



Universidad Nacional Autónoma de México

Tecnología Orientada a Objetos

Diseño del software

Versión 1.0

Fecha: 03/Noviembre/2022

Integrantes:

- **Hernández Ruiz Leny Javier**
- **Ochoa Lozano Miguel Angel**
- **Silverio Martínez Andrés**
- **Trujillo Salazar Wendy Jazmin**



Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software

Contenido

Histórico de Cambios	3
Descripción de la Arquitectura	4
Paquetes de la arquitectura	4
Definición del ambiente de implementación	6
Vista de despliegue	6
Descripción	6
Modelo de despliegue	6
Vista lógica	6
Descripción	6
Identificación de las clases	6
Vista de datos	8
Descripción	8
Diagrama de la base de datos	8
Vista dinámica	9
Descripción	9
Diagramas de secuencia	9
Diagramas de navegación	16



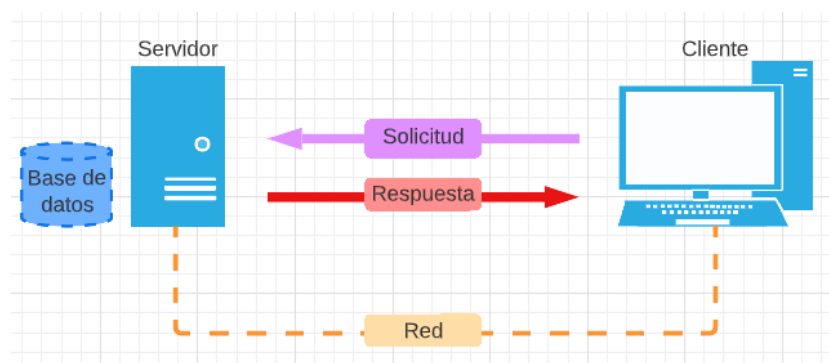
Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software

Histórico de Cambios

Versión	Descripción	Responsable de Actualización	Fecha de actualización
v0.1	Creación del documento y relleno de requerimientos (Entregables 1, 2 y 3)	Trujillo Salazar Wendy Jazmin	26/10/2022
v0.2	Modificación del documento y requisitos funcionales (Entregables 1, 2 y 3)	Hernández Ruiz Leny Javier Ochoa Lozano Miguel Ángel Silverio Martínez Andrés Trujillo Salazar Wendy Jazmin	27/10/2022
v1.0	Actualización del documento	Trujillo Salazar Wendy Jazmin	01/11/2022

1. Descripción de la Arquitectura

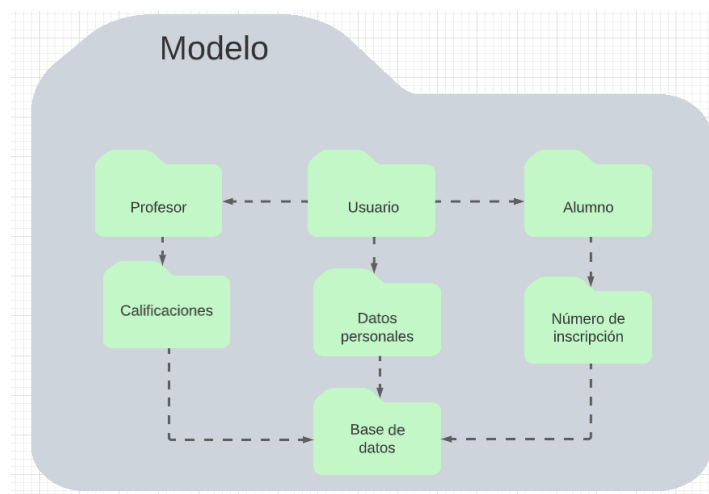
En el sistema se requiere priorizar las características de facilidad de modularidad, soporte de almacenamiento de datos y garantizar la portabilidad para que los usuarios puedan acceder desde cualquier dispositivo en las diferentes plataformas. Tomando en cuenta principalmente estas características, el sistema será implementado utilizando una arquitectura basada en el patrón de arquitectura *cliente-servidor*



1.1. Paquetes de la arquitectura

1.1.1. Modelo (Model)

Es la representación de la información con la cual el sistema va a operar, por lo tanto gestiona todos los accesos a dicha información, tanto consultas como actualizaciones de datos. Mostrando al usuario aquella parte de la información que se le solicita para que esta sea mostrada. Las peticiones de acceso o manipulación de información llegan al 'modelo' a través del 'controlador'.

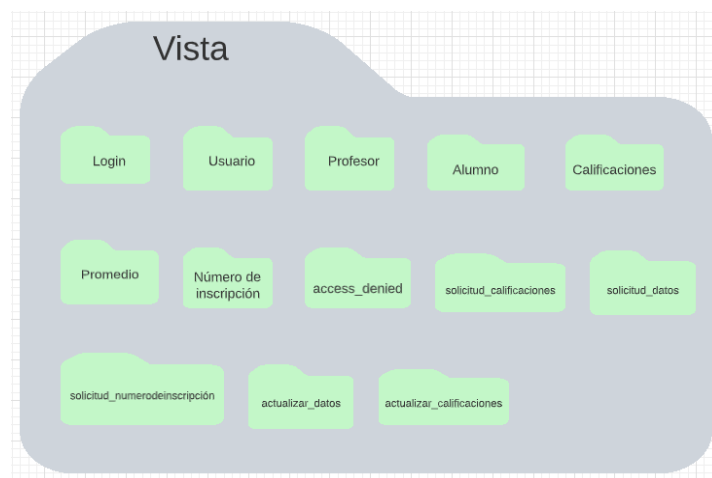




Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software

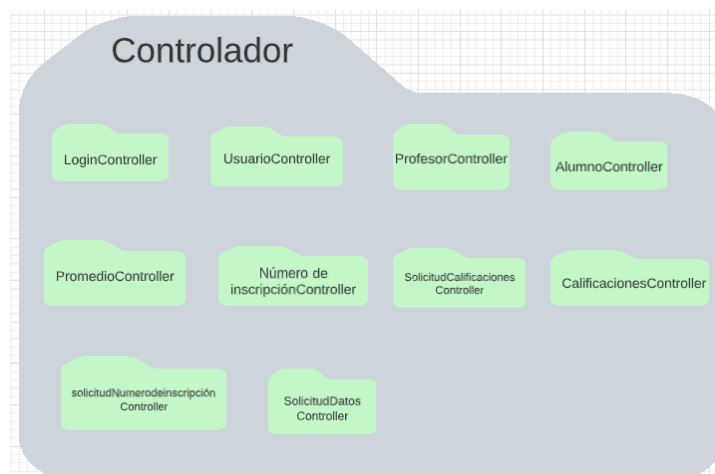
1.1.2. Vista (View)

