Jesús Alejandro Flores Hernández

FCI

Manual de practicas

Diseño OO

Contenido

[Objetivo 1](#_Toc92878678)

[Introducción 1](#_Toc92878679)

[Modelado OO 2](#_Toc92878680)

[Narrativa 3](#_Toc92878681)

[Ejercicio 1 Narrativa 4](#_Toc92878682)

[Diseño de clases 4](#_Toc92878683)

[Ejercicio 2 Diseño de clases 4](#_Toc92878684)

[Diseño Atributos 5](#_Toc92878685)

[Ejercicio 3 Atributos 6](#_Toc92878686)

[Diseño Operaciones 7](#_Toc92878687)

[Ejercicio 4 Modelado de clases, atributos y operaciones 7](#_Toc92878688)

[Relaciones entre objetos 11](#_Toc92878689)

[Agregación 12](#_Toc92878690)

[Multiplicidad en asociaciones 14](#_Toc92878691)

[Asociación reflexiva 14](#_Toc92878692)

[Agregación 15](#_Toc92878693)

[Herencia 15](#_Toc92878694)

[Ejercicio 5 Modelado UML 16](#_Toc92878695)

[Ejercicio 6 Modelado UML 16](#_Toc92878696)

# Objetivo

Entender los conceptos de diseño de clases y sus relaciones tales como asociación, agregación y herencia, diseñar modelos de clases y sus asociaciones usando la simbología UML.

# Introducción

La programación orientada a objetos es un paradigma que nos permite manejar de una mejor forma la complejidad de la construcción de sistemas, implica sin embargo un cambio radical en la forma de pensamiento que con tanto trabajo construyo un programador estructurado. Las ventajas de utilizar la POO valen la pena el esfuerzo del cambio de paradigma, es importante tomar en cuenta el lenguaje sobre el que se trabaja, entre los lenguajes OO modernos están: Java, Ruby, D, C++, C#, Visual Basic, Object Pascal, debido a razones de disponibilidad, popularidad, disponibilidad de material de ayuda, se elige VB para este curso sin embargo el lector debería probar otros lenguajes y compararlos de forma analítica

# Modelado OO

Un modelo se define como una abstracción del mundo real. Primeramente, debemos identificar que vamos a modelar y en nuestro caso son los elementos que interactúan en un problema a los que normalmente llamaremos objetos. Ejemplo:

Problema: Reinscripción de alumnos.

"Juan llega con la secretaria académica de la FCI para realizar su reinscripción, solicita ser atendido, espera su turno, da su matrícula y entrega la lista de materias que tomará en el próximo ciclo…".

En este fragmento de problema encontramos a "Juan" a la "secretaria académica" y las "materias" estos elementos interactúan en el problema y son llamados objetos, como sabemos que algo es un objeto, bueno desde el momento que debo distinguirlo de otros, por ejemplo, las personas se distinguen de otras por medio de sus atributos como nombre sexo, edad, las materias debo distinguirlas una de otra también.

Es de importancia crítica conocer muy bien el problema a modelar, es frecuente cometer el error de empezar a modelar sin un conocimiento pleno del problema, si somos capaces de realizar manualmente las operaciones que el problema conlleva entonces lo entendemos, cuando pido a mis alumnos que escriban un modelo que describa la multiplicación de dos matrices, en algunos casos, incluso empiezan a escribir líneas de código sin saber siquiera como se multiplican matrices. El primer paso es asegurarse de que se comprende bien el proceso que modelaremos, esto se logra preguntando, tomando notas, estudiar las notas, preguntar más para resolver dudas, y finalmente escribir una narrativa que describa el problema, esta será de 1 a 10 cuartillas dependiendo del problema y puede ser incluso mayor para sistemas más complejos.

Los conjuntos de objetos son clasificados es decir agrupados según su tipo, y esas clases forman parte del modelo del problema, primeramente, ubicamos los objetos. Una vez identificados los agrupamos según su tipo, por ejemplo si en un problema encontramos a Juan, María, Pedro, José, ... todos son objetos y todos son del mismo tipos el tipo persona, entonces abstraemos en una clase, la clase Persona y la modelamos,  para modelar la clase debemos analizar el ámbito en que queremos aplicar el modelo, por ejemplo un objeto persona la puedo ver desde el punto de vista de un empleado o de un alumno o de un esposo o esposa, etc., por tanto el ámbito será: Empresa, Escuela, Hogar, etc. En dependencia del ámbito identificamos un conjunto de características (también llamadas atributos) del objeto, notemos que una entidad puede tener miles de características, pero la mayoría no serían pertinentes en un ámbito en particular, si estoy modelando una persona como alumno en el ámbito de una escuela, serian importantes características como: Nombre, Edad, Calificación, pero serian irrelevantes otras como: Estatura, Peso, Estado Civil, nuestro modelo de persona seria entonces:

Modelo Persona con atributos:

            Nombre

            Edad

            Calificación

Este es un modelo por qué no incluye todas las características de la persona, sino solo algunas de ellas, el resto se perdieron como parte del proceso de abstracción, notemos además que este modelo incluye solo la abstracción de los atributos de la entidad, pero no incluye aun las operaciones que la persona puede realizar tales como: Estudiar, Leer, Caminar etc.

Una vez que encontramos los atributos de una clase de objetos debemos buscar sus operaciones, estas son las acciones que pueden realizar los objetos o que puedo realizar sobre ellos, por ejemplo un alumno puede almacenar (en su memoria) su nombre esta operación en un alumno real se realiza en su más tierna infancia pero para el alumno que estamos modelando se debe realizar al iniciar el programa, necesitamos por tanto una operación que permita almacenar el nombre del alumno en el atributo correspondiente, esta operación puede llamarse "guardarNombre", "bautizar", "setNombre" etc. lo importante es que entendamos que esta operación implica actualizar la información en el atributo "nombre" del alumno, en general las operaciones se ven reflejadas por un cambio en los valores de los atributos (es decir cambia su estado).

# Narrativa

Para modelar cualquier sistema debemos partir de una base que describa el problema a resolver, para ello nosotros usaremos una narrativa. Una narrativa es una descripción de un problema en particular, debe describir el problema los actores que intervienen en él y de manera muy importante los datos y los procesos que se utilizan en la solución del problema.

# Ejercicio 1 Narrativa

Escriba una narrativa para el control de la información de una farmacia:

Solución:

La farmacia CarmelosFarm se dedica a la compra venta de medicamentos con una única sucursal, cuenta con una lista de medicamentos de los cuales maneja una clave, una descripción, un proveedor, fecha de compra, fecha de caducidad, precio de compra, precio de venta, existencia, stock mínimo y stock máximo, Los medicamentos se guardan en un almacén desde donde se sacan al mostrador para ser vendidos, cada que un medicamento se vende se genera un registro de la venta que incluye número de venta, clave del medicamento y costo y el medicamento se resta de la existencia en almacén, cuando la existencia en almacén de algún medicamento se encuentra por debajo del stock mínimo, el sistema genera un pedido del medicamento al proveedor por una cantidad que se obtiene de restar stock máximo menos existencia, para esto la empresa cuenta con una cartera de proveedores de los que tiene una clave de proveedor, un nombre de proveedor, teléfono, correo electrónico, y tiempo de entrega. También se debe manejar un control de pedidos de medicamentos a los proveedores con una clave de pedido, clave de medicamento, clave de proveedor, cantidad y fecha de pedido.

