



**NOTA:** este documento es solo una guía de repaso y de autoevaluación, no se debe responder para entregar.

## Repaso Teoría

- Ingeniería de Software
  - ¿Cuál es la diferencia entre programa y software?
  - ¿Qué es la ingeniería de software?
  - ¿Cuáles son los roles en la ingeniería de software?
  - ¿Cuáles son las principales diferencias entre los modelos de software tradicionales y ágiles?
- Ciclo de vida del software
  - ¿Cuáles son las etapas del ciclo de vida del software?
  - ¿De qué se trata la etapa de requerimientos?
  - ¿Cuál es la diferencia de requerimientos en tradicional y en ágil?
  - ¿Cuál es la diferencia entre requerimientos funcionales y no funcionales?
  - ¿Cuáles son los tipos de requerimientos no funcionales?
  - ¿De qué se trata la etapa de análisis y diseño?
  - ¿Cuál es la diferencia de análisis y diseño tradicional y ágil?
  - ¿De qué se trata la etapa de implementación?
  - ¿Cuál es la diferencia de implementación en tradicional y en ágil?
  - ¿De qué se trata la etapa de pruebas?
  - ¿Cuál es la diferencia de pruebas en tradicional y en ágil?
- Metodologías ágiles
  - Definición de scrum
  - Definición de lean
  - Definición de xp
  - Definición de kanban
- Roles en Scrum
  - Scrum master
  - Product Owner
  - Stakeholder
  - DevTeam
- Ceremonias en Scrum
  - Release planning
  - Sprint Planning
  - Daily
  - Grooming o refinamiento
  - Sprint Review
  - Retrospective
- Artefactos en Scrum
  - Backlog
  - Burndown chart
  - Historias de usuario

- Implementación en ágil, clean code, pruebas unitarias, coverage
  1. Nombre 5 principios de clean code
  2. ¿Qué son las pruebas unitarias?
  3. ¿Qué significa coverage en las pruebas unitarias?
  4. ¿Cuáles prácticas se usan en implementación ágil? ¿Cuál es la diferencia con la implementación tradicional?
  5. ¿Qué significa code review?
- Arquitectura de software ágil.
  1. ¿Cuál es la diferencia entre arquitectura tradicional y arquitectura ágil?
  2. ¿Qué significa MVC? ¿Qué significa MPV?
  3. ¿Qué es la arquitectura monolítica?
  4. ¿Qué es la arquitectura SOA?
  5. ¿Qué es la arquitectura microservicios?
  6. ¿Qué es la arquitectura serverless?
  7. ¿Cuáles son las tareas de un arquitecto de software?
- Pruebas en metodologías ágiles: Herramientas, Bug tracking. Pruebas funcionales, pruebas no funcionales, casos de pruebas, regresiones, automatización
  1. ¿Cómo definiría a Pruebas ágiles de software?
  2. Describa el proceso de bug tracking
  3. ¿Cuáles herramientas de automatización se usan en las pruebas ágiles?
  4. ¿Cuál es la diferencia entre pruebas tradicionales y pruebas ágiles?
  5. Defina qué es un bug y cómo se puede clasificar
  6. ¿Qué son las pruebas funcionales ?
  7. ¿Qué son las pruebas no funcionales?
  8. ¿Cuáles son las tareas de un ingeniero de pruebas de software?
- Deployment en ágil: despliegues, builds, releases, versionamiento, herramientas
  1. ¿Cuál es la diferencia entre deployment tradicional y deployment en ágil?
  2. ¿Qué significa CI?
  3. ¿Qué significa CD?
  4. ¿Cómo se puede realizar el versionamiento de un software por número?
  5. Nombre 3 buenas prácticas de Deployment
  6. ¿Cuáles son las tareas de un ingeniero devops?



## Repaso Práctica

Se define un problema y se debe plantear su solución con un software. Para el software propuesto, desarrollar los siguientes puntos (practicar con el tema del proyecto):

- Proponga 3 requerimientos no funcionales. Indique en cada uno a cuál tipo de requerimiento no funcional corresponde.
- Redacte 3 historias de usuario para el software con todas sus partes (título, descripción, criterios de aceptación, definition of done)
- Dibujar/modelar arquitectura del proyecto
- Diseñe 3 casos de prueba para el proyecto
- Reportar 3 issues de un problema con todas sus partes
- Dibujar/modelar proceso de deployment del proyecto