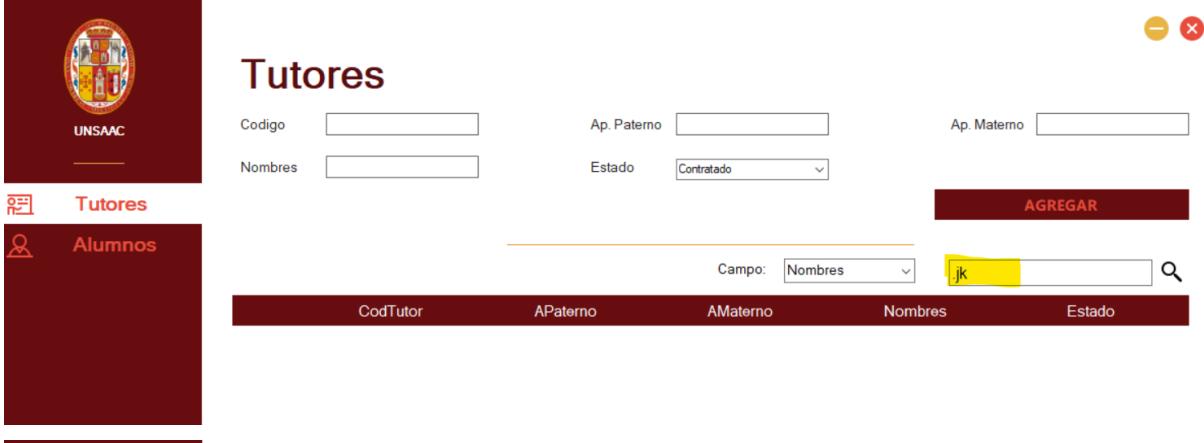
```
private void cmbCategoria_TextChanged(object sender, EventArgs e)
{
    refreshDataGridView();
    tbBuscar.Text = "";
}
```

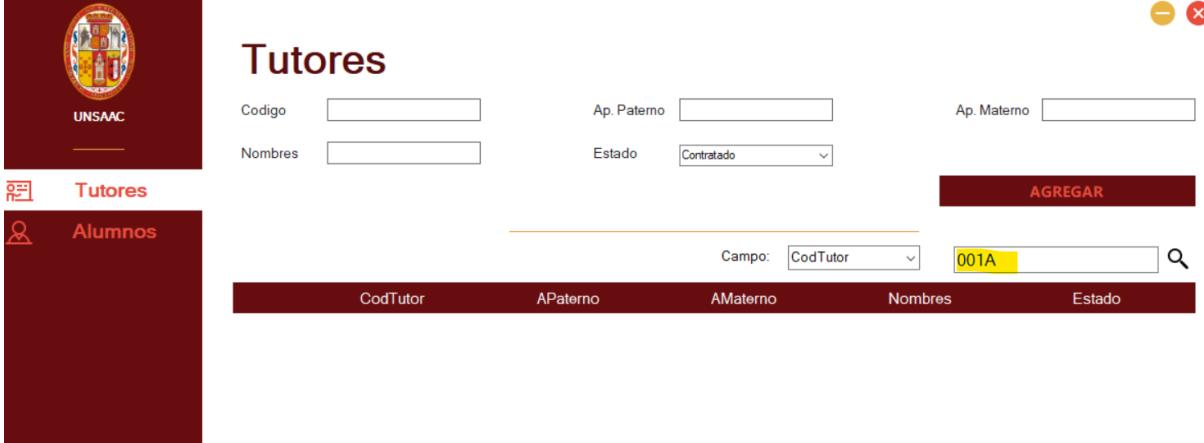
#### Pruebas:

a). Prueba 1 : Ingresamos datos válidos, es decir datos correctos a buscar

CodTutor	APaterno	AMaterno	Nombres	Estado
010	Ugarte	Rojas	Hector Eduardo	Nombrado
011	Nina	Hanco	Hernan	Nombrado
012	Cesar	Medrano	Ivan	Nombrado
013	Rozas	Huacho	Javier Arturo	Nombrado
014	Chavez	Centeno	Javier David	Nombrado
015	Soncco	Alvarez	Jose Luis	Nombrado
016	Pilco	Quispe	Jose Mauro	Nombrado
017	Carvajal	Luna	Julio Cesar	Nombrado
018	Medina	Miranda	Karelia	Nombrado
019	Enciso	Rodas	Lauro	Nombrado

b).Prueba 2: Donde ingresamos datos inválidos , es decir si ingresamos cualquier tipo de carácter (<;jk) o también cuando ingresamos códigos de tutor no válidos.



The screenshot shows the 'Tutores' (Tutors) section of the UNSAAC system. On the left sidebar, there are links for 'Tutores' and 'Alumnos'. The main area has fields for 'Codigo', 'Ap. Paterno', 'Ap. Materno', 'Nombres', and 'Estado'. Below these is a search bar with dropdown menus for 'Campo' (set to 'Nombres') and a text input containing the invalid character sequence '<;jk'. A magnifying glass icon is next to the search bar. At the bottom, there's a table header with columns 'CodTutor', 'APaterno', 'AMaterno', 'Nombres', and 'Estado'.  
  


This screenshot shows the same 'Tutores' section. The search bar now has 'Campo' set to 'CodTutor' and the text input contains the invalid code '001A'. The rest of the interface is identical to the first screenshot.

c).Prueba 3: Si no ingresamos ningún tipo de dato a la caja de busqueda(null)



**Tutores**

Codigo	<input type="text"/>	Ap. Paterno	<input type="text"/>	Ap. Materno	<input type="text"/>
Nombres	<input type="text"/>	Estado	<input type="text" value="Contratado"/>	<b>AGREGAR</b>	

**Tutores**

**Alumnos**

Campo: APaterno

CodTutor	APaterno	AMaterno	Nombres	Estado
001	Chullo	Llave	Boris	Nombrado
002	Quispe	Onofre	Carlos Ramon	Nombrado
003	Candia	Oviedo	Dennis Ivan	Nombrado
004	Carrasco	Poblete	Edwin	Nombrado
005	Palomino	Olivera	Emilio	Nombrado
006	Gamarra	Saldivar	Enrique	Nombrado
007	Pacheco	Vasquez	Esther	Nombrado
008	Cutipa	Arapa	Gladys Efraima	Nombrado
009	Ticona	Pari	Guzman	Nombrado
010	Ugarte	Rojas	Hector Eduardo	Nombrado
011	Nina	Hanco	Hernan	Nombrado
012	Cesar	Medrano	Ivan	Nombrado
013	Rozas	Huacho	Javier Arturo	Nombrado

### Casos de Prueba:

N	test case	Input	expected output	Actual output	Remarks
1	Caso de prueba Buscar por CodTutor,APaterno,AMaterno,Nombres,Estado que sean válidos	CodTutor,APaterno,AMaterno,Nombres,Estado válidos	Información de Tutor de acuerdo al campo de búsqueda ingresada	Información de Tutor de acuerdo al campo de búsqueda ingresada	pass
2	Caso de prueba Buscar por CodTutor,APaterno,AMaterno,Nombres,Estado que sean inválidos	CodTutor,APaterno,AMaterno,Nombres,Estado inválido(cualquier tipo de carácter)	Message box:"Datos Invalido: Ingrese alguna informacion de busqueda correcta"	Información sin datos (fila vacía sin datos)	no pass
3	Caso de prueba Buscar por CodTutor,APaterno,AMaterno,Nombres,Estado null	CodTutor,APaterno,AMaterno,Nombres,Estado null	Message box:"Error:In grese al menos un dato a buscar"	DataGridView de todos los datos tutores	no pass

	mbres,Estado que sean null		”		
--	----------------------------	--	---	--	--

### 1.2.2 Busqueda Alumnos

#### Código:

```
private void tbBuscar_TextChanged(object sender, EventArgs e)
{
    if (tbBuscar.Text != "")
    {
        dgvPrincipal.AutoSizeColumnsMode = DataGridViewAutoSizeColumnsMode.Fill;
        ConectarSQL conn = new ConectarSQL();
        dgvPrincipal.DataSource = conn.BuscarAlumno(cmbCategoria.Text.ToString(), tbBuscar.Text.ToString());
    }
    else if (tbBuscar.Text == "")
    {
        refreshDataGridView();
    }
}
```

#### Pruebas:

- a).Prueba 1 : Ingresamos datos válidos, es decir datos correctos a buscar de alumnos

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
143833	MELO	GUTIERREZ	RAUL ELVER	No riesgo	021	IN
204809	TORRES	MENDOZA	RAUL	No riesgo	024	IN
204856	Quispe	Zamalloa	Raul	No Riesgo	023	IN

- b).Prueba 2: Donde ingresamos datos inválidos , es decir si ingresamos cualquier tipo de carácter (<;,KJ,...) o también cuando ingresamos códigos de Alumnos(CodAlumno) no válidos.



