

Ingeniería del Software I



Guías Trabajos Prácticos INGENIERIA DE SOFTWARE I

Profesores: Dr. Ramon García Martínez
Dr. Dario Rodriguez
Mg. Hernán Amatriain

Ayudantes: Lic. Santiago Bianco
Lic. Sebastian Martins

2017

Índice

<i>Guía de Trabajo práctico 1: Ciclo de Vida ("El Robot Asesino")</i>	<i>Pág. 3</i>
<i>Guía de Trabajo Práctico 2: Diagramas de Flujo de Datos (DFDs)</i>	<i>Pág. 32</i>
2.1. Ejercicio 1: Títulos	<i>Pág. 32</i>
2.2. Ejercicio 2: Gestoría "Placente, Gancedo y CIA."	<i>Pág. 33</i>
2.3. Ejercicio 3: Distribuidora "Su Remedio"	<i>Pág. 34</i>
2.4. Ejercicio 4: Comprador Frecuente	<i>Pág. 35</i>
2.5. Ejercicio 5: Laboratorium	<i>Pág. 36</i>
2.6. Ejercicio 6: Ticketronn	<i>Pág. 37</i>
2.7. Ejercicio 7: Comunicación	<i>Pág. 38</i>
2.8. Ejercicio 8: Demoras	<i>Pág. 39</i>
2.9. Ejercicio 9: MacMickey	<i>Pág. 39</i>
2.10. Ejercicio 10: Administración de Alumnos	<i>Pág. 40</i>
2.11. Ejercicio 11: Importac	<i>Pág. 41</i>
2.12. Ejercicio 12: Circ_Event	<i>Pág. 41</i>
2.13. Ejercicio 13: Rascacie	<i>Pág. 42</i>
2.14. Ejercicio 14: Sargo	<i>Pág. 43</i>
<i>Guía de Trabajo práctico 3: Diagramas de Entidad Relación (DER)</i>	<i>Pág. 44</i>
3.15. Ejercicio 15: Películas para completar	<i>Pág. 44</i>
3.16. Ejercicio 16: Peluquería 1	<i>Pág. 44</i>
3.17. Ejercicio 17: Banco	<i>Pág. 45</i>
3.18. Ejercicio 18: Proyecto	<i>Pág. 45</i>
3.19. Ejercicio 19: Rayo	<i>Pág. 46</i>
3.20. Ejercicio 20: Banco Mar	<i>Pág. 46</i>
3.21. Ejercicio 21: Administración de Actas de Finales	<i>Pág. 47</i>
3.22. Ejercicio 22: Gerox	<i>Pág. 47</i>
3.23. Ejercicio 23: DER Completo	<i>Pág. 48</i>
3.24. Ejercicio 24: Editoriales	<i>Pág. 49</i>
3.25. Ejercicio 25: Buques	<i>Pág. 49</i>
3.26. Ejercicio 26: Becas RCI	<i>Pág. 50</i>
3.27. Ejercicio 27: Taller	<i>Pág. 50</i>
3.28. Ejercicio 28: Peluquería 2	<i>Pág. 51</i>
<i>Guía de Trabajo práctico 4: Diseño de Sistemas OO</i>	<i>Pág. 52</i>
4.29. Ejercicio 29	<i>Pág. 52</i>

Guía de Trabajo práctico 1: Ciclo de Vida ("El Robot Asesino")

Dinámica de trabajo:

- A continuación se presentan 9 artículos relatando un caso que la prensa denominó “El Robot Asesino”.
- Cada grupo deberá tomar varios artículos, analizarlos y responder las siguientes preguntas:
 1. Identifique los errores de Gestión de Proyecto.
 2. ¿Estaban las pruebas diseñadas correctamente? ¿Quién es el responsable del diseño de las pruebas?. Justifique.
 3. Identifique los errores en la aplicación de la metodología para Análisis y Diseño de Sistemas de Información en relación a los requerimientos y el diseño lógico.
 4. Identifique los errores de codificación.
 5. Enumere las causas que generaron el accidente. Seleccione dos causas que a su criterio sean las más relevantes. Justifique su selección.
 6. Identifique los aspectos éticos que fueron vulnerados en el caso analizado.
- Una vez respondidas las preguntas cada grupo deberá elaborar un informe en que se indique:
 - Nombre de los integrantes del grupo,
 - Artículos seleccionados, y respuestas a las preguntas del cuestionario.

1.1. INTRODUCCIÓN

El caso del Robot Asesino es una historia que combina elementos de Ingeniería del Software y de ética en la computación.

Los artículos de esta historia comienzan con la acusación por homicidio no premeditado a un programador. Este programador había escrito un código defectuoso que causó la muerte de un operador de robot, lentamente. También se presentan, a lo largo de varios artículos, factores de la corporación que también contribuyeron en el accidente. Se muestra el desarrollo del software como un proceso social, presentando varios aspectos de la compleja tarea de construir software para un mundo real, y algunos temas éticos relacionados con esta complejidad.

