Primera Parte de la teoría

1. **Muchos de los fallos del SW son consecuencia de dos factores. ¿Cuáles son?**

* Demandas crecientes
* Expectativas Bajas

1. **¿Cuáles son los mejores métodos y técnicas de la ingeniería del software?**

Por ejemplo, los juegos siempre deben diseñarse usando una serie de prototipos, mientras que los sistemas críticos de control de seguridad requieren de una especificación completa y analizable para su desarrollo. Por lo tanto, no puede decirse que un método sea mejor que otro.

1. **Existen dos tipo de productos software:**

* Productos genéricos y productos específicos.
* Productos genéricos y productos a medida.
* Software vertical y productos personalizados.
* **Ninguna de las anteriores.**

1. **En el SW horizontal, ¿Quién controla la especificación?**

* **La empresa que lo desarrolla**
* La empresa que lo contrata
* El cliente
* El usuario

1. **Definición de Ingeniería del Software**

Es una disciplina de la ingeniería que se interesa por todos los aspectos de la producción de software.

1. **“Cada vez con más frecuencia se requieren sistemas que operen como distribuidos a través de redes que incluyan diferentes tipos de ordenadores, dispositivos móviles. Se tendrá que integrar con frecuencia el nuevo software con sistemas legados más viejos escritos en diferentes lenguajes de programación”. ¿De qué problema general del SW estamos hablando?**

Heterogeneidad

1. **Explique en que consiste y qué contexto se encuentra el concepto del “software como servicio”**

Se encuentra en el contexto de que el software no correría en computadoras locales, sino en “nubes de computo” (un enorme número de sistemas de cómputo vinculados que comparten muchos usuarios) a las que se accede usando internet.

1. **En un desarrollo web, ¿qué podemos decir en cuento al diseño y desarrollo de las interfaces de usuario?**

Que anteriormente las aplicaciones empresariales eran monolíticas, haciendo que los programas corriesen en computadores individuales o en grupos de computadores, y ahora estas aplicaciones se encuentran distribuidas a lo largo del todo el mundo.

1. **¿Cuáles son los tres desafíos del software que afectan a todos los desarrollos?**

* El SW es intangible
* Cada proyecto es distinto
* En los proyectos grandes, es inútil la reutilización

1. **“Son sistemas para modelar procesos o situaciones físicas que incluyen muchos objetos separados interactuantes. Dichos sistemas a menudo son computacionalmente intensivos y para que se ejecuten requieren sistemas paralelos de alto rendimiento”. ¿De qué tipo de aplicaciones estamos hablando?**

Sistemas para modelado y simulación

1. **Existen fundamentos de ingeniería del SW que se aplican a todos los tipos de sistemas de SW. Por ejemplo:**

* **Tiene que usar de manera tan efectiva como sea posible los recursos existentes**
* Debe existir una planificación que no dé lugar a improvisaciones
* Los fundamentos de la Ingeniería del SW cambian según los sistemas.
* Ninguna de las anteriores.

1. **Explique en qué consiste y en qué contexto se encuentra el concepto del “SW como servicio”**

Se encuentra en el contexto de que el software no correría en computadoras locales, sino en “nubes de computo” (un enorme número de sistemas de cómputo vinculados que comparten muchos usuarios) a las que se accede usando internet.

1. **Desde un punto de vista ético, ¿Qué significado tiene la competencia?**

Que la competencia en dos países distintos no es la misma ya que la ingeniería del SW se realiza dentro de un marco social y legal que limita la libertad de la gente que trabaja en dicha área.

1. **Enumerar los modelos de procesos software**

Modelo en cascada, desarrollo incremental, ingeniería del SW orientada a la reutilización.

1. **Una de las siguientes no es exactamente una etapa del modelo de procesos en cascada.**

* **Análisis y definición de requerimientos**
* Diseño del sistema
* Implementación y prueba de unidad
* Integración y prueba del sistema

1. **Las descripciones de los procesos deben incluir los productos, las pre y postcondiciones y…**

Los roles

1. **¿Qué es un proceso de software?**

Son una serie de actividades relacionadas que conduce a la elaboración de un producto de software

1. **El desarrollo incremental tiene tres beneficios importantes**

Se reduce el corte de adaptar los requerimientos cambiantes del cliente, es más sencillo obtener una realimentación del cliente y la entrega e implementación de SW útil al cliente sea más rápida.

