

**Benemérita Universidad Autónoma de Puebla**

**Facultad de Ciencias de la Computación**

**Ingeniería en Tecnologías de la Información**

**Desarrollo Basado en Modelos**

**Generación de CRUD a base de un diagrama de clases**

**INTEGRANTES:**

Cortes Escalante Elihu

Nolasco Toledo Monserrat

Xochitemol Pérez Esleban

**Otoño 2019**

Introducción

Metodología

**Descripción del proyecto**

Generación de una plataforma web.

Utilizando una herramienta (jsUML2), crearemos un diagrama de clases que posteriormente exportaremos en XML.

El XML es el que se utilizara para crear una interfaz de usuario del tipo CRUD.

**Objetivos**

* Creación de un modelo de clases en la plataforma jsUML2.
* Importar el diagrama de clases a un XML o JSON.
* Generación de código para implementar una plataforma web a través de un CRUD.

**Alcances**

Creación de un prototipo de la plataforma web

**Limitaciones**

* Tiempo de trabajo.
* Integrantes del equipo.
* Ambiente de trabajo nuevo.
* Conocimiento.

**Requisitos**

Conocimientos necesarios en modelado de clases.

Conocimientos en la plataforma jsUML2.

**Herramientas.**

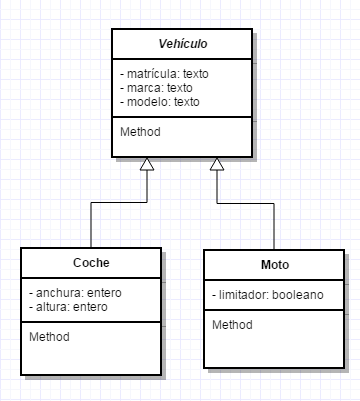
-jsUML2.

-SublimeText.

**Capitulo 1.- Análisis de Clases UML.**

Los diagramas de clases son uno de los tipos de diagramas más útiles en UML, ya que trazan claramente la estructura de un sistema al modelar sus clases, atributos, operaciones y relaciones entre objetos.

Si tenemos un buen modelado del proyecto, es mas sencillo comprender el proyecto y desarrollarlo.



Visibilidad

Relacion

Atributos

Metodos

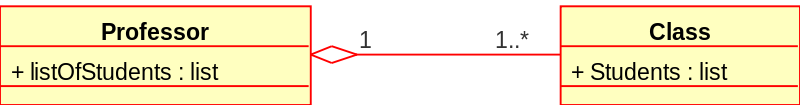
Nombre

* Nombre: Representa el nombre de la clase.
* Atributos: Representan los datos asociados a los objetos instanciados por esa clase. Los atributos pueden ser del tipo: int, string, float, booleano, entre otros.
* Visibilidad: La visibilidad puede ser publica (+), privada (-) o protegida (#).
* Relaciones: Los tipos más importantes de relaciones estáticas entre clases son los siguientes:
  + Asociación: Las relaciones de asociación representan un conjunto de enlaces entre objetos o instancias de clases. Es el tipo de relación mas general, y denota básicamente una dependencia semántica. Ejemplo:

https://upload.wikimedia.org/wikipedia/commons/4/4d/UML_role_example.gif

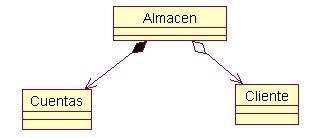
* + Agregación: Es un tipo de relación jerárquica entre un objeto que representa la totalidad de ese objeto y las partes que lo componen. Los objetos “son parte de” otro objeto complemento.

Se representa con un rombo hueco en la clase cuya instancia es una agregación de las instancias de la otra.



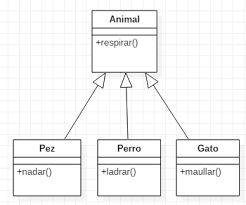
* + Composición: La composición es una forma de agregación donde la relación de propiedad es más fuerte, e incluso coinciden los tiempos de vida del objeto completo y las partes que lo componen.

Se representa con un rombo lleno en la clase cuya instancia contiene las instancias de la otra clase.



* + Herencia: Es el mecanismo que permite a una clase de objetos incorporar atributos y métodos de otra clase, añadiéndolos a los que ya posee. La clase de la cual se hereda se denominada superclase, y la que hereda subclase.

Esta relación se representa como una línea continua con una flecha hueca en el extremo que apunta a la superclase.



* Métodos:

**Capítulo 2.- Análisis de XML**

XML proviene de eXtensible Markup Language (“Lenguaje de Marcas Extensible”). Se trata de un metalenguaje (un [lenguaje](https://definicion.de/lenguaje/) que se utiliza para decir algo acerca de otro) extensible de etiquetas que fue desarrollado por el Word Wide Web Consortium (W3C).

El XML es una adaptación del SGML (Standard Generalized Markup Language), un lenguaje que permite la organización y el etiquetado de [documentos](https://definicion.de/documento/). Esto quiere decir que el XML no es un lenguaje en sí mismo, sino un [sistema](https://definicion.de/sistema) que permite definir lenguajes de acuerdo a las necesidades.



XML se plantea como un lenguaje estándar para el intercambio de información entre diferentes programas de una manera segura, fiable y libre, ya que no pertenece a ninguna compañía.

Muchas instituciones públicas ya están utilizando XML para almacenar su información, siguiendo estos criterios de organización de datos. Esto es lo que llamamos una base de datos en XML.

Otra posibilidad interesante del XML es que a partir de un documento en XML se pueden generar archivos PDF y en otros formatos. De esta forma, la información puede ser presentada de una manera visual para su lectura por las personas y el XML sólo quedaría para ser entendido por los programas; aunque si hacemos un esfuerzo, vemos que es fácil para una persona extraer la información de un documento XML directamente.

**Capítulo 3.- Representación de información de clases UML en XML**

Para la representación del diagrama de clases en UML, ocupamos la herramienta de jsUML2.