

TEMA2

INGENIERIA DEL SOFTWARE

Entornos de Desarrollo
1 DAM/DAW





Índice

1. Introducción.
2. Ciclo de vida del software.
3. Fases del desarrollo.
4. Caso práctico 1: “Fases del desarrollo”.
5. Metodologías del desarrollo del software.
6. Roles en el desarrollo del software.
7. ¿Cómo hacer un plan de proyecto?



Definiciones importantes



REPASO

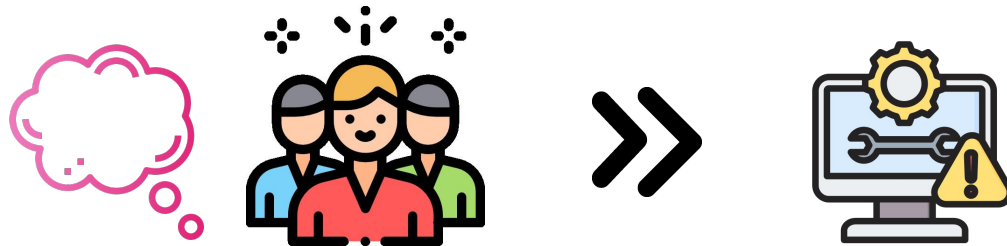
- Programa:** conjunto de pasos o instrucciones que indican al ordenador cómo realizar un proceso.
- Ejecutar:** consiste en iniciar un programa y ponerlo en ejecución.
- Entorno de desarrollo (IDE):** herramienta que facilita y posibilita el desarrollo de software.
- Código fuente:** código de un programa, escrito por un programador en un lenguaje de programación.
- Compilar:** proceso que traduce el código fuente en código objeto (comprensible por un ordenador).
- Código objeto:** código máquina generado tras la compilación del código fuente.
- Librería:** código externo que se incluye al programa para proporcionar funcionalidad adicional.
- Archivo ejecutable:** programa ejecutable que puede ser ejecutado en el ordenador.
- Lenguaje máquina:** lenguaje que sólo es comprensible por el ordenador (0101011).
- Instrucción:** cada una de las sentencias u órdenes que forman parte de un programa.
- Palabra reservada:** cada uno de los símbolos (léxico) que componen la sintaxis de un lenguaje.
- Semántica:** reglas que definen la combinación de los símbolos de un lenguaje de programación.

1. Introducción



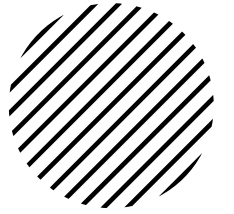
Ingeniería del software = disciplina formada por un conjunto de métodos, herramientas y técnicas que se utilizan en el desarrollo de programas informáticos.

Muchos proyectos fracasan durante el desarrollo ➡ ¿POR QUÉ?



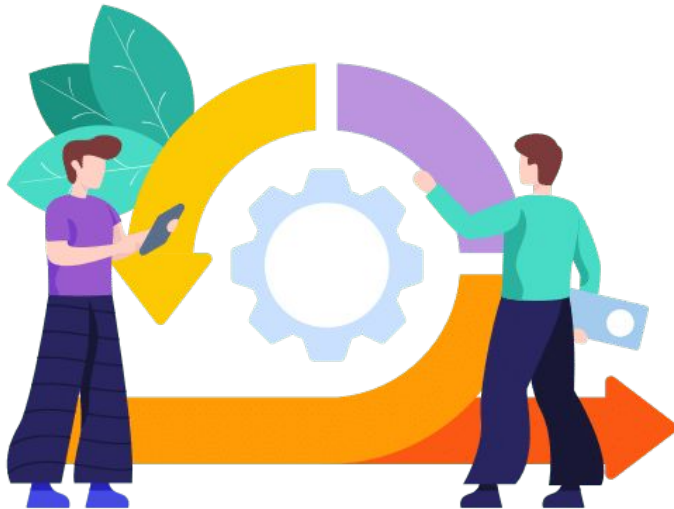
IMPORTANTE

Establecer un proceso que ayude a realizar cada una de las etapas que requiere la implementación de cualquier aplicativo.



