Actividad Práctica capítulo 3 UF1286

Enunciado:

Este caso práctico tiene como objetivo que el estudiante aplique conocimientos adquiridos en las distintas fases del desarrollo de software: análisis, diseño, desarrollo e implementación. Para ello, se presentan cuatro ejercicios prácticos que abarcan desde la elección de un ciclo de vida de software, hasta la validación y verificación de código fuente.

Ejercicios:

1. Se quiere desarrollar un sistema de software que controle la venta de entradas de un cine denominado Cine +. Paralelamente a la venta de entradas en la taquilla, el sistema debe dar soporte a la venta de entradas por internet, debiendo estar ambos coordinados. Como técnico de software, debe seleccionar, justificando la elección, el ciclo de vida más adecuado para este sistema.

2. Imagine una página web con un sistema de autenticación en la que se permita crear mensajes en un foro de discusión. Como técnico responsable de análisis de requisitos, dé una posible descripción del caso de uso “Crear mensaje foro” para el sistema. Se debe poder añadir un título de mensaje y un comentario más detallado. El moderador debe aceptar el comentario antes de que pueda ser publicado para toda la comunidad. Además, podría rechazar el comentario si lo determina no adecuado al hilo de discusión y solicitar al internauta algún dato que no ha sido rellenado correctamente.

3. Forma parte del equipo de desarrollo de un proyecto de software y le piden que obtenga un diagrama de entidad-relación del sistema. El sistema debe llevar un control de los proveedores, clientes, productos y ventas, teniendo en cuenta que cada proveedor provee a la empresa con productos y que estos se clasifican en categorías. Cada venta está compuesta de uno o varios de estos artículos y el cliente puede realizar todas las compras que desee. Dé una posible solución.

4. Se ha finalizado la fase de implementación en su proyecto y, a continuación, se debe validar y verificar el programa. Para detectar posibles anomalías, se van a utilizar métodos automatizados de análisis. Elabore un informe con el resultado de las posibles anomalías que encontrarían estos métodos si se aplicasen al siguiente trozo de código (proporcionado en el ejercicio).

```
<in binsearch(char *word, char *wordaux, int n)
{
   int cond;
   int low, high, mid;
   int suma;

   high = n - 1;
   while (low <= high) {
       mid = (low+high) / 2;
       cond = strcmp(word, wordaux));
       low = low + 1;
   }
   return -1;

   low = low + 1;
}
```

Forma de Entrega:

El caso práctico deberá entregarse en formato pdf:

ApellidoNombre_CasoPractico3.pdf

1. Se quiere desarrollar un sistema de software que controle la venta de entradas de un cine denominado Cine +. Paralelamente a la venta de entradas en la taquilla, el sistema debe dar soporte a la venta de entradas por internet, debiendo estar ambos coordinados. Como técnico de software, debe seleccionar, justificando la elección, el ciclo de vida más adecuado para este sistema.

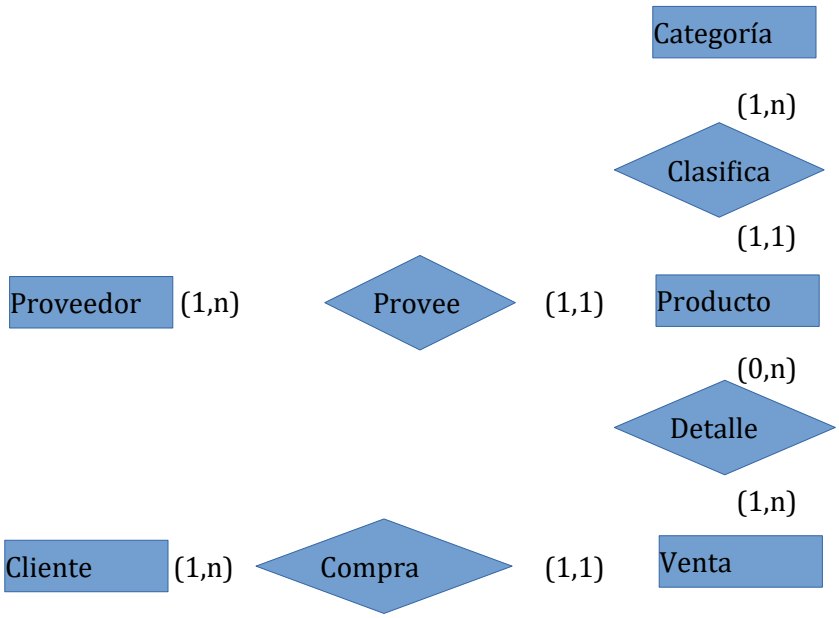
El ciclo de vida debe ser iterativo, ya que el cliente no ha dado muchos detalles del programa que quiere, y tampoco se indica un plazo de desarrollo, con lo que cuando vea en funcionamiento la primera versión, se percatará de lo que quiere, para realizar la siguiente version, o si necesita mas funcionalidades pasar de iterativo a incremental, que son muy parecidos

2. Imagine una página web con un sistema de autenticación en la que se permita crear mensajes en un foro de discusión. Como técnico responsable de análisis de requisitos, dé una posible descripción del caso de uso "Crear mensaje foro" para el sistema. Se debe poder añadir un título de mensaje y un comentario más detallado. El moderador debe aceptar el comentario antes de que pueda ser publicado para toda la comunidad. Además, podría rechazar el comentario si lo determina no adecuado al hilo de discusión y solicitar al internauta algún dato que no ha sido rellenado correctamente.

