



Politécnico
Internacional

INGENIERIA DE SOFTWARE - I

“Si lo puedes imaginar, se puede programar.”



ESPACIOS DE FORMACIÓN

1	Viernes	6 PM a 10:30 PM - 1 Hora
2	Break	Por acordar.

A TENER EN CUENTA



1. Cerrar los micrófonos y desactivar las cámaras.



2. Solicitar la palabra mediante el chat.



3. Preguntar

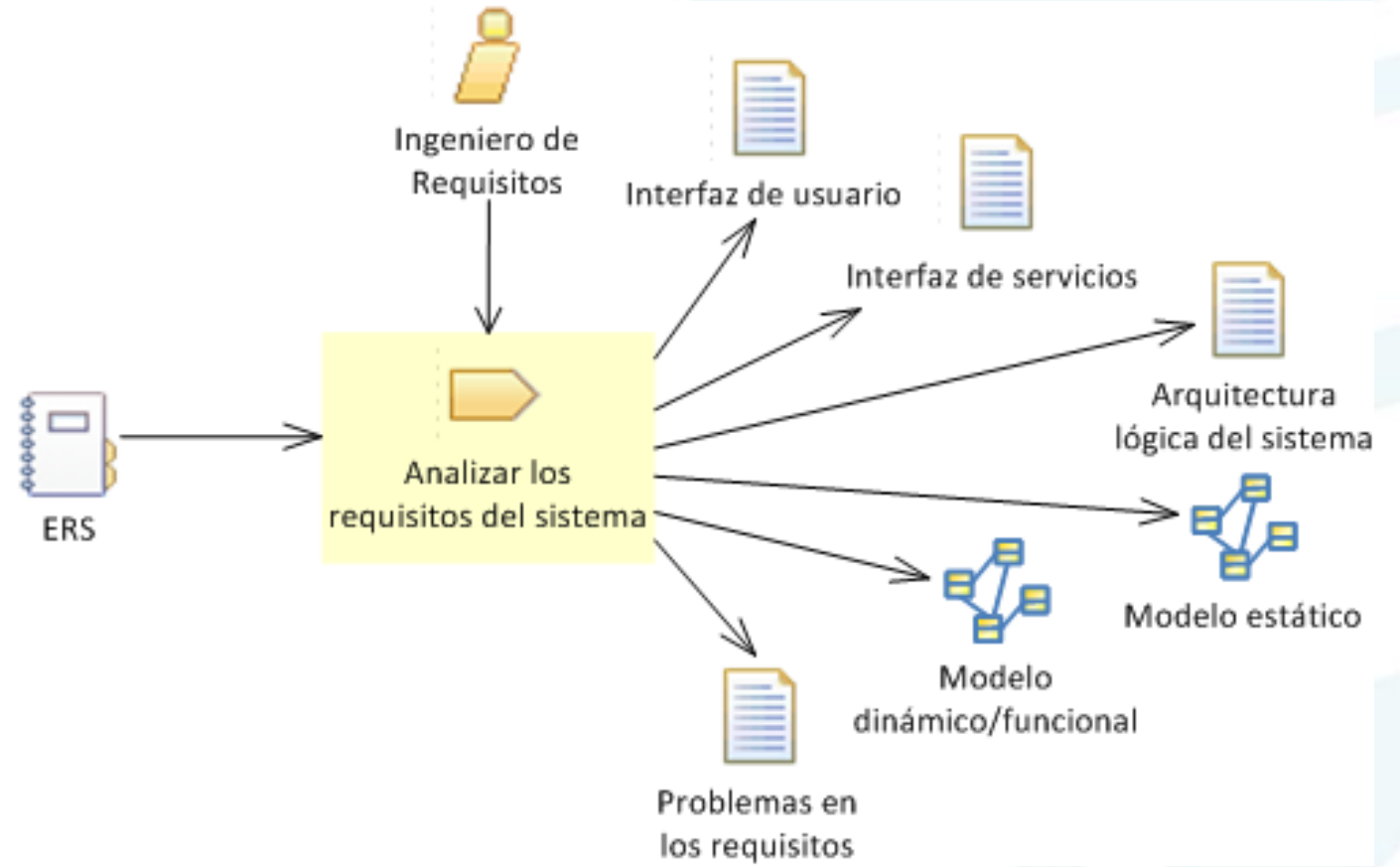
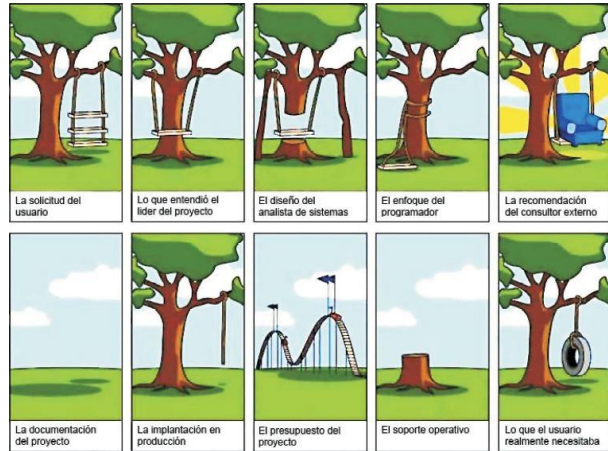


CONOCIMIENTOS PREVIOS

- *¿Conocen que es UML?*

AGENDA

- Requisitos Funcionales y no Funcionales.
- Empecemos con UML.
- Proyecto INNOVA.

