

SENA

Centro De Electricidad Electrónica Y Telecomunicaciones (CEET)

FEBRERO ,2021

BOGOTA D.C

Ficha:2057746

Diego Alejandro Cáceres Salazar

Guisli Nataly Ramírez Sanabria

Tecnólogo Análisis Y Desarrollo De Sistemas De Información (ADSI)



CARBOGRES SAS

1. Aprendizaje Cooperativo:

Con sus equipos de trabajo, se debatirán los interrogantes propuestos y generarán un ensayo de máximo una (2) página de extensión donde se exprese la opinión del grupo respecto a:

- o Como desarrollador de software, ¿Qué errores considera que se cometieron? ¿Fue un problema de programación?
- o ¿Cuáles lecciones cree que le dejó este incidente a la Agencia Espacial Europea?
- o ¿Reusar un código que funciona bien será siempre una buena opción?
- o ¿Cuáles crees que debieron ser las medidas preventivas que el equipo desarrollador del Ariane 5 tenían que aplicar?



CARBOGRES SAS

• Recorte rectangular

El en desarrollo de este software hubieron mucho errores al implantarlo ya que hubieron sucesos que no eran admisibles a la hora de diseñar este programa para implementarlo en un cohete ya que estos estaba diseñados para versiones de cohetes antiguos y no contaban con las mejoras en estructura que podían hacer más pesado a la hora del despegue ya que el programa estaba realizado para cohetes mas livianos que este haciéndolos, a la hora del despegue el cohete Ariane 5 despegó más rápido incrementado su velocidad ya que el programa determino que necesitaba mas velocidad para poder despegar, aparte de esto el código tenía que transformarse mientras se usaba el despegue es decir tenía que pasar de numero número almacenado en coma flotante de 64 bits a entero de 16 bits.

Este incidente le dejó como lección a la Agencia Espacial Europea que se debe revisar todos los posibles errores de software antes de realizar un lanzamiento de un cohete ya que entre estos podemos creer que se puede realizar de tal manera que no tenga falla y pueden incurrir a estas fallas las cuales pueden llegar a ser mortales.

Reusar código es bueno y malo a la vez es bueno porque a la hora de programar reducimos una gran cantidad de tiempo también cuando las tareas que realiza este código son repetitivas ya que se puede reutilizar ya que va a estar realizando esta actividad siempre pero tiende a tener fallas ya que al la hora de implementar puede ocasionar fallas ya que podemos tener algunas clases llamadas de distintas formas y esto puede generar errores incluso podemos llegar a tener errores con los nombres de las bases de datos, es malo ya que si no estamos pendientes a la hora de reusar este tipo de código podemos pasar por alto algunos nombres pueden ser de las clases, variables, objetos o incluso el nombre de base de datos y estos errores podrían generar atrasos a la hora de crear o desarrollar un programa de software.

Las experiencias que le pudieron quedar a los desarrolladores del Ariane 5 son:

CARBOGORES SAS

Que a la hora de programar revisar muy bien cada tipo de valor que utilizamos ya que podemos tener varios tipos de datos que no son compatibles por ejemplo los un tipo de dato INT o un tipo de dato VARCHAR, ya que el un tipo de dato VARCHAR puede almacenar datos de todo tipo en cambio el un tipo de dato INT solo almacena datos numéricos y esto a la hora de que el programa lo interprete va a generar errores operacionales ya que tomamos la hora o ponemos una cuenta regresiva en un tipo de dato VARCHAR no va a funcionar de la misma manera que lo realiza un tipo de dato INT

A la hora de implementar un software debemos ver la compatibilidad y la funcionalidad que va a realizar ya que podemos pensar que el código que estamos diseñando va a cumplir con esta función, pero en realidad estamos haciendo algo parecido pero que no cumple con toda su funcionalidad en pocas palabras debemos estar muy atentos de cumplir con los requerimientos del sistema que se desea implementar ya que si no ponemos atención a esto podemos generar grandes fallas las cuales pueden ser muy costosas o asta pueden llegar a la muerte de algunas personas como en los ejemplos vistos en la actividad de reflexión inicial en el por culpa de un error humano y del programa hubieron sobredosis de radiación a la hora de realizar las radiografías y esto provocó la muerte de 7 personas y que otras 20 tuvieron problemas de salud, o también como el de la fallas en las redes de las aerolíneas que provocaron pérdidas económicas, retraso en los vuelos y afectar las agendas de las personas que tenían planeado viajar.

