

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



**Práctica 7:**  
**Diseño estructural**  
**Shimoda Emily Tomomi**  
**216579882 - Ingeniería en Computación**  
**Luis Daniel Zamora Delgadillo**  
**217761153 - Ingeniería Informática**

**21B**

## Índice

Índice	2
Introducción	3
Desarrollo	4
Modelo entidad-relación	4
Diccionario de datos	4
Resultados	7
Diagrama de bloques	7
Modelo Relacional	8
Especificación	8
Conclusiones	8
Referencias	9

## Introducción

El Lenguaje Unificado de Modelado (UML) desempeña un rol importante no solo en el desarrollo de software, sino también en los sistemas que no tienen software en muchas industrias, ya que es una forma de mostrar visualmente el comportamiento y la estructura de un sistema o proceso. UML ayuda a mostrar errores potenciales en las estructuras de aplicaciones, el comportamiento del sistema y otros procesos empresariales. Su objetivo principal fue desarrollar una forma menos caótica de representar el cada vez más complejo desarrollo de software, a la vez que separaban la metodología del proceso.<sup>(1)</sup>

Existen dos tipos principales de diagramas UML: diagramas de estructura y diagramas de comportamiento con varias subcategorías.

**Los diagramas estructurales** representan la estructura estática de un software o sistema, y también muestran diferentes niveles de abstracción e implementación. Estos se usan para ayudar a visualizar las diversas estructuras que componen un sistema, como una base de datos o aplicación. Muestran la jerarquía de componentes o módulos y cómo se conectan e interactúan entre sí. Estas herramientas ofrecen orientación y garantizan que todas las partes de un sistema funcionen según lo previsto en relación con todas las demás partes.<sup>(1)</sup>

Algunos diagramas estructurales son los siguientes:

- **Diagrama de clases.** Este diagrama, el más común en el desarrollo de software, se usa para representar el diseño lógico y físico de un sistema, y muestra sus clases. Tiene un aspecto similar al del diagrama de flujo porque las clases se representan con cuadros. Este diagrama ofrece una imagen de las diferentes clases y la forma en la que se interrelacionan, y cada clase posee tres compartimientos: Nombre de la clase (sección superior), atributos de la clase (sección central) y métodos u operaciones de la clase (sección inferior) <sup>(1)</sup>. De la mano del diagrama de clases, podemos encontrar las tarjetas CRC y los diccionarios que al igual que el anterior describen las clases y que se van a usar, sus métodos y atributos.
- **Diagrama de objetos.** Diagrama de objetos. A menudo, este diagrama se usa como una forma de comprobar la revisión de un diagrama de clases para fines de precisión. En otras palabras, ¿funcionará en la práctica? Muestra los objetos de un sistema y sus relaciones, y ofrece una mejor visión de los potenciales defectos de diseño que necesitan reparación. <sup>(1)</sup>

