



## *ENTORNOS DE DESARROLLO: ACTIVIDAD 2.*

## UNIDAD 1

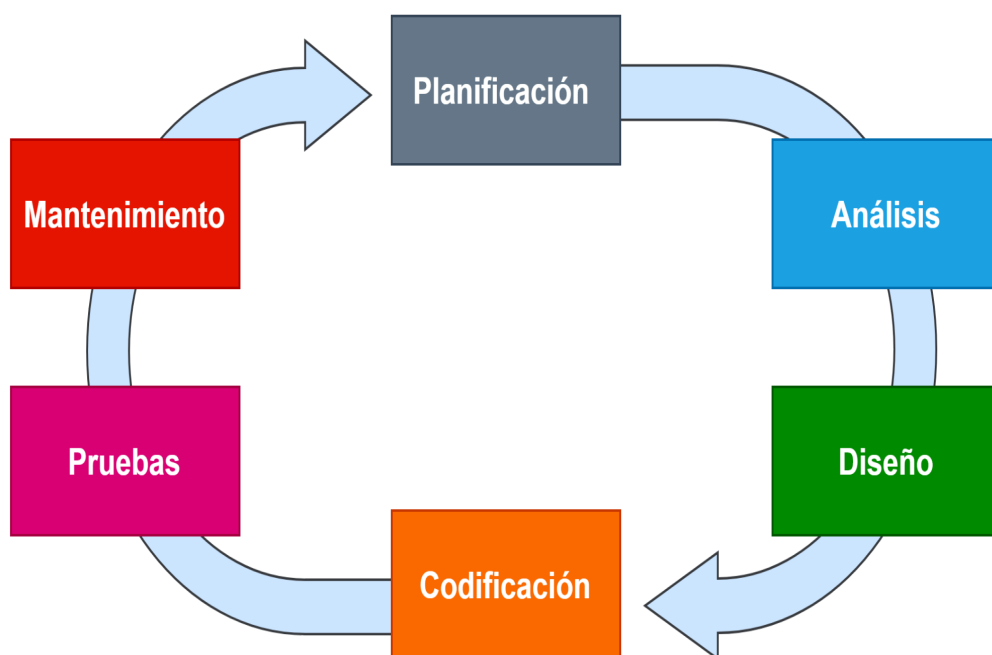
### ACTIVIDAD 2

#### 1.- Define "Ciclo de vida del software".

El **ciclo de vida del software** o Systems Development Life Cycle son las fases que se llevan a cabo a la hora de desarrollar un software. Estas fases nos permiten administrar, y validar el proyecto, así como garantizar que cumpla con los requisitos necesarios.

Existen diferentes tipos de ciclos de vida, entre ellos están:

- Modelo en cascada.
- Modelo en V
- Modelo en espiral
- Modelo de construcción de prototipos
- Modelo en espiral
- Metodologías ágiles (o adaptativos)



#### 2.- Nombra las fases principales del desarrollo de software y explica brevemente qué se hace en cada una de ellas.

Aunque en cada modelo de desarrollo pueden variar las fases, un ciclo de vida básico contiene las siguientes fases:

### Fase 0 - Planificación.

Algunas de las tareas que se llevan a cabo en esta fase son algunas de las más importantes, ya que una buena planificación evitará errores y retrasos a lo largo del proyecto. Planificación temporal, asignación de recursos para cada etapa, estimación de los costes, estudio de la viabilidad del proyecto.

### Fase 1 - Análisis.

En esta fase de análisis se determinan los requisitos que deberá tener el software. Es este momento se pone de manifiesto que es lo que realmente se necesita y que requerimientos hace el cliente. También se orienta tanto al cliente como al propio proyecto hacia lo que mejor vaya a funcionar para lo que se necesita.

Aquí se ponen las bases sobre las características del sistema, de la manera más específica posible.

### Fase 2 - Diseño.

En esta siguiente fase se debe decidir cuál será la estructura del software. Así como dividir las partes en elementos que puedan ser desarrollados por separado.

Aparte de la estructura del software escogiendo de manera adecuada las diferentes herramientas que se van a destinar a cada elemento, también se deben decidir tanto qué lógica van a seguir los datos, así como su interfaz. También se debe escoger el entorno de desarrollo y el lenguaje de programación más adecuado.

