1. **Responda las preguntas a final del capítulo: (1-12), incluyendo las que se refieren a temas más amplios como la 9.4, donde debe diseñar una BD que le permita administrar los ECS.**

**9.1 ¿Por qué es cierta la primera ley de la ingeniería de sistemas? ¿Cómo afecta a nuestra percepción de los paradigmas de la ingeniería del software?**

Es totalmente cierta porque siempre existirán nuevos requerimientos o ,como se da muchas veces, los usuarios no tienen completamente claro que es lo que quieren o que necesitan realmente y a medida que lo van notando piden mas cosas.

Nosotros solemos creer que luego de que se hace entrega formal del software ya no hay que trabajar mas en el, que esta completamente cerrado, y es falso. Un software nunca estará terminado en su totalidad, como dije antes, siempre habrán mejoras, correcciones que se tendremos que hacer para satisfacer las necesidades del cliente.

**9.2 Exponga las razones de la existencia de líneas base con sus propias palabras.**

Para tener una base para futuros proyectos ademas de controlar los cambios realizados y asegurar que cualquier cambio hecho sea justificado y formalmente evaluado y documentado.

**9.3 Asuma que es el gestor de un pequeño proyecto. ¿Qué líneas base definiría para el proyecto y cómo las controlaría?**

* Linea Base :Especificación del sistema
  + Como las controlaría:
    - realizando reuniones constante con los clientes para analizar sus necesidades
    - Mantener una documentación detallada y clara de todo el proceso
    - dar seguimiento a las documentaciones para validar que vayan acorde a las especificaciones dadas por los clientes.
* Linea Base :Especificación de pruebas
  + Como las controlaría:
    - Evaluar junto al equipo de trabajo todos los casos que deben ser probados
    - Realizar un plan documentado de los procedimientos a realizar durante las pruebas

**9.4 Diseñe un sistema de base de datos que permita a un ingeniero del software guardar, obtener referencias de forma cruzada, buscar, actualizar, cambiar, etc., todos los elementos de la configuración del software importantes.**



**- ¿Cómo manejaría la base de datos de diferentes versiones de un mismo programa?**

Mediante tabla Cambio en la misma se almacenan un código para dicho cambio así como la configuracion afectada y el programa. Igualmente se almacena la versión anterior al cambio y la nueva versión.

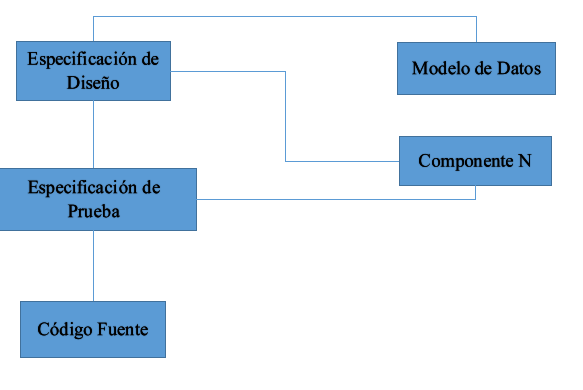
**- ¿Se manejaría de forma diferente el código fuente que la documentación?**

Si, La documentación estará asignada al programa por lo que al hacer un cambio en la documentación se deberá hacer mediante el o los programas directamente afectados. Si luego se desea saber cuales documentaciones fueron afectadas se harían una del cambio cruzada la tabla programa\_documento donde se indican cuales documentos se ven afectados por el programa cambiado.

**- ¿Cómo se evitaría que dos programadores hicieran cambios diferentes sobre el mismo ECS al mismo tiempo?**

Existiría un trigger en la base de datos para impedir que un mismo programa sea modificado al mismo tiempo por programadores diferentes.

**9.6 Utilice un modelo E-R (Capítulo 12) para describir las interrelaciones entre los ECS (objetos) de la Sección 9.1.2.**



Aquí se puede observar como los modelos de datos y los componentes N se relacionan con la especificación del diseño así como el mismo componente N y el código fuente forman parte crucial de la especificación de pruebas. Igualmente se puede ver que tanto la especificación de diseño como la especificación de pruebas dependen una de la otra.

