

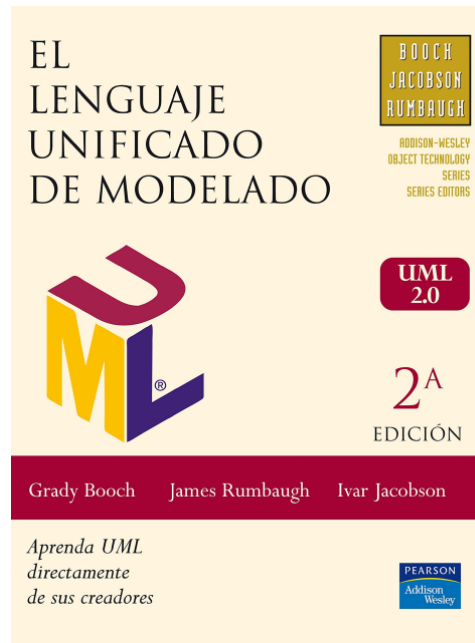


Práctica #2

Unified Modeling Language

1. UML

UML (Unified Modeling Language) es un lenguaje estándar de modelado para visualizar, especificar, construir y documentar artefactos de un sistema software orientado a objetos.



Es un estándar para describir los planos de un sistema, incluyendo cosas conceptuales como procesos de negocio y funciones de sistemas, tanto como cosas concretas como sentencias de lenguajes de programación, esquemas de bases de datos y componentes software reutilizables. Es un medio para: comunicarse con los clientes, comunicarse entre desarrolladores, o usar herramientas de generación automática de código/documentación.

La notación UML ha sido pensada para servir de lenguaje de modelado de objetos, independientemente del método implementado. Con UML se busca ante todo la simplicidad ya que es intuitivo, homogéneo y coherente.