2. Ciclo de vida del software

El **ciclo de vida del software** se refiere al proceso general y completo que sigue un proyecto de desarrollo de software desde su **concepción** hasta su finalización y **mantenimiento o retirada**. Es una representación de alto nivel que abarca todas las etapas, actividades y decisiones que involucran la creación, entrega y mantenimiento de un software.

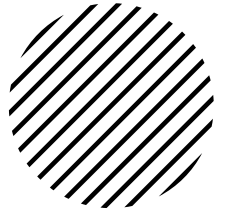


Hace más referencia a la metodología de trabajo que seguimos.





3. Fases del desarrollo del software

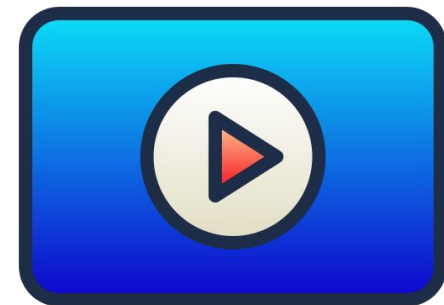


Las **fases del desarrollo del software** son las etapas específicas que un proyecto de desarrollo de software atraviesa dentro de su ciclo de vida. Estas fases representan los pasos concretos que deben completarse para llevar a cabo el desarrollo del software. Las fases pueden variar según la metodología de desarrollo utilizada, pero típicamente incluyen:

VUESTRO TURNO

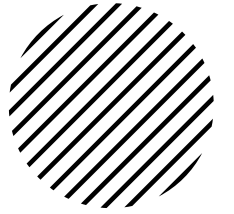


**ORDENA LAS SIGUIENTES
TARJETAS**



<https://www.youtube.com/watch?v=Z9LNk12ndm4>

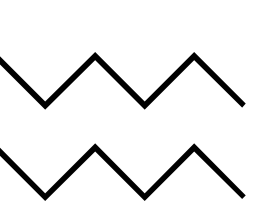




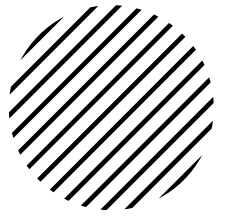
3. Fases del desarrollo

- 1) **Análisis:** En esta fase se crea un diagrama sencillo en el que especificamos cada uno de los requisitos que nos ha indicado el cliente pudiendo también tener alguna relación, por ejemplo, a base de datos. Análisis de restricciones, entrevistas con el cliente y los usuarios finales. Se genera un documento vinculante (a modo de contrato) entre el cliente y el desarrollador
- 1) **Diseño:** Consiste en determinar los requisitos generales de la arquitectura de la aplicación. Se define cada subconjunto de la aplicación. Los documentos son mucho más técnicos. Esta fase involucra a los jefes de proyecto, arquitectos de software y analistas. Los programadores aún no intervienen en esta fase.
- 1) **Implementación:** En este punto nos centraremos en codificar todo lo diseñado previamente para tener una a primera aproximación al software.
- 1) **Pruebas:** Las pruebas de nuestra aplicación deben permitir identificar posibles errores que hayamos cometido a la hora de desarrollar. Se distingue entre prueba funcional (con el cliente, se revisa de qué hace lo que se ha definido) y prueba estructural (pruebas técnicas del desarrollo).

