

# Hoja 1 de Ejercicios

## *Ciclo de Vida del Software*

**Inicio: Semana del 28 de Enero.**

**Duración: 1 semana.**

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

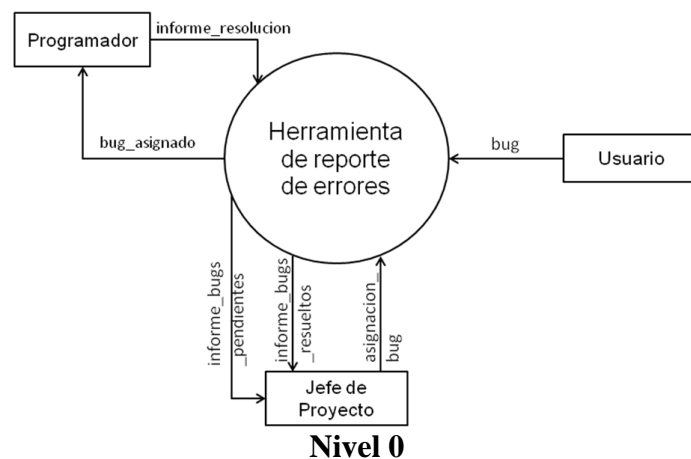
2) Clasifica los siguientes requisitos en funcionales, y no funcionales. Dentro de estos últimos indica el tipo.

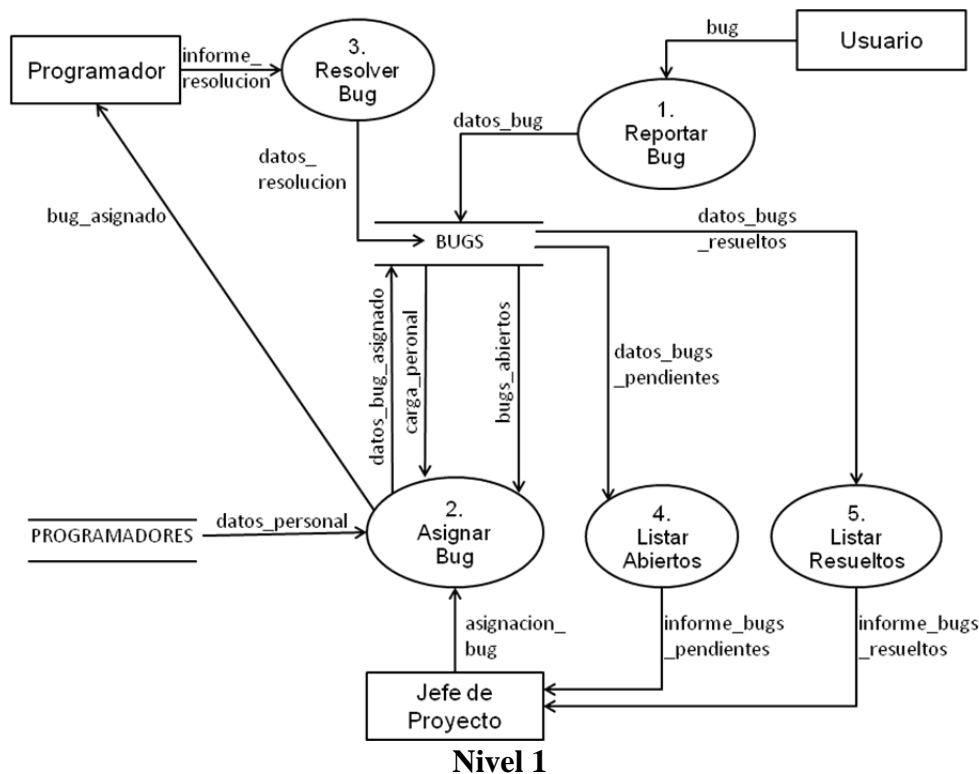
1. La aplicación debe incluir una opción de menú para la impresión de todos los listados.
2. La Base de Datos de la aplicación debe soportar hasta un máximo de 5 millones de registros.
3. La respuesta de autorización de crédito debe ser en menos de 30 secs, el 90% de las veces.
4. El dispositivo de punto 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) Modela mediante un *Diagrama de Casos de Uso* la aplicación del apartado 4. Describe el escenario “Asignar Bug” en el que el Jefe de Proyecto asigna un bug pendiente a un Programador.

6) La cadena de restauración “*las 1001 tostas*” te encarga la realización de una aplicación informática para la automatización del servicio a los clientes de sus distintos locales. El funcionamiento es el siguiente:

Cuando un cliente entra en el establecimiento, un camarero le asigna una mesa. Una vez en la mesa, el cliente rellena un formulario con las tostas y bebidas que desea. Ahora esta selección se hace con lápiz y papel, pero se pretende que el cliente lo haga mediante un dispositivo con pantalla táctil. El cliente puede seleccionar bien tostas específicas de entre una lista de 100, o bien un número de entre “las más populares”, o de “la selección del chef”. El sistema debe pues presentar cuáles son las 20 tostas más pedidas de ese día, así como las 12 tostas seleccionadas por el chef. Si un cliente quiere más comida o bebida realiza una nueva selección y el proceso se repite. Cuando el cliente realiza la orden, se le presenta el precio actual de su consumición.

La información sobre la comida seleccionada por cada cliente se presenta inmediatamente al cocinero (en un dispositivo instalado en la cocina), y el de la bebida al barman (en un dispositivo instalado en la barra). Cuando la comida o la bebida están listas, el cocinero o el barman lo indican en el sistema. Este le presenta un aviso al camarero asignado a la mesa correspondiente (que está equipado de un dispositivo móvil), que se encargan de llevarlas a la mesa.

El sistema debe además controlar el stock de alimentos en almacén. El cocinero se encarga de introducir dos veces al día la información sobre los nuevos alimentos adquiridos o los utilizados. Cuando un ingrediente se agota en el almacén, las correspondientes tostas no deben poder ser seleccionadas por los clientes.

Finalmente, cuando un cliente se va, lo indica mediante el dispositivo de la mesa. Esto hace que se imprima su recibo en la caja central, y se le presenta un aviso al camarero asignada a esas mesas, que le cobra (sólo se acepta dinero en efectivo). En cualquier momento antes de abandonar su mesa, de manera opcional el cliente puede rellenar un formulario para evaluar el servicio recibido. Este formulario ha de ser personalizable.

- a) Realiza un diagrama de casos de uso.
  - b) Describe el escenario correspondiente a realizar una petición de comida/bebida por parte del cliente. Incluye tanto el flujo principal y los alternativos si los hubiera.
- 

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

---

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

---

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

---

10) 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.