Construcción de software y toma de decisiones

TC2005B

Dr. Esteban Castillo Juarez
ITESM, Campus Santa Fe


esteban.castillojz@tec.mx

Tecnológico de Monterrey

Agenda

- UML
- UML y SCRUM
- Diagramas de estructura
  - Diagrama de clases
  - Diagrama entidad-relación
  - Diagrama de objetos
  - Diagrama de componentes
- Diagramas de comportamiento
  - Diagrama de casos de uso

Tecnológico de Monterrey

UML

- El lenguaje de modelado unificado (UML; Unified Modeling Language) es un estándar para la representación visual de objetos, estados y procesos dentro de un sistema.

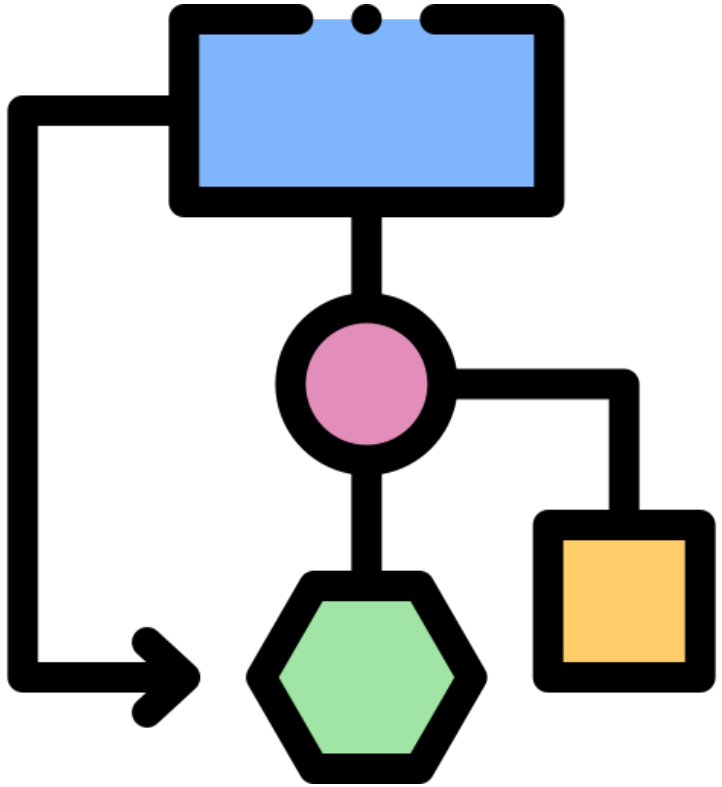
- Por un lado, el lenguaje de modelado puede servir de modelo para un proyecto y garantizar así una arquitectura de información estructurada; por el otro, ayuda a los desarrolladores a presentar la descripción del sistema de una manera que sea comprensible para quienes están fuera del campo.

- UML se utiliza en el diseño de componentes de software (bases de datos, componentes web, etc.) así como en su planificación y posterior despliegue.

UML

- UML proporciona un estándar de representación de software el cual hace que el proceso de desarrollo sea mas simple e intuitivo entre desarrolladores.

- UML puede verse como un puente entre el desarrollador y el cliente/dueño de un sistema de software, donde el primero utiliza diagramas estándar para visualizar el estado de un producto y el segundo puede ver y abstraer sin ningún conocimiento el progreso de un sistema.



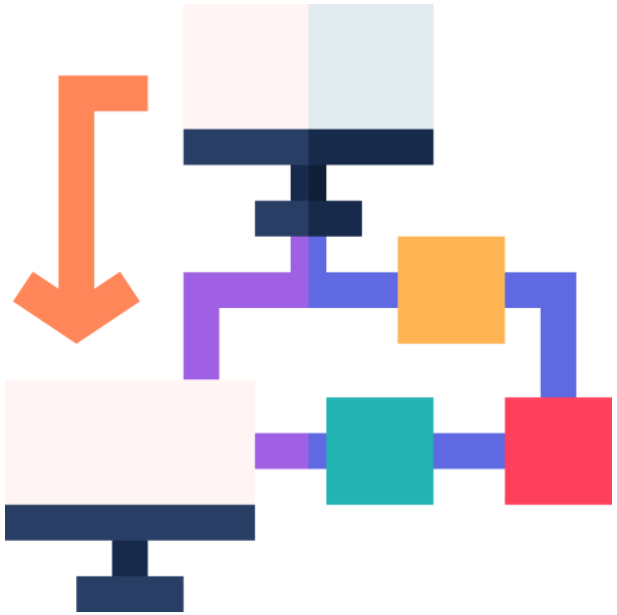
Tecnológico de Monterrey

UML

- UML permite comprender, evaluar y criticar el diseño y la viabilidad de un producto de software más rápido que si tuviera que profundizar en el sistema real.

- Al ser un modelo, abstrae y enfatiza los elementos mas relevantes de un sistema de software sin perder la practicidad de representar todo en un lenguaje estándar que el desarrollador y el cliente entienden.

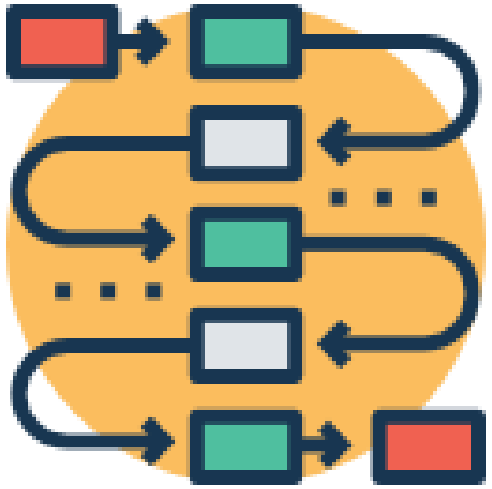
- Por tanto, UML contiene una serie de diagramas y elementos visuales que sirven de apoyo para la documentación y buen entendimiento de un sistema.



Tecnológico de Monterrey

UML

- UML Tiene las siguientes ventajas para el desarrollo e implementación de sistemas de software:
  - **Es un lenguaje formal:** Cada elemento tiene un significado fuertemente definido, por lo que se puede estar seguro de que cuando se modele una faceta particular del sistema, no se malinterpretará.
  - **Es conciso**: Todo el lenguaje se compone de notación simple, directa y ampliamente usada por desarrolladores alrededor del mundo.



Tecnológico de Monterrey

UML

- UML Tiene las siguientes ventajas para el desarrollo e implementación de sistemas de software:
  - **Es autocontenido:** Describe todos los aspectos importantes de un sistema sin ambigüedad ☺.
  - **Es escalable**: Donde sea necesario, el lenguaje es lo suficientemente formal para manejar proyectos de modelado de sistemas masivos, pero también se reduce a proyectos pequeños, evitando la exageración.

Tecnológico de Monterrey

UML

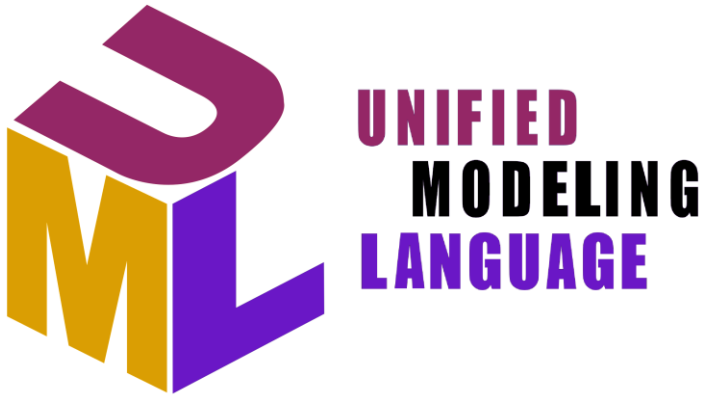
- UML Tiene las siguientes ventajas para el desarrollo e implementación de sistemas de software:
  - **Se basa en lecciones aprendidas:** UML es la culminación de las mejores prácticas en la comunidad de ingeniería de software a lo largo de los últimos 20 años.
  - **Es el estándar**: UML está controlado por un grupo de contribuciones activas (expertos en empresa y academia). Un estándar garantiza la capacidad de transformación e interoperabilidad de UML, lo que significa que no está atado a un producto en particular.

Tecnológico de Monterrey

UML

Resumiendo:

- UML es un lenguaje de modelado formal, el cual tiene una notación simple cuyo significado está bien definido.
- La notación de UML es lo suficientemente pequeña para aprenderla fácilmente y tiene una definición inequívoca del significado de la notación



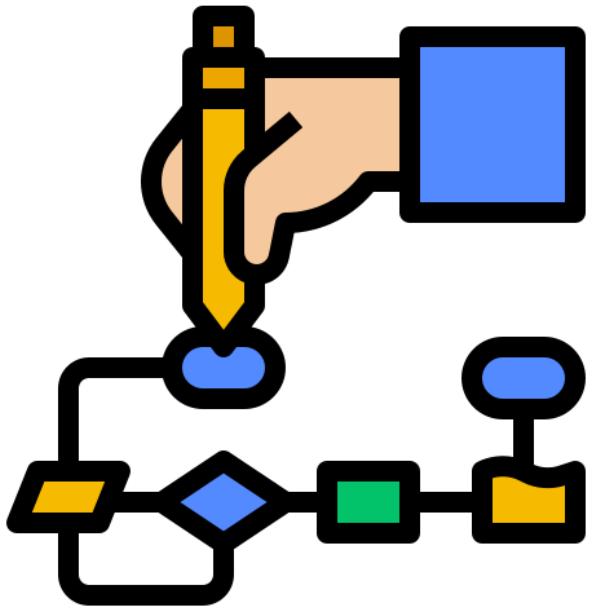
Tecnológico de Monterrey

UML

- Muchos recién llegados a UML se enfocan en los diferentes tipos de diagramas usados para modelar un sistema.

- Es muy fácil suponer que el conjunto de diagramas que se pueden crear son en realidad el modelo de un sistema de software.

