



**UNIVERSIDAD AUTONOMA DEL ESTADO DE MEXICO
CENTRO UNIVERSITARIO UAEM ATLACOMULCO**

INGENIERIA EN COMPUTACIÓN

MATERIA:
ANALISIS DE SISTEMAS

PROFESOR:
ING. GREGORIO GRACIA ESTRADA

ALUMNOS:
LUÍS ALBERTO GARCÍA SOSA
EDGAR JUÁREZ SEGUNDO
JUAN CARLOS MEJIA GONZÁLEZ
JOSÉ CARLOS MONROY PLATA

ICO-10

AVANCE #3

Abril DE 2009

Caso de uso y escenarios.

Los casos de uso se componen de una secuencia de eventos iniciada por el usuario; describen el sistema a desarrollarse por lo tanto es de vital importancia saber quiénes son sus usuarios.

Los casos de uso son ideas simples y practicas que no requieren muchas habilidades tecnológicas para ser utilizadas.

Cada caso de uso constituye un flujo completo de eventos, que especifican la interacción que toma lugar entre usuarios y el sistema.

La instancia de caso de uso existe mientras este se siga ejecutando.

La ejecución del caso de uso nuevo termina cuando el actor genera un evento que requiere un caso de uso nuevo.

Las diferentes instancias de los casos de uso se conocen como **escenarios**.

Como varios casos de uso pueden comenzar de una misma forma, no es siempre posible decidir que caso de uso se ha instanciado, hasta que este se haya completado.

Retomando el aspecto de los escenarios comenzaremos con: Escenario peor, escenario medio y escenario perfecto en estos aspectos nos imaginaremos o presentaremos que aspectos se verán en cada escenario y como se resolverán

Peor escenario:

Como debemos saber lo peor que puede ocurrir cuando se implementa un nuevo sistema es que el software no funcione. Y esto es algo preocupante; sin embargo, en algo así tan malo se agrega la documentación es decir que los usuarios sigúan ciertas instrucciones y que el sistema se comporte de manera distinta. Y esto afecta pues tanto nosotros como desarrolladores y creadores perdemos credibilidad sin dejar afuera nuestro sistema. Obviamente al perder credibilidad pues será muy difícil recuperarla. Por lo que para este escenario hemos recreado algunos problemas que en el peor de los casos que podrían ocurrir.

Se tiene el programa desarrollado totalmente y nosotros como desarrolladores hemos realizado pruebas con datos ficticios pero nos encontramos con problemas para dar altas, mientras lo demás funciona perfectamente se verifica el código y rectifica el error, se vuelven a realizar pruebas encontrando nuevos problemas en los módulos que funcionaban perfectamente, se deshizo el código implementado y nos encontramos con el mismo problema solo falla un modulo, por lo que se decidió quitarlo y realizar ese modulo de nuevo solucionando el problema.

Nos dirigimos a la empresa donde se iba a instalar el programa, hablamos con el responsable de el área (gerente, jefe de piso, presidente de empresa) asignándonos el equipo en el cual al momento de intentar instalar nos marca errores pues el equipo que nos proporcionaron no cumple los requisitos necesarios, se le notifico al responsable el cual nos proporciono en seguida el equipo solicitado.

Se intenta instalar el programa ahora en el nuevo equipo y aparece un nuevo error que nos habla de la falta de un archivo .dll necesario para la ejecución de nuestro programa lo corregimos descargando dicha .dll logrando instalar el programa correctamente.

Procedemos a hacer pruebas con datos reales y nuestra sorpresa es que el modulo que nos genero errores y el que se volvió a

realizar es el único que funciona correctamente haciendo que todo nuestro tiempo invertido para la realización del programa y en la instalación, sea en vano provocando enojo en nuestros clientes los cuales casi nos demandan por no cumplir con el contrato y por lo tanto se fueron con el software del vecino.

Escenario Medio:

En este escenario se han hecho pruebas del sistema comprobando la integración de cada modulo del sistema.

También hemos buscado las discrepancias entre cada modulo y el objetivo original (lo que nuestro cliente a solicitado que realice nuestro programa), especificaciones.

Nos hemos preocupado más por la compatibilidad de los módulos independientes.

Tratamos de hallar áreas en donde los módulos hayan sido diseñados con especificaciones distintas para la longitud y tipo de datos y los nombres de los elementos de los datos.

Por ejemplo, un modulo puede pedir o esperar que el tipo de dato para un platillo o cliente sea un campo numérico, mientras que otros esperan que sea de tipo carácter, es posible que nuestro sistema no reporte un error en la introducción de datos pero la salida si puede mostrarnos resultados inesperados y sin saber porque o en el peor de los casos; si un registro creado y guardado en un modulo, usando la identificación como tipo numérico, se busca posteriormente como tipo carácter, el campo podría ser no reconocido y nos aparecerá el mensaje “el registro no fue encontrado”, provocándonos perdida o teniendo información parcial además de que se pierde tiempo(registrar los datos, búsqueda del error o encontrar el motivo de dicho mensaje).

También hemos verificado que los tamaños de los archivos son adecuados y que los índices se han construido en forma adecuada. Pues hemos hecho todo lo posible para encontrar errores dentro de nuestro sistema colocándolo en situaciones exageradas como sencillas. Todo esto para encontrar errores en la forma en que el sistema los procesa.

El mejor escenario:

Si bien el peor escenario que podríamos encontrar es el que no funcione correctamente nuestro sistema el mejor escenario es todo lo contrario.

Nuestro sistema funciona correctamente de pies a cabeza, la documentación es buena y entendible y va acorde a lo que realiza nuestro programa y nuestro cliente quedo muy satisfecho por nuestro trabajo.

Se han realizado las pruebas necesarias para examinar las especificaciones que señalan lo que el programa debe hacer y como lo debe llevar a cabo bajo diferentes condiciones.

Después se procede a desarrollar casos de prueba para cada uno de todas las posibles condiciones o situaciones que se puedan presentar.

Se realizo un estudio de resultados para determinar si el programa funciona perfectamente.

En una de las pruebas realizadas se coloco a nuestro sistema como una caja negra; en esta situación no fue de vital importancia estudiar el código o verificar si se ocuparon todas y cada una de las instrucciones o rutas dentro del programa ya que hipotéticamente si el programa cumple las especificaciones, no fallara.

Entre algunas circunstancias que nos ayudaron a determinar o verificar que nuestro sistema tiene un mayor desempeño o que se encuentra en un buen escenario son:

-Con carga máxima: todas las terminales se activan al mismo tiempo y quieren acceder al mismo tiempo al servidor.

-Almacenando: determinando la capacidad del sistema para almacenar datos de transacciones en un disco u otros archivos.

-Ejecutándolo: Tiempo de respuesta a las consultas cuando el sistema esta cargando en su totalidad con datos operativos

-Recuperando: cargando copias de respaldo de datos y reiniciar el procedimiento sin la preocupación de perder datos o integridad.

-Procedimientos: haciendo que los usuarios lleven a cabo exactamente lo que el manual pide para su buen funcionamiento.

-Factores humanos: determinamos como utilizan los usuarios el sistema al procesar datos o preparar informes y si su experiencia es favorable