**UNSAAC**

**Tutores**

**Alumnos**

## Alumnos

Codigo:  Ap. Paterno:  Ap. Materno:   
 En riesgo

Nombres:  Escuela P.:

Cod. Tutor:

**AGREGAR**

Campo: Nombres

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
012117	PINARES	BUSTAMANTE	SAMMY YASSER	Riesgo	003	IN
032648	LIMPE	ZEVALLOS	LUIS ANDRES	Riesgo	004	IN
052113	PUMA	HUILCA	RICHARD JESUS	Riesgo	005	IN
070730	YAÑEZ	SANZ	ANTHONY JOS...	No riesgo	003	IN
071106	CCAHUANTICO	MENDOZA	JULIO CESAR	No riesgo	004	IN
073289	BALLON	SEGOVIA	HENRY LUIS	Riesgo	006	IN
080219	MOREANO	BRICEÑO	GROVER	No riesgo	005	IN
081561	HUACANI	DE LA CRUZ	DANY DARWIN	No riesgo	006	IN
083221	TRUJILLO	TORBISCO	LUIS ANDERSON	Riesgo	007	IN
083222	VARGAS	QUISPE	CHARLY	Riesgo	009	IN
091613	RAMIREZ	APAZA	EBERT	Riesgo	012	IN
091622	VALDEZ	CGUNO	JULIO CESAR	Riesgo	013	IN



**UNSAAC**

**Tutores**

**Alumnos**

## Alumnos

Codigo:  Ap. Paterno:  Ap. Materno:   
 En riesgo

Nombres:  Escuela P.:

Cod. Tutor:

**AGREGAR**

Campo: CodAlumno  02

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
012117	PINARES	BUSTAMANTE	SAMMY YASSER	Riesgo	003	IN
032648	LIMPE	ZEVALLOS	LUIS ANDRES	Riesgo	004	IN
052113	PUMA	HUILCA	RICHARD JESUS	Riesgo	005	IN
070730	YAÑEZ	SANZ	ANTHONY JOS...	No riesgo	003	IN
071106	CCAHUANTICO	MENDOZA	JULIO CESAR	No riesgo	004	IN
073289	BALLON	SEGOVIA	HENRY LUIS	Riesgo	006	IN
080219	MOREANO	BRICEÑO	GROVER	No riesgo	005	IN
081561	HUACANI	DE LA CRUZ	DANY DARWIN	No riesgo	006	IN
083221	TRUJILLO	TORBISCO	LUIS ANDERSON	Riesgo	007	IN
083222	VARGAS	QUISPE	CHARLY	Riesgo	009	IN
091613	RAMIREZ	APAZA	EBERT	Riesgo	012	IN
091622	VALDEZ	CGUNO	JULIO CESAR	Riesgo	013	IN

c).Prueba 3: Si no ingresamos ningn tipo de dato a la caja de busqueda(null)



**UNSAAC**

**Tutores**

**Alumnos**

## Alumnos

Codigo:  Ap. Paterno:  Ap. Materno:   
 En riesgo

Nombres:  Escuela P.:

Cod. Tutor:

**AGREGAR**

Campo: Situacion

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
012117	PINARES	BUSTAMANTE	SAMMY YASSER	Riesgo	003	IN
032648	LIMPE	ZEVALLOS	LUIS ANDRES	Riesgo	004	IN
052113	PUMA	HUILCA	RICHARD JESUS	Riesgo	005	IN
070730	YAÑEZ	SANZ	ANTHONY JOS...	No riesgo	003	IN
071106	CCAHUANTICO	MENDOZA	JULIO CESAR	No riesgo	004	IN
073289	BALLON	SEGOVIA	HENRY LUIS	Riesgo	006	IN
080219	MOREANO	BRICEÑO	GROVER	No riesgo	005	IN
081561	HUACANI	DE LA CRUZ	DANY DARWIN	No riesgo	006	IN
083221	TRUJILLO	TORBISCO	LUIS ANDERSON	Riesgo	007	IN
083222	VARGAS	QUISPE	CHARLY	Riesgo	009	IN
091613	RAMIREZ	APAZA	EBERT	Riesgo	012	IN
091622	VALDEZ	CGUNO	JULIO CESAR	Riesgo	013	IN

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
012117	PINALES	BUSTAMANTE	SAMMY YASSER	Riesgo	003	IN
032648	LIMPE	ZEVALLOS	LUIS ANDRES	Riesgo	004	IN
052113	PUMA	HUILLCA	RICHARD JESUS	Riesgo	005	IN
070730	YANEZ	SANZ	ANTHONY JOS...	No riesgo	003	IN
071106	CCAHUANTICO	MENDOZA	JULIO CESAR	No riesgo	004	IN
073289	BALLON	SEGOVIA	HENRY LUIS	Riesgo	006	IN
080219	MOREANO	BRICEÑO	GROVER	No riesgo	005	IN
081561	HUACANI	DE LA CRUZ	DANY DARWIN	No riesgo	006	IN
083221	TRUJILLO	TORBISCO	LUIS ANDERSON	Riesgo	007	IN
083222	VARGAS	QUISPE	CHARLY	Riesgo	009	IN
091613	RAMIREZ	APAZA	EBERT	Riesgo	012	IN
091622	VALDEZ	CGUNO	JULIO CESAR	Riesgo	013	IN
093160	CHARA	TACURI	CESAR	Riesgo	014	IN

### Casos de Prueba:

N	test case	Input	expected output	Actual output	Remarks
1	Caso de prueba Buscar por CodAlumno,APaterno,AMaterno,Nombres,Situacion,CodTutor,CodEscuela que sean válidos	CodAlumno,APaterno,AMaterno,Nombres,Situacion,CodTutor,CodEscuela válidos	Información del Alumno de acuerdo al campo de búsqueda ingresada	Informacion de Alumno de acuerdo al campo de búsqueda ingresada	pass

2	Caso de prueba Buscar por CodAlumno,APaterno,AMaterno,Nombres,Situacion,CodTutor,Cod Escuela que sean inválidos	CodAlumno,APaterno,AMaterno,Nombres,Situacion,CodTutor,CodEscuela inválido(cualquier tipo de carácter)	Message box:"Datos Invalido: Ingrese alguna informacion de busqueda correcta"	Información sin datos (fila vacía sin datos)	no pass
3	Caso de prueba Buscar por CodAlumno,APaterno,AMaterno,Nombres,Situacion,CodTutor,Cod Escuela que sean null	CodAlumno,APaterno,AMaterno,Nombres,Situacion,CodTutor,CodEscuela null	Message box:"Error:Ingrese al menos un dato a buscar "	DataGridView de todos los datos de alumno	no pass

### 1.3. Formulario Tutores:

#### Código

##### a. Agregar

```
private void btnAgregar_Click(object sender, EventArgs e)
{
    if (btnAgregar.Text == "AGREGAR")
    {
        try
        {
            ConectarSQL conn = new ConectarSQL();

            conn.AgregarTutor(tbCodigo.Text, tbApPaterno.Text, tbApMaterno.Text, tbNombres.Text, cmbEstado.Text);
            MessageBox.Show("Agregado Correctamente");
            Clear();
        }
        catch
        {
            MessageBox.Show("Datos incorrectos o existentes");
        }
    }
    else
    {
        try
        {
            ConectarSQL conn = new ConectarSQL();

            conn.EditarTutor(tbCodigo.Text, tbApPaterno.Text, tbApMaterno.Text, tbNombres.Text, cmbEstado.Text);

            MessageBox.Show("Fila editada correctamente");
            btnCancelar.Visible = false;
            btnAgregar.Text = "AGREGAR";
            tbCodigo.Enabled = true;
            Clear();
        }
        catch
        {
            MessageBox.Show("Datos incorrectos o existentes");
        }
    }
    refreshDataGridView();
}
```

## b. Eliminar

```
if (e.ColumnIndex >= 0 && dgvPrincipal.Columns[e.ColumnIndex].Name == "Eliminar")
{
    // Código para eliminar elemento
    //
    string Código = dgvPrincipal.Rows[e.RowIndex].Cells["CodTutor"].FormattedValue.ToString();
    DialogResult result = MessageBox.Show("Seguro que desea Eliminar Tutor de codigo:" + Código + "?", "Eliminar Tutor", MessageBoxButtons.YesNo);

    switch (result)
    {
        case DialogResult.Yes:
            try
            {
                ConectarSQL conn = new ConectarSQL();
                conn.EliminarTutor(Código);
                MessageBox.Show("Eliminado");
            }
            catch
            {
                MessageBox.Show("No es posible eliminar un tutor asignado a un alumno");
            }
            break;

        case DialogResult.No:
            break;
    }
    //MessageBox.Show("Eliminado");
    refreshDataGridView();
}
```

## c. Editar

```
else if (e.ColumnIndex >= 0 && dgvPrincipal.Columns[e.ColumnIndex].Name == "Editar")
{
    dgvPrincipal.CurrentRow.Selected = true;
    tbCodigo.Text = dgvPrincipal.Rows[e.RowIndex].Cells["CodTutor"].FormattedValue.ToString();
    tbNombres.Text = dgvPrincipal.Rows[e.RowIndex].Cells["Nombres"].FormattedValue.ToString();
    tbApPaterno.Text = dgvPrincipal.Rows[e.RowIndex].Cells["APaterno"].FormattedValue.ToString();
    tbApMaterno.Text = dgvPrincipal.Rows[e.RowIndex].Cells["AMaterno"].FormattedValue.ToString();
    cmbEstado.Text = dgvPrincipal.Rows[e.RowIndex].Cells["Estado"].FormattedValue.ToString();

    btnAgregar.Text = "EDITAR";
    btnCancelar.Visible = true;
    tbCodigo.Enabled = false;
}
```

### Pruebas:

Prueba 1: Clic en botón agregar, código null, Nombres null, Ap.Paterno null, Ap. Materno null, estado null.