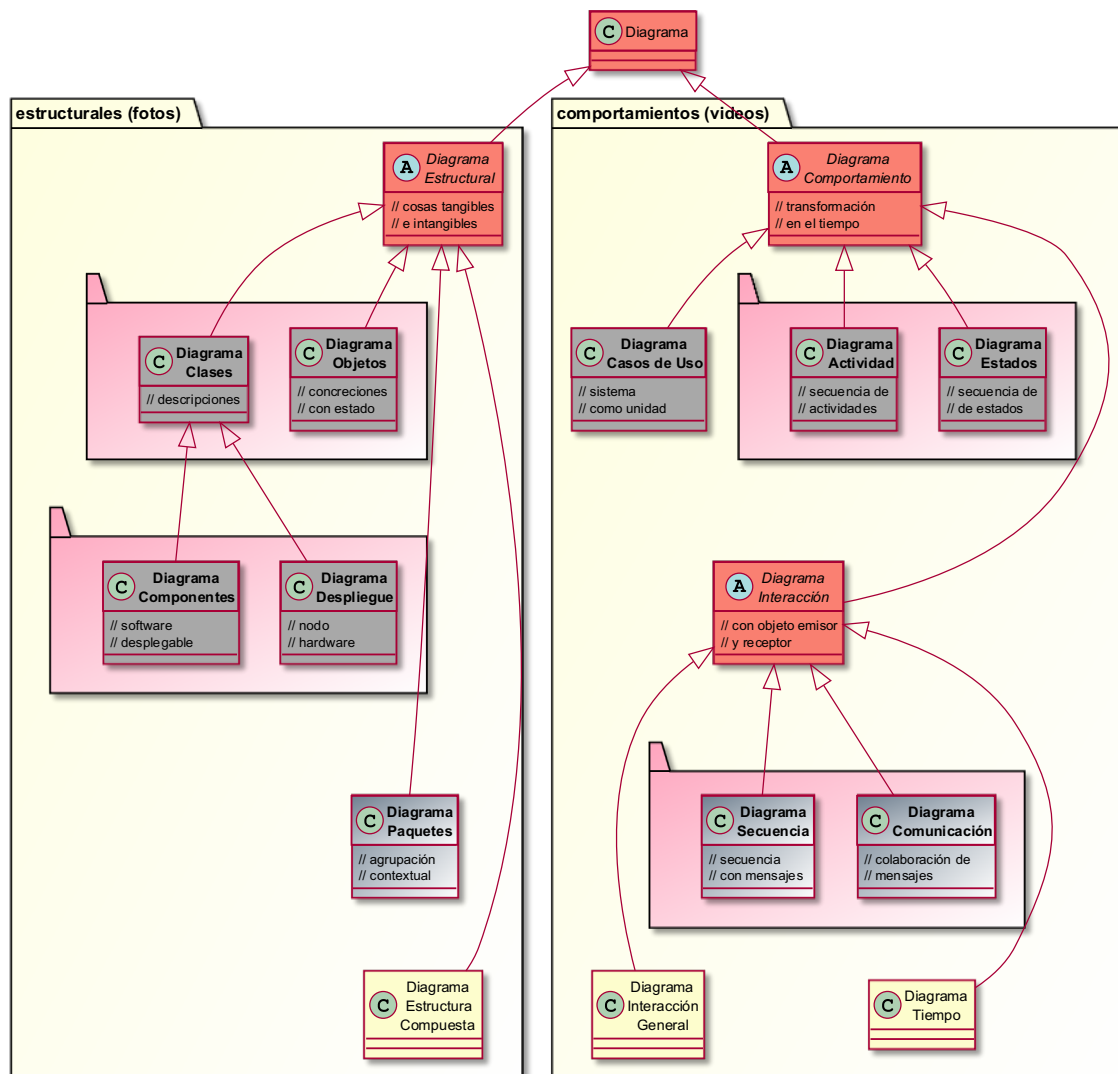
UML proporciona a los desarrolladores un vocabulario que incluye tres categorías: elementos, relaciones y diagramas. Hay cuatro tipos de elementos: estructurales, de comportamiento, de agrupación y de anotación. Hay 7 tipos principales de elementos estructurales: casos de usos, clases, clases activas (clase cuyas instancias son objetos activos), interfaces, componentes, colaboraciones y nodos. Hay 2 tipos de elementos de comportamiento: interacciones y máquinas de estado. Hay 4 tipos de agrupaciones: paquetes, modelos, subsistemas y marcos de trabajo. Solo hay un tipo de elemento de anotación: notas.



Dentro de la categoría relaciones encontramos 3 tipos: relación de dependencia, relación de asociación y relación de generalización. Dentro de la categoría diagramas, vemos que UML proporciona varios tipos divididos en diagrama estructurales y de comportamiento (ver imagen abajo). Los diagramas estructurales muestran los aspectos estáticos de un sistema representando el esqueleto relativamente estable, mientras que los diagramas de comportamiento muestran los aspectos dinámicos de un sistema tales como sus partes mutables.

Entre los diagramas estructurales tenemos: diagrama de clases, diagrama de objetos, diagrama de componentes, diagrama de despliegue y diagrama de paquetes. Entre los diagramas de comportamiento tenemos: diagrama de caso de uso, diagrama de estados, diagrama de actividad, y los diagramas de interacción: diagrama de secuencia, diagrama de comunicación.

Los diagramas son una presentación gráfica de un conjunto de elementos, que la mayoría de las veces se dibuja como un grafo conexo de nodos (elementos) y arcos (relaciones).