# Diseño de clases

Una narrativa nos describe un problema, dentro del problema debemos descubrir clases y objetos. Los objetos son los elementos que “actúan” en el problema o sobre los que se “actúa”, por ejemplo, ***El alumno Juan Pérez calcula el promedio de sus calificaciones de la materia POO***, otra puede ser que ***Al Alumno Juan Pérez el profesor le asigna una calificación en la secuencia 1***, en este ejemplo Juan Pérez es un objeto y el cálculo de promedio es una operación que realiza el alumno y asignar calificación es una transformación (en sus datos) a la que se ve sujeto el alumno. Como vimos antes, los objetos también tienen atributos que son representados por datos, debemos reunir los datos y operaciones de un conjunto de objetos afines en un modelo, a este modelo se le llama **clase**, por ejemplo, ubique a todos sus compañeros del curso de programación, todos son objetos y todos son diferentes sin embargo tienen características comunes, a saber, todos tienen matricula, nombre, y dos calificaciones para dos experiencias de aprendizaje en el curso de POO. Soportan también las operaciones de calcular promedio y asignar calificación dada por el profesor a una experiencia, deben también tener métodos para alimentar la información inicial del objeto como el nombre y matricula y para recuperarla cuando sea necesario, Esto último es muy importante porque estamos trabajando con un modelo en la computadora, el alumno real sabe su nombre, pero el modelo informático de alumno debe ser alimentado con estos datos para que sea un reflejo del alumno real.

# Ejercicio 2 Diseño de clases

Diseñe la clase persona a partir de la narrativa anterior

Solución

Para encontrar las clases, lea su narrativa, subraye los sustantivos y haga una lista, en la narrativa anterior los sustantivos son:

* Farmacia carmelosFarm
* Lista de Medicamentos
* Sucursal
* Almacén
* Mostrador
* Medicamento
* Cartera de proveedores
* Proveedor
* Pedido
* Registro de venta

Estos son los candidatos a clases. A continuación, haga un análisis para eliminar los que no nos interesan, por ejemplo, el sustantivo carmelosFarm no lo voy a modelar, más bien estoy haciendo un modelo dentro de la farmacia, sucursal tampoco se modelará, pues no nos piden llevar el control de ella, a menos que fueran varias si necesitaríamos distinguirlas y almacenar datos diferentes de cada una, almacén si lo modelamos porque es uno, pero queremos hacer operaciones sobre él. nos quedan:

* Lista de Medicamentos
* Almacén
* Mostrador
* Medicamento
* Cartera de proveedores
* Proveedor
* Pedido
* Registro de venta

Y estas son las clases de nuestro problema.

# Diseño Atributos

Los atributos son las características de los objetos, para un Auto podemos tener color, marca, número de cilindros. Los atributos son datos y debemos cuidar que puedan ser representables, por ejemplo la marca del auto puede ser representado por una cadena de caracteres: “Volks Wagen”, el número de cilindros por un número entero: 4. Ahora pensemos en el atributo belleza, ¿cómo podríamos representar el atributo belleza?, como podríamos decir que un auto es bello, tal vez con un si o un no, tal vez sería mejor una escala como (si, no, regular) o (muy bonito, bonito, regular, feo, muy feo) o una escala del 0 al 10, con esto queremos hacer notar que cada que pensemos en un atributo para una entidad, debemos pensar también en el tipo de dato para su representación. Ahora cuales son los tipos datos con los que puedo representar atributos, los acostumbrados son:

*Tabla 1 Tipos de datos.*

|  |  |
| --- | --- |
| Tipos de datos | Identificador computacional |
| Cadenas de caracteres | String |
| Números enteros | int |
| Números reales | float, double |
| Valores lógicos (si, no) | boolean |
| Caracteres | char |

Cualquier atributo deberá ser representado usando estos tipos de datos a los que se denominan Tipos Primitivos.

# Ejercicio 3 Atributos

Diseñe los atributos para un alumno de la materia de POO queremos controlar la información de nombre edad, calificación 1 y calificación 2.

Solución:

*Clase alumno*

|  |
| --- |
| persona |
| nombre  edad  calif1  calif2 |
| guardarNombre  guardarEdad  devolverNombre  devolverEdad  guardarCalifExp1  guardarCalifExp2  calcularPromedio |

# Diseño Operaciones

Las operaciones de igual forma dependen del ámbito en el que estemos modelando al objeto, por ejemplo, las operaciones que realiza una persona como alumno no son las mismas que realiza una persona como profesor.

Una persona tendrá operaciones como comer, caminar, dormir, pero recuerde que estamos haciendo un modelo y las operaciones que nos interesan son las que tiene que ver con el ámbito de nuestro problema, por ejemplo, si un profesor quiere llevar el control de las calificaciones de los alumnos, las operaciones como caminar, comer, dormir, no le interesan, ¿Qué le interesa entonces como operaciones?, tal vez piense en la operación estudiar pero yo le diría que no, recuerde que el objetivo del profesor es tener el control de las calificaciones y no que el alumno aprenda, es decir para el profesor será importante que el alumno aprenda pero en otro ámbito, en un sistema que modele la enseñanza aprendizaje por ejemplo, pero en nuestro problema solo queremos llevar el control de las calificaciones. Que querría el profesor entonces, para levar el control necesita, almacenar una calificación, recuperarla cuando la necesite, saber el nombre de un alumno que saco tal o cual calificación. Por tanto las operaciones para un alumno serían: guardarNombre, devolverNombre, guardarCalificación, devolverCalificación. Es decir, yo tendre un conjunto de objetos alumnos a los que asignare calificación y la podré recuperar cuando la necesite.

# Ejercicio 4 Modelado de clases, atributos y operaciones

Problema:

Diseñe un modelo OO para la siguiente narrativa:

La empresa PROVPAR.S.A. Dedicada a la venta de refacciones automotrices desea llevar el control de su información para los proveedores, partes, pedidos y notas de venta, para esto la empresa cuenta con una agenda de proveedores en donde guarda el rfc, nombre, teléfono, dirección, e-mail y calificación del proveedor, donde calificación es un número del 1 al 10 con el que nuestra empresa califica el servicio prestado por el proveedor, si una empresa tiene un servicio excelente tendrá una calificación de 10 y si es pésimo será de 0. La empresa también cuenta con una lista de partes donde almacena una clave, nombre, existencia, stock mínimo, stock máximo, peso, cc y cv, donde stock mínimo es la cantidad mínima de este artículo que la empresa requiere en existencia es decir su existencia en almacén no debe bajar de esta cantidad, stock máximo es la cantidad máxima de artículos o partes que puede almacenar la empresa en su almacén, peso es el peso de la parte en kg y será un número entero, finalmente cc y cv son el costo que la empresa paga al proveedor par la parte y cv es el costo al que la empresa vende la parte. Para las notas de venta la empresa necesita la relación de todas las ventas efectuadas y para esto requiere almacenar la siguiente información por cada venta: número de venta, clave del artículo o parte vendido, cantidad, costo, iva y total, caber aclarar que cada nota incluye información de un solo tipo de artículo, es decir si un cliente compra 200 tornillos se realiza una sola nota, pero si compra 200 tornillos y 150 tuercas se realizarán dos notas una para los tornillos y otra para las tuercas . También necesita llevar el control de sus pedidos de partes a sus proveedores para lo que requiere una lista de pedidos con  clave del pedido, mes del pedido, clave del artículo o parte, clave del proveedor y cantidad solicitada, el mes del pedido es un número de 1 a 12, la cantidad solicitada es un numero entero. Los datos de clave de parte, clave de proveedor, clave de pedido son cadenas de 1 a 10 caracteres, el número de venta es un entero

Solución:

Subrayamos los sustantivos como candidatos a clases, aquí se muestran de azul, subrayamos los adjetivos como candidatos a atributos, aquí se muestran de rojo, y subrayamos los verbos como candidatos a operaciones.

