

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA DE SISTEMAS COMPUTACIONALES

INFORME DE PROYECTO FINAL

TÍTULO: **DISEÑO E IMPLEMENTACION DE UN SISTEMA DE REGISTRO DE PAGOS Y PLANIFICACION CURRICULAR DEL COLEGIO INTEGRAL CLASS**

**Autores:**

Cruzado Laiza, Joseph Lenin

Graus Vera, Jhoel David

Mariños Pérez, Milton Cesar

Ñiquin Valdez, Junnior Samir

**Curso:**

Modelamiento y Análisis de Software

**Docente del Curso:**

Diaz Amaya, Lourdes Roxana

Sede – Perú

2021

**Contenido**

[**I.** **RESUMEN.** 8](#_Toc75881334)

[**1.1.** **Motivación del proyecto** 8](#_Toc75881335)

[**1.2.** **Hallazgos** 9](#_Toc75881337)

[**1.3.** **Propuestas** 9](#_Toc75881338)

[**II.** **ANÁLISIS DEL PROBLEMA** 9](#_Toc75881339)

[**2.1.** **Antecedentes** 9](#_Toc75881341)

[**2.2.** **Identificación y formulación del problema** 10](#_Toc75881342)

[**2.2.1.** **Identificación del problema** 10](#_Toc75881343)

[**2.2.2.** **Formulación interrogativa del problema** 11](#_Toc75881344)

[**2.3.** **Definición de objetivos** 11](#_Toc75881346)

[**2.4.** **Requisitos del cliente** 11](#_Toc75881349)

[**III.** **GENERACIÓN SOLUCIONES** 11](#_Toc75881350)

[**IV.** **METODOLOGÍA DE DESARROLLO A UTILIZAR** 12](#_Toc75881357)

[**V.** **HERRAMIENTAS DE INGENIERÍA UTILIZADAS** 12](#_Toc75881358)

[**VI.** **JUSTIFICACION OPERATIVA , TECNICA, ECONOMICA** 12](#_Toc75881359)

[**6.1.** **Justificación Operativa** 12](#_Toc75881362)

[**6.2.** **Justificación Técnica** 12](#_Toc75881363)

[**6.3.** **Justificación Económica** 13](#_Toc75881364)

[**VII.** **RECOMENDACIONES FINALES** 13](#_Toc75881365)

[**7.1. Plan de despliegue de la solución** 13](#_Toc75881368)

[**7.2. Recomendaciones respaldadas por información y análisis** 13](#_Toc75881372)

[**X.** **REFERENCIAS O BIBLIOGRAFÍA** 13](#_Toc75881373)

[**XI.** **ANEXOS** 13](#_Toc75881374)

[**11.1.** **Modelamiento del negocio** 13](#_Toc75881375)

[**11.1.1.** **Unidades organizacionales** 13](#_Toc75881376)

[**11.1.2.** **Objetivos de negocio** 14](#_Toc75881377)

[**11.1.2.1.** **Control de estudio** 14](#_Toc75881378)

[**11.1.2.2.** **Gestión de pagos** 15](#_Toc75881379)

[**11.1.3.** **Reglas del negocio** 15](#_Toc75881380)

[**11.1.3.1.** **Control de estudio** 15](#_Toc75881381)

[**11.1.3.2.** **Gestión de pagos** 15](#_Toc75881382)

[**11.1.4.** **Glosario del negocio** 15](#_Toc75881383)

[**11.1.5.** **Identificación de procesos del negocio** 16](#_Toc75881384)

[**11.1.6.** **Identificación de actores del negocio** 16](#_Toc75881386)

[**11.1.7.** **Identificación de trabajadores de negocio** 16](#_Toc75881389)

[**11.1.8.** **Diagrama de casos de uso del negocio** 16](#_Toc75881393)

[**11.1.9.** **Realización de casos de uso del negocio** 17](#_Toc75881395)

[**11.1.9.1.** **Diagrama de actividad del negocio – subproceso** 17](#_Toc75881396)

[**11.1.10.** **Modelo de dominio** 22](#_Toc75881397)

[**11.2.** **Modelo de requerimientos del software** 22](#_Toc75881399)

[**11.2.1.** **Documento visión del software** 22](#_Toc75881400)

[**11.2.2.** **Diagrama de casos de uso** 26](#_Toc75881401)

[**11.2.3.** **Requerimientos funcionales y no funcionales** 28](#_Toc75881404)

[**11.2.4.** **Priorización de casos de uso** 30](#_Toc75881405)

[**11.2.5.** **Plan de iteraciones** 32](#_Toc75881406)

[**11.2.6.** **Especificaciones de casos de uso** 33](#_Toc75881407)

[**11.3.** **Realización del análisis y diseño** 40](#_Toc75881408)

[**11.3.1.** **Para cada iteración** 40](#_Toc75881409)

**INDICE DE TABLAS**

**INDICE DE FIGURAS**

**INDICE DE ANEXOS**

# **RESUMEN.**

* 1. **Motivación del proyecto**

El motivo por el que decidimos realizar este proyecto es porque vimos ampliamente los procesos que hoy en día requieren cambios en el ámbito escolar por la actual pandemia, por ello nos basamos en las operaciones que comúnmente se realizan de forma habitual para llegar a automatizar dichos procedimientos y de esta manera conseguir resultados más eficaces.

* 1. **Hallazgos**

## **Descripción de la organización.**

### **Razón social:**

* **Nombre:** Colegio Integral Class
* **Ubicación:** Carlos Marcelo Corne 345, Trujillo 13008
* **Teléfono:** 949 356 997
* **E-mail:** [colegio@integral.edu.pe](mailto:colegio@integral.edu.pe)

### **Logotipo de organización:**



***Ilustración 1 de google***

* 1. **Propuestas**
* Realizar un software que permita ejercer distintas operaciones en el rubro educativo, como son las matrículas, los pagos, las planificaciones curriculares y las sesiones de clase en un colegio.

# **ANÁLISIS DEL PROBLEMA**

1. 1. **Antecedentes**

* Nora Buena Guevara, Alexander Daniel, en la tesis ″Análisis, diseño e implementación de un sistema de información para la gestión académica de un instituto superior tecnológico″, en la Pontificia Universidad Católica del Perú, se centró en construir un software que incremente la calidad de los servicios que brinda a los alumnos de la institución de estudio. El plan del proyecto de implementación del sistema desarrollado se distribuyó en cuatro procesos principales: la gestión del proyecto, la concepción, la elaboración y la construcción. Finalmente, el investigador concluye que la metodología RUP en las fases elegidas para el desarrollo de este proyecto, guiaron de forma efectiva el desarrollo del software en todas sus etapas, desde el análisis hasta la implementación, brindando un mecanismo fiable y eficiente que describía cada componente considerado para la implementación final.
* Romero Galindo, Raúl Miguel, en la tesis de grado ″Análisis, diseño e implementación de un sistema de información aplicado a la gestión educativa en centros de educación especial″, en la Pontificia Universidad Católica del Perú, tuvo como propósito desarrollar una plataforma que posibilite la administración y atención de los planes curriculares especiales funcionales (programas educativos) y terapéuticos para personas con necesidades especiales. La administración del proyecto desarrollado adoptó prácticas establecidas por el Project Management Institute, y a nivel de software se consideró la metodología Agile Unified Process (AUP) por su afinidad y claridad de actividades en las etapas de diseño y construcción. Finalmente se indica que el monitoreo continuo del cronograma de proyecto y de la estructura de descomposición del trabajo posibilitó el cumplimiento de los tiempos estipulados, además de la culminación satisfactoria las fases de desarrollo del software junto con los entregables adecuados y establecidos por la metodología AUP.
  1. **Identificación y formulación del problema**
     1. **Identificación del problema**

Hasta hace un año atrás a la pandemia, dentro del ámbito educativo debemos de reconocer que la mayoría de las instituciones aplicaban un habitual método para el registro de notas e incluso para diversos procesos que se requerían, y este era mediante hojas impresas formuladas con la intención de que posteriormente se rellenen por los docentes con la información necesaria de cada estudiante que además se anexaba con ayuda de la herramienta Excel y no solo se completaban notas sino matriculas o asistencias que es lo con lo que se necesitaba contar por los diversos colegios. Por ello ante el cambio tan radical de un ámbito presencial a uno virtual dentro del rubro académico, se vio afectada la metodología con la que se contaba para tener los registros solicitados y al no tener un plan de respaldo ante este tipo de situaciones, se decidió recurrir así a lo más factible con lo que se cuenta hoy en día como lo son las herramientas de Microsoft para llevar una adecuada organización de información. Sin embargo, sabemos que de cierta manera se complica el hecho de buscar diversos registros en relación con cada estudiante porque toma un determinado tiempo detectarlos y si es crucial el acceso a dicha información se originaria inconvenientes. Por lo cual la institución educativa “INTEGRAL CLASS” particularmente ha sido una de las que más necesita la ayuda de un programa efectivo que facilite su organización logrando así tener un mejor control de información, ya que referido a este aspecto es que hemos visto en una situación un tanto complicada como agraviada a diversas instituciones producto de la pandemia. Es así que ante esta necesidad nosotros desarrollaremos un software adecuándose a las necesidades que aquejan a nuestro ámbito educativo que les permitirá ingresar todos los datos necesarios organizándolos ya sea en cuanto a matriculas, pensiones, etc. Con ayuda de este innovador software la institución tendrá garantizado que toda la información registrada se encontrará en un respectivo orden y no llevará tanto tiempo acceder a ella, garantizando además la calidad con la que cuenta la institución educativa en cuanto al manejo de operaciones realizadas de aquí en adelante.

* + 1. **Formulación interrogativa del problema**
* ¿Podrá el Software integrado mejorar el diseño y gestión de sus procesos de matrícula, pagos, sesión de clase y planificación curricular?
  1. **Definición de objetivos**

**Objetivo general:**

* Integrar un software al colegio Integral Class con la capacidad de mejorar las operaciones de matrícula, control de pagos y planificación curricular.

**Objetivos específicos:**

* Implementar un sistema de registro de notas para alumnos.
* Validar el registro de calificaciones de alumnos en el sistema.
* Implementar un sistema de matrículas para el registro de alumnos.
* Efectuar un registro interno de la planificación curricular y sesión de clase.
* Automatizar el proceso en la planificación curricular y gestión de pagos.
  1. **Requisitos del cliente**

# **GENERACIÓN SOLUCIONES**

1. 1. **Identificación y análisis de todas las restricciones, criterios y limitaciones**
2. Debido a la falta de experiencia al recolectar datos primarios, existe una gran probabilidad de que la implementación de los mismos este defectuosa.
3. Basar la investigación en un ámbito sin pandemia podría haber generado resultados más precisos.
4. Los procesos ya están definidos, sólo pueden ser rediseñados (matricula, pensión y planificación curricular); esto quiere decir que no se pueden crear nuevos procesos.
   1. **Generación y análisis de múltiples soluciones**

* Se pudo haber estudiado y analizado los procesos principales por más tiempo, logrando así una optimización de matrícula y registro de notas más precisa.
* La implementación de la investigación, se pudo haber realizado en una plataforma dedicada al sector educativo.
  1. **Criterios para seleccionar la solución**
* Impacto ambiental
* Seguridad de datos
* Entorno Político
* Presupuesto
* Optimización de procesos
* Sostenibilidad
* Impacto educativo

# **METODOLOGÍA DE DESARROLLO A UTILIZAR**

Se utilizo la metodología clásica para elaborar la solución:

1. **Proceso unificado racional(RUP):** utiliza el enfoque de la orientación a objetos en su diseño y está diseñado y documentado el uso de la notación UML ( Unified Modeling Language ) para ilustrar los procesos en acción. Utiliza técnicas y prácticas probadas comercialmente.
2. Programación orientada a objeto(OOP): Es un paradigma de programación que usa los objetos en su iteración para diseñar aplicaciones, con las técnicas que requiera.

Se eligió la metodología RUP, se adapto a nuestra problemática, enfocándose en el diseño y documentación de cada proceso, y se avala con su uso comercial que ha tenido, previniendo resultados en situaciones realistas.

# **HERRAMIENTAS DE INGENIERÍA UTILIZADAS**

Las herramientas que se utilizaron fueron:

1. Draw.io: Se utilizo para la creación de diagramas del proyecto.
2. Word: Se utilizo para la descripción de cada proceso en un archivo de texto.
3. Visual Studio: Se utilizo para la creación de interfaces y programación.
4. Git Hub: Se utilizo como repositorio grupal y actividades de cada trabajo realizado.
5. Enterprise: Se utilizo para hacer diagramas de negocio, basado en casos de uso y objetos.
6. SQL: Se utilizo para la creación de la base de datos y las consultas que se van a realizar.

Concluimos que el uso correcto de las herramientas, nos ayudo a alcanzar el resultado esperado.

# **JUSTIFICACION OPERATIVA, TECNICA, ECONOMICA**

2. 1. **Justificación Operativa**

Se garantizará el acceso al sistema que solo el personal autorizado y/o registrado que ingresaran con un código que permitirá la navegación a los procesos a realizar.

* 1. **Justificación Técnica**

La investigación cuenta como propósito la integración de un software al colegio con el objetivo principal de lograr una mejoría notable en cuanto a las operaciones realizadas en distintas gestiones académicas. Por ello el dimensionamiento del proyecto se encuentra enfocado en las necesidades de un sistema adaptado a los tiempos de hoy en plena pandemia donde se debe contar con el menor contacto posible. Con la integración del software la institución será capaz de llevar adelante un mejor orden en cuanto a los archivos y poder así cumplir con los objetivos y metas previstos.

* 1. **Justificación Económica**

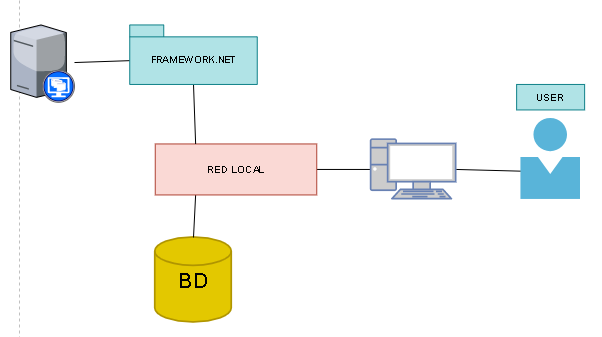
La investigación tendrá una repercusión en la economía de la institución, porque hará procesos de forma digital sin necesidad de elementos físicos para su gestión, por ejemplo, excesivo uso de papel, etc. Este proyecto se justifica porque aportará ventajas económicas para la institución en general, en el uso de un servicio óptimo para cada gestión.

# **RECOMENDACIONES FINALES**



**7.1. Plan de despliegue de la solución**

La solución de desplegará en un servidor que sea capaz de usar framework .net y un motor de base de datos SQL-Server, a través de red – Local.

****

***Ilustración 2 elaborado por equipo***

**7.2. Recomendaciones respaldadas por información y análisis**

* Para el ingreso de información que se dará masivamente en alguno de los mantenedores, se podrían realizar varias iteraciones de forma que no sea necesario volver a la interfaz principal para la realización de la misma operación para el ingreso de un paquete de datos o información.
* Bajo circunstancias de pandemia existe un posible escenario en el que las clases se digan dictando de manera remota y/o virtual, por ello se piensa que puede haber una expansión con respecto al límite de vacantes o en otros procedimientos que requieran de una presencia colectiva.

# **REFERENCIAS O BIBLIOGRAFÍA**

* Piattini et al., 2007. Análisis y diseño de Aplicaciones Informáticas de Gestión. Una perspectiva de Ingeniería del Software. Ra-Ma. Junio 2007.
* Pressman, 2005. Ingeniería del Software: Un Enfoque Práctico. 6ª Edición. McGraw-Hill, 2005.
* Pfleeger, 2002. Ingeniería del Software. Teoría y Práctica. Prentice Hall, 2002.
* Sommerville, 2005. Ingeniería del Software. 7ª Edición, Addison-Wesley. Julio 2005.