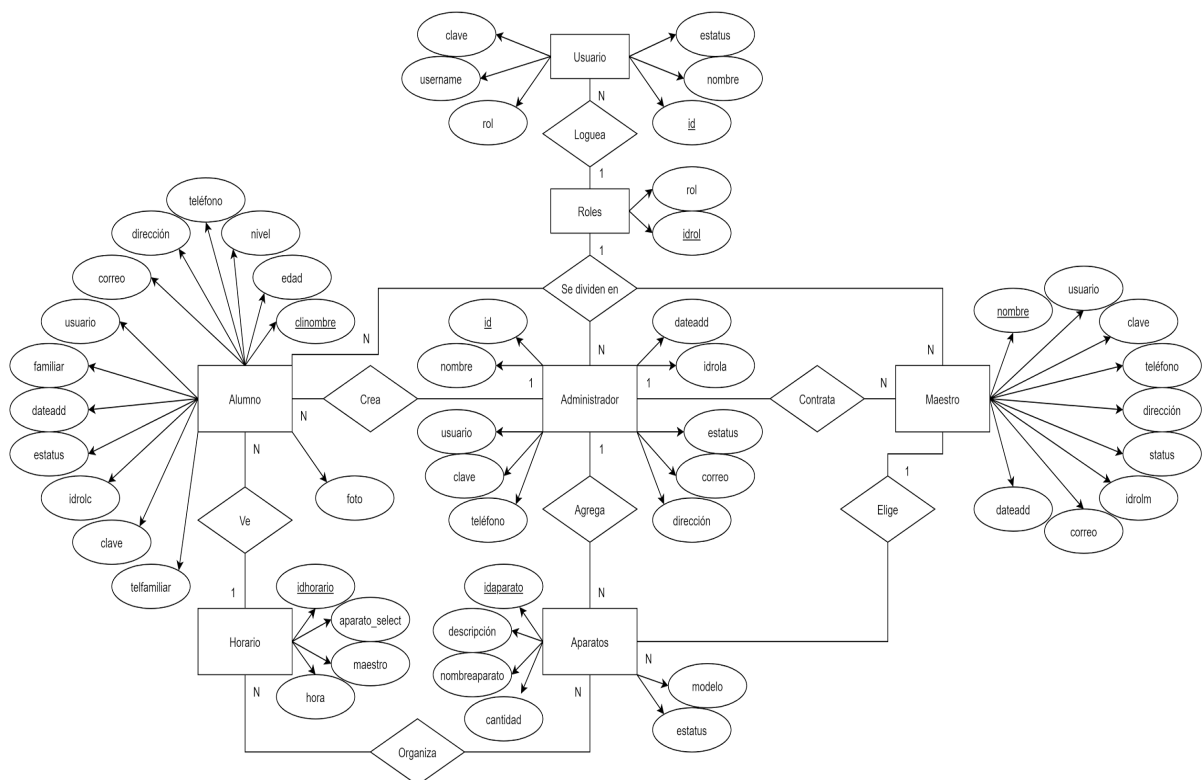
El UML también ha ganado popularidad como indicación para **modelar bases de datos**. Estos modelos son una gran herramienta visual para generar ideas, diagramas de forma libre y colaborar en ideas. <sup>(1)</sup>

Si bien el UML no tiene especificaciones para el modelado de datos, puede ser una herramienta útil para la creación de diagramas, especialmente porque los datos de las bases de datos se pueden usar en la programación orientada a objetos. Estos diagramas también son estructurales y algunos ejemplos de este tipo de diagramas son:

- **Modelo relacional.** Aquí, los datos se estructuran utilizando relaciones o estructuras matemáticas similares a una cuadrícula que tienen columnas y filas. Básicamente, es una tabla. <sup>(1)</sup>
- **Modelo entidad-relación.** Este se compone de tipos de entidad (personas, lugares o cosas). Muestra las relaciones que pueden existir entre ellos. Al definir las entidades, sus atributos y mostrar las relaciones entre ellas, un diagrama ER ilustra la estructura lógica de las bases de datos. <sup>(1)</sup>