Presenta el 'modelo' (gestión de datos) en un formato adecuado para interactuar (usualmente la interfaz de usuario), por tanto requiere de dicho 'modelo' la información que debe representar como salida.



1.1.3. Controlador (Controller)

Responde a eventos (usualmente acciones del usuario) e invoca peticiones al 'modelo' cuando se hace alguna solicitud sobre la información (por ejemplo, un registro en la base de datos). También puede enviar comandos a su 'vista' asociada si se solicita un cambio en la forma en que se presenta el 'modelo', por tanto se podría decir que el 'controlador' hace de intermediario entre la 'vista' y el 'modelo'.





Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software

1.2. Definición del ambiente de implementación

Concepto	Herramienta	Versión
Lenguaje de programación	Java, Javascript	8
Framework	Spring	5.0
Diagramador UML	Lucidchart	0.97.2
Manejador de bases de datos	MySQL	8.0.15
Servidor de aplicaciones	Glassfish	4.1.2
IDE	Netbeans	14.0
Control de versiones	Github	2.21.0

2. Vista de despliegue

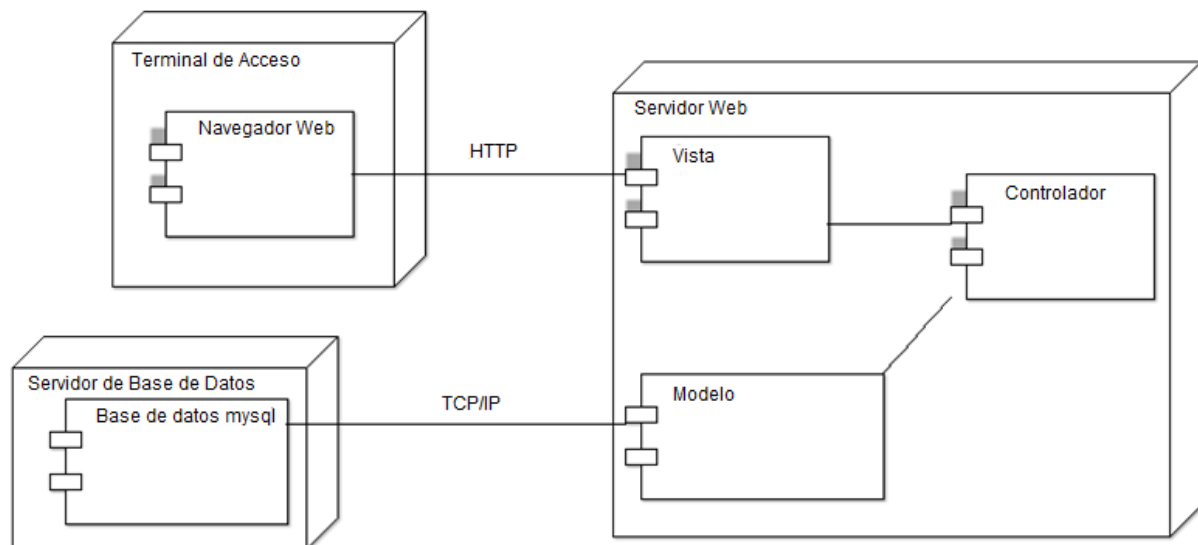
2.1. Descripción

El sistema tendrá 3 nodos principales:

- Usuario: navegador web independiente de la plataforma (windows, linux, android, ios, etc.), este navegador se conectará mediante el protocolo HTTP con el servidor web.
- Servidor web: servidor que correrá en glassfish y utilizará un proveedor de servicios en la nube, ya que es más rentable que adquirir el equipo técnico necesario para montar un servidor propio.
- Servidor de base de datos: utilizará la plataforma mysql (libre y gratuita) debido al tipo y volumen de información que se almacenará. Éste realiza la conexión con el servidor web mediante el protocolo TCP/IP.



2.2. Modelo de despliegue



3. Vista lógica

3.1. Descripción

Esta sección describe de manera general la descomposición del modelo mediante jerarquía de paquetes.