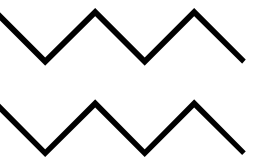




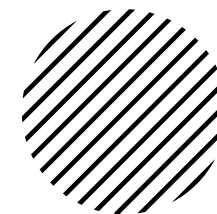
3. Fases del desarrollo



- 5) **Documentación**: En la fase de documentación es necesario dejar por escrito todas las decisiones tomadas durante el desarrollo para facilitar el posterior mantenimiento de la aplicación. En muchas ocasiones es incremental, es decir, se va generando y completando en cada una de las fases del desarrollo.
- 6) **Explotación**: En esta parte del proceso, es necesario dejar preparado todo el software para poder ser lanzado a un entorno real para trabajar con usuarios reales y de esta forma entregarlo al cliente. Se recogen los errores y las correcciones en un nuevo documento de mantenimiento. Los programadores y analistas revisan esos fallos para mejorar el software y aprender de los errores
- 7) **Mantenimiento**: Una vez que el software se encuentra en ejecución en un entorno real es necesario mantenerlo, pues normalmente puede haber errores tanto lógicos como externos. Se realizan procedimientos correctivos sobre el programa, siempre hay que tener delante la documentación técnica de la aplicación y las operaciones de mantenimiento se deben documentar para dejar constancia de los cambios.
- 8) **Retirada**: El software ha llegado al final de su vida útil. No resulta rentable seguir ampliándolo manteniéndolo



3. Fases del desarrollo



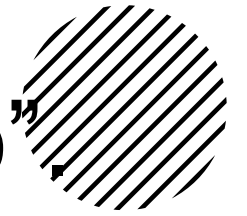


Tarea 1

1. Realiza una infografía o diagrama que represente las diferentes fases del desarrollo del software.



4. Caso práctico 1: “Fases del desarrollo”



A la hora de crear un nuevo producto software, estimamos que tendrá una complejidad alta. En este contexto, nuestro responsable nos pide que nos pongamos cuanto antes **con el diseño del nuevo producto**, ya que el tiempo disponible para el proyecto es limitado.

¿Crees que se trata de una buena práctica?

▪ DEBATAMOS.



5. Metodologías del desarrollo del software.

Una mala elección en la metodología



¿QUÉ CREÉIS QUE PUEDE PROVOCAR?

IMPORTANTE

Necesidades de los clientes en cada punto del ciclo en el que nos encontremos.

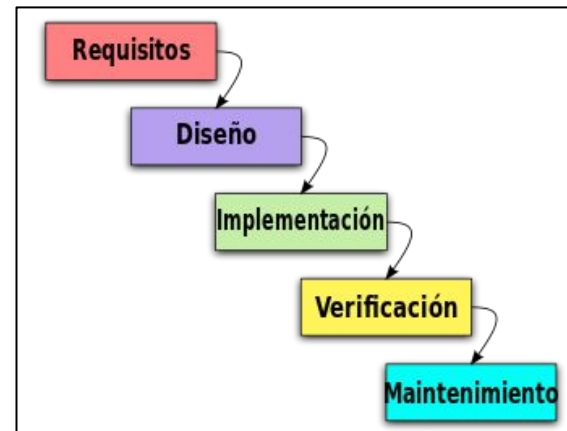
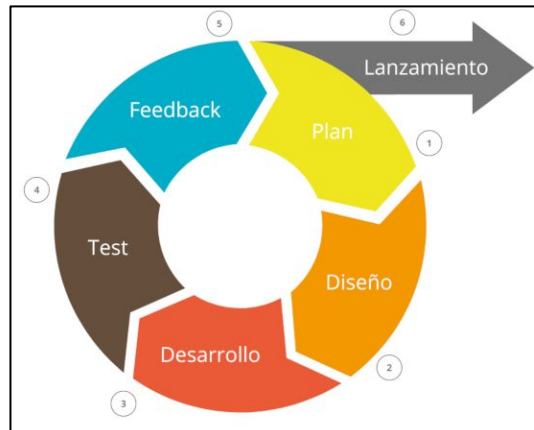


¿A QUÉ METODOLOGÍA CORRESPONDE CADA IMAGEN?

EN ESPIRAL

EN CASCADA

ÁGIL



5.1. Metodología en cascada

Es una metodología lineal que es fácil de seguir por todos los programadores.

CARACTERÍSTICA PRINCIPAL



No se pasa a la siguiente fase hasta que no se completa la fase anterior.