La **empresa PROVPAR.S.A**. Dedicada a la venta de **refacciones automotrices** desea llevar el control de su información para los **proveedores**, **partes**, **pedidos** y **notas de venta**, para esto la empresa cuenta con una **agenda de proveedores** en donde guarda el **rfc, nombre, teléfono, dirección, e-mail y calificación del proveedor**, donde calificación es un número del 1 al 10 con el que nuestra empresa califica el servicio prestado por el proveedor, si una empresa tiene un servicio excelente tendrá una calificación de 10 y si es pésimo será de 0. La empresa también cuenta con una **lista de partes** donde almacena una clave, nombre, existencia, stock mínimo, stock máximo, peso, cc y cv, donde stock mínimo es la cantidad mínima de este artículo que la empresa requiere en existencia es decir su existencia en **almacén** no debe bajar de esta cantidad, stock máximo es la cantidad máxima de artículos o partes que puede almacenar la empresa en su almacén, peso es el peso de la parte en kg y será un número entero, finalmente cc y cv son el costo que la empresa paga al proveedor par la parte y cv es el costo al que la empresa vende la parte. Para las notas de venta la empresa necesita la **relación de todas las ventas efectuadas** y para esto requiere almacenar la siguiente información por cada venta: número de venta, clave del artículo o parte vendido, cantidad, costo, iva y total, caber aclarar que cada nota incluye información de un solo tipo de artículo, es decir si un cliente compra 200 tornillos se realiza una sola nota, pero si compra 200 tornillos y 150 tuercas se realizarán dos notas una para los tornillos y otra para las tuercas . También necesita llevar el control de sus pedidos de partes a sus proveedores para lo que requiere una lista de pedidos con clave del pedido, mes del pedido, clave del artículo o parte, clave del proveedor y cantidad solicitada, el mes del pedido es un número de 1 a 12, la cantidad solicitada es un numero entero. Los datos de clave de parte, clave de proveedor, clave de pedido son cadenas de 1 a 10 caracteres, el número de venta es un entero.

Se ubican los sustantivos en la narrativa (mostrados aquí en azul) y se hace una lista de ellos, estas serán las clases.

Clases

* Empresa Provpar
* Refacciones automotrices
* Agenda de proveedores
* Proveedores
* Partes
* Pedidos
* Notas de venta
* Lista de partes
* Almacén
* Relación de ventas

 Revisamos las clases anteriores para refinar el modelo, de aquí, notamos que según la narrativa solo se mencionan atributos y operaciones para las siguientes clases y nos quedamos con ellas eliminando las demás ya sea porque están repetidas o por que no las controlará el sistema:

Clases finales

* Refacciones
* Proveedores
* Agenda\_proveedores
* Partes
* Pedidos
* Notas\_venta
* Lista\_partes
* Almacen
* Relacion\_ventas

A continuación, asociamos atributos, por ejemplo

**rfc, nombre, teléfono, dirección, e-mail y calificación del proveedor son atributos del proveedor**

|  |
| --- |
| Proveedor |
| **rfc**  **nombre**  **teléfono**  **dirección**  **e-mail**  **calificación** |
|  |

 A continuación, identificamos las operaciones, algunas son implícitas, por ejemplo en la frase "llevar el control de su información para los proveedores,” llevar el control tiene implícito que deseamos altas, bajas, modificaciones de la información, esta funcionalidad de altas, bajas, cambios de parte a quien se le asigna?, bueno cambios en los datos de la parte se le asigna a la clase parte por lo que queda:

|  |
| --- |
| Proveedor |
| **rfc**  **nombre**  **teléfono**  **dirección**  **e\_mail**  **calificación** |
| Setrfc  getRfc  setNombre  getNombre  setTelefono  getTelefono  setDirección  getDireccion  setTelefono  getTelefono  setEmail  getEmail |

Las operaciones set y get servirán para introducir y sacar datos de la parte.

Las operaciones de Alta y baja de parte ¿a qué clase las asignamos?

No las podemos asignar a la clase parte porque no tendría sentido, es decir una parte la doy de alta en ¿dónde? Pues en la lista de partes que también podría ser el almacén y tenemos la clase almacén así:

|  |
| --- |
| Almacen |
| listaPartes |
| altaParte  bajaParte |

En este punto también nos damos cuenta de que si bien en un principio ubicamos lista de partes como una clase, queda mejor como un atributo de la clase almacén.

También notamos que para dar altas y bajas de pedidos necesitamos una nueva clase que podríamos llamar listaDpedidos.

**Por lo tanto, las clases que quedan son:**

1. Proveedor
2. AgendaProveedores
3. Parte
4. Almacen
5. Pedido
6. ListaPedidos
7. NotaVenta
8. ListaNotasVenta

# Relaciones entre objetos

Las clases se relacionan entre si y es necesario representar esas relaciones en la programación, por ejemplo, un profesor se relaciona con un alumno por medio de un curso, un hombre con una mujer con un matrimonio, un motor con un auto porque el auto contiene un motor, etcétera. Para representar las relaciones entre las clases de objetos usaremos la notación UML.