El texto consta de una introducción y 9 artículos:

- Los involucrados: En la cual se caracterizan los personajes de la historia.
- Artículo 1: Programador de Silicon Valley acusado por homicidio no premeditado.
- Artículo 2: Los que desarrollaron el “Robot Asesino” trabajaron bajo una enorme presión.
- Artículo 3: Los compañeros acusan: el programador del “Robot Asesino” era una estrella.
- Artículo 4: El proyecto del “Robot Asesino” controvertido tema desde el vamos.
- Artículo 5: Silicon Techtronics prometió entregar un robot seguro.
- Artículo 6: La interfaz del “Robot Asesino”.
- Artículo 7: Ingeniero del software cuestiona la autenticidad de las pruebas del software del “Robot Asesino”.
- Artículo 8: Empleado de Silicon Techtronics admite la falsificación de las pruebas del software.
- Artículo 9: Conversación con el Dr. Harry Yoder.

Vale aclarar que este punto y el texto donde se caracterizan los personajes de la historia no se deben utilizar como texto de análisis.

Este artículo fue utilizado por Richard G. Epstein. "El uso de escenarios sobre ética de la computación en la educación de Ingeniería del Software: el caso del robot asesino." Enseñanza de Ingeniería de Software: Anticipo de la 7ma. Conferencia SEI CSEE, San Antonio. Jorge L. Díaz-Herrera, editor. Notas del discurso en Computer Science 750. Springer-Verlag 1994.

1.2. LOS INVOLUCRADOS

El caso del robot asesino consta de nueve artículos publicados en periódicos, un artículo de un jornal y de una entrevista publicada en una revista. Esta historia intenta llamar la atención sobre tópicos de ética en la computación y en la ingeniería de software.

Las personas e instituciones involucradas en esta historia son enteramente ficticias (excepto por las referencias a las Universidades de Carnegie-Mellon y Purdue y a los venerables computadores científicos Ben Shneiderman y Jim Foie). Se eligió Silicon Valley para la ubicación del accidente debido a que éste es un icono de la alta tecnología. Todas las personas e instituciones nombradas en Silicon Valley son ficticias.

La caracterización de los personajes:

- **Alex Allendale**, Abogado, contratado para defender a Randy Samuels.
- **Jan Anderson**, programadora y analista de Silicon Techtronics. Se oponía al uso del modelo de cascada en el proyecto del robot y fue despedida por ser honesta.
- **Turina Babbage**, presidente de la Association for Computing Machinery (ACM). Anuncia que ACM realizaría una investigación sobre violaciones al Código de ética de la ACM por parte de los empleados de Silicon Techtronics.
- **Robert Franklin**, periodista del SENTINEL-OBSERVER de Silicon-Valley. Entrevistó al Profesor Harry Yoder para conocer la visión de un experto en ética sobre la evolución del caso del robot asesino. La entrevista fue publicada en la revista dominical del SENTINEL- OBSERVER.
- **Horace Gritty**, Profesor de Ciencias de la Computación.
- **Sandra Herdenson**, estudiante reciba en la Universidad de Silicon Valley. Coperó en la investigación sobre procedimientos de aseguramiento de la calidad en la Universidad de Silicon Valley.
- **Ray Jonson**, Jefe de División Robótica en Silicon Techtronics. La División Robótica necesitaba un robot exitoso.
- **Marta**, fuente anónima de un periódico. Una persona interna de Silicon Valley que dio al SENTINELA-OBSERVER información sobre la dinámica de grupo del proyecto del robot Robbie CX30.
- **Bart Matthews**, operador del robot. Un programa de computación con defectos causó que el robot Robbie CX30 lo matara.
- **Roberta Matthews**, viuda de Bart.
- **Jane McMurdock**, Fiscal de la Ciudad de Silicon Valley. Fue quien incrimina a Randy Samuels con los cargos de asesinato no premeditado.
- **Mabel Muckraker**, periodista del Silicon Valley Sentinel Observer. Fue designada a la historia del robot asesino por su reputación de eficaz reportera investigadora.
- **Bill Park**, profesor de física de la Universidad de Silicon Valley. Confirio que Randy Samuels malinterpreto las ecuaciones de la dinámica del robot.
- **Randy Samuels**, programador. Escribió el código del programa que causó que el robot Robbie CX30 oscilara con gran amplitud, matando a su operador, Bart Matthews.