3.2. Identificación de las clases

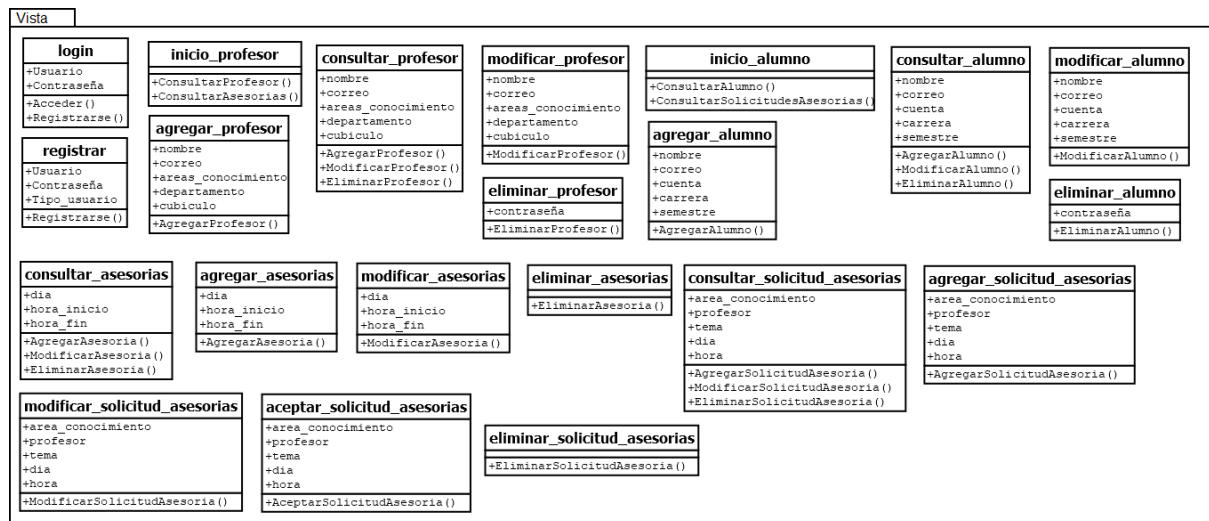
Esta sección contendrá los diagramas que describen el comportamiento estático del sistema y los tipos de relaciones existentes entre las clases. Se generará un diagrama de clases por cada paquete que contenga el sistema.

