



# Diagramas en ingeniería de software

Alberto Fernández Azcoaga **UO179637**  
David Álvarez Díaz **UO283196**  
Pablo López Tamargo **UO287694**

# Que son los diagramas

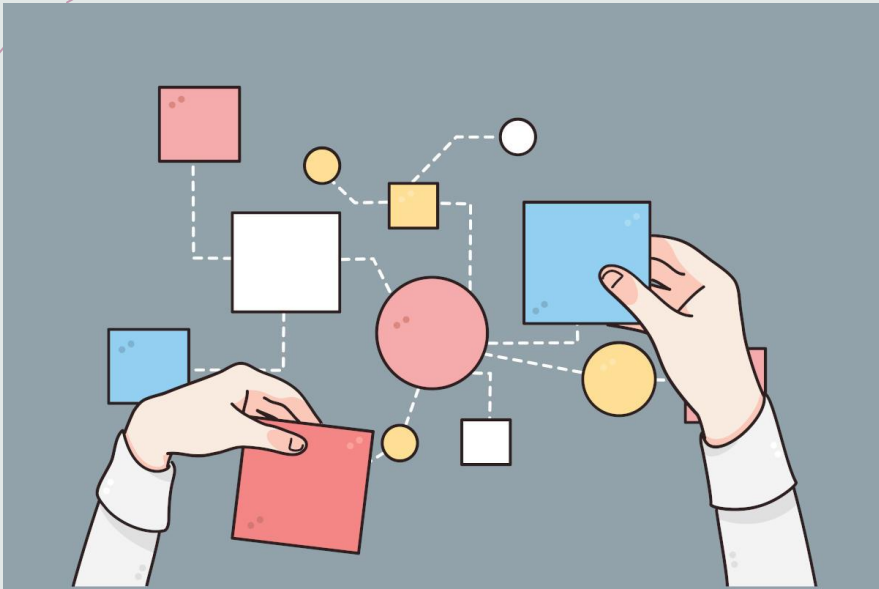
Representaciones gráficas

- Ayudan a:

- Visualizar
- Entender
- Comunicar

- Herramientas fundamentales:

- Diseñadores
- Desarrolladores
- Stakeholders



# UML (Unified Modeling Language)



James Rumbaugh  
ADOO



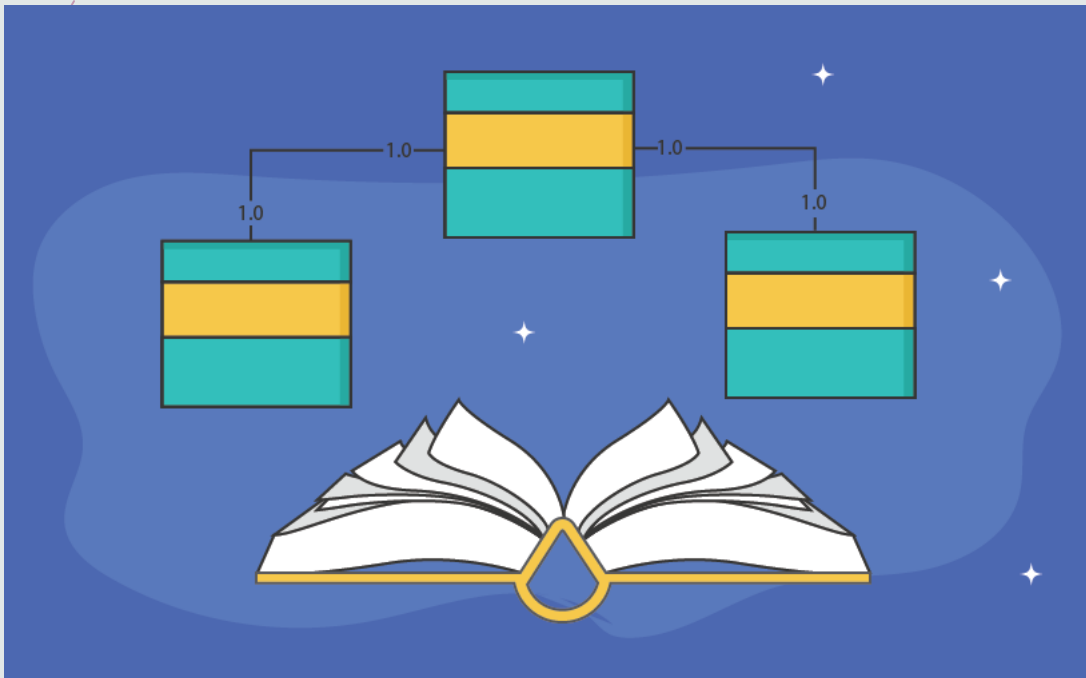
Grady Booch  
DOO








Ivar Jacobson  
SDL



# Funciones de UML

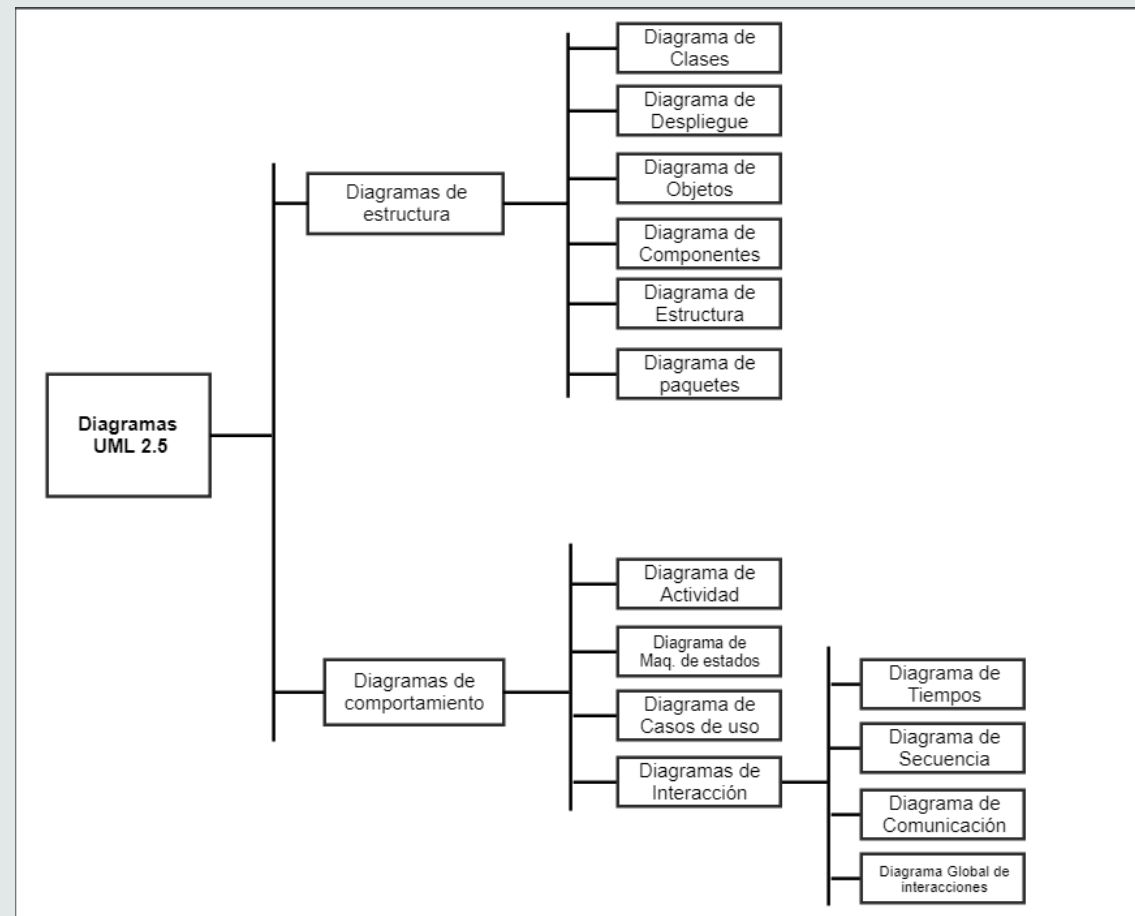


-  Modelado Visual
-  Comunicación
-  Documentación
-  Análisis y diseño
-  Estandarización

# Tipos de diagramas

Separados en dos grandes bloques:

- Diagramas de Estructura
- Diagramas de Comportamiento



# Diagramas De Estructura

## **Objetivo:**

Representar la organización y relación de los elementos estáticos en el sistema.

