



**I.E.S Jacarandá**

---

# **Unidad 1 Desarrollo de Software: Elementos**

**1º DAW**

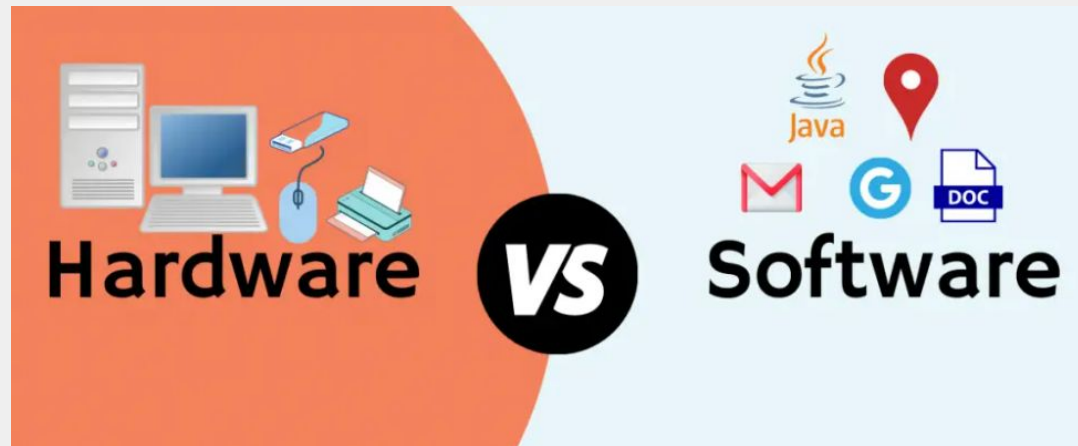
---

---

# Objetivos

- Describir qué software y su clasificación
- Conocer el ciclo de vida de desarrollo software.
- Conocer los distintos tipos de lenguaje de programación
- Entender y conocer el proceso de obtención de código ejecutable

# 1.1 Introducción



En cualquier sistema informático podemos distinguir:

- El hardware: componentes físicos
- El software: parte lógica del ordenador. Comprende el conjunto de programas y aplicaciones que actúan sobre el hardware del ordenador y facilitan al usuario la realización de diferentes tareas.

En este capítulo se estudiarán conceptos básicos que tienen que ver con el desarrollo de software

# 1.2.1 El software y tipos

Es el conjunto de programas informáticos que interactúan sobre el hardware para ejecutar una tarea.

## DE SISTEMAS

- + Gestiona la comunicación entre hardware y software.
- + Interfaz de alto nivel.
- + SSOO y controladores de dispositivos

