

## Conceptos de las herramientas e imágenes de como realizarlas.

las herramientas para utilizar son:

**Enterprise Architect:** Sparx Systems Arquitecto Empresarial es una herramienta visual de modelado y diseño basada en el OMG UML. La plataforma admite: el diseño y la construcción de sistemas de software; modelar procesos comerciales; y modelar dominios basados en la industria.

**Rational Software Architect:** IBM Rational Software Architect es una herramienta de desarrollo y diseño integrado que fortalece el desarrollo dirigido por modelos UML para la creación de servicios y aplicaciones con arquitecturas sólidas. Esta herramienta permite especificar y mantener aspectos clave de las arquitecturas para servicios, infraestructuras y aplicaciones, colaborar más efectivamente con los miembros del equipo, comunicarse mejor con los participantes del proyecto y asegurarse de que la salida cumple los requisitos establecidos.

**Visual Paradigm:** Visual Paradigm es una herramienta de diseño y gestión para sistemas de TI que proporciona a los desarrolladores de software una plataforma de desarrollo de vanguardia para crear aplicaciones de calidad más rápidamente, de mejor manera y a menor costo.

Esta herramienta soporta el ciclo de vida completo del desarrollo de software a través de la representación de todo tipo de diagramas y está disponible en varias ediciones para satisfacer diferentes necesidades, incluyendo Enterprise, Professional, Community, Standard, Modeler y Personal.

Visual Paradigm es una herramienta visual UML de modelado y CASE, que es poderosa, multiplataforma y fácil de usar.

Además, ofrece una versión gratuita y de código abierto, la versión Visual Paradigm UML 6.4 Community Edition

**Star UML:** StarUML es una herramienta de código abierto para el modelado de software basado en los estándares UML (Unified Modeling Language) y MDA (Model Driven Architecture). Originalmente era un producto comercial llamado Plastic, pero pasó a ser de código abierto.

Ofrece soporte completo para diagramas UML e ingeniería inversa, y puede generar código para lenguajes como Java, C++ y C#, así como documentos para aplicaciones de Microsoft Office.

**ArgoUML:** ArgoUML es una herramienta de modelado de sistemas utilizada para realizar diseños en UML (Unified Modeling Language) en el análisis y pre-diseño de sistemas de software. Está desarrollada en Java, por lo que es independiente de la plataforma y puede instalarse en múltiples sistemas operativos.

Proporciona la generación de código que puede ser utilizado en lenguajes como Java, C++, C#, y PHP.

**Miro:** Miro es una plataforma en línea versátil e innovadora que ayuda a mejorar la colaboración, agilizar los procesos y fomentar la creatividad en entornos empresariales y educativos.

Esta herramienta permite a equipos e individuos tener un espacio de trabajo flexible e intuitivo para la planificación colaborativa y la visualización de ideas.

Entre sus funciones destacan la creación de diagramas, esquemas y plantillas, y cuenta con más de 160 integraciones para adaptarse a diferentes necesidades.

Además, Miro ofrece la posibilidad de visualizar y desarrollar sistemas complejos con una extensa biblioteca de formas y permite que la inteligencia artificial cree diagramas de flujo, ERD, clases UML, diagramas de secuencia y más en segundos.

**Diagramas de clases:** Los diagramas de clases son representaciones gráficas que muestran las clases de un sistema y las relaciones entre ellas. Estas clases crean y operan objetos, y cada clase se representa con un rectángulo dividido en tres partes: el nombre de la clase, los atributos y los métodos.