## **Uso:**

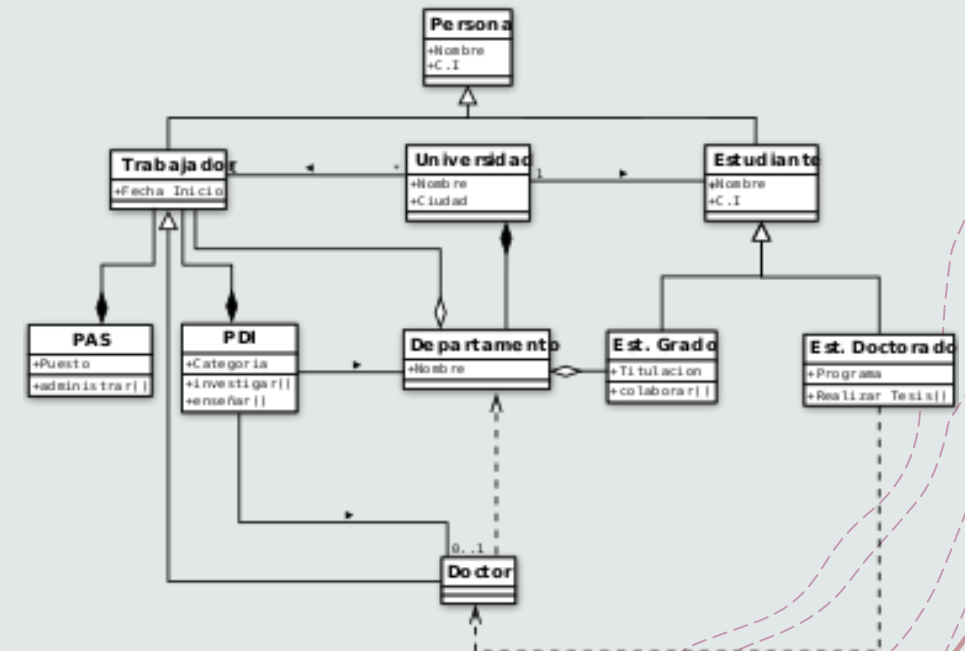
Facilitan la comprensión de la organización y la estructura estática del sistema.

Ayudan en la toma de decisiones relacionadas con la implementación y la modularidad del software.

# Diagrama de clases

Representa la estructura estática del sistema.  
Muestra clases y sus interconexiones.  
Detalla atributos y métodos de cada clase.

Diagrama de Clases

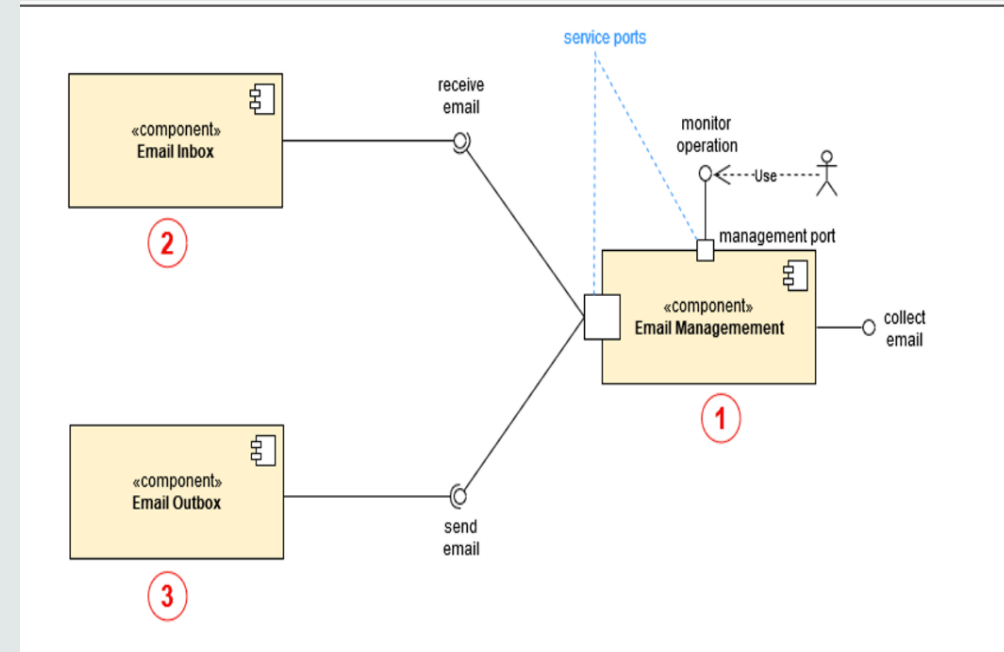


# Diagrama de Componentes

Representa componentes físicos y sus relaciones.