**Tutores**

Codigo	<input type="text"/>	Ap. Paterno	<input type="text"/>	Ap. Materno	<input type="text"/>
Nombres	<input type="text"/>	Estado	<input type="text"/>		

**AGREGAR**

**Alumnos**

CodTutor	APaterno	AMaterno	Nombres	Estado
000	A		Armin	Nombrado
001	C	Datos incorrectos o existentes	Boris	Nombrado
002	Q		Carlos Ramon	Nombrado
003	C		Dennis Ivan	Nombrado
004	Cafresco	Fabrieta	Edwin	Nombrado
005	Palomino	Olivera	Emilio	Nombrado
006	Gamarra	Saldivar	Enrique	Nombrado
007	Pacheco	Vasquez	Esther	Nombrado
008	Cutipa	Arapa	Gladys Efraima	Nombrado
009	Ticona	Pari	Guzman	Nombrado
010	Ugarte	Rojas	Hector Eduardo	Nombrado
011	Nina	Hanco	Heman	Nombrado
012	Cesar	Medrano	Ivan	Nombrado
013	Rozas	Huacho	Javier Arturo	Nombrado

Campo:  CodTutor

**b) Prueba 2:** Clic en botón agregar, código null, Nombres null, Ap.Paterno null, Ap. Materno null, estado valido.

CodTutor	APaterno	AMaterno	Nombres	Estado
001	Chu	X e	Boris	Nombrado
002	Quis	re	Carlos Ramon	Nombrado
003	Can	Agregado Correctamente	Dennis Ivan	Nombrado
004	Carra	do	Edwin	Nombrado
005	Palon	ste	Emilio	Nombrado
006	Gamme	ra	Enrique	Nombrado
007	Pacheco	Vasquez	Esther	Nombrado
008	Cutipa	Arapa	Gladys Efraina	Nombrado
009	Ticona	Pari	Guzman	Nombrado
010	Ugarte	Rojas	Hector Eduardo	Nombrado
011	Nina	Hanco	Hernan	Nombrado
012	Cesar	Medrano	Ivan	Nombrado
013	Rozas	Huacho	Javier Arturo	Nombrado
014	Chavez	Centeno	Javier David	Nombrado
015	Sonoco	Alvarez	Ines Irie	Nombrado

Prueba 3: Clic en botón editar sin hacer cambio alguno.

CodTutor	APaterno	AMaterno	Nombres	Estado
001	Chullo	X e	Boris	Nombrado
002	Quis	re	Carlos Ramon	Nombrado
003	Can	Fila editada correctamente	Dennis Ivan	Nombrado
004	Carra	o	Edwin	Nombrado
005	Palon	ste	Emilio	Nombrado
006	Gamme	ra	Enrique	Nombrado
007	Pacheco	Vasquez	Esther	Nombrado
008	Cutipa	Arapa	Gladys Efraina	Nombrado
009	Ticona	Pari	Guzman	Nombrado
010	Ugarte	Rojas	Hector Eduardo	Nombrado
011	Nina	Hanco	Hernan	Nombrado
012	Cesar	Medrano	Ivan	Nombrado
013	Rozas	Huacho	Javier Arturo	Nombrado
014	Chavez	Centeno	Javier David	Nombrado
015	Sonoco	Alvarez	Ines Irie	Nombrado

**Prueba 4: Clic en botón editar, Nombres null, Ap. Paterno null, Ap. Materno null, estado valido.**

The screenshot shows a web application interface for managing tutors. At the top, there's a header with the logo of 'UNSAAC' and navigation links for 'Tutores' and 'Alumnos'. The main section is titled 'Tutores' and contains search fields for 'Codigo' (036), 'Nombres', 'Ap. Paterno', and 'Ap. Materno'. Below these are dropdown menus for 'Estado' (set to 'Nombrado') and 'Cancelar'/'EDITAR' buttons. A modal dialog box is open over the table, indicating that the row for tutor code 036 has been edited correctly. The table lists various tutor records with columns: CodTutor, APaterno, AMaterno, Nombres, and Estado. The record for tutor code 036 is highlighted in orange.

CodTutor	APaterno	AMaterno	Nombres	Estado
023	Tito	Luis Beltran	Nombrado	
024	Figueroa	Manuel	Nombrado	
025	Vergara	Maria del Pilar	Nombrado	
026	Cusimayta	Maritza	Nombrado	
027	Ucsa	Nila Sonia	Nombrado	
028	Alzamora	Paredes	Robert Wilbert	
029	Cusuhaman	Phocco	Roger Mario	
030	Villafuerte	Serna	Rony	
031	Quintanilla	Portugal	Roxana Lisette	
032	Sosa	Jauregui	Victor David	
033	Ibarra	Zambrano	Waldo Elio	
034	Zamalloa	Paro	Willian	
035	Ormeño	Ayala	Yeshica Isela	
036	Arlet	Arlet	Armin	
037			Nombrado	

**Prueba 5: Clic en el icono de eliminar**

This screenshot shows the same 'Tutores' page after an edit operation. The modal dialog now displays a confirmation message asking if the user wants to delete the tutor with code 000. The 'Si' (Yes) button is highlighted. The table below shows a list of tutors, with the row for tutor code 000 highlighted in orange.

CodTutor	APaterno	AMaterno	Nombres	Estado
000	Palomino	Olivera	Armin	Nombrado
001	Gamarra	Saldivar	Boris	Nombrado
002	Pacheco	Vasquez	Carlos Ramon	Nombrado
003			Dennis Ivan	Nombrado
004			Edwin	Nombrado
005	Cutipa	Arapa	Emilio	Nombrado
006	Ticona	Pari	Esther	Nombrado
007	Ugarte	Rojas	Gladys Efraina	Nombrado
008	Nina	Hanco	Guzman	Nombrado
009	Cesar	Medrano	Hector Eduardo	Nombrado
010	Rozas	Huacho	Ivan	Nombrado
011			Javier Arturo	Nombrado
012			Javier David	Nombrado
013				
014				

CodTutor	APaterno	AMaterno	Nombres	Estado
000	Arlet	Armin	Nombrado	
001	Llave	Boris	Nombrado	
002	Onofre	Carlos Ramon	Nombrado	
003	Oviedo	Dennis Ivan	Nombrado	
004	Poblete	Edwin	Nombrado	
005	Olivera	Emilio	Nombrado	
006	Saldivar	Enrique	Nombrado	
007	Vasquez	Esther	Nombrado	
008	Cutipa	Gladys Efraima	Nombrado	
009	Ticona	Pari	Guzman	
010	Ugarte	Rojas	Hector Eduardo	
011	Nina	Hanco	Herman	
012	Cesar	Medrano	Ivan	
013	Rozas	Huacho	Javier Arturo	
014	Chavez	Centeno	Javier David	

## Caso de prueba

N	test case	Input	expected output	Actual output	Remarks
1	Caso de prueba Agregar Tutor invalido	Clic en botón agregar, código null, Nombres null, Ap.Paterno null, Ap. Materno null, estado null.	msgbox: “Datos incompletos”	msgbox: “Datos incorrectos o existentes”	Pass
2	Caso de Prueba Editar Tutor invalido	Clic en botón agregar, código null, Nombres null, Ap.Paterno null, Ap. Materno null, estado valido.	msgbox: “Datos incompletos”	msgbox: “Fila editada correctamente”	No Pass
3	Caso de Prueba Editar Tutor invalido	Clic en botón editar sin hacer cambios.	msgbox: “No se hicieron cambios”	msgbox: “Fila editada correctamente”	No Pass
4	Caso de Prueba Editar Tutor invalido	Clic en botón editar, Nombres null, Ap. Paterno null,	msgbox: “Datos incompletos”	msgbox: “Fila editada correctamente”	No Pass

		Ap. Materno null, estado valido.			
5	Caso de prueba eliminar Tutor válido	Hacer click en el ícono del tacho	msgbox: “¿Seguro que desea eliminar al Tutor?”	msgbox: “¿Seguro que desea eliminar al Tutor?”	Pass