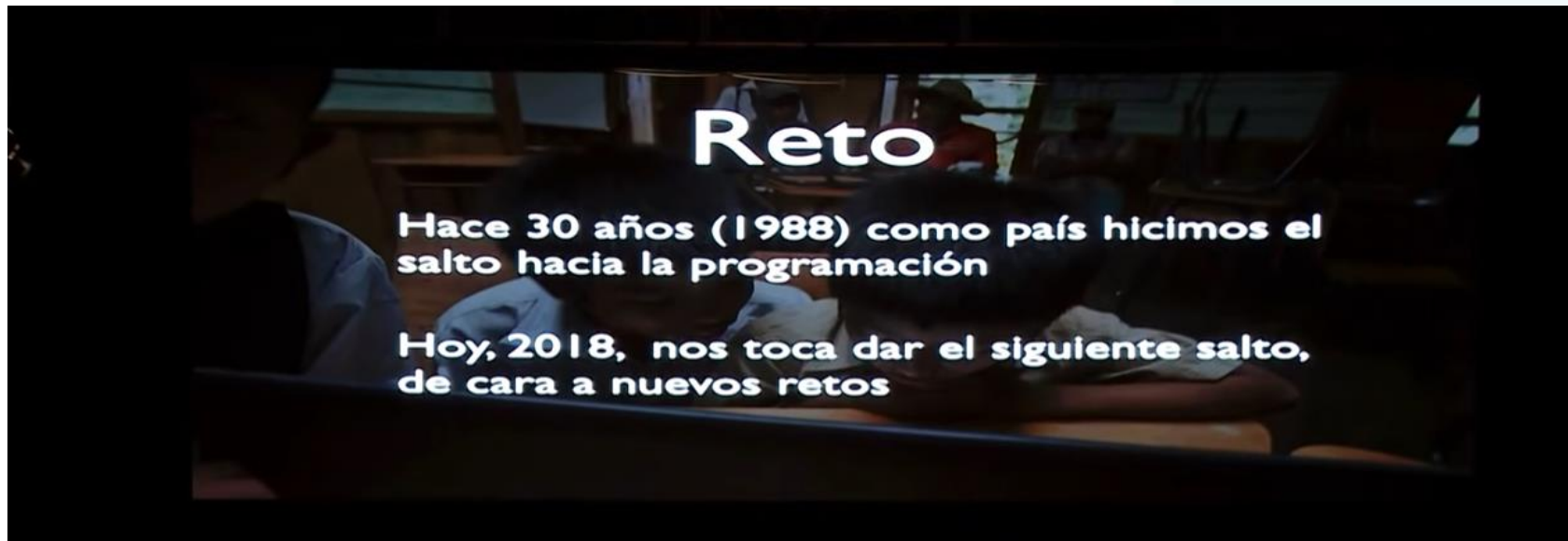


Avances de Nuestro Proyecto



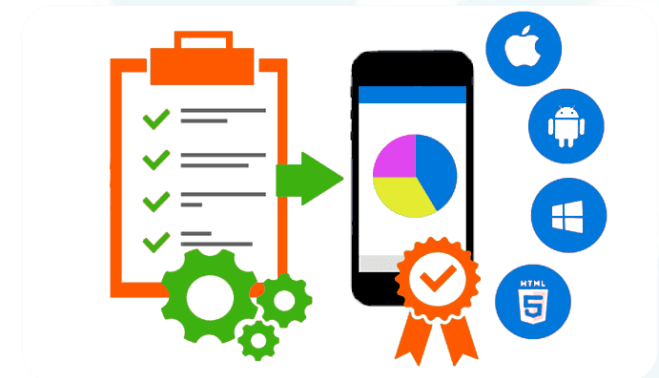
<https://www.youtube.com/watch?v=0TFZfwB67Y&t=848s>

[Pensamiento computacional | Alberto Cañas | TEDxPuraVidaED](#)



- Funcionales
- No funcionales

REQUISITOS DE SOFTWARE



REQUISITOS FUNCIONALES: Describen las interacciones entre el sistema y su ambiente, en forma independiente a su implementación. El ambiente incluye al usuario y cualquier otro sistema externo con el cual interactúe el sistema.

REQUISITOS DE SOFTWARE



REQUISITOS NO FUNCIONALES: Describen atributos sólo del sistema o del ambiente del sistema que no están relacionados directamente con los requisitos funcionales. Los requisitos no funcionales incluyen restricciones cuantitativas, como el tiempo de respuesta o precisión, tipo de plataforma (lenguajes de programación y/o sistemas operativos, etc.)

REQUISITOS DE SOFTWARE



+ Requisitos funcionales

Un requisito funcional define una función del sistema de software o sus componentes. Una función es descrita como un conjunto de entradas, comportamientos y salidas. Los requisitos funcionales pueden ser: cálculos, detalles técnicos, manipulación de datos y otras funcionalidades específicas que se supone, un sistema debe cumplir.

REQUISITOS DE SOFTWARE

ANÁLISIS DE REQUISITOS DE SOFTWARE



+ Requisitos no funcionales

Un requisito no funcional o atributo de calidad es, en la ingeniería de sistemas y la ingeniería de software, un requisito que sabe bien y especifica criterios que pueden usarse para juzgar la operación de un sistema en lugar de sus comportamientos específicos, ya que estos corresponden a los requisitos funcionales. Por tanto, se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar, sino características de funcionamiento, por eso suelen denominarse Atributos de calidad de un sistema. Queda entonces el requisito no funcional, que son las restricciones o condiciones que impone el cliente al programa que necesita, por ejemplo el tiempo de entrega del programa, el lenguaje o la cantidad de usuarios

REQUISITOS DE SOFTWARE

ANÁLISIS DE REQUISITOS DE SOFTWARE



El Lenguaje unificado de modelado (UML) desempeña un papel importante en el desarrollo de software, pero también en otros sistemas de muchos sectores de la industria, ya que es un medio de mostrar visualmente el comportamiento y la estructura de un sistema o un proceso.

UML ayuda a identificar posibles errores en las estructuras de la aplicación, el comportamiento del sistema u otros procesos empresariales.



UML

UML apareció en escena por primera vez en la década de los 90 de la mano de tres ingenieros de software: Grady Booch, Ivar Jacobson y James Rumbaugh.

Estos ingenieros querían desarrollar una forma menos caótica de representar el desarrollo de software cada vez más complejo y separar al mismo tiempo la metodología del proceso.

Actualmente, UML sigue siendo la notación estándar a la que recurren los desarrolladores, así como los jefes de proyecto, las empresas tecnológicas y profesionales de todos los sectores.

UML

¿Por qué usar UML?

UML apareció en escena por primera vez en la década de los 90 de la mano de tres ingenieros de software: Grady Booch, Ivar Jacobson y James Rumbaugh.

Estos ingenieros querían desarrollar una forma menos caótica de representar el desarrollo de software cada vez más complejo y separar al mismo tiempo la metodología del proceso.