# **ANEXOS**

* 1. **Modelamiento del negocio**
     1. **Unidades organizacionales**



***Ilustración 3 Elaborado por equipo***

* + 1. **Objetivos de negocio**
       1. **Control de estudio**
* Asegurar la evaluación diagnostica de cada estudiante
* Garantizar que se registren las notas de todos los alumnos en un periodo determinado.
* Garantizar que se registren las evidencias de los estudiantes dentro de la fecha establecida.
  + - 1. **Gestión de pagos**
* Garantizar el registro de matrículas de los estudiantes en tiempo óptimo.
* Garantizar que las vacantes de matrícula cumplan con sus especificaciones de grado académico.
* El pago de pensiones se realizará en los tiempos pactados del tutor.
  + 1. **Reglas del negocio**
       1. **Control de estudio**
* Las notas no podrán ser modificadas si han transcurrido 24hrs de haber sido ingresadas.
* Las notas luego de ser ingresadas deben ser enviadas al director para su aprobación.
* El software permitirá ingresar notas solo si estas están entre el rango de 0 a 20.
  + - 1. **Gestión de pagos**
* Los estudiantes deben contar con documento de identidad para su matrícula.
* Los alumnos solo podrán matricularse según las fechas asignadas.
* El alumno(a) debe tener antecedentes académicos.
* Ningún alumno puede matricularse si debe cursos reprobados.
* Se aceptarán estudiantes hasta límite de vacantes.
  + 1. **Glosario del negocio**
* **Evaluación:** Valoración de los conocimientos del alumno.
* **Vacante:** Es la disponibilidad del acceso a una matrícula.
* **Tutor:** Persona encargada de cuidar, velar y responder por el alumno, suelen ser los padres.
* **Pensión:** Cantidad de dinero a pagar continuamente por la estancia en el colegio.
* **Calendarización:** Es un calendario resumido de todo el año escolar.
* **Consolidado:** Es un registro de notas del año anterior.
* **Evaluación Diagnostica:** Es una evaluación que se encarga de diagnosticar el nivel académico de los estudiantes.
  + 1. **Identificación de procesos del negocio**



***Ilustración 4 elaborado por equipo***

* + 1. **Identificación de actores del negocio**
* Alumno
* Tutor
  + 1. **Identificación de trabajadores de negocio**
* Director
* Docente
* Secretaria
  + 1. **Diagrama de casos de uso del negocio**



***Ilustración 5 elaborado por equipo***

* + 1. **Realización de casos de uso del negocio**
       1. **Diagrama de actividad del negocio – subproceso**

Diagrama de Planificación



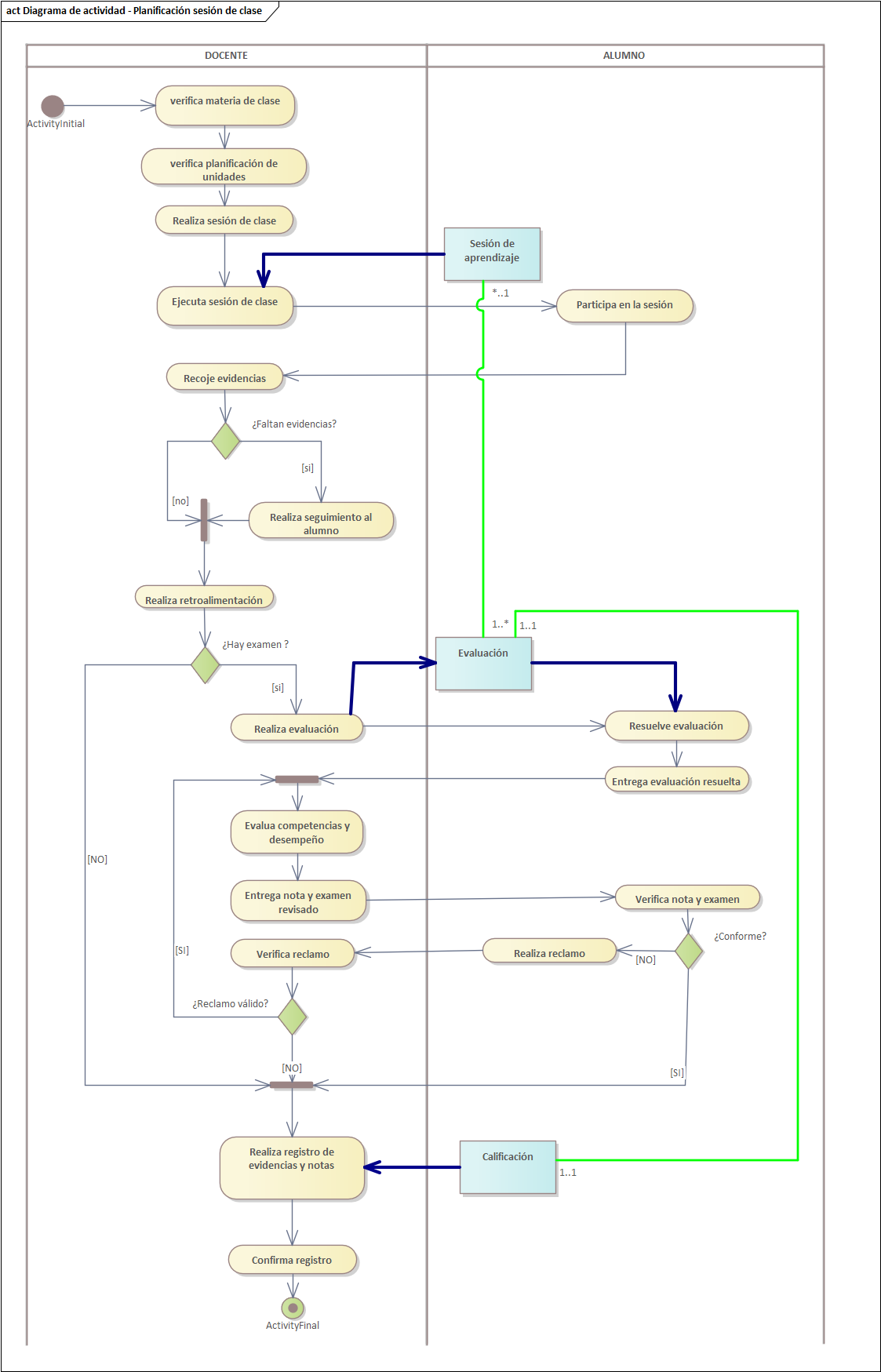
***Ilustración 6 elaborado por equipo***

Diagrama de actividad de pension



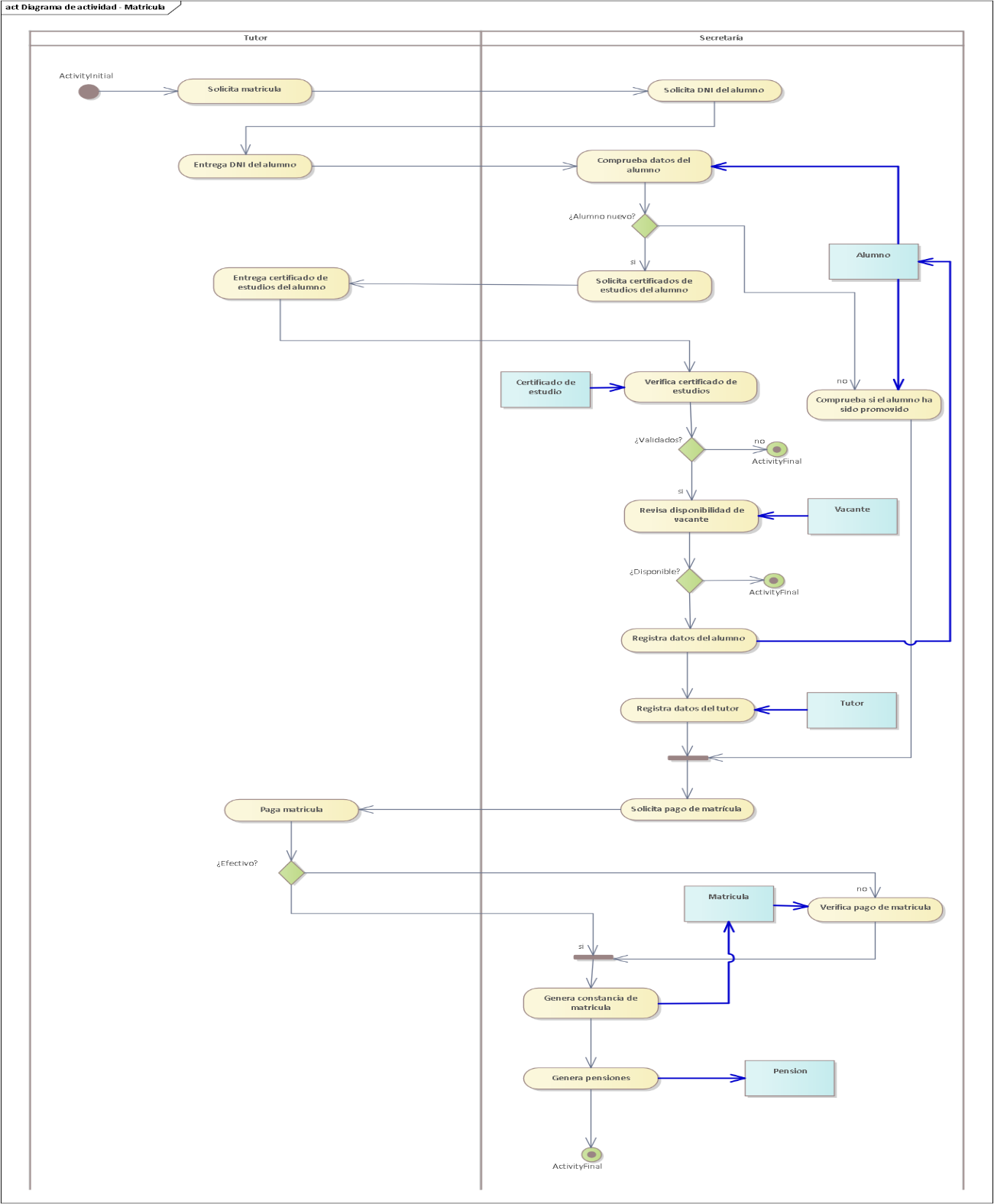
***Ilustración 7 elaborado por equipo***

Diagrama planificación sesión de clase (Control De Estudio)

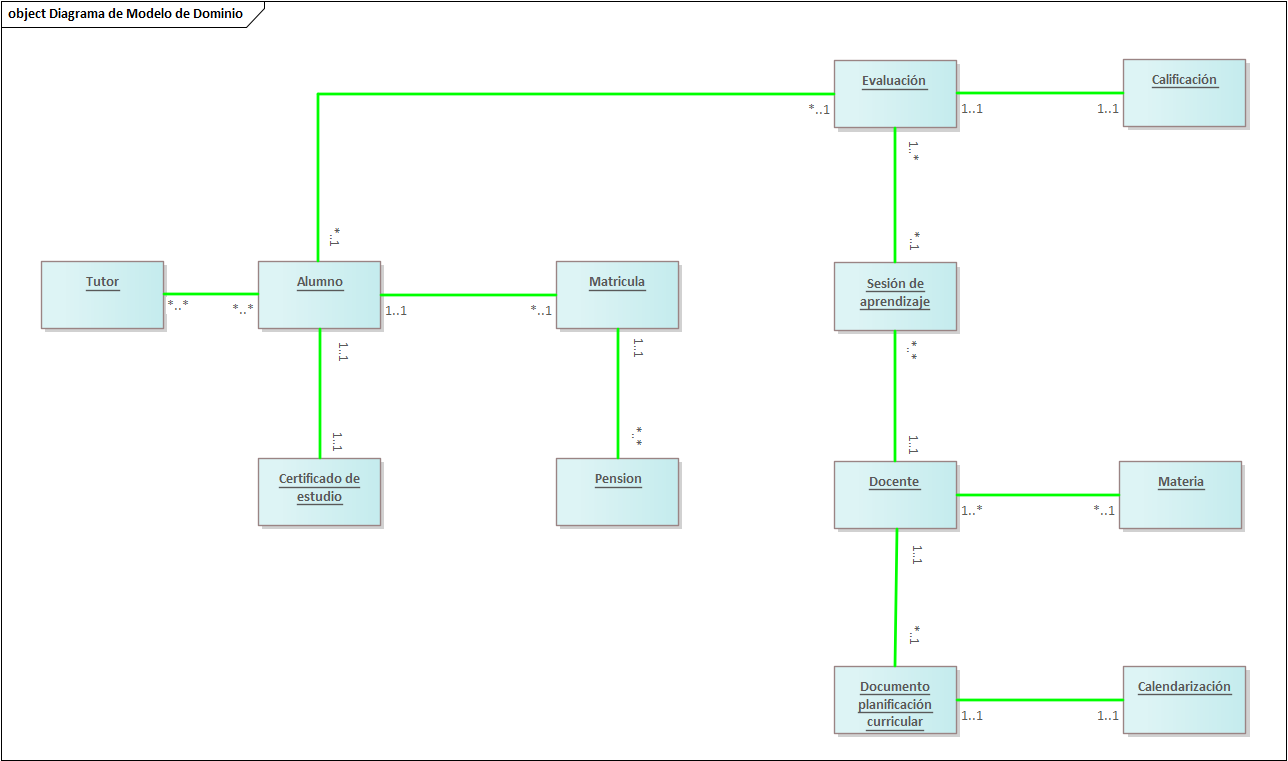


***Ilustración 8 elaborado por equipo***

Diagrama de actividad Matricula



***Ilustración 9 elaborado por equipo***

* + 1. **Modelo de dominio**

***Ilustración 10 elaborado por equipo***

* 1. **Modelo de requerimientos del software**
     1. **Documento visión del software**

**TABLA DE USUARIOS**

|  |  |
| --- | --- |
| **Rol** | **Descripción** |
| Alumno | Es el encargado de realizar las evaluaciones y consultar calificaciones que ejecuta el docente. |
| Tutor | Es el encargado de ofrecer los datos del estudiante para la respectiva matrícula y cancelar las pensiones. |
| Docente | Es el encargado de realizar la planificación curricular y la sesión de clase. |
| Secretaria | Es el encargado de realizar la matricula del alumno y registrar los pagos de las pensiones. |

