

Hoja 1 de Ejercicios

Ciclo de Vida del Software

Inicio: Semana del 23 de Enero.

Duración: 1 semana.

1) ¿Para qué vale el estudio de viabilidad? ¿Qué aspectos hay que estudiar?. ¿Cuál es más importante?

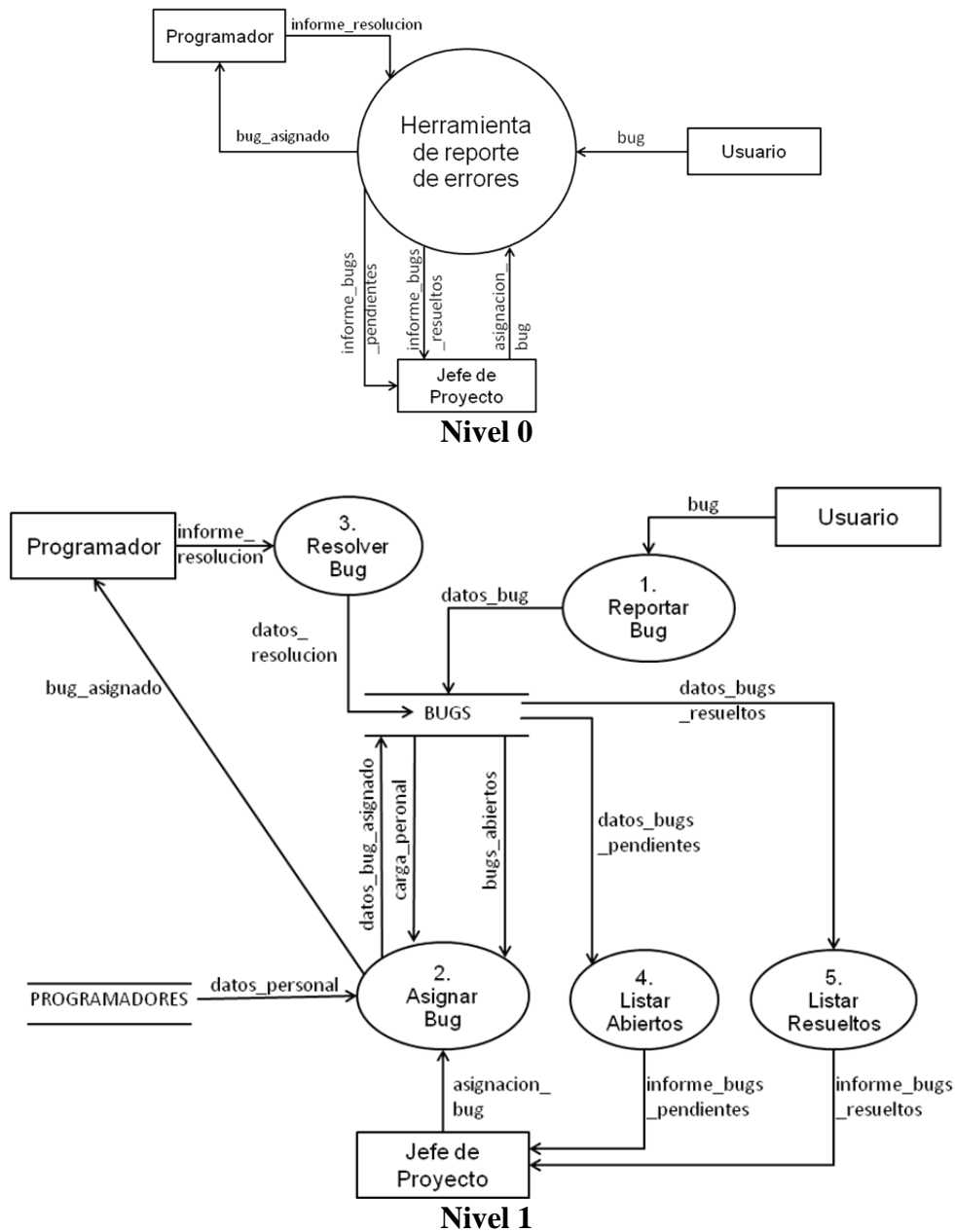
2) Clasifica los siguientes requisitos en funcionales, y no funcionales. Dentro de estos últimos indica el tipo.

1. La aplicación debe incluir una opción de menú para la impresión de todos los listados.
 2. La Base de Datos de la aplicación debe soportar hasta un máximo de 5 millones de registros.
 3. La respuesta de autorización de crédito debe ser en menos de 30 secs, el 90% de las veces.
 4. El dispositivo de punto de venta tendrá una pantalla táctil en panel grande y plano. El texto debe ser visible desde un metro.
 5. El sistema debe tener una recuperación robusta cuando el acceso a sistemas externos (tales como el inventario, impuestos, etc.) falla.
 6. El sistema deberá incluir versiones de la ayuda en español e inglés.
-

3) De los siguientes requisitos, indica cuál de ellos no cumple alguno de los atributos de corrección:

1. "... hasta 15 autobuses se dibujarán dentro de la misma ventana. Si excede el número se utilizará otra ventana diferente."
 2. "El sistema tendrá una interfaz de usuario sencilla de utilizar."
 3. "Los usuarios deben escribir su contraseña en un tiempo menor de 15 segundos desde que escribieron su identificación."
 4. "El tiempo de respuesta para todos los comandos será menor de 0.1 segundos. El tiempo de respuesta para el comando 'DELETE' será menor de 5 segundos."
 5. "El sistema tendrá un tiempo de respuesta aceptable."
-

4) Describe el funcionamiento de la siguiente aplicación, modelada mediante un Diagrama de Flujo de Datos. ¿Qué otros aspectos de la aplicación habría que modelar en la fase de requisitos?



5) En el diseño de software, ¿Qué es la modularidad? ¿y la ocultación de información?.

6) ¿Qué son las pruebas de unidad y cuándo hay que hacerlas?

7) Si se hacen pruebas de caja blanca ya no hay que hacer pruebas de caja negra. ¿Verdadero o Falso?

8) Selecciona el modelo de ciclo de vida más adecuado para los siguientes escenarios.

1. Un software que controle un reactor nuclear con los objetivos de prever posibles fallos, detectar anomalías, recomendar sugerencias de tácticas y realizar un seguimiento de las acciones llevadas a cabo por los operadores. La definición del sistema software es conflictiva y presenta altos riesgos en diferentes etapas del desarrollo. Además, las pruebas son difíciles de realizar, ya que el sistema sólo puede pasar a la fase de explotación (uso) cuando esté completamente probado, por lo que se pretende construir un simulador como ayuda a esta fase.
2. Un software que realice las gestiones de un departamento de ventas de una editorial donde las funciones están claramente identificadas. El sistema software que se tiene que implementar apenas presenta riesgos, sus fases de desarrollo están bien definidas y tu empresa tiene gran experiencia en el desarrollo de este tipo de aplicaciones
3. Un software para planificación del stock de vinos que una bodega tiene en distintos almacenes. El sistema utilizará algoritmos de planificación basándose en la demanda del mercado y en el suministro de los proveedores. Esta aplicación no presenta, a priori, grandes riesgos aunque se prevé que puedan surgir requisitos nuevos durante el desarrollo.
4. Se quiere construir una aplicación que permita acceder a través de Internet a información sobre las obras que hay en un museo. El cliente cree tener muy claros los requisitos de la aplicación. Sin embargo, el Ingeniero de Software no opina lo mismo por lo que quiere verificar los requisitos de usuario lo antes posible, en particular los relacionados con la funcionalidad y la interfaz de usuario. Técnicamente no parece presentar ningún problema.