- **Sam Reynolds**, Gerente del proyecto del CX30, Ray Johson era su superior inmediato. SU experiencia fue adquirida en el campo del procesamiento de datos, pero fue puesto al mando del proyecto CX30, muy a su pesar. Estaba a favor de aplicar el modelo de cascada del desarrollo del software.
- **Robbie CX30**, el robot. Robbie jamás tuvo un mal pensamiento hacia nadie, aún así se torno un salvaje asesino.
- **Wesley Silber**, profesor de Ingeniería del Software en la Universidad del Silicon Valley. Condujo una revisión e los procedimientos de aseguramiento de la calidad en Silicon Techtronics.
- **Sharon Skinner**, profesora de psicología en la Universidad de Silicon Valley. Veía a Randy Samuels como una persona aplicada a su tarea, susceptible por demás a las críticas.
- **Valeria Thomas**, abogada, contratada por Sam Reynolds.
- **Michael Waterson**, presidente y presidente ejecutivo de Silicon Techtronics. Puso a Sam Reynolds a cargo del proyecto Robbie CX30 como una medida para recortar gastos. Contribuyo generosamente a la campaña de reelección de James McMurdock. Contrato al Dr. Silber para llevar adelante una investigación sobre aseguramiento de calidad en Silicon Techtronics.
- **Max Worthington**, jefe de seguridad de Silicon Techtronics. Monitoreaba la comunicación electrónica entre los empleados y así descubrió a Cindy Yardley.
- **Ruth Whitterspoon**, analista y programadora vocera del comité “Justicia para Randy Samuels”. Defendía a Randy Samuels sobre la base que Silicon Techtronics estaba legalmente obligada a entregar un robot confiable y seguro.
- **Cindy Yardley**, empleada y probadora de software de Silicon Techtronics. Admitió la falsificación de las pruebas de software con el fin de resguardar el puesto de trabajo de sus compañeros.
- **Harry Yoder**, profesor de tecnología de computación y ética. Examino, en una entrevista publicada en la revista dominical del SENTINEL – OBSERVER, la tensión entre las responsabilidades individuales y corporativas.

1.3. LOS ARTICULOS

1.3.1. Artículo 1- Programador de Silicon Valley acusado por homicidio no premeditado. El error del programa causó la muerte del operador del robot

Especial para el SENTINEL – OBSERVER de Silicon Valley.

Jane McMurdock, fiscal de la ciudad de Silicon Valley, anunció en la fecha la acusación de Randy Samuels con los cargos de asesinato no premeditado. Samuels trabajaba como programador en Silicon Techtronics, Inc., una de las empresas mas nuevas de Silicon Valley en la arena de la alta tecnología. El cargo involucra la muerte de Bart Matthews, quien fuera muerto el pasado mes de mayo por un robot de la línea de armado.

Matthews, quien trabajaba como operador de robot en Cybernetics, Inc., en Silicon Heights, fue aplastado y murió como consecuencia de ello, cuando el robot que estaba operando produjo un malfuncionamiento y comenzó a oscilar su “brazo” violentamente. El brazo del robot alcanzó a Matthews, arrojándolo contra una pared y aplastando su cráneo. Matthews murió en forma casi instantánea en un caso que conmocionó e indignó a muchos en Silicon Valley. De acuerdo con el dictamen de cargos, Samuels fue quien escribió la pieza del programa de computadora en particular, que fue la responsable de la falla del robot. “Hay una evidencia incriminatoria”, anunció triunfante, McMurdock en una conferencia de prensa mantenida en la Corte.

“Tenemos la fórmula manuscrita, suministrada por el físico del proyecto, que se suponía que tenía que programar Samuels. Pero negligentemente malinterpretó la formula, y esto llevó a una horrible muerte. La sociedad debe protegerse de los programadores que comenten errores descuidadamente o de lo contrario nadie estará a salvo, y menos que nadie nuestra familia e hijos”, dijo:

- El SENTINEL – OBSERVER ha podido obtener una copia de la muestra de la fórmula manuscrita en cuestión. En realidad, existen tres fórmulas similares, garabateadas en un papel amarillo de un block borrador tamaño oficio. Cada una de las fórmulas describe el movimiento del brazo del robot en una dirección: este – oeste, norte- sur y arriba – abajo.
- El SENTINEL – OBSERVER mostró entonces las fórmulas a Bill Park, profesor de física de la Universidad de Silicon Valley. Este confirmó que estas ecuaciones podían ser usadas para describir el movimiento del brazo del robot.
- El SENTINEL – OBSERVER mostró entonces el código del programa a Bill Park, escrito por el acusado en lenguaje C de programación. Preguntamos al Profesor Park, quien está muy familiarizado con este y muchos otros lenguajes de programación, si el código era o no correcto para las formulas dadas del brazo del robot.
La respuesta del Profesor Park fue inmediata: “¡Por Júpiter!”. Parece que interpretó los puntos y de las fórmulas como barras y, e hizo lo mismo con las x y las z. Se suponía que tenía que usar derivadas, pero en su lugar tomo los promedios!. Si me preguntan, ¡es culpable como el mismo demonio!”.
 - El SENTINEL – OBSERVER no pudo contactar a Samuels para entrevistarlo. “Se encuentra profundamente deprimido por esto”, nos dijo la novia por teléfono. “Pero, Randy cree que va a aliviarse en cuanto pueda decir su versión de la historia.”.