## Desarrollo

### Modelo entidad-relación



### Diccionario de datos

Nombre	Clasificación	Tipo De Dato	Tamaño	Restricción
Admin	Clase	N/A	N/A	N/A

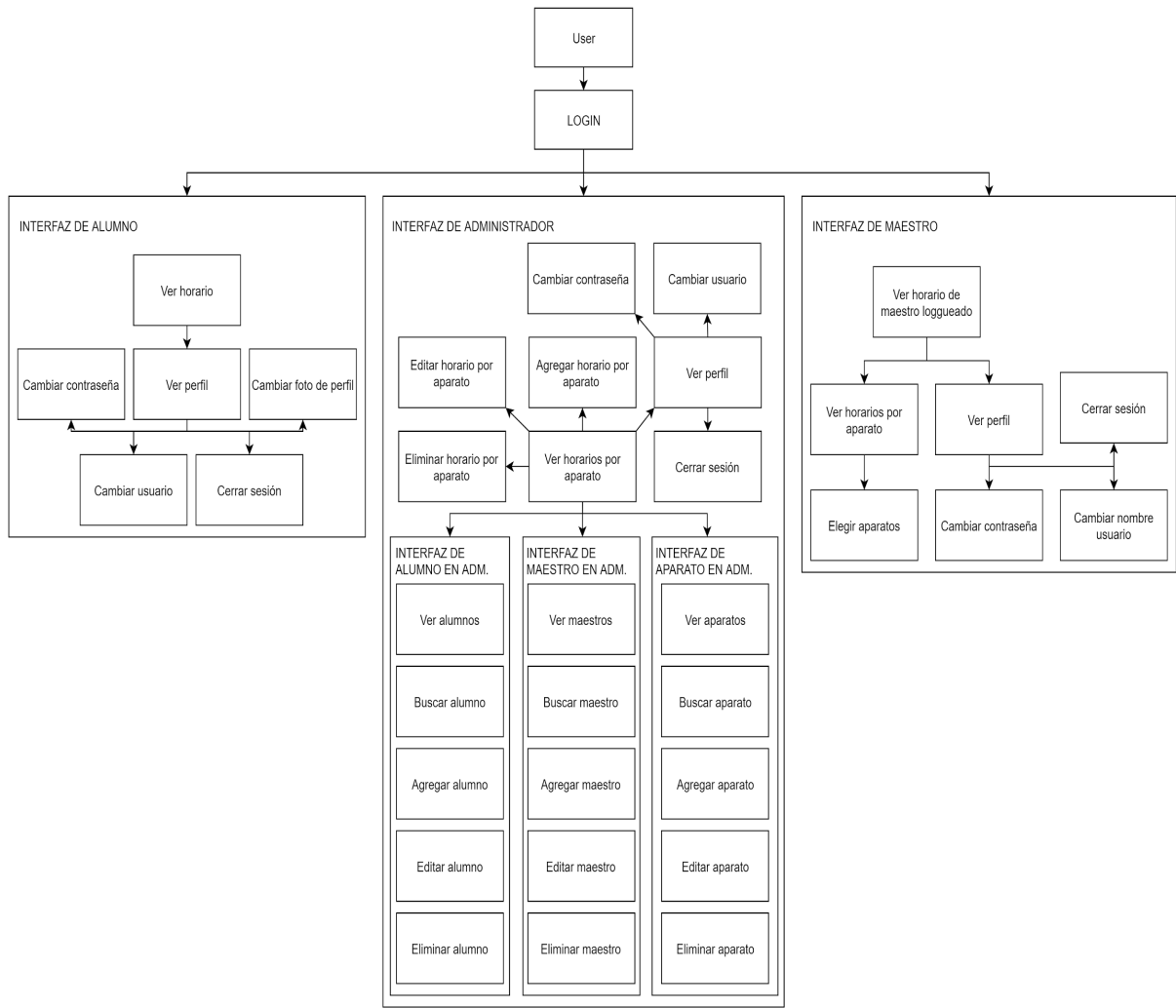
<b>Adminnombre</b>	<b>Atributo</b>	<b>Varchar</b>	<b>80</b>	<b>Not Null</b>
<b>Agrega</b>	<b>Relación</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>
<b>Aparato</b>	<b>Clase</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>
<b>Aparato_select</b>	<b>Atributo</b>	<b>Varchar</b>	<b>60</b>	<b>FK</b>
<b>Clave</b>	<b>Atributo</b>	<b>Varchar</b>	<b>50</b>	<b>Not Null</b>
<b>Cliente</b>	<b>Clase</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>
<b>Clinombre</b>	<b>Atributo</b>	<b>Varchar</b>	<b>80</b>	<b>PK</b>
<b>Correo</b>	<b>Atributo</b>	<b>Varchar</b>	<b>100</b>	<b>Ninguna</b>
<b>Dateadd</b>	<b>Atributo</b>	<b>Datetime</b>	<b>Preestablecido</b>	<b>Not Null</b>
<b>Descripción</b>	<b>Atributo</b>	<b>Text</b>	<b>Preestablecido</b>	<b>Not Null</b>
<b>Dirección</b>	<b>Atributo</b>	<b>Text</b>	<b>Preestablecido</b>	<b>Not Null</b>
<b>Elige</b>	<b>Relación</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>
<b>Estatus</b>	<b>Atributo</b>	<b>Int</b>	<b>11</b>	<b>Not Null</b>
<b>Familiar</b>	<b>Atributo</b>	<b>Varchar</b>	<b>60</b>	<b>Not Null</b>
<b>Fecha_nac</b>	<b>Atributo</b>	<b>Date</b>	<b>Preestablecido</b>	<b>Not Null</b>
<b>Hora</b>	<b>Atributo</b>	<b>Datetime</b>	<b>Preestablecido</b>	<b>Not Null</b>
<b>Horario</b>	<b>Clase</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>
<b>Foto</b>	<b>Atributo</b>	<b>LONGBLOB</b>	<b>Preestablecido</b>	<b>Ninguno</b>