## DE PROGRAMACIÓN

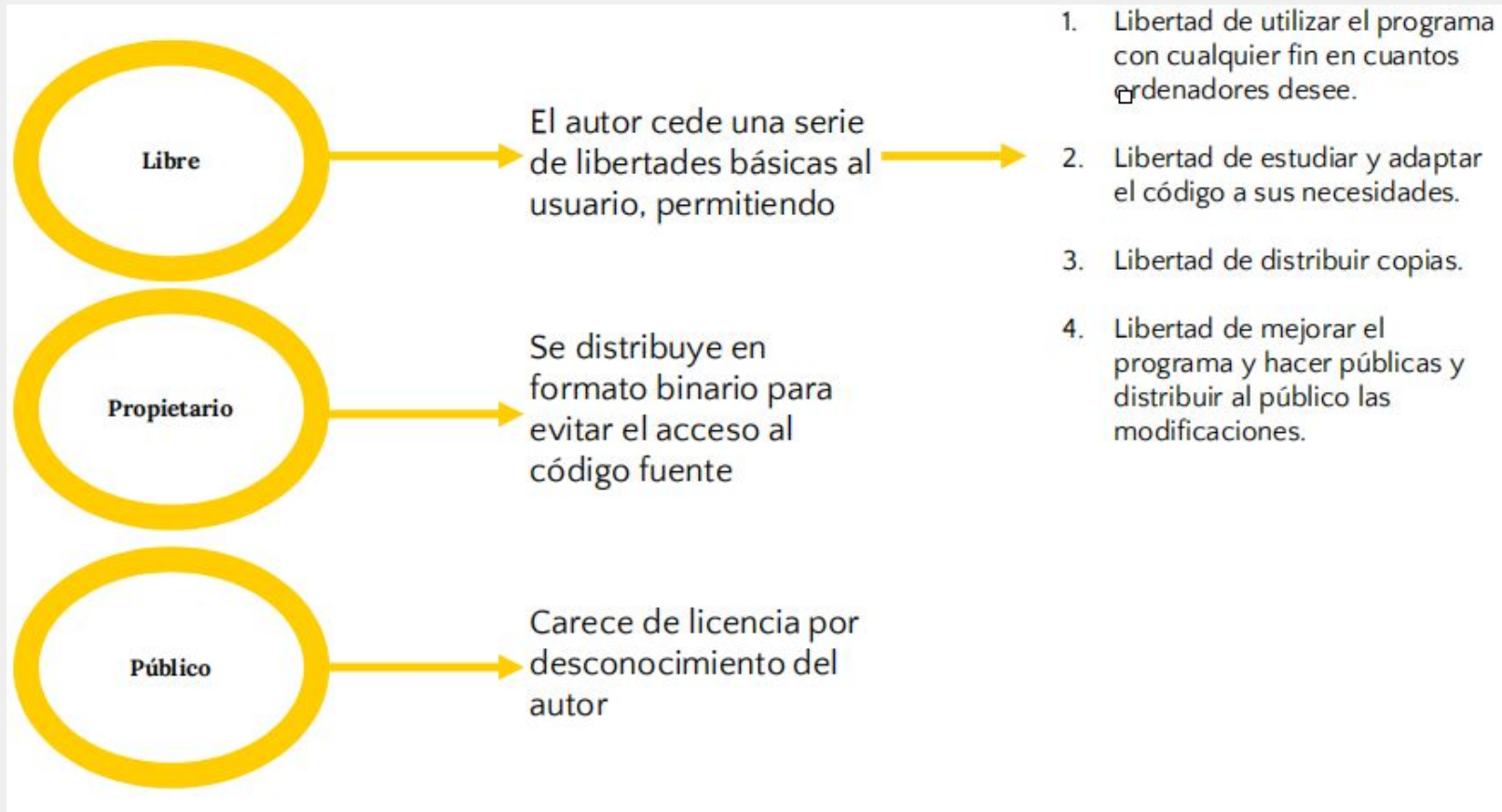
- + Herramientas para que el usuario desarrolle aplicaciones informáticas.
- + Entornos Integrados de Desarrollo (IDE), compiladores, ...

## DE APLICACIÓN

- + Permite al usuario realizar una o varias tareas específicas.
- + Apps ofimáticas, bases de datos, juegos, ...

## 1.2.2 El software: licencias

Una licencia de SW es un contrato que se establece entre el desarrollador de un SW sometido a propiedad intelectual y a derechos de autor y el usuario, en el cual se definen con precisión los derechos y deberes de ambas partes.



# 1.2 Ciclo de vida del software

El ciclo de vida de un producto SW comprende el período que transcurre desde que la primera idea sobre él hasta que el producto deja de estar disponible o es retirado (deprecated)

Se compone de varias etapas. Cada etapa tiene como entrada uno o varios productos o documentos procedentes de las etapas anteriores y produce otros documentos de salida. Es por ello que es tan importante la documentación de cada etapa.

A continuación, puedes observar las etapas más usuales:

# 1.2 Ciclo de vida del software



# 1.2.1 Modelos de Ciclo de Vida

Existen varios modelos de ciclo de vida, es importante tener en cuenta las características del proyecto SW para elegir un modelo u otro:

Modelos más importante:

- Cascada
- Incremental
- Evolutivo



# 1.2.1 Modelos de Ciclo de Vida

## Modelo en Cascada.

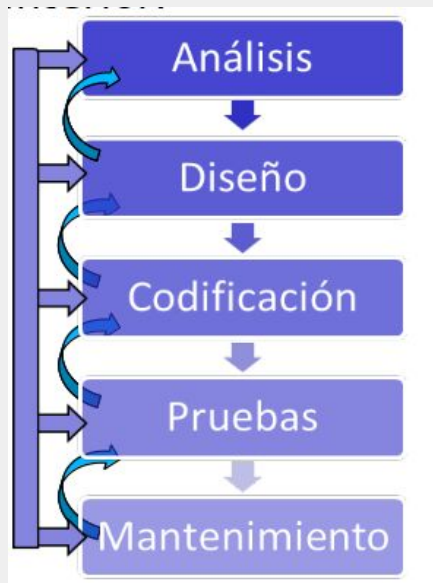
Las etapas para el desarrollo del SW tienen un orden, de tal forma que para empezar una etapa es necesario finalizar la etapa anterior; después de cada etapa se realiza una revisión para comprobar si se puede pasar a la siguiente



# 1.2.1 Modelos de Ciclo de Vida

## Modelo en Cascada con Retroalimentación.

Es una de las variantes del modelo en cascada más utilizada. Produce una retroalimentación entre etapas, pudiendo volver de una etapa a la anterior o incluso varias etapas a la anterior.