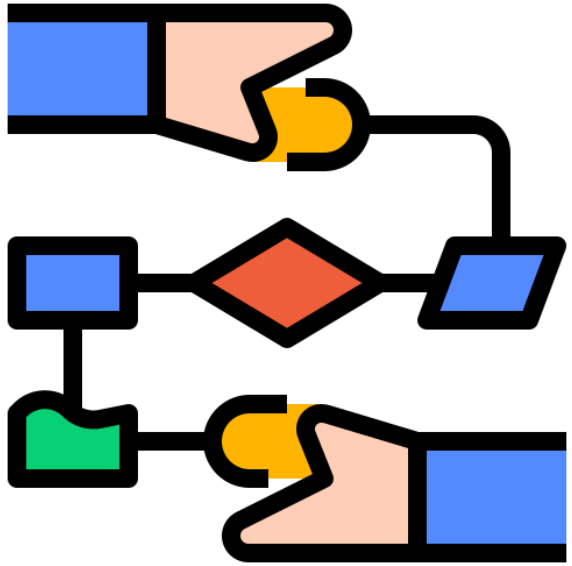
- Lo anterior, es un error fácil de cometer, pero el modelado UML no se trata solo de diagramas; se trata de capturar su sistema como un todo. **Los diagramas son en realidad ventanas hacia su modelo o forma de crear software.**



Tecnológico de Monterrey

UML

- Un diagrama particular le mostrará algunas partes de su modelo, pero no necesariamente todo.
- Lo anterior tiene sentido, ya que no desea un diagrama que muestre todo su modelo a la vez; desea poder dividir el contenido de su modelo en varios diagramas. Sin embargo, no todo en su modelo necesita existir en un diagrama para que sea parte de su modelo.
- La colección de todos los elementos que describen su sistema, incluidas sus conexiones entre sí, conforman un modelo de software.

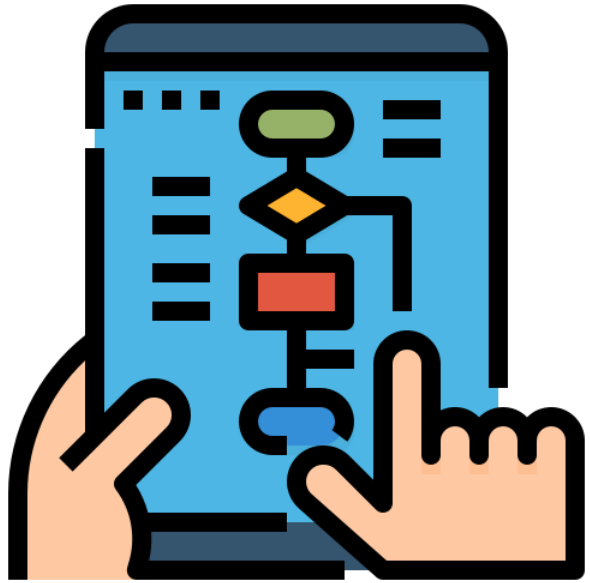


Tecnológico de Monterrey

UML

- Entonces, la próxima vez que use UML para trabajar, vale la pena recordar que lo que está manipulando es una vista del contenido de su sistema de software a través de un modelo ejemplificado con diagramas.

- Puede cambiar elementos de su modelo dentro del diagrama, pero el diagrama en sí no es el modelo, es solo una forma útil de presentar una pequeña parte de la información que contiene su sistema.



Tecnológico de Monterrey

UML

How the UML diagram describes the software

How the code is actually written

Tecnológico de Monterrey

UML y SCRUM

- Cuando usa UML para modelar un sistema de software, el "grado de UML" que aplica está parcialmente influenciado por la metodología de desarrollo de software (metodologías tradicionales vs agiles).

- En el caso de las metodologías agiles, UML se puede utilizar para explicar a los clientes como funciona/puede-funcionar un software, incentivando la participación de los clientes/usuarios y su interacción con el equipo de desarrollo.

Tecnológico de Monterrey

UML y SCRUM

- Esta comunicación entre los grupos (así como entre UML y SCRUM) da lugar a un software que responde mejor a las necesidades del cliente. En caso de encontrarse un conflicto o malentendido sobre un elemento, este puede ser corregido durante la siguiente interacción o versión.

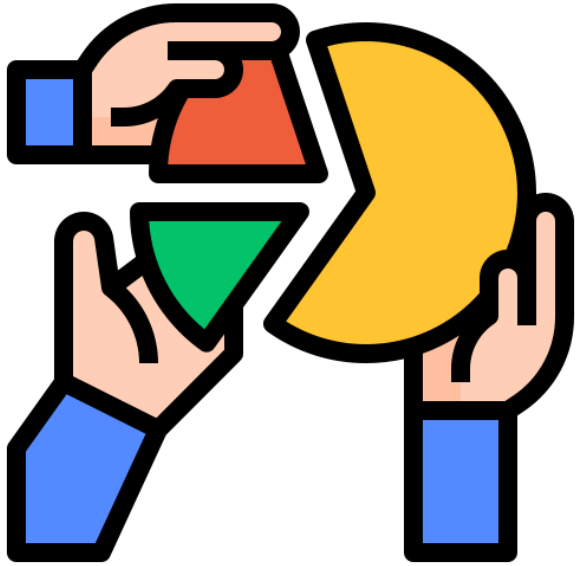
- UML constituye una referencia común. Los modelos ayudan al equipo a trabajar juntos, ya que pueden ser utilizados por los miembros del equipo para visualizar, compartir y discutir ideas y soluciones a los problemas.

Tecnológico de Monterrey

UML y SCRUM

- Los modelos de UML pueden ayudar a los nuevos miembros del equipo a entender la visión de alto nivel y participar rápidamente, dado que una representación visual puede ser más fácil y más clara de entender y analizar que el texto. **Una imagen vale más que mil palabras.**

- Por supuesto que si nos enfocamos en los modelos y la documentación, no estamos siendo ágiles; pero si usamos los modelos para enfocarnos en el problema a resolver, entonces son artefactos valiosos.

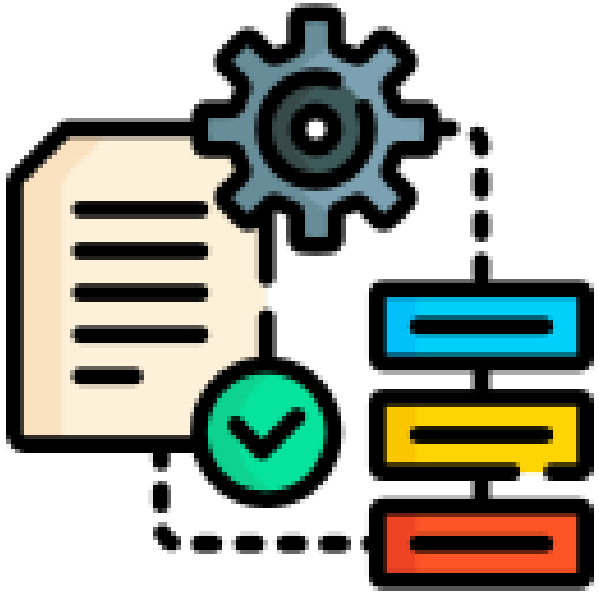


Tecnológico de Monterrey

UML y SCRUM

- Los modelos pueden utilizarse como fuente de documentación actualizada, dado que la arquitectura general de un sistema usualmente no cambia a lo largo de varias iteraciones. Por ello los modelos pueden reutilizarse en fases posteriores.

- Por otro lado los equipos de desarrollo enfrentan otros problemas al utilizar herramientas de modelado de escritorio: Las herramientas son bastante complejas de usar, toman mucho tiempo en instalar y configurar, compartir modelos es complicado, etc.



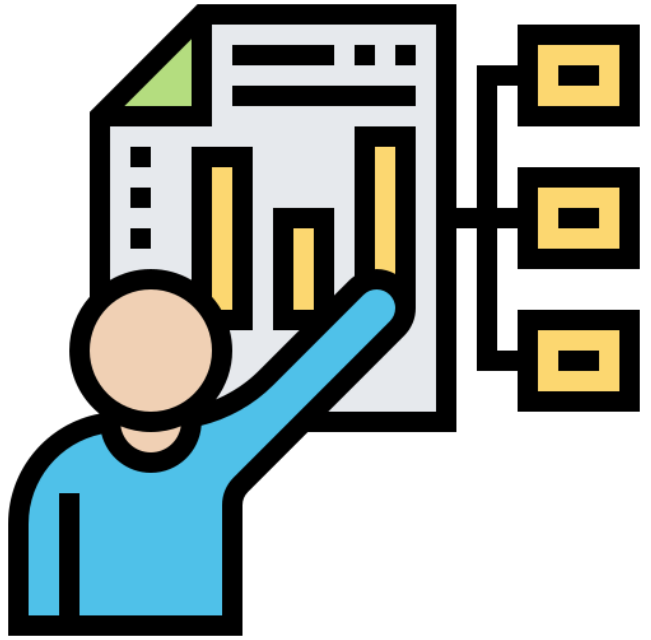
Tecnológico de Monterrey

UML y SCRUM

En resumen:

- Las metodologías de desarrollo tienen como objetivo aportar disciplina y previsibilidad al proceso de creación de software, aumentando las posibilidades de éxito de un proyecto.

- Dado que UML es el lenguaje para modelar software con estándares por defecto, es una parte importante del proceso de desarrollo de software y del entendimiento de desarrolladoras y potenciales clientes.