<b>Id</b>	<b>Atributo</b>	<b>Int</b>	<b>11</b>	<b>PK</b>
<b>Idaparato</b>	<b>Atributo</b>	<b>Int</b>	<b>11</b>	<b>Not Null</b>
<b>Idhorario</b>	<b>Atributo</b>	<b>Int</b>	<b>11</b>	<b>PK</b>
<b>Idrol</b>	<b>Atributo</b>	<b>Int</b>	<b>11</b>	<b>PK</b>
<b>Idrola</b>	<b>Atributo</b>	<b>Int</b>	<b>11</b>	<b>FK</b>
<b>Idrole</b>	<b>Atributo</b>	<b>Int</b>	<b>11</b>	<b>FK</b>
<b>Idrolm</b>	<b>Atributo</b>	<b>Int</b>	<b>11</b>	<b>FK</b>
<b>Logea</b>	<b>Relación</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>
<b>Maestro</b>	<b>Clase</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>
<b>maestro</b>	<b>Atributo</b>	<b>Varchar</b>	<b>80</b>	<b>FK</b>
<b>Maestronombre</b>	<b>Atributo</b>	<b>Varchar</b>	<b>80</b>	<b>PK</b>
<b>Modelo</b>	<b>Atributo</b>	<b>Varchar</b>	<b>50</b>	<b>Not Null</b>
<b>Nivel</b>	<b>Atributo</b>	<b>Varchar</b>	<b>30</b>	<b>Not Null</b>
<b>Nombre</b>	<b>Atributo</b>	<b>Varchar</b>	<b>80</b>	<b>Not Null</b>
<b>Nombreatarato</b>	<b>Atributo</b>	<b>Varchar</b>	<b>60</b>	<b>PK</b>
<b>Rol</b>	<b>Clase</b>	<b>N/A</b>	<b>N/A</b>	<b>N/A</b>
<b>rol</b>	<b>Atributo</b>	<b>Int</b>	<b>11</b>	<b>FK</b>
<b>rol</b>	<b>Atributo</b>	<b>Varchar</b>	<b>20</b>	<b>Not Null</b>