3.2.1. Clases de vista

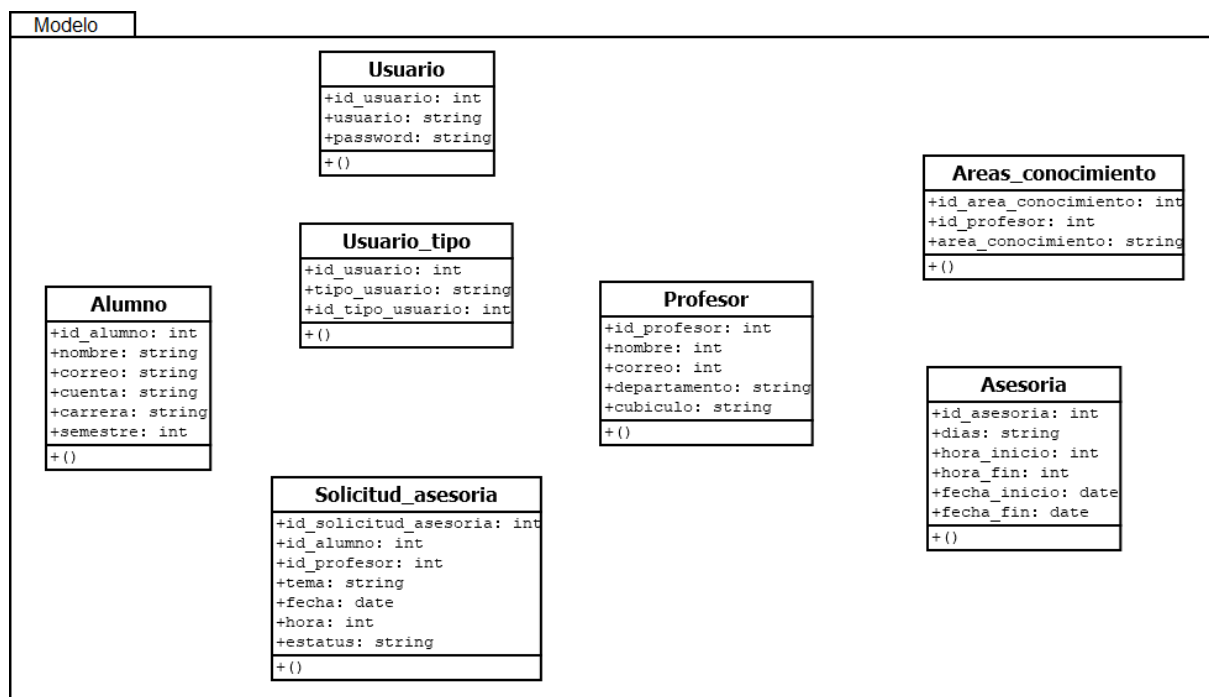


Ingeniería de la Computación, Universidad Nacional Autónoma de México

Programación Orientada a Objetos - Diseño de software



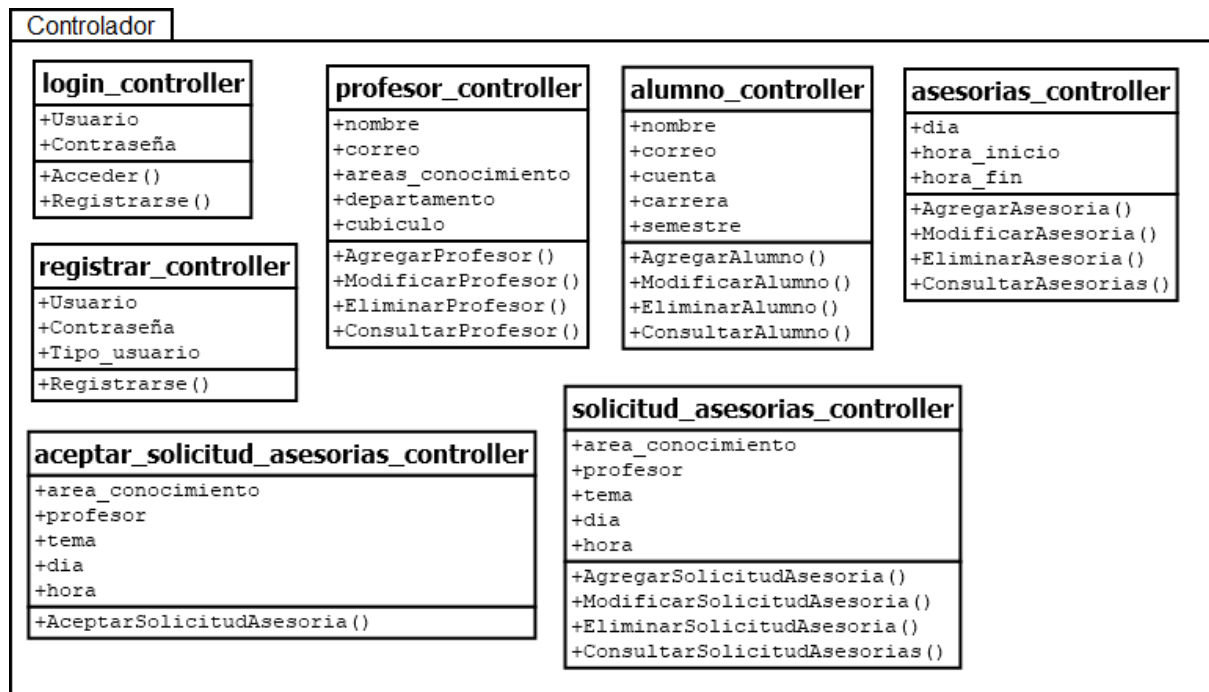
3.2.2. Clases de modelo



3.2.3. Clases de controlador



Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software



4. Vista de datos

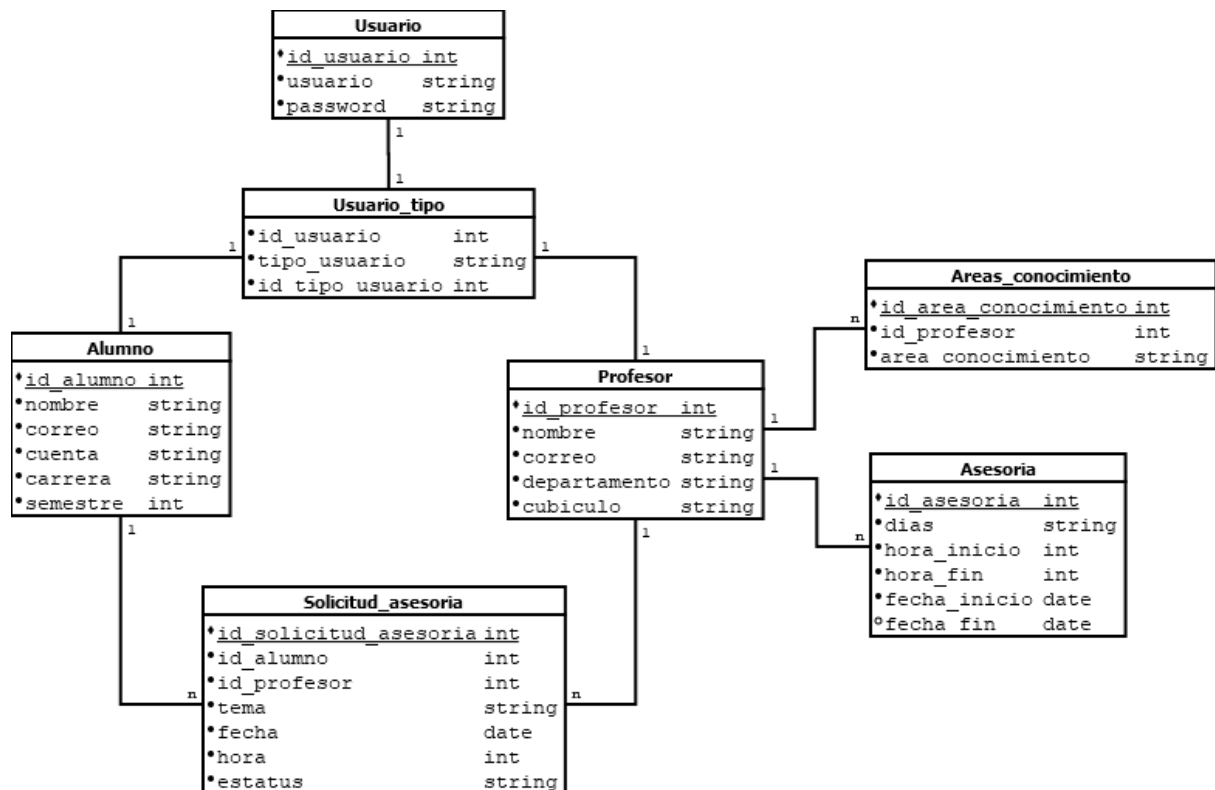
4.1. Descripción

Para hacer persistente la información del sistema de administración de asesorías se utilizará una base de datos que es un conjunto de datos pertenecientes a un mismo contexto y almacenados sistemáticamente para su posterior uso.

4.2. Diagrama de la base de datos



Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software



5. Vista dinámica

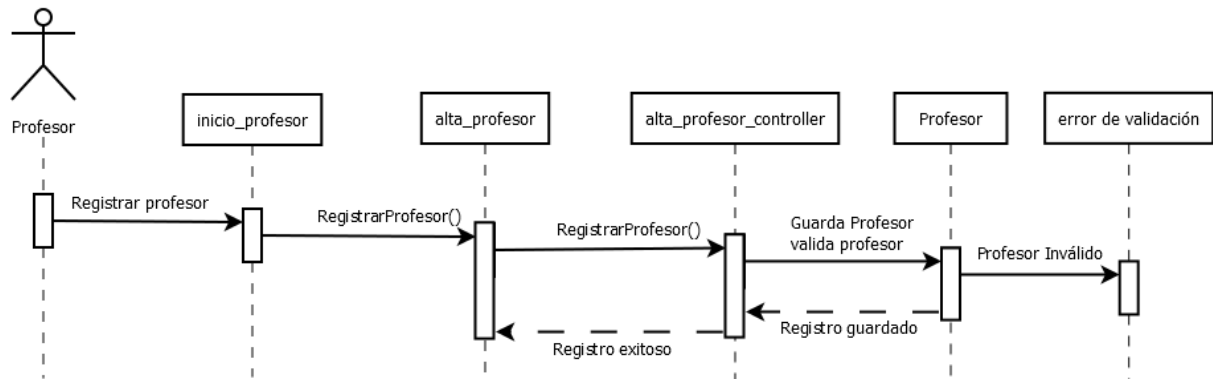
5.1. Descripción

Cada caso de uso es representado con un diagrama de secuencia, éste muestra cómo se comportan los objetos entre ellos a través del tiempo. Los diagramas de secuencia representarán el comportamiento dinámico de los casos de uso. El diagrama de estados, indicará cómo navega el usuario a través de las interfaces del sistema. Cada diagrama tendrá un estado inicial que se identificará con una circunferencia negra que rodea un punto negro, los estados serán representados por un óvalo con el nombre de la interfaz en la que se encuentra el usuario al realizar alguna acción, las acciones estarán determinadas en el documento por flechas con un estado origen, una acción y el estado destino.