Tecnológico de Monterrey

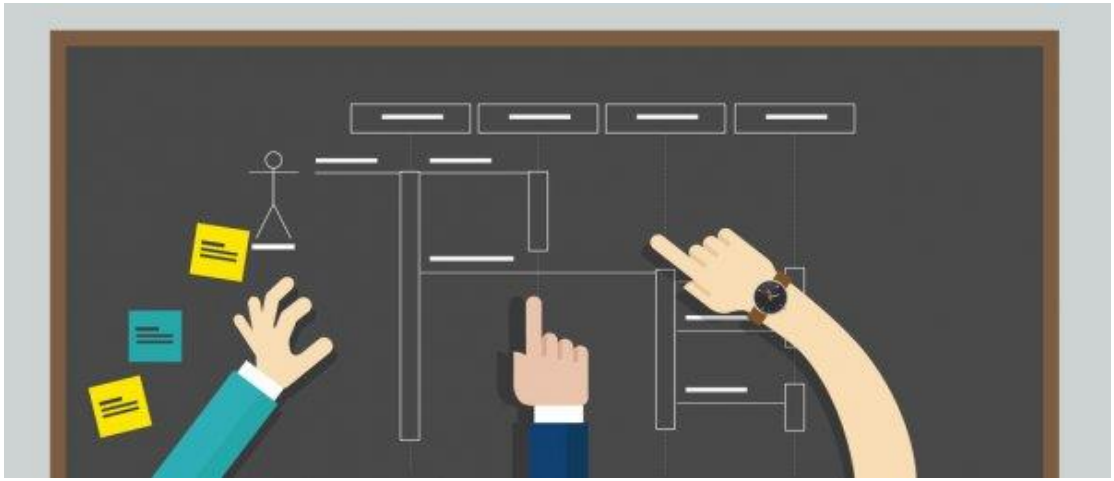
UML y SCRUM

Tecnológico de Monterrey

Diagramas UML

- El lenguaje de modelado UML define 14 tipos de diagramas que se dividen en dos categorías principales.

- Las categorías de **estructura y comportamiento** representan los conceptos básicos representados por los diagramas UML en la creación de un sistema.



UNIFIED MODELING LANGUAGE

Tecnológico de Monterrey

Diagramas de estructura

- Los diagramas de estructura representan los elementos individuales de un sistema. Por lo tanto, son especialmente adecuados para la representación de la arquitectura de software.

- La representación estática no representa un cambio, sino estados y dependencias en un momento determinado. Los elementos individuales u objetos están relacionados entre sí. Por ejemplo, un objeto pertenece a una clase.

- Los diagramas UML de esta categoría representan un sistema completo o la arquitectura de un proceso funcional.

Diagramas de estructura – diagramas de clases

- El diagrama de clases es un diagrama puramente orientado al modelo de Programación Orientado a Objetos (POO), ya que define las plantillas que se utilizarán en la resolución de un problema.

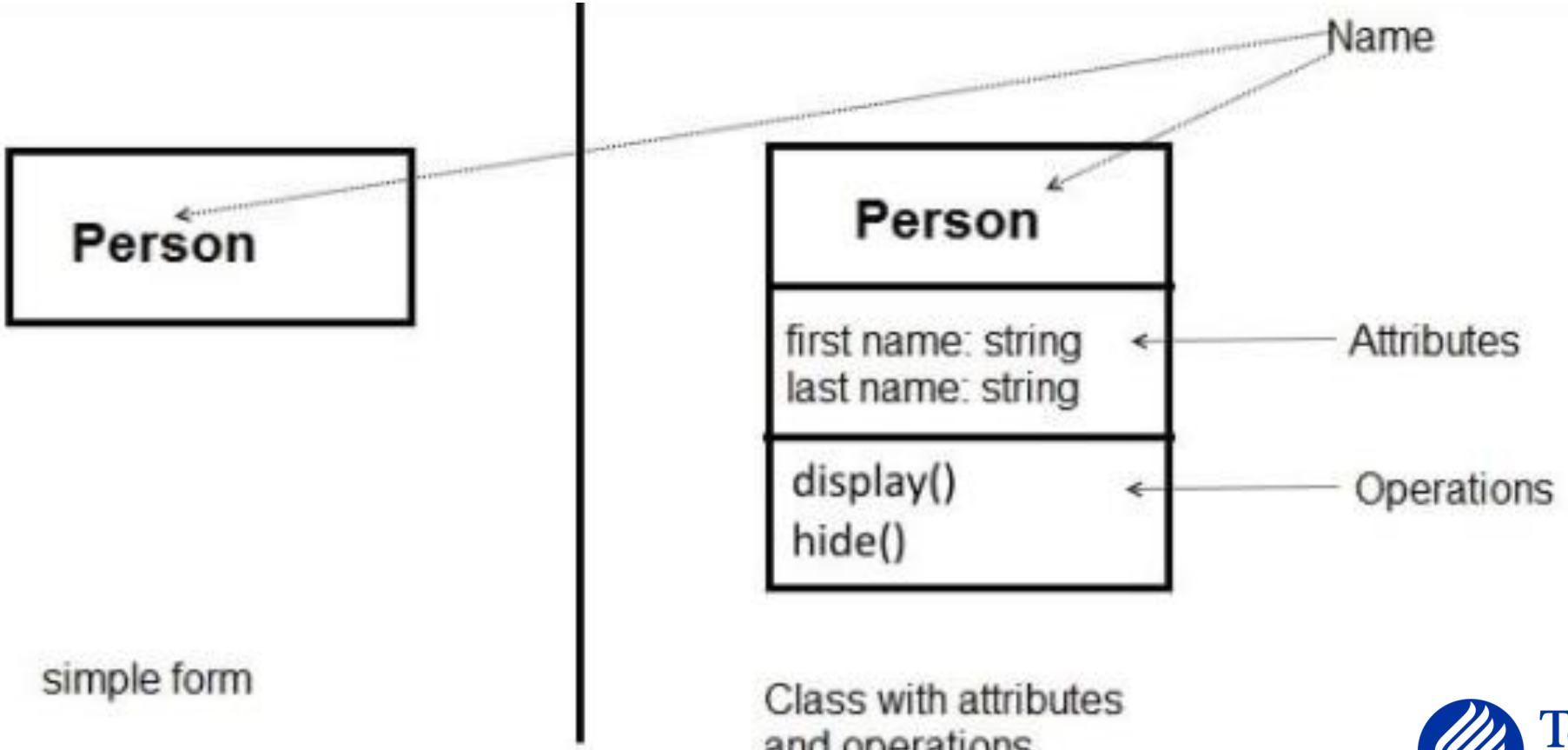
- Se utiliza para representar los elementos que componen un sistema de información desde un punto de vista estático.

- Este diagrama no incluye la forma en la que se comportan a lo largo de la ejecución los distintos objetos de un programa.

Tecnológico de Monterrey

Diagramas de estructura – diagramas de clases

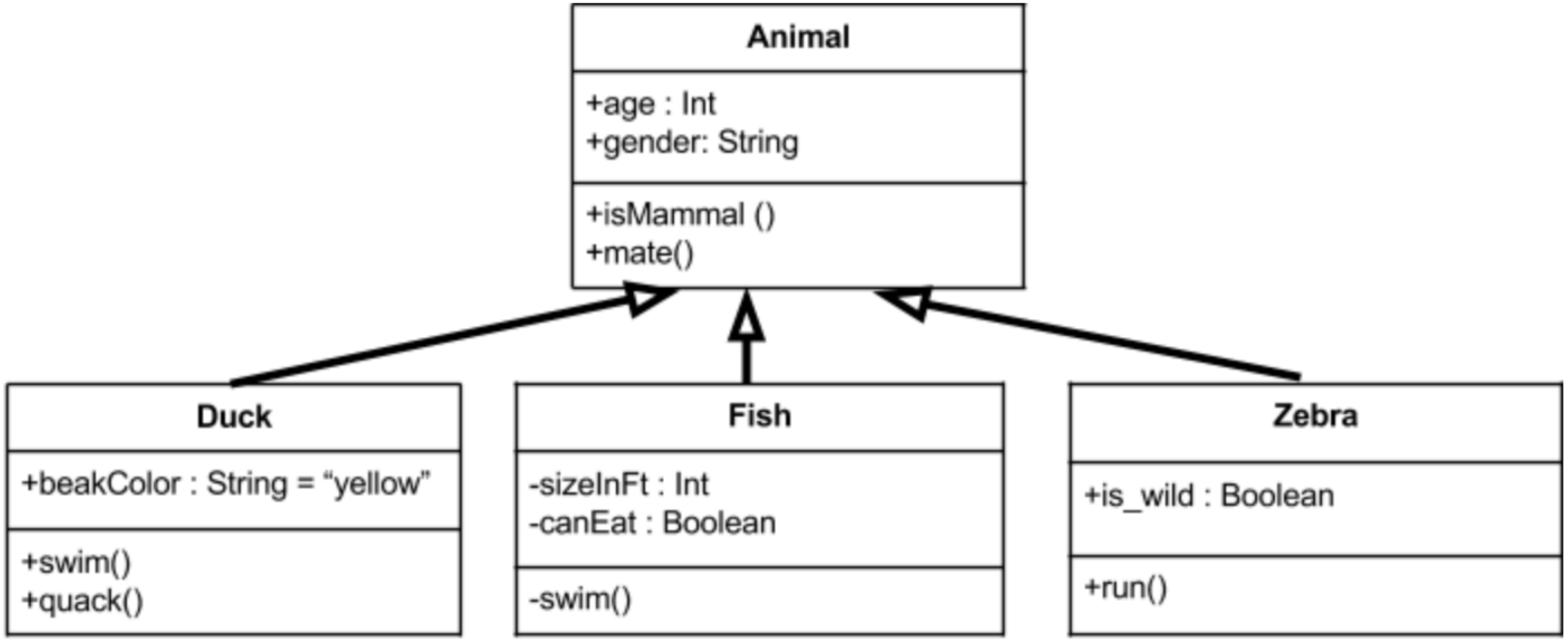
Usado en la Programación Orientada a Objetos (POO).