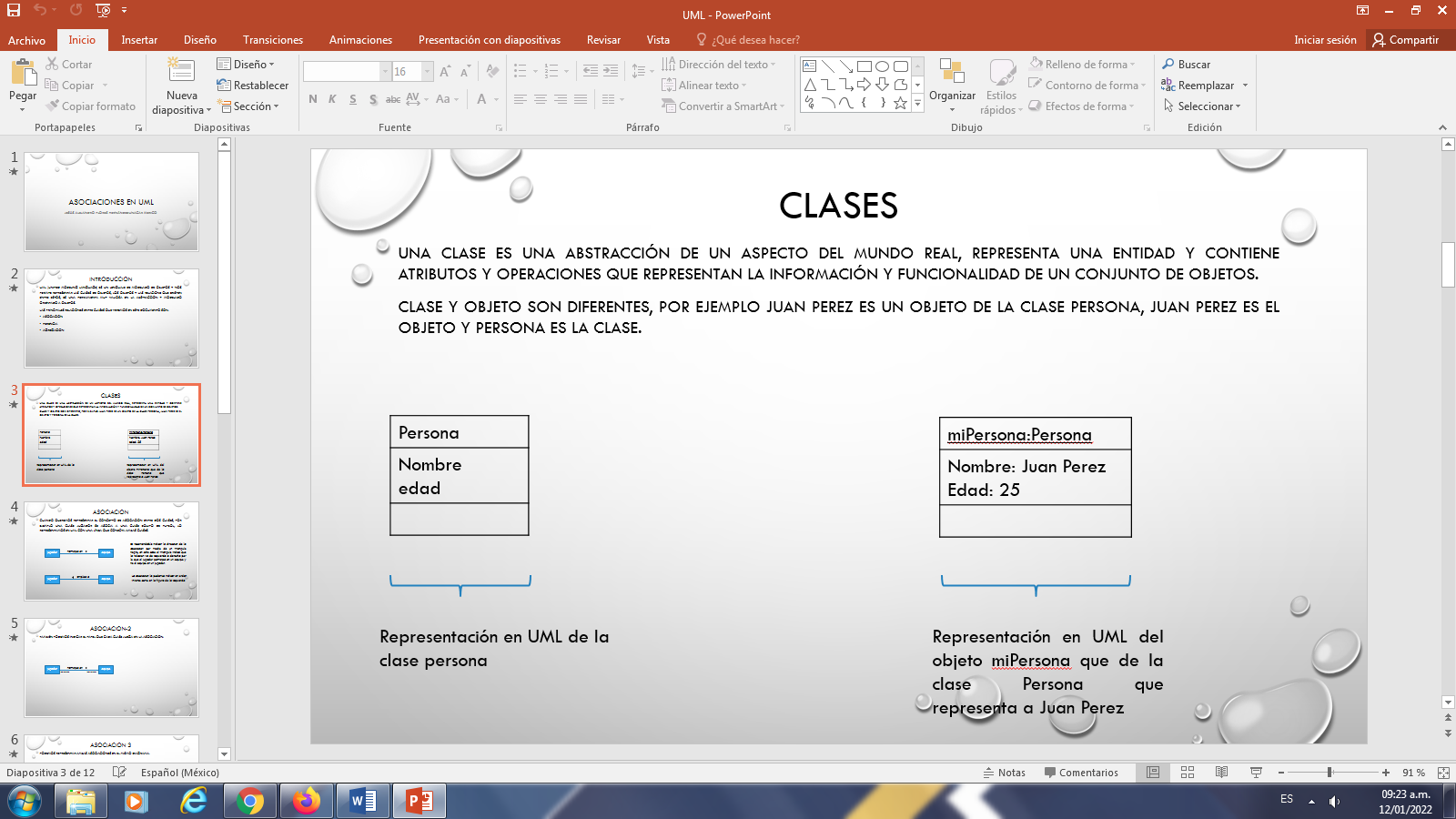
Uml (unified modeling language) Es un lenguaje de modelado de objetos y nos permite representar las clases de objetos, los objetos y las relacione que existen entre estos, es una herramienta muy valiosa en la abstracción y modelado orientado a objetos.

Las principales relaciones entre clases que veremos en este documento son:

* Asociación
* Herencia
* Agregación

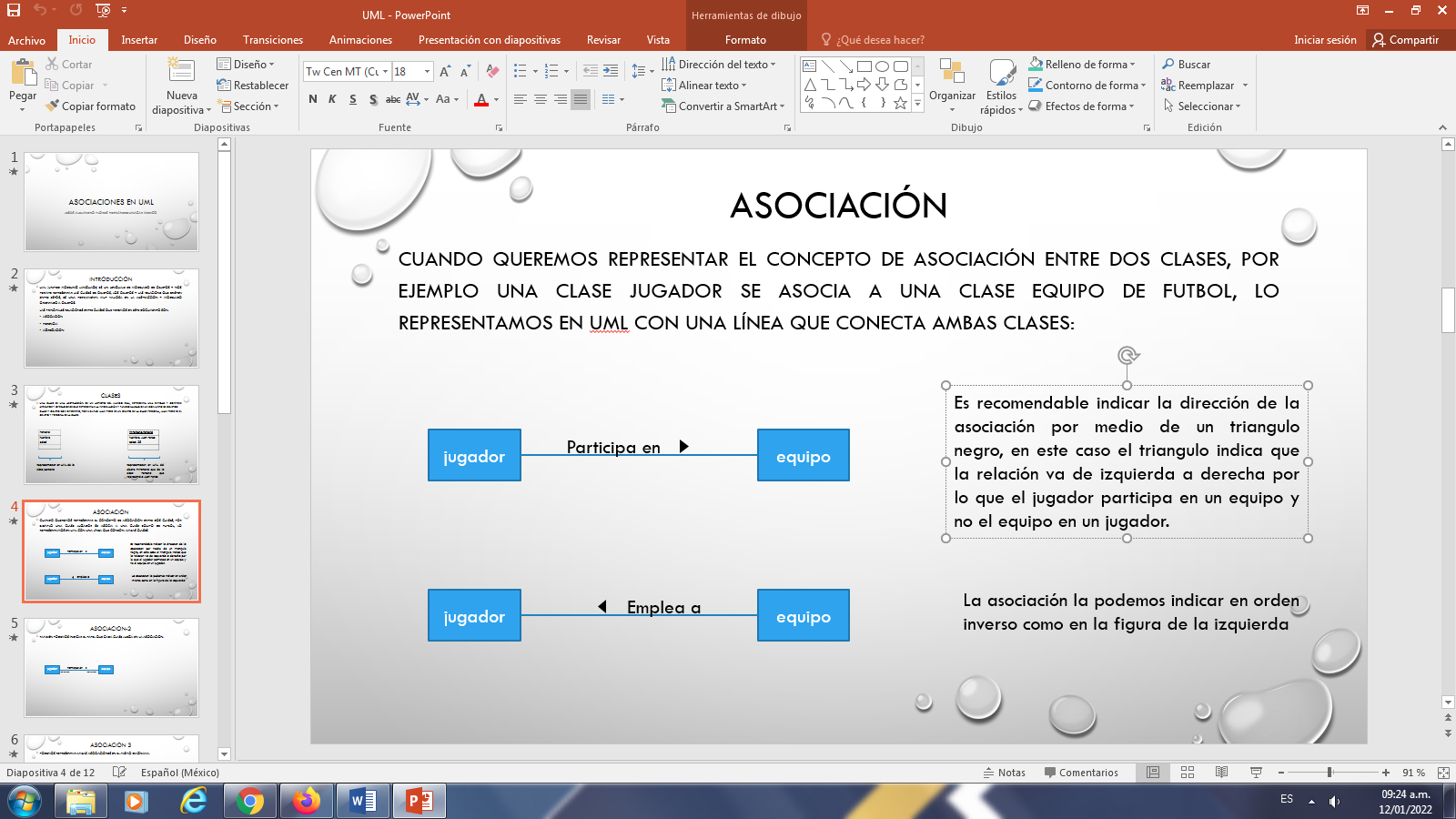
Una clase es una abstracción de un aspecto del mundo real, representa una entidad y contiene atributos y operaciones que representan la información y funcionalidad de un conjunto de objetos.

Clase y objeto son diferentes, por ejemplo, juan PEREZ es un objeto de la clase persona, juan PEREZ es el objeto y persona es la clase.