Actualmente, UML sigue siendo la notación estándar a la que recurren los desarrolladores, así como los jefes de proyecto, las empresas tecnológicas y profesionales de todos los sectores.

UML

¿Por qué usar UML?

- Simplifica la complejidad
- Mantiene las líneas de comunicación abiertas
- Automatiza la producción de software y los procesos
- Ayuda a resolver problemas de diseño persistentes
- Aumenta la calidad del trabajo
- Reduce los costes y el tiempo de comercialización

UML

¿Cuáles son los beneficios de UML?

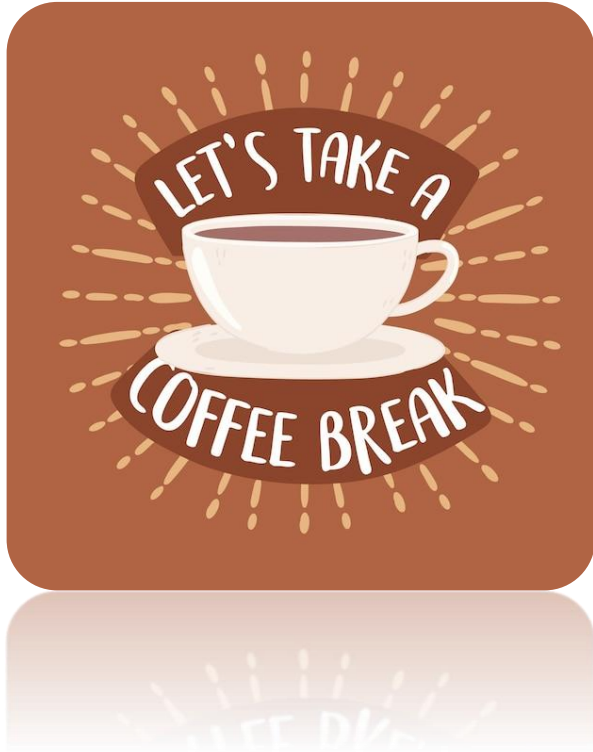
Existe dos tipos principales de diagramas de UML:

Diagramas de estructuras y diagramas de comportamiento (y de estos grupos se derivan muchos otros). Estas variaciones existen para representar los diversos tipos de escenarios y diagramas que usan diferentes tipos de personas.

Desde clientes a jefes de proyecto, pasando por autores técnicos, diseñadores, analistas, programadores y responsables del control de calidad, cada rol utilizará un diagrama concreto adecuado a sus necesidades. El objetivo de UML es expresar visualmente los diagramas de forma que todos puedan entenderlos fácilmente.