1.3.2. Artículo 2 - Los que desarrollaron al “Robot Asesino” trabajaron bajo una enorme presión

Especial para el SENTINEL – OBSERVER de Silicon Vallley. Silicon Valley, EEUU por Mabel Muckraker

El SENTINEL – OBSERVER tomó conocimiento hoy que Randy Samuels y otros que trabajaron en el proyecto del “robot asesino” en Silicon Techtronics, estuvieron bajo tremendas tensiones para finalizar el software del robot para el 1° de enero de este año. Según una fuente bien informada, los altos niveles gerenciales advirtieron a los integrantes del staff del proyecto que “rodarían cabezas” si no se cumplía con el objetivo del 1° de enero.

Randy Samuels, programador de Silicon Techtronics, fue acusado la semana pasada bajo cargo de asesinato no premeditado en el ahora famoso caso del “Robot asesino”. Samuels escribió el software defectuoso que causó que el robot industrial de Silicon Techtronics, Robbie CX30, aplastara y lesionara fatalmente al operador, Bart Matthews. Matthews era un operador de robot de Cybernetics, Inc. Conforme a la fiscal de Silicon Valley, Jane McMurdock, Samuels malinterpretó la formula matemática, “volviéndolo al inofensivo Robbie un salvaje asesino”.

Nuestra fuente informada, quien desea mantenerse en el anonimato y a quien llamaremos “Marta” por el resto de este artículo, tiene un íntimo conocimiento de todos los aspectos del proyecto Robbie CX30. Marta dijo al SENTINEL – OBSERVER que existía una enorme fricción entre el jefe de División de Robótica, Ray Jonson y el gerente del proyecto de Robbie CX30, Sam Reynolds. “Se odiaban a muerte” manifestó Marta al SENTINEL – OBSERVER en una entrevista exclusiva.

“Hacia junio del año pasado el proyecto se encontraba atrasado seis meses y Johnson se puso furioso. Había rumores que echarían a toda la División de Robótica, que él lideraba, si Robbie (el robot CX30) no daba muestras de ser un éxito comercial. Él (Jonson) llamo a Sam (Reynolds) a su oficina y realmente lo destruyó. Quiero decir, uno podía oír los gritos desde el fondo de la oficina. Jonson le dijo a Sam que terminara el proyecto para el 1° de enero o de lo contrario “rodarían cabezas”.

“Yo no estoy diciendo que Jonson le ordenara a Sam acortar camino”, agrego Marta. “Creo que la idea de cortar camino estaba implícita. El mensaje fue: “acortá camino si querés mantener tu puesto”.

De acuerdo con documentos provistos por Marta al SENTINEL – OBSERVER, el 12 de julio del año pasado fueron agregados al proyecto Robbie CX30 veinte nuevos programadores. Esto ocurrió días después de la tomentosa reunión entre Jonson y Reynolds que Marta contó.

De acuerdo a Marta, los nuevos contratados eran un desastre. “Jonson, unilateralmente, hizo los arreglos de estas contrataciones, seguramente desviando recursos de otros aspectos del proyecto Robbie (CX30). Reynolds se oponía con vehemencia a esto. Jonson solo conocía acerca de la fabricación del hardware. Esa era su especialidad. No pudo haber entendido de las dificultades que nosotros estamos teniendo con el software de la robótica. Usted no puede acelerar un proyecto de software agregando mas gente. No es como una línea de montaje”.

Según Marta y otras fuentes, la contratación de estos nuevos veinte programadores llevó a que se hiciera una reunión entre Johson, Reynolds y todos los integrantes del proyecto de software de Robbie CX30. “Esta vez fue Sam (Reynolds) el que se puso furioso. Se quejó de que el proyecto no necesitaba más gente. Sostuvo que el problema principal era que Jonson y otros miembros a nivel directivo no entendían que el Robbie CX30 era fundamentalmente diferente a otras versiones anteriores del robot”. Estas fuentes dijeron al SENTINEL – OBSERVER que los nuevos empleados no estaban totalmente integrados al proyecto, aún seis meses después de su ingreso, cuando diez robots Robbie CX30, incluido el robot que mato a Bart Matthews, ya habían sido despachados. Según Marta, “Sam solo quería mantener las cosas lo mas simples posible. No quería que el nuevo personal complicara las cosas”. “Se pasaron seis meses leyendo manuales. La mayoría de los empleados nuevos no sabia nada de robots y Sam no estaba como para perder tiempo tratando de enseñarles”. Según Marta, la reunión del 12 de junio se hizo famosa en la corporación Silicon Techtronics porque fue en esa reunión donde Ray Jonson anuncio su “Teoría Ivory Snow [“no existe el blanco perfecto, o bien, no hay más blanco que el blanco nieve”] de diseño y desarrollo de software”. De acuerdo a Marta, “Ray (Johson) nos dio una gran presentación en multimedia, con diapositivas y todo. La esencia de esta “Teoría Ivory Snow” es simplemente que el blanco nieve es 99,44 % puro y que no hay razón por la que el software de robótica deba ser más puro que esto. Dijo repetidas veces que “El software perfecto era un oximoron”.