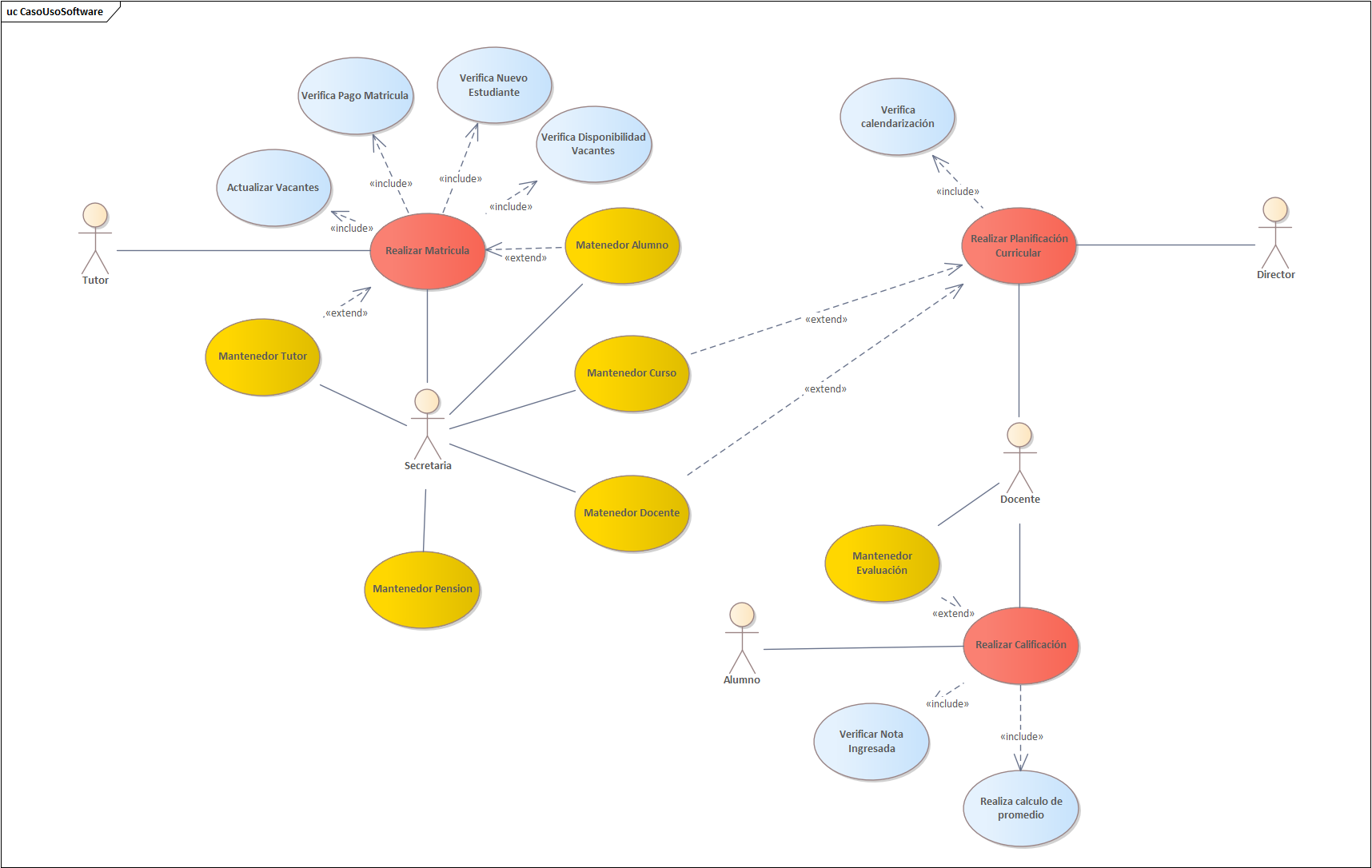
**NECESIDADES DE SOFTWARE**

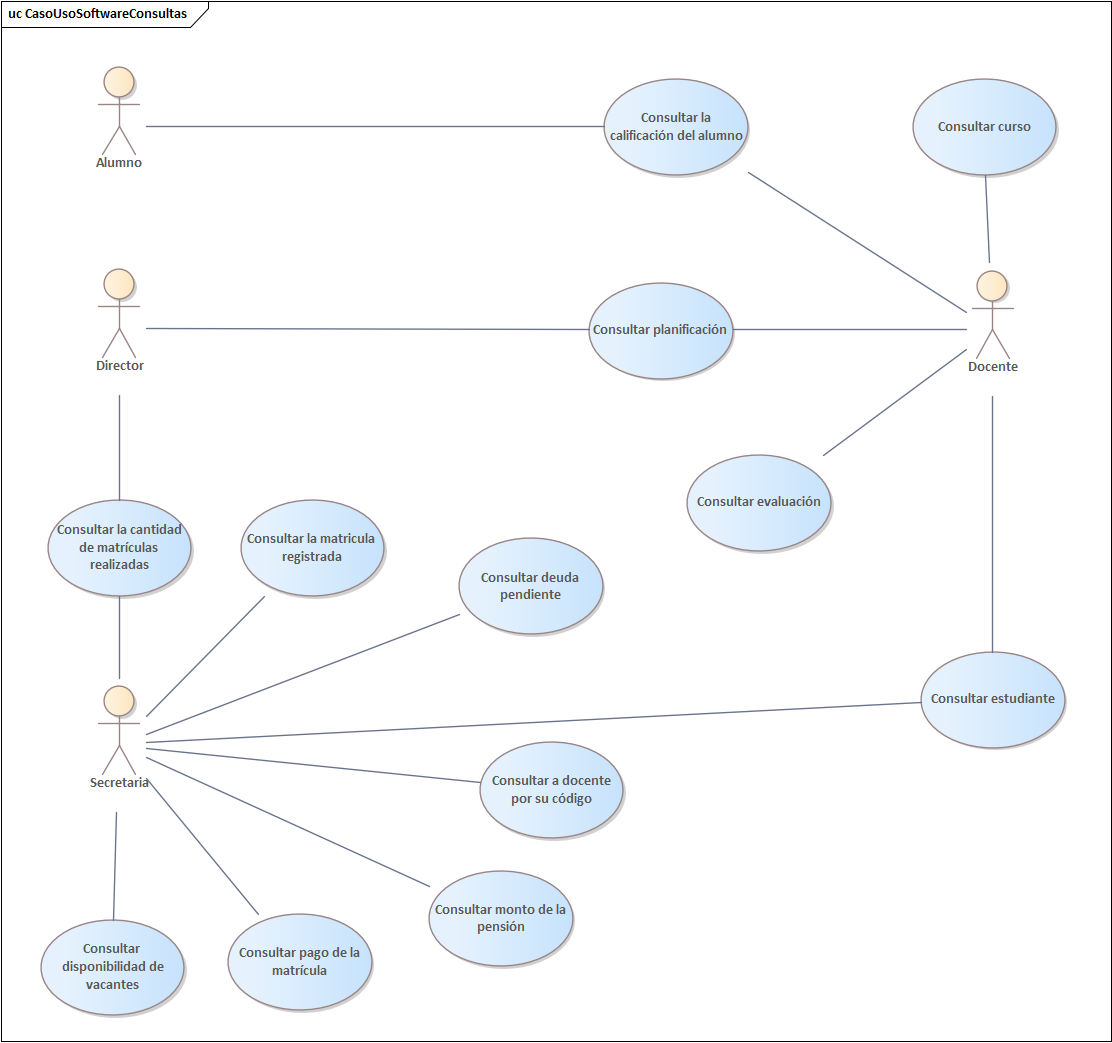
| **Id** | **Descripción** |
| --- | --- |
| NEC – 01 | El sistema deberá permitir registrar, modificar, inhabilitar y mostrar la matrícula del estudiante. |
| NEC – 02 | El sistema deberá registrar, modificar, anular y mostrar el pago de pensiones. |
| NEC – 03 | El sistema deberá permitir registrar y mostrar las calificaciones de los estudiantes. |
| NEC – 04 | El sistema deberá Registra y mostrar la planificación curricular |
| NEC – 05 | El sistema deberá Registrar, inhabilitar, modificar y mostrar al alumno. |
| NEC – 06 | El sistema deberá registrar, modificar, inhabilitar y mostrar al docente. |
| NEC – 07 | El sistema deberá Registrar, modificar, inhabilitar y mostrar la materia o curso. |
| NEC – 08 | El sistema deberá Registrar, inhabilitar, modificar y mostrar al tutor |
| NEC – 09 | El sistema deberá Registrar y mostrar la evaluación |
| NEC – 10 | Consultar la matrícula registrada por código de matrícula. |
| NEC – 11 | Consultar la cantidad de matrículas realizadas por fecha. |
| NEC – 12 | Consultar la cantidad de vacantes disponibles. |
| NEC – 13 | Consultar pago de matrícula por código de estudiante. |
| NEC – 14 | Consultar pago de pensión por código de estudiante. |
| NEC – 15 | Consulta deuda pendiente por código de estudiante. |
| NEC – 16 | Consultar la calificación del alumno por código de estudiante. |
| NEC – 17 | Consultar planificación curricular por id curso. |
| NEC – 18 | Consultar planificación curricular por id docente. |
| NEC – 19 | Consulta docente por código de docente. |
| NEC – 20 | Consulta materia por código o nombre de materia. |
| NEC – 21 | Consultar alumno por código de alumno. |
| NEC – 22 | Consultar evaluación por código de evaluación. |

CARACTERÍSTICAS DE SOFTWARE

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Id** | **Id NEC** | | | **Descripción** |
| CAR – 01 | NEC – 01 | | | El sistema deberá permitir registrar y mostrar la matricula ingresando código de estudiante. |
| CAR – 02 | NEC – 01 | | | El sistema deberá permitir modificar o inhabilitar la matrícula. |
| CAR – 03 | NEC – 01 | | | Verificar si es un estudiante es nuevo. |
| CAR – 04 | NEC – 01 | | | Verificar disponibilidad de vacantes por año académico. |
| CAR – 05 | NEC – 02 | | | El sistema deberá permitir registrar las pensiones ingresando código de estudiante, fecha de inicio y fin. |
| CAR – 06 | NEC – 02 | | | El sistema deberá permitir mostrar las pensiones por código de estudiante. |
| CAR – 07 | NEC – 02 | | | El sistema deberá poder actualizar el estado de la deuda luego de haber cancelado el monto. |
| CAR – 08 | NEC – 02 | | | El sistema deberá verificar el estado de las pensiones, en caso de no haber deuda, no podrá realizar pago. |
| CAR – 09 | NEC – 03 | | | El sistema deberá permitir registrar la calificación ingresando el código de evaluación. |
| CAR – 10 | NEC – 03 | | | El sistema deberá poder mostrar el detalle de la calificación luego de ser registrada. |
| CAR – 11 | NEC – 03 | | | El sistema deberá verificar las calificaciones registradas anteriormente, para que no se sobrescriban. |
| CAR – 12 | NEC – 04 | | | El sistema deberá registrar la planificación curricular ingresando código de docente. |
| CAR – 13 | NEC – 04 | | | El sistema deberá mostrar el detalle de la planificación curricular, luego de ser registrada. |
| CAR – 14 | NEC – 04 | | | El sistema deberá verificar si hay planificaciones con el mismo curso. |
| CAR – 15 | NEC – 05 | | | El sistema deberá Registrar al estudiante, ingresando nombre, grado, edad etc. |
| CAR – 16 | NEC – 05 | | | El sistema deberá modificar y inhabilitar al estudiante, ingresando código de estudiante. |
| CAR – 17 | NEC – 05 | | | El sistema deberá mostrar al estudiante luego de ser registrado. |
| CAR – 18 | NEC – 06 | | | El sistema deberá Registrar al docente, ingresando, nombre, edad, nivel de educación, etc. |
| CAR – 19 | NEC – 06 | | | El sistema deberá modificar datos e inhabilitar al docente, ingresando código de docente. |
| CAR – 20 | NEC – 06 | | | El sistema deberá mostrar al docente luego de ser registrado. |
| CAR – 21 | NEC – 07 | | | El sistema deberá Registrar a la materia o curso, ingresando nombre de curso, grado, año etc. |
| CAR – 22 | NEC – 07 | | | El sistema deberá poder modificar o inhabilitar curso ingresando código de curso. |
| CAR – 23 | NEC – 07 | | | El sistema deberá poder mostrar el curso luego de ser registrado. |
| CAR – 24 | NEC – 08 | | | El sistema deberá Registrar al tutor, ingresando DNI, nombre, edad, fecha de nacimiento, etc. |
| CAR – 25 | NEC – 08 | | | El sistema deberá modificar o inhabilitar al tutor, ingresando DNI del tutor. |
| CAR – 26 | NEC - 08 | | | El sistema deberá mostrar al tutor luego de ser registrado. |
| CAR – 27 | NEC – 09 | | | El sistema deberá Registrar la evaluación ingresando, nombre, curso etc. |
| CAR – 28 | NEC – 09 | | | El sistema deberá mostrar la evaluación luego de ser registrada por código de evaluación. |
| CAR – 29 | NEC – 10 | | | El sistema deberá consultar la matrícula por código de matrícula mostrando el detalle matrícula. |
| CAR – 30 | NEC – 11 | | | El sistema deberá consultar la cantidad de matrículas registradas, ingresando la fecha. |
| CAR – 31 | NEC – 12 | | | El sistema deberá consultar las vacantes disponibles por fecha de año académico. |
| CAR – 32 | NEC – 13 | | | El sistema deberá consultar el pago de matrícula por código de estudiante, mostrando el detalle del pago. |
| CAR – 33 | NEC – 14 | | | El sistema deberá consultar el pago de pensión por código de estudiante, mostrando el detalle del pago. |
| CAR – 34 | NEC – 15 | | | El sistema deberá consultar la deuda pendiente por código de estudiante, mostrando únicamente el detalle de la deuda. |
| CAR – 35 | NEC – 16 | | | El sistema deberá Consultar la calificación del estudiante por código de alumnos y posteriormente mostrar su detalle calificación. |
| CAR – 36 | | NEC – 19 | El sistema deberá consultar la planificación curricular por id curso, mostrando el detalle de la planificación encontrada. | |
| CAR – 37 | | NEC – 20 | El sistema deberá consultar la planificación por id docente, mostrando el detalle de la planificación. | |
| CAR – 38 | | NEC – 23 | El sistema deberá consultar al docente por código docente, mostrando datos del docente. | |
| CAR – 39 | | NEC – 24 | El sistema deberá consultar la materia por código de materia o nombre de materia, mostrando los datos de la materia. | |
| CAR – 40 | | NEC – 25 | El sistema deberá consultar al alumno por código de alumno, mostrando los datos del alumno. | |
| CAR – 41 | | NEC – 26 | El sistema deberá consultar la evaluación por id de evaluación, mostrando el detalle de la evaluación | |

* + 1. **Diagrama de casos de uso**





* + 1. **Requerimientos funcionales y no funcionales**
* REQUERIMIENTOS FUNCIONALES

|  |  |
| --- | --- |
| **ID** | **Descripción** |
| RF01 | El sistema deberá permitir registrar, modificar, inhabilitar y mostrar la matrícula del estudiante. |
| RF02 | El sistema deberá registrar, modificar, anular y mostrar el pago de pensiones. |
| RF03 | El sistema deberá permitir registrar y mostrar las calificaciones de los estudiantes. |
| RF04 | El sistema deberá Registra y mostrar la planificación curricular |
| RF05 | El sistema deberá Registrar, inhabilitar, modificar y mostrar al alumno. |
| RF06 | El sistema deberá registrar, modificar, inhabilitar y mostrar al docente. |
| RF07 | El sistema deberá Registrar, modificar, inhabilitar y mostrar la materia o curso. |
| RF08 | El sistema deberá Registrar, inhabilitar, modificar y mostrar al tutor |
| RF09 | El sistema deberá Registrar y mostrar la evaluación |
| RF10 | Consultar la matrícula registrada por código de matrícula. |
| RF11 | Consultar la cantidad de matrículas realizadas por fecha. |
| RF12 | Consultar la cantidad de vacantes disponibles. |
| RF13 | Consultar pago de matrícula por código de estudiante. |
| RF14 | Consultar pago de pensión por código de estudiante. |
| RF15 | Consulta deuda pendiente por código de estudiante. |
| RF16 | Consultar la calificación del alumno por código de estudiante. |
| RF17 | Consultar planificación curricular por id curso. |
| RF18 | Consultar planificación curricular por id docente. |
| RF19 | Consulta docente por código de docente. |
| RF20 | Consulta materia por código o nombre de materia. |
| RF21 | Consultar alumno por código de alumno. |
| RF22 | Consultar evaluación por código de evaluación. |

* REQUERIMIENTOS NO FUNCIONALES

|  |  |
| --- | --- |
| RNF01 | * El sistema debe ser interactivo, con interfaz sencillo manejo para el usuario, y con características graficas de una institución educativa. |
| De apariencia o interfaz externa |
| RNF02 | * El sistema debe ser fácil de usar para el personal de la institución educativa y para los estudiantes. |
| De Usabilidad |
| RNF03 | * El sistema podrá ser usado en la plataforma Windows, a través de una aplicación de escritorio. |
| De portabilidad |
| RNF04 | * Se debe restringir las operaciones según el tipo de rol que posea el usuario que ingrese al sistema. |
| De seguridad de accesos |
| RNF05 | * El sistema seguirá la arquitectura de N-Capas. * El sistema contemplará el uso del Enterprise Architect para su interpretación, análisis, elaboración de modelos de negocio y del sistema. * El sistema se implementará en Visual Studio 2015 o 2019 y como Motor de Base de datos en SQL server Management Studio 2014. |
| De restricciones en el Diseño y la Implementación |

* + 1. **Priorización de casos de uso**

|  |  |  |
| --- | --- | --- |
| CRITERIOS | PESO | RANGO |
| RI: Riesgo tecnológico, Tecnología Nueva, Interface especializada, hardware o dispositivos nuevos, etc. | 3 | 0-3 |
| SA: Significativo para la arquitectura. | 2 | 0-3 |
| NC: Naturaleza critica, de valor para el negocio | 1 | 0-3 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| CASO DE USO | RI (3) | SA (2) | NC (1) | PUNTAJE |
| Realiza matricula | 3 | 3 | 3 | 18 |
| Realiza planificación curricular | 3 | 3 | 3 | 18 |
| Realiza calificaciones | 3 | 2 | 2 | 15 |
| Mantenedor alumno | 2 | 3 | 2 | 15 |
| Mantenedor docente | 2 | 3 | 3 | 15 |
| Mantenedor curso | 2 | 2 | 1 | 11 |
| Mantenedor tutor | 1 | 1 | 2 | 7 |
| Mantenedor Evaluación | 2 | 2 | 2 | 12 |
| Mantenedor Pensión | 2 | 2 | 2 | 12 |
| Consultar la calificación del alumno | 2 | 1 | 1 | 09 |
| Consultar Curso | 2 | 1 | 1 | 09 |
| Consultar planificación | 2 | 1 | 1 | 09 |
| Consultar evaluación | 2 | 1 | 1 | 09 |
| Consultar la cantidad de matrículas realizadas | 2 | 1 | 1 | 09 |
| Consultar deuda pendiente | 2 | 1 | 1 | 09 |
| Consultar docente | 2 | 1 | 1 | 09 |
| Consultar monto de la pensión | 2 | 1 | 1 | 09 |
| Consultar estudiante | 2 | 1 | 1 | 09 |
| Consultar pago matricula | 2 | 1 | 1 | 09 |
| Consultar disponibilidad de vacantes | 2 | 1 | 1 | 09 |

|  |  |  |
| --- | --- | --- |
| PRIORIDAD | CASOS DE USO | COMENTARIO |
| Alta | * Realiza matricula * Realiza planificación curricular * Realiza calificaciones * Mantenedor alumno * Mantenedor docente | Puntación alta en todos los criterios de clasificación. Difícil de añadir tarde, deben planificarse para las primeras iteraciones del proyecto. |
| Media | * Mantenedor curso * Mantenedor tutor * Mantenedor Evaluación * Mantenedor Pensión | Proceso importante pero no demasiado difícil.  Forman parte de los requisitos funcionales |
| Baja | * Consultar la calificación del alumno * Consultar Curso * Consultar planificación * Consultar evaluación * Consultar la cantidad de matrículas realizadas * Consultar deuda pendiente * Consultar docente * Consultar monto de la pensión * Consultar estudiante * Consultar pago matricula * Consultar disponibilidad de vacantes | Fácil, efecto mínimo en la arquitectura. |

* + 1. **Plan de iteraciones**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| N° de iteración: | Contenido | Fecha de Inicio | Fecha de Limite | Responsables |
| 1 | -Mantenedor Estudiante  -Transacción Matricula | 3/06/2021 | 09/06/2021 | Mariños Pérez, Milton Cesar |
| 2 | Mantenedor Docente  Mantenedor Curso | 3/06/2021 | 13/06/2021 | Graus Vera, Jhoel David |
| 3 | Mantenedor Evaluación | 13/06/2021 | 15/06/2021 | Cruzado Laiza, Joseph Lenin |
| 4 | Transacción Planificación  Curricular | 15/06/2021 | 18/06/2021 | Cruzado Laiza, Joseph Lenin |
| 5 | Transacción Calificación | 18/06/2021 | 21/06/2021 | Ñiquin Valdez, Junnior Samir |

1° Iteración: Mariños Pérez, Milton Cesar

* Realizar un mantenedor Estudiante por lo cual implementara un modelado de diseño y interfaces para el registro, modificación y inhabilitación de los datos del Estudiante.
* Realizar una Transacción Matrícula por lo cual implementara un modelado de diseño e interfaces para el registro de los datos de la Matricula.

