

Universidad de Guadalajara
C.U.C.E.I.
Sem. de ingeniería de software 1
Sección: Do1
Mtra.: Karla Avila Cárdenas



Práctica 8:
Diagramas de integración

Shimoda Emily Tomomi
216579882 - Ingeniería en Computación
Luis Daniel Zamora Delgadillo
217761153 - Ingeniería Informática

21B

Índice

Índice	2
Introducción	3
Desarrollo	4
Diagrama de bloques	7
Resultados	7
Diagrama de colaboración	7
Diagrama de secuencia	8
Especificación	8
Conclusiones	8
Referencias	9

Introducción

Tal como su nombre lo sugiere, un diagrama de integración es un tipo de diagrama UML que se emplea para captar el comportamiento interactivo de un sistema. Los diagramas de integración se centran en describir el flujo de mensajes dentro de un sistema y ofrecen contexto para una o más líneas de vida dentro de un sistema. Además, los diagramas de integración pueden emplearse para representar las secuencias ordenadas dentro de un sistema, y actúan como medio para visualizar los datos en tiempo real vía UML.⁽²⁾

Los diagramas de integración entran dentro de los **diagramas de comportamiento**. El enfoque aquí está en los aspectos dinámicos del sistema de software o proceso. En estos diagramas se muestra la funcionalidad de un sistema y se enfatiza lo que debe ocurrir en el sistema que se está modelando.⁽¹⁾

Este tipo de diagramas traen una serie de beneficios con ellos, tales como:

- Modelar un sistema como una serie de secuencia de eventos ordenados por tiempo
- Hacer ingeniería inversa o directa de un sistema o proceso
- Organizar la estructura de diversos eventos interactivos
- Transmitir de manera simple el comportamiento de los mensajes y de las líneas de vida dentro de un sistema
- Identificar las posibles conexiones entre los elementos de la línea de vida⁽²⁾

Así como con los diagramas estructurales, existen varios tipos de diagramas de integración o de comportamiento. Para esta práctica se realizarán diagramas de secuencia, de comunicación y de objetos:

- Los **diagramas de colaboración** representan las relaciones e interacciones entre distintos objetos de software.⁽²⁾ Enfatizan los aspectos estructurales de un diagrama de interacción y se centran en la arquitectura de los objetos, más que en el flujo de los mensajes⁽²⁾

Tiene los siguientes beneficios:

- Destacan cómo se conectan las líneas de vida
- Se centran en los elementos dentro de un sistema, más que en el flujo de mensajes
- Proporcionan más énfasis en la organización que en la cronología

Este tipo de diagramas también presentan algunas desventajas, así como:

- Pueden resultar demasiado complejos
 - Dificultan explorar objetos específicos dentro de un sistema
 - Crearlos puede demandar demasiado tiempo
- El **diagrama de secuencia** es otra opción para representar las interacciones. Es popular más allá de la comunidad de diseño, este diagrama visualmente atractivo es bueno para mostrar todo tipo de procesos empresariales. Simplemente revela la

estructura de un sistema, mostrando la secuencia de mensajes e interacciones entre actores y objetos cronológicamente. Los diagramas de secuencia muestran iteraciones y ramificaciones simples. Es favorable al realizar múltiples tareas.⁽¹⁾

Si cambia la secuencia de los eventos, pueden producirse demoras o puede llegar a caerse el sistema. Es importante seleccionar la notación que corresponda a la secuencia particular dentro de tu diagrama. Sus beneficios son:⁽²⁾

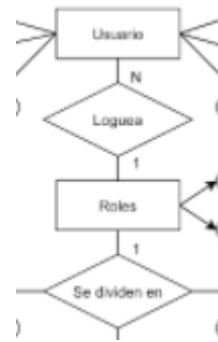
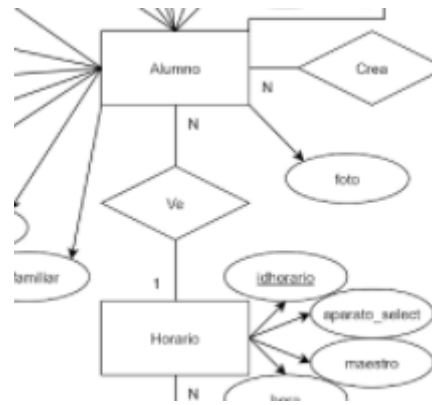
- Son fáciles de mantener y de generar
- Son fáciles de actualizar según los cambios en el sistema
- Permiten realizar ingeniería exacta y directa