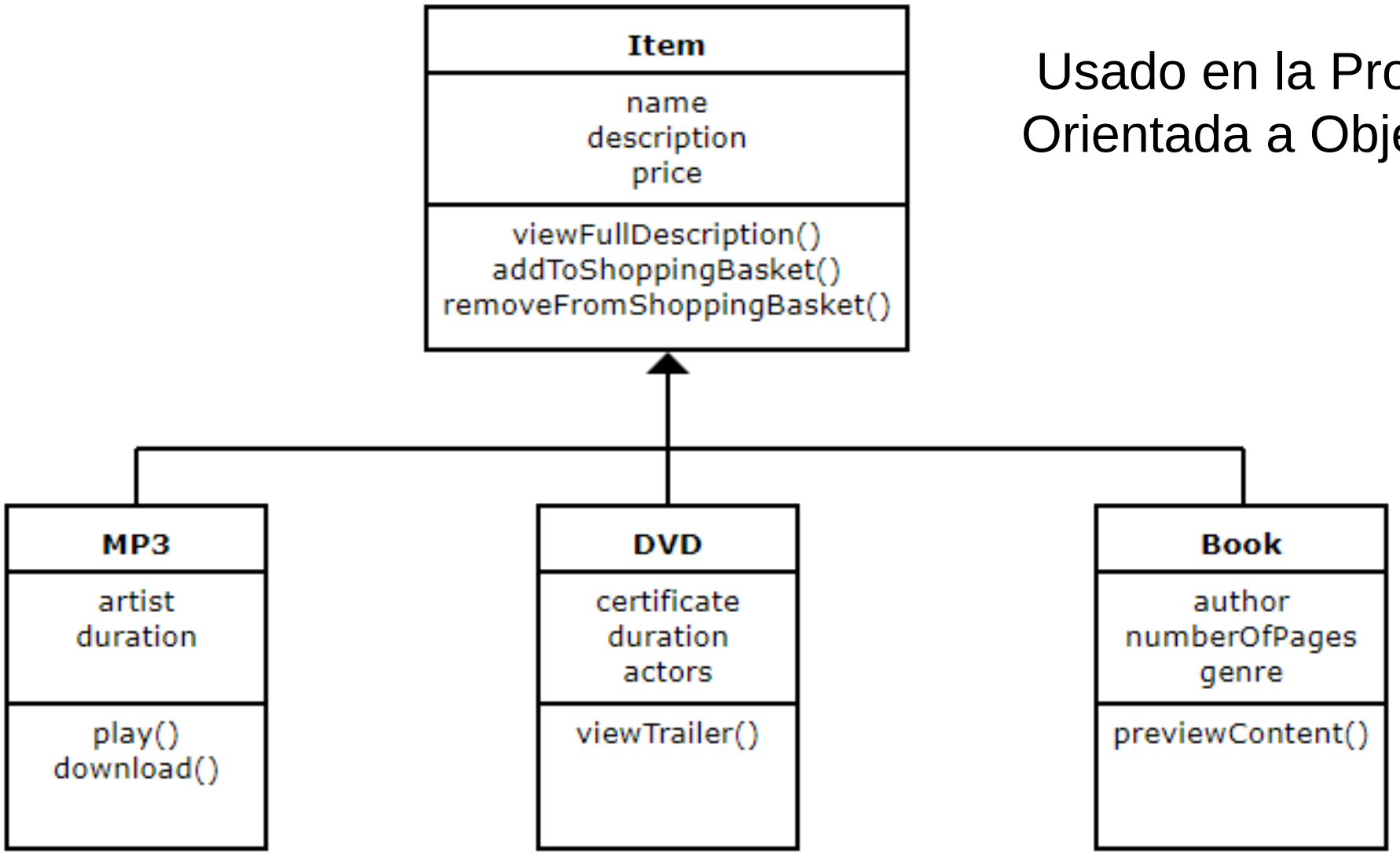
Tecnológico de Monterrey

Diagramas de estructura – diagramas de clases

Usado en la Programación Orientada a Objetos (POO).



Diagramas de estructura – diagramas de clases



Usado en la Programación Orientada a Objetos (POO).

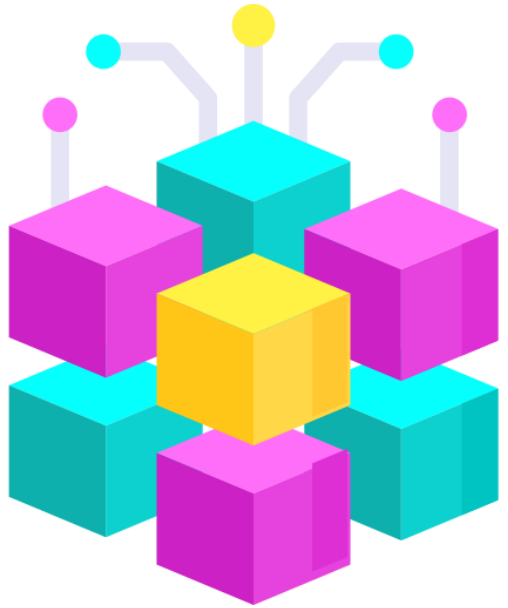
Tecnológico de Monterrey

Diagramas de estructura – diagramas entidad relación

- El modelo entidad-relación es un herramienta para representar de forma abstracta la información que esta presente en una base de datos tradicional.

- Los diagramas entidad-relación ayudan a modelar el componente de representación de datos de un sistema software. La representación de datos, sin embargo, sólo forma parte de un diseño completo de un sistema.

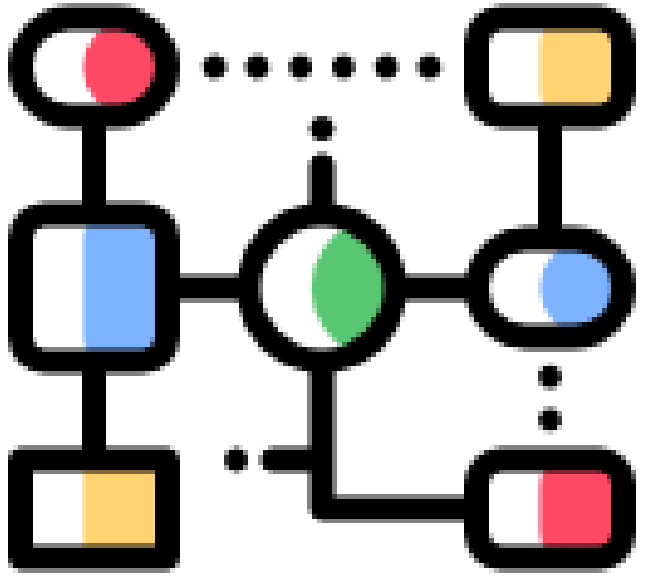
- En corto: Los diagramas entidad-relación muestran el diseño conceptual de aplicaciones de bases de datos.



Tecnológico de Monterrey

Diagramas de estructura – diagramas entidad relación

- Este tipo de diagramas describen las distintas entidades (conceptos) del sistema de información y las relaciones y restricciones existentes entre ellas.
- Se puede decir que estos diagramas se utilizan para la modelización de datos que describe las asociaciones que existen entre las diferentes categorías de información dentro de un sistema de software.
- En otras palabas son usados para entender el dominio del mundo real que se va a modelar.