2° Iteración: Graus Vera, Jhoel David

* Realizar un mantenedor Docente por lo cual implementara un modelado de diseño y interfaces para el registro, modificación y inhabilitación de los datos del Docente.
* Realizar una Transacción Matrícula por lo cual implementara un modelado de diseño y interfaces para el registro de los datos de la Matricula.

3° Iteración: Cruzado Laiza, Joseph Lenin

* Realizar una Mantenedor Evaluación por lo cual implementara un modelado de diseño y interfaces para el registro de los datos de la Matricula.

4° Iteración: Cruzado Laiza, Joseph Lenin

* Realizar una Transacción Planificación Curricular por lo cual implementara un modelado de diseño y interfaces para el registro de los datos de la Matricula.

2° Iteración: Ñiquin Valdez, Junnior Samir

* Realizar una Transacción Calificación por lo cual implementara un modelado de diseño y interfaces para el registro de los datos de la Matricula.
  + 1. **Especificaciones de casos de uso**

|  |  |  |  |
| --- | --- | --- | --- |
| **IDENTIFICADOR:**  CU01 | **NOMBRE:** Matricula | | |
| **CATEGORIA:** Core | | **COMPLEJIDAD:** Alta | **PRIORIDAD:** Alta |
| **ACTORES:** Secretaria | | | |
| **PROPOSITO:** Realizar el registro de la matrícula de alumnos. | | | |
| **PRECONDICIONES:**   1. Debe haber disponibilidad de vacantes o no se registrará al alumno. | | | |
| **FLUJO BÁSICO:**  La secretaria elige la opción **“Matricular”**   1. El sistema muestra la interfaz de matrícula con la opción: “Confirmar”. 2. El sistema muestra las opciones: “Nuevo estudiante”, “Estudiante existente”. 3. La secretaria selecciona la opción de “Estudiante existente”.   Si el estudiante no existe se registra estudiante.   1. La secretaria registra la matrícula. 2. El sistema muestra las opciones: “Tutor nuevo”, “Tutor existente”. 3. La secretaria selecciona la opción “Tutor existente”.   Si el tutor no existe se registra al tutor.   1. La secretaria selecciona el botón “Confirmar”. 2. El sistema registra al estudiante. 3. El sistema registra la matricula. 4. El sistema registra al tutor. 5. El sistema actualiza la cantidad de vacantes. | | | |
| **POSCONDICION:** Registro de matrícula almacenada. | | | |
| **FLUJOS ALTERNATIVOS:**   1. El sistema verifica si aún está dentro del rango de vacantes. | | | |
| **REQUERIMENTOS ESPECIALES O SUPLEMENTARIOS:**   1. Los mensajes de advertencia aparecen cuando no se efectuó la matricula exitosamente. 2. Los campos se deben llenar obligatoriamente. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **IDENTIFICADOR:** CU02 | **NOMBRE:** Mantenedor Estudiante | | |
| **CATEGORÍA:** Mantenedor | | **COMPLEJIDAD:** Media | **PRIORIDAD:** Alta |
| **ACTORES:** Secretaria | | | |
| **PROPÓSITO:** El sistema realizará la modificación de datos de los alumnos. | | | |
| **PRECONDICIONES:**   1. El estudiante debe estar registrado. | | | |
| **FLUJO BÁSICO:**   1. La secretaria selecciona el botón “Estudiante”. 2. El sistema muestra la interfaz estudiante con las opciones “Detalles”, “Inhabilitar”, “Buscar” y   “Actualizar”.   1. La secretaria ingresa el DNI del estudiante que desea buscar. 2. La secretaria selecciona el botón “Buscar”. 3. El sistema verifica si el DNI del estudiante existe.   Si no existe el DNI del estudiante se mostrará la tabla vacía.   1. El sistema muestra en la tabla al estudiante buscado. 2. La secretaria selecciona en la tabla al estudiante. 3. La secretaria selecciona la opción “Inhabilitar”. 4. El sistema inhabilitara al estudiante. 5. La secretaria selecciona el botón “Detalles”. 6. El sistema muestra la interfaz de detalles con la opción: “Editar”, “Confirmar”. 7. La secretaria selecciona la opción “Editar”. 8. La secretaria modifica los datos del estudiante. 9. La secretaria selecciona la opción “Confirmar”. 10. El sistema guardara las modificaciones del estudiante. 11. La secretaria selecciona el botón “Actualizar”. 12. El sistema actualiza la tabla. | | | |
| **POSCONDICION:** Estudiante modificado. | | | |
| **FLUJOS ALTERNATIVOS:** | | | |
| **REQUERIMIENTOS ESPECIALES O SUPLEMENTARIOS:**   1. Los mensajes de advertencia aparecen cuando no se realizó la modificación del estudiante correctamente. 2. Los campos se deben llenar obligatoriamente. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **IDENTIFICADOR:** CU03 | **NOMBRE:** Mantenedor Tutor | | |
| **CATEGORÍA:** Mantenedor | | **COMPLEJIDAD:** Baja | **PRIORIDAD:** Media |
| **ACTORES:** Secretaria | | | |
| **PROPÓSITO:** El sistema realizará la modificación de datos del tutor. | | | |
| **PRECONDICIONES:**   1. Los tutores deben estar registrados. | | | |
| **FLUJO BÁSICO:**   1. La secretaria selecciona el botón “Tutor”. 2. El sistema muestra la interfaz tutor con las opciones “Registrar Tutor”, “Detalles”, “Inhabilitar”, “Buscar” y “Actualizar”. 3. La secretaria selecciona la opción “Registrar tutor”. 4. El sistema muestra la interfaz registrar tutor con las opciones: “Buscar estudiante” y “Confirmar Registro”. 5. La secretaria registra los datos del tutor. 6. La secretaria selecciona la opción “Buscar estudiante”. 7. El sistema muestra la interfaz estudiante con las opciones “Confirmar”. 8. La secretaria selecciona al estudiante. 9. La secretaria selecciona el botón “Confirmar”. 10. El sistema copia DNI del estudiante. 11. La secretaria selecciona la opción “Confirmar Registro”. 12. El sistema registra al tutor. 13. La secretaria ingresa el DNI del tutor que desea buscar. 14. La secretaria selecciona el botón “Buscar”. 15. El sistema verifica si el DNI del tutor existe.   Si no existe el DNI del tutor se mostrará la tabla vacía.   1. El sistema muestra en la tabla al tutor buscado. 2. La secretaria selecciona en la tabla al tutor. 3. La secretaria selecciona la opción “Inhabilitar”. 4. El sistema inhabilitara al tutor. 5. La secretaria selecciona el botón “Detalles”. 6. El sistema muestra la interfaz de detalles con la opción: “Editar”, “Confirmar”. 7. La secretaria selecciona la opción “Editar”. 8. La secretaria modifica los datos del tutor. 9. La secretaria selecciona la opción “Confirmar”. 10. El sistema guardara las modificaciones del tutor. 11. La secretaria selecciona el botón “Actualizar”. 12. El sistema actualiza la tabla. | | | |
| **POSCONDICION:** Tutor modificado. | | | |
| **FLUJOS ALTERNATIVOS:** | | | |
| **REQUERIMIENTOS ESPECIALES O SUPLEMENTARIOS:**   1. Los mensajes de advertencia aparecen cuando no se realizó la modificación del tutor correctamente. 2. Los campos se deben llenar obligatoriamente. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **IDENTIFICADOR:** CU05 | **NOMBRE:** Mantenedor Docente | | |
| **CATEGORÍA:** Mantenedor | | **COMPLEJIDAD:** Baja | **PRIORIDAD:** Alta |
| **ACTORES:** Secretaria | | | |
| **PROPÓSITO:** El sistema realizará el registro de docente, modificara datos de docente e inhabilitara los datos del docente. | | | |
| **PRECONDICIONES:**   1. Los tutores deben tener como mínimo sus datos primordiales. | | | |
| **FLUJO BÁSICO:**   1. La secretaria selecciona el botón “Docente”. 2. El sistema muestra la interfaz docente con las opciones “Registrar Docente”, “Modificar”, “Inhabilitar”, “Buscar” y “Actualizar”. 3. La secretaria selecciona el botón “Registrar Docente”. 4. El sistema muestra la interfaz de registrar docente con la opción Guardar. 5. La secretaria Ingresa los datos del docente. 6. La secretaria selecciona el botón “Guardar”. 7. El sistema registra los datos del docente. 8. La secretaria ingresa el DNI del docente que desea buscar. 9. La secretaria selecciona el botón “Buscar”. 10. El sistema verifica si el DNI del docente existe.   Si no existe el DNI del docente se mostrará la tabla vacía.   1. El sistema muestra en la tabla al docente buscado. 2. La secretaria selecciona en la tabla al docente. 3. La secretaria selecciona el botón “Inhabilitar”. 4. El sistema inhabilita al docente. 5. La secretaria selecciona el botón “Modificar”. 6. El sistema muestra la interfaz Docente con las opciones “Editar” y “Confirmar”. 7. El sistema copia los datos del docente en las cajas de texto de la interfaz Docente. 8. La secretaria selecciona el botón “Editar”. 9. La secretaria modifica los datos del docente. 10. La secretaria selecciona el botón “Confirmar”. 11. El sistema guarda y actualiza los datos. 12. La secretaria selecciona el botón “Actualizar”. 13. El sistema actualiza la tabla del Docente. | | | |
| **POSCONDICION:** Docente registrado, modificado e inhabilitado. | | | |
| **FLUJOS ALTERNATIVOS:** | | | |
| **REQUERIMIENTOS ESPECIALES O SUPLEMENTARIOS:**   1. Los mensajes de advertencia aparecen cuando no se realizó el registro de docente correctamente. 2. Los campos se deben llenar obligatoriamente. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **IDENTIFICADOR:** CU06 | **NOMBRE:** Mantenedor Curso | | |
| **CATEGORÍA:** Mantenedor | | **COMPLEJIDAD:** Baja | **PRIORIDAD:** Alta |
| **ACTORES:** Secretaria | | | |
| **PROPÓSITO:** El sistema realizará el registro del curso, modificará los datos del curso e inhabilita el curso. | | | |
| **PRECONDICIONES:**   1. Los cursos deben tener como mínimo sus datos primordiales. | | | |
| **FLUJO BÁSICO:**   1. La secretaria selecciona el botón “Curso”. 2. El sistema muestra la interfaz docente con las opciones “Registrar Curso”, “Modificar”, “Inhabilitar”, “Buscar” y “Actualizar”. 3. La secretaria selecciona el botón “Registrar Curso”. 4. El sistema muestra la interfaz de registrar curso con la opción Guardar. 5. La secretaria Ingresa los datos del curso. 6. La secretaria selecciona el botón “Guardar”. 7. El sistema registra los datos del curso. 8. La secretaria ingresa el código del curso que desea buscar. 9. La secretaria selecciona el botón “Buscar”. 10. El sistema verifica si el código del curso existe.   Si no existe el código del curso se mostrará la tabla vacía.   1. El sistema muestra en la tabla al curso buscado. 2. La secretaria selecciona en la tabla el curso. 3. La secretaria selecciona el botón “Inhabilitar”. 4. El sistema inhabilita el curso seleccionado. 5. La secretaria selecciona el botón “Modificar”. 6. El sistema muestra la interfaz curso con las opciones “Editar” y “Confirmar”. 7. El sistema copia los datos del curso en las cajas de texto de la interfaz curso. 8. La secretaria selecciona el botón “Editar”. 9. La secretaria modifica los datos del curso. 10. La secretaria selecciona el botón “Confirmar”. 11. El sistema guarda y actualiza los datos. 12. La secretaria selecciona el botón “Actualizar”. 13. El sistema actualiza la tabla del Curso. | | | |
| **POSCONDICION:** Curso registrado, modificado e inhabilitado. | | | |
| **FLUJOS ALTERNATIVOS:** | | | |
| **REQUERIMIENTOS ESPECIALES O SUPLEMENTARIOS:**   1. Los mensajes de advertencia aparecen cuando no se realizó el registro de curso correctamente. 2. Los campos se deben llenar obligatoriamente. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **IDENTIFICADOR:** CU08 | **NOMBRE:** Mantenedor Evaluación | | |
| **CATEGORÍA:** Mantenedor | | **COMPLEJIDAD:** Alta | **PRIORIDAD:** Media |
| **ACTORES:** Docente | | | |
| **PROPÓSITO:** El sistema realizará el registro de la evaluación. | | | |
| **PRECONDICIONES:**   1. La evaluación debe tener todos sus datos primordiales. 2. El curso debe estar registrado. | | | |
| **FLUJO BÁSICO:**   1. La secretaria selecciona el botón “Evaluación”. 2. El sistema muestra la interfaz Evaluación con las opciones “Registrar Evaluación”. 3. La secretaria ingresa el código o nombre de curso. 4. El sistema verifica si el código o nombre del curso existe.   Si no existe el código o nombre del curso se va a “Registrar Curso”.   1. La secretaria Ingresa los datos de la evaluación. 2. La secretaria selecciona el botón “Registrar Evaluación”. 3. El sistema guarda los datos de la evaluación en la base de datos. | | | |
| **POSCONDICION:** Evaluación registrada. | | | |
| **FLUJOS ALTERNATIVOS:** | | | |
| **REQUERIMIENTOS ESPECIALES O SUPLEMENTARIOS:**   1. Los mensajes de advertencia aparecen cuando no se realizó el registro de evaluación correctamente. 2. Los campos se deben llenar obligatoriamente. | | | |

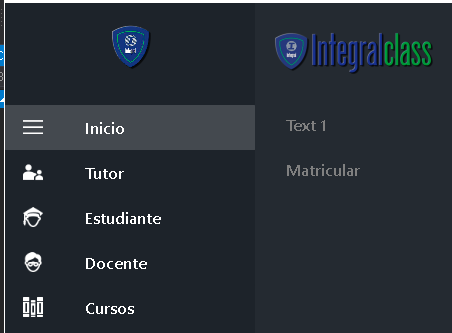
|  |  |  |  |
| --- | --- | --- | --- |
| **IDENTIFICADOR:**  CU07 | **NOMBRE:** Calificación | | |
| **CATEGORIA:** Core | | **COMPLEJIDAD:** Alta | **PRIORIDAD:** Alta |
| **ACTORES:** Docente | | | |
| **PROPOSITO:** Realizar el registro de calificaciones. | | | |
| **PRECONDICIONES:**   1. El estudiante debe estar registrado. 2. La evaluación debe estar registrada. 3. El curso debe estar registrado | | | |
| **FLUJO BÁSICO:**  La secretaria elige la opción **“Calificación”**   1. El sistema muestra la interfaz de calificaciones con la opción: “Confirmar”. 2. El docente ingresa el código o nombre del curso. 3. El sistema verifica el código o nombre del curso.   Si no existe el código o nombre del curso se va “Registrar curso”.   1. El docente ingresa el código de la evaluación. 2. El sistema verifica si el código de la evaluación existe.   Si el código de la evaluación no existe va a “Registrar evaluación”.   1. El sistema mostrara los estudiantes pertenecientes al curso ingresado. 2. El docente seleccionara a los estudiantes. 3. El sistema verifica si el estudiante seleccionado no tiene nota.   Si el estudiante tiene nota se mostrará un mensaje “El estudiante ya tiene una calificación”.   1. El docente ingresa las notas de los estudiantes. 2. El docente selecciona el botón “Confirmar”. 3. El sistema registra las calificaciones en la tabla. | | | |
| **POSCONDICION:** Registro de calificaciones almacenada. | | | |
| **FLUJOS ALTERNATIVOS:**   1. Si el curso no está registrado, va al “Mantenedor curso”. 2. Si la evaluación no está registrada, va al “Mantenedor evaluación”. 3. El sistema verifica si el estudiante tiene calificación. | | | |
| **REQUERIMENTOS ESPECIALES O SUPLEMENTARIOS:**   1. Los mensajes de advertencia aparecen cuando no se efectuó la calificación exitosamente. 2. Los campos se deben llenar obligatoriamente. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **IDENTIFICADOR:**  CU04 | **NOMBRE:** Planificación curricular | | |
| **CATEGORIA:** Core | | **COMPLEJIDAD:** Alta | **PRIORIDAD:** Alta |
| **ACTORES:** Docente | | | |
| **PROPOSITO:** Realizar el registro de la planificación curricular. | | | |
| **PRECONDICIONES:**   1. Los cursos deben estar registrados. | | | |
| **FLUJO BÁSICO:**  El docente elige la opción **“Planificación”**   1. El sistema muestra la interfaz de planificación con la opción: “Guardar”. 2. El docente ingresa el código o nombre del curso. 3. El sistema verifica el código o nombre del curso.   Si no existe el código o nombre del curso va a “Registrar curso”.   1. El docente ingresa la fecha de calendarización. 2. El docente ingresa los datos de la planificación en las cajas de texto. 3. El docente selecciona el botón “Guardar”. 4. El sistema verifica la fecha actual.   Si la fecha no es la actual el registro de la planificación se cancela.   1. El sistema registra la planificación en la tabla. | | | |
| **POSCONDICION:** Registro de planificación almacenada. | | | |
| **FLUJOS ALTERNATIVOS:**   1. Si el curso no está registrado, va al “Mantenedor Curso”. 2. El sistema verifica si la fecha es la actual. | | | |
| **REQUERIMENTOS ESPECIALES O SUPLEMENTARIOS:**   1. Los mensajes de advertencia aparecen cuando no se efectuó el registro de la planificación exitosamente. 2. Los campos se deben llenar obligatoriamente. | | | |