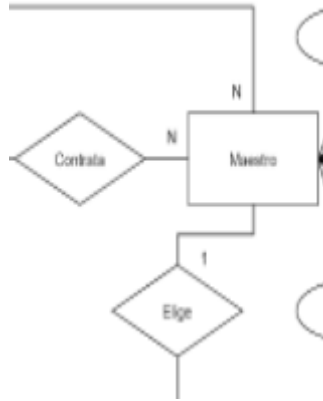
Sus desventajas son:

- Pueden resultar complejos, con demasiadas líneas de vida y notaciones variadas
 - Es fácil hacerlos incorrectamente y dependen de que ingreses la secuencia de manera correcta
- Un **diagrama de objeto** representa una instancia específica de un diagrama de clases en un momento determinado en el tiempo,⁽³⁾ tiene muchas similitudes con el diagrama de clases realizado en la práctica anterior. Este tipo de diagrama se enfoca en los atributos de un conjunto de objetos y cómo esos objetos se relacionan entre sí. Algunas de sus aplicaciones son:⁽³⁾
- Revisión de una iteración específica de un sistema general
 - Obtención de un avista de nivel alto del sistema a desarrollar
 - Prueba de un diagrama de clases creado para la estructura general del sistema, por medio de diagramas de objeto para casos de uso específicos

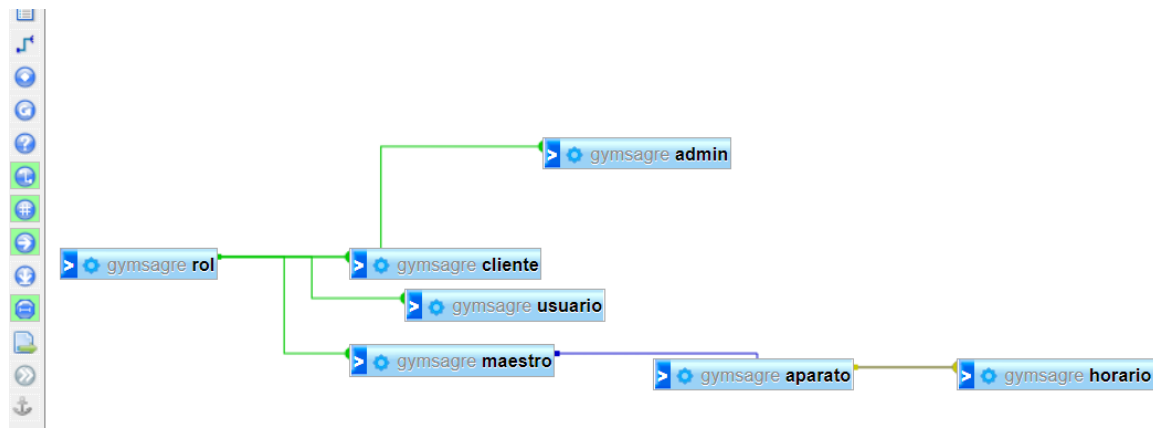
Desarrollo

Gracias al diagrama de entidad relación que se hizo previamente en las cuales utilizamos las siguientes entidades con sus respectivas relaciones





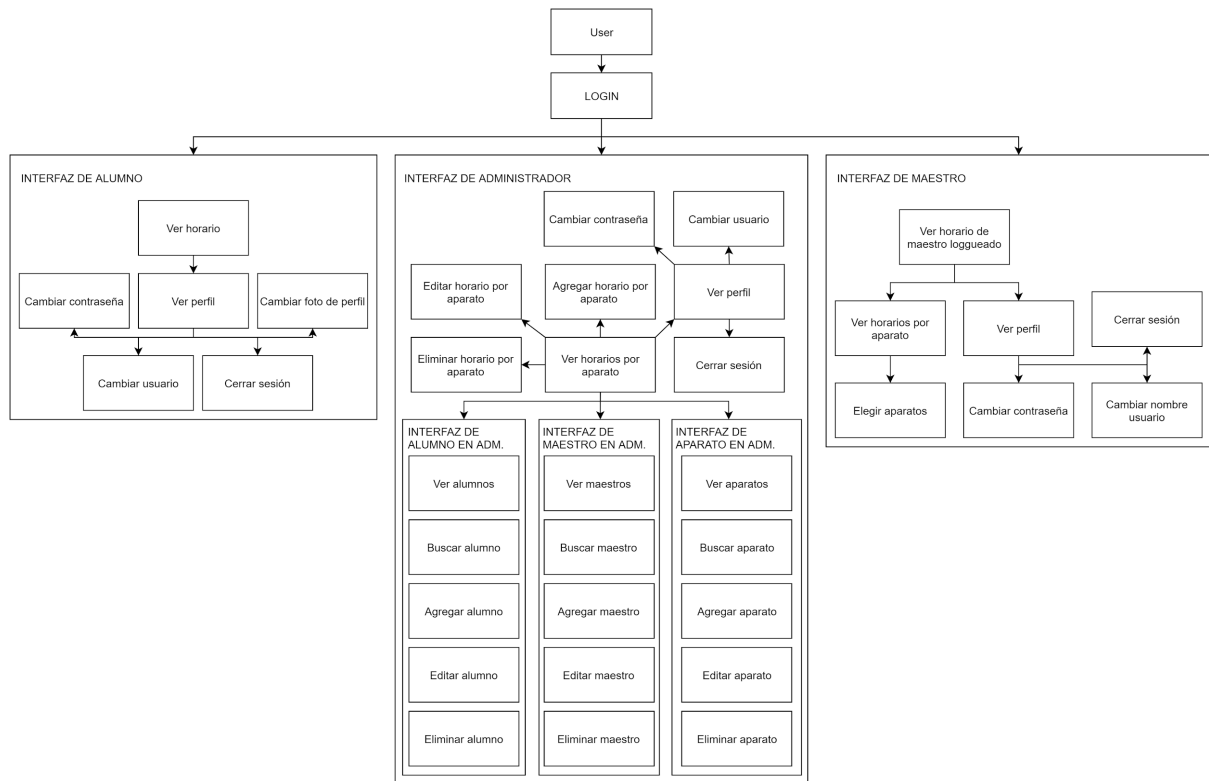
y un pequeño diagrama de como esta el modelo relacional de la base de datos