PROBLEMA PRINCIPAL



El tiempo que se requiere para finalizar todo el proyecto: requiere que todas las fases previas estén finalizadas para pasar a la siguiente.

CASO DE USO

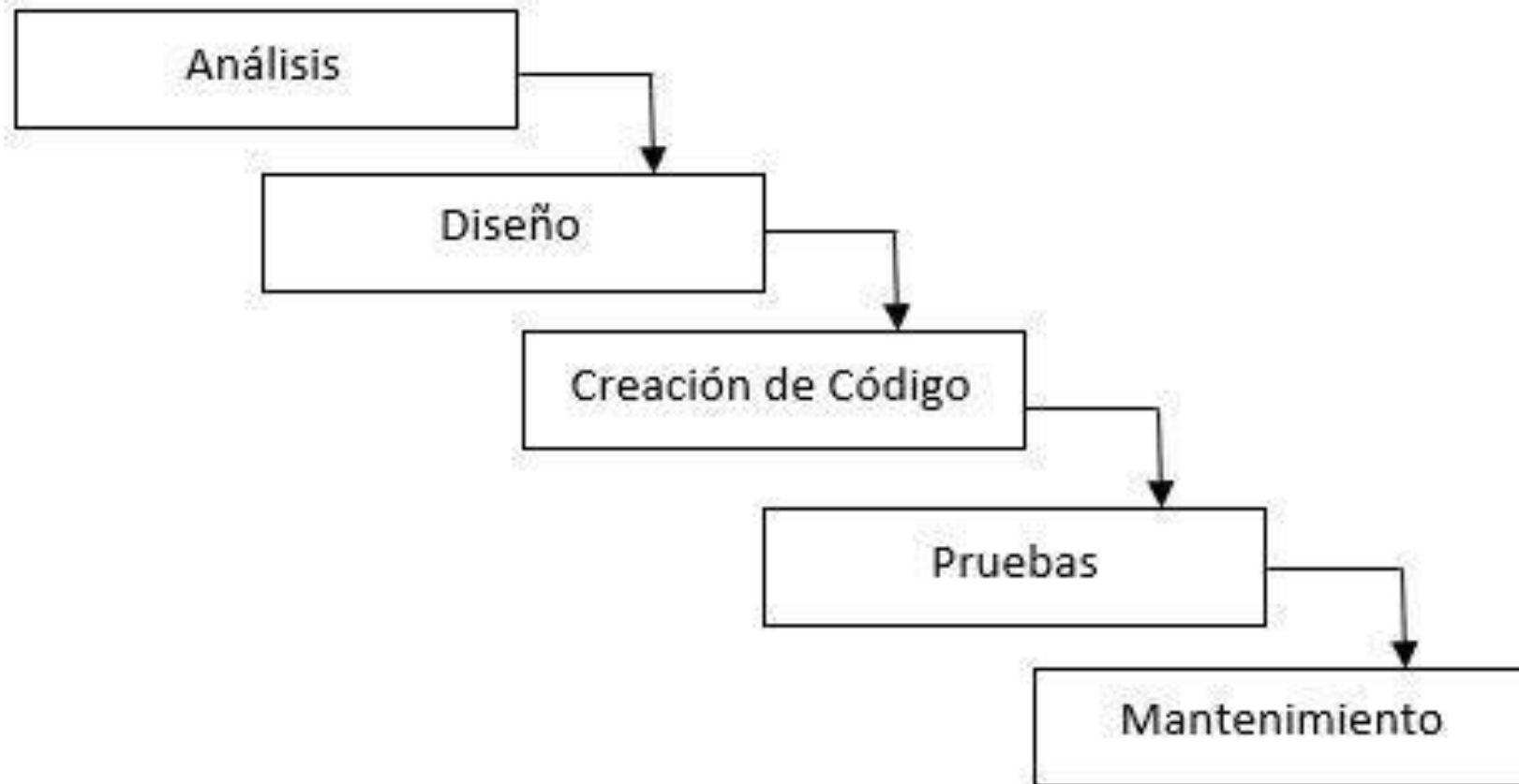


Cuando hay una visión clara de lo que se deberá ser el producto final.