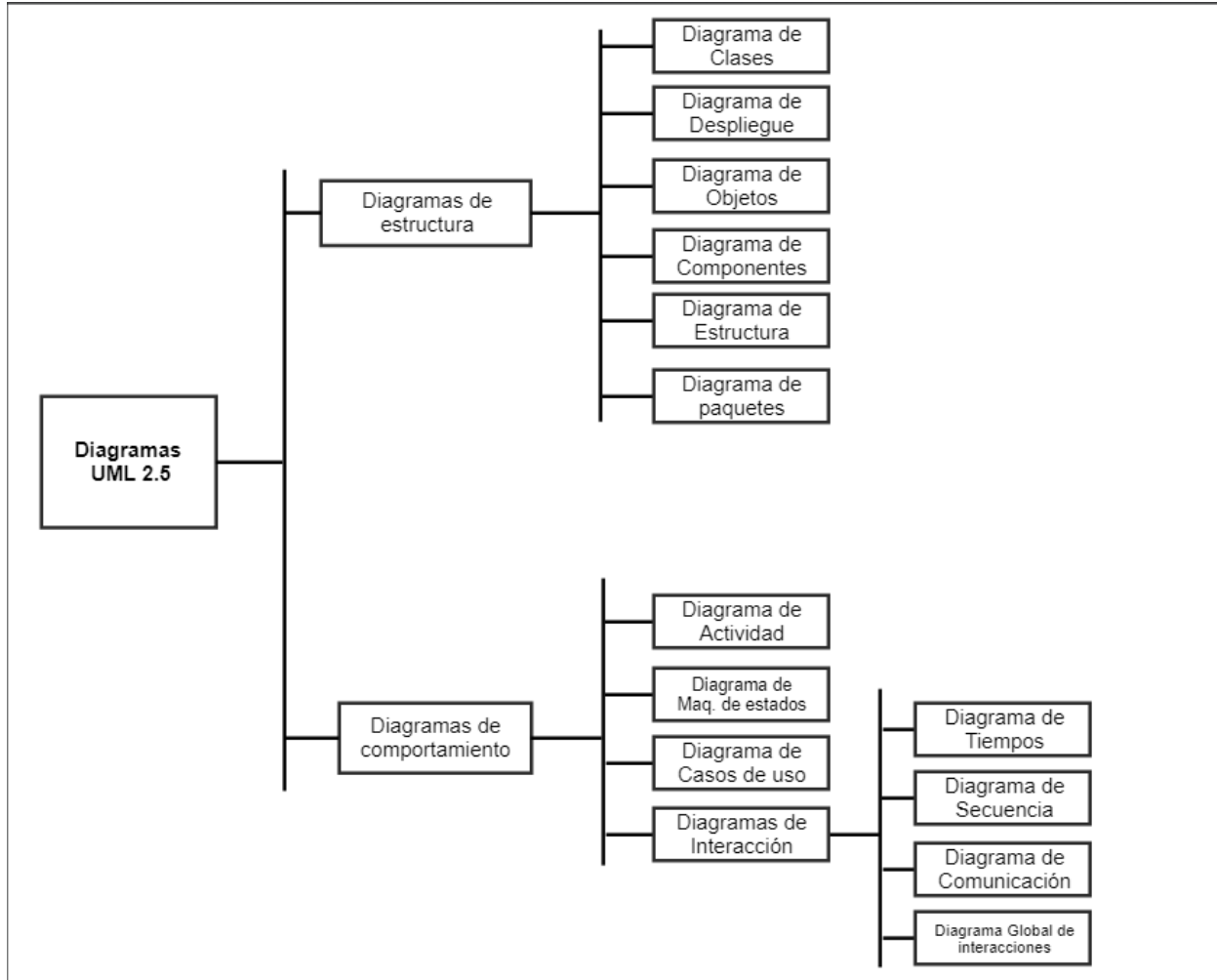
UML

Tipos de diagramas de UML



PAUSA ACTIVA

15 Min



UML

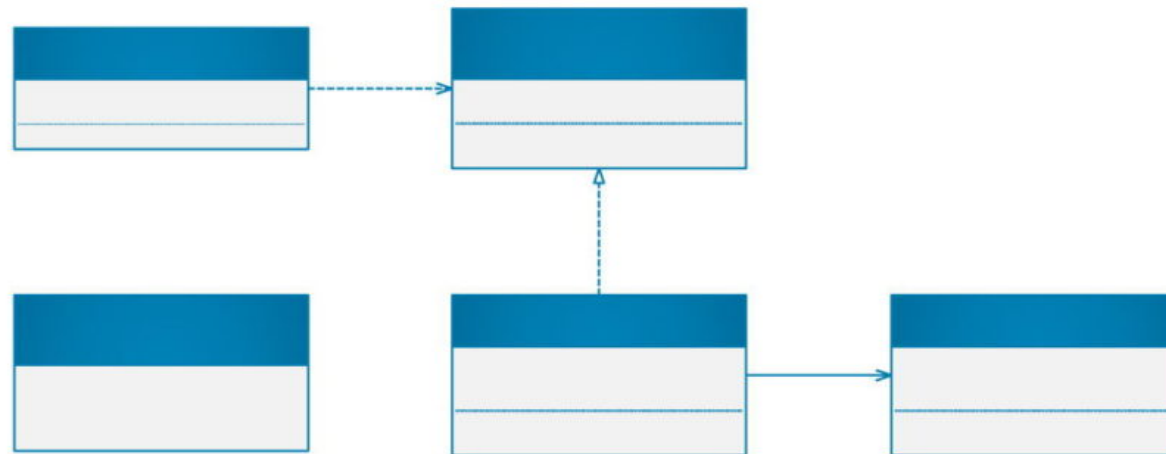
Tipos de diagramas de UML

Los diagramas estructurales representan la estructura estática de un programa de software o sistema, y también muestran los diferentes niveles de abstracción e implementación. Se utilizan para poder visualizar las distintas estructuras que componen un sistema, como una base de datos o una aplicación. Muestran la jerarquía de los componentes o módulos, y la forma en que se conectan o interactúan entre sí. Estas herramientas ofrecen pautas y garantizan que todas las partes de un sistema funcionen según lo previsto en relación con las demás.

UML

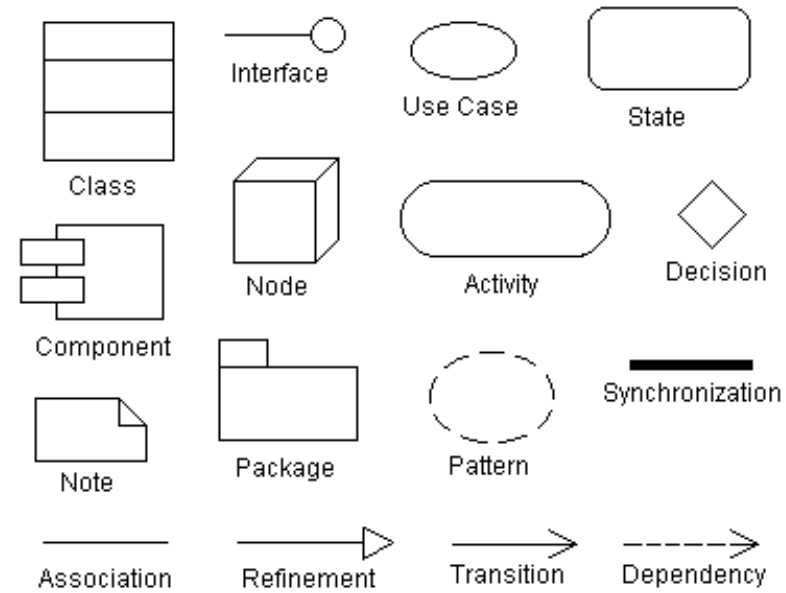
Diagramas estructurales

- **Diagrama de clases.** Este diagrama, el más común en el desarrollo de software, se utiliza para representar el diseño físico y lógico de un sistema y mostrar sus clases. Es similar a un diagrama de flujo en cuanto a que las clases se representan con cuadros. Este diagrama ofrece una representación visual de las distintas clases y de la forma en que están interrelacionadas. Cada clase tiene tres compartimentos:
- Sección superior: nombre de la clase
- Sección central: atributos de la clase
- Sección inferior: métodos u operaciones de la clase



- **Diagrama de objetos.** Este diagrama se utiliza a menudo como una forma de comprobar la exactitud de un diagrama de clases, es decir, determinar si funcionará en la práctica. Muestra los objetos de un sistema y sus relaciones, y ofrece una perspectiva mejor de los posibles errores de diseño que deben corregirse.
- **Diagrama de componentes.** Denominado también “diagrama de flujo de componentes”, muestra los grupos lógicos de elementos y sus relaciones. Es decir, ofrece una vista más simplificada de un sistema complejo descomponiéndolo en componentes más pequeños. Cada una de las partes se representa mediante una caja rectangular, con su nombre escrito dentro. Los conectores definen la relación o las dependencias entre los distintos componentes.
- **Diagrama de estructuras compuestas.** Este diagrama apenas se utiliza fuera del campo de desarrollo de software. La razón de ello es que, aunque es similar a un diagrama de clases, describe con mayor detalle la estructura interna de varias clases y muestra las interacciones entre ellas. A menos que seas un desarrollador, una vista más general será normalmente suficiente.
- **Diagrama de implementación.** Este diagrama muestra los componentes de hardware (los nodos) y de software (los artefactos) y sus relaciones. Ofrece una representación visual del lugar exacto donde se implementa cada componente de software.
- **Diagrama de paquetes.** Se utiliza para ilustrar las dependencias entre los paquetes que componen un modelo. El objetivo principal es mostrar la relación entre los distintos componentes de gran tamaño que conforman un sistema complejo.
- **Diagrama de perfiles.** Este diagrama es más un lenguaje que un diagrama. Un diagrama de perfiles ayuda a crear nuevas propiedades y semántica para los diagramas UML mediante la definición de estereotipos, valores etiquetados y restricciones. Estos perfiles te permiten personalizar un metamodelo UML para diferentes plataformas (por ejemplo, Java Platform, Enterprise Edition (Java EE) o Microsoft .NET Framework) y dominios (por ejemplo, modelado de procesos empresariales, arquitectura orientada a servicios, aplicaciones médicas, etc).