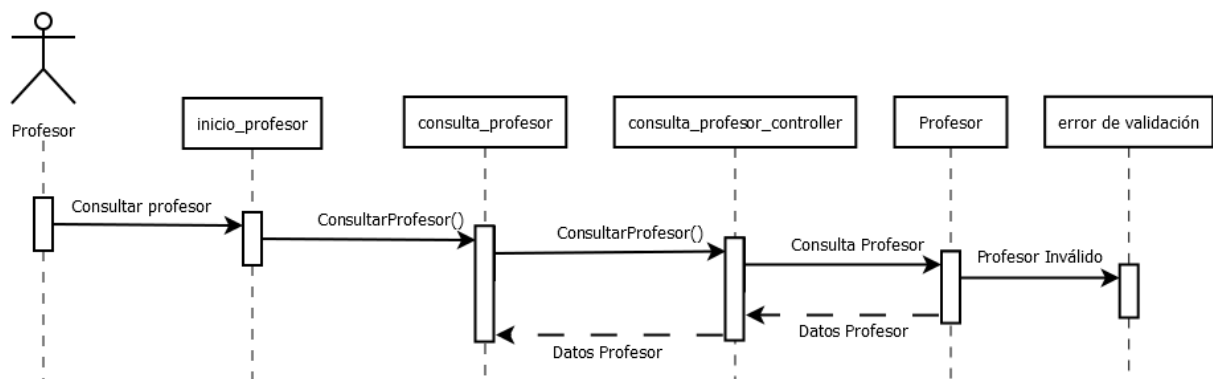
5.2. Diagramas de secuencia

5.2.1. Agregar Profesor

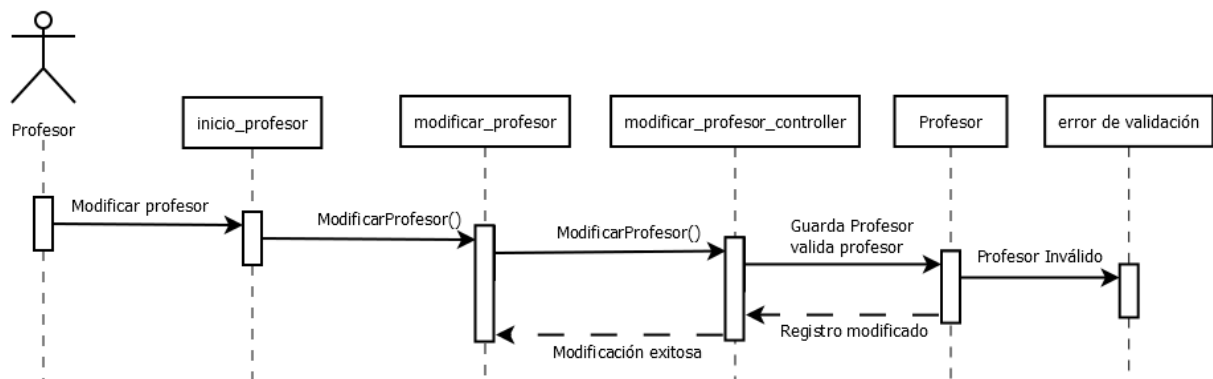
Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software



