**INSTITUTO TECNOLÓGICO DE CHIHUAHUA II**

**INGENIERÍA EN SISTEMAS COMPUTACIONALES**

**TALLER DE BASE DE DATOS**



**Tarea 2 “Traducción de textos”**

**Primera Evaluación Parcial**

Martin Andrés Alanís Rivera 16550022

Chihuahua, Chih. a 16 de Febrero de 2018

Indice

[INTRODUCCIÓN 3](#_Toc506584810)

[OBJETIVO 4](#_Toc506584811)

[Errores de Software de computadora seriamente épicos 5](#_Toc506584812)

[Fallo de "Sangre Corrupta" de World of Warcraft 7](#_Toc506584813)

[Apagón norteamericano 8](#_Toc506584814)

[Incidente USS Yorktown 9](#_Toc506584815)

[Crisis de misiles de la Guerra Fría 10](#_Toc506584816)

[Protección contra copia maliciosa CD SONY 11](#_Toc506584817)

[Error Misil Patriot 12](#_Toc506584818)

[CONCLUSIONES 13](#_Toc506584819)

[RECOMENDACIONES 14](#_Toc506584820)

[Referencias 15](#_Toc506584821)

[Ilustración 1 Therac -25 5](file:///C:\Users\Martin\Desktop\Taller%20base%20de%20datos\ISC-TBD-Tarea2-16550022.docx#_Toc506584822)

[Ilustración 2 World of Warcraft 7](file:///C:\Users\Martin\Desktop\Taller%20base%20de%20datos\ISC-TBD-Tarea2-16550022.docx#_Toc506584823)

[Ilustración 3 Gente caminando atravez del puente de Queenbor en Nueva York durante el apagon 8](file:///C:\Users\Martin\Desktop\Taller%20base%20de%20datos\ISC-TBD-Tarea2-16550022.docx#_Toc506584824)

[Ilustración 4 USS Yorktown (CG-48) 9](file:///C:\Users\Martin\Desktop\Taller%20base%20de%20datos\ISC-TBD-Tarea2-16550022.docx#_Toc506584825)

[Ilustración 5 Stanislav Petrov recibiendo el Premio Dresden, 2013. 10](file:///C:\Users\Martin\Desktop\Taller%20base%20de%20datos\ISC-TBD-Tarea2-16550022.docx#_Toc506584826)

[Ilustración 6 Lanzamiento de misil Patriot 12](file:///C:\Users\Martin\Desktop\Taller%20base%20de%20datos\ISC-TBD-Tarea2-16550022.docx#_Toc506584827)

# INTRODUCCIÓN

En este documento se llevara a cabo la traducción de una serie de textos de diferentes estilos que tienen por común el error en los software y sus respectivas consecuencias

# OBJETIVO

Promover entre el estudiante la importancia del dominio del idioma inglés en su carrera profesional, debido a que la mayoría de la información relevante en el área se encuentra en este idioma.

# Errores de Software de computadora seriamente épicos

La mayoría de los errores de software son pequeños inconvenientes que pueden ser superados o solucionados por el usuario, pero hay algunos de esos notables casos donde un simple error ha afectado a millones, de un grado u otra e incluso ha causado lesiones y pérdida de vidas. El software es escrito por humanos y cada parte del software tiene errores o “Características no documentadas” como un hombre de ventas las puede llamar.

Eso es, que un software haga algo que no debería, o no hace algo que debería hacer. Estos errores se pueden deber a un mal diseño, malinterpretando un problema o solo un simple error humano como un error en un libro. Sin embargo, cuando el libro es leído por un humano quien usualmente puede deducir el significado de la palabra mal escrita, el software es leído por las computadoras, quienes son comparablemente estúpidas y harán únicamente lo que les fue dicho. Aquí hay diez casos donde las consecuencias de estos errores fueron enormes, de una forma u otra. (Jones, 2012)

El Therac-25 era una máquina para la administración de radioterapia, generalmente para tratar pacientes con cáncer. Tenía dos modos de operación, la primera consistía en un haz de electrones apuntando directamente al paciente en pequeñas dosis por un periodo pequeño de tiempo. El segundo apuntaba el haz de electrones a altos niveles de energía hacia un “blanco” de metal primero lo que esencialmente convertiría el rayo en rayos X que luego pasarían al paciente. En modelos previos de la maquina Therac, para este segundo modo de operación, había seguros físicos para asegurar que su objetivo estuviera en posición, que de no tenerlo, rayos de energía muy elevados podrían disparar directamente al paciente.