#### 1.4 Alumnos:

##### Código:

###### a. Eliminar

```
if (e.ColumnIndex >= 0 && dgvPrincipal.Columns[e.ColumnIndex].Name == "Eliminar")
{
    // Código para eliminar elemento
    //
    string Código = dgvPrincipal.Rows[e.RowIndex].Cells["CodAlumno"].FormattedValue.ToString();
    DialogResult result = MessageBox.Show("Seguro que desea Eliminar Alumno de código:" + Código + "?",
                                         "Eliminar Alumno", MessageBoxButtons.YesNo);

    switch (result)
    {
        case DialogResult.Yes:
            try
            {
                ConectarSQL conn = new ConectarSQL();
                conn.EliminarAlumno(Código);
                MessageBox.Show("Eliminado");
            }
            catch
            {
                MessageBox.Show("No es posible eliminar el Alumno");
            }

            break;
        case DialogResult.No:
            break;
    }
    //MessageBox.Show("Eliminado");
    refreshDataGridView();
}
```

###### b. Agregar:

```
private void btnAgregar_Click(object sender, EventArgs e)
{
    if (btnAgregar.Text == "AGREGAR")
    {
        try
        {
            ConectarSQL conn = new ConectarSQL();
            string Situacion;
            if (cbRiesgo.Checked == true)
            {
                Situacion = "Riesgo";
            }
            else
            {
                Situacion = "No Riesgo";
            }
            conn.AgregarAlumno(tbCodigo.Text, tbApPaterno.Text, tbApMaterno.Text, tbNombres.Text, Situacion,
                tbCodTutor.Text, tbCodEscuela.Text);
            MessageBox.Show("Agregado Correctamente");
            Clear();
        }
        catch
        {
            MessageBox.Show("Datos incorrectos o existentes");
        }
    }
}
```

```
else
{
    try
    {
        ConectarSQL conn = new ConectarSQL();
        string Situacion;
        if (cbRiesgo.Checked == true)
        {
            Situacion = "Riesgo";
        }
        else
        {
            Situacion = "No Riesgo";
        }
        conn.EditarAlumno(tbCodigo.Text, tbApPaterno.Text, tbApMaterno.Text, tbNombres.Text, Situacion,
            tbCodTutor.Text, tbCodEscuela.Text);

        MessageBox.Show("Fila editada correctamente");
        btnCancelar.Visible = false;
        btnAgregar.Text = "AGREGAR";
        tbCodigo.Enabled = true;
        Clear();

    }
    catch
    {
        MessageBox.Show("Datos incorrectos o existentes");
    }
}

refreshDataGridView();
}
```

c. Editar:

```

else if (e.ColumnIndex >= 0 && dgvPrincipal.Columns[e.ColumnIndex].Name == "Editar")
{
    dgvPrincipal.CurrentRow.Selected = true;
    tbCodigo.Text = dgvPrincipal.Rows[e.RowIndex].Cells["CodAlumno"].FormattedValue.ToString();
    tbNombres.Text = dgvPrincipal.Rows[e.RowIndex].Cells["Nombres"].FormattedValue.ToString();
    tbApPaterno.Text = dgvPrincipal.Rows[e.RowIndex].Cells["APaterno"].FormattedValue.ToString();
    tbApMaterno.Text = dgvPrincipal.Rows[e.RowIndex].Cells["AMaterno"].FormattedValue.ToString();
    tbCodEscuela.Text = dgvPrincipal.Rows[e.RowIndex].Cells["CodEscuela"].FormattedValue.ToString();
    tbCodTutor.Text = dgvPrincipal.Rows[e.RowIndex].Cells["CodTutor"].FormattedValue.ToString();

    if (dgvPrincipal.Rows[e.RowIndex].Cells["Situacion"].FormattedValue.ToString() == "Riesgo")
    {
        cbRiesgo.Checked = true;
    }
    else
    {
        cbRiesgo.Checked = false;
    }
    btnAgregar.Text = "EDITAR";
    btnCancelar.Visible = true;
    tbCodigo.Enabled = false;
}

else
{
    try
    {
        ConectarSQL conn = new ConectarSQL();
        string Situacion;
        if (cbRiesgo.Checked == true)
        {
            Situacion = "Riesgo";
        }
        else
        {
            Situacion = "No Riesgo";
        }
        conn.EditarAlumno(tbCodigo.Text, tbApPaterno.Text, tbApMaterno.Text, tbNombres.Text, Situacion,
                           tbCodTutor.Text, tbCodEscuela.Text);

        MessageBox.Show("Fila editada correctamente");
        btnCancelar.Visible = false;
        btnAgregar.Text = "AGREGAR";
        tbCodigo.Enabled = true;
        Clear();
    }
    catch
    {
        MessageBox.Show("Datos incorrectos o existentes");
    }
}

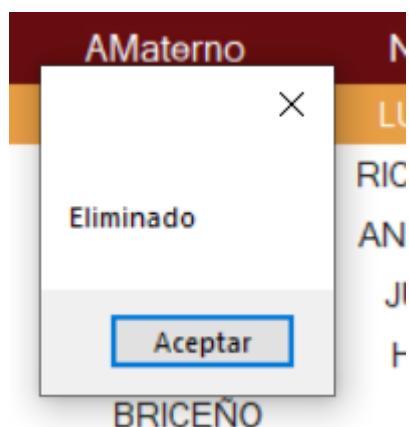
```

## Pruebas:

- Prueba 1: Click en botón de eliminar

The screenshot shows the 'Alumnos' (Students) page. At the top, there are search fields for 'Codigo', 'Ap. Paterno', 'Ap. Materno', 'Nombres', 'Escuela P.', and 'Cod. Tutor'. There is also a checkbox for 'En riesgo' and a 'AGREGAR' button. Below the search fields is a table listing students with columns: CodAlumno, APaterno, AMaterno, Nombres, Situacion, CodTutor, and CodEscuela. A modal dialog titled 'Eliminar Alumno' is open, asking 'Seguro que desea Eliminar Alumno de codigo:032648?' with 'Sí' and 'No' buttons.

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
032648	LIMPE	ZEVALLOS	LUIS ANDRES	Riesgo	004	IN
052113	PUMA	HUILLCA	RICHARD JESUS	Riesgo	005	IN
070730	YAÑEZ	SANZ	ANTHONY JOS...	No riesgo	003	IN
071106	CCAHUANTICO	MENDOZA	JULIO CESAR	No riesgo	004	IN
073289	BALLON	SEGOVIA	HENRY LUIS	Riesgo	006	IN
080219	MOREANO	BRICEÑO	GROVER	No riesgo	005	IN
081561	HUACANI	DE LA CRUZ	DANY DARWIN	No riesgo	006	IN
083221	TRUJILLO	TORBISCO	LUIS ANDERSON	Riesgo	007	IN
083222	VARGAS	QUISPE	CHARLY	Riesgo	009	IN
091613	RAMIREZ	APAZA	EBERT	Riesgo	012	IN
091622	VALDEZ	CGUNO	JULIO CESAR	Riesgo	013	IN
093160	CHARA	TACURI	CESAR	Riesgo	014	IN
093169	HUAMANI	FLOREZ	RONALD	Riesgo	016	IN
093175	LAYME	MAMANI	NELSON	Riesgo	017	IN
093178	MAMANI	CRISPIN	ISAIASDAO	Riesgo	018	IN



b. Prueba 2: Click en botón de Editar del data grid view

The screenshot shows the 'Alumnos' (Students) page with a student record for 'RICHARD JESUS' selected. The edit form at the top includes fields for 'Codigo' (052113), 'Ap. Paterno' (PUMA), 'Ap. Materno' (HUILLCA), 'Nombres' (RICHARD JESUS), 'Escuela P.' (IN), and 'Cod. Tutor' (005). A checkbox for 'En riesgo' is checked. Below the form is a table with columns: CodAlumno, APaterno, AMaterno, Nombres, Situacion, CodTutor, and CodEscuela. The student record for 'RICHARD JESUS' is highlighted in the table.

CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
052113	PUMA	HUILLCA	RICHARD JESUS	Riesgo	005	IN
070730	YAÑEZ	SANZ	ANTHONY JOS...	No riesgo	003	IN
071106	CCAHUANTICO	MENDOZA	JULIO CESAR	No riesgo	004	IN
073289	BALLON	SEGOVIA	HENRY LUIS	Riesgo	006	IN
080219	MOREANO	BRICEÑO	GROVER	No riesgo	005	IN
081561	HUACANI	DE LA CRUZ	DANY DARWIN	No riesgo	006	IN
083221	TRUJILLO	TORBISCO	LUIS ANDERSON	Riesgo	007	IN
083222	VARGAS	QUISPE	CHARLY	Riesgo	009	IN
091613	RAMIREZ	APAZA	EBERT	Riesgo	012	IN
091622	VALDEZ	CGUNO	JULIO CESAR	Riesgo	013	IN
093160	CHARA	TACURI	CESAR	Riesgo	014	IN

c. Prueba 3: Click en botón de Editar nuevo, APaterno en null

## Alumnos

Codigo	052113	Ap. Paterno		Ap. Materno	HUILCA
Nombres	RICHARD JESUS	Escuela P.	IN	<input checked="" type="checkbox"/> En riesgo	
Cod. Tutor	005	<b>CANCELAR</b>			<b>EDITAR</b>