5.2.2. Consultar Profesor

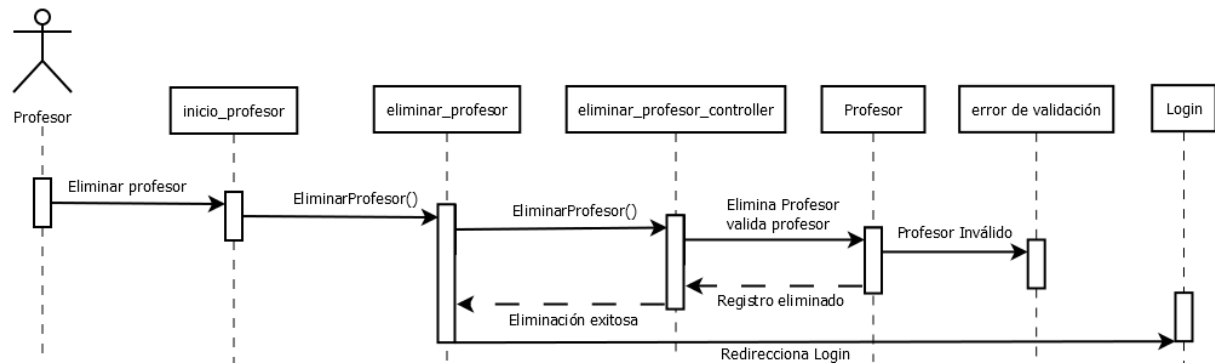


5.2.3. Modificar Profesor

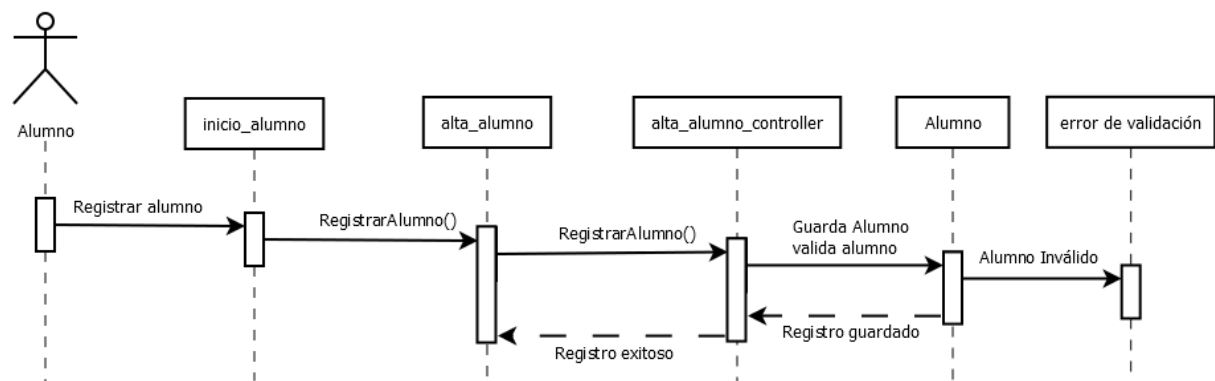


5.2.4. Eliminar Profesor

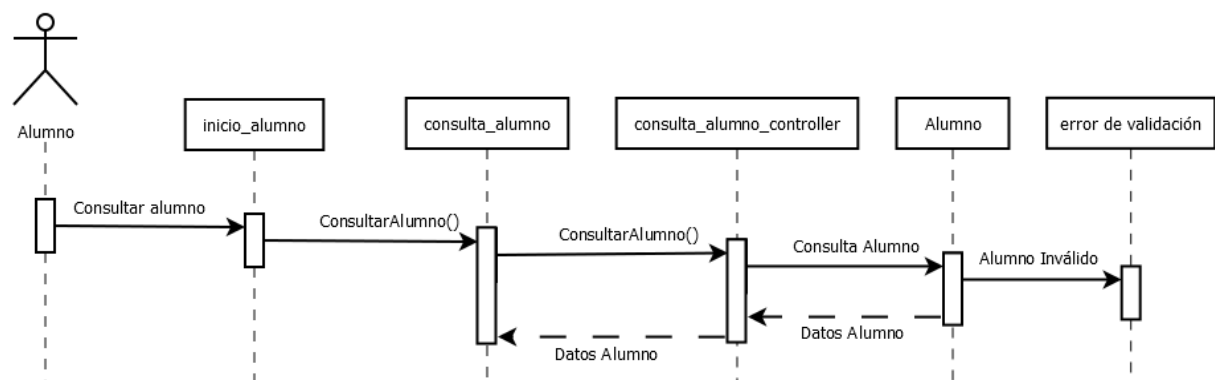
Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software