Ilustración 1 Therac -25

En el nuevo modelo, estos seguros físicos fueron remplazados por unos de Software  
Lamentablemente, hubo un error en el software: a veces se producía un "desbordamiento aritmético" durante las comprobaciones automáticas de seguridad. Esto básicamente significa que el sistema usaba un número dentro de sus cálculos internos que era demasiado grande para que lo manejara. Si, en este preciso momento, el operador estaba configurando la máquina, las comprobaciones de seguridad fallarían y el objetivo de metal no se movería a su lugar.

El resultado fue que los haces 100 veces más altos que la dosis deseada se dispararía a un paciente, dándoles envenenamiento por radiación. Esto sucedió en 6 ocasiones conocidas, causando la muerte posterior de 4 pacientes.

# Fallo de "Sangre Corrupta" de World of Warcraft

El enormemente exitoso World of Warcraft (WoW), un juego de computadora en línea creado por Blizzard Entertainment, sufrió un problema embarazoso después de una actualización de su juego el 13 de septiembre de 2005, lo que provocó la muerte masiva (ficticia)

Después de una actualización del contenido del juego, se presentó un nuevo personaje enemigo, Hakkar, que tenía la capacidad de infligir una enfermedad, llamada sangre corrupta, sobre los personajes que consumirían su salud durante un período de tiempo. Esta enfermedad podría transmitirse de jugador a jugador, al igual que en el mundo real, y podría matar a cualquier personaje que la contraiga. Este efecto estaba destinado a estar estrictamente localizado en el área del juego en la que Hakkar habitaba.

Sin embargo, una cosa se pasó por alto: los jugadores pudieron teletransportarse a otras áreas del juego mientras aún estaban infectados y transmitir la enfermedad a otros, que es exactamente lo que sucedió.

No puedo encontrar ninguna cifra sobre el recuento de cadáveres, pero ciudades enteras dentro del mundo del juego eran áreas prohibidas, con cadáveres de jugadores muertos ensuciando las calles.

Afortunadamente, la muerte del jugador no es permanente en WoW y el evento finalizó cuando los administradores del juego restablecieron los servidores y aplicaron más actualizaciones de software. Particularmente interesante es la forma en que las reacciones de los jugadores en el juego podrían reflejar de cerca sus reacciones a un incidente similar en la vida real.

Ilustración 2 World of Warcraft



Ilustración 3 Gente caminando atravez del puente de Queenbor en Nueva York durante el apagon

# Apagón norteamericano

Afectando a alrededor de 55 millones de personas, principalmente en el noreste de los Estados Unidos, pero también en Ontario, Canadá, este fue uno de los apagones de energía más grandes de la historia. Todo comenzó cuando una planta de energía a lo largo de la costa sur del lago Erie, Ohio, se desconectó debido a la gran demanda que puso al resto de la red de energía bajo una mayor presión. Cuando las líneas eléctricas tienen una carga eléctrica más pesada, se calientan, lo que significa que el material que forma el cable (generalmente aluminio y acero) se expande. Varias líneas eléctricas colgaban más bajas a medida que se expandían y atrapaban árboles, derribándolos y sometiendo al sistema a más presión.

Esto condujo a un efecto de cascada que eventualmente redujo la red eléctrica al 20% de la salida normal. Aunque las causas de este apagón no tenían nada que ver con un error de software, podría haberse evitado si no fuera por un error de software en el centro de control Sistema de alarmas.

En el llamado escenario de "condición de carrera", dos partes del sistema competían por el mismo recurso y no pudieron resolver el conflicto, lo que provocó que el sistema de alarma se congelara y detuviera el procesamiento de alertas.