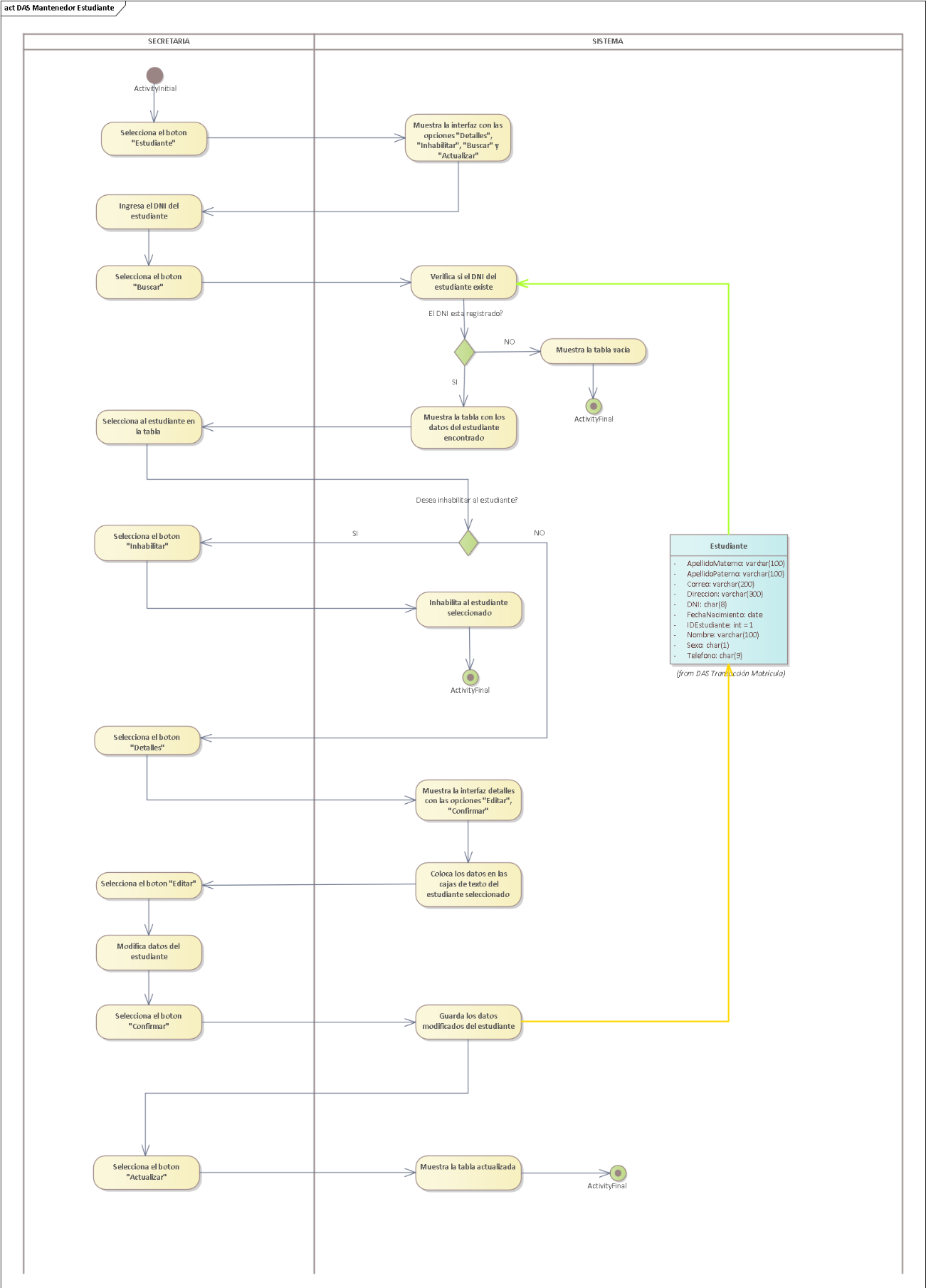
* 1. **Realización del análisis y diseño**
     1. **Para cada iteración**

