

## **Unidad 1**

### **CAPITULO 1 (PRESSMAN)**

¿Cuáles son las competencias de ingeniería de software?

¿Qué es software y como se define? ¿porque es más complicado realizar el producto

Dificultades y riesgos de hacer software

Tipos de software

Las etapas del software

Cuáles son los elementos que caracterizan a un

Tipos de roles en un equipo de desarrollo

## **Unidad 2**

Que es ingeniería de requerimientos, porque existe, problemáticas

Como iniciar el proceso de requerimientos

Como ´priorizar los requerimientos

Tipos de requerimientos

Que es un caso de uso y para qué sirve

Que son las excepciones en un caso de uso

Porque se hace el análisis de riesgos

Que significa que se validen los requerimientos

Que artefactos se producen en la fase de requerimientos y cuánto dura la fase de requerimientos

Cuáles son las características principales de la fase de diseño

Que significa que la base de datos sea relacional

Características que se deben reflejar del software como parte del diseño

Que es arquitectura, patrón de diseño, modularidad

Elementos de UML

¿Qué es lo involucra UI, y que involucra UX?

¿Qué es una estructura de datos, para que sirve y de donde se obtiene?

¿Qué es la administración de la configuración?

¿Qué es un software legado?

¿Qué es un estándar de codificación?

¿Qué es una métrica de software?

¿Cómo medimos la calidad de un software?

¿Cuál es el proceso de verificación que ocurre en el diseño?

¿Cuáles son los 4 tipos de mantenimiento?

Guías para establecer cuando hacer mantenimiento

Elementos de la calidad de construcción

Que es un IDE

¿Qué es un editor?

¿Qué es un API y como se utiliza?

¿Cuáles son los tipos de pruebas que podemos hacer?

¿Cuáles son los parámetros para determinar el costo de mantenimiento?

El Software y la Ingeniería de Software. ¿Qué es la ingeniería de software? Terminologías. Clases de software, de base y las distintas áreas. Software de sistema, aplicaciones web, etc. Elementos propios de ingeniería de software. Las tres capas de ingeniería de software: herramientas, métodos y proceso. ¿Cómo se define la cantidad? Proceso de desarrollo de software (generalizado y especificado). Estrategias que se plantean a la hora de buscar soluciones en el desarrollo de software. Método genérico. Principios básicos y generales. Mitos de software. Terminología básica y roles en el equipo de desarrollo. Tareas básicas en el equipo de desarrollo. Análisis, pruebas, etc. 5 fases del desarrollo de software. Comprensión de requerimientos. Ingeniería de requerimientos, sus etapas y lo que hace cada una. ¿Por qué existe? ¿En qué se diferencia de ingeniería de software? Problemáticas en la fase de requerimientos. ¿Cómo iniciamos el proceso de requerimiento? Las preguntas iniciales. Cómo se administra y priorizamos los requerimientos. Cuestiones particulares que tienen que ver con la documentación. Requerimientos funcionales y no funcionales y otras clases de requerimientos. Desarrollo de caso de uso. ¿Para qué sirven los casos de uso? Diagrama de casos de uso. ¿Qué es un actor y cómo se representa? Stay Holders. ¿Qué significa generar un escenario? Validación de requerimientos. ¿Qué artefactos se reproducen en la fase de requerimientos? Plática con Javier. :0 Fase de diseño. Qué es trazabilidad. Características principales de la fase de diseño. Abstracción y diferentes conceptos. 4 productos resultantes de la fase de diseño. Datos relacionales y no relacionales. Arquitectura de software. Qué representa. Usabilidad, rendimiento etc. Experiencias de usuario.

Competencias del programa de ingeniería de software. Comparación de la malla curricular de la UADY con el de CONAIC. Competencias que debería saber un ingeniero de software al graduarse.

## Unidad 2

C:

-¿?

¿Por qué las 5 fases comparten el nombre “proceso” con las 3 capas?

- Construcción de software

¿Cuáles son los productos resultantes de la fase de implementación?

--codificación

¿Qué es un estándar de codificaciones?

¿Qué es un controlador de versiones?

¿Administración de la configuración?

- ¿Que es administración, y que es configuración?

- Proponer soluciones algorítmicas.

-Que es una estructura de datos?

¿Para qué sirve?

¿De dónde se obtiene?

--Implementación

¿Qué es una métrica?

¿Cuáles son las partes de la métrica que se pueden aplicar en software?

¿Como medimos la calidad del código que se está produciendo?

¿Cuáles son las habilidades para trabajo remoto?

- Pruebas

-Proceso de testing

-V&V

- Que es verificación y validad

Diferentes tipos de pruebas (¿estáticas, dinámicas, etc.?)

¿Qué es una prueba de caja negra?

¿Qué es una prueba de caja blanca?

¿Qué es una prueba de aceptación?

¿Qué es una prueba de sistema?

¿Que son las inspecciones? (Validaciones o verificaciones)

Planes de pruebas

Desde la parte del diseño, se podría especificar una parte para que el software

pueda ser probado.

validación

Mostrarle al cliente que se cumplió con el requerimiento funcional.

Al no especificar un requerimiento no funcional, es difícil validarlos.

¿Como hago una validación?

Dar ejemplos de pruebas

¿Como se realizan las pruebas de estrés? ¿y en qué momento se aplican?

¿Para qué se usan las pruebas de usuarios? ¿que resulta con esto?

¿Qué es una prueba de unidad?

¿Qué es integración continua?

¿Qué se hace en las pruebas de sistema? ... que involucra?

Test driven development

¿Cuál es el rol de un QA?

¿Cuáles son los parámetros de calidad que podemos verificar?

¿Cuál es un ejemplo de métrica para aplicar en los parámetros de calidad?

- Mantenimiento (Evolución del software)

Tipos de mantenimiento (4)

Ejemplificar cada uno.

Costo del mantenimiento

¿Vale la pena darle mantenimiento al sistema, o es mejor hacerlo de nuevo?

guías para establecer .. ¿Cuándo se debe hacer mantenimiento?

¿Qué es un sistema legado?