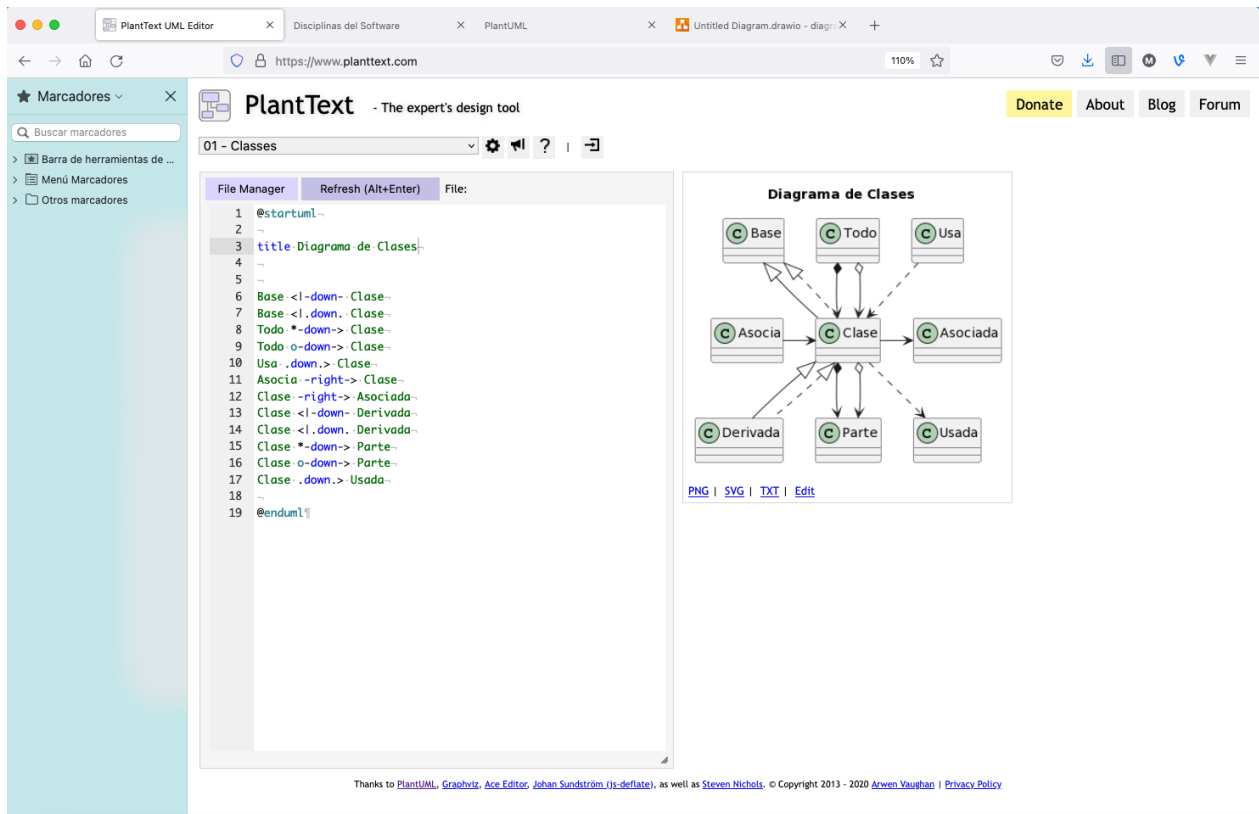




2. PlantUML

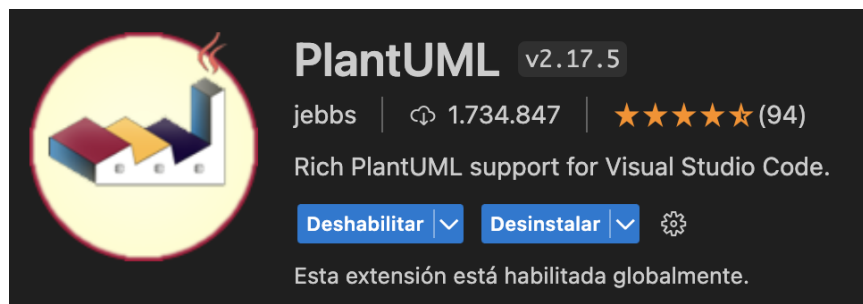
PlantUML es un generador de diagramas UML a partir de la descripción textual de cada elemento del diagrama y sus relaciones. Es una aplicación web que te permite a partir de unas instrucciones en un archivo de text generar distintos diagramas UML

URL: <https://plantuml.com/es/>



Guía de uso: <https://plantuml.com/es/guide>

Desde Visual Code podemos usar esta herramienta al instalar la extensión PlantUML. Esta extensión da soporte para editar los archivos de extensión “.plantuml”, ver previsualizaciones de los diagramas y exportarlos en formato de imagen.