LA PLANIFICACIÓN ES MUY IMPORTANTE !!!



5.1. Metodología en cascada





5.2. Metodología en espiral

Es un modelo que se centra en poder crear una pequeña funcionalidad que continuamente se va a ir agrandando a medida que vamos desarrollando el proyecto, terminando por tanto siguiendo un ciclo de vida en espiral.

CARACTERÍSTICA PRINCIPAL



Tiene un ciclo continuo de actividades que permiten la identificación temprana y la mitigación de riesgos.

PROBLEMA PRINCIPAL



Es complejo de implementar y debe ser seguido de manera estricta.

CASO DE USO



En proyectos de alta complejidad con partes independientes o tecnologías independientes o cambiantes.



5.2. Metodología en espiral





5.3. Metodologías ágiles

Son modelos capaces de adaptar las formas de trabajo a las condiciones del proyecto, consiguiendo inmediatez y flexibilidad.

Busca distribuir de forma permanente sistemas de software en funcionamiento diseñado con iteraciones rápidas.

CARACTERÍSTICA PRINCIPAL



Su enfoque en la flexibilidad, la colaboración y adaptabilidad en el desarrollo.

PROBLEMA PRINCIPAL



La toma de decisiones rápida y frecuente puede ser un desafío para algunos equipos o líderes.

CASO DE USO



Cuando en el desarrollo se necesita la capacidad de adaptación constante.