Marta y otros personajes anónimos que se acercaron con información, retrataron a Jonson como un gerente con una desesperada necesidad de ser ayudado por un éxito en el proyecto. Versiones anteriores de Robbie, CX10 y CX20, fueron experimentales por naturaleza y nadie esperaba que fueran éxitos comerciales. De hecho, la División Robótica de Silicon Techtronis estaba operando con sus finanzas en rojo desde su concepción 6 años atrás. O triunfaba el CX30 o Silicon Techtronics quedaría fuera del negocio de robótica industrial. “Los robots Robbie anteriores tuvieron mucha prensa, especialmente aquí en Silicon Valley”, dijo otra fuente que también quiere permanecer anónima. “Robbie CX30 iba a capitalizarse con la buena publicidad generada por los proyectos anteriores. Lo único es que Robbie CX30 era mas revolucionario de lo que Jonson quería admitir. CX30 representaba un paso gigante hacia delante en términos de sofisticación. Había muchísimas preguntas acerca de los parámetros industriales en los que debería trabajar el CX30. Mucho de lo que debía ejecutarse era completamente nuevo, pero Jonson nunca lo pudo entender. Él solo nos veía como unos perfeccionistas. Uno de sus dichos favoritos era ‘La perfección es enemiga de lo bueno’”.

1.3.3. Artículo 3 - Los compañeros acusan: “El programador del ‘Robot Asesino’ era una estrella”

Especial para el SENTINEL – OBSERVER de Silicon Vallley. Silicon Valley, EEUU por Mabel Muckraker

Randy Samuels, el que fuera el programador de Silicon Techtronics que fue acusado por escribir el software que causó el horrible incidente del “robot asesino” el pasado mes de mayo, era aparentemente una “prima donna” que encontraba muy difícil aceptar criticas, aseguraron hoy varios compañeros de trabajo.

En una rueda de prensa con varios compañeros de trabajo de Samuels en el proyecto del “robot asesino”, el SENTINEL – OBSERVER pudo obtener importantes revelaciones acerca de la psiquis del hombre que pudo haber sido criminalmente responsable de la muerte de Bart Matthews, operador de robot y padre de 3 criaturas.

Con el permiso de los entrevistados, el SENTINEL – OBSERVER permitió a la profesora Sharon Skinner del Departamento de Psicología de Software en la Universidad de Silicon Valley, escuchar una grabación de la entrevista. La profesora Skinner estudia la psicología de los programadores y otros factores psicológicos que tienen impacto en el proceso de desarrollo del software.

“Estaría de acuerdo con la mujer que lo llamó “prima donna””, explicó la profesora Skinner. “Este es un término utilizado para referirse a un programador que simplemente no puede aceptar las críticas, o más precisamente, no puede aceptar su propia falibilidad.”

“Randy Samuels tiene lo que nosotros, psicólogos de programadores, llamamos una personalidad orientada hacia una tarea, lindando con una personalidad orientada hacia sí mismo. Le gusta poder completar cosas, pero su ego esta muy densamente involucrado en su trabajo. En el mundo de la programación esto se considera “no, no”, agregó la profesora Skinner en su oficina tapizada en libros.

La profesora Skinner continuó explicando algunos hechos adicionales sobre equipos de programación y personalidades del programador. “Básicamente, hemos encontrado que un buen equipo de programación requiere de una mezcla de personalidades, incluyendo a una persona que esté orientada hacia la interacción, que saca una enorme satisfacción del hecho de trabajar con otra gente, alguien que pueda ayudar a mantener la paz y a que las cosas se muevan en una dirección positiva. Muchos programadores están orientados hacia lo que es la tarea, y esto puede ser problemático si se tiene un equipo donde son todos de este modo”.

Los compañeros de trabajo de Samuels se mostraron muy reticentes a culpar a alguien por el desastre del robot, pero cuando se los presionó para que comentaran la personalidad de Samuels y sus hábitos laborales, surgieron varios hechos importantes. Samuels trabajaba en un equipo formado aproximadamente por una docena de analistas, programadores y probadores de software. (Esto no incluye a 20 programadores que fueron incorporados posteriormente y que nunca llegaron a estar activamente involucrados en el desarrollo del software de la robótica). Si bien cada individuo del equipo poseía una especialidad, casi todos están comprometidos en todo el proceso del software del principio al fin.