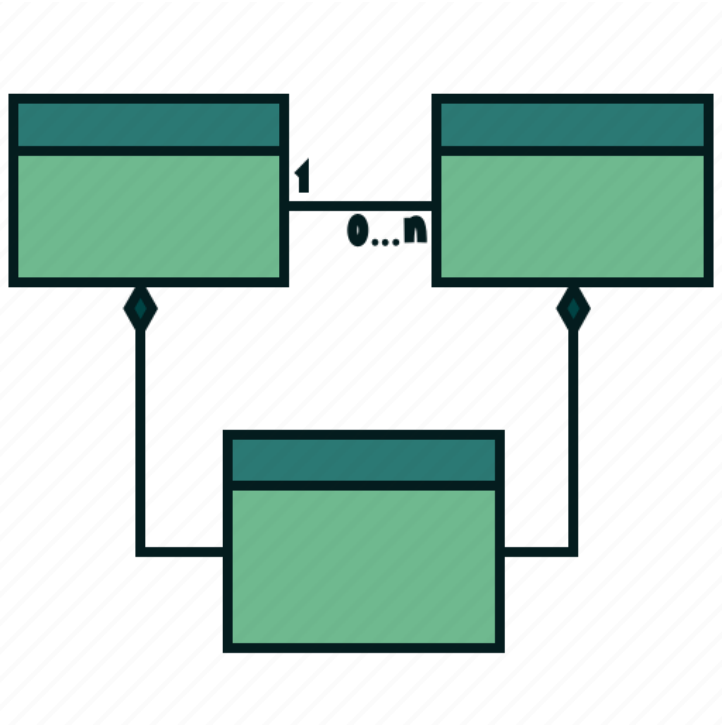


Tecnológico de Monterrey

Diagramas de estructura – diagramas entidad relación

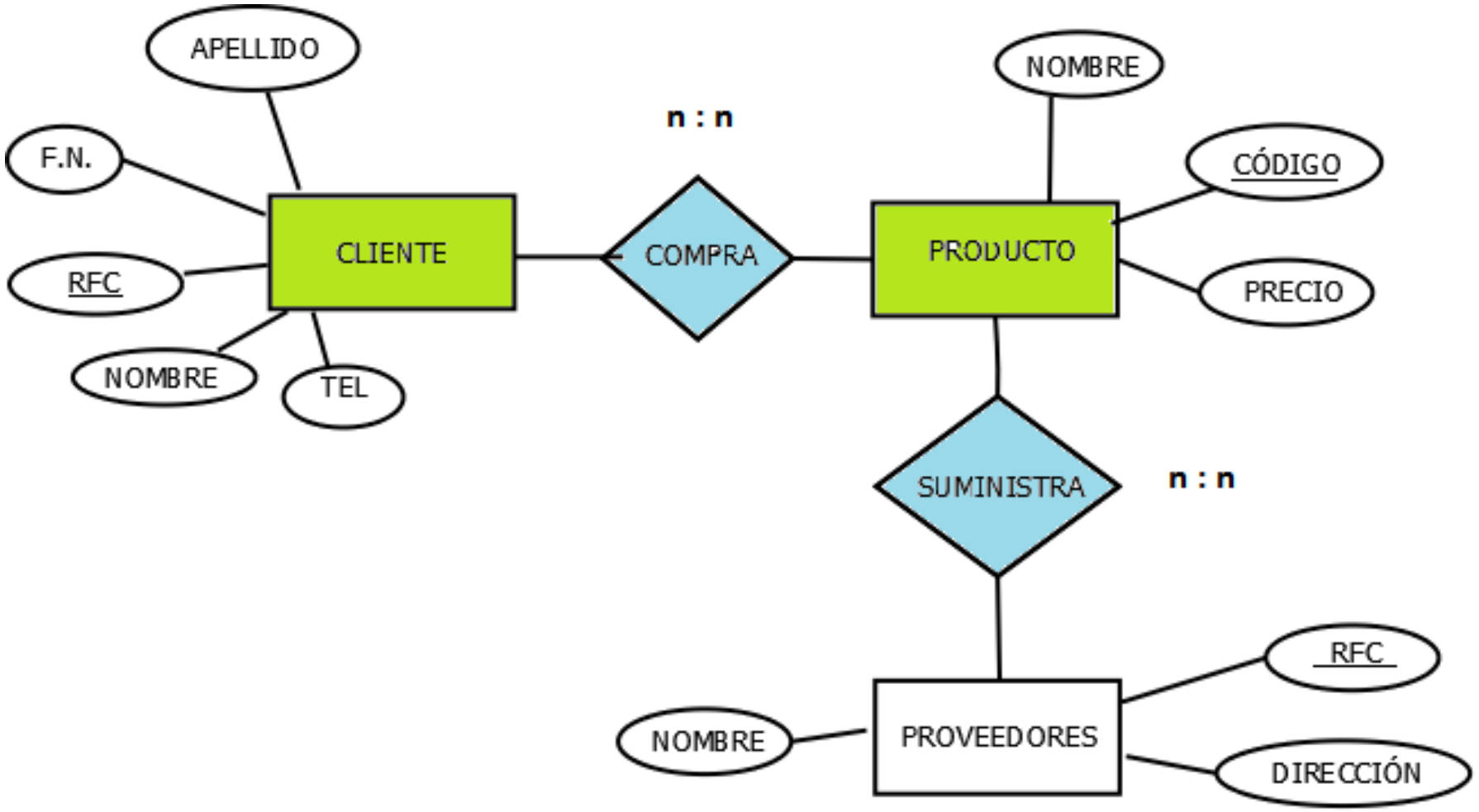
- Los diagramas entidad relación son los que nos permiten crear bases de datos bien especificadas y normalizadas en el contexto de un problema concreto.

- En la parte de bases de datos profundizaremos específicamente en su uso, control y depuración pero de momento este será el modelo lógico de una base de datos.