### Fase 3 - Codificación.

En esta fase de codificación, se debe escribir el código fuente con el lenguaje adecuado

### Fase 4 - Pruebas.

Una vez que el código ha sido creado hay que comprobar que el programa ejecute las tareas especificadas. Además, en esta fase se detectan fallas y se analiza el rendimiento del software. Tras las pruebas también se puede llevar a cabo una implementación del programa, es decir, habilitarlo para que el usuario lo utilice en los diferentes entornos sin problemas de compatibilidad.

### Fase 5 - Mantenimiento.

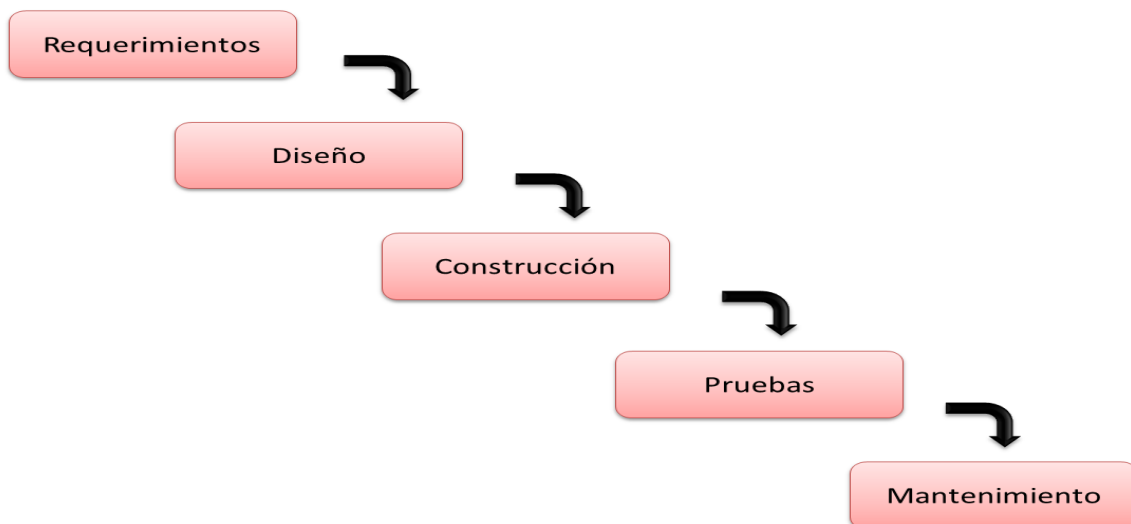
Tras las pruebas realizadas, y puesta en marcha del programa, hay que mantenerlo, optimizarlo, y mejorarlo para eliminar los errores detectados, adaptar nuevas funciones o requerimientos. Es decir, actualizarlo.

### 3.- Explica brevemente en qué consiste el modelo en cascada cuando hablamos de desarrollo de software.

El modelo de desarrollo en cascada es un tipo de procedimiento lineal se caracteriza por dividirse en fases que se suceden entre sí, las cuales se ejecutan tan solo una vez a lo largo del proyecto. La fase anterior es el punto de partida para la siguiente.

Las fase que se plantean en él son las siguiente:

1. Requisitos del sistema y del software.
2. Análisis.
3. Diseño.
4. Construcción o codificación.
5. Pruebas
6. Mantenimiento.



### 4.- Ventajas e inconvenientes del modelo en cascada.

VENTAJAS	DESVENTAJAS
Estructura sencilla. Fases muy diferenciadas.	Los proyectos más complejos o de varios niveles no permiten su división en fases de proyecto claramente diferenciadas.
Buena documentación del proceso en cada fase. Verificación tras cada fase.	Poco margen para ajustes si cambian las exigencias.
Los costes y la carga de trabajo se pueden calcular al comienzo.	El usuario final no se integra hasta que no se termina la programación.

### 5.- ¿Qué se entiende por verificación? ¿Y por validación?

La verificación son las pruebas que se realizan en cada paso del proceso que tienen lugar en cada etapa del ciclo de vida para asegurarse que se han cumplido los requisitos necesarios o bien para realizar las correcciones necesarias.