Enfocado en la distribución física y la arquitectura de implementación.

Muy útil para identificar microservicios y dependencias.



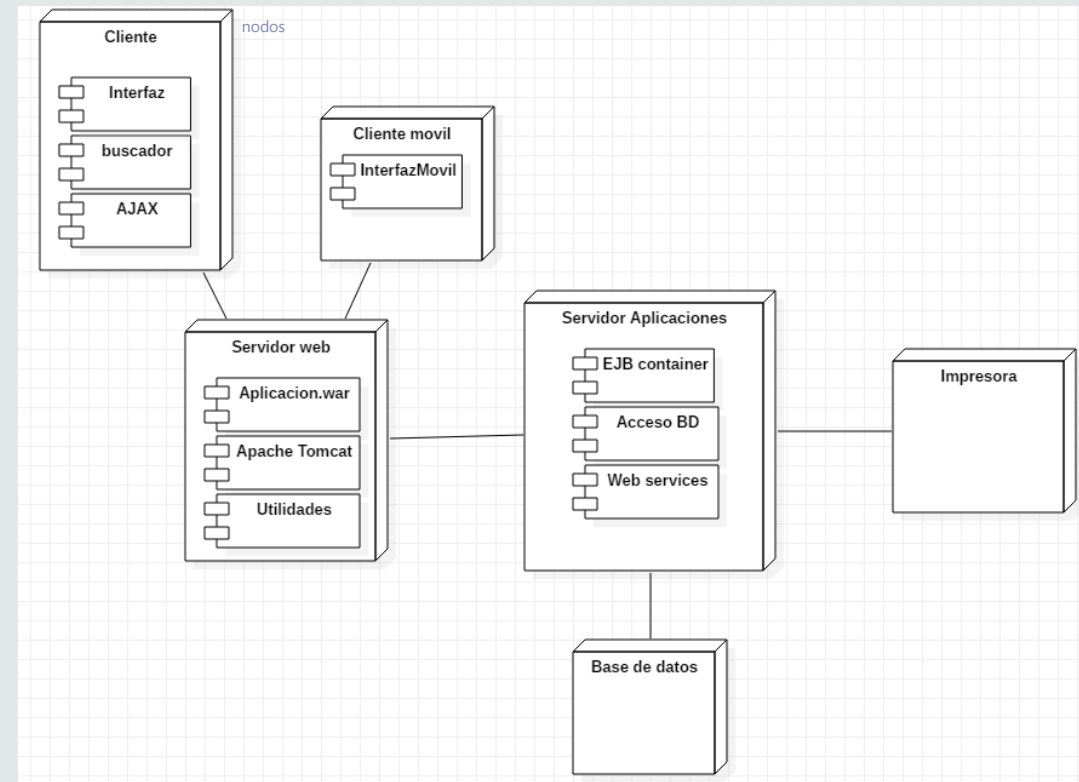
Muestra la interacción de 3 módulos software de un correo.



# Diagrama de Despliegue

Representa el despliegue físico de los componentes de un sistema.