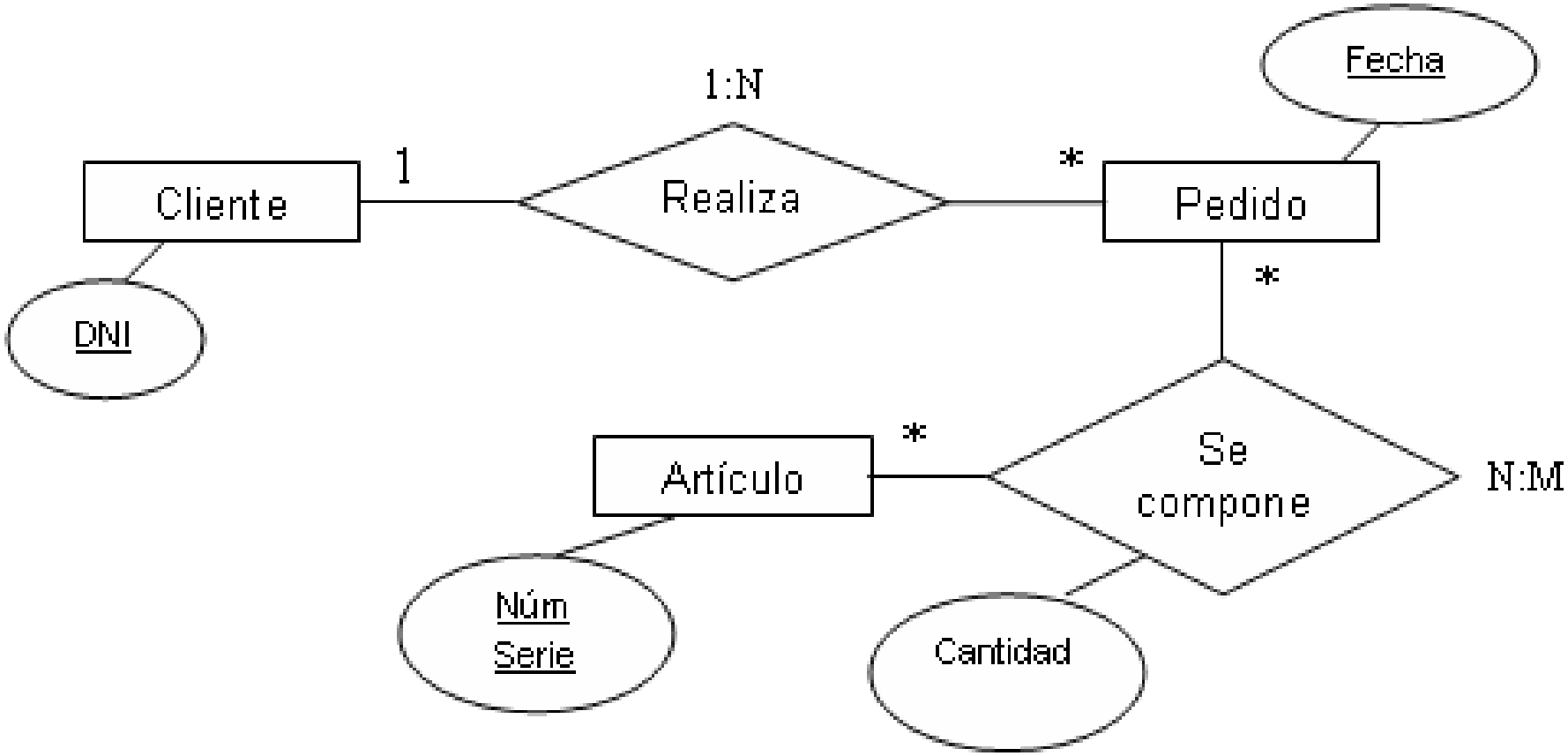
Tecnológico de Monterrey

Diagramas de estructura – diagramas entidad relación



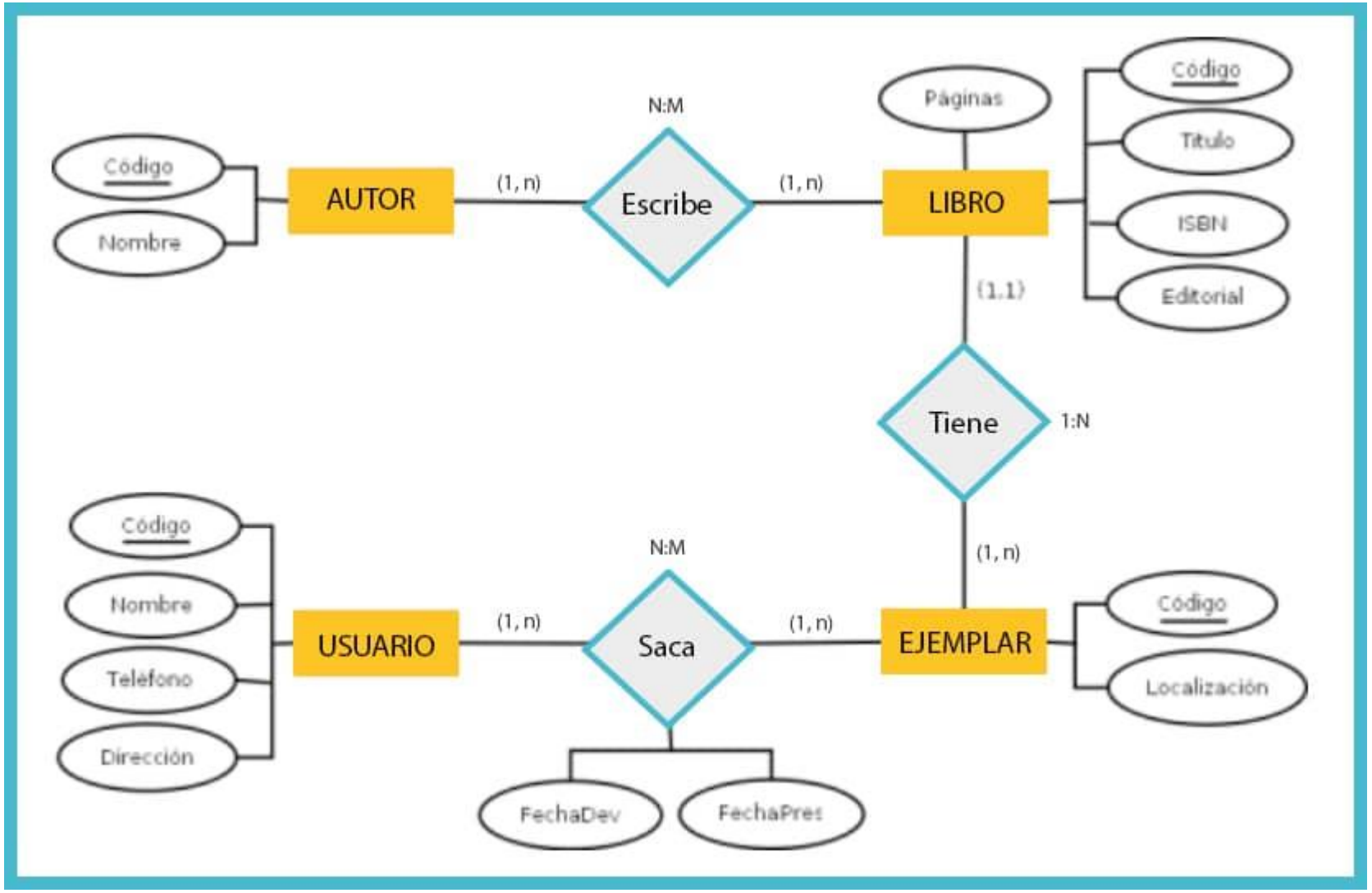
Tecnológico de Monterrey

Diagramas de estructura – diagramas entidad relación



Tecnológico de Monterrey

Diagramas de estructura – diagramas entidad relación



Tecnológico de Monterrey

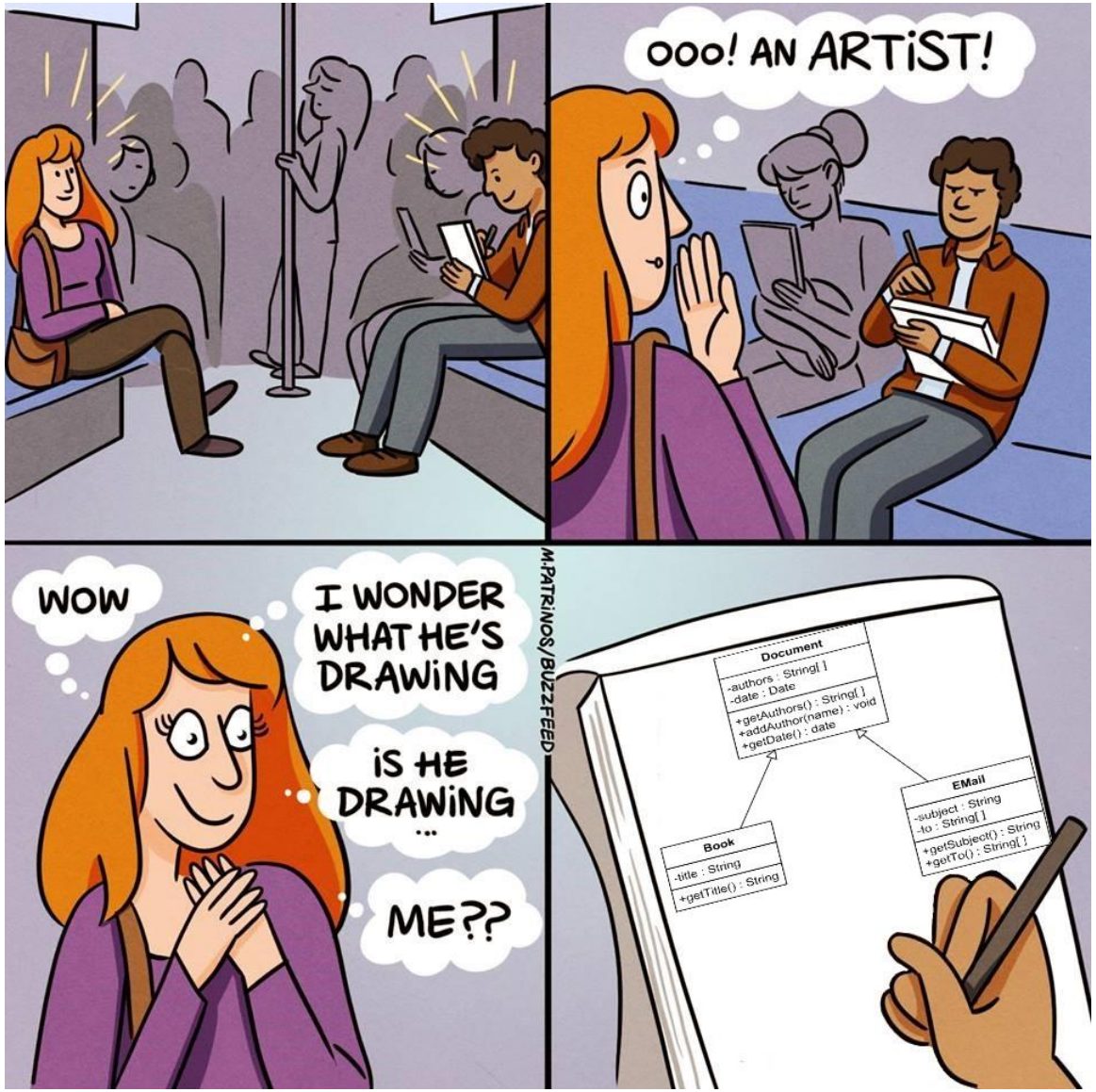
Diagramas de estructura – diagramas entidad relación