“Sam Reynolds tenía un background en el procesamiento de datos. Dirigió unos cuantos proyectos de software de esta naturaleza”, dijo uno de los integrantes del equipo, refiriéndose al gerente del proyecto Robbie CX30. “Pero su rol en el proyecto era mas que nada de líder. Asistía a todas las reuniones importantes y lo mantenía a Ray (Ray Johnson jefe del Departamento de robótica) sobre nuestras espaldas lo mas posible”. Sam Reynolds, como ya fuera informado en el SENTINEL – OBSERVER de ayer, se encontraba bajo una severa presión para lograr producir un robot Robbie CX30 operativo para el 1° de enero de este año. Sam Reynolds no pudo ser ubicado para entrevistarlo ya sea sobre su rol en el incidente o sobre Samuels y sus hábitos de trabajo.

“Éramos un equipo democrático, a excepción del liderazgo provisto por Sam (Reynolds)”, observó otro miembro del equipo. En el mundo del desarrollo del software, un equipo democrático es un equipo en donde todos los miembros de este tienen un decir en el proceso de toma de decisiones. “Desafortunadamente, nosotros éramos un equipo de individualistas muy ambiciosos, muy talentosos – si debo referirme a mi mismo – y muy opinadores. Randy (Samuels) era justo el peor del grupo. Lo que quiero decir es que teníamos, por ejemplo, a dos chicos y una chica con grados de maestría de la CMU1 , y no eran tan arrogantes como Randy.”

¹ CMU significa Universidad de Carnegie - Mellon, una líder nacional en enseñanza de Ingeniería del Software.

Un compañero comentó sobre un incidente que Samuels causó en una reunión de aseguramiento de la calidad. Esta reunión involucraba a Samuels y a tres revisores de un módulo de software que Samuels había diseñado e implementado. Tales reuniones son llamadas “revisiones del código”. Uno de los revisores menciona que Samuels había usado un algoritmo sumamente ineficiente (programa) para lograr un determinado resultado y Samuels “se puso todo colorado”. Empezó a gritar una sarta de obscenidades y después se levantó y se fue. Y nunca regresó.

“Le enviamos un memo con un algoritmo más rápido y a su tiempo usó este algoritmo en su módulo”, agregó el colega.

El módulo de software del incidente de la reunión de aseguramiento de la calidad fue el primero en ser identificado como una falla en el “asesino” del operador de robot. No obstante, este colega se apuró a señalar que la eficacia del algoritmo no era un tópico en el malfuncionamiento del robot. Era solo que Randy hacía muy difícil para la gente el poderle comunicar las observaciones. Se tomaba todo muy a pecho. Se graduó con el puntaje más alto de la clase y luego se recibió con honores en Ingeniería del Software en Purdue. Definitivamente es muy inteligente”.

“Randy, en su pared, tiene este inmenso cartel hecho en Banner”, continuó este colega, “Decía: “DENME LA ESPECIFICACIÓN Y LES DARE UN PROGRAMA DE COMPUTACIÓN”. Ese es el tipo de arrogancia que tenía y también demuestra que tenía muy poca paciencia para desarrollar y verificar las especificaciones. Amaba el aspecto de solucionar el problema, la programación propiamente dicha”. No parecía que Randy Samuels quedó atrapado en el espíritu de la “programación sin egolatría”, observó la profesora Skinner cuando escuchó esta parte de la entrevista con los colegas de trabajo de Samuels. “La idea de una programación sin egocentrismo es que el producto de software pertenece al equipo y no a los programadores individuales. La idea es estar abierto a las críticas y estar menos atado al trabajo propio. Ciertamente que la tarea de revisión de código es coherente con esta filosofía general”. Una colega habló acerca de otro aspecto de la personalidad de Samuels: su capacidad de ayuda. “Randy odiaba las reuniones, pero era muy bueno con las relaciones uno a uno. Siempre estaba ansioso por ayudar. Recuerdo una vez que me encontraba encerrada en un camino sin salida y él, en vez de tan solo señalarme la dirección correcta, se hizo cargo del problema y lo resolvió el mismo. Se paso cerca de 5 días completos en mi problema”. “Por supuesto que mirando en retrospectiva, hubiera sido mejor para el pobre Sr. Matthews y su familia que Randy se hubiese dedicado tan solo a sus propias cosas”, agregó luego de una larga pausa.