5.3. Metodologías ágiles

SCRUM PROCESS

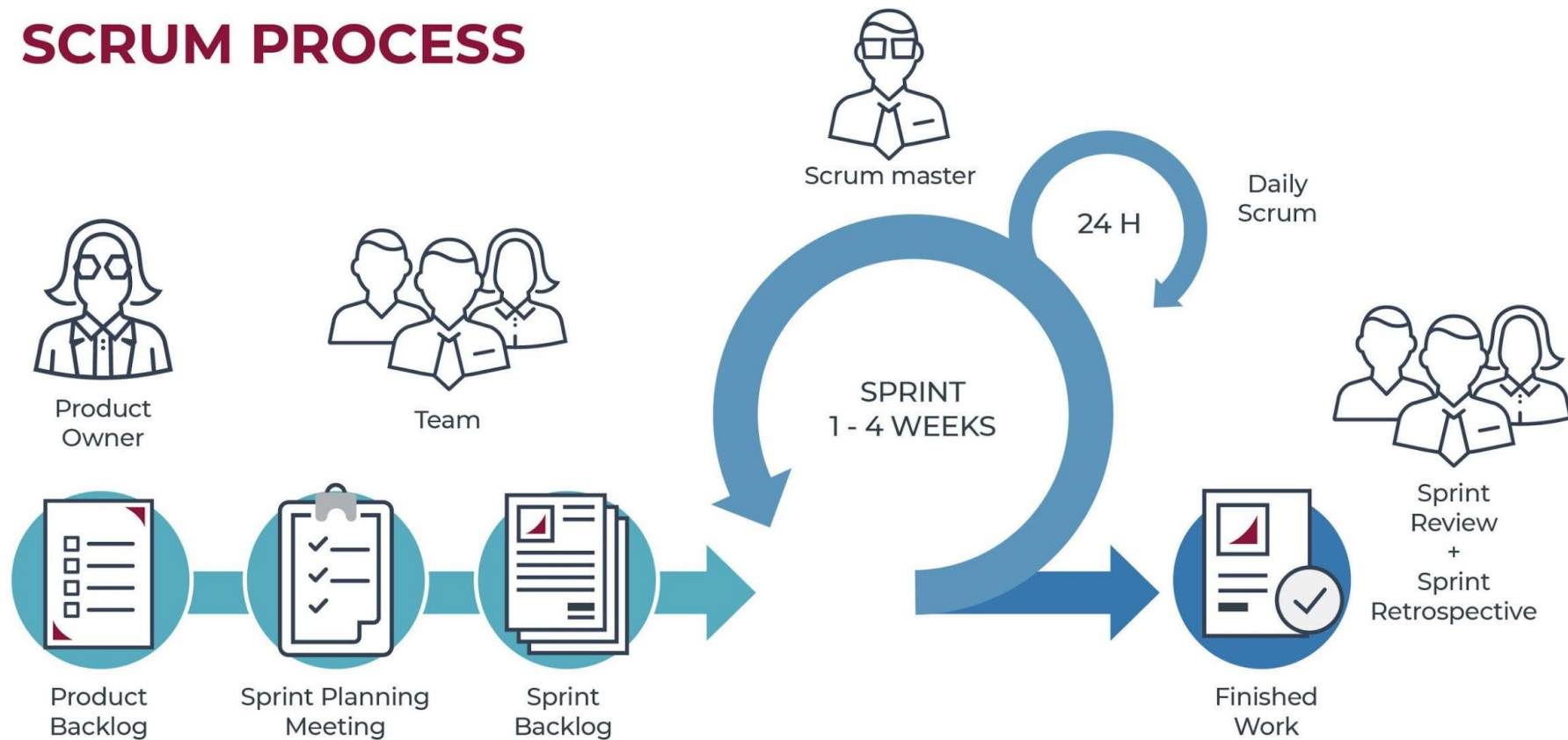
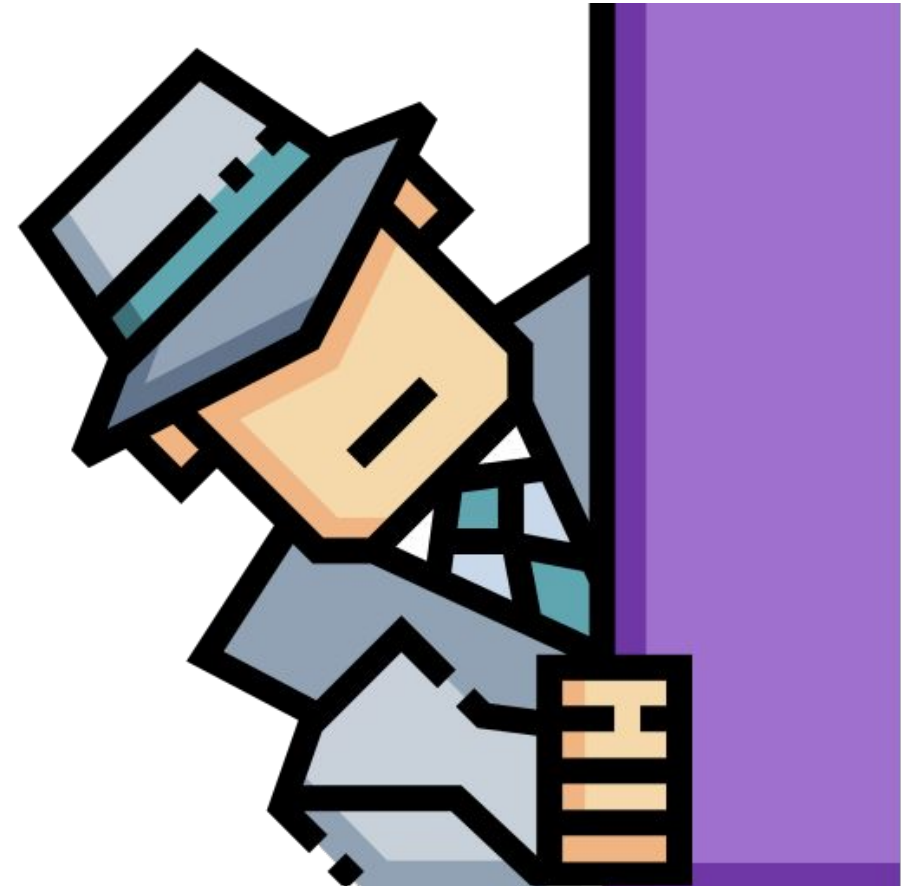


Diagrama de Gantt

<https://www.atlassian.com/es/agile/project-management/gantt-chart>

DE INTERÉS: INVESTIGA



6. Roles en el desarrollo del software.

Durante el ciclo de vida del software es normal que existan diferentes individuos involucrados en el proyecto, que pueden ir saliendo y entrando acorde con las necesidades de éste.