COCHE
PK idCoche FK idMotor
Color Longitud Peso N° Puertas

MOTOR
PK idMotor FK idCoche
Potencia Rendimiento Consumo

Tecnológico de Monterrey

Diagramas de estructura – diagramas entidad relación

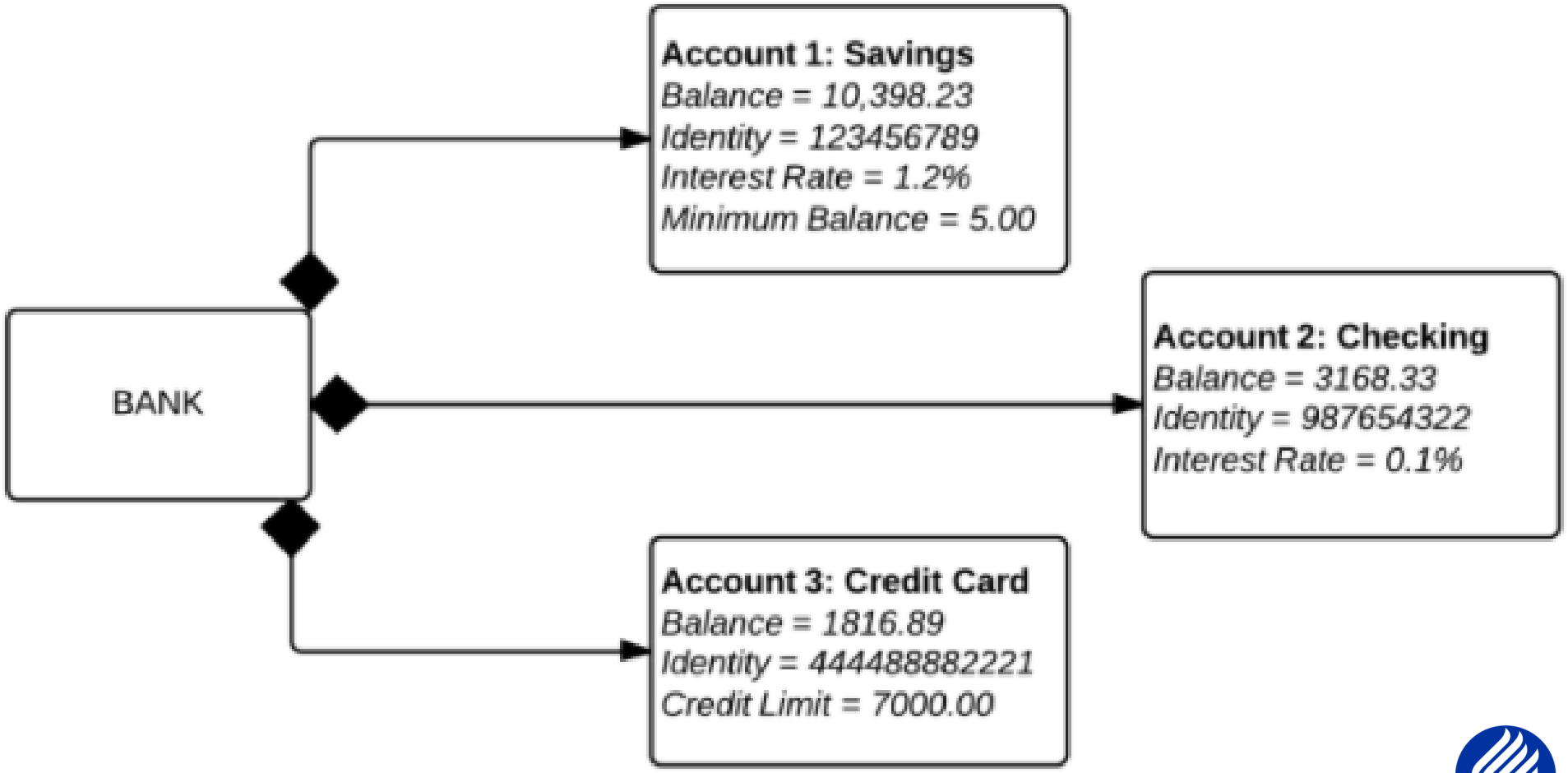


Tecnológico de Monterrey

Diagramas de estructura – diagrama de objetos

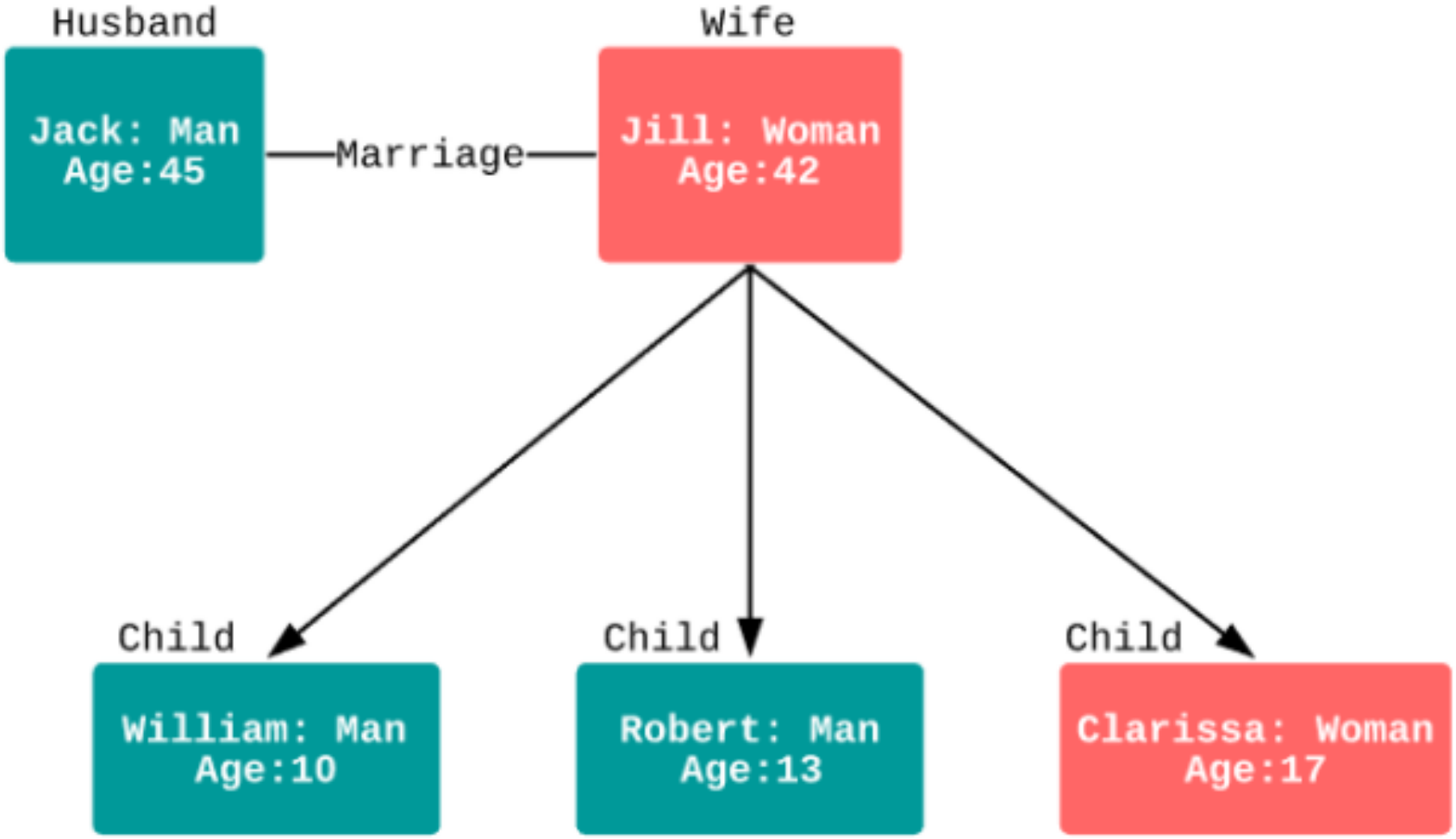
- Un diagrama de objetos UML representa una instancia específica de un diagrama de clases en un momento determinado en el tiempo.
- Un diagrama de objetos se enfoca en los valores de los atributos de un conjunto de objetos y cómo esos objetos se relacionan entre sí.
- Los diagramas de objetos son sencillos de crear: se componen de objetos, representados por rectángulos, conectados mediante líneas.

Diagramas de estructura – diagrama de objetos



Tecnológico de Monterrey

Diagramas de estructura – diagrama de objetos



Tecnológico de Monterrey

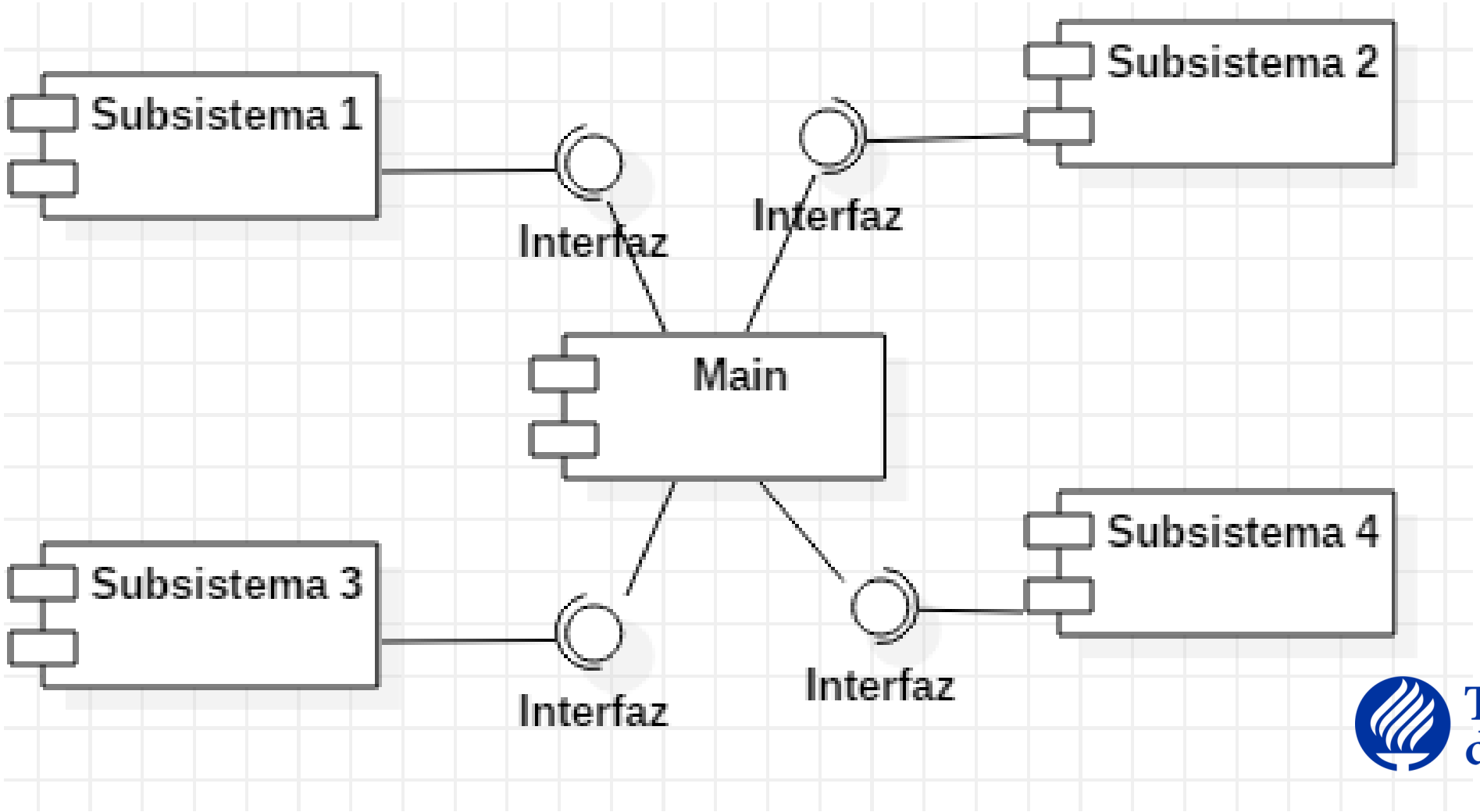
Diagramas de estructura – diagrama de componentes

- Un componente es un módulo que está aislado del sistema externo e interactúa con otros componentes mediante mensajes definidos en las clases.

- Los diagramas de componentes representan las relaciones entre los componentes individuales del sistema mediante una vista de diseño estática. Pueden ilustrar aspectos de modelado lógico y físico.

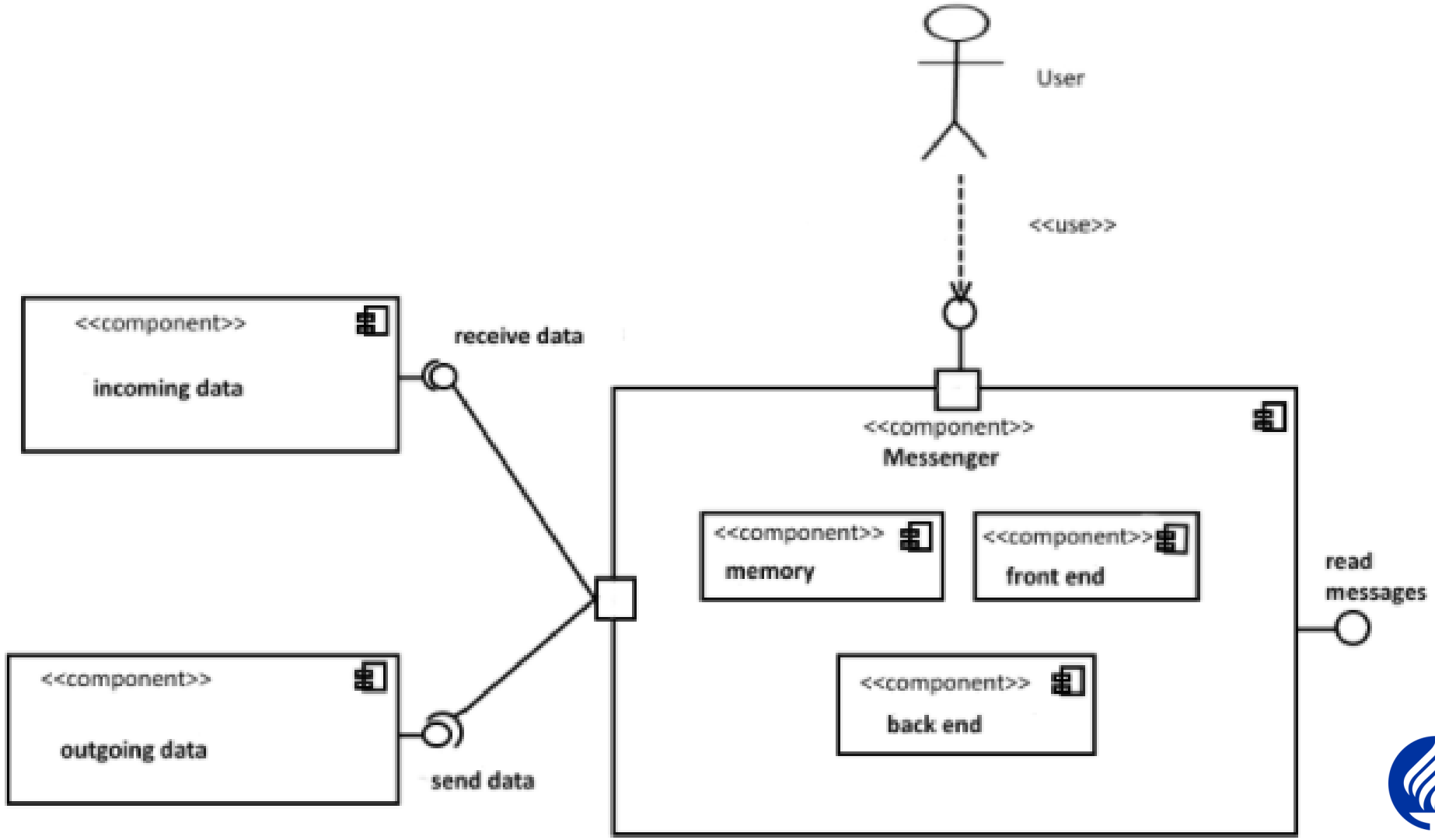
- En el contexto del UML, los componentes son partes modulares de un sistema independientes entre sí, que pueden reemplazarse con componentes equivalentes. Son autocontenidos y encapsulan estructuras de cualquier grado de complejidad. Los elementos encapsulados solo se comunican con los otros a través de interfaces.