---

Lista de Alumnos								
	CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela	
	052113	PUMA	HUILCA	RICHARD JESUS	Riesgo	005	IN	
	070730	YAÑEZ	SANZ	ANTHONY JOS...	No riesgo	003	IN	
	071106	CCAHUANTI	X	JULIO CESAR	No riesgo	004	IN	
	073289	BALLON		ENRY LUIS	Riesgo	006	IN	
	080219	MOREANO		GROVER	No riesgo	005	IN	
	081561	HUACANI		DANY DARWIN	No riesgo	006	IN	
	083221	TRUJILLO	Aceptar	SANDERSON	Riesgo	007	IN	
	083222	VARGAS	QUISPE	CHARLY	Riesgo	009	IN	
	091613	RAMIREZ	APAZA	EBERT	Riesgo	012	IN	

d. Prueba 4: Click en botón de Editar nuevo, AMaterno en null

## Alumnos

Codigo	052113	Ap. Paterno	PUMA	Ap. Materno	
Nombres	RICHARD JESUS	Escuela P.	IN	<input checked="" type="checkbox"/> En riesgo	
Cod. Tutor	005	<b>CANCELAR</b>			<b>EDITAR</b>

---

Lista de Alumnos								
	CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela	
	052113	X	HUILCA	RICHARD JESUS	Riesgo	005	IN	
	070730		ANZ	ANTHONY JOS...	No riesgo	003	IN	
	071106		DOZA	JULIO CESAR	No riesgo	004	IN	
	073289		GOVIA	HENRY LUIS	Riesgo	006	IN	
	080219		CEÑO	GROVER	No riesgo	005	IN	
	081561	HUACANI	DE LA CRUZ	DANY DARWIN	No riesgo	006	IN	
	083221	TODISCO		SANDERSON	Riesgo	007	IN	

e. Prueba 5: Click en botón de Editar nuevo, Nombres en null



## Alumnos

Codigo	052113	Ap. Paterno	PUMA	Ap. Materno	HUILCA
Nombres		Escuela P.	IN	<input checked="" type="checkbox"/> En riesgo	
Cod. Tutor	005	<input type="button" value="CANCELAR"/> <input type="button" value="EDITAR"/>			

Campo: CodAlumno

	CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
	052113	PUMA		RICHARD JESUS	Riesgo	005	IN
	070730	YAÑEZ		THONY JOS...	No riesgo	003	IN
	071106	CCAHUANT		JULIO CESAR	No riesgo	004	IN
	073289	BALLON		ENRY LUIS	Riesgo	006	IN
	080219	MOREAN		GROVER	No riesgo	005	IN
	081561	HUACANI	DE LA CRUZ	DANY DARWIN	No riesgo	006	IN
	083221	TRUJILLO	TORBISCO	LUIS ANDERSON	Riesgo	007	IN
	083222	VARGAS	QUISPE	CHARLY	Riesgo	009	IN

f. Prueba 6: Click en botón de Editar nuevo, Escuela en null

Codigo	052113	Ap. Paterno	PUMA	Ap. Materno	HUILCA
Nombres	RICHARD JESUS	Escuela P.		<input checked="" type="checkbox"/> En riesgo	
Cod. Tutor	005	<input type="button" value="CANCELAR"/> <input type="button" value="EDITAR"/>			

Campo: CodAlumno

	CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
	052113	PUMA		RICHARD JESUS	Riesgo	005	IN
	070730	YAÑEZ		THONY JOS...	No riesgo	003	IN
	071106	CCAHUANT		JULIO CESAR	No riesgo	004	IN
	073289	BALLON		ENRY LUIS	Riesgo	006	IN
	080219	MOREAN		GROVER	No riesgo	005	IN
	081561	HUACANI	DE LA CRUZ	DANY DARWIN	No riesgo	006	IN
	083221	TRUJILLO	TORBISCO	LUIS ANDERSON	Riesgo	007	IN

g. Prueba 7: Click en botón de Editar nuevo, CodTutor inexistente

Codigo	052113	Ap. Paterno	PUMA	Ap. Materno	HUILCA
Nombres	RICHARD JESUS	Escuela P.	IN	<input checked="" type="checkbox"/> En riesgo	
Cod. Tutor		<input type="button" value="CANCELAR"/> <input type="button" value="EDITAR"/>			

Campo: CodAlumno

	CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
	052113	PUMA		RICHARD JESUS	Riesgo	005	IN
	070730	YAÑEZ		THONY JOS...	No riesgo	003	IN
	071106	CCAHUANT		JULIO CESAR	No riesgo	004	IN
	073289	BALLON		ENRY LUIS	Riesgo	006	IN
	080219	MOREAN		GROVER	No riesgo	005	IN
	081561	HUACANI	DE LA CRUZ	DANY DARWIN	No riesgo	006	IN

h. Prueba 8: Click en botón de Editar nuevo con datos correctos

## Alumnos

Codigo	052113	Ap. Paterno	LEÓN	Ap. Materno	DEL MONTE
Nombres	RICHARD JESUS	Escuela P.	IN	<input type="checkbox"/> En riesgo	
Cod. Tutor	003	<b>CANCELAR</b>			<b>EDITAR</b>

Campo: CodAlumno

	CodAlumno	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
	052113	PUMA		X RICHARD JESUS	Riesgo	005	IN
	070730	YAÑEZ		THONY JOS...	No riesgo	003	IN
	071106	CCAHUANT	Fila editada correctamente	JULIO CESAR	No riesgo	004	IN
	073289	BALLON		ENRY LUIS	Riesgo	006	IN
	080219	MOREANC		GROVER	No riesgo	005	IN
	081561	HUACANI	DE LA CRUZ	DÁNY DARWIN	No riesgo	006	IN

- i. Prueba 9: Llenar datos del Alumno con un código ya existente, y dar click en el botón agregar.

## Alumnos

Codigo	182898	Ap. Paterno	Cerrillo	Ap. Materno	Miranda
Nombres	Gianela	Escuela P.	IN	<input checked="" type="checkbox"/> En riesgo	
Cod. Tutor	003	<b>AGREGAR</b>			

Campo: CodAlumno

	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
	PINARES	BUSTAMANTE	SAMMY YASSER	Riesgo	003	IN
	DURAND	NAVARRO	LUISA SHIRLEY	RIESGO	019	IN
	ARONI	SOTO	WARREN STEP...	RIESGO	020	IN

- j. Prueba 10: Llenar datos del Alumno con Cod. Tutor vacío, y dar click en el botón agregar

## Alumnos

Codigo	204810	Ap. Paterno	CERRILLO	Ap. Materno	MIRANDA
Nombres	GIANELA	Escuela P.	IN	<input type="checkbox"/> En riesgo	
Cod. Tutor		<b>AGREGAR</b>			

Campo: CodAlumno

	APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
	PACHARI	LIPA	MILTON ALEXIS	NO RIESGO	018	IN
	PEÑA	CARALIFRO	JOSE LUIS	NO RIESGO	019	IN

- k. Prueba 11: Llenar datos del Alumno con un Cod. Tutor inexistente, y dar click en el botón agregar.

## Alumnos

Codigo: 204810 Ap. Paterno: CERRILLO Ap. Materno: MIRANDA  
 Nombres: GIANELA Escuela P.: IN En riesgo  
 Cod. Tutor: 050 AGREGAR

Datos incorrectos o existentes

APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
CUSI	QUISPE	YANET	No riesgo	010	IN
GUEVARA	DELGADO	TIRSSA IVONNE	Riesgo	012	IN

- Prueba 12: Llenar datos del Alumno con datos correctos, y dar click en el botón agregar.

## Alumnos

Codigo: 204810 Ap. Paterno: CERRILLO Ap. Materno: MIRANDA  
 Nombres: GIANELA Escuela P.: IN En riesgo  
 Cod. Tutor: 003 AGREGAR

Agregado Correctamente

APaterno	AMaterno	Nombres	Situacion	CodTutor	CodEscuela
PINARES	BUSTAMANTE	SAMMY YASSER	Riesgo	003	IN
DURAND	NAVARRO	LUISA SHIRLEY	RIESGO	019	IN

