PREGUNTAS CORTAS

1. Modelos de ciclo de vida en cascada y en V. Expliquelos y compárelos.

El **cliclo de vida en cascada** es la secuencia de las distintas fases de la producción del software. Como elementos de unión entre cada fase aparecen los diferentes documentos que se generan en cada fase. (Analisis, Diseño, Codificacion, Integracion, Mantenimiento).

El modelo en cascada obliga a terminar cada fase antes de comenzar la siguiente, se basa en el trabajo de la anterior fase.

Para detectar errores y evitarlos en otras fases, se dan procesos de revisión al completar cada fase, antes de pasar a la siguiente.

El **ciclo de vida en V**  se basa en una secuencia de fases a la del modelo en cascada pero le da mayor importancia a la visión jerarquizada que se va teniendo de las distintas partes del sistema a medida que se avanza con el sistema. La comprobación de una parte del sistema se llama verificación. La comporbación de que un elemento satisface las necesidades del usuario identificadas durante el análisis se llama validación.

La diferencia de ambos es que el CV en cascada realiza todas las fases dependiendo unas de otras (diseño secuencial lineal). Es además un proceso continuo. Un error impide la secuencia del diseño totalmente. Los defectos se detectan en la fase d epruebas.

Mientras que el CV en V jerarquiza modulando las fases y verificando sin depender al 100% de la fase anterior (diseño secuencial en V). Es un proceso simultaneo. Un error impide la secuencia del diseño en menor medida, respecto al cascada. Los defectos se detectan en la fase inicial.

1. Qué tres objetivos fundamentales o cualidades mínimas es deseable alcanzar al hacer la **descomposición modular** de un sistema? Explica cada uno de ellos y en cada caso cómo se puede medir o qué factores intervienen.

Los tres objetivos son:

Son:

* **Claridad**. Siempre es más fácil de entender y manejar cada módulos de un sistema que tratar de entenderlo como un todo compacto.
* **Reducción de costes**. Es más barato desarrollar, documentar, probar, depurar y mantener un sistema modular que un único bloque de sistema.
* **Reutilizabilidad**. Si los módulos se diseñan teniendo en cuenta otras opciones posibles será inmediata su reusabilidad.

Para lograr una descomposición modular es necesario identificar:

* Los modulos.
* Descripción de cada modulo.
* Describir las relaciones entre modulos

Para poderse medir su descomposición modular es necesario tener:

* **Independencia funcional**. Matriz de requsitos/componentes del final de los documentos ADD Y DDD es necesario indicar que modulo se encarga de realizar cada uno de los requisitos o funciones del SRD. Pudiendo decirse que cada función se puede realizar en un modulo distinto. Para poder ser funcional debe realizar una función concreta o conjunto de ellas sin apenas relación con el resto de modulos del sistema.

Para medir la independencia funcional se usan dos criterios: acoplamiento y cohesión

* **Adaptabilidad.** Al diseñar un istema se pretende resolver un problema concreto y se trata de obtener prácticamente un sistema optimo. Sus factores son:
  + **Previsión. Resulta complicado prever que evolución futura tendrá un determinado sistema. Por ello ha de tener un acomplamiento débil con el resto de modulos, asi los cambios no afectan a todo el sistema.**
  + **Accesibilidad. Es necesario conocer lo mejor posible el sistema, para poder realizarle adaptaciones futuras, para ello es necesario estudiar su estructura, y esto es sencillo si resulta accesible todos los documentos de especificación, diseño e implementación.**
  + **Consistencia. Cuando se realizan adaptaciones sucesivas es vital mantener la consistencia entre los documentos de especificación, diseño e implementación para cada adaptación. La tendencia es usar aplicaciones para llevar un control de versiones y configuración.**

Las técnicas de diseño para conseguirlo son:

* Diseño funcioanl descendente.
* Diseño orientado a objetos.
* Diseño de datos.