Ejemplo:

Supongamos que la etapa de Análisis (toma de requisitos) ha finalizado durante la fase de diseño se detectan fallos entonces será necesario retornar a la etapa anterior, realizar lo ajustes pertinentes y continuar de nuevo con el Diseño.

# 1.2.1 Modelos de Ciclo de Vida

## Modelo en Cascada

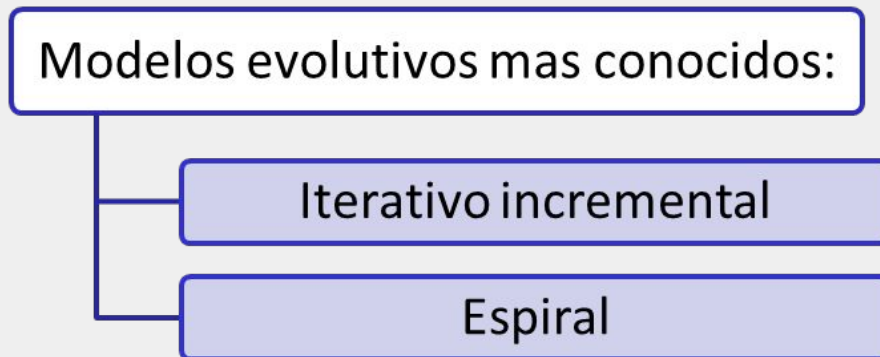
VENTAJAS	INCONVENIENTES
<ul style="list-style-type: none"><li>• Fácil de comprender, planificar y seguir</li><li>• La calidad del producto resultante es alta.</li><li>• Permite trabajar con personal poco cualificado.</li></ul>	<ul style="list-style-type: none"><li>• La necesidad de tener todos los requisitos definidos desde el principio (algo que no siempre ocurre ya que pueden surgir necesidades imprevistas).</li><li>• Es difícil volver atrás si se cometen errores en una etapa.</li><li>• El producto no está disponible para su uso hasta que no está completamente terminado.</li></ul>
SE RECOMIENDA CUANDO:	
<ul style="list-style-type: none"><li>• El proyecto es similar a alguno que ya se haya realizado con éxito anteriormente.</li><li>• Los requisitos son estables y están bien comprendidos.</li><li>• Los clientes no necesitan versiones intermedias.</li></ul>	