# Asociación

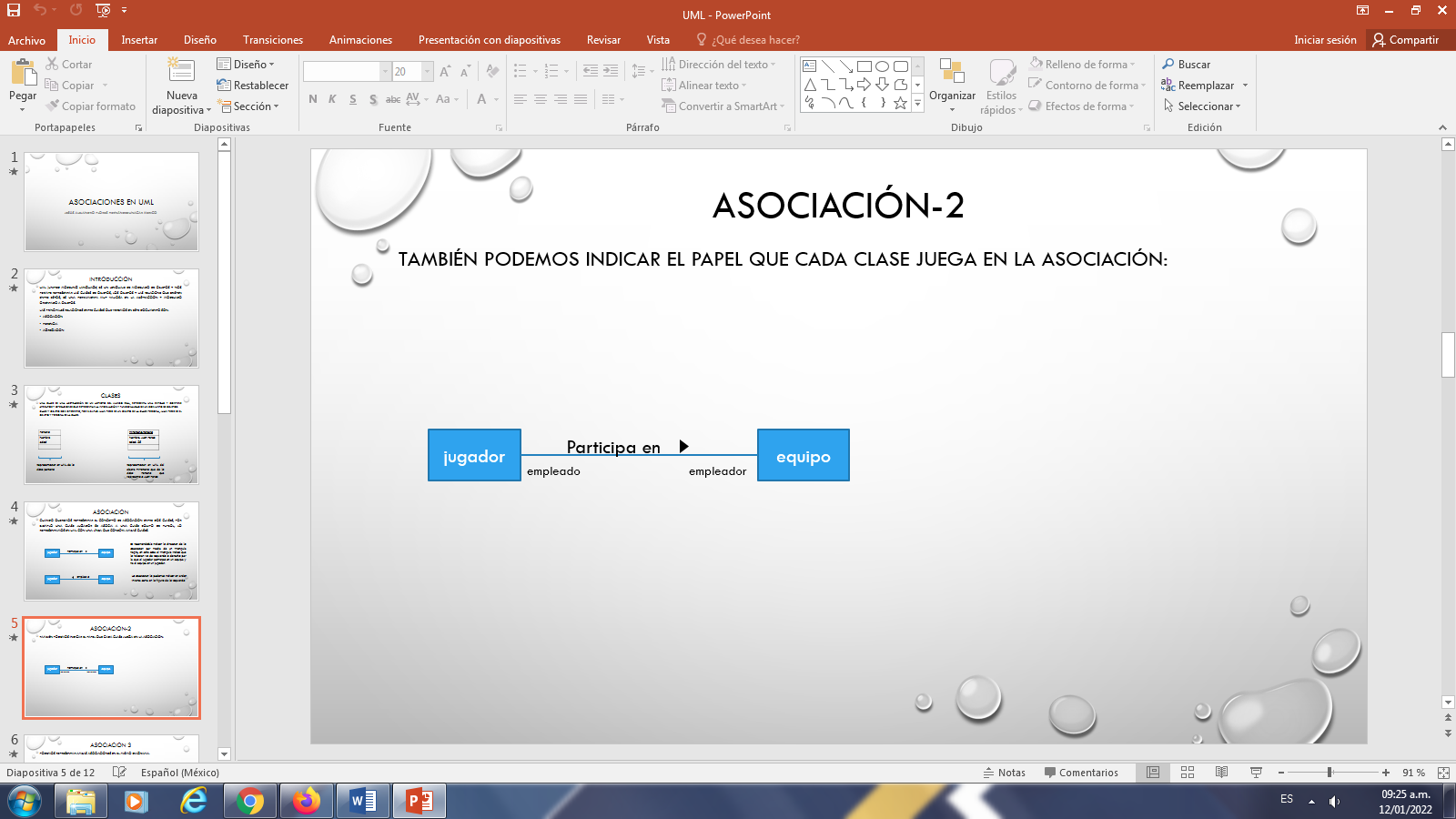
Cuando queremos representar el concepto de asociación entre dos clases, por ejemplo, una clase jugador se asocia a una clase equipo de futbol, lo representamos en UML con una línea que conecta ambas clases:



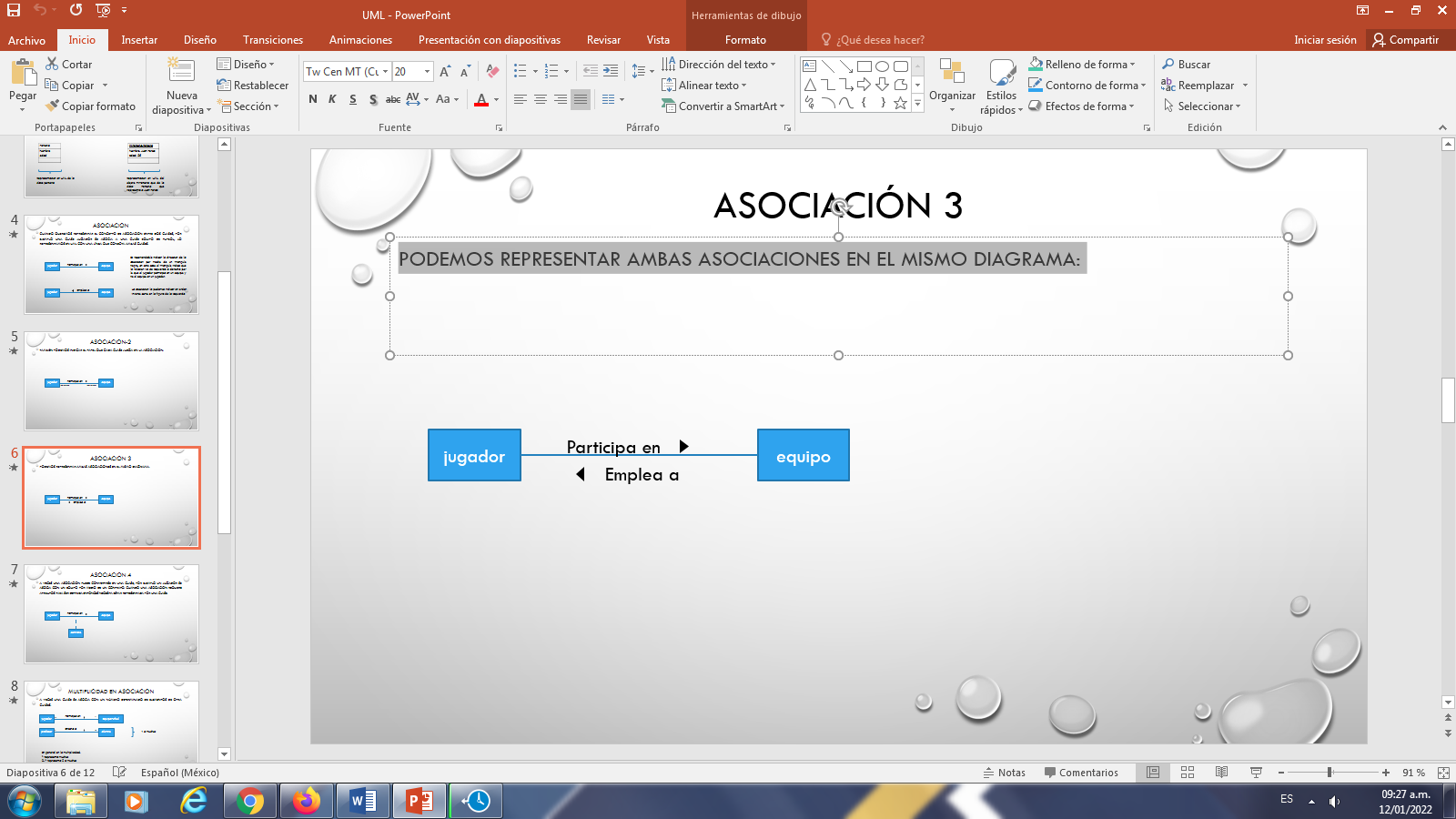
También podemos indicar el papel que cada clase juega en la asociación:

Podemos representar ambas asociaciones en el mismo diagrama:

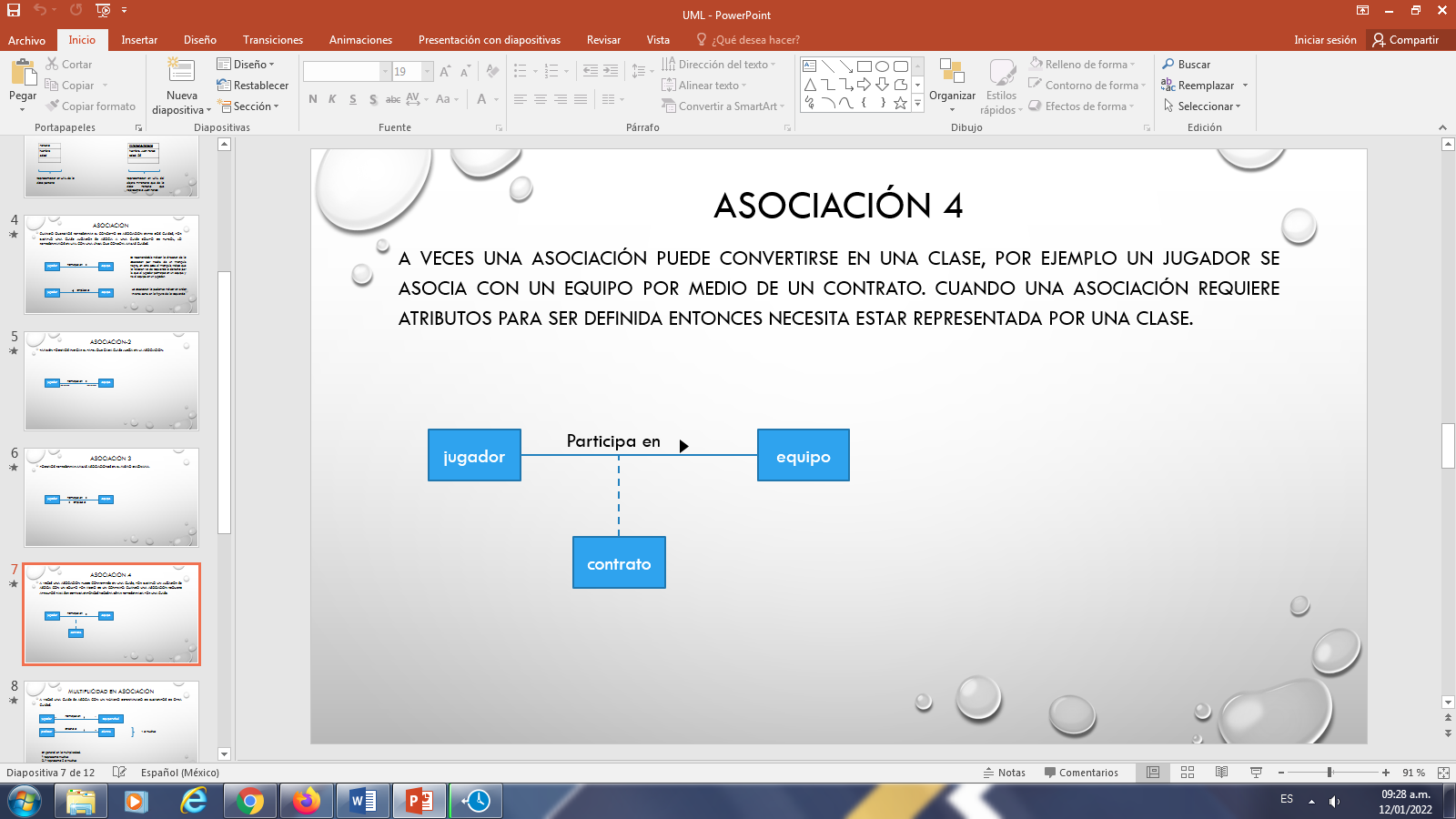
Podemos representar ambas asociaciones en el mismo diagrama:



Podemos representar ambas asociaciones en el mismo diagrama:

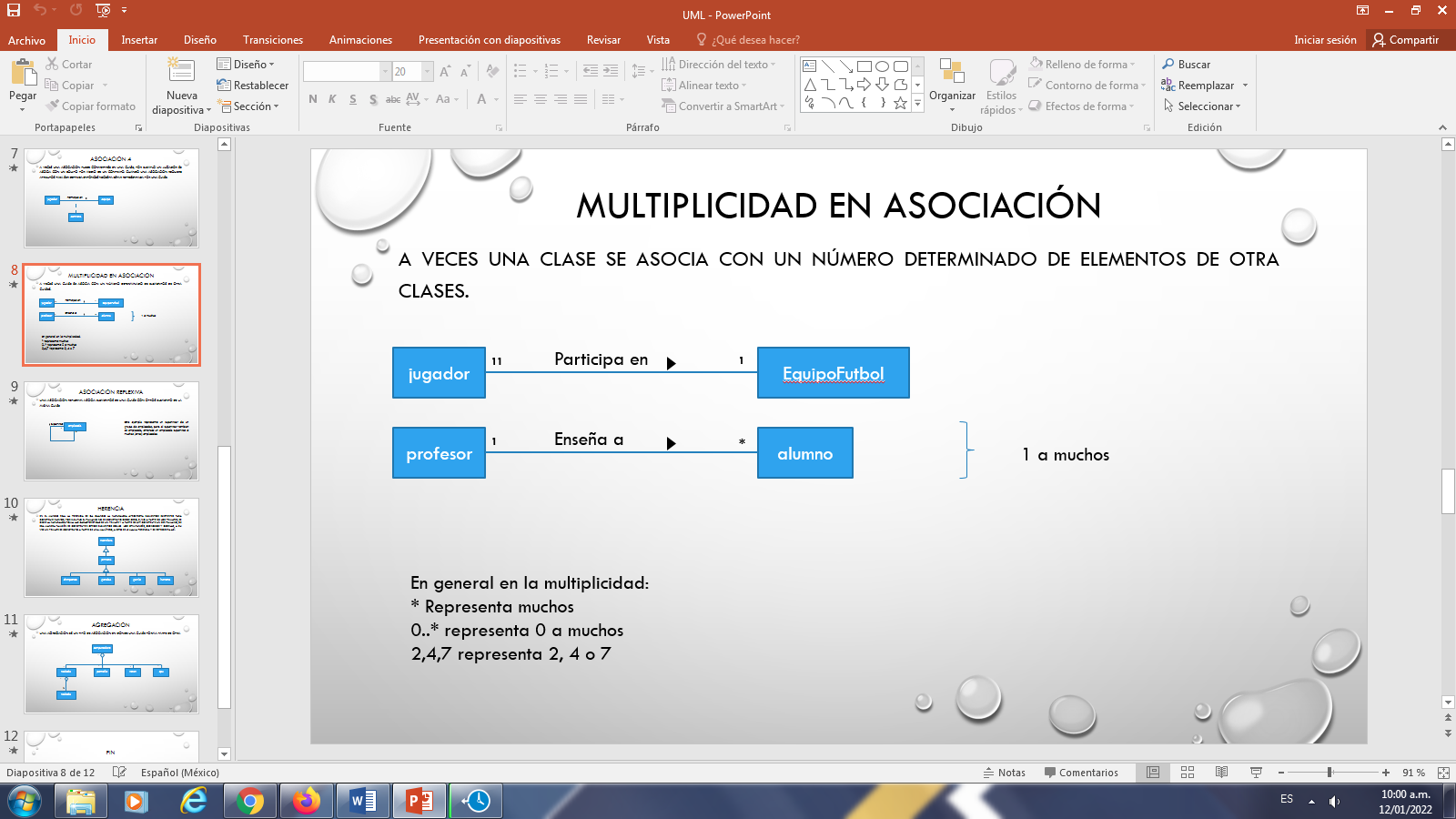


A veces una asociación puede convertirse en una clase, por ejemplo, un jugador se asocia con un equipo por medio de un contrato. Cuando una asociación requiere atributos para ser definida entonces necesita estar representada por una clase.