Nomenclatura



Object Diagram

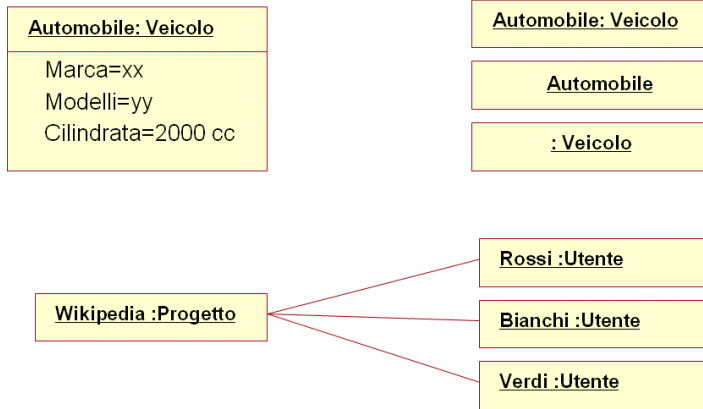


Diagrama de Componentes

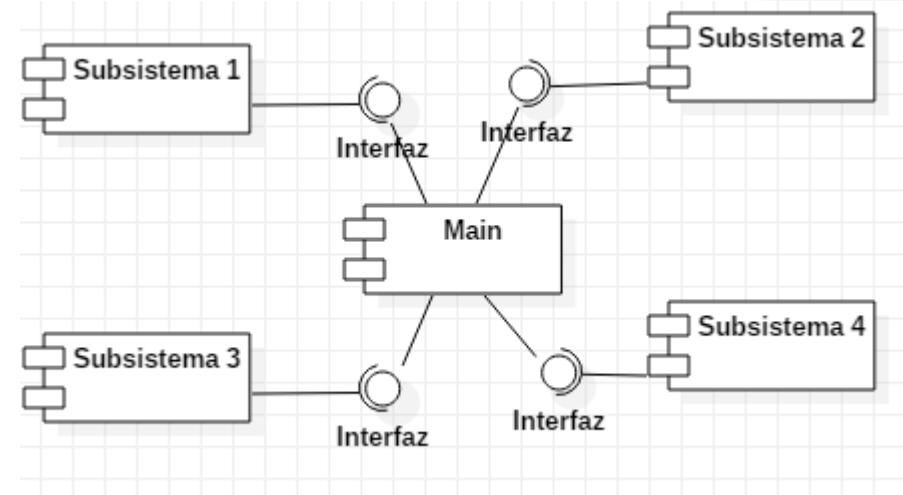


Diagrama de Estructuras Compuestas

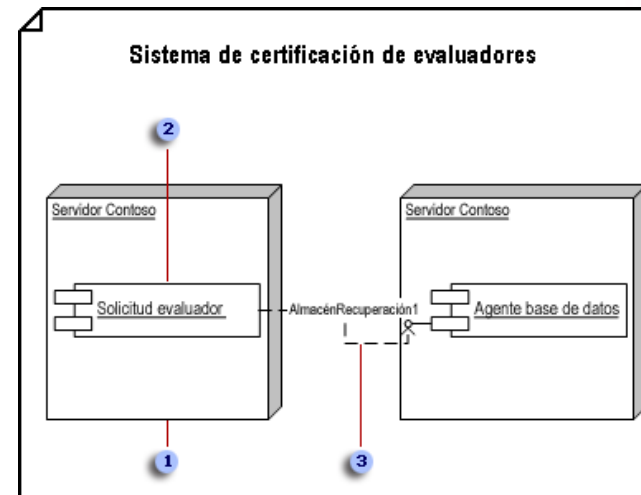
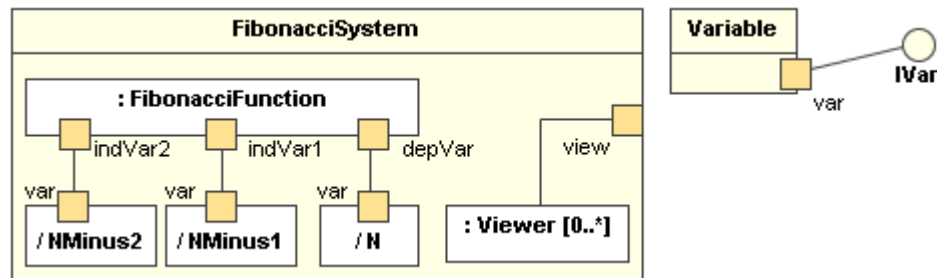