Desafortunadamente, el sistema de alarma falló "en silencio", lo que significa que se rompió, pero no notificó a nadie que se había roto. Esto significaba que no se proporcionaban alertas sonoras o visuales al personal de la sala de control, que confiaba en tales cosas para la conciencia situacional. Las consecuencias fueron bien informadas y dejaron muchas áreas sin energía durante varios días y afectaron la industria, los servicios públicos, la comunicación. También se culpó como al menos un factor contribuyente en varias muertes.

# Incidente USS Yorktown

Ilustración 4 USS Yorktown (CG-48)

En el mundo del desarrollo de software, hay varios errores comúnmente conocidos que los programadores encuentran y tienen que atender. Un ejemplo es el error 'divide por cero', donde se realiza un cálculo que divide cualquier número por cero.

Tal cálculo no es posible de resolver, al menos no sin utilizar matemáticas superiores, y la mayoría del software, para todo, desde supercomputadoras a calculadoras de bolsillo: está escrito para tener en cuenta este escenario.

Fue con cierta vergüenza, entonces, que el USS Yorktown sufrió una falla completa de su sistema de propulsión y estuvo muerto en el agua durante casi 3 horas cuando un miembro de la tripulación tipeó " 0 "en el sistema de gestión de base de datos de a bordo que luego se utilizó en un cálculo de división. El software se instaló como parte de una operación más amplia para utilizar las computadoras a fin de reducir la potencia humana necesaria para que funcionen algunos buques. Afortunadamente, el barco estaba realizando maniobras en el momento del incidente, en lugar de desplegarse en un entorno de combate, lo que podría haber tenido consecuencias más graves.

# Crisis de misiles de la Guerra Fría

Stanislav Petrov era el oficial de guardia de un búnker secreto cerca de Moscú, responsable de monitorear el sistema satelital de alerta temprana soviético. Justo después de la medianoche, recibieron una alerta de que Estados Unidos había lanzado cinco misiles balísticos intercontinentales Minuteman.

Como parte de la doctrina de destrucción mutuamente segura que prevaleció durante la Guerra Fría, la respuesta a un ataque de un poder sería un ataque de venganza del otro. Esto significaba que si el ataque era genuino, tenían que responder rápidamente.

Ilustración 5 Stanislav Petrov recibiendo el Premio Dresden, 2013.

Sin embargo, parecía extraño que Estados Unidos atacara con solo un puñado de ojivas nucleares: aunque causarían un daño masivo y la pérdida de vidas, ni siquiera sería suficiente para acabar con la oposición soviética.

Además, las estaciones de radar en el suelo no captaban ningún contacto, aunque no podían detectar más allá del horizonte debido a la curvatura de la Tierra, lo que podría haber explicado la demora. Otra consideración fue el sistema de alerta temprana sí mismo, que se sabía que tenía fallas y que había sido llevado al servicio en primer lugar.

Petrov sopesó todos estos factores y decidió descartar la alerta como una falsa alarma. Aunque Petrov no tenía el dedo en el botón nuclear como tal, si hubiera recomendado a sus superiores que tomaran el ataque como real, podría haber llevado a una guerra nuclear total. Ya sea por experiencia, intuición o simplemente suerte, la decisión de Petrov fue la correcta. Más tarde se determinó que el software de detección temprana había captado el reflejo del sol desde la parte superior de las nubes y lo malinterpretó a medida que se lanzaba el misil.

# Protección contra copia maliciosa CD SONY

La guerra aparentemente interminable entre los medios y los piratas va y viene cada año. Tan pronto como se encuentran nuevas formas de proteger y distribuir de manera segura los medios, se descubren nuevas formas de eludir y poner en peligro estas medidas. Algunos argumentarían que Sony BGM fue un paso demasiado lejos en 2005, cuando introdujeron una nueva forma de protección contra copias en algunos de sus CD de audio.

Cuando se juega con una computadora con Windows, estos CD instalarían un software llamado "rootkit". Un rootkit es una forma de software que se adentra en una computadora y altera ciertos procesos fundamentales.