# 1.2.1 Modelos de Ciclo de Vida

## Modelo en Evolutivos

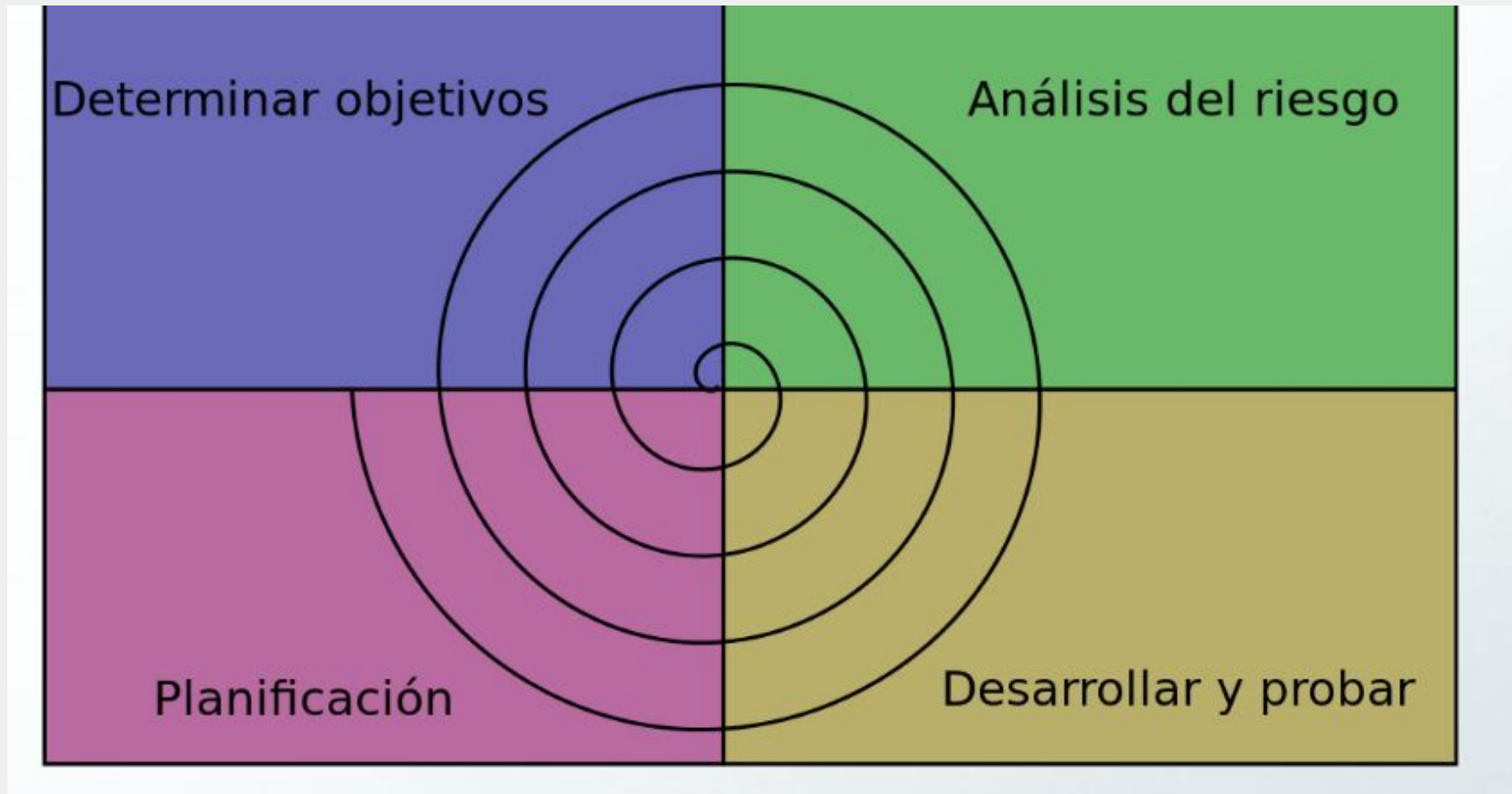
Al contrario del modelo en cascada que asume que se va a entregar un producto completo, los modelos evolutivos permiten desarrollar versiones cada vez más completas hasta llegar al producto final deseado.

En estos modelos se asume que las necesidades del usuario no están completas y se requiere una vuelta a planificar y diseñar después de cada implantación de los entregables.