**9.7 Investigue sobre herramientas de GCS existentes y describa cómo implementan el control de versiones, de cambios de objetos de configuración de software**

Una herramienta muy utiliza es Git, la misma posee una interfaz gráfica fácil de usar. Ademas de que tiene varias características como:

-Cada versión registrada en el tiempo es una revisión.

-Cada operación se realiza en el repositorio local.

-Permite enviar cambios a repositorios remotos.

-Es multiplataforma.

-Permite crear varias ramas para evitar hacer cambios directos en la rama principal hasta estar totalmente probados.

**9.8 Las relaciones ≪parte-de≫ e ≪interrelacionado≫ representan relaciones sencillas entre los objetos de configuración. Describa cinco relaciones adicionales que pudieran ser útiles en el contexto de la base de datos del proyecto.**

- contenido-por

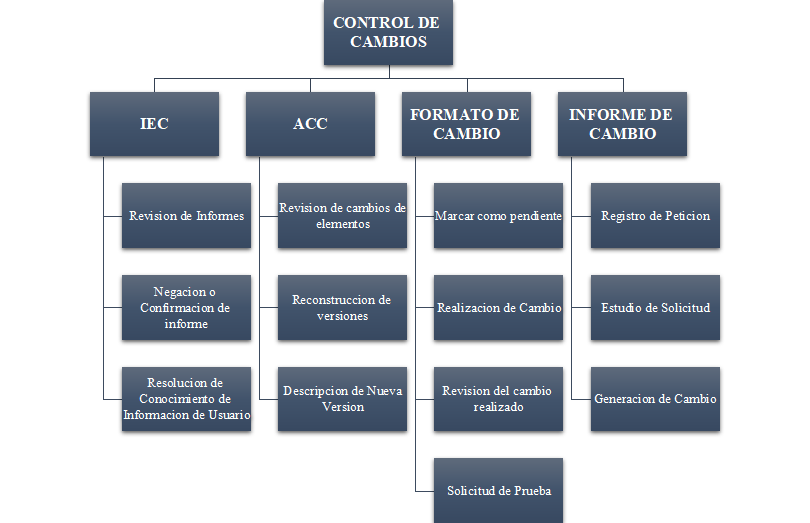
-hasta-de

-desde-de

-depende-de

-conjunto-a

**9.10 Utilizando la Figura 9.5 como guía, desarrolle un esquema de trabajo más detallado aún para el control de cambios. Describa el papel de la ACC y sugiera formatos para la petición de cambio, el informe de cambios e IEC.**

****

**9.11 Desarrolle una lista de comprobaciones que se pueda utilizar en las auditorías de configuración.**

* ¿Se ha realizado el cambio en el OCI?
* ¿Las modificaciones adicionales han sido asociadas?
* ¿Se ha llevado a cabo una revisión técnica formal para evaluar la corrección técnica?
* ¿El proceso del software ha sido seguido y aplicado adecuadamente en base a los estándares de ingeniería del software?
* ¿En el ECS se resaltaron los cambios?
* ¿Fue especificada la fecha de cambio y autor?
* ¿Los cambios los caracteres del objeto de Configuración son figurados?
* ¿Se marcaron los cambios en el ECS?
* ¿Se detalla la fecha y el autor del cambio?
* ¿Se refleja la caracterización del ECS los cambios?

**9.12 ¿Cuál es la diferencia entre una auditoría de GCS y una revisión técnica formal? ¿Se pueden juntar sus funciones en una sola revisión? ¿Cuáles son los pros y los contras?**

Las auditorías de GCS se concentran en el cumplimiento de estándares ya preestablecidos por las organizaciones internacionales o globales, mientras que la revisión técnica formal permite evaluar los resultados obtenidos, los cuales son evaluados por los desarrolladores y clientes.