Analista de sistema



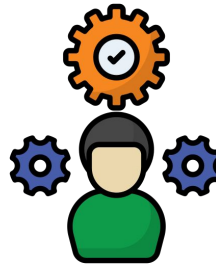
Diseñador del sistema



Analista programador



Programador



Arquitecto de software

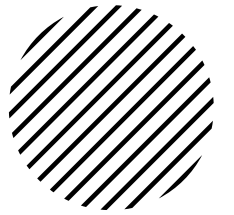


Jefe de proyecto/equipo





6. Roles en el desarrollo del software.



- ❑ **Analista de sistema.** Es el encargado de realizar el estudio del sistema para llevar a cabo la resolución del problema y garantizar que se cumplen las expectativas del cliente. Interviene principalmente en la fase de análisis.
- ❑ **Diseñador de software.** Es el encargado de realizar el diseño del sistema que se va a implementar trabajando principalmente en la fase de diseño.
- ❑ **Analista programador.** Es el individuo que posee mayor dominio de la programación centrándose más en los detalles del proyecto. Trabaja principalmente en las fases de diseño, implementación, pruebas y mantenimiento.
- ❑ **Programador.** Es el encargado de codificar el estudio realizado por los analistas y diseñadores.
- ❑ **Arquitecto de software.** Se trata de un perfil híbrido que permite unir el proceso de desarrollo, pues es el encargado de conocer e investigar los entornos de desarrollo, frameworks y tecnologías diferentes.
- ❑ **Jefe de proyecto.** Dirige el curso del proyecto. Puede ser un analista con experiencia, un arquitecto o una persona dedicada a ese puesto en exclusividad. Debe saber gestionar un equipo y lidiar con los tiempos. Trata de manera continua y fluida con el cliente.





Tarea 2

1. Asigna a cada tarea quién del equipo (rol) la desempeña.

(Siguiendo diapositiva)

Tarea 2 ➡ ¿QUIÉN HACE QUÉ?



- a) Identificar los requisitos funcionales y no funcionales de la aplicación en una reunión con el cliente.
- a) Crear los bocetos y maquetas visuales de la interfaz de la aplicación para el diseño de la interfaz de usuario.
- a) Diseñar la estructura general del sistema y la interacción entre los componentes.
- a) Implementar la funcionalidad para que los usuarios se registren en la aplicación que se está desarrollando.
- a) Detallar los flujos de búsqueda de libros y sus requisitos
- a) En la creación del back-end para la gestión de libros, el desarrollo de sistemas de dicha gestión, incluyendo la base de datos.
- a) Analizar y optimizar las consultas a la base de datos mejorando el rendimiento.
- a) Realizar pruebas de usabilidad con usuarios para identificar posibles problemas de diseño.
- a) Revisar el código, identificar errores y depurarlos para mejorar la calidad.
- a) Crear documentación técnica y de usuario para el proyecto.





Tarea 3

1. Investiga sobre SCRUM y crea una presentación (en parejas)

- ✓ ROLES EN SCRUM
- ✓ EVENTOS
- ✓ ARTEFACTOS
- ✓ PRINCIPIOS Y VALORES DE SCRUM
- ✓ VENTAJAS E INCONVENIENTES.
- ✓ HERRAMIENTAS Y SOFTWARE PARA APOYAR SCRUM.

LA INGENIERÍA DE *software*



**¿ALGUNA
DUDA?**

GRACIAS