1. **El desarrollo incremental presenta dos serios inconvenientes**

El proceso de creación no es visible, y la estructura del sistema tiende a degradarse conforme se tienen nuevos incrementos

1. **Existen cuatro actividades fundamentales en el proceso de ingeniería de requerimientos, estudio de viabilidad y…**

Obtención y análisis de requerimientos, especificación de requerimientos y la validación de requerimientos.

1. **Si hablamos de la ética de la profesión de ingeniería del SW, hay aspectos difusus que no están plenamente regulados por las leyes. Entre estos aspectos encontramos la confidencialidad, la competencia, el mal uso de los ordenadores, y…**

Derechos de propiedad intelectual

1. **La prueba de defectos y la depuración son el mismo proceso.**

* Sí, pero la primera la desarrolla el equipo de integración y la segunda el programador.
* Sí, pero la primera la desarrolla el programador y la segunda el equipo de integración.
* **Sí, pero ambas la realiza el programador.**
* No

1. **Las pruebas…**

* Alfa están asociadas al SW horizontal
* Beta están asociadas al SW vertical
* Beta están asociadas a productos personalizadas
* **Ninguna de las anteriores**

1. **Las pruebas FAT también se denominan pruebas de [1] y las pruebas SAT se denominan también prueba de [2]**

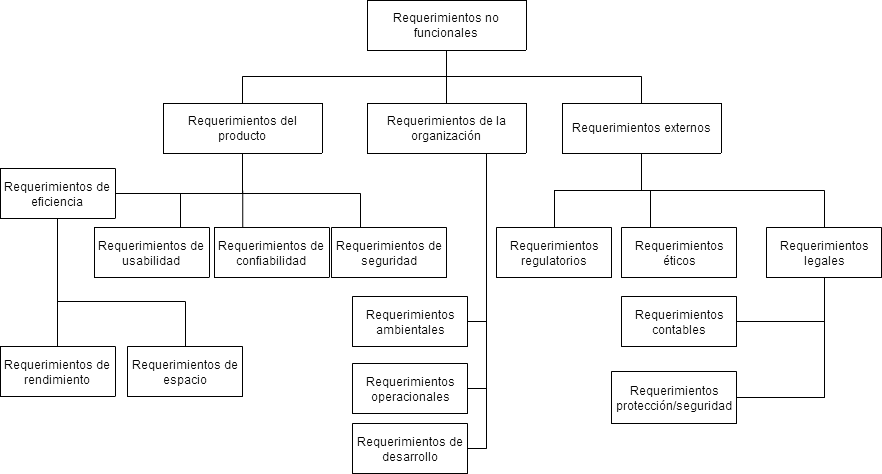
Pruebas del sistema

Pruebas de aceptación

1. **El cambio es inevitable. Tenemos dos formas de enfrentar el cambio para que sus consecuencias no hipotequen o retrasen todo el proyecto. ¿Qué formas?**

Mediante el uso de prototipos y usar una entrega incremental.

1. **Clasificación de los requisitos no funcionales.**

****

SEGUNDA PARTE

1. **¿Qué es un prototipo?**

Es una aplicación que presenta alguna funcionalidad completa, y el resto de la aplicación se encuentra incompleto

1. **Problemas de usar prototipos.**

* Propiedades emergentes
* Mala documentación
* Estructura deficiente
* La calidad

1. **Define el término requerimiento.**

Es el proceso de recopilar información sobre el sistema requerido y los sistemas existentes

1. **Tipos de requisitos**

* Funcionales
* No funcionales
* De dominio

1. **Fases de desarrollo del SW**

Especificación, diseño, desarrollo, validación/pruebas y evaluación

1. **Tipos de datos que hay en el SW**

* Primarios
* Secundarios

1. **¿Qué es un algoritmo?**

Es una composición de datos primarios para obtener un resultado mediante el cálculo de la combinación de datos

1. **¿Qué es una transacción y “roll back”?**

Una transacción son un conjunto de operaciones que se tienen que realizar hasta el final para terminar el proceso (como sacar dinero de un cajero)

Roll back es la devolución de todas las transacciones

1. **Atributos de un buen SW**

* Dar funcionalidad y el desempeño requeridos
* Debe ser sustentable
* Confiable
* Utilizable

1. **Tipos de sistemas**

* Sistemas en tiempo real (entradas pueden afectar a las salidas)
* Sistemas en tiempo útil (la respuesta a la entrada es coherente con los tiempos de entrada)

Segunda Parte de la teoría

1. **En que ocasiones no podremos aplicar nunca una metodología ágil**

* Sistemas críticos
* Proyectos grandes (más de 2 años)
* El equipo de desarrollo no sea un equipo equilibrado

1. **Explique en que consiste la programación extrema XP**

Consiste en la programación conjunta por parejas de desarrollo donde, mientras uno programa las funcionalidades del sistema, el compañero está trabajando en programar pruebas para verificar el trabajo del compañero. Pasado un tiempo se intercambian los roles

1. **Una de las características de las metodologías agiles es que los procesos de especificación, diseño e implementación están:**

Entrelazados.