1.3.4. Artículo 4 - El proyecto del “Robot Asesino” controvertido desde el vamos. Bandos enfrentados por el modo en que debía proseguir el proyecto

Especial para el SENTINEL – OBSERVER de Silicon Vallley. Silicon Valley, EEUU por Mabel Muckraker

Dos grupos, comprometidos con diferentes filosofías de desarrollo de software, casi se enfrentan violentamente durante las reuniones iniciales de planeamiento para el Robbie CX30, el robot de Silicon Techtronics que mató a un obrero de la línea de ensamble el pasado mes de mayo. Estaba en cuestionamiento si el proyecto Robbie CX30 debía proseguir de acuerdo con el “modelo de cascada” o el “modelo de prototipo”.

El modelo de cascada y el prototipo son dos métodos comunes para organizar un proyecto de software. En el modelo de cascada, el proyecto de software pasa a través de etapas definidas de desarrollo. La primera etapa es la de análisis de requerimientos y especificaciones, durante la cual se intenta arribar a un acuerdo en cuanto a la funcionalidad detallada del sistema. A medida que el proyecto pasa de una etapa a la siguiente, existen limitadas oportunidades de dar marcha atrás y cambiar decisiones ya tomadas. Una desventaja es que los usuarios potenciales no tienen oportunidad de interactuar con el sistema hasta bien entrado el ciclo de vida del mismo.

En el modelo de prototipo, se pone un gran énfasis en producir un modelo de prototipo bien temprano durante el ciclo de vida del sistema. El prototipo es construido con el propósito de arribar a una especificación final de la funcionalidad del sistema propuesto. Los usuarios potenciales pueden interactuar con el prototipo en forma temprana y con frecuencia hasta que son acordados los requerimientos. Este enfoque le da los potenciales usuarios la oportunidad de interactuar con un sistema prototipo en forma temprana durante el ciclo de desarrollo y mucho antes que el sistema final esté diseñado y codificado.

En un memorando de fecha 11 de diciembre del anteaño pasado, Jan Anderson, miembro del equipo original del proyecto CX30, ataco duramente la decisión tomada por el gerente del proyecto Sam Reynolds de emplear el modelo en cascada. El SENTINEL – OBSERVER obtuvo una copia del memo de Anderson, dirigido a Reynolds, y Anderson verificó la autenticidad del memorando para este diario. Reynolds despidió a Anderson el 24 de diciembre, justo dos semanas después de que ella escribiera el memo.

El memo de Anderson hace referencia a una reunión anterior en la que ocurrió un fuerte intercambio de opiniones relacionadas con la filosofía del desarrollo del software.

En el memo, Anderson subrayó el siguiente párrafo:

“No fueron mis intenciones impugnar su competencia durante la reunión de ayer, pero debo protestar con mi mayor vehemencia contra la idea de que completemos el software de Robbie CX30 siguiendo el modelo de cascada que Usted ya usó en otros proyectos. No necesito recordarle que aquellos eran proyectos de procesamiento de datos que involucraban el procesamiento de transacciones de negocio. El proyecto Robbie CX30 llevará un alto grado de interacción, tanto entre robot y componentes como entre robot y su operador. Dado que la interacción del operador con el robot es tan importante, la interfaz no puede estar diseñada como una idea de último momento”. Randy Samuels, a quien se lo acuso de asesinato no premeditado por la muerte de Bart Matthews, padre de 3 niños, había participado de la reunión del 11 de diciembre.

En una conservación con este diario, Anderson dijo que Samuels no tenía mucho que decir sobre la controversia cascada – prototipo, pero sí afirmó que daría “una mano” con tal de que exoneraran a Samuels.

“El proyecto fue sentenciado a muerte mucho antes de que Samuels malinterpretara las fórmulas”, aclaró Anderson enfáticamente en la sala de su casa en los suburbios.

En conversación con este diario, Anderson hizo lo mejor de sí para explicar la controversia del método cascada vs. prototipo en términos simples. “El punto principal en realidad era si podíamos llegar a ponernos de acuerdo con los requerimientos del sistema sin dejar que los operadores del robot presintieran lo que teníamos en mente. Reynolds ha estado en el negocio del procesamiento de datos por tres décadas y es bueno en eso, pero nunca debería haber sido gerente de este proyecto”.

Conforme a registros obtenidos por el SENTINEL – OBSERVER, Silicon Techtronics, Michael Waterson. Reynolds reemplazaba a John Cramer, quien gerenciaba al anterior proyecto Robbie CX10 y CX20. Cramer fue puesto a cargo del proyecto CX30, pero murió inesperadamente en un accidente aéreo. Al colocar a Reynolds a cargo del proyecto CX30, nos dice nuestra fuente, que Waterson iba en contra del consejo de Ray Jonson, jefe de la División de robótica. De acuerdo con estas fuentes, Jonson se oponía fuertemente a la alternativa de ponerlo a Reynolds como jefe del proyecto Robbie CX30. Estas fuentes dijeron al SENTINEL – OBSERVER que la elección de Waterson por Reynolds fue puramente una decisión de recorte de gastos. Era más barato transferir a Reynolds a la División robótica que incorporar a un nuevo líder de proyecto de fuera de la corporación.