### Casos de Prueba:

N	Test Case	Input	expected output	Actual output	Remarks
1	Caso de prueba eliminar alumno válido al hacer click en el icono del tacho	Hacer click en el icono del tacho	msgbox: “¿Seguro que desea eliminar al alumno?”	msgbox: “¿Seguro que desea eliminar al alumno?”	Pass
2	Caso de Prueba Editar válido al hacer click en el icono del lápiz	hacer click en el icono del lápiz	Llenar los textbox con los datos del alumno, deshabilitar textbox de código, hacer aparecer botón de cancelar y cambiar nombre de	Llenar los textbox con los datos del alumno, deshabilitar textbox de código, hacer aparecer botón de cancelar y cambiar nombre de	Pass

			botón de agregar a editar	botón de agregar a editar	
3	Caso de Prueba Editar invalido por APaterno en null	APaterno en null	msgbox: “Datos incompletos en APaterno ”	msgbox: “Fila editada correctamente”	No Pass
4	Caso de Prueba Editar invalido por AMaterno en null	AMaterno en null	msgbox: “Datos incompletos en AMaterno ”	msgbox: “Fila editada correctamente”	No Pass
5	Caso de Prueba Editar invalido por Nombres en null	Nombres en null	msgbox: “Datos incompletos en Nombres ”	msgbox: “Fila editada correctamente”	No Pass
6	Caso de Prueba Editar invalido por Escuela Null	Escuela Null	msgbox: “Datos incompletos en Escuela ”	msgbox: “Datos incorrectos o inexistentes”	No Pass
7	Caso de Prueba Editar invalido por CodTutor null	CodTutor null	msgbox: “Datos incompletos en CodTutor ”	msgbox: “Datos incorrectos o inexistentes”	No Pass
8	Caso de Prueba Válido por Variables no null y datos válidos	Variables no null y datos válidos	msgbox: “Fila editada correctamente”	msgbox: “Fila editada correctamente”	Pass
9	Caso de Prueba Agregar invalido por Código ya existente	Código ya existente	msgbox: “ Datos incorrecto o existentes”	msgbox: “ Datos incorrecto o existentes”	Pass
10	Caso de Prueba Agregar invalido por Cod Tutor en null	Cod Tutor null	msgbox: “ Datos incorrecto o existentes”	msgbox: “ Datos incorrecto o existentes”	Pass
11	Caso de Prueba Agregar invalido por Cod Tutor Inexistente	Cod Tutor Inexistente	msgbox: “ Datos incorrecto o existentes”	msgbox: “ Datos incorrecto o existentes”	Pass
12	Caso de Prueba Agregar válido	Variables no null y datos válidos	msgbox: “Agregado Correctamente”	msgbox: “Agregado Correctamente”	Pass

## 2. Pruebas con Mock para el formulario de alumnos

- Agregar:

- Prueba agregar alumno válido:

```
[TestMethod]
0 referencias | Fiorella2919, Hace 35 minutos | 1 autor, 1 cambio
public void TestMockAgregarAlumnoCorrecto()
{
    Mock<ConecitarSQL> Alumno = new Mock<ConecitarSQL>();

    //variables de prueba
    string CodAlumno = "666666";
    string Nombres = "SOYLA";
    string APaterno = "BACA";
    string AMaterno = "DEL CAMPO";
    string Situacion = "EN RIESGO";
    string CodTutor = "004";
    string CodEscuela = "IN";

    bool Expected = true;

    //Indicamos el resultado del objeto, Que retorne una fila
    Alumno.Setup(a => a.AgregarAlumnoSiPosible(It.IsAny<string>(), It.IsAny<string>(), It.IsAny<string>(), It.IsAny<string>(),
        It.IsAny<string>(), It.IsAny<string>(), It.IsAny<string>())).Returns(true);

    //Para obtener el resultado usamos .Object y lo almacenamos
    bool actual = Alumno.Object.AgregarAlumnoSiPosible(CodAlumno, APaterno, AMaterno, Nombres, Situacion, CodTutor,
        CodEscuela);

    //Funcion Assert verifica si los resultados son iguales
    Assert.AreEqual(actual, Expected);

}
```

- Prueba agregar alumno invalido:

```
[TestMethod]
0 referencias
public void TestMockAgregarAlumnoIncorrecto()
{
    Mock<ConecitarSQL> Alumno = new Mock<ConecitarSQL>();

    //variables de prueba
    string CodAlumno = "666666";
    string Nombres = "SOYLA";
    string APaterno = ""; // variable faltante
    string AMaterno = "DEL CAMPO";
    string Situacion = "EN RIESGO";
    string CodTutor = "004";
    string CodEscuela = "IN";

    bool Expected = false;

    //Indicamos el resultado del objeto, Que retorne una fila
    Alumno.Setup(a => a.AgregarAlumnoSiPosible(It.IsAny<string>(), It.IsAny<string>(), It.IsAny<string>(),
        It.IsAny<string>(), It.IsAny<string>(), It.IsAny<string>(), It.IsAny<string>())).Returns(false);

    //Para obtener el resultado usamos .Object y lo almacenamos

    bool actual = Alumno.Object.AgregarAlumnoSiPosible(CodAlumno, APaterno, AMaterno, Nombres, Situacion, CodTutor,
        CodEscuela);
    //Funcion Assert verifica si los resultados son iguales
    Assert.AreEqual(actual, Expected);

}
```

- Select:

- Prueba seleccionar Alumnos

```

[TestMethod]
0 referencias
public void TestMockSeleccionarAlumnos()
{
    Mock<ConectarSQL> Conectar = new Mock<ConectarSQL>();

    //variable
    int Expected = 1;

    //Indicamos el resultado del objeto, Que retorne una fila
    Conectar.Setup(expression:a:ConectarSQL => a.SeleccionAlumnos()).Returns(1);
    //Para obtener el resultado usamos .Object y lo almacenamos

    int actual = Conectar.Object.SeleccionAlumnos();
    //Funcion Assert verifica si los resultados son iguales
    Assert.AreEqual(expected:actual, actual:Expected);

}

```

- **Editar:**

Prueba editar alumno válido:

```

public void EditarAlumnoCorrecto()
{
    Mock<ConectarSQL> Alumno = new Mock<ConectarSQL>();
    //variables de prueba
    string CodAlumno = "666666";
    string Nombres = "SOYLA";
    string APaterno = ""; // variable incompleta
    string AMaterno = "DEL CAMPO";
    string Situacion = "EN RIESGO";
    string CodTutor = "004";
    string CodEscuela = "IN";

    bool Expected = true;
    //Indicamos el resultado del objeto, Que retorne una fila
    Alumno.Setup(a => a.EditarAlumnoSiEsPosible(It.IsAny<string>(), It.IsAny<string>(),
        It.IsAny<string>(), It.IsAny<string>(), It.IsAny<string>())).Returns(true);
    //Para obtener el resultado usamos .Object y lo almacenamos

    bool actual = Alumno.Object.EditarAlumnoSiEsPosible(CodAlumno, APaterno, AMaterno, Nombres,
        Situacion, CodTutor, CodEscuela);
    //Funcion Assert verifica si los resultados son iguales
    Assert.AreEqual(actual, Expected);
}

```

Prueba editar alumno inválido:

```

public void EditarAlumnoIncorrecto()
{
    Mock<ConectarSQL> Alumno = new Mock<ConectarSQL>();
    //variables de prueba
    string CodAlumno = "666666";
    string Nombres = "SOYLA";
    string APaterno = ""; // variable incompleta
    string AMaterno = "DEL CAMPO";
    string Situacion = "EN RIESGO";
    string CodTutor = "004";
    string CodEscuela = "IN";

    bool Expected = false;
    //Indicamos el resultado del objeto, Que retorne una fila
    Alumno.Setup(a => a.EditarAlumnoSiEsPosible(It.IsAny<string>(), It.IsAny<string>(),
        It.IsAny<string>(), It.IsAny<string>(), It.IsAny<string>())).Returns(false);
    //Para obtener el resultado usamos .Object y lo almacenamos
    bool actual = Alumno.Object.EditarAlumnoSiEsPosible(CodAlumno, APaterno, AMaterno, Nombres,
        Situacion, CodTutor, CodEscuela);
    //Funcion Assert verifica si los resultados son iguales
    Assert.AreEqual(actual, Expected);
}

```

- **Buscar:**

- Prueba buscar alumno valido:

```

[TestMethod]
0 referencias
public void Test_Buscar_Alumno_Valido()
{

    Mock<ConectarSQL> Conectar = new Mock<ConectarSQL>();
    //variables
    string categoria = "Codigo";
    string buscar = "182905";

    int Expected = 1;

    //Indicamos el resultado del objeto, Que retorne una fila
    Conectar.Setup(a => a.TestBuscarAlumno(It.IsAny<string>(), It.IsAny<string>())).Returns(1);
    //Para obtener el resultado usamos .Object y lo almacenamos

    int actual = Conectar.Object.TestBuscarAlumno(categoria, buscar);
    //Funcion Assert verifica si los resultados son iguales
    Assert.AreEqual(actual, Expected);
}

```

- Prueba buscar alumno invalido:

```

        }

    [TestMethod]
    0 referencias
    public void Test_Buscar_Alumno_Invalido()
    {

        Mock<ConectarSQL> Conectar = new Mock<ConectarSQL>();
        //variables
        string categoria = "Codigo";
        string buscar = "jhon";

        int Expected = 0;

        //Indicamos el resultado del objeto, Que retorne una fila
        Conectar.Setup(a => a.TestBuscarAlumno(It.IsAny<string>(), It.IsAny<string>())).Returns(1);
        //Para obtener el resultado usamos .Object y lo almacenamos

        int actual = Conectar.Object.TestBuscarAlumno(categoria, buscar);
        //Funcion Assert verifica si los resultados son iguales
        Assert.AreEqual(actual, Expected);
    }
}

[TestMethod]
0 referencias
public void Test_Buscar_Alumno_Invalido1()
{
}

    Mock<ConectarSQL> Conectar = new Mock<ConectarSQL>();
    //variables
    string categoria = "Codigo";
    string buscar = "";

    int Expected = 0;

    //Indicamos el resultado del objeto, Que retorne una fila
    Conectar.Setup(a => a.TestBuscarAlumno(It.IsAny<string>(), It.IsAny<string>())).Returns(1);
    //Para obtener el resultado usamos .Object y lo almacenamos

    int actual = Conectar.Object.TestBuscarAlumno(categoria, buscar);
    //Funcion Assert verifica si los resultados son iguales
    Assert.AreEqual(actual, Expected);
}