1. **Los procesos de software los podemos clasificar entre aquellos que están dirigidos por un plan y aquellos**

Dirigidos por una metodología ágil

1. **Citar tres principios definitorias de las metodologías agiles**

* Client in the loop (el cliente está presente durante el proceso)
* Entrega incremental (se entrega todo el tiempo cosillas)
* Diseño simple
* Personas no procesos
* Rápida adopción del cambio

1. **Cuáles son los roles que contempla SCRUM**

PD 🡪 Propietario del desarrollo

SM🡪 Scrum Manager

E 🡪 Equipo

I 🡪 Interesados

1. **Duraciones de SCRUM**

Planificacion del sprint: Una jornada de trabajo

Reunion diaria: 15 minutos

Revision del Sprint: Maximo 4 horas y moderada por el SM

1. **Enumere las ventajas que presenta el diseño arquitectónico**

* Comunicación con los participantes
* Análisis del sistema
* Reutilización a gran escala

1. **¿Qué es el diseño arquitectónico?**

Es un proceso creativo cuyo fin es entender cómo debe organizarse el sistema y como debe diseñarse su estructura global.

1. **¿Quién decide la arquitectura del sistema?**

Los requisitos no funcionales

1. **El punto de vista es una forma de descubrir, obtener y recopilar elementos**

* No, se llama entrevista
* Sí, pero se llama Rol
* No, se llama escenario
* **Ninguna de las anteriores**

1. **¿Cómo se llama el enlace crucial entre el diseño y la ingeniería de requerimientos que identifica los principales componentes estructurales y la relación entre ellos?**

Diagrama de flujo

1. **¿Cuáles son las vistas que componen el modelo 4 + 1?**

* Vista lógica
* Vista de proceso
* Vista de desarrollo
* Vista física
* Vista conceptual

1. **Si el rendimiento es un factor crítico, los componentes del sistema deberán ser:**

Grandes

1. **Si la disponibilidad es un factor crítico, la arquitectura del sistema deberá ser:**

Redundante

1. **En cuanto a la arquitectura en repositorio hay dos nombres que definen la forma en que los datos pueden ser actualizados en las diversas bases del modelo**

Replicación y sincronización.

Replicación 🡪 Para modelos de datos diferentes

1. **Donde se ha empleado profundamente un símil del patrón de arquitectura y filtro en esta asignatura**

Diagrama de flujos de datos.

1. **En qué tipo de documento se detalla la arquitectura del sistema**

En el diseño arquitectónico.

1. **Enumera 5 tipos de arquitectura de aplicaciones**

* Arquitectura en capas
* Arquitectura de repositorios
* Arquitectura de tubería y filtro
* Arquitectura cliente servidor
* Arquitectura del modelo vista controlador

1. **Dibujar un esquema de la arquitectura de filtro**

PROCESOS

DATOS DATOS TRANSFORMADOS

1. **Uno de los siguientes no es una forma de especificación de requisitos del sistema.**

Notación científica.

1. **Como se define un patrón**

* Nombre
* Descripción
* Descripción del problema
* Descripción de la solución
* Consecuencias

1. **Esquema MVC (Modelo Vista Controlador)**

Es un patrón de arquitectura de SW que separa los datos y la lógica de negocio de una aplicación de la interfaz de usuario y el módulo encargado de gestionar los eventos y comunicaciones

1. **De que tres modelos generales se derivan la mayoría de código abierto**

La licencia publica general GNU se conoce como licencia “reciproca”

La licencia pública menos general GNU es una variante de la licencia anterior

La licencia Berkeley Standard Distribution es una licencia no reciproca

1. **Licencias GPL y GNU**

GPL 🡪 si uso código GPL mi código también debe serlo. Publicamos el código fuente

GNU 🡪 no estás obligado a publicar lo que has hecho pero, si modifico algo tengo que publicarlo.