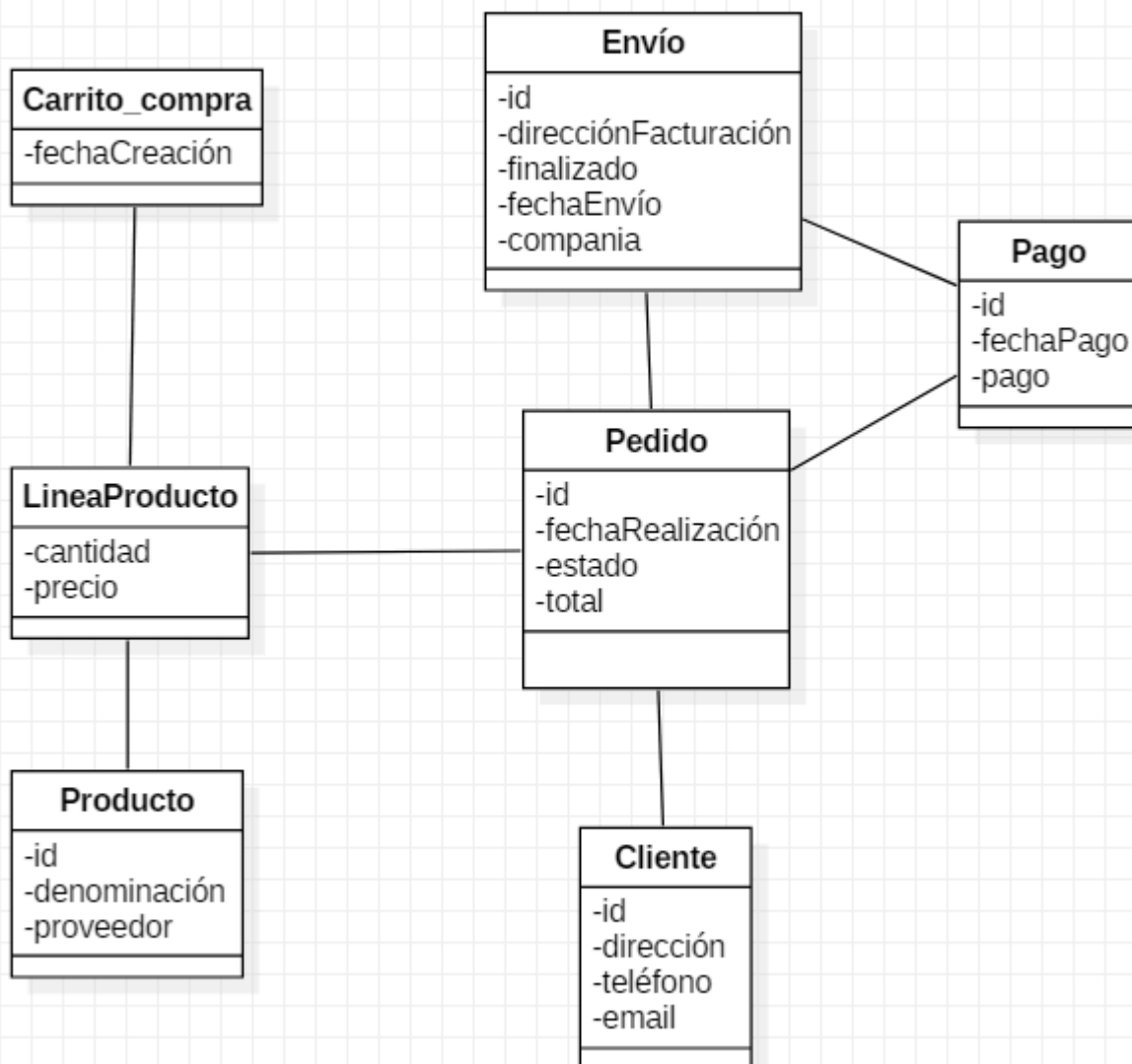
Los atributos son las características o propiedades que definen a la clase, mientras que los métodos son las acciones o funciones que puede realizar la clase.

Además, los diagramas de clases incluyen información sobre la relación entre un objeto y otro, la herencia de propiedades de otro objeto, y conjuntos de operaciones/propiedades que son implementadas para una interfaz gráfica.

Un ejemplo de un diagrama de clases puede ser un sistema administrativo hotelero, donde se pueden mostrar las relaciones entre cada objeto, incluyendo la información de huéspedes, las responsabilidades del personal y la ocupación por habitación.

Otro ejemplo puede ser un sistema de préstamos y reservas de una biblioteca, donde las clases pueden incluir libros, usuarios, préstamos y reservas, y las relaciones entre ellas pueden ser de asociación, herencia, agregación o composición.

Los diagramas de clases son esenciales en el diseño orientado a objetos y ofrecen una visualización clara y ordenada de cómo están organizadas las diferentes partes del sistema y cómo interactúan entre sí.