Nombre:	Crear mensaje foro
Descripcion	El usuario escribe un mensaje para publicarlo en el foro
Actores	Usuario/moderador
Precondiciones	- El usuario está autenticado - El usuario puede publicar mensajes
Flujo común	- El usuario pulsa en responder o en nuevo mensaje - El usuario escribe el titulo del mensaje - Y el mensaje de forma mas detallada - Y le da al boton de publicar
Flujo normal	- El sistema comprueba que el mensaje tiene cumplimentados los campos obligatorios - El moderador verifica que el mensaje no infringe ninguna norma del foro ni ninguna ley,

	- El moderador autoriza la publicacion
Flujo alternativo 1	- El mensaje esta incompleto - El sistema envia un mensaje al usuario indicandole el campo que tiene que cumplimentar
Flujo alternativo 2	- El mensaje infringe una norma o una ley - El moderador no autoriza la publicacion - El moderador Evalua bloquear la cuenta del usuario y/o tomar acciones legales
Postcondiciones	El mensaje se guarda sin publicar a la espera de la decision del moderador

3. Forma parte del equipo de desarrollo de un proyecto de software y le piden que obtenga un diagrama de entidad-relación del sistema. El sistema debe llevar un control de los proveedores, clientes, productos y ventas, teniendo en cuenta que cada proveedor provee a la empresa con productos y que estos se clasifican en categorías. Cada venta está compuesta de uno o varios de estos artículos y el cliente puede realizar todas las compras que desee. Dé una posible solución.



4. Se ha finalizado la fase de implementación en su proyecto y, a continuación, se debe validar y verificar el programa. Para detectar posibles anomalías, se van a utilizar métodos automatizados de análisis. Elabore un informe con el resultado de las posibles anomalías que encontrarían estos métodos si se aplicasen al siguiente trozo de código (proporcionado en el ejercicio).

Un plan de pruebas debería tener los siguientes apartados:

• **El proceso de prueba:** una descripción de las principales fases del proceso de prueba.

• **Trazabilidad de requerimientos:** es necesario hacer un seguimiento de los requerimientos y dependencias entre ellos cuando se producen las pruebas. De todas formas, es primordial que cada requerimiento se someta a pruebas de forma individual.

• **Elementos probados:** especificación de los elementos que van siendo probados.

• **Calendario de pruebas:** un calendario donde se asignen recursos a los procesos de prueba. Procedimientos de registro de las pruebas: se debe llevar una documentación de pruebas coherente y consistente. De esta forma, se podrá realizar cualquier tipo de auditoría del proceso de pruebas para comprobar que ha sido satisfactorio y óptimo.

• **Requerimientos de hardware y software:** existen herramientas que permiten hacer una gestión más completa de los procesos de prueba.

• **Restricciones:** durante el proceso de prueba pueden aparecer eventualidades que afecten negativamente al desarrollo del plan. Esta sección es un intento de anticipación a todas aquellas circunstancias.

Los métodos formales son técnicas que utilizan métodos matemáticos para describir las propiedades del sistema y poder verificar a través de análisis matemático que el sistema realiza las operaciones correctas.

El modelo de **Sala Limpia** se basa en cinco puntos clave:

1. Especificación formal: el software se debe especificar formalmente, utilizando un modelo de transición de estados donde se muestren las respuestas a los diversos tipos de eventos.

2. Desarrollo incremental: el software se divide en incrementos que se desarrollan y validan de forma independiente utilizando el proceso de sala limpia. Estos incrementos se especifican, con la información de los clientes,en una etapa temprana del proceso. La ingeniería del software de sala limpia es un enfoque que defiende la necesidad de corregir el software a medida que este se desarrolla.

3. Programación de estructura: se utiliza solo un número limitado de estructuras de control y de abstracciones de datos.

4. Verificación estática: el software desarrollado se verifica estáticamente utilizando inspecciones de software rigurosas. No existe ningún proceso de prueba de unidades o módulos para los componentes del código.

5. Pruebas estadísticas del sistema: el incremento del software integrado es probado estadísticamente, para determinar su fiabilidad. Estas pruebas estadísticas se basan en un perfil operacional, el cual se desarrolla paralelo a la especificación del sistema.

Los métodos de análisis estático son métodos formales que permiten validar y verificar el software sin necesidad de ejecutarlo. A estos métodos se les conoce como métodos automatizados de análisis y su principal objetivo es avisar a los técnicos de pruebas de posibles anomalías en el sistema.A continuación, se muestra un listado de posibles anomalías:

Clase de defecto	Comprobación del análisis estático
Defectos de datos	Variables utilizadas antes de su inicialización. Variables declaradas pero nunca utilizadas. Variables asignadas dos veces pero nunca utilizadas entre asignaciones. Posibles violaciones de los límites de los vectores. Variables no declaradas.
Defectos de control	Código no alcanzable. Saltos incondicionales en bucles.
Defectos de entrada/salida	Las variables salen dos veces sin intervenir ninguna asignación.
Defectos de interfaz	Inconsistencias en el tipo de parámetros. Inconsistencias en el número de parámetros. Los resultados de las funciones no se utilizan. Existen funciones y procedimientos a los que no se les llama.
Defectos de gestión de almacenamiento	Punteros sin asignar. Aritmética de punteros.