```

- **Eliminar:**

```

        0 referencias
    public void Eliminar_Alumno_Valido()
    {

        Mock<ConectarSQL> Alumno = new Mock<ConectarSQL>();

        //variables de prueba
        string CodAlumno = "164563";

        bool Expected = true;

        //Indicamos el resultado del objeto, que nos retorne un mensaje de que se elimino
        Alumno.Setup(a => a.EliminarAlumnoCorrectamente(It.IsAny<string>())).Returns(true);
        //Para obtener el resultado usamos .Object y lo almacenamos

        bool actual = Alumno.Object.EliminarAlumnoCorrectamente(CodAlumno);
        //Funcion Assert verifica si los resultados son iguales
        Assert.AreEqual(actual, Expected);

    }

```

## Visualización de los resultados:

Explorador de pruebas					
Prueba		Duración	Rasgos	Mensaje de error	
▲ <span style="color: green;">✓</span> TestMock (10)		711 ms			
▲ <span style="color: green;">✓</span> TestMock (10)		711 ms			
▲ <span style="color: green;">✓</span> UnitMock (10)		711 ms			
<span style="color: green;">✓</span> EditarAlumnoCorrecto		675 ms			
<span style="color: green;">✓</span> EditarAlumnoIncorrecto		5 ms			
<span style="color: green;">✓</span> Eliminar_Alumno_Valido		4 ms			
<span style="color: green;">✓</span> Test_Buscar_Alumno_Invalido		9 ms			
<span style="color: green;">✓</span> Test_Buscar_Alumno_Invalido1		< 1 ms			
<span style="color: green;">✓</span> Test_Buscar_Alumno_Valido		2 ms			
<span style="color: green;">✓</span> TestMockAgregarAlumnoCorrecto		5 ms			
<span style="color: green;">✓</span> TestMockAgregarAlumnoIncorrecto		4 ms			
<span style="color: green;">✓</span> TestMockIngresoLoginCorrecto		6 ms			
<span style="color: green;">✓</span> TestMockSeleccionarAlumnos		1 ms			

## Cuadro de Pruebas Formulario Alumno:

N	Test case	Input	Expected output	Actual output	Remarks
<b>Agregar</b>					
1	Prueba unitaria agregar alumno válido	<b>Parámetros de entrada no vacíos</b>	True	True	Pass
2	Prueba unitaria agregar alumno invalido	<b>Algún parámetro de entrada vacío</b>	False	False	Pass
<b>Select</b>					
4	Prueba unitaria para seleccionar alumnos válidos	Ningun parametro de entrada	True	True	Pass
<b>Editar</b>					
5	Prueba unitaria para editar alumno válido.	<b>categoría:</b> cadena de caracteres. <b>editar:</b> cadena de caracteres.	True	True	Pass

<b>6</b>	Prueba unitaria para editar alumno invalido.	<b>categoría:</b> cadena de caracteres. <b>editar:</b> cadena de caracteres.	False	False	Pass
<b>Buscar</b>					
<b>7</b>	Prueba unitaria para buscar alumno válido.	<b>categoría:</b> cadena de caracteres. <b>buscar:</b> cadena de caracteres.	1	1	Pass.
<b>8</b>	Prueba unitaria para buscar alumno inválido.	<b>categoría:</b> cadena de caracteres. <b>buscar:</b> cadena de caracteres.	0	0	Pass.
<b>9</b>	Prueba unitaria para buscar alumno inválido.	<b>categoría:</b> cadena de caracteres. <b>buscar:</b> null.	0	0	Pass.
<b>Eliminar</b>					

<b>10</b>	Prueba unitaria para eliminar alumno válido.	<b>categoría:</b> cadena de caracteres. <b>eliminar:</b> cadena de caracteres.	True	True	Pass.
-----------	--	---	------	------	-------

### 3. Pruebas de Integración

#### 3.1 Login

Nº	Objetivo	Descripción de la prueba	Resultados Esperados
1	Validar el enlace entre el login y el formulario Docente y Alumno	Introducir el usuario y la contraseña y pulsar el botón <b>entrar</b> .	La aplicación nos dirige al panel del formulario Docente y Alumno y muestra una notificación de bienvenida al administrador logueado.
2	Validar el enlace entre el login y el formulario Tutorados	Introducir el usuario y la contraseña y pulsar el botón <b>entrar</b> .	La aplicación nos dirige al panel del formulario Tutorados y muestra una notificación de bienvenida al administrador logueado.

### 3.2 Agregar Alumno

Nº	Objetivo	Descripción de la prueba	Resultados Esperados
1	Validar el proceso de agregar alumno del módulo agregar y la comunicación con el DataGridView de alumnos.	Ingresar los datos correctos y válidos que se desean agregar :Código,Nombres,Cod.Tutor Ap. Materno,Ap Paterno,Escuela P. en los diferentes TextBoxs correspondientes y seleccionar el CheckedListBox en caso el alumno a agregar esté en Riesgo para luego pulsar el botón AGREGAR.	La aplicación abre una ventana de confirmación con un mensaje de “Se agregó correctamente” y el botón <b>ACEPTAR</b> para completar el proceso.Una vez presionado el botón la aplicación muestra el DataGridView de alumnos con el nuevo elemento agregado.
2	Invalidar el proceso de agregar alumno del módulo agregar y la comunicación con el DataGridView de alumnos si algún dato ingresado está vacío, a excepción de Situación.	Completar los TextBoxs correspondientes a los datos del alumno dejando uno o varios en vacío.	La aplicación muestra un mensaje de error : “Datos incorrectos o existentes” y el botón <b>ACEPTAR</b> que una vez pulsado la aplicación continúa mostrando el DataGridView de alumnos sin ninguna modificación o datos nuevos a la vista.
3	Invalidar el proceso de agregar alumno del módulo agregar y la comunicación con el DataGridView de alumnos si el dato ingresado CodAlumno es invalido o incorrecto y los demás datos correctos.	Agregar el código de alumno nuevo con un formato invalido. Ejemplo: -Código (formato correcto): 182908,tiene máximo 6 caracteres -Código agregado (formato incorrecto): A01B01B	La aplicación muestra un mensaje de error indicando:“Datos incorrectos o existentes” y el botón <b>ACEPTAR</b> que una vez pulsado la aplicación nos indica el TextBoxs del <b>Código</b> errado y continúa mostrando el DataGridView de alumnos sin ninguna modificación o datos nuevos a la vista.
4	Invalidar el proceso de agregar alumno del módulo agregar y la comunicación con el	Agregar los Nombres del alumno nuevo con un formato invalido. Ejemplo: -Nombres (formato	La aplicación muestra un mensaje de error indicando:“Datos incorrectos o existentes” y el botón <b>ACEPTAR</b> que una vez

	DataGridView de alumnos si el dato ingresado Nombres es invalido o incorrecto y los demás datos correctos.	correcto): LUCERO,LUZ LUCERO,etc. -Nombres agregados (formato incorrecto): LUCER4,456JK,10211.	pulsado la aplicación nos indica el TextBoxs del <b>Nombres</b> errado y continúa mostrando el DataGridView de alumnos sin ninguna modificación o datos nuevos a la vista.
5	Invalidar el proceso de agregar alumno del módulo agregar y la comunicación con el DataGridView de alumnos si el dato ingresado Cod. Tutor es invalido o incorrecto y los demás datos correctos.	Agregar el código de docente con un formato invalido o con un código que no existe en la base de datos. -Cód. Tutor (formato correcto): 001 -Cód. Tutor agregado (formato incorrecto): AO1,0KL,etc.	La aplicación muestra un mensaje de error indicando: "Datos incorrectos o existentes" y el botón <b>ACEPTAR</b> que una vez pulsada la aplicación nos indica el TextBoxs del <b>Cod. Tutor</b> errado y continúa mostrando el DataGridView de alumnos sin ninguna modificación o datos nuevos a la vista.
6	Invalidar el proceso de agregar alumno del módulo agregar y la comunicación con el DataGridView de alumnos si el dato ingresado Ap. Materno o Ap. Paterno es invalido o incorrecto y los demás datos correctos.	Agregar el Ap. Materno o Ap. Paterno con un formato invalido . -Ap. Materno o Ap. Paterno(formato correcto): QUISPE,MOLINA,etc. -Ap. Materno o Ap. Paterno agregado (formato incorrecto): 2LO1,QUIS98,etc.	La aplicación muestra un mensaje de error indicando: "Datos incorrectos o existentes" y el botón <b>ACEPTAR</b> que una vez pulsada la aplicación nos indica el TextBoxs del <b>Ap. Materno o Ap. Paterno</b> errado y continúa mostrando el DataGridView de alumnos sin ninguna modificación o datos nuevos a la vista.
	Invalidar el proceso de agregar alumno del módulo agregar y la comunicación con el DataGridView de alumnos si el dato ingresado Escuela P. es invalido o incorrecto y los demás datos correctos.	Agregar la Escuela P. con un formato invalido . - Escuela P.(formato correcto): IN,AR,CC,etc. - Escuela P. agregado (formato incorrecto):2569,Ilp98,etc.	La aplicación muestra un mensaje de error indicando: "Datos incorrectos o existentes" y el botón <b>ACEPTAR</b> que una vez pulsada la aplicación nos indica el TextBoxs del <b>Escuela P.</b> errado y continúa mostrando el DataGridView de alumnos sin ninguna modificación o datos nuevos a la vista.