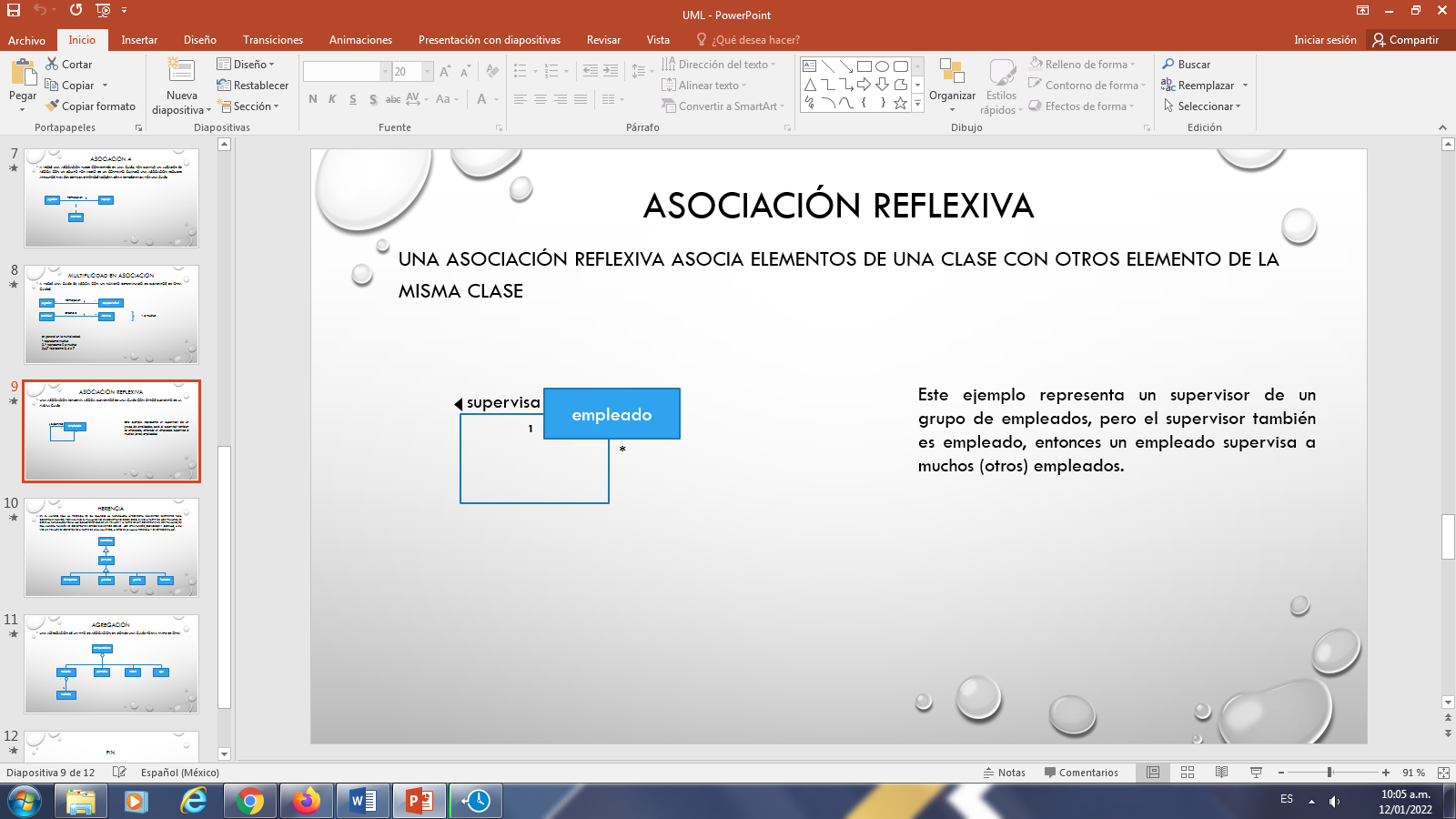
# Multiplicidad en asociaciones

A veces Una clase se asocia con un número determinado de elementos de otras clases.



# Asociación reflexiva

Esta asociación, asocia elementos de una clase con otros elementos de la misma clase.



# Agregación

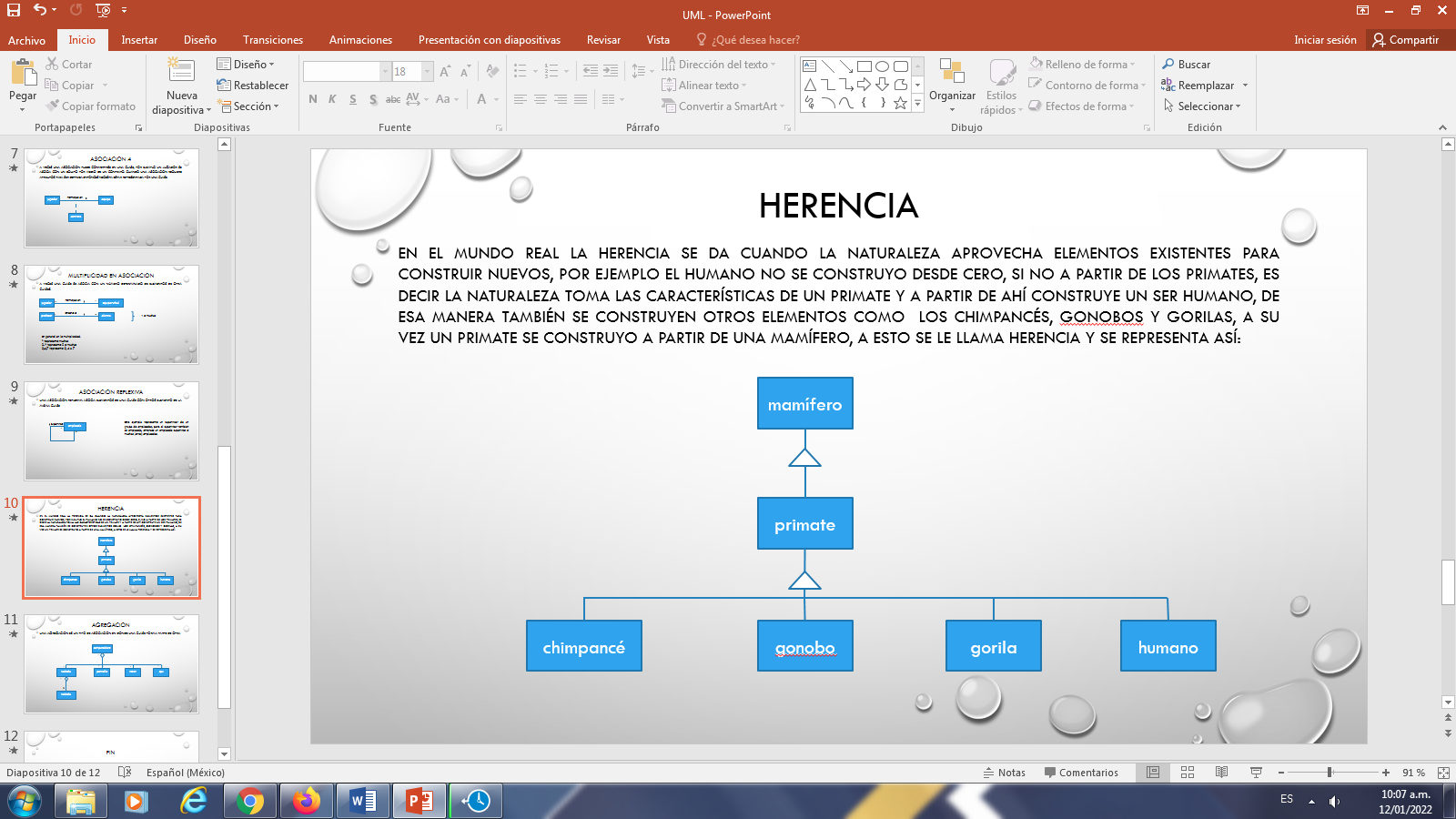
Una agregación es un tipo de asociación en donde una clase forma parte de otra