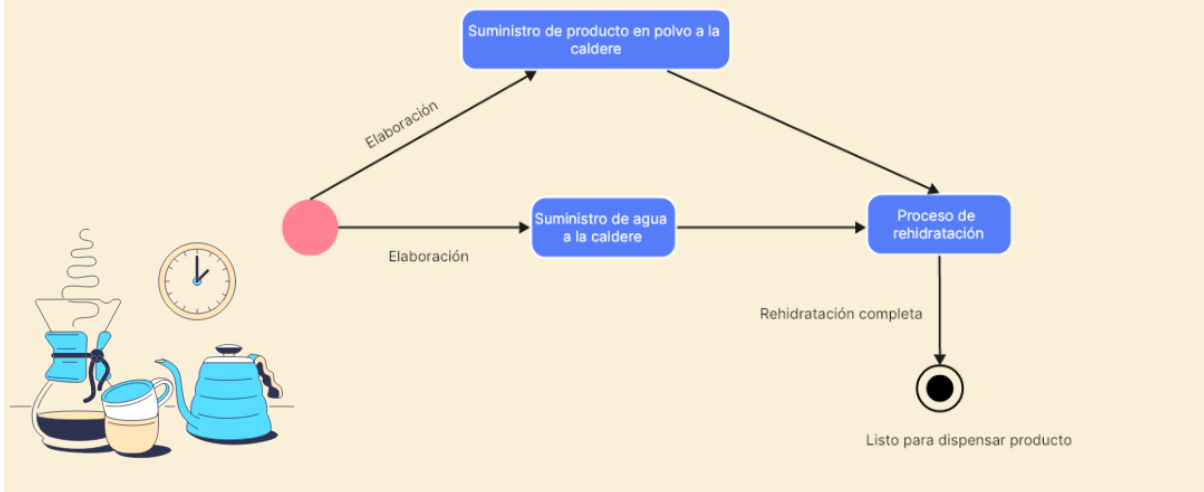


**Diagrama de estado:** Los diagramas de estado, también conocidos como diagramas de máquina de estados, son un tipo de diagrama de comportamiento en el Lenguaje Unificado de Modelado (UML) que muestran transiciones entre diversos objetos.

Estos diagramas representan estados y transiciones, mostrando cómo los elementos actúan de forma diferente según el estado del programa que se está desarrollando.

Similar a los diagramas de actividades, describen el comportamiento de objetos que se comportan de diversas formas en su estado actual.

# Diagrama de estados



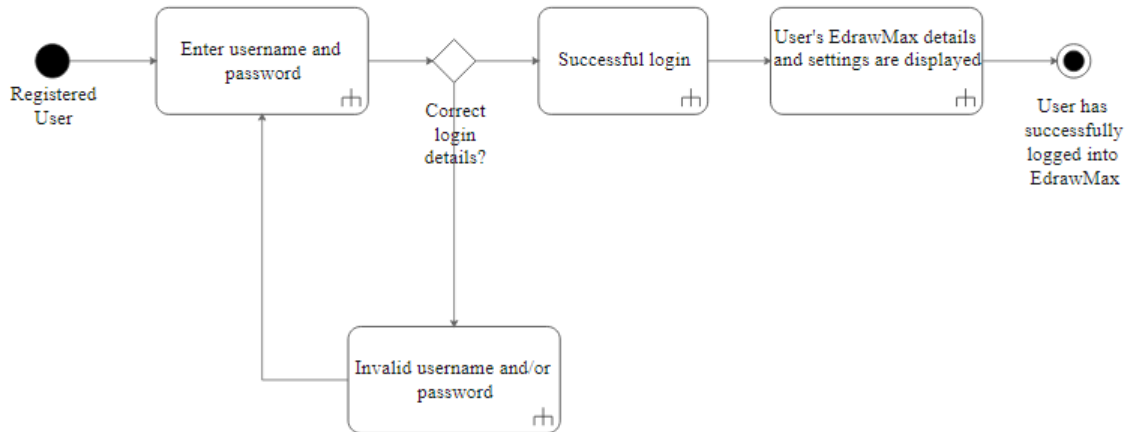
**Diagrama de actividad:** Los diagramas de actividad son herramientas gráficas utilizadas para describir el flujo de trabajo desde un punto inicial hasta un punto final, mostrando una secuencia de acciones y decisiones que pueden ocurrir secuencialmente o simultáneamente.

Estos diagramas son especialmente útiles para modelar procesos empresariales y de software, permitiendo visualizar cómo se coordinan varias actividades para proporcionar un servicio o resultado final.

Los diagramas de actividad se diferencian de los diagramas de flujo en que pueden representar procesos paralelos, mientras que los diagramas de flujo generalmente se limitan a procesos secuenciales.

En un diagrama de actividad, los nodos representan acciones, decisiones, objetos e inicio/fin, y los flujos de control muestran cómo se pasa de una actividad a otra.

### Activity diagram login example



**Diagramas de caso de uso:** Los diagramas de casos de uso son representaciones gráficas que muestran cómo los usuarios interactúan con un sistema, como una aplicación de software. Estos diagramas ayudan a esclarecer roles, tareas y responsabilidades, y se utilizan para describir los objetivos de los usuarios, las interacciones entre los usuarios y el sistema, así como el comportamiento que debe ejecutar el sistema para satisfacerlos.

En un diagrama de casos de uso, los actores representan roles que pueden incluir usuarios humanos, hardware externo u otros sistemas. Los casos de uso son una sola unidad de trabajo significativa que proporciona una vista de alto nivel del comportamiento observable para alguien o algo fuera del sistema.