### 3.3 Eliminar Alumno

Nº	Objetivo	Descripción de la prueba	Resultados Esperados
1	Validar proceso de eliminación de un alumno de la tabla alumnos	Pulsar el botón de <b>eliminar</b> (Figura de bote de basura) en la fila del alumno que se desea borrar.	La aplicación abre una ventana de confirmación con un mensaje de “¿Está seguro que desea eliminar a este alumno?” y los botones Si o No, para realizar la acción. De presionar Si se realiza la conexión, caso contrario regresamos al formulario de alumno sin realizar cambios.
2	Validar proceso de eliminación de un tutor de la tabla tutores	Pulsar el botón de <b>eliminar</b> (Figura de bote de basura) en la fila del tutor que desea borrar.	La aplicación abre una ventana de confirmación con un mensaje de “¿Está seguro que desea eliminar a este tutor?” y los botones Si o No, para realizar la acción. De presionar Si se realiza la conexión, caso contrario regresamos al formulario de tutor sin realizar cambios.

### 3.4 Buscar Alumno

Nº	Objetivo	Descripción de la prueba	Resultados Esperados
1	Validar la comunicación entre el módulo buscar por CodAlumno y DataGridView de Alumnos	Seleccionar el campo CodAlumno e introducir el código del alumno en el TextBox y pulsar botón de Buscar	La aplicación muestra DataGridView con los datos del alumno del código buscado.
2	Validar la comunicación entre el módulo buscar por APaterno y DataGridView de Alumnos	Seleccionar el campo APaterno,introducir el Apellido Paterno del alumno en el TextBox y pulsar el botón de Buscar	La aplicación muestra el DataGridView con los alumnos que tienen el Apellido Paterno Buscado.
3	Validar la	Seleccionar el campo	La aplicación muestra el

	comunicación entre el módulo buscar por AMaterno y DataGridView de Alumnos	AMaterno e introducir el Apellido Materno del alumno en el TextBox y pulsar el botón de Buscar	DatagridView con los alumnos que tienen el Apellido Materno Buscado.
4	Validar la comunicación entre el módulo buscar por Nombres y DataGridView de Alumnos	Seleccionar el campo Nombres e introducir los Nombres del alumno en el TextBox y pulsar el botón de Buscar	La aplicación muestra el DatagridView con los alumnos que tienen los Nombres buscados.
5	Validar la comunicación entre el módulo buscar por Situación y DataGridView de Alumnos	Seleccionar el campo Situación e introducir la situación del alumno en el TextBox y pulsar el botón de Buscar	La aplicación muestra el DataGridView de los alumnos que se encuentren con la situación buscada.
6	Validar la comunicación entre el módulo buscar por CodTutor y DataGridView de Alumnos	Seleccionar el campo CodTutor e introducir en el TextBox el Código del tutor del alumno y pulsar el botón de Buscar	La aplicación muestra DataGridView con los datos del alumno que corresponde al código del tutor buscado.
7	Validar la comunicación entre el módulo buscar por Escuela y DataGridView de Alumnos	Seleccionar el campo Escuela e introducir la escuela del alumno en el TextBox y pulsar botón de Buscar	La aplicación muestra el DataGridView de los alumnos que se encuentren en la escuela buscada.

### 3.5 Buscar Tutor

Nº	Objetivo	Descripción de la prueba	Resultados Esperados
1	Validar la comunicación entre el módulo buscar por CodAlumno y DataGridView de Tutores	Seleccionar el campo CodTutor e introducir el código del tutor en el TextBox y pulsar botón de Buscar	La aplicación muestra DataGridView con los datos del Tutor del código buscado.
2	Validar la comunicación entre el módulo buscar por	Seleccionar el campo APaterno,introducir el Apellido Paterno del tutor	La aplicación muestra el DataGridView con los tutores que tienen el Apellido Paterno

	APaterno y DataGridView de Tutores	en el TextBox y pulsar el botón de Buscar	Buscado.
3	Validar la comunicación entre el módulo buscar por APaterno y DataGridView de Tutores	Seleccionar el campo AMaterno e introducir el Apellido Materno del tutor en el TextBox y pulsar el botón de Buscar	La aplicación muestra el DataGridView con los tutores que tienen el Apellido Materno Buscado.
4	Validar la comunicación entre el módulo buscar por Nombres y DataGridView de Tutores	Seleccionar el campo Nombres e introducir los Nombres del tutor en el TextBox y pulsar el botón de Buscar	La aplicación muestra el DataGridView con los tutores que tienen los Nombres buscados.
5	Validar la comunicación entre el módulo buscar por Estado y DataGridView de Tutores	Seleccionar el campo Situación e introducir el estado del tutor en el TextBox y pulsar el botón de Buscar	La aplicación muestra el DataGridView de los tutores que se encuentren con el estado buscado.

### 3.6 Editar Alumno

Nº	Objetivo	Descripción de la prueba	Resultados Esperados
1	Validar la comunicación entre el módulo cambiar texto del botón de guardar y el botón con el icono de lápiz dentro del DataGridView de alumnos.	Pulsar el icono del lápiz dentro del DataGridView de alumnos.	La aplicación cambia el texto del botón “Guardar” por “Editar”.
2	Validar la comunicación del módulo editar alumno con el DataGridView de alumnos.	Pulsar el icono del lápiz dentro de una fila del DataGridView de alumnos.	La aplicación autocompleta los campos de entrada Alumno: CodAlumno, Nombres, APaterno, AMaterno, Situación, CodTutor, CodEscuela
3	Invalidar el proceso de editar alumno del	Editar algún dato del alumno y borrar el	La aplicación muestra un mensaje de error indicando qué

	módulo editar y la comunicación con el DataGridView de alumnos si algún dato editado está vacío, a excepción de Situación.	contenido dejándolo vacío.	campo de entrada se dejó vacío y continúa mostrando el DataGridView de alumnos sin ninguna modificación..
4	Invalidar el proceso de editar alumno del módulo editar y la comunicación con el DataGridView de alumnos si el código de alumno editado es invalido.	Editar el código de alumno con un formato invalido. Ejemplo: -Código (formato correcto): 182908 -Código editado (formato incorrecto): A01B01	La aplicación muestra un mensaje de error indicando que el código ingresado es invalido y continua mostrando el DataGridView de alumnos sin ninguna modificación.
5	Invalidar el proceso de editar alumno del módulo editar y la comunicación con el DataGridView de alumnos si el código de docente editado es invalido o no existe.	Editar el código de docente con un formato invalido o con un código que no existe en la base de datos. -Código (formato correcto): 001 -Código editado (formato incorrecto): AO1	La aplicación muestra un mensaje de error indicando si el código de docente editado es invalido o si no existe.

### 3.7 Editar Tutor

Nº	Objetivo	Descripción de la prueba	Resultados Esperados
1	Validar la comunicación entre el módulo cambiar texto del botón de guardar y el botón con el icono de lápiz dentro del DataGridView de tutores.	Pulsar el icono del lápiz dentro del DataGridView de tutores.	La aplicación cambia el texto del botón “Guardar” por “Editar”.
2	Validar la comunicación del módulo editar tutor con el DataGridView de tutores.	Pulsar el icono del lápiz dentro de una fila del DataGridView de tutores.	La aplicación autocompleta los campos de entrada tutor: CodTutor, Nombres, APaterno, AMaterno, Estado.
3	Invalidar el proceso de editar tutor del módulo editar y la	Editar algún dato del tutor y borrar el contenido dejándolo vacío.	La aplicación muestra un mensaje de error indicando qué campo de entrada se dejó vacío

	comunicación con el DataGridView de tutores si algún dato editado está vacío, a excepción de Estado.		y continúa mostrando el DataGridView de tutores sin ninguna modificación..
4	Invalidar el proceso de editar tutor del módulo editar y la comunicación con el DataGridView de tutores si el código de tutor editado es invalido.	Editar el código de tutor con un formato invalido. Ejemplo: -Código (formato correcto): 011 -Código editado (formato incorrecto): A01B01	La aplicación muestra un mensaje de error indicando que el código ingresado es invalido y continua mostrando el DataGridView de tutores sin ninguna modificación.