Diagramas de estructura – diagrama de componentes



Tecnológico de Monterrey

Diagramas de estructura – diagrama de componentes



Tecnológico de Monterrey

Diagramas de comportamiento

- Los diagramas de comportamiento a diferencia de los diagramas estructurales, no son estáticos, sino que representan procesos y situaciones dinámicas.

- Este tipo de diagramas muestran como se comporta un sistema de información de forma dinámica. Es decir, describe los cambios que sufre un sistema a través del tiempo cuando está en ejecución.

- Los diagramas de comportamiento se emplean para visualizar, especificar, construir y documentar los aspectos dinámicos de un sistema.

Tecnológico de Monterrey

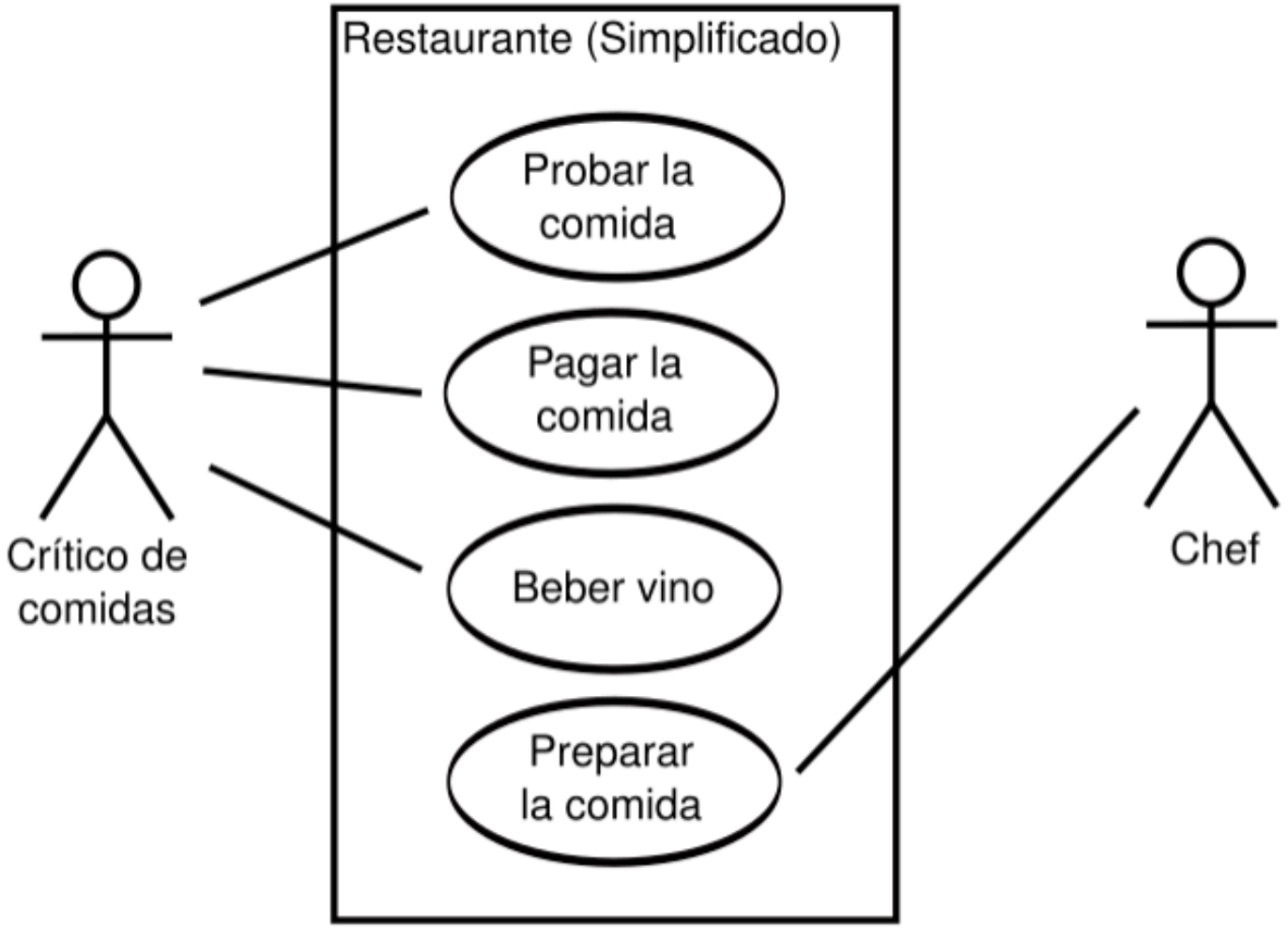
Diagramas de comportamiento – diagrama de casos de uso

- Los diagramas de comportamiento se emplean para visualizar, especificar, construir y documentar los aspectos dinámicos de un sistema.

- Los casos de uso son el mecanismo para capturar la conducta deseada de un sistema que está bajo desarrollo. Estas especificaciones no contienen detalles de cómo esta conducta es implantada.

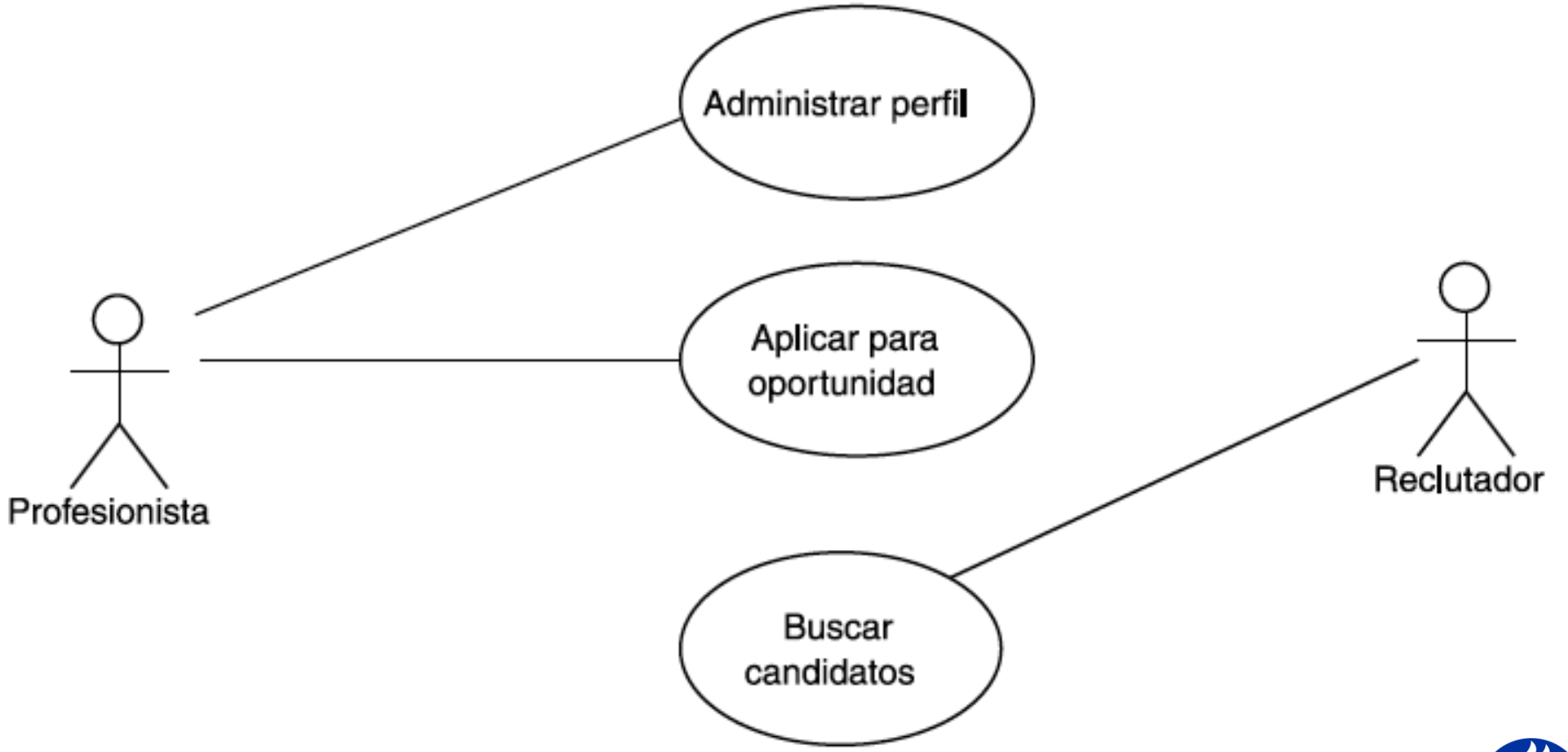
- Un modelo de casos de uso describe los requerimientos funcionales de un sistema. Este modelo contiene las funciones deseadas y sirve como un contrato entre el cliente y los desarrolladores.

Diagramas de comportamiento – diagrama de casos de uso



Tecnológico de Monterrey

Diagramas de comportamiento – diagrama de casos de uso



Tecnológico de Monterrey

Referencias

- Sommerville, I., Software Engineering, 10th Edition, Pearson, 2016, IN, 1292096144, 9781292096148.
- Connolly Thomas M, Database systems : a practical approach to design, implementation and management, 5thed., London : Addison-Wesley, 2010, 9780321523068.
- Martel, A., Gestión practica de proyectos con SCRUM, , EEUU, : libre, 2016.
- https://www.becas-santander.com/es/blog/metodologias-desarrollo-software.html

Tecnológico de Monterrey

Gracias!

Preguntas...

Tecnológico de Monterrey