5.2.5. Agregar Alumno

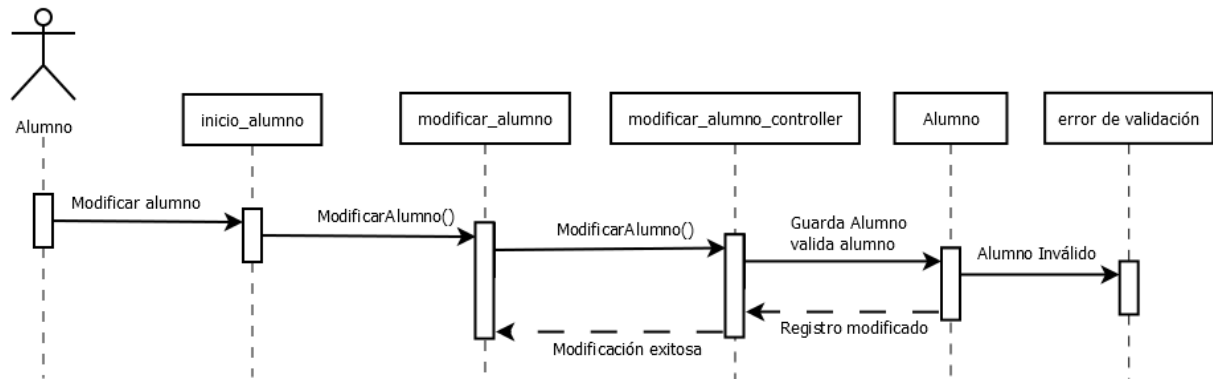


5.2.6. Consultar

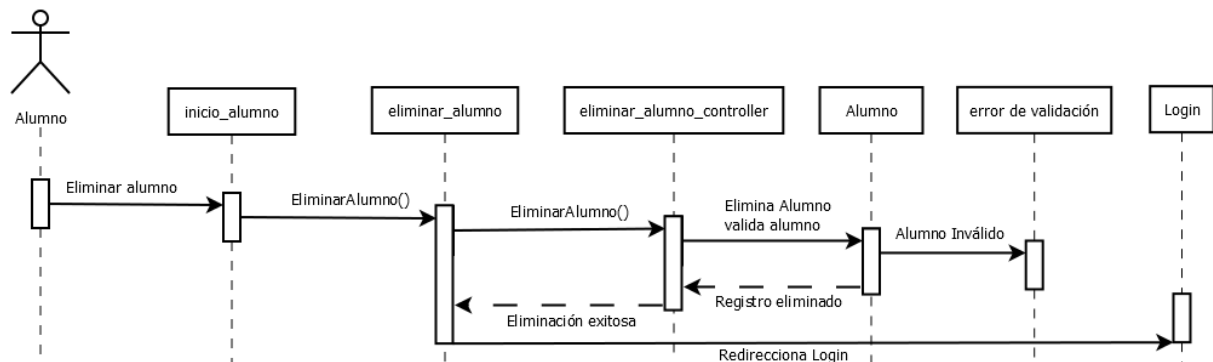


5.2.7. Modificar Alumno

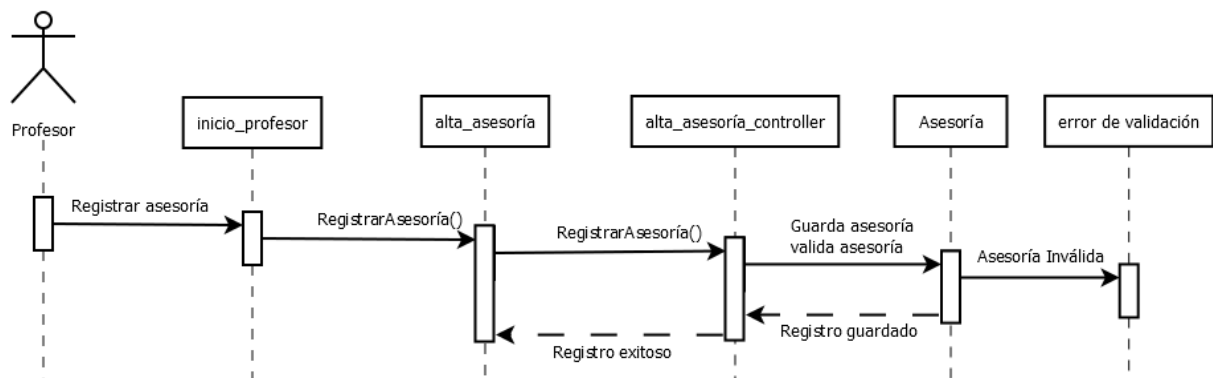
Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software



5.2.8. Eliminar Alumno

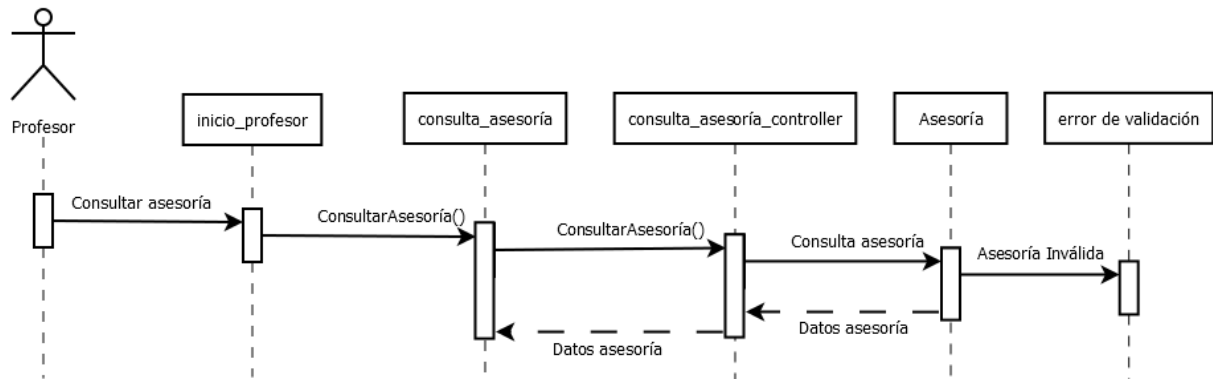


5.2.9. Agregar Asesoría

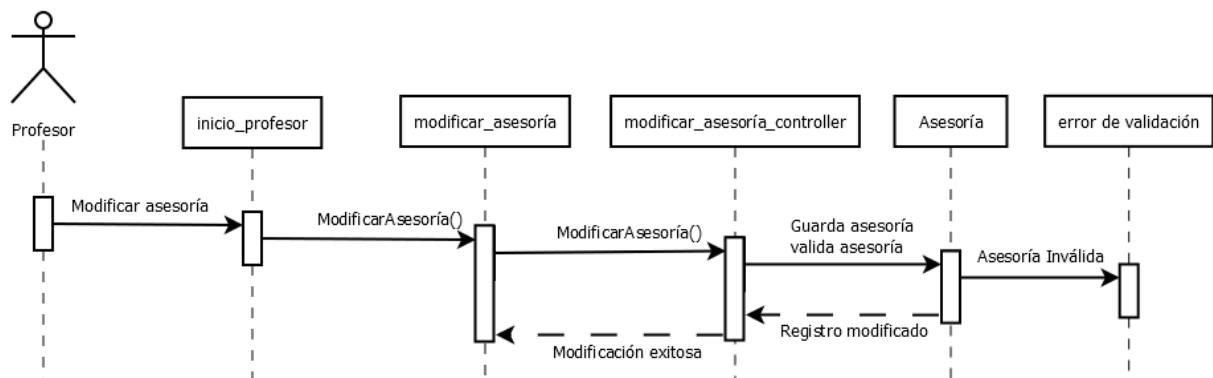


5.2.10. Consultar Asesorías

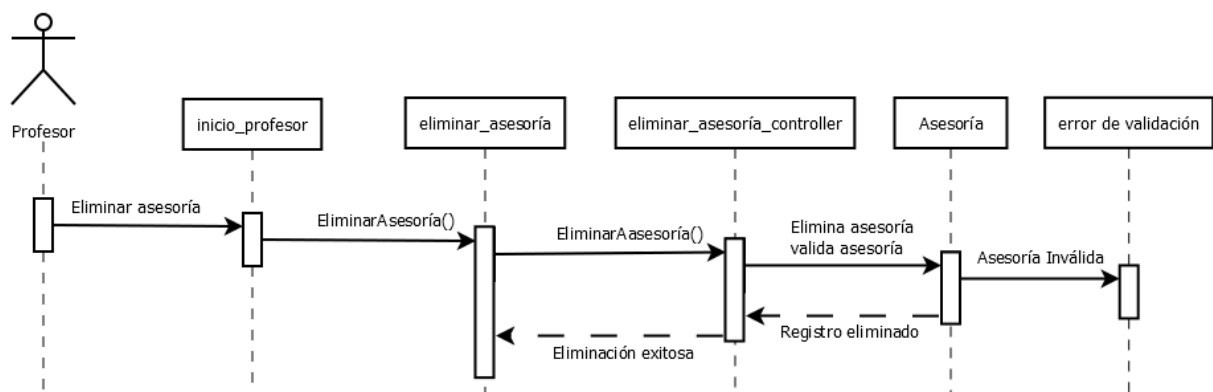
Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software