3. Ejercicios

a) Practicar el uso de VisualCode y PlantUML:

1. Instalar extensión PlanUML y configurar el directorio de exportar imágenes para que sea el ".", y configurar el formato por defecto de la imagen para que sea "png"
2. Abrir el proyecto creado en la práctica #1 y buscar dentro del directorio "docs/diagrams" el archivo "actorsUseCaseDiagram.plantuml"
3. Hacer una previsualización del archivo
4. Exportar el diagrama en formato "png"

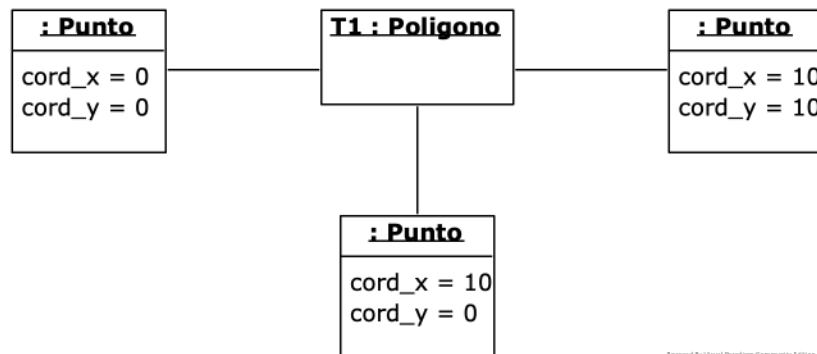
b) Modele las siguientes descripciones y conceptos utilizando un diagrama de clase:

1. Un teclado tiene por lo menos 105 teclas
2. Los estudiantes, los profesores y los empleados son personas
3. Un disco duro esta formado por pistas, y las pistas están formadas por sectores
4. Un triángulo es un polígono que esta formado por 3 lados
5. Un equipo de baseball está formado por jugadores y tienen asignado como homeclub un estadio
6. Un comité esta integrado por dos o más personas. Una persona puede o no pertenecer a comités. Cada comité tiene un coordinador que forma parte del comité y una misma persona puede coordinar más de un comité.
7. Una caracterización se fundamenta en al menos un criterio, un elemento clasificado se genera a partir de una caracterización que depende del criterio utilizado.
8. Un laboratorio tiene una diversidad de equipos, pero solamente un laboratorio nuclear puede tener equipos nucleares
9. La ecuación de FP tiene 2 operandos, el primero es UFC, el segundo es TCF. Los TCF dependen de la valoración de 14 subfactores de complejidad técnica (Fi)
10. Un trabajo de grado especial (TEG) y un trabajo de grado de maestría (TGM) son trabajos de investigación. Todo trabajo de investigación es dirigido por uno o a lo sumo dos profesores que juegan el rol de tutor. Un mismo profesor puede dirigir varios trabajos de investigación. Un TGM siempre se realiza de manera individual, es decir, un solo estudiante de postgrado mientras que un TEG puede ser realizado por uno o dos estudiantes de pregrado.



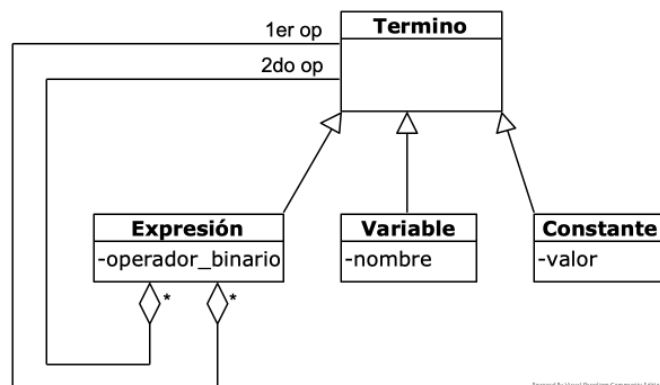
11. Suponga que un documento gráfico está compuesto de hojas. Cada hoja contiene un objeto dibujo. Un objeto dibujo puede ser: texto, figura 2D, o grupo. Un grupo consta de al menos dos objetos dibujo. Un objeto de dibujo puede ser a lo sumo miembro de un grupo. Una figura 2D puede ser círculo o cuadrado.