Por otro lado la validación, se basa en las verificaciones y la documentación que se aportó en ellas. En la validación se lleva a cabo pruebas completas del proyecto y a diferencia de la verificación esta se lleva a cabo después de haber sido instalado el software.

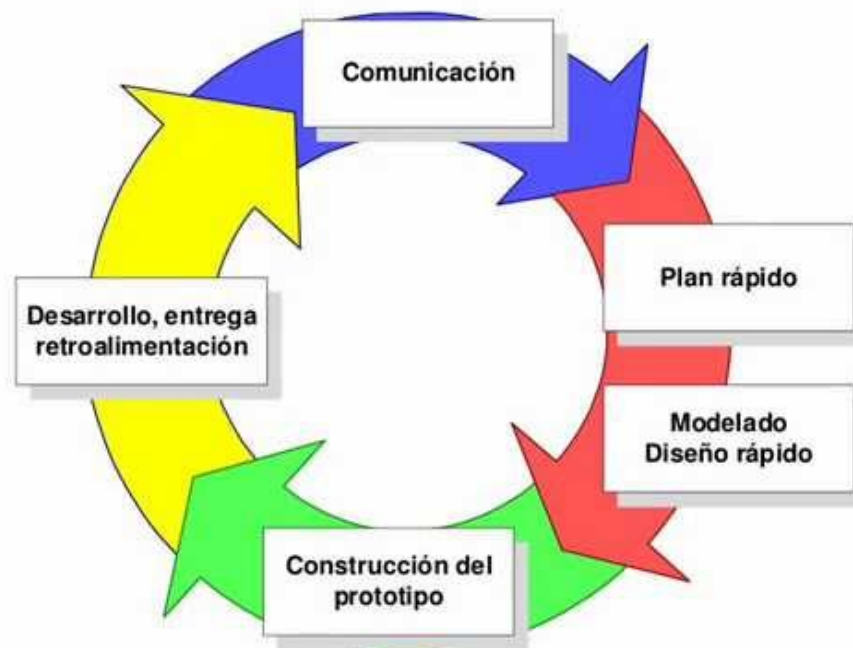
### 6.- Explica cómo funciona el modelo de desarrollo mediante creación de prototipos.

El modelo de prototipos se centra en un **diseño rápido que representa las características principales del programa** que el usuario podrá utilizar. De esta manera pueden probarlo y dar su opinión sobre distintos aspectos como la usabilidad, la utilidad o el rendimiento.

El prototipo se puede **modificar cuando sea necesario** y todos los resultados obtenidos de las presentaciones y pruebas se deben anotar para utilizar posteriormente como ayuda en el desarrollo del producto final.

Fases del modelo de construcción de prototipos:

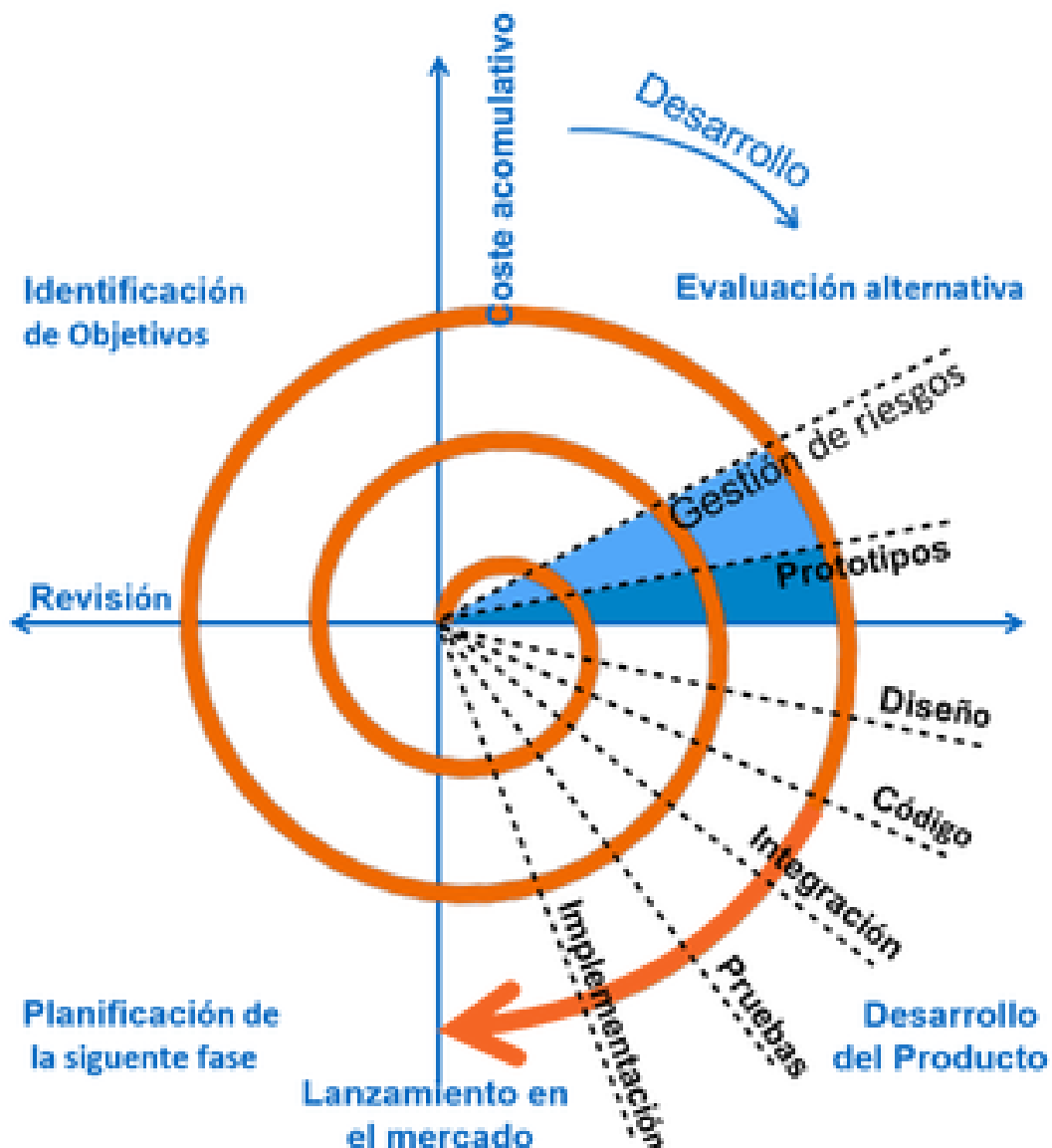
- 1 - Requisitos.
- 2 - Modelajes y desarrollo de código.
- 3 - Evaluación.
- 4 - Modificación.
- 5 - Documentación.
- 6 - Pruebas.



## 7.- Explica cómo funciona el modelo espiral cuando se aplica al desarrollo orientado a objetos.

Este es un modelo que está basado en el riesgo. A diferencia de un modelo lineal como es el modelo en cascada este modelo tiene un ciclo de vida por medio de espirales que hay que recorrer. Este enfoque se parece bastante a la creación de prototipos. Al centrarse en la minimización del riesgo, mediante el análisis y la evaluación de riesgos, este modelo tiene un componente financiero que puede ser muy relevante.

El progreso del proyecto va a depender de cómo se vayan resolviendo los riesgos identificados. El objetivo de este modelo es producir un proyecto en continua mejora desde el inicio.



**8 - Webgrafía.**

- <https://intelequia.com/es/blog/post/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>
- <https://evotic.es/software-a-medida/ciclo-de-vida-del-software/>
- <https://intelequia.com/es/blog/post/ciclo-de-vida-del-software-todo-lo-que-necesitas-saber>
- <https://raknarrok.blogspot.com/2015/05/cap-3-software-development-life-cycle.html>
- <https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/el-modelo-en-cascada/>
- <https://www.emerson.com/documents/automation/training-sis-203-verification-validation-on-es-es-41624.pdf>
- [https://es.ryte.com/wiki/Modelo\\_en\\_Espiral#Informaci.C3.B3n\\_general](https://es.ryte.com/wiki/Modelo_en_Espiral#Informaci.C3.B3n_general)