Es que podemos sacar el diagrama de colaboración.

También con la ayuda del diagrama de bloques previamente realizado podemos hacer el diagrama de secuencia.

Diagrama de bloques



Resultados

Diagrama de colaboración

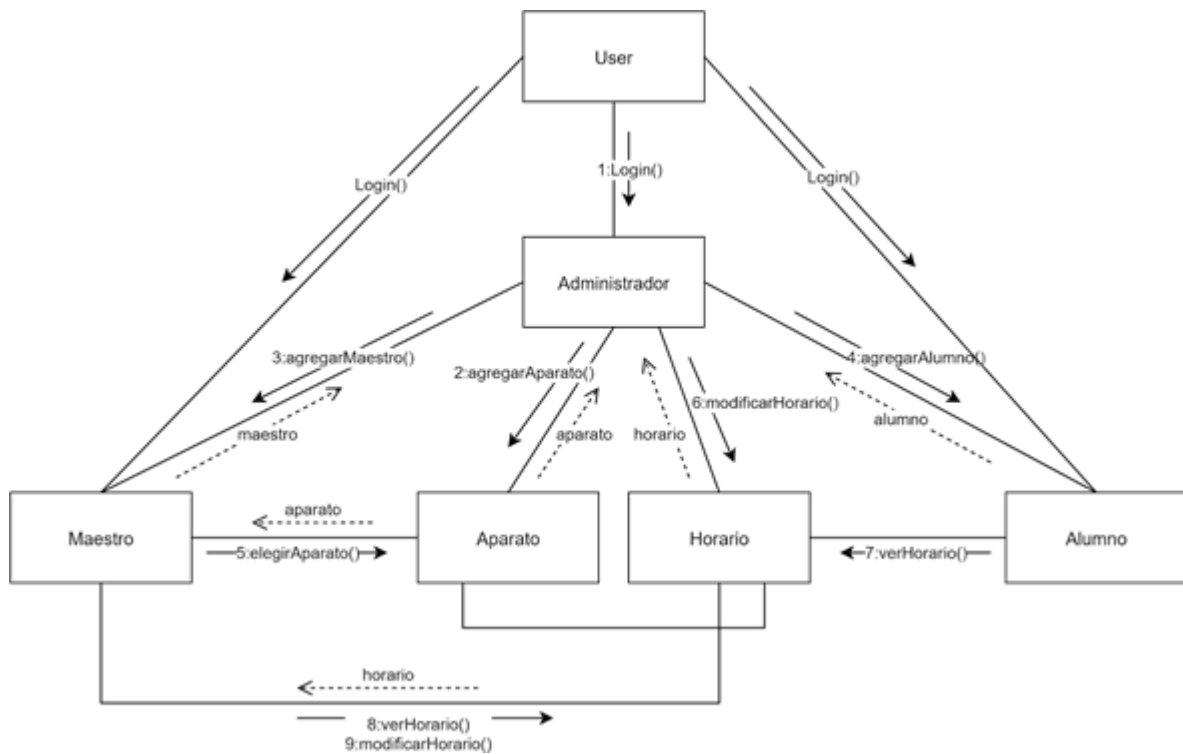
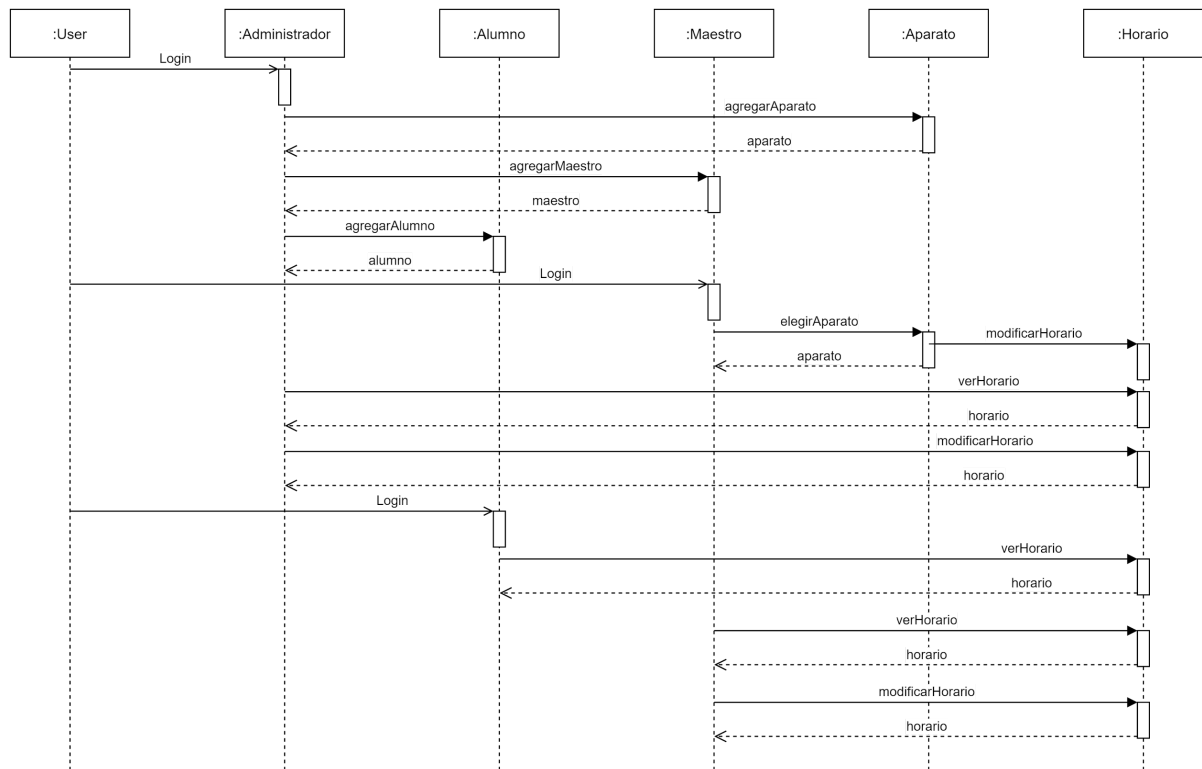