**Interfaces Principales**

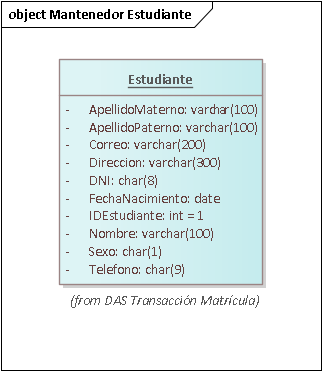
**Interfaz Principal**



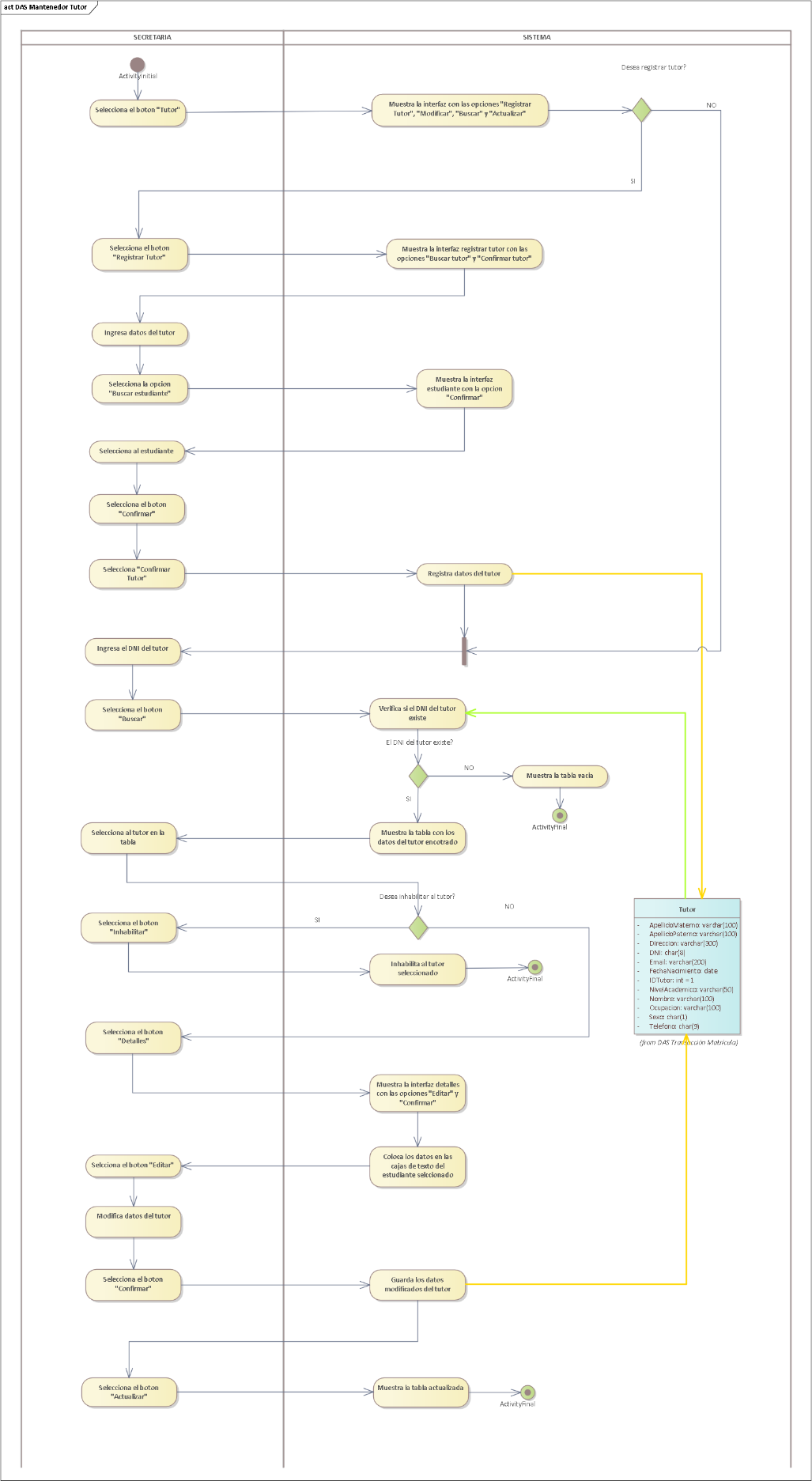
**Diagrama de Actividad del Software**

**DAS Mantenedor Estudiante**

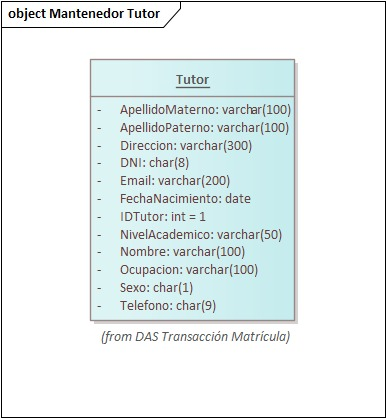
**Modelo Lógico Mantenedor Alumno**



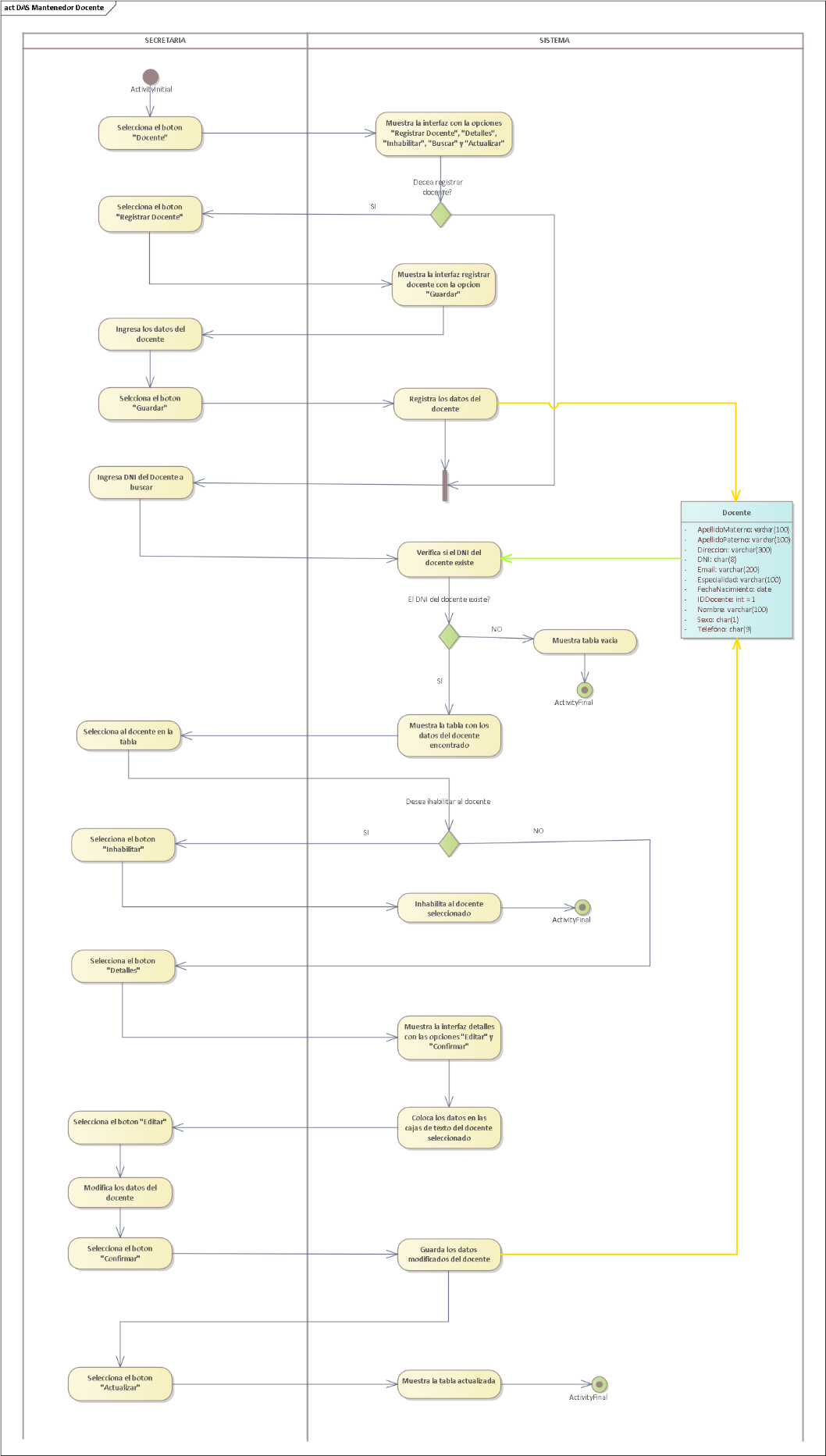
**DAS Mantenedor Tutor**



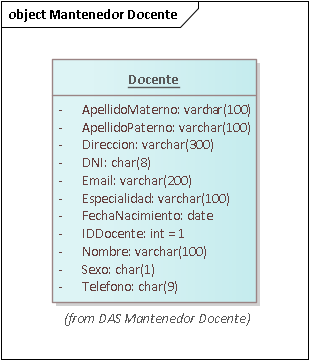
**Modelo lógico Mantenedor Tutor**



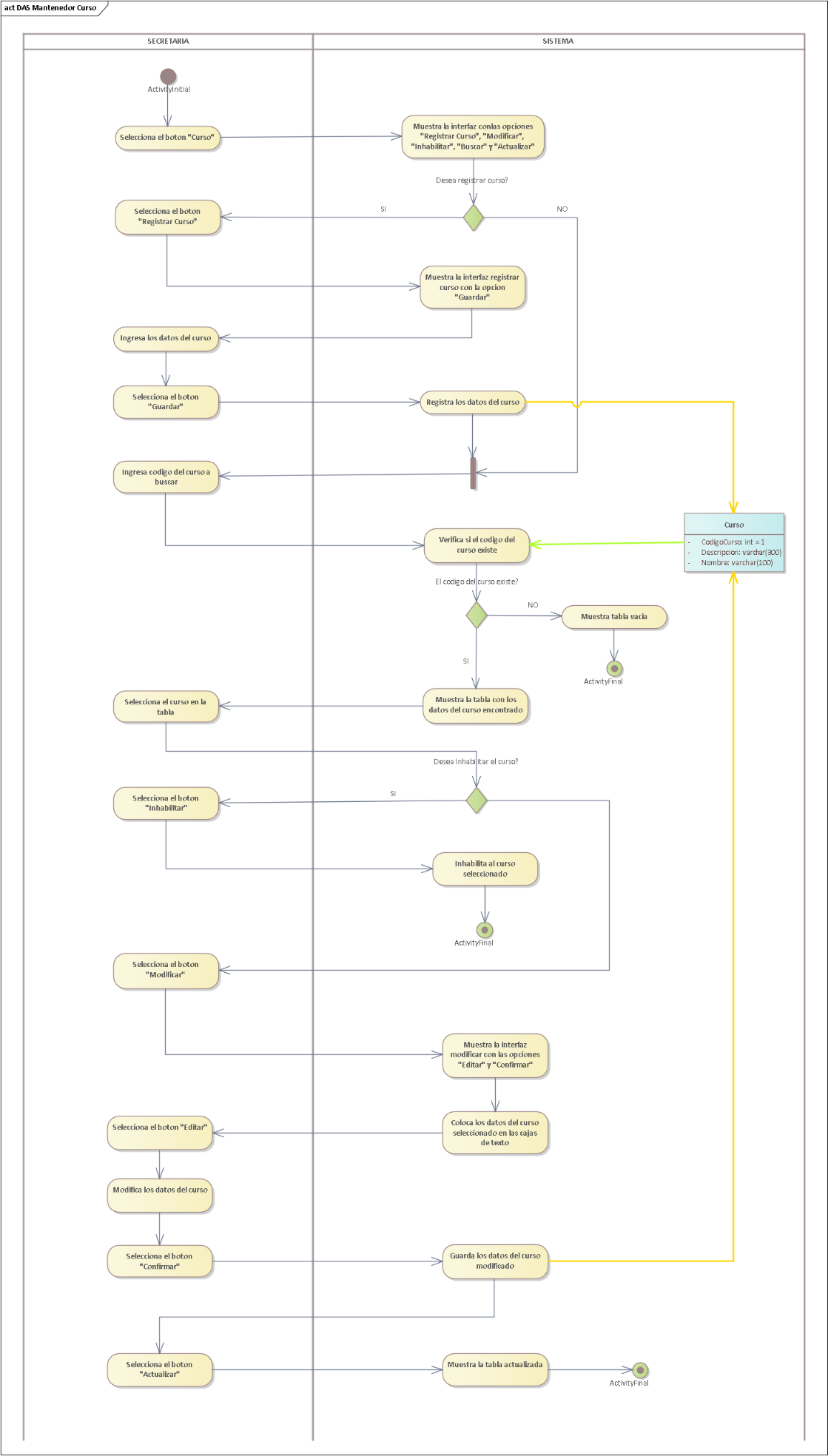
**DAS Mantenedor Docente**



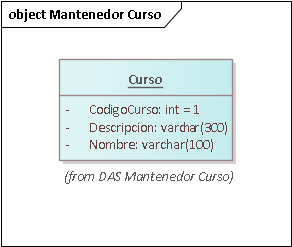
**Modelo lógico Mantenedor Docente**



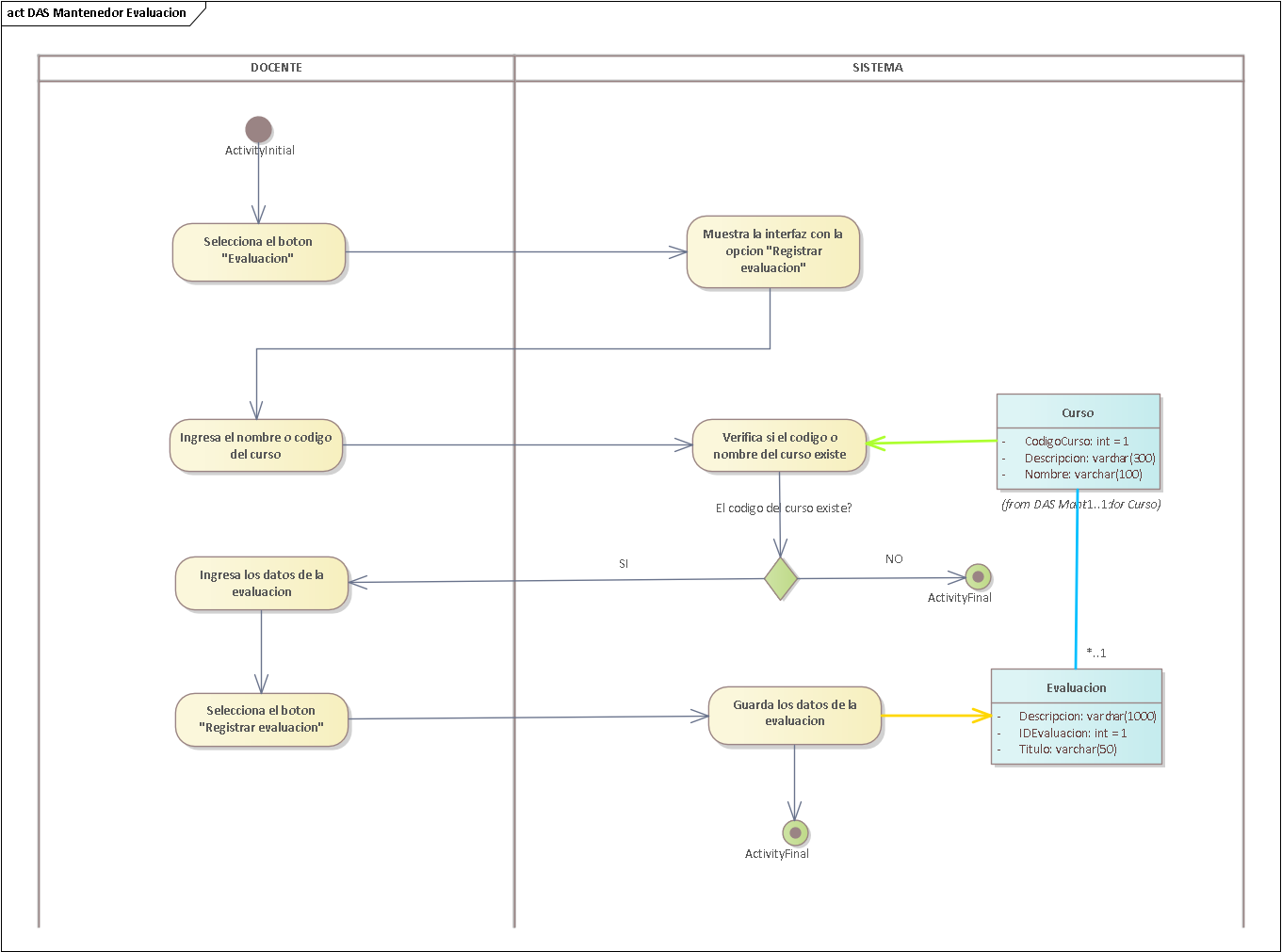
**DAS MANTENEDOR CURSO**



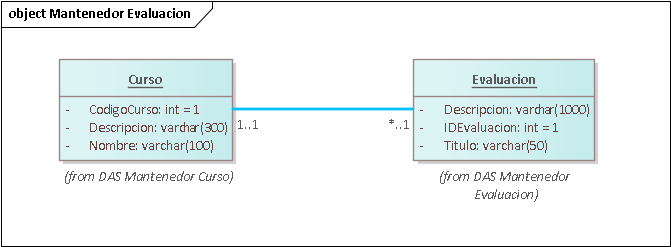
**Modelo lógico Mantenedor Curso**

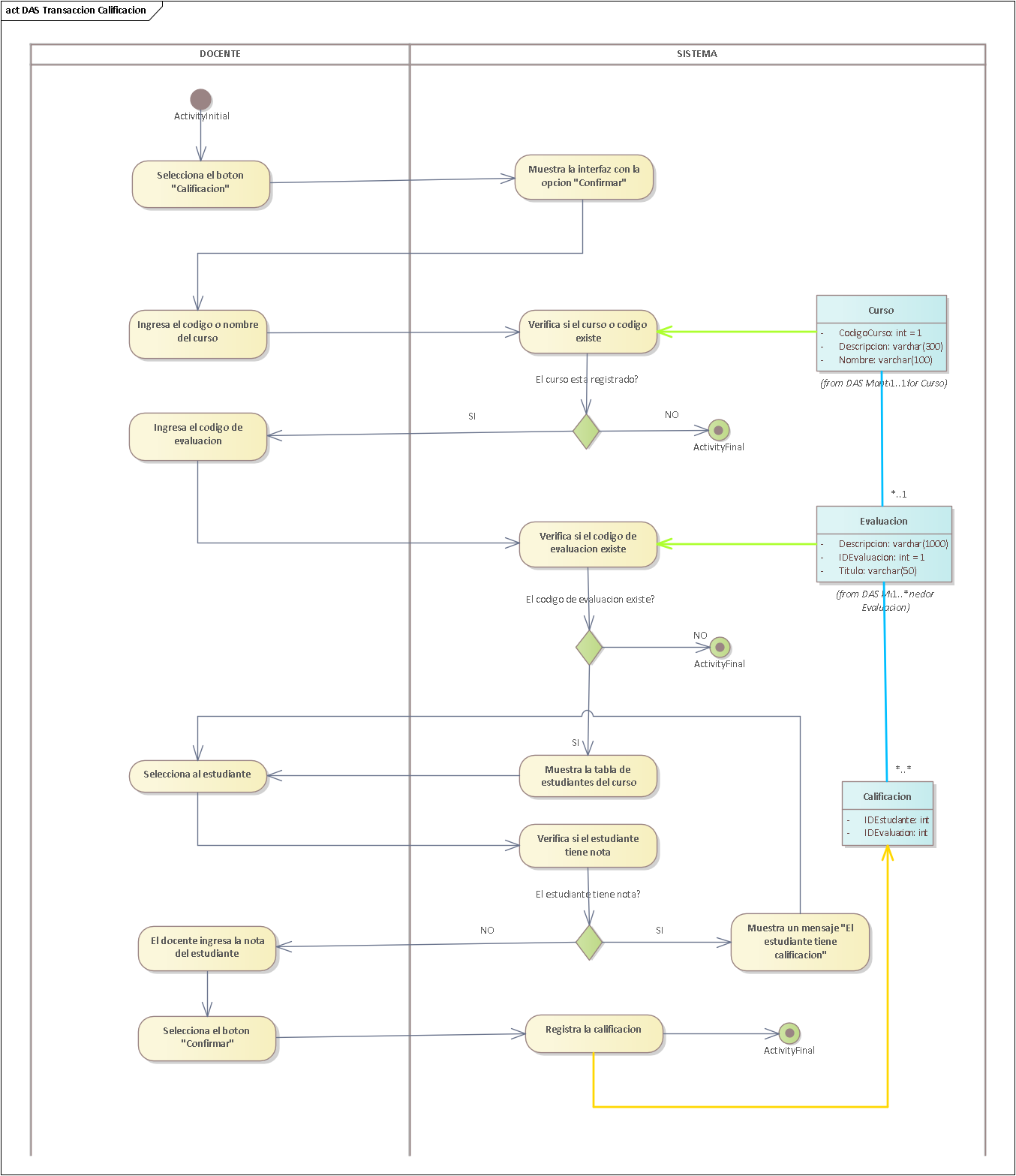


**DAS MANTENEDOR EVALUACIÓN**

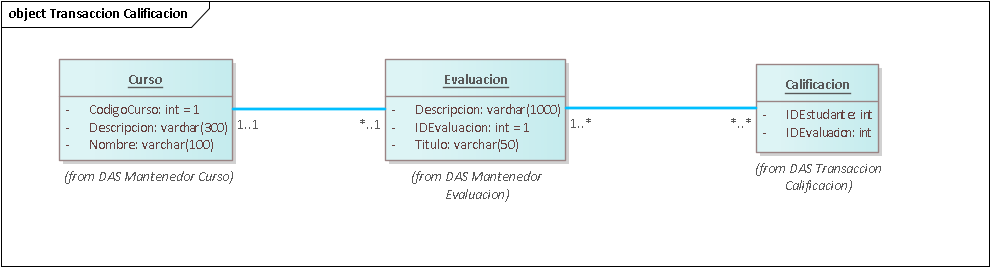


**Modelo Lógico Mantenedor Evaluación**

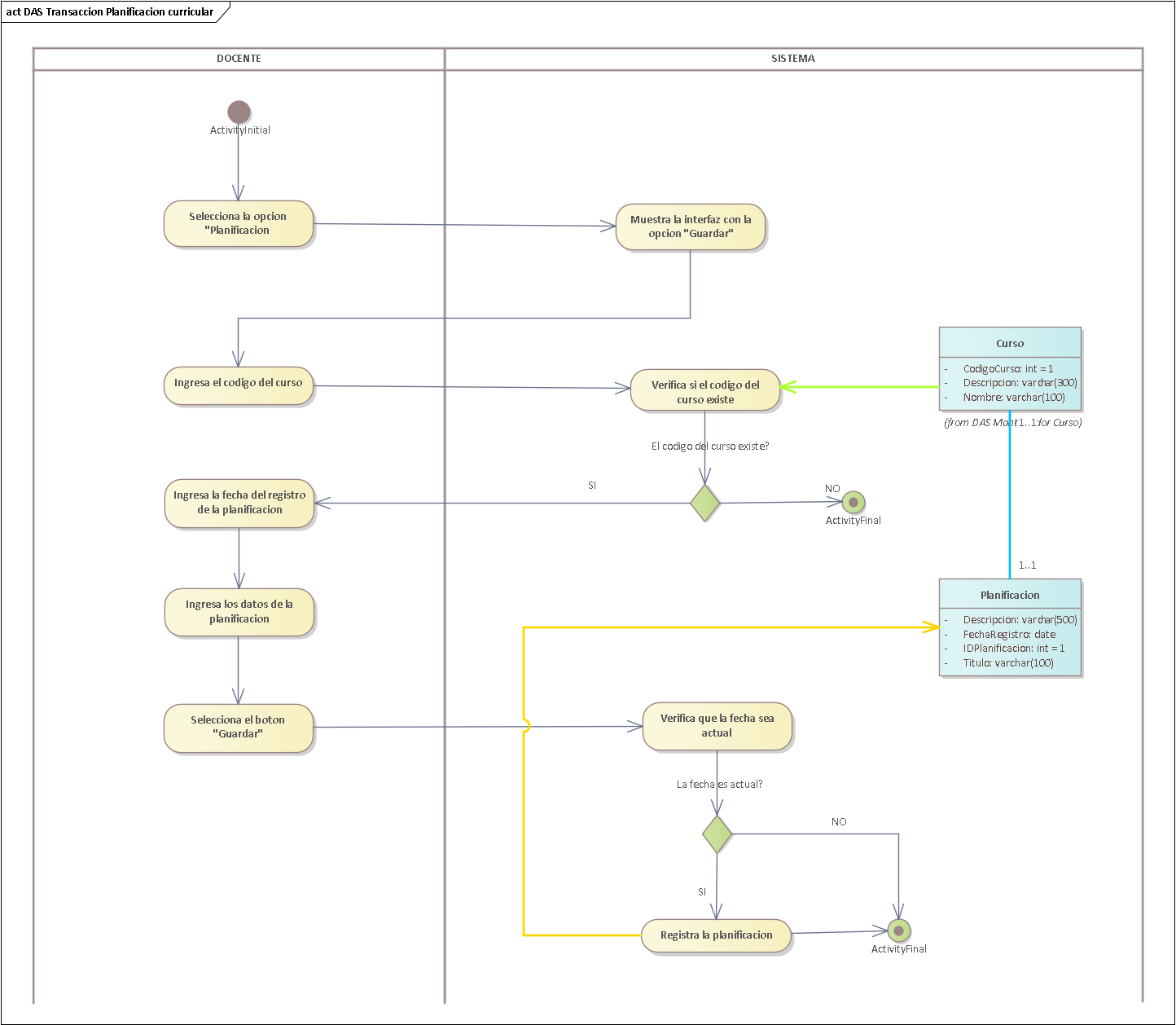


**DAS Transacción Realizar Calificación**

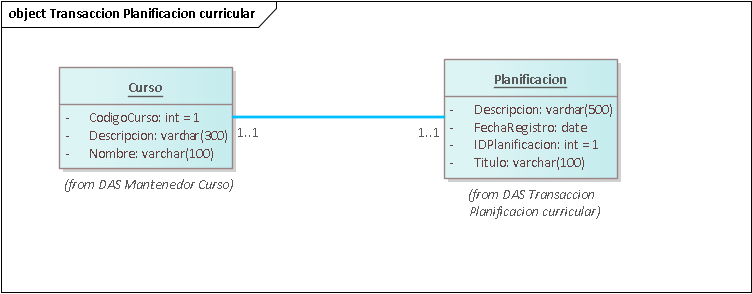
**Modelo lógico Transacción Calificación**



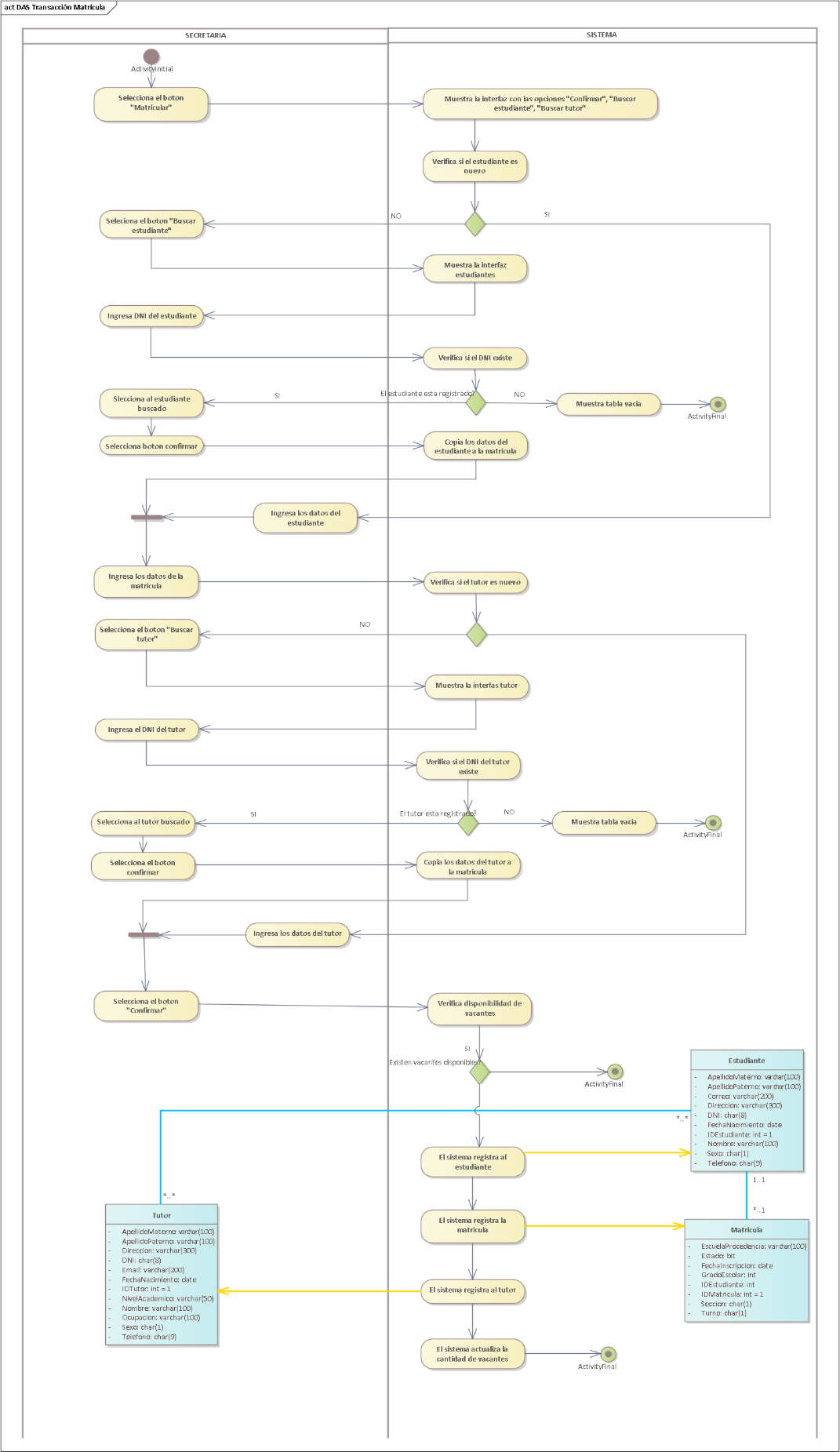
**DAS Transacción Planificación Curricular**