Diagrama de Implementación

Diagrama de Estructuras Compuestas

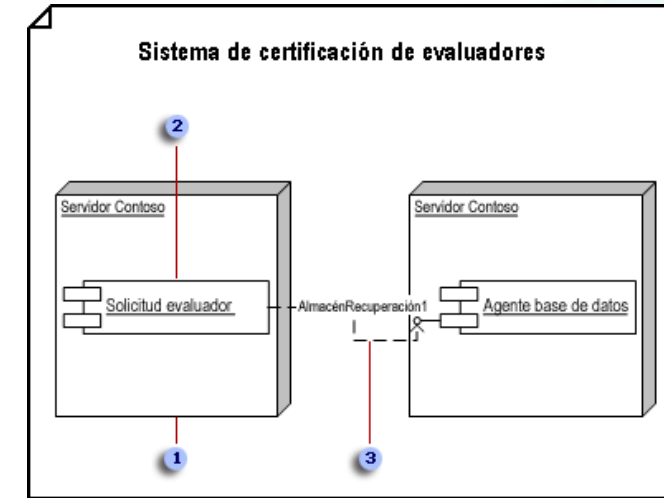
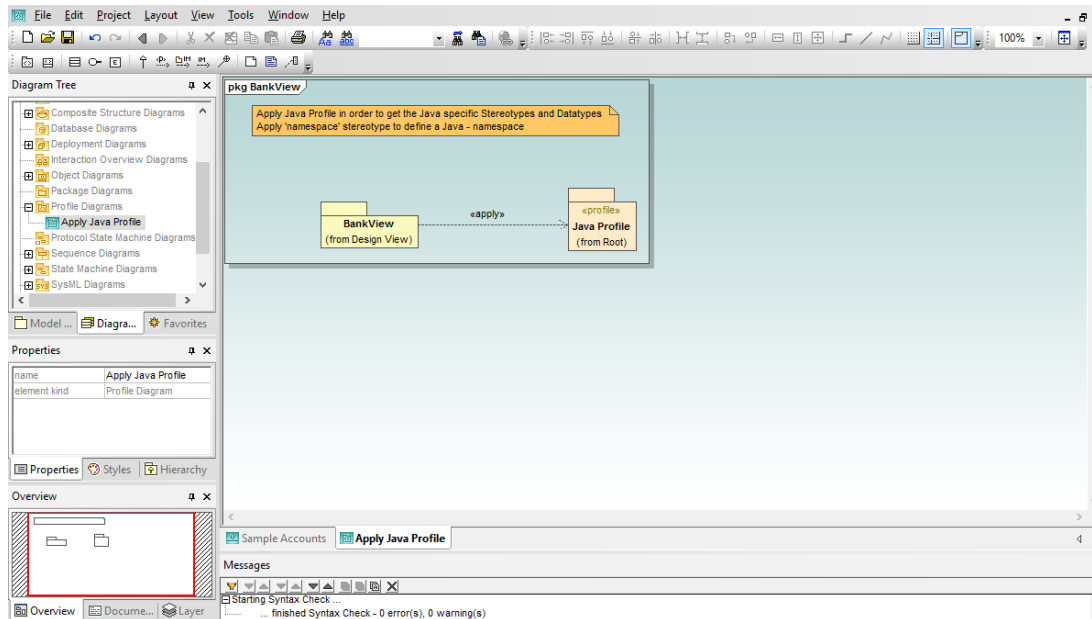
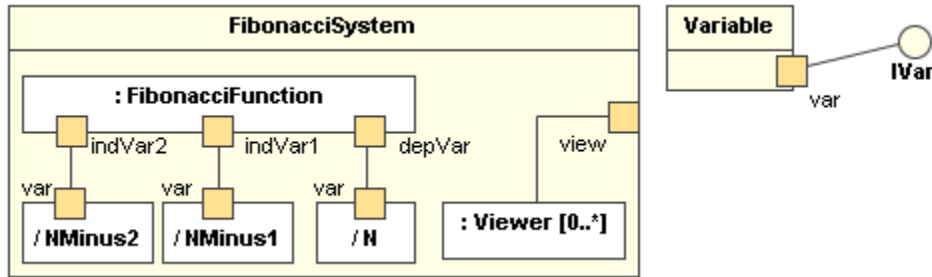


Diagrama de Implementación

Diagrama de perfiles



Manos a la obra

- ***Vamos a interactuar con nuestra Especificación de Requisitos de Software.***

<https://www.lucidchart.com>

 **Lucidchart**

[Contactar a Ventas](#)

Paso 1 de 3

Elige tu plan

	Gratis	Individual	Team	Corporativo
Precio por mes	\$0	A partir de \$7.95	A partir de \$9.00 por usuario 3 usuarios como mínimo.	¡Comunícate con nosotros para una cotización!
Documentos editables	3	Ilimitado	Ilimitado	Ilimitado
Plantillas profesionales	100	+1000	+1000	+1000
Colaboración	Básica	Básica	Avanzada	Avanzada
Integraciones	Básica	Básica	Avanzada	Avanzada
Seguridad	✓	✓	✓	✓
Administra los controles fácilmente			Básica	Avanzada
Datos avanzados y automatización				✓
SSO				✓

<https://www.genmymodel.com/products/>



Products

Support/Contact

About

Online Modeling made easy



Archimate®



Journey Mapping



Data Modeling



BPMN



UML



Flowcharting



Easy login

Log into GenMyModel to perform online modeling using your existing google, github or just an email



Easy diagramming

Copy/paste and undo/redo, fine...
GenMyModel diagramming is intuitive with alignment helpers, smart placement, overview and tree view. Easily locate any element.



Easy search

Easily search any diagram, object or attribute within the repository.

<https://www.gliffy.com/pricing>

Pricing for Gliffy's Online Diagramming App

Professional

For everyone from single users to mid-sized teams, get powerful diagramming features to capture your ideas.

1 to 9 Users	10 to 50 Users
\$8	\$6
per user/month, billed annually Or \$10 per user, billed monthly	per user/month, billed annually Or \$8 per user, billed monthly

- Unlimited diagrams
- Access to templates
- Import from other applications
- 24/7 email support
- Live diagram embedding
- Google Drive plug-in
- Private sharing
- Commenting tool

Save up to 25% when you choose annual billing.

BUY NOW

Enterprise

Enterprise-grade functionality for large teams or entire organizations.

Any Size Team

Contact our sales team for custom pricing packages.