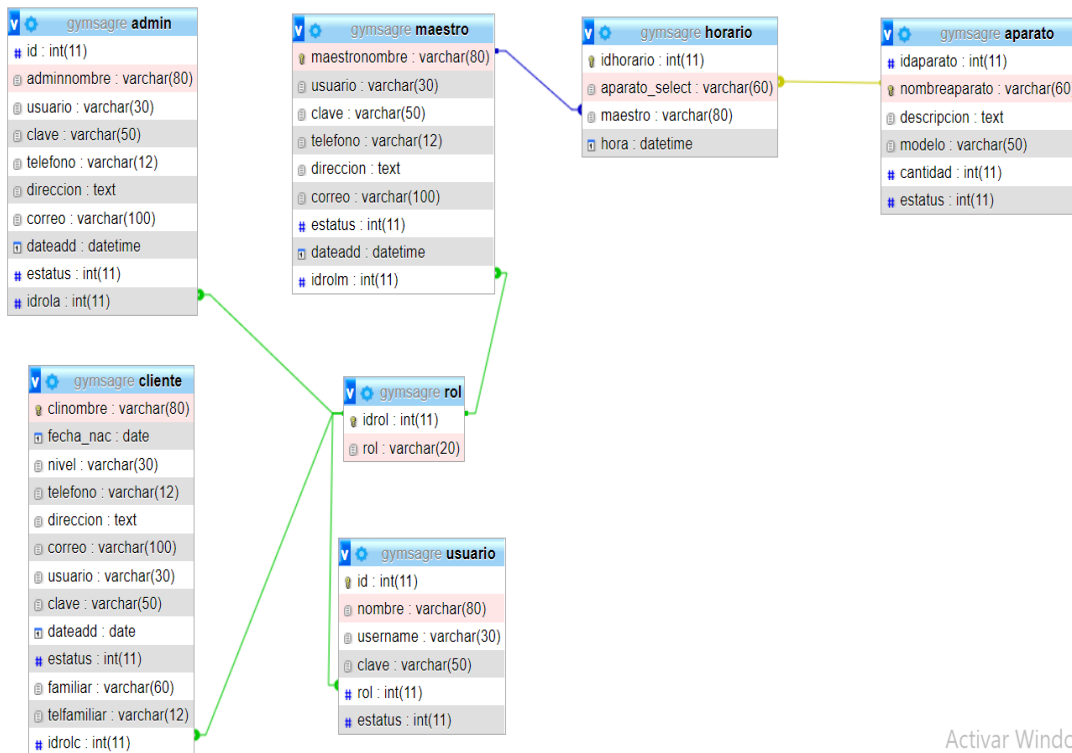
Organiza	Relación	N/A	N/A	N/A
Teléfono	Atributo	Varchar	12	Not Null
Telfamiliar	Atributo	Varchar	12	Not Null
Usuario	Clase	N/A	N/A	N/A
usuario	Atributo	Varchar	80	Not Null
Ve	Relación	N/A	N/A	N/A

### Resultados

#### Diagrama de bloques



## Modelo Relacional



Activar Windows  
Ve a Configuración para activar Win

## Especificación

Se decidió que el sistema fuera programado de manera secuencial, y no con poo (Programación orientada a objetos), por ende no se realizaron los diagramas de clases y por ende las tarjetas CRC, ya que con pequeños scripts y funciones ya establecidas propias de la base de datos y php como lo son los triggers (Disparadores) y Sesiones, se puede realizar lo requerido de una manera más sencilla y especializada ya que este sistema está más enfocado a una página web que requiere el despliegue de información correcta y un inicio de sesión por roles y no un CMS o un ERP como tal.

## Conclusiones

**Luis Daniel:** En mi opinión todos estos diagramas que hemos hecho y visto en el presente documento nos sirve para ya hacer una clara definición de cómo se empezara a implementar la estructura del sistema, la gestión de la base de datos y la función que esta tendrá, además de facilitarnos la representación visual del modelo de datos, sus atributos, relaciones, sus tipos y cómo se utilizará, dejándonos como resultado dejar bien aterrizadas las ideas de lo que se pretende lograr con el sistema.

**Emily:** Los diagramas estructurales son muy útiles para identificar las partes en que se dividirá un sistema, además de marcar un buen punto de inicio al definir cómo será estructurada la base de datos, el definir las tablas que van a usarse, así como sus campos y relaciones con anticipación puede ayudar a facilitar etapas más avanzadas del proceso, como



el código e incluso al diseño, al tener una idea de cómo van a relacionarse las distintas partes del sistema y qué debe incluir cada una. Creo que estos diagramas también ayudan a organizar mejor las ideas que se tienen de cómo van a relacionarse los elementos del sistema

### **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]