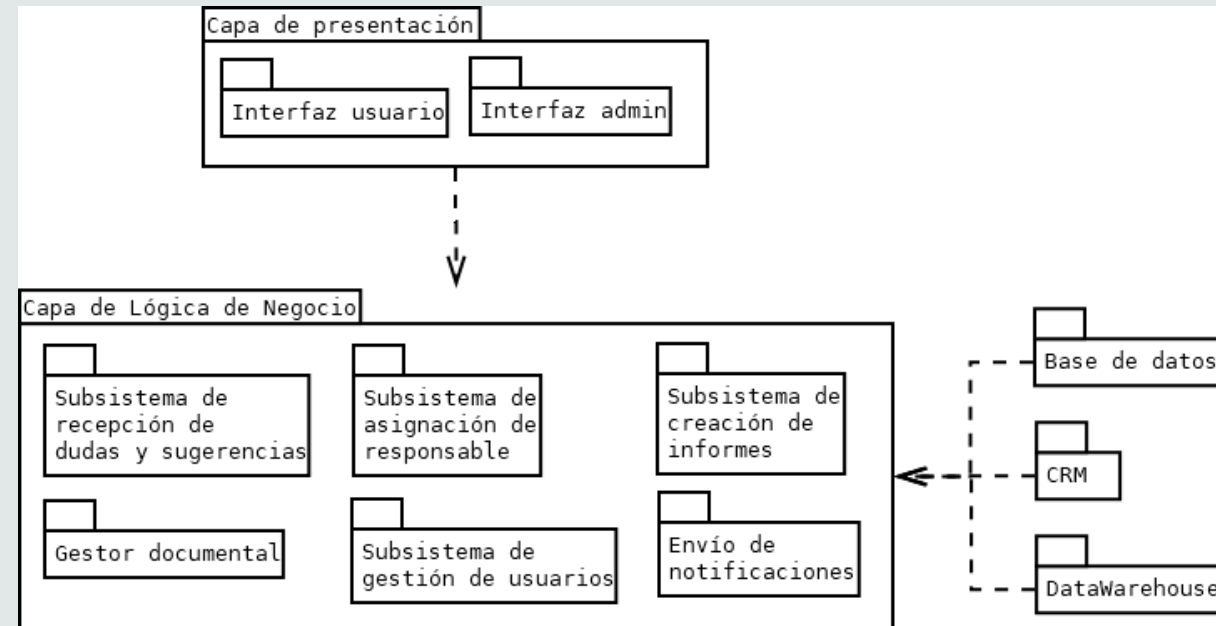
Muestra la relación del Software con el Hardware.



# Diagrama de paquetes

Agrupar funcionalidades en paquetes.

Facilita modularidad y mantenibilidad.

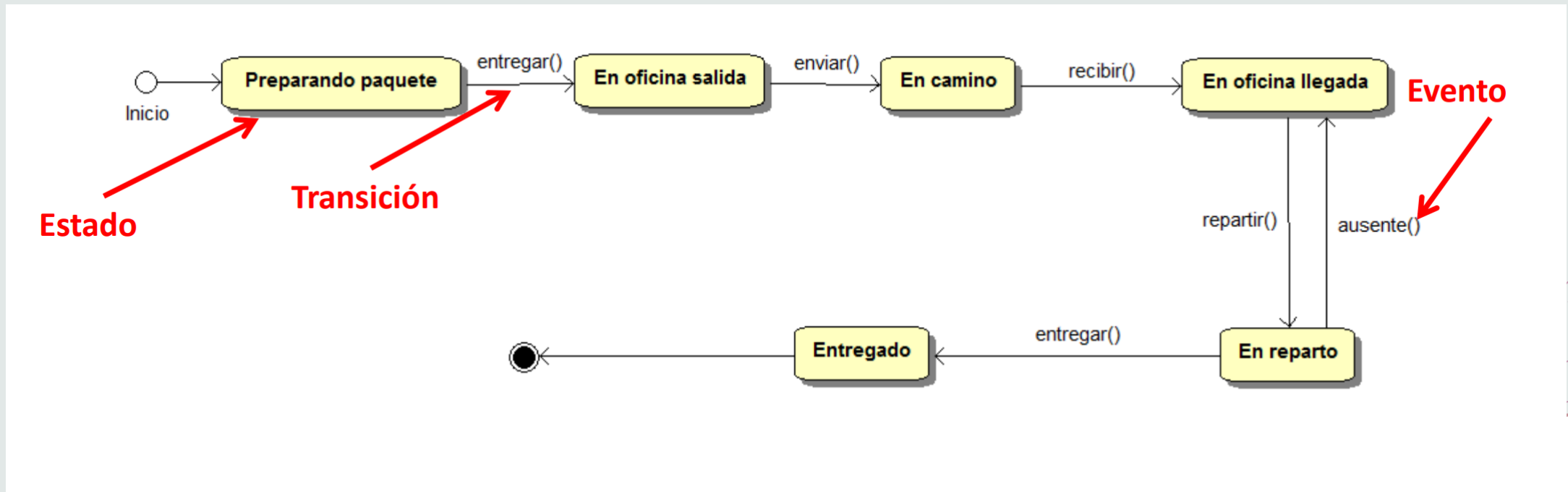


# Diagramas de Comportamiento

- Tipo de diagramas que persiguen mostrar el comportamiento dinámico de un sistema
- Reflejan como determinadas actividades del sistema cambian a lo largo del tiempo