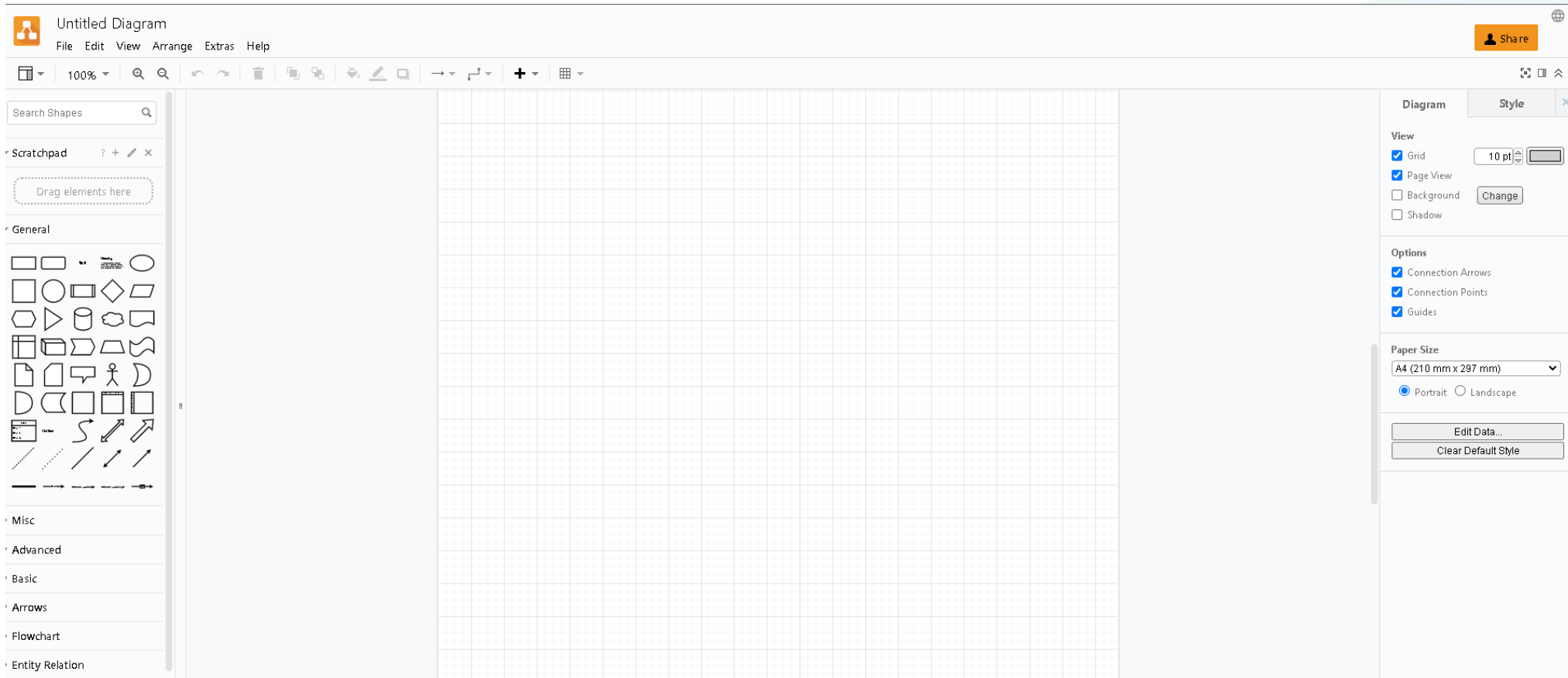
GET A QUOTE

Everything in Professional, plus:

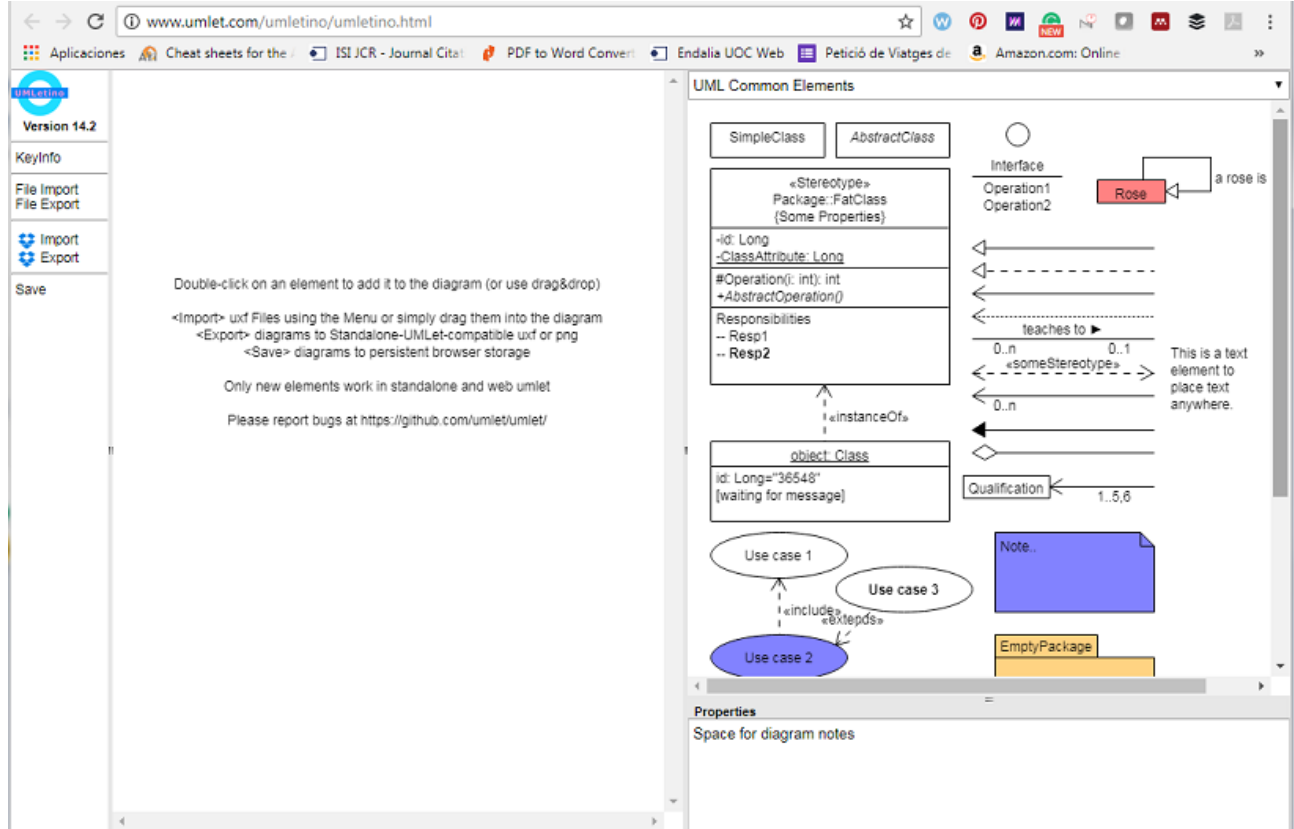
- Easy admin controls
- Centralized security
- Team management
- Single sign-on (SSO) integration
- Automatic diagram migration
- Dedicated phone and email support
- Shared online workspace
- and more.