5.2.11. Modificar Asesoría

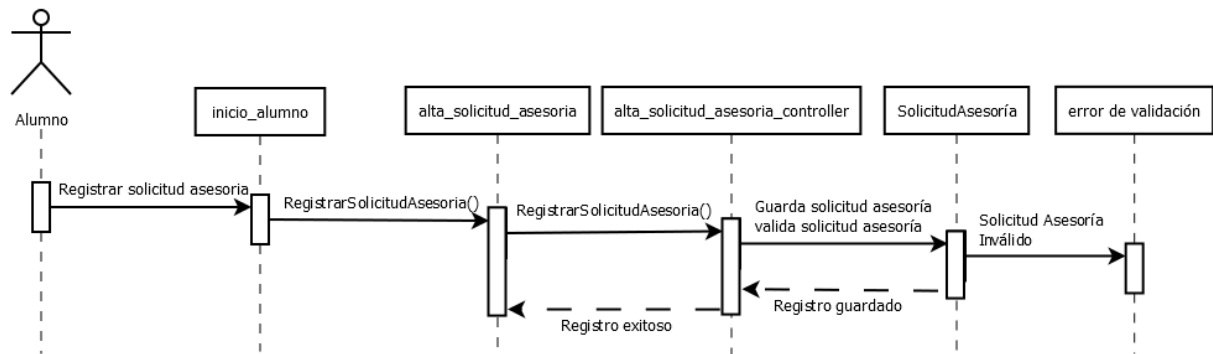


5.2.12. Eliminar Asesoría

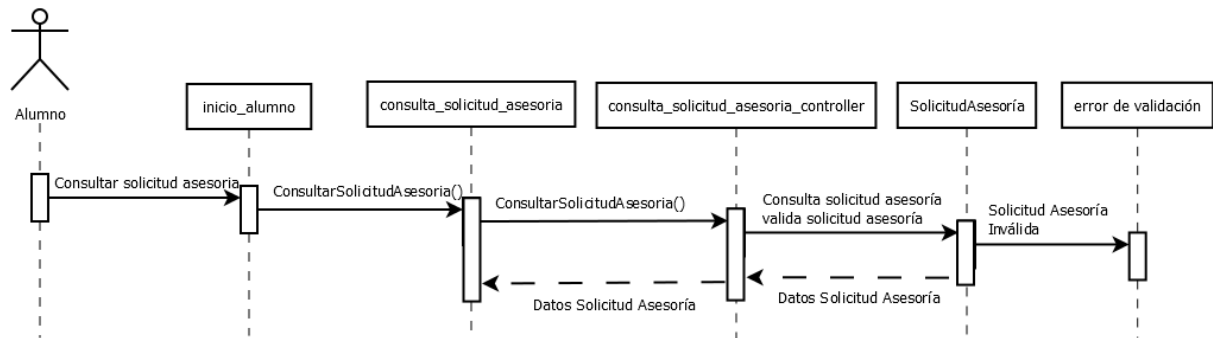


5.2.13. Agregar Solicitud de Asesoría

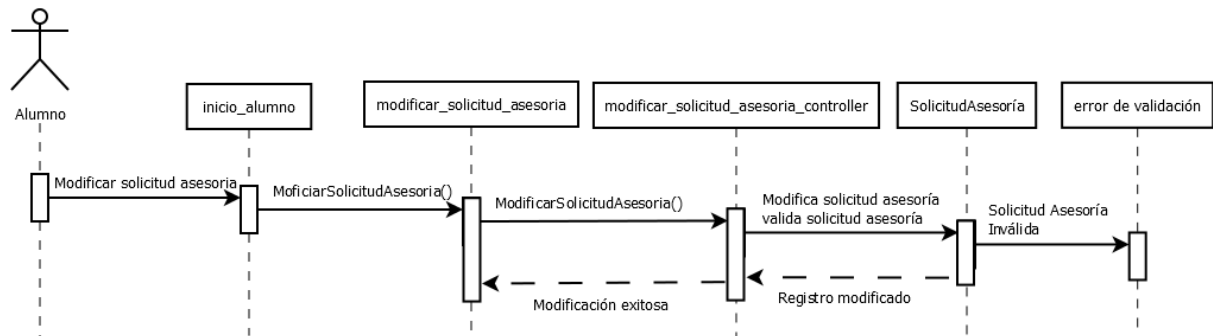
Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software



5.2.14. Consultar Solicitud de Asesoría



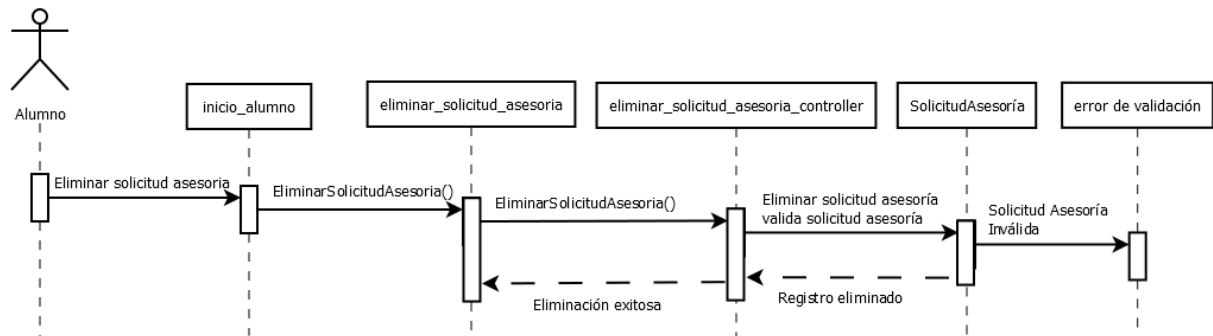
5.2.15. Modificar Solicitud de Asesoría



5.2.16. Eliminar Solicitud de Asesoría



Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software





5.3. Diagramas de navegación



Ingeniería de la Computación, Universidad Nacional Autónoma de México
Programación Orientada a Objetos - Diseño de software