c) Suponga el siguiente diagrama de objetos:



1. Haga el diagrama de clases correspondiente
2. Suponga dos triángulos con un lado común. Hacer un diagrama de objetos donde: “un punto pertenece a uno o más polígonos”
3. Hacer un diagrama de objetos de un polígono con cuatro puntos: (0,0), (1,0), (0,1) y (1,1)

d) Suponga el siguiente diagrama de clases:

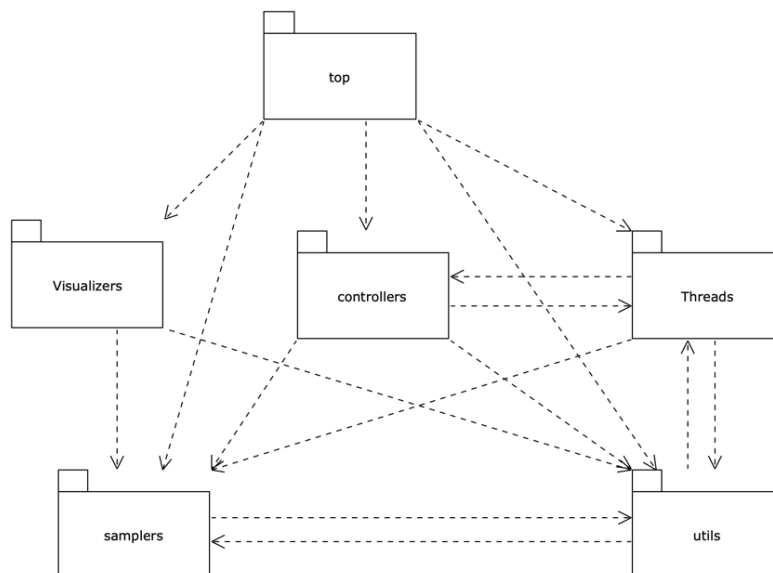


1. Preparar un diagrama de objetos para la siguiente expresión: $(x + y/2) / (x/3 + y)$

* indica que un término puede ser utilizado en más de una expresión



e) Supongamos el siguiente diagrama de paquetes:



1. ¿Qué significa que el paquete “top” tenga una dependencia con el paquete “controllers”?
 2. ¿Con cuántos paquetes tiene dependencias el paquete “Threads”? ¿Cuántas dependencias son de entrada y cuántas de salida? ¿Si se hace un cambio en alguna de las clases del paquete “Threads” qué podría suceder?
 3. Hablando en términos de modularidad, ¿este diagrama muestra un código con bajo o alto acoplamiento entre los paquetes? ¿Esto es bueno o malo?
- f) Obtenga de la siguiente descripción el diagrama de clase y un diagrama de objetos que represente un juego entre Venezuela y Colombia en los Panamericanos que se realizan en Chile (complete los datos del juego con la información que quiera, no es necesario poner a todos los jugadores):

Un equipo de volleyball tiene de 6 a 20 jugadores, uno de los cuales es el capitán del equipo. Todo equipo tiene un nombre y un registro, y todo jugador tiene un número y una posición. Los equipos compiten en juegos y cada juego tiene una localidad y una puntuación. Un equipo puede tener un entrenador que tiene un nivel de acreditación y un número de años de experiencia. Un mismo entrenador puede entrenar diferentes equipos. Cualquier persona, ya sea jugador de un equipo, así como el entrenador, tienen un teléfono, nombre y dirección.

Al terminar, colocar en el reto del Aula Virtual del ejercicio “f”, subir la imagen resultante y en GitHub los archivos de plantUML. También puede subir a GitHub todos los ejercicios que haga con PlantUML. Coloque de nuevo el URL de su repositorio de GitHub en la entrega del reto 04.