[Learn more about Gliffy's enterprise diagramming app >>](#)

GET A QUOTE



<https://app.diagrams.net/>



UMLETINO

Diagramo
diagrams with HTML5

Fork us on : [Bitbucket](#) or [GitHub](#)

Home Features Download Register Buy Login

Diagramo is an online flowchart software

- ✓ It's FREE and Open Source ([GPL](#))
- ✓ It's pure [HTML5](#)
- ✓ Share and collaborate
- ✓ Export to SVG, Gif and JPEG
- ✓ Life time support & storage

Try it Now!
No account required

Download it
Completely free

HTML5
Based ONLY on HTML5. No Flash, no Java or other plugins.

Share and collaborate
Share diagrams with other teammates. Or make them public if you want.

Install it on your server
[Download it](#) and deploy it on your server. All you need is PHP and a MySQL database.

Customize it!
[Take the sources](#) and tune them to your needs.

Free! GPL!
Yes, it's a GPL application. You can follow with its code.

Export to SVG, Gif and JPEG
You can export diagrams in SVG, Gif or JPEG format.

DIAGRAMO



Vamos a interactuar

Encuentra cuantas bombas de servientrega hay en la imagen.

Vamos a interactuar

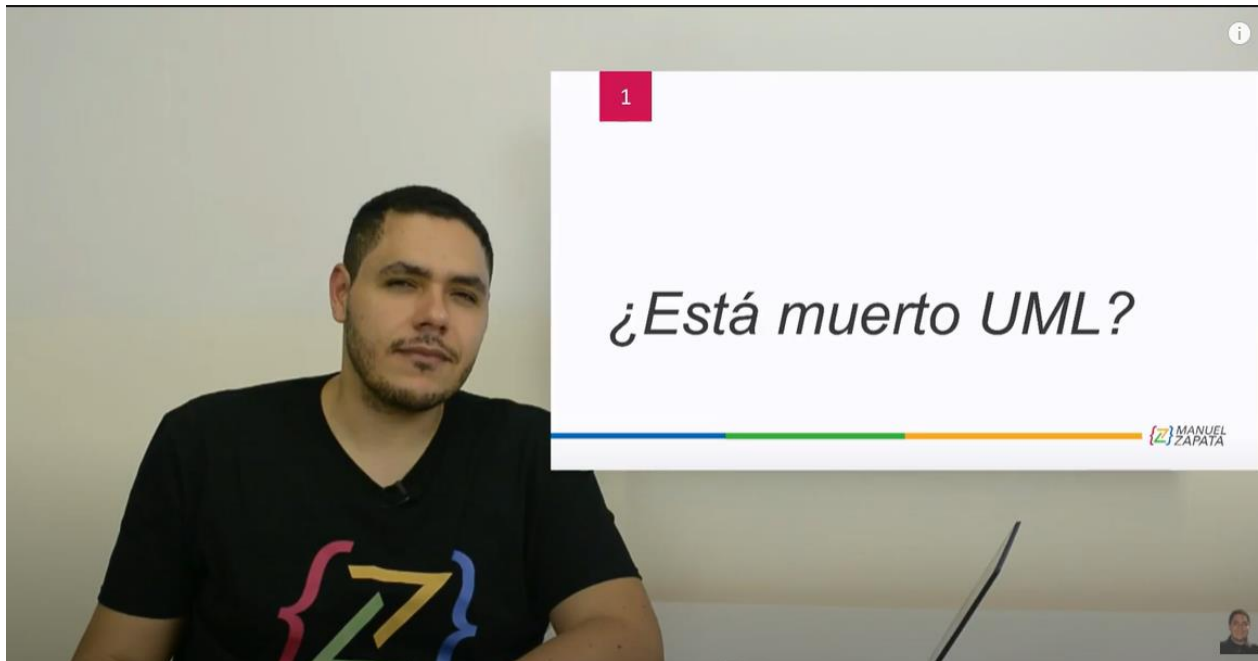
Revisemos las herramientas presentadas:

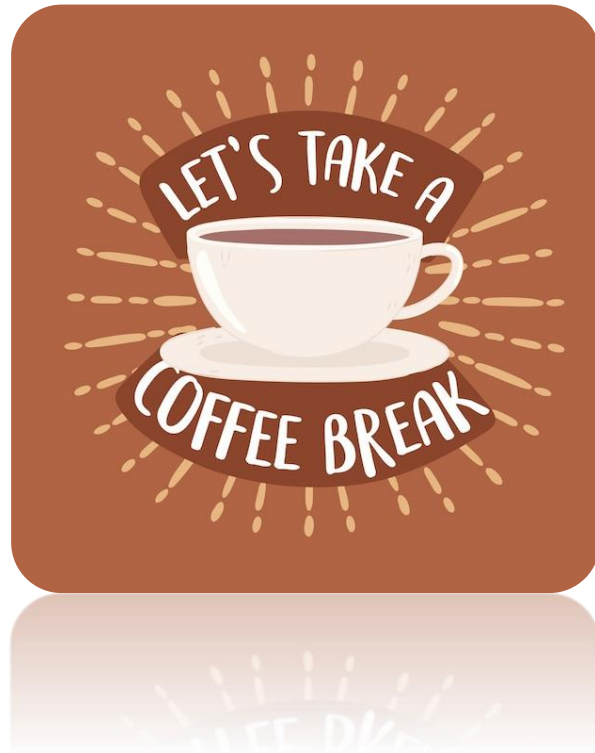
- Lucidchart
- Genmymodel
- Gliffy
- App.diagrams.net
- UMLETINO
- Diagramo

Elija tres Herramientas y responda las siguientes preguntas:

1. Indique tres fortalezas que encuentra en la herramienta
2. Indique una debilidad que encuentra en la herramienta

<https://www.youtube.com/watch?v=Quv-XtIWVBc>





PAUSA ACTIVA

15 Min

Diagrama UML Automático en NETBEANS con EasyUML





Manos a la obra

- ***Vamos a interactuar con con UML.***

¿preguntas?

CONCLUSIONES