



## TAREA 1. Investigación de conceptos clave de los procesos de software:

### 1. Roles y responsabilidades:

- Analistas:
- Diseñadores:
- Programadores:
- Testers:

### 2. Metodologías:

#### a. Modelo en Cascada (Waterfall): Descripción

Fases

Ventajas/Desventajas

#### b. Modelo Ágil (Agile):

Descripción

Fases

Ventajas/Desventajas



- ¿Qué roles son esenciales en un proyecto de software y por qué?
- ¿En qué situaciones sería más adecuado usar el modelo en cascada?
- ¿Cuáles son las principales diferencias entre el modelo en cascada y el ágil?

- **Actividad 1: Cascada vs. Ágil**
  - Divididos en dos grupos: uno defiende el modelo en cascada y otro el modelo ágil.
  - Cada grupo debe presentar:
    - Una descripción de la metodología.
    - Ejemplos de proyectos donde sería útil.
    - Ventajas y desventajas.



## **TAREA 2.** Investigación sobre:

- Importancia de verificar el funcionamiento del software en el ambiente de producción.
- Diferencias entre ambientes de desarrollo, pruebas y producción.
- Impacto de un mal funcionamiento en producción (pérdidas económicas, reputación, etc.).

<https://www.youtube.com/watch?v=CM9yvzHwuDg>

## **TAREA 3.** Investigación sobre estándares vigentes para la verificación:

Normas y estándares de calidad (ISO/IEC 25010, IEEE 829, etc.).

Buenas prácticas de la industria (ITIL, DevOps, etc.).

Cumplimiento de regulaciones y políticas de la organización.

<https://www.iso.org/standard/35733.html>

<https://www.atlassian.com/es/devops>

## **TAREA 4.** Investigación sobre:

Verificación del funcionamiento del Software:

- Pruebas de Funcionalidad.
- Pruebas de Rendimiento.
- Pruebas de Seguridad.
- Pruebas de Usabilidad.
- Pruebas de Compatibilidad.

<https://www.youtube.com/watch?v=hSxXuRxA9mo>

## **TAREA 5.** Investigación sobre:

Las herramientas y tecnologías utilizadas en la verificación del software, incluyendo pruebas automatizadas, monitoreo, gestión de incidencias y plataformas de CI/CD



1. ¿Por qué es importante verificar el software en producción?
2. ¿Qué diferencias existen entre los ambientes de desarrollo, pruebas y producción?
3. ¿Qué consecuencias puede tener un fallo en producción?
4. ¿Qué normas y estándares de calidad son más relevantes en el desarrollo de software?
5. ¿Cómo contribuyen las buenas prácticas como ITIL y DevOps a la calidad del software?
6. ¿Por qué es importante cumplir con regulaciones y políticas en el desarrollo de software?



7. ¿Qué son las pruebas de funcionalidad y por qué son importantes?
  
8. ¿Cómo se realizan las pruebas de rendimiento y qué métricas se utilizan?
  
9. ¿Qué aspectos se evalúan en las pruebas de seguridad?
  
10. ¿Cómo se realizan las pruebas de usabilidad y qué beneficios aportan?
  
11. ¿Qué se evalúa en las pruebas de compatibilidad?
  
12. ¿Qué herramientas de pruebas automatizadas conocen y cómo se utilizan?
  
13. ¿Cómo ayudan las herramientas de monitoreo y alertas a garantizar la calidad del software?
  
14. ¿Qué ventajas ofrecen las plataformas de CI/CD en el desarrollo de software?



o Los estudiantes asumen roles (analista, diseñador, programador, tester) y simulan un pequeño proyecto de software.

- 1. Nombre del proyecto de software:
- 2. Definir los roles:

Rol	Nombre	Funciones

o Deben definir cómo colaborarán en cada metodología (cascada o ágil).

Rol	Metodología casaca	Metodología ágil





### *Estudio de Casos 1*

Noviembre de 2018, Target, cadena de retail más grandes de Estados Unidos, experimentó un fallo en su sistema en línea durante el *Black Friday*, uno de los días de compras más importantes del año. El sitio web y la aplicación móvil colapsaron, dejando a los clientes sin poder realizar compras durante varias horas.



En equipo, comparten la información que investigaron y responde a las siguientes preguntas:

1. ¿Qué causó el fallo en el sistema de Target durante el Black Friday?
2. ¿Qué consecuencias tuvo este fallo para la empresa y los clientes?
3. ¿Qué medidas podrían haberse tomado para evitar este problema?
4. ¿Cómo crees que Target podría mejorar su sistema para evitar fallos en el futuro?



### *Estudio de Casos 2*

En 2018, entró en vigor el **Reglamento General de Protección de Datos (GDPR)** en la Unión Europea, una regulación diseñada para proteger la privacidad y los datos personales de los ciudadanos. Facebook, como una de las plataformas más grandes del mundo, tuvo que realizar cambios significativos en su software y procesos para cumplir con esta normativa.



1. ¿Qué cambios implementó Facebook para cumplir con el GDPR?
2. ¿Cómo mejorar la calidad del software de Facebook como resultado de estos cambios?
3. ¿Qué impacto tuvo el cumplimiento del GDPR en la experiencia del usuario?
4. ¿Qué lecciones se pueden extraer de este caso para otros proyectos de software?



### *Estudio de Casos 3*

En 2015, Microsoft lanzó Windows 10, una de las actualizaciones más importantes de su sistema operativo. Para garantizar la calidad del producto, Microsoft implementó un programa masivo de pruebas de software que involucró a millones de usuarios y una variedad de técnicas de prueba.



1. ¿Qué tipos de pruebas realizó Microsoft para garantizar la calidad de Windows 10?
2. ¿Cómo impactan estas pruebas en la calidad del producto final?
3. ¿Qué lecciones se pueden extraer de este caso para otros proyectos de software?
4. ¿Cómo podrían aplicarse estas prácticas de prueba en un proyecto de menor escala?



- ¿Qué aprendí?
  
- *Si eres responsables de un proyecto de software, ¿qué medidas tomarían para evitar fallos en producción?*
  
- ¿Qué normas o prácticas te parecieron más útiles y por qué?
  
- ¿Cómo aplicarías lo aprendido en un proyecto real?



## PRACTICA 5. MAPA CONCEPTUAL

**Instrucciones:** de forma individual realizan un mapa conceptual, donde coloquen todo lo aprendido del tema: de prueba de software en ambiente productivo.