1. **Software intermediario que maneja las comunicaciones entre componentes**

Middleware

1. **Características de los componentes**

Componente🡪 Parte del sistema independiente y distribuido cuya funcionalidad está estrictamente definida y es accesible a través de interfaces bien definidas invocadas por mecanismos bien definidos.

* Estandarizado
* Independiente
* Componible
* Implementable
* Documentado

1. **Conflictos de diseño**

* Transparencia
* Apertura
* Escalabilidad
* Seguridad
* Calidad de servicio
* Gestión de fallos

1. **Sistemas distribuidos, estilos arquitectónicos**

Arquitectura:

* Maestro-Esclavo
* Cliente-Servidor (dos niveles)
* Cliente-Servidor multinivel
* Componentes distribuidos
* Peer to peer

1. **¿Qué es un sistema distribuido?**

Es un conjunto de ordenadores que están trabajando haciendo que parezca solo uno

1. **Ventajas de un sistema distribuido**

* Compartición de recursos
* Apertura
* Concurrencia
* Escalabilidad
* Tolerancia a fallos

1. **Tipos de ataque de los que se defiende un sistema distribuido**

* Intercepción (atacan las comunicaciones entre partes del sistema)
* Interrupción (atacan a los servicios del sistema)
* Modificación (se cambian los datos o secuencias del sistema)
* Fabricación (se crea información q no debería existir y luego la usa para conseguir ciertos privilegios)

1. **Soporte que da el middleware**

* Interacción entre componentes del sistema
* La provisión de servicios comunes

1. **Características de las metodologías ágiles**

* Los procesos de especificación, diseño e implementación están entrelazados.
* El sistema se desarrolla en diferentes versiones
* Las interfaces de usuario del sistema se desarrollan usando con frecuencia un sistema de elaboración interactivo

1. **¿Por qué la documentación en las metodologías ágiles es mínima?**

Por qué usa un desarrollo incremental y el código usado por los programadores está muy bien comentado

1. **¿En dónde han tenido mucho éxito las metodologías ágiles?**

* En desarrollar un producto mediano o pequeño para poder vender
* Diseñar un sistema a medida dentro de un organización

1. **¿Cuál es el fundamento más importante de las metodologías ágiles?**

Mantener la simplicidad

1. **Las metodologías ágiles presentan unos problemas con la documentación del SW. ¿A qué se debe?**