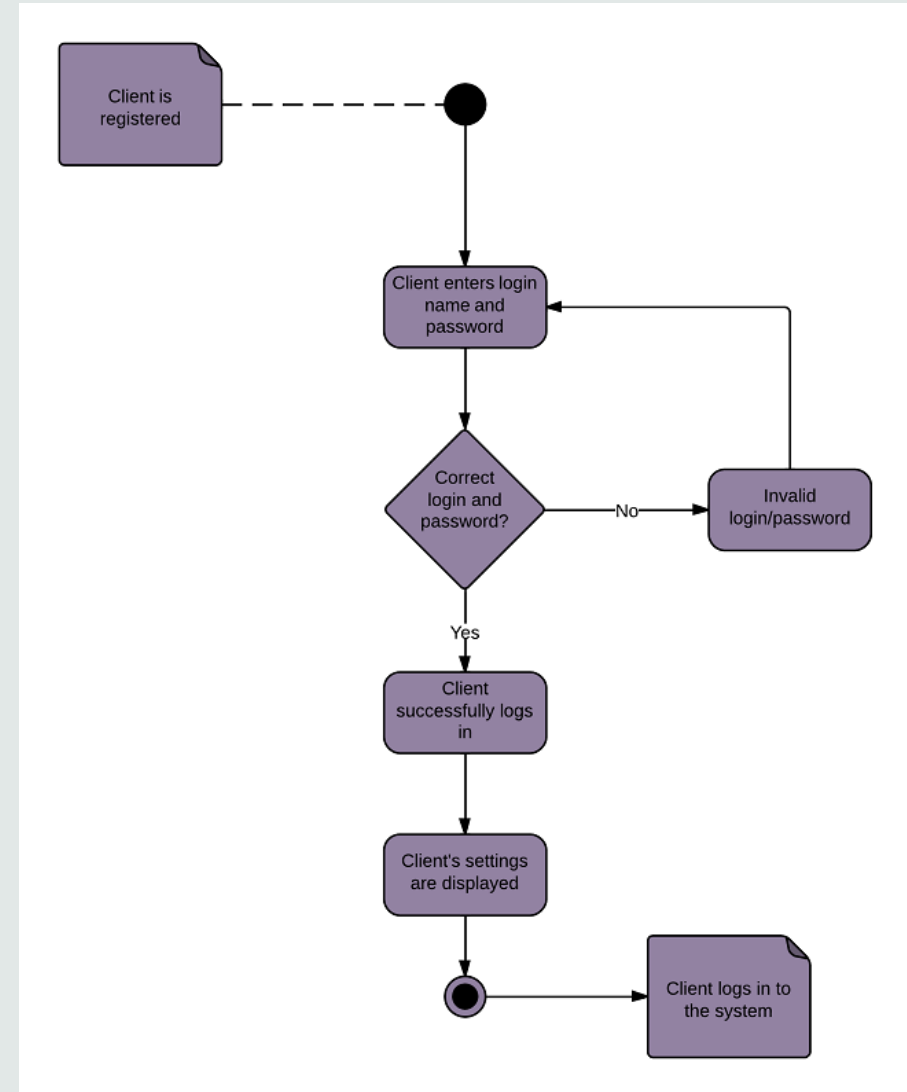
# Diagramas de Estado

Muestra los estados por los que pasa un objeto durante su vida.