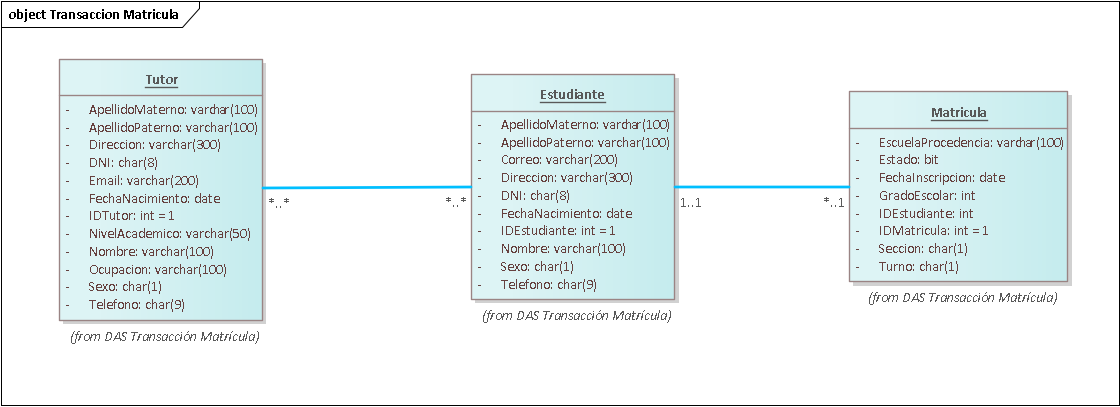
**Modelo lógico Transacción Planificación Curricular**