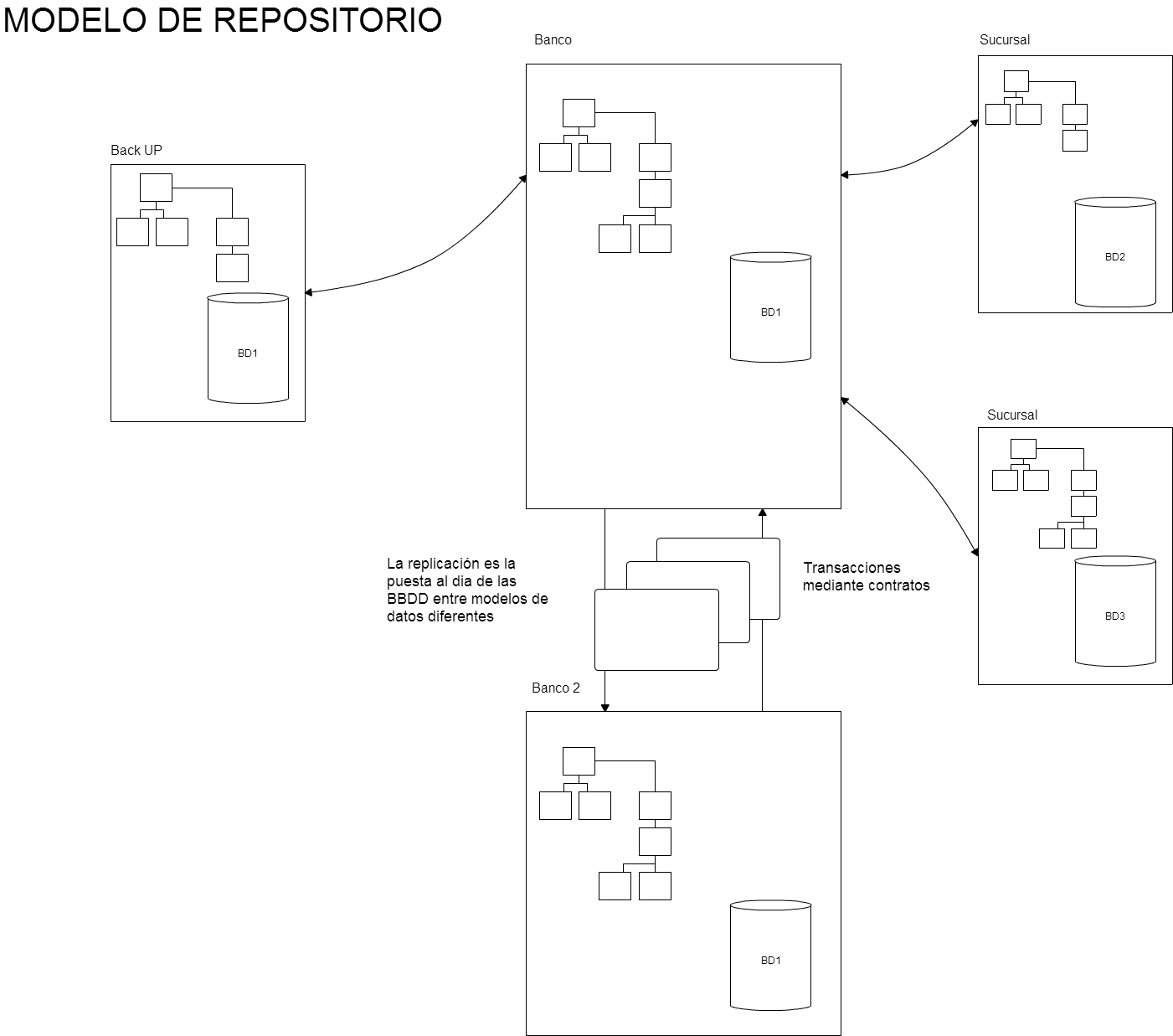
* El cliente puede no estar disponible
* El equipo que creo el SW, puede estar disuelto
* Poca documentación al crearse con el proceso incremental

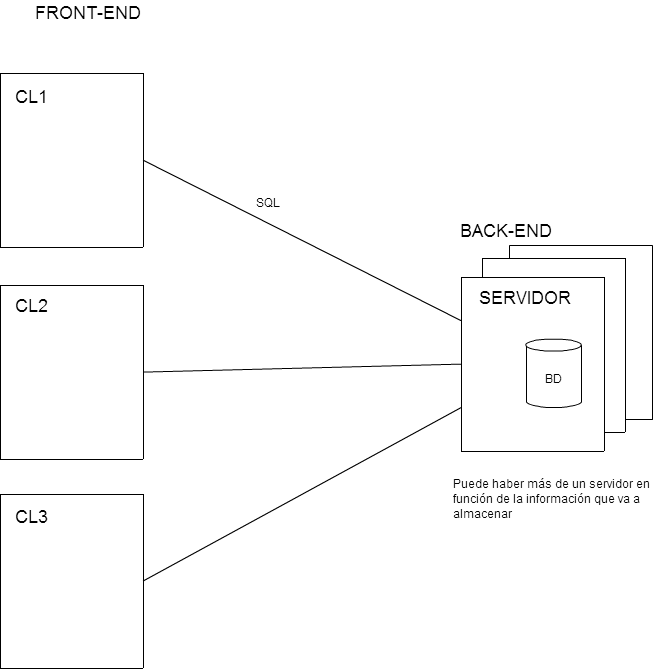
1. **Debido a la estrecha relación entre los requerimientos no funcionales y la arquitectura de SW, depende de una serie de factores. ¿Cuáles son?**

* Rendimiento (debe localizar operaciones críticas dentro de los componentes)
* Seguridad (proteger los activos más críticos en las capad más internas del sistema)
* Protección (tener las operaciones relacionadas con la protección en un lugar determinado)
* Disponibilidad (se debe poder incluir, modificar, sustituir y actualizar componentes del sistema)
* Mantenibilidad (se debe poder sustituir los componentes con facilidad)

1. **¿En qué niveles se puede llevar a cabo la reutilización de software?**

* Nivel de abstracción, usando el conocimiento de abstracciones exitosas
* Nivel objeto, reutilizando directamente los objetos de una librería
* Nivel componente, reutilizando los componentes
* Nivel sistema, reutilizando la aplicación al completo

****

****

Modelo Cliente - Servidor