

Administración de la configuración del software 22.5

Diseño del sistema de base de datos del proyecto (Repositorio de Configuración)

Este sistema de base de datos permitirá a un ingeniero de software almacenar, referenciar, rastrear, actualizar y cambiar todos los ítems de configuración de software importantes. A continuación, se describe su diseño y funcionamiento.

1. Entidades principales

- Item de Configuración (IC): ID_IC, Nombre, Tipo, Descripción, Estado, Proyecto_asociado.
- Versión: ID_Versión, ID_IC, Número de versión, Fecha de creación, Autor, Comentarios, Ruta.
- Historial de cambios: ID_Cambio, ID_Versión, Fecha, Desarrollador, Descripción, Aprobado por.
- Usuario: ID_Usuario, Nombre, Rol.
- Bloqueo de edición: ID_Bloqueo, ID_IC, ID_Usuario, Fecha_inicio, Fecha_fin, Estado.

2. Manejo de diferentes versiones del mismo programa

Cada cambio a un IC crea una nueva entrada en la tabla 'Versión'. Se utiliza una convención tipo SemVer (major.minor.patch). Las versiones anteriores se conservan para referencia o rollback.

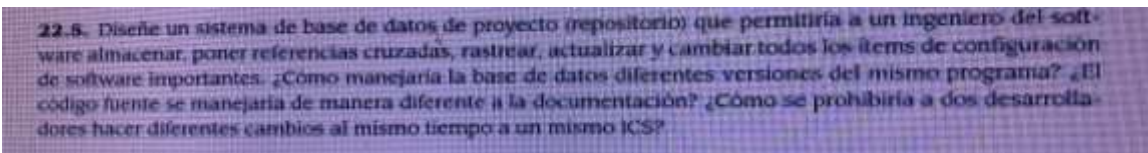
3. ¿El código fuente se manejaría de manera diferente a la documentación?

Sí. Ambos se tratan como ICs, pero el código fuente se integra con sistemas como Git y activa pruebas automáticas, mientras que la documentación se almacena como archivos y su control de versiones es más sencillo.

4. ¿Cómo prohibir que dos desarrolladores editen el mismo IC al mismo tiempo?

Se implementa un sistema de bloqueo lógico. Cuando un desarrollador edita un IC, se crea un registro de bloqueo que impide a otros editarlo simultáneamente. Alternativamente, puede usarse un sistema de ramas y merge.

Explicación :



22.8. Diseñe un sistema de base de datos de proyecto (repositorio) que permitiría a un ingeniero del software almacenar, poner referencias cruzadas, rastrear, actualizar y cambiar todos los ítems de configuración de software importantes. ¿Cómo manejaría la base de datos diferentes versiones del mismo programa? ¿El código fuente se manejaría de manera diferente a la documentación? ¿Cómo se prohibiría a dos desarrolladores hacer diferentes cambios al mismo tiempo a un mismo IC?

1. Almacenamiento de ítems de configuración

Ítems de configuración (ICs) son todos los elementos que forman parte del software y deben mantenerse bajo control, como:

- Código fuente
- Documentación
- Manuales de usuario
- Casos de prueba
- Diagramas

Estos se almacenan con información adicional (metadatos), como:

- Quién lo creó
- Cuándo se creó
- Qué versión es
- Qué cambios se le han hecho

2. Referencias cruzadas

Permiten ver cómo se relacionan entre sí los diferentes ítems. Por ejemplo:

- Un diagrama UML puede estar relacionado con cierto código fuente.
- Un documento de requisitos puede tener relación con casos de prueba específicos.

Esto facilita la trazabilidad, es decir, seguir cómo se implementa cada requerimiento.

3. Control de versiones

Cada vez que se hace un cambio en un archivo, el sistema guarda una nueva versión sin eliminar la anterior. Así se puede:

- Volver a una versión anterior si algo sale mal.
- Ver el historial completo de cambios.
- Saber quién cambió qué y cuándo.

Se puede usar Git para esto, que es una herramienta muy común en el desarrollo de software.

4. Diferenciación entre código y documentación

Aunque todo se guarda en el mismo sistema, el tratamiento es diferente:

Código fuente:

- Cambia más seguido.
- Se necesita probar automáticamente que funcione (por ejemplo, usando Jenkins).
- Se revisa cuidadosamente antes de integrarlo al proyecto.

Documentación:

- Cambia menos.
- Es más visual o textual.
- Se revisa entre colegas, no necesariamente con pruebas automáticas.

5. Prevención de conflictos de edición

Evita que dos personas modifiquen el mismo archivo al mismo tiempo:

- Bloqueo (check-out exclusivo): solo una persona puede editar el archivo hasta que lo devuelva.
- Edición concurrente: todos pueden editar, pero luego hay que fusionar los cambios usando herramientas como Git.

Esto protege la integridad del proyecto y evita sobrescribir el trabajo de otros.

Ventajas del sistema

- Trazabilidad: puedes seguir todo el historial de cada archivo.
- Seguridad: se evitan pérdidas o cambios no autorizados.
- Colaboración: varias personas pueden trabajar ordenadamente.
- Auditoría: se puede verificar fácilmente qué se hizo y por quién.