Aunque no siempre es de naturaleza maliciosa, un rootkit a menudo se usa para plantar sigilosamente software malicioso y difícil de detectar (o eliminar), como virus, troyanos, etc. En el caso de Sony BMG, el objetivo era controlar la forma en que una computadora con Windows utilizaron los CD de Sony para evitar copiarlos o convertirlos a MP3, lo que los ayudaría a reducir la piratería de sus medios. El rootkit logró esto, pero al tomar medidas para ocultarse del usuario, habilitó virus y otro software malicioso para escóndete junto con eso.

La implementación poco pensada, y la creciente percepción de que Sony BMG no tenía ningún negocio manipulando furtivamente las PC de los usuarios, significaba que todo el plan fracasó. Resultó en la clasificación del rootkit como malware por parte de muchas compañías de seguridad informática, así como en varios juicios y en la retirada de productos de los CD ofensivos.

# Error Misil Patriot

Durante la Operación Escudo del Desierto, el ejército estadounidense desplegó el Sistema de misiles Patriot como defensa contra aviones y misiles, en este caso misiles iraquíes Al Hussein (SCUD). El software de rastreo para el misil Patriot usa la velocidad de su objetivo y el tiempo actual para predecir dónde estará el objetivo de un instante a otro. Dado que varios objetivos pueden viajar a velocidades de hasta MACH 5, estos cálculos deben ser muy precisos.

En ese momento, había un error en el software de segmentación, lo que significaba que, con el tiempo, el reloj interno "derivaría" (como cualquier reloj) más y más de la hora exacta cuanto más tiempo se dejara funcionando el sistema.

Ilustración 6 Lanzamiento de misil Patriot

El error ya era conocido y simplemente se solucionó reiniciando regularmente el sistema y restableciendo así el reloj del sistema. Desafortunadamente, los encargados no entendieron claramente qué tan 'regularmente' deberían reiniciar el sistema, y ​​se dejó funcionando para 100 horas. Cuando se lanzó un misil iraquí, apuntando a un aeródromo estadounidense en Dhahran, Arabia Saudita, fue detectado por el sistema de misiles Patriot.

Sin embargo, en este punto, el reloj interno se había desviado en 0,34 segundos, por lo que cuando intentó calcular dónde estaría el siguiente misil, estaba mirando una zona del cielo a más de medio kilómetro de distancia de la ubicación real de los misiles. Inmediatamente asumió que no había misiles enemigos y canceló la intercepción. El misil se llevó a su destino donde mató a 28 soldados e hirió a otros 98.

# CONCLUSIONES

Los errores siempre están en el minúsculo detalle, pero principalmente el factor más importante que encuentro dentro de los errores mencionados, es que el factor humano que lo programo fue el que dio origen a todos y cada uno de los errores.

Por ejemplo, el de Fallo de Sangre corrupta de World of warcraft fue un error que no se pudo ver venir ni mucho menos sus consecuencias sobre la comunidad Gamer de World of Warcraft

Otros ejemplos enseñan que el intelecto humano puede superar los errores de los equipos como en el caso de la crisis de los misiles de la guerra fría que por medio de un pensamiento lógico se pudo determinar que era un error en el sistema.

Sin embargo tampoco se descarta el como la pereza para corregir un error puede llevar a un terrible resultado como el caso del misil Patriot. Quienes diseñadores, ingenieros y técnicos de mantenimiento conocían del error sobre el diseño del misil y sencillamente optaron por una medida rápida pero no efectiva: el reinicio. Llevando al posterior suceso terrible

# RECOMENDACIONES

Para los diseñadores de software que seremos, el que nos inculquen la cultura del Double-check&Test (termino improvisado) para poder asegurarnos que nuestros programas cumplan con todos los parámetros solicitados dentro del software al igual el que nos impulsen a tener una mayor visión sobre cada programa que hagamos para poder tener el habito de poder prever cualquier posibilidad que se nos pueda presentar.

# Referencias

Jones, A. (24 de Diciembre de 2012). *ListVerse*. Recuperado el 16 de 02 de 2018, de http://listverse.com/2012/12/24/10-seriously-epic-computer-software-bugs/