# Diagrama de Actividad

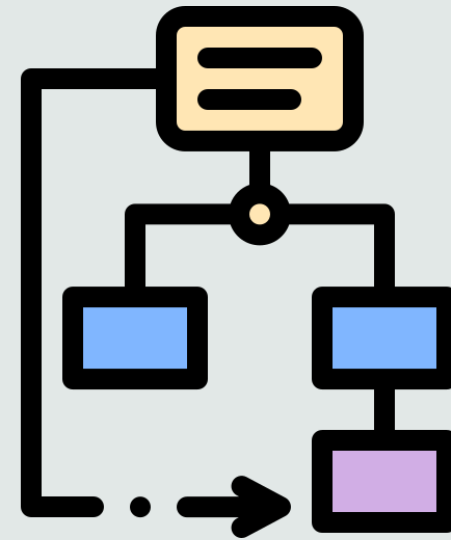
Muestra el paso a paso de las diferentes actividades del sistema.



# Herramientas



**A mano**



**Aplicaciones**

# Aplicaciones

Arreglan muchos de los inconvenientes de hacer diagramas en papel, pero mantienen las ventajas. Hay dos tipos:



**UI**



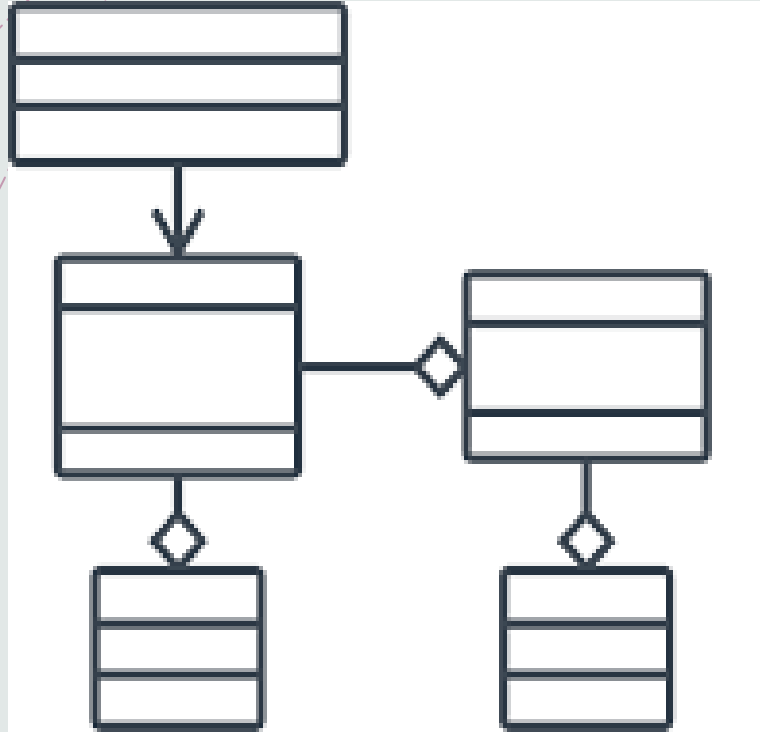
**Script**

# Aplicación UI

The image shows a screenshot of a diagramming application interface. The main workspace displays a flowchart with a rectangular box on the left connected by a line to a circular box on the right. The interface includes a top menu bar with options like 'Archivo', 'Editar', 'Vista', 'Organizar', 'Extras', and 'Ayuda'. A toolbar below the menu contains various drawing tools. On the left, there is a sidebar with a search bar 'Buscar formas' and a list of shapes categorized under 'General', 'Miscelánea', 'Avanzado', 'Básico', 'Flechas', 'Diagrama de flujo', 'Relación de la entidad', and 'UML'. On the right, a 'Diagrama' panel is open, showing settings for 'Vista' (Grid, Page view, Background, Shadow, Sketch), 'Opciones' (Connection arrows, Connection points, Guides), and 'Tamaño del papel' (A4, Orientation). The status bar at the bottom indicates 'Página-1'.



¿Qué pasa si queremos meter una nueva funcionalidad en un sistema ya estructurado?



**Sistema completo**



**Nueva funcionalidad**

En el peor de los casos, tenemos que reestructurar el diagrama entero.

# Aplicación script

```
@startuml
Alice -> Bob: Authentication Request
Bob --> Alice: Authentication Response

Alice -> Bob: Another authentication Request
Alice <-- Bob: another authentication Response
@enduml
```



Sintaxis no muy compleja. Diagramas mantenibles.



**¿Alguna pregunta?**