**DAS Transacción Matrícula**

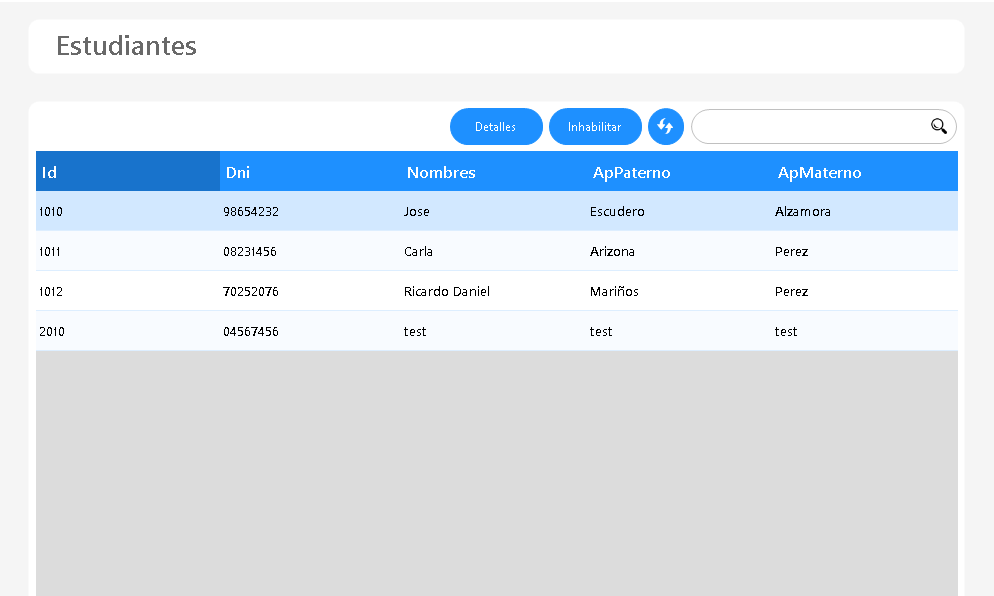


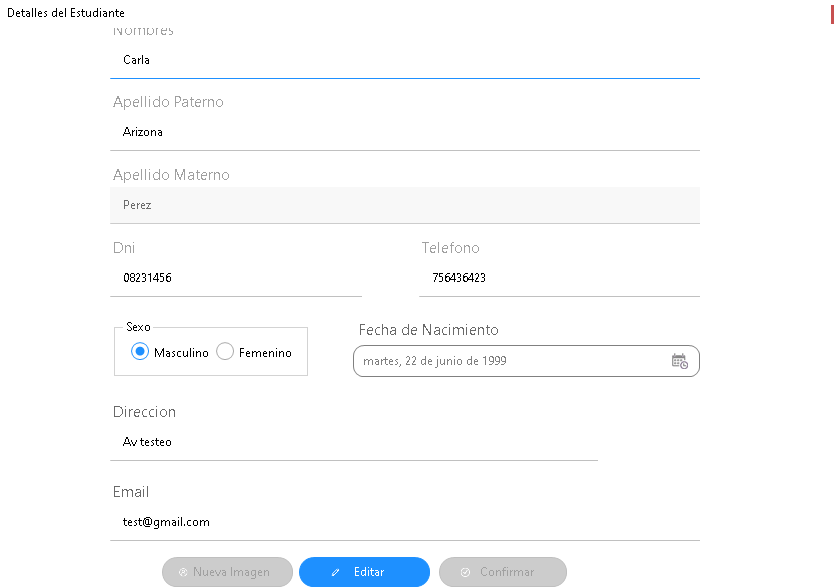
**Modelo Lógico Transacción Matricula**

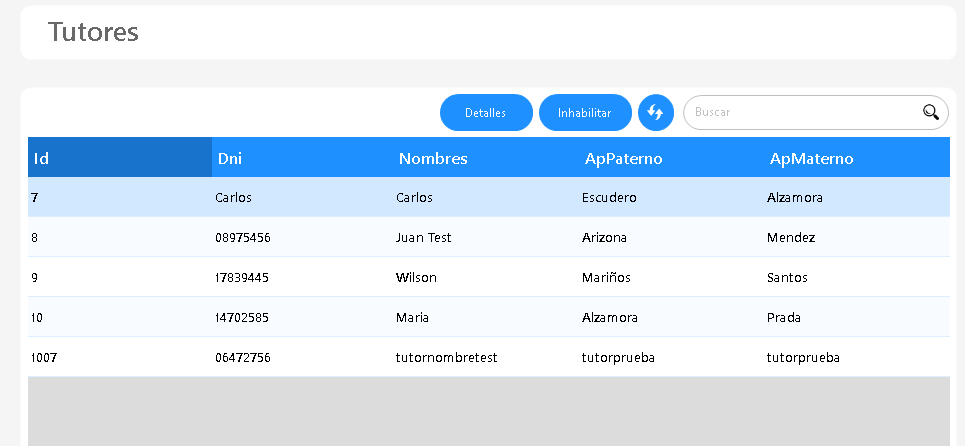


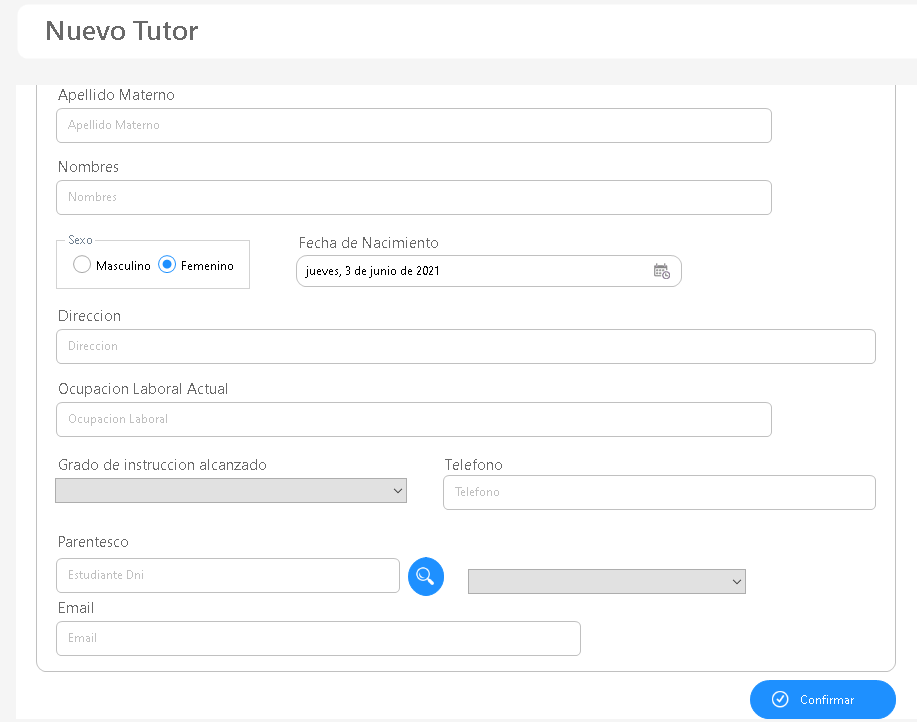
**Interfaces**

**Mantenedor Estudiante**

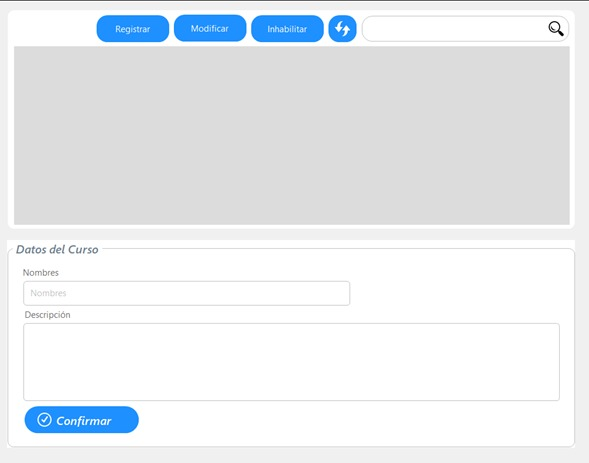




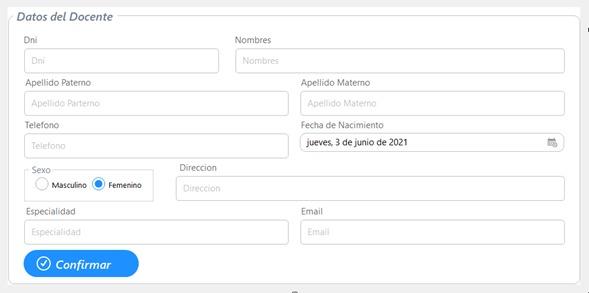
**Mantenedor Tutor**

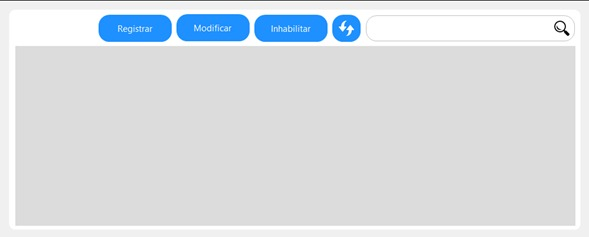


**MANTENEDOR CURSO**

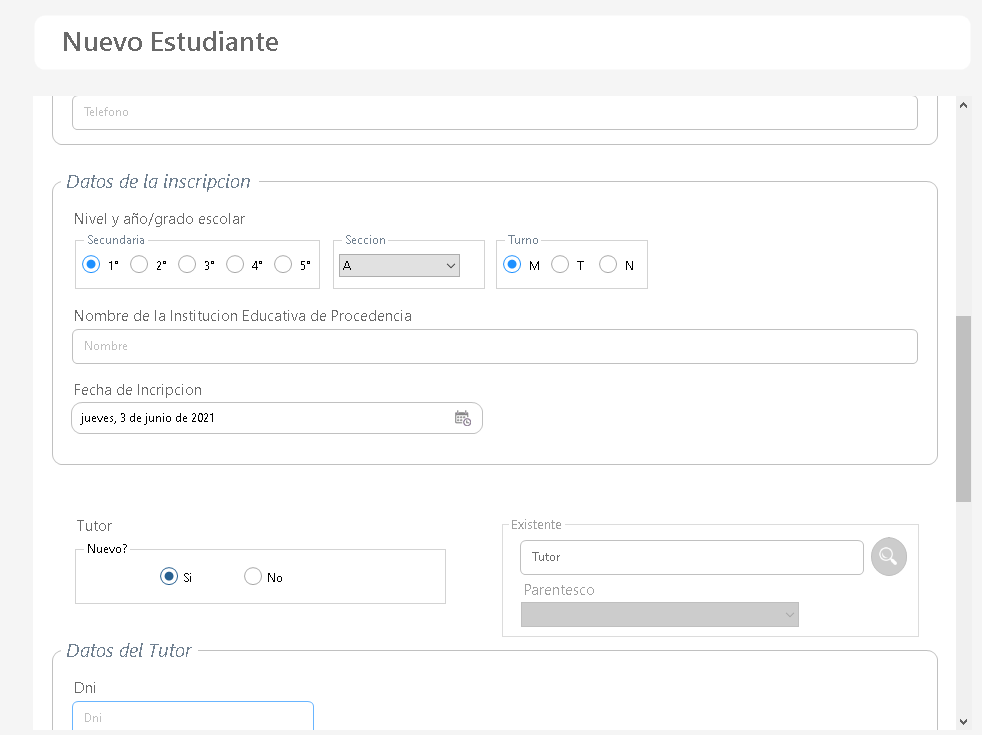


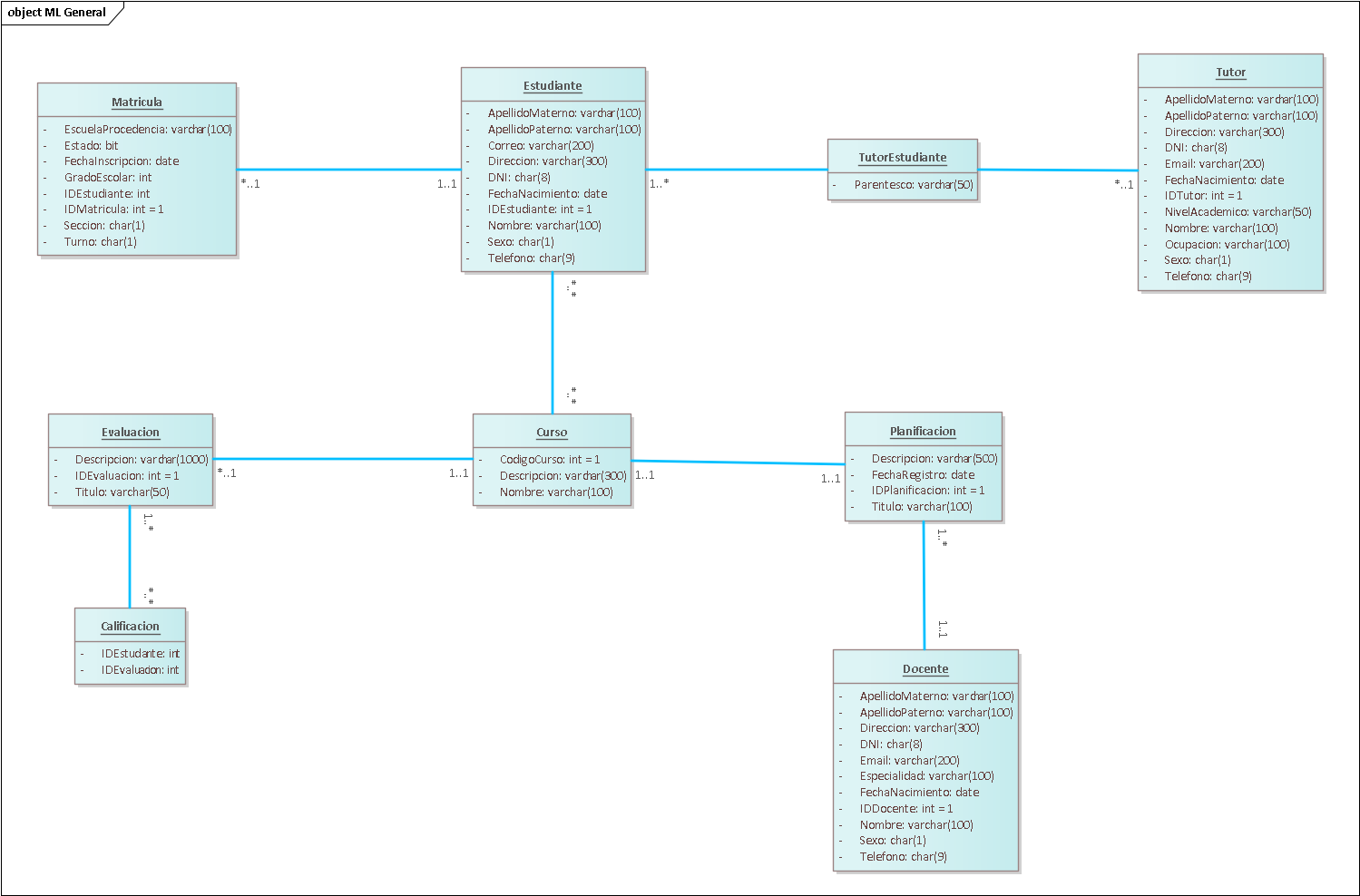
**MANTENEDOR DOCENTE**

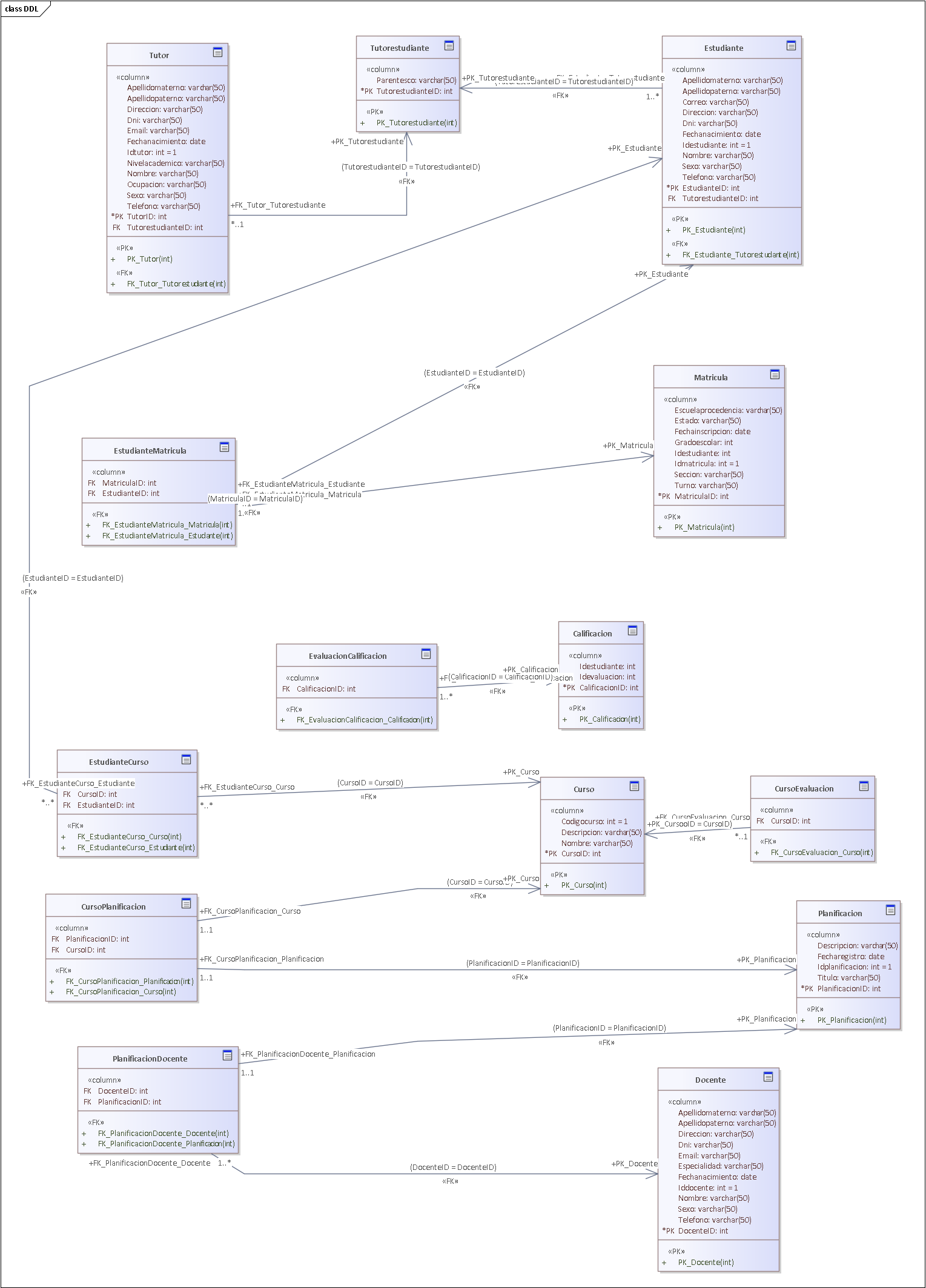




**Transacción Realizar Matricula**

****

Diagrama lógico y físico completo



Ingeniería directa (Script de migración de base de datos en SQL a SERVER)

/\* ---------------------------------------------------- \*/

/\* Generated by Enterprise Architect Version 15.2 \*/

/\* Created On : 30-Jun.-2021 17:04:54 \*/

/\* DBMS : SQL Server 2012 \*/

/\* ---------------------------------------------------- \*/

/\* Drop Foreign Key Constraints \*/

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[FK\_CursoEvaluacion\_Curso]') AND OBJECTPROPERTY(id, N'IsForeignKey') = 1)

ALTER TABLE [CursoEvaluacion] DROP CONSTRAINT [FK\_CursoEvaluacion\_Curso]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[FK\_CursoPlanificacion\_Planificacion]') AND OBJECTPROPERTY(id, N'IsForeignKey') = 1)

ALTER TABLE [CursoPlanificacion] DROP CONSTRAINT [FK\_CursoPlanificacion\_Planificacion]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[FK\_CursoPlanificacion\_Curso]') AND OBJECTPROPERTY(id, N'IsForeignKey') = 1)

ALTER TABLE [CursoPlanificacion] DROP CONSTRAINT [FK\_CursoPlanificacion\_Curso]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[FK\_Estudiante\_Tutorestudiante]') AND OBJECTPROPERTY(id, N'IsForeignKey') = 1)

ALTER TABLE [Estudiante] DROP CONSTRAINT [FK\_Estudiante\_Tutorestudiante]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[FK\_EstudianteCurso\_Curso]') AND OBJECTPROPERTY(id, N'IsForeignKey') = 1)

ALTER TABLE [EstudianteCurso] DROP CONSTRAINT [FK\_EstudianteCurso\_Curso]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[FK\_EstudianteCurso\_Estudiante]') AND OBJECTPROPERTY(id, N'IsForeignKey') = 1)

ALTER TABLE [EstudianteCurso] DROP CONSTRAINT [FK\_EstudianteCurso\_Estudiante]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[FK\_EstudianteMatricula\_Matricula]') AND OBJECTPROPERTY(id, N'IsForeignKey') = 1)

ALTER TABLE [EstudianteMatricula] DROP CONSTRAINT [FK\_EstudianteMatricula\_Matricula]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[FK\_EstudianteMatricula\_Estudiante]') AND OBJECTPROPERTY(id, N'IsForeignKey') = 1)

ALTER TABLE [EstudianteMatricula] DROP CONSTRAINT [FK\_EstudianteMatricula\_Estudiante]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[FK\_EvaluacionCalificacion\_Calificacion]') AND OBJECTPROPERTY(id, N'IsForeignKey') = 1)

ALTER TABLE [EvaluacionCalificacion] DROP CONSTRAINT [FK\_EvaluacionCalificacion\_Calificacion]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[FK\_PlanificacionDocente\_Docente]') AND OBJECTPROPERTY(id, N'IsForeignKey') = 1)

ALTER TABLE [PlanificacionDocente] DROP CONSTRAINT [FK\_PlanificacionDocente\_Docente]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[FK\_PlanificacionDocente\_Planificacion]') AND OBJECTPROPERTY(id, N'IsForeignKey') = 1)

ALTER TABLE [PlanificacionDocente] DROP CONSTRAINT [FK\_PlanificacionDocente\_Planificacion]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[FK\_Tutor\_Tutorestudiante]') AND OBJECTPROPERTY(id, N'IsForeignKey') = 1)

ALTER TABLE [Tutor] DROP CONSTRAINT [FK\_Tutor\_Tutorestudiante]

GO

/\* Drop Tables \*/

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[Calificacion]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [Calificacion]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[Curso]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [Curso]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[CursoEvaluacion]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [CursoEvaluacion]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[CursoPlanificacion]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [CursoPlanificacion]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[Docente]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [Docente]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[Estudiante]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [Estudiante]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[EstudianteCurso]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [EstudianteCurso]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[EstudianteMatricula]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [EstudianteMatricula]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[EvaluacionCalificacion]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [EvaluacionCalificacion]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[Matricula]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [Matricula]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[Planificacion]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [Planificacion]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[PlanificacionDocente]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [PlanificacionDocente]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[Tutor]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [Tutor]

GO

IF EXISTS (SELECT 1 FROM dbo.sysobjects WHERE id = object\_id(N'[Tutorestudiante]') AND OBJECTPROPERTY(id, N'IsUserTable') = 1)

DROP TABLE [Tutorestudiante]

GO

/\* Create Tables \*/

CREATE TABLE [Calificacion]

(

[Idestudiante] int NULL,

[Idevaluacion] int NULL,

[CalificacionID] int NOT NULL

)

GO

CREATE TABLE [Curso]

(

[Codigocurso] int NULL DEFAULT 1,

[Descripcion] varchar(50) NULL,

[Nombre] varchar(50) NULL,

[CursoID] int NOT NULL

)

GO

CREATE TABLE [CursoEvaluacion]

(

[CursoID] int NULL

)

GO

CREATE TABLE [CursoPlanificacion]

(

[PlanificacionID] int NULL,

[CursoID] int NULL

)

GO

CREATE TABLE [Docente]

(

[Apellidomaterno] varchar(50) NULL,

[Apellidopaterno] varchar(50) NULL,

[Direccion] varchar(50) NULL,

[Dni] varchar(50) NULL,

[Email] varchar(50) NULL,

[Especialidad] varchar(50) NULL,

[Fechanacimiento] date NULL,

[Iddocente] int NULL DEFAULT 1,

[Nombre] varchar(50) NULL,

[Sexo] varchar(50) NULL,

[Telefono] varchar(50) NULL,

[DocenteID] int NOT NULL

)

GO

CREATE TABLE [Estudiante]

(

[Apellidomaterno] varchar(50) NULL,

[Apellidopaterno] varchar(50) NULL,

[Correo] varchar(50) NULL,

[Direccion] varchar(50) NULL,

[Dni] varchar(50) NULL,

[Fechanacimiento] date NULL,

[Idestudiante] int NULL DEFAULT 1,

[Nombre] varchar(50) NULL,

[Sexo] varchar(50) NULL,

[Telefono] varchar(50) NULL,

[EstudianteID] int NOT NULL,

[TutorestudianteID] int NULL

)

GO

CREATE TABLE [EstudianteCurso]

(

[CursoID] int NULL,

[EstudianteID] int NULL

)

GO

CREATE TABLE [EstudianteMatricula]

(

[MatriculaID] int NULL,

[EstudianteID] int NULL

)

GO

CREATE TABLE [EvaluacionCalificacion]

(

[CalificacionID] int NULL

)

GO

CREATE TABLE [Matricula]

(

[Escuelaprocedencia] varchar(50) NULL,

[Estado] varchar(50) NULL,

[Fechainscripcion] date NULL,

[Gradoescolar] int NULL,

[Idestudiante] int NULL,

[Idmatricula] int NULL DEFAULT 1,

[Seccion] varchar(50) NULL,

[Turno] varchar(50) NULL,

[MatriculaID] int NOT NULL

)

GO

CREATE TABLE [Planificacion]

(

[Descripcion] varchar(50) NULL,

[Fecharegistro] date NULL,

[Idplanificacion] int NULL DEFAULT 1,

[Titulo] varchar(50) NULL,

[PlanificacionID] int NOT NULL

)

GO

CREATE TABLE [PlanificacionDocente]

(

[DocenteID] int NULL,

[PlanificacionID] int NULL

)

GO

CREATE TABLE [Tutor]

(

[Apellidomaterno] varchar(50) NULL,

[Apellidopaterno] varchar(50) NULL,

[Direccion] varchar(50) NULL,

[Dni] varchar(50) NULL,

[Email] varchar(50) NULL,

[Fechanacimiento] date NULL,

[Idtutor] int NULL DEFAULT 1,

[Nivelacademico] varchar(50) NULL,

[Nombre] varchar(50) NULL,

[Ocupacion] varchar(50) NULL,

[Sexo] varchar(50) NULL,

[Telefono] varchar(50) NULL,

[TutorID] int NOT NULL,

[TutorestudianteID] int NULL

)

GO

CREATE TABLE [Tutorestudiante]

(

[Parentesco] varchar(50) NULL,

[TutorestudianteID] int NOT NULL

)

GO

/\* Create Primary Keys, Indexes, Uniques, Checks \*/

ALTER TABLE [Calificacion]

ADD CONSTRAINT [PK\_Calificacion]

PRIMARY KEY CLUSTERED ([CalificacionID] ASC)

GO

ALTER TABLE [Curso]

ADD CONSTRAINT [PK\_Curso]

PRIMARY KEY CLUSTERED ([CursoID] ASC)

GO

ALTER TABLE [Docente]

ADD CONSTRAINT [PK\_Docente]

PRIMARY KEY CLUSTERED ([DocenteID] ASC)

GO

ALTER TABLE [Estudiante]

ADD CONSTRAINT [PK\_Estudiante]

PRIMARY KEY CLUSTERED ([EstudianteID] ASC)

GO

ALTER TABLE [Matricula]

ADD CONSTRAINT [PK\_Matricula]

PRIMARY KEY CLUSTERED ([MatriculaID] ASC)

GO

ALTER TABLE [Planificacion]

ADD CONSTRAINT [PK\_Planificacion]

PRIMARY KEY CLUSTERED ([PlanificacionID] ASC)

GO

ALTER TABLE [Tutor]

ADD CONSTRAINT [PK\_Tutor]

PRIMARY KEY CLUSTERED ([TutorID] ASC)

GO

ALTER TABLE [Tutorestudiante]

ADD CONSTRAINT [PK\_Tutorestudiante]

PRIMARY KEY CLUSTERED ([TutorestudianteID] ASC)

GO

/\* Create Foreign Key Constraints \*/

ALTER TABLE [CursoEvaluacion] ADD CONSTRAINT [FK\_CursoEvaluacion\_Curso]

FOREIGN KEY ([CursoID]) REFERENCES [Curso] ([CursoID]) ON DELETE No Action ON UPDATE No Action

GO

ALTER TABLE [CursoPlanificacion] ADD CONSTRAINT [FK\_CursoPlanificacion\_Planificacion]

FOREIGN KEY ([PlanificacionID]) REFERENCES [Planificacion] ([PlanificacionID]) ON DELETE No Action ON UPDATE No Action

GO

ALTER TABLE [CursoPlanificacion] ADD CONSTRAINT [FK\_CursoPlanificacion\_Curso]

FOREIGN KEY ([CursoID]) REFERENCES [Curso] ([CursoID]) ON DELETE No Action ON UPDATE No Action

GO

ALTER TABLE [Estudiante] ADD CONSTRAINT [FK\_Estudiante\_Tutorestudiante]

FOREIGN KEY ([TutorestudianteID]) REFERENCES [Tutorestudiante] ([TutorestudianteID]) ON DELETE No Action ON UPDATE No Action

GO

ALTER TABLE [EstudianteCurso] ADD CONSTRAINT [FK\_EstudianteCurso\_Curso]

FOREIGN KEY ([CursoID]) REFERENCES [Curso] ([CursoID]) ON DELETE No Action ON UPDATE No Action

GO

ALTER TABLE [EstudianteCurso] ADD CONSTRAINT [FK\_EstudianteCurso\_Estudiante]

FOREIGN KEY ([EstudianteID]) REFERENCES [Estudiante] ([EstudianteID]) ON DELETE No Action ON UPDATE No Action

GO

ALTER TABLE [EstudianteMatricula] ADD CONSTRAINT [FK\_EstudianteMatricula\_Matricula]

FOREIGN KEY ([MatriculaID]) REFERENCES [Matricula] ([MatriculaID]) ON DELETE No Action ON UPDATE No Action

GO

ALTER TABLE [EstudianteMatricula] ADD CONSTRAINT [FK\_EstudianteMatricula\_Estudiante]

FOREIGN KEY ([EstudianteID]) REFERENCES [Estudiante] ([EstudianteID]) ON DELETE No Action ON UPDATE No Action

GO

ALTER TABLE [EvaluacionCalificacion] ADD CONSTRAINT [FK\_EvaluacionCalificacion\_Calificacion]

FOREIGN KEY ([CalificacionID]) REFERENCES [Calificacion] ([CalificacionID]) ON DELETE No Action ON UPDATE No Action

GO

ALTER TABLE [PlanificacionDocente] ADD CONSTRAINT [FK\_PlanificacionDocente\_Docente]

FOREIGN KEY ([DocenteID]) REFERENCES [Docente] ([DocenteID]) ON DELETE No Action ON UPDATE No Action

GO

ALTER TABLE [PlanificacionDocente] ADD CONSTRAINT [FK\_PlanificacionDocente\_Planificacion]

FOREIGN KEY ([PlanificacionID]) REFERENCES [Planificacion] ([PlanificacionID]) ON DELETE No Action ON UPDATE No Action

GO

ALTER TABLE [Tutor] ADD CONSTRAINT [FK\_Tutor\_Tutorestudiante]

FOREIGN KEY ([TutorestudianteID]) REFERENCES [Tutorestudiante] ([TutorestudianteID]) ON DELETE No Action ON UPDATE No Action

GO

Diagrama de BD en SQL SERVER