2. actividad aprendizaje autónomo:

En este momento de APROPIACIÓN DEL CONOCIMIENTO es propicia la conceptualización con el fin de establecer un marco teórico inherente al presente resultado de aprendizaje desarrollado.

Por lo anterior, creará un mapa conceptual utilizando los recursos digitales en línea suscritos por el SENA (<http://biblioteca.sena.edu.co/paginas/bases.html>) donde se incluyan los siguientes conocimientos:

CARBÓGRÉS SAS

PRUEBAS DE SOFTWARE

¿Qué Son?

Las pruebas de software (en inglés software testing) son las investigaciones empíricas y técnicas cuyo objetivo es proporcionar información objetiva e independiente sobre la calidad del producto a la parte interesada o stakeholder.
Es una actividad más en el proceso de control de calidad

¿Cuáles hay?

PRUEBAS
UNITARIA


PRUEBAS
CAJA
BLANCA

PRUEBAS
CAJA
NEGRA


PRUEBAS
FUNCIONARIAS

PRUEBAS
HUMO


PRUEBAS
REGRESION




En programación, una prueba unitaria es una forma de comprobar el correcto funcionamiento de una unidad de código. Por ejemplo en diseño estructurado o en diseño funcional una función o un procedimiento




Las pruebas de caja blanca se centran en los detalles procedimentales del software, por lo que su diseño está fuertemente ligado al código fuente.




Las Pruebas de Caja Negra, es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software



Una prueba funcional es una prueba de tipo caja negra basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software.



Las pruebas de humo son una revisión rápida de un producto de software para comprobar que funciona y no tiene defectos evidentes que interrumpen la operación básica del mismo.



Las pruebas de regresión son cualquier tipo de pruebas de software con el objeto de descubrir errores, carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software

7. ... 1GEK ... A

PRUEBAS SISTEMA

se prueba el sistema de forma global. En esta etapa pueden distinguirse los siguientes tipos de pruebas, cada uno con un objetivo claramente diferenciado

PRUEBAS ESTRESS

consiste en probar los límites que un sistema puede soportar. En este tipo de pruebas se suele enviar más peticiones de las que el software podría atender normalmente para saber el comportamiento de la aplicación.

PRUEBAS DESEMPEÑO

las pruebas de rendimiento son las pruebas que se realizan, desde una perspectiva, para determinar lo rápido que realiza una tarea un sistema en condiciones particulares de trabajo

PRUEBAS DE INTEGRIDAD Y BASES DE DATOS

Pruebas de integridad de base de datos son pruebas de los métodos y procesos utilizados para acceder y gestionar datos para asegurar que los métodos de acceso, los procesos y las reglas de los datos funcionan como se espera y que durante el acceso a la base

PRUEBAS DE SEGURIDAD Y CONTROL DE ACCESO

Las pruebas de seguridad se podrían definir como el conjunto de actividades que se llevan a cabo para encontrar fallas y vulnerabilidades en aplicaciones web

PRUEBAS DE GUI

pruebas de GUI y pruebas basadas en GUI, necesitan ejecutar eventos de forma manual o automática, simulando el uso de la aplicación y reconocer el estado de la GUI y detectar un error en esta.

Actividad aprendizaje autónomo:

Remitiéndonos a los casos descritos en la ACTIVIDAD DE REFLEXIÓN INICIAL y usando el conocimiento adquirido, usted creará una tabla donde identifique los sucesos ocurridos, acciones o medidas correctas que se debieron tomar y lección aprendida o recomendaciones para tener en cuenta en proyectos futuros.

Acontecimiento presentado	Acción / Medida Correcta para solucionar	Recomendación / Lección aprendida
Colapso en el Sistema aeroportuarios en los ángeles	Análisis del código para impedir el bloqueo de el sistema para mantener buen funcionamiento	Mejorar el manejo de código del software y que sea lo suficientemente liviano para no provocar bloqueos en los sistemas
Sobredosis radiológica en el instituto nacional del cáncer de Panamá	Mejorar el procedimiento y el código para así poder administrar la dosis correcta a la hora de realizarse las radiografías	Tener mas cuidado a la hora de realizar los procedimientos hechos por las personas implicadas en el funcionamiento del sistema
Problemas en diversos aeropuertos del mundo por fallas en el sistema de facturación de varias aerolíneas	Mejorar la conexión a la red de las aerolíneas para que no sucedan estas fallas y así tener un buen funcionamiento de ellas	Mejorar y actualizar constantemente la red a la cual están conectadas las aerolíneas en cada uno de estos países

CARBOGRES SAS