1. Descripción de las estrategias básicas de integración de los modulos de un producto software.

Los modulos de un producto SW se han integrado para formar un sistema completo, las trategias de integración son:

* Integración Big Bang. Realizar la integración de todas las unidades en un único paso, la cantidad de errores que aparecen puede ser inmensa y difícil de gestionar. Esta integración funciona para sistemas pequeños, su ventaja es que evita la realización de ‘andamiaje’.
* Integracion descendente. Parte de un modulo principal que se prueba con modulos de andamiaje de los otros modulos usados directamente. Los modulos sustitutos se van reemplazando, uno por uno, por los verdaderos y se realizan las pruebas de integración correspondientes. La codificación de los mod. Sustitutos es un trabajo adicional que ha de simplicarse lo máximo posible para poder adaptar distintas soluciones. La ventaja fundamental es que se ven las posibilidades de la aplicación y de prototipo para mostrar
* Integración ascendente. Empieza codificando por separado y en paralelo todos los modulos de nivel mas bajo. Para probarlos ponen drivers que los hacen trabajar de forma independiente. Su interés radica fundamentalmente en su capacidad de probar explícitamente situaciones especiales o infrecuentes. Las ventajas que tiene es que facilita el trabajo en paralelo y el ensayo de situaciones especiales pero su desventaja es que para situaciones globales no es tan funcional.

1. Convencer a un cliente de la importancia de la Ing. De la producción del software.
2. Explique con un ejemplo que es un diagrama de transición de estados.

Los diagramas de estado muestran los diferentes estados de un objeto durante su vida, y los estímulos que provocan los cambios de estado en un objeto. Todos los objetos o ‘maquinas o personas’ tienen un estado. Por ejemplo vender pan en una panaderia

* Tendremos un punto de inicio – la panaderia
* Transiciones de estados – texto de la actividad que se realiza – comprando pan a los proveedores, exponiendo tipos de panes, atendiendo a cliente, gestionando venta en caja, etc.
* Estado- rectángulo con vértices redondos – comprar pan, exponer panes, atender cliente, atender caja, etc
* Punto de terminación – circulo relleno en el interior de otro. – Fin – cerramos venta o cerramos la tienda hasta mañana, final del dia.

1. Explique que es la genericidad.

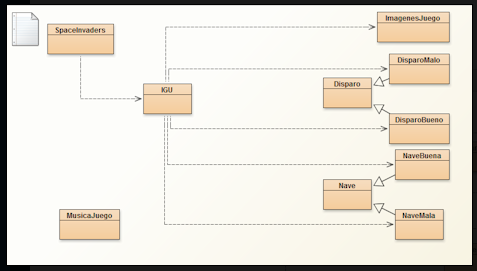
Agrupar aquellos elementos del diseño que usan estructuras similares o funcionan de forma similar. Se generalizan los elementos que comparten y después se crean los elementos específicos para cada objeto. Dando lugar a acciones simples y fáciles de mantener. En el modelo orientado a objetos se llaman clases, y estos objetos que comparten ciertas características se les llama clases genéricas o plantillas, de las cuales las clases especificas heredan de la plantillas sus características, naciendo asi el termino herencia en POO.

1. El ciclo en espiral. Esquema y explicación de fases. Describa en que fase se diseña del software.

Es el refinamiento evolutivo del ciclo de vida, introduce la actividad de análisis de riesgo como elemento fundamental para guiar la evolución del proceso de desarrollo. Este ciclo se convierte en espiral al añadir como dimensión radial una indicación del esfuerzo total realizado hasta el momento, que va creciendo.

Se divide en 4 zonas de actividad. – planificación, análisis del riesgo, ingeniería, evaluación

1. DFD de space invader.



1. Defina abstracción.

‘Sacado a partir de sus limites externos’, se trata de cualquier cosa que partiendo de la apariencia extrerna de una realidad, se aleja de ella. Se trata de cualquier elemento que forma parte del sistema que se quiere modelar, con entidad o importancia para distinguirlo del resto y esté dotado de funciones relevantes para el sistema. Se pueden realizar sin tener en cuenta el sistema de control de acceso concreto en el que se usará, facilitando los cambios futuros y su reutilización en diversos sistemas.