La fuente anónima que el SENTINEL – OBSERVER llamará “Marta” describió la situación de este modo: “Waterson pensaba que sería mas barato transferir a Reynolds a robótica antes que intentar encontrar afuera un nuevo gerente para el proyecto Robbie. Además, Waterson tendía a sospechar de la gente de

afuera del grupo. Con frecuencia mandaba memos sobre cuánto tarda la gente en aprender “el modo de hacer las cosas de Silicon Techtronics”. Desde el punto de vista de Waterson, Reynolds era el gerente y fue transferido a su nuevo puesto de robótica como un gerente y no como un experto técnico. Claramente, Reynolds se veía a sí mismo tanto gerente como experto técnico. Reynolds no tenía conciencia de sus propias limitaciones técnicas”.

Según Marta, Reynolds era muy renuente a gerenciar proyecto que no usaran el modelo de cascada que tan bien le había servido en el procesamiento de datos. Tildó al modelo prototipo como un “modelo de moda” en la reunión del 11 de diciembre, y después de una serie de intercambios verbales la cosa se puso muy personal.

“Anderson estaba especialmente expresiva”, recuerda Marta. “Tenía mucha experiencia con interfaces con usuarios y desde su perspectiva, la interfaz robot – operador era crítica para el éxito del CX30, dado que la intervención del operador sería frecuente y a veces crítica”. En su entrevista con el SENTINEL – OBSERVER, Jan Anderson comentó sobre el aspecto de la reunión del 11 de diciembre:

“Reynolds estaba en contra de “perder el tiempo” – para usar sus propias palabras – con cualquier tipo de análisis formal de las propiedades de los factores humanos y su interfaz con el usuario. Para él, la interfaz con el usuario real eran un tema periférico”.

“Para él (Reynolds), cualquier cosa nueva era “moda”, agrega Anderson. “Las interfaces de las computadoras eran una moda, el diseño orientado a objetos era una moda, la especificación formal y las técnicas de verificación eran una moda, y por sobre todo, el modelo en prototipo era una moda”. Justo una semana después de la reunión del 11 de diciembre, el grupo de proyecto Robbie recibió un memo de Sam Reynolds concerniente al plan para el proyecto Robbie CX30.

“Era el modelo de cascada, como salido de un libro”, Anderson dijo a este reportero mientras revisaba una copia del memo con el plan del proyecto. “Análisis de requerimiento y especificación, luego diseño de arquitectura y diseño detallado, codificación, prueba, entrega y mantenimiento. En el modo de ver de Reynolds, no hacía falta tener ninguna interacción del usuario con el sistema hasta muy, pero muy avanzado el proyecto”. El SENTINEL – OBSERVER se ha enterado de que el primer operador que realmente uso a Robbie CX30 en una función industrial fue Bart Matthews, el hombre que fue muerto en la tragedia del robot asesino. Este primer uso de Robbie CX30 en un uso industrial fue cubierto por los medios, incluyendo este periódico. Como una gran ironía, el Informe Anual de Silicon Techtronics para los accionistas, publicado el pasado mes de marzo, contiene en la brillante portada una foto de un sonriente Bart Matthews. A Matthews se lo muestra operando al mismísimo robot Robbie CX30 que aplastó hasta tan solo dos meses después de la toma fotográfica.

1.3.5. Artículo 5 - Silicon Techtronics prometió entregar un robot seguro. Cuestionada la calidad del entrenamiento del operador.

Especial para el SENTINEL – OBSERVER de Silicon Vallley. Silicon Valley, EEUU por Mabel Muckraker

En una conferencia de prensa de esta tarde, un grupo de programadores que se autodenominan “Comité de Justicia para Randy Samuels”, distribuyó documentos que muestran que Silicon Techtronics se obligó a sí misma a hacer entrega de robots que “no causarían daño corporal a los operadores humanos”. Randy Samuels es el programador que ha sido acusado del asesinato infame del caso del “robot asesino”.

“No podemos entender como el Fiscal pudo acusar a Randy con esos cargos cuando, de hecho, la compañía Silicon Techtronics estaba legalmente obligada a producir y entregar robots seguros a Cybernetics”, dijo el vocero del comité, Ruth Witherspoon. “Creemos que en todo esto hay un encubrimiento y que hay algún tipo de confabulación entre la gerencia de SiliTech (Silicon Techtronic) y la oficina del Fiscal. Michael Waterson era uno de los mas grandes contribuyentes de la campaña de reelección de la Sra. McMurdock