Diagrama de secuencia



Especificación

Como en el diseño estructural no se hizo la elaboración del diagrama de clases ni las tarjetas CRC, en esta parte no se realizó el diagrama de objetos ya que este es para detallar y dar mas especificaciones de los dos diagramas previamente mencionados los cuales son el de clases y las tarjetas.

Conclusiones

Daniel: En mi opinión con este tipo de diagramas nos adentramos a una mayor especificación y un mayor detalle de lo que se espera de la funcionalidad del programa cuando éste ejecuta sus procesos, además de ver la secuencia y cronología que los eventos llevan a cabo, como también lo que involucra algún evento determinado.

Emily: Creo que esos diagramas nos ayudan a entender mejor el flujo de actividades que se espera que lleve el usuario al interactuar con el programa, lo que ayudará también a identificar posibles errores humanos que pueden afectar este flujo y por lo tanto, alterar el comportamiento del programa, que puede desencadenar en un error. Conocer estos flujos pueden ayudarnos a prevenir estos errores, además de ayudar al diseño de una interfaz que facilite el seguimiento del flujo esperado.

Referencias

- (1) Microsoft 365 Team (2019) “La guía sencilla para la diagramación de UML y el modelado de la base de datos” [online] Microsoft. Disponible en: <https://www.microsoft.com/es-mx/microsoft-365/business-insights-ideas/resources/guide-to-uml-diagramming-and-database-modeling> [Obtenido el 23 de octubre 2021]
- (2) Lucidchart (s/a) “Todo acerca de los diagramas de interacción UML” [online] Lucidchart. Disponible en: <https://www.lucidchart.com/pages/es/diagrama-de-interaccion-uml> [Obtenido el 25 de octubre 2021]
- (3) Lucidchart (s/a) “Tutorial de diagrama de objetos” [online] Lucidchart. Disponible en: <https://www.lucidchart.com/pages/es/diagrama-de-objetos-uml> [Obtenido el 25 de octubre 2021]