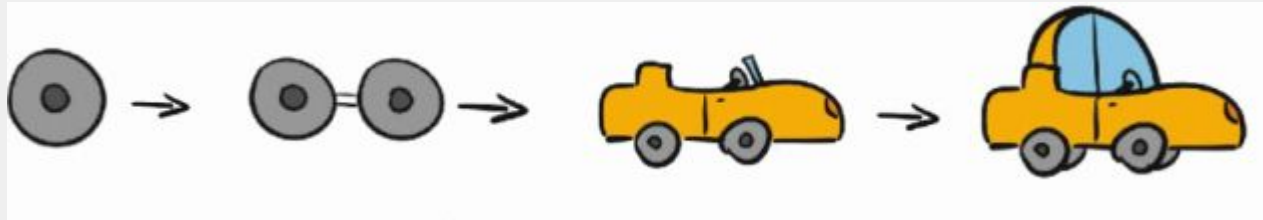
# 1.2.1 Modelos de Ciclo de Vida

## Modelo en Evolutivos: Modelo en Espiral

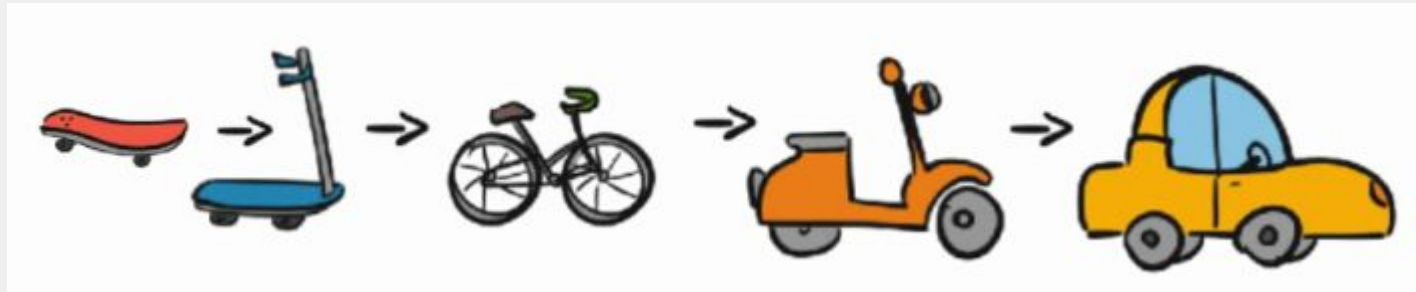


# 1.2.1 Modelos de Ciclo de Vida

## Modelo Incremental



Modelo en Espiral

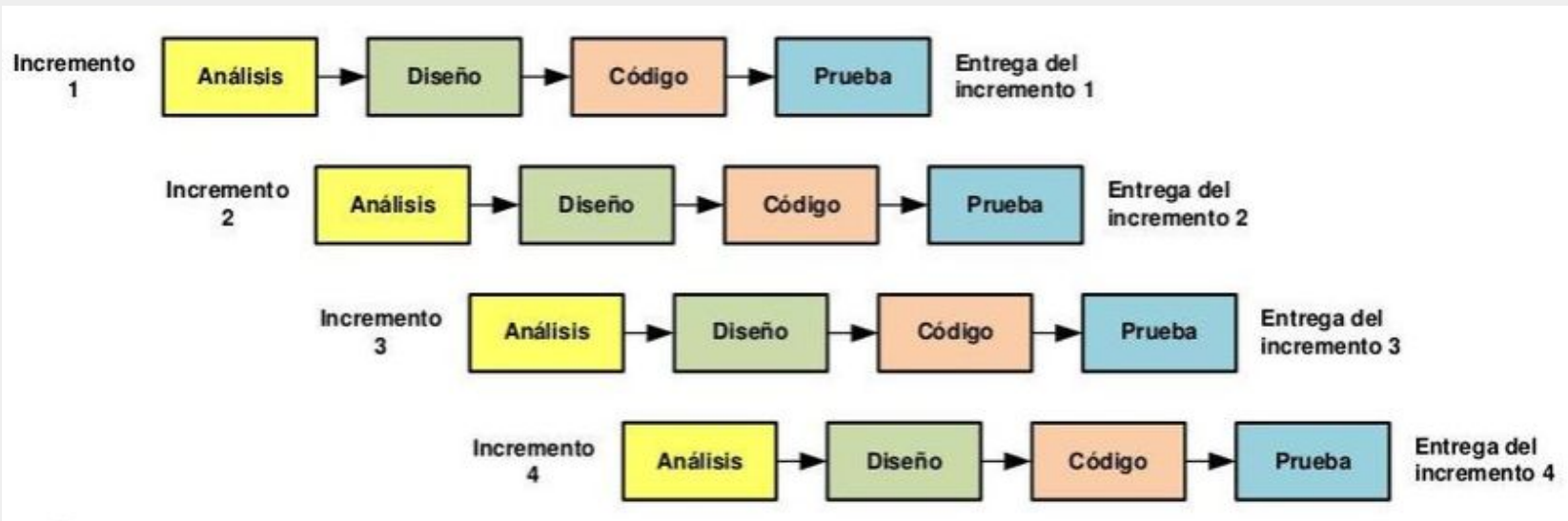


Modelo Incremental

# 1.2.1 Modelos de Ciclo de Vida

## Modelo Incremental

Se definen diferentes fases, en cada una de ellas se define el producto mínimo con valor para negocio. Minimum Value Product (MVP)



# Investiga:

Busca información sobre metodologías ágiles. ¿Qué son y qué ventajas tienen respecto a las metodologías clásicas?

Indica algunas de estas metodologías y explica un poco de cada una.

¿Cuál te parece más interesante o beneficiosa?

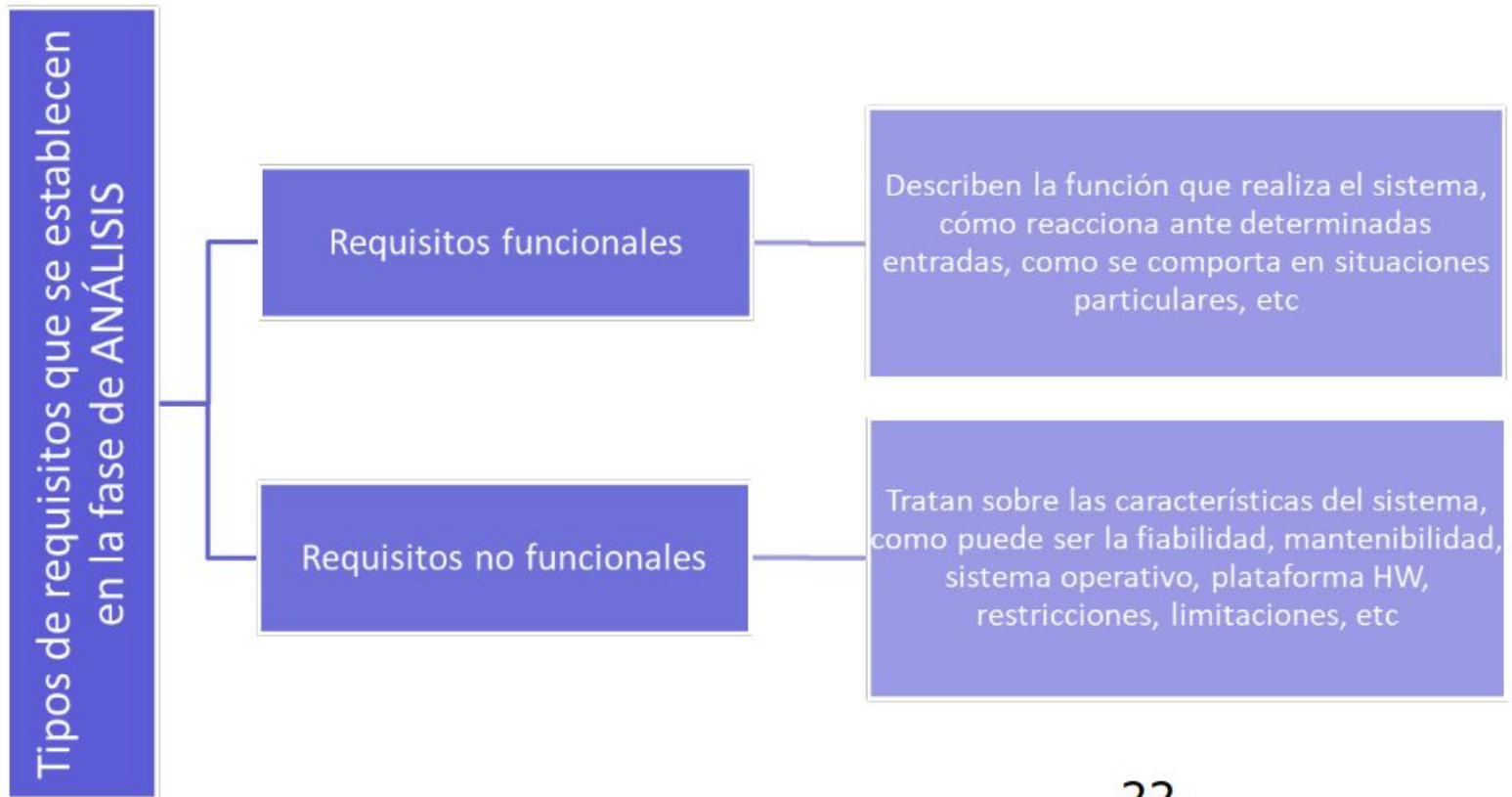
Busca información sobre la metodología SCRUM. Describe:

- Describe brevemente en qué consiste
- Qué tipo de ciclo de vida es
- Qué roles existen



# 1.2 Ciclo de vida del software: Análisis

En esta fase se analizan y especifican los requisitos o capacidades que el sistema debe tener porque el cliente así lo ha pedido.



# 1.2 Ciclo de vida del software: Análisis

Tarea:

- **Actividad de Grupo.** Dos alumnos salen a la pizarra. La clase plantea una solicitud de un proyecto determinado y entre todos indicamos algunos requisitos funcionales y no funcionales.

Documento con conclusiones

1. Explica en una frase en qué consiste el proyecto
2. Indica 3 posibles requisitos funcionales y 3 no funcionales.

# 1.2 Ciclo de vida del software: Análisis

Para representar los requisitos se utilizan diferentes técnicas:



# 1.2 Ciclo de vida del software: Diseño

Ya sabemos qué hacer... Ahora nos planteamos: ¿Cómo se hace?