Aquí decimos que una computadora se compone de un teclado, una pantalla un ratón, un cpu, y que un teclado se compone de muchas teclas.

# Herencia

En el mundo real la herencia se da cuando la naturaleza aprovecha elementos existentes para construir nuevos, por ejemplo, el humano no se construyó desde cero, si no a partir de los primates, es decir la naturaleza toma las características de un primate y a partir de ahí construye un ser humano, de esa manera también se construyen otros elementos como los chimpancés, gonobos y gorilas, a su vez un primate se construyó a partir de un mamífero, a esto se le llama herencia y se representa así:

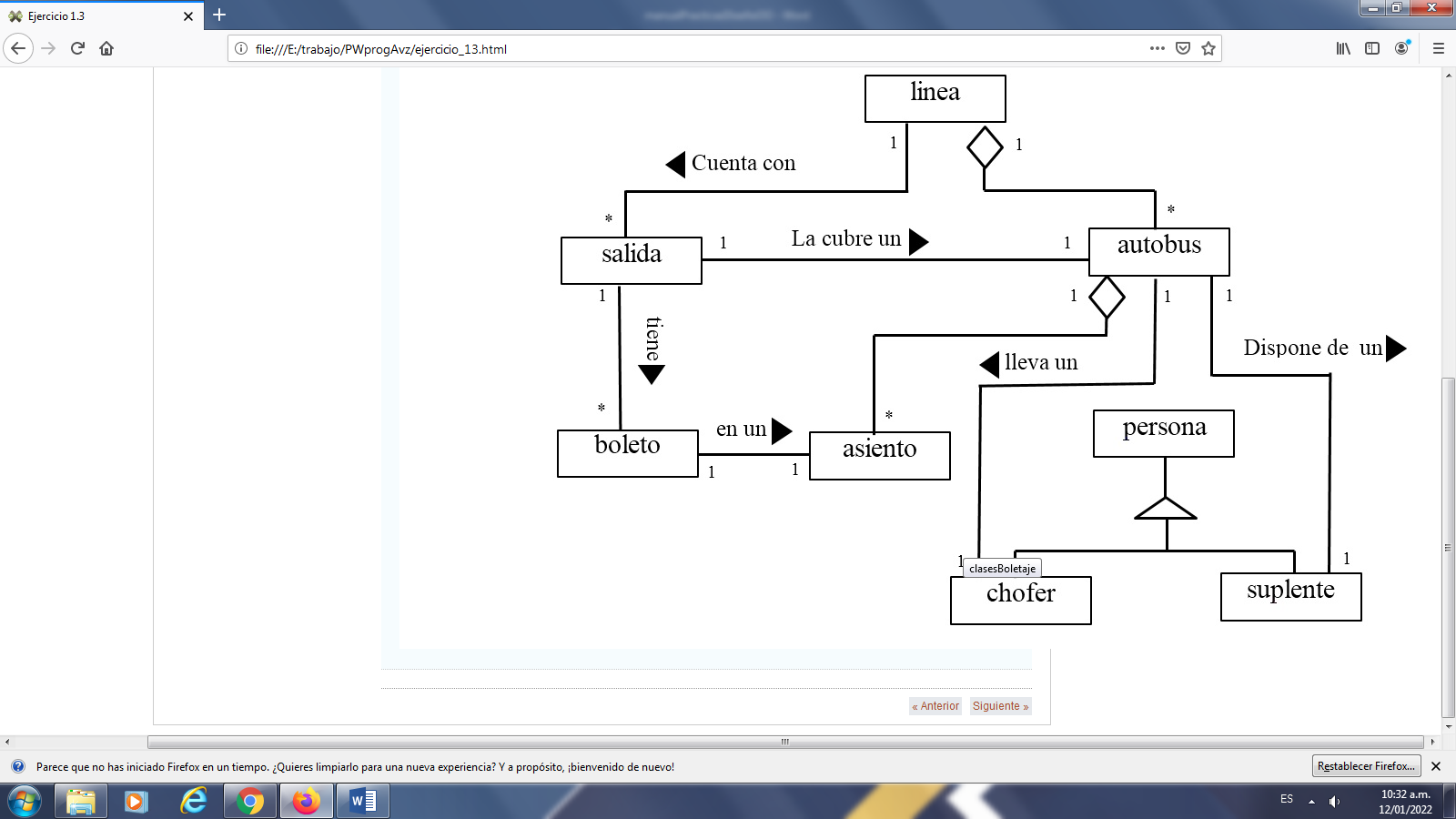


# Ejercicio 5 Modelado UML

Diseñe el modelo de clases para la siguiente narrativa:

La línea de autobuses "The Totoler’s Bus Company Incorporation" cubre la ruta de Cd del Carmen a Champotón sin escalas, se desea la construcción de un sistema cliente servidor que maneje el boletaje de Carmen a Champotón, para esto la línea cuenta con 5 salidas diarias en distintos horarios, para cada salida se tiene un camión asignado por cada horario, y un chofer asignado a cada camión y un chofer suplente en caso de descanso o vacaciones, cada autobús tiene un total de 40 asientos y se desea que el sistema controle la venta de boletos.

Solución:



# Ejercicio 6 Modelado UML

Diseñe el modelo de clases para la siguiente narrativa:

Suponga que queremos modelar los elementos presentes en el control de cursos se la facultad ciencias de la información, sabemos que la facultad se compone de un conjunto de carreras, un conjunto de alumnos y un conjunto de docentes, que las carreras tienen un gestor cada una y una cantidad de alumnos cada carrera, que cada alumno tiene un conjunto de materias aprobadas y un conjunto de materias por aprobar, que un alumno lleva un conjunto de cursos en el semestre, el curso indica que grupo de alumnos tomo el curso, que profesor lo dio y en que aula, también que cada alumno tiene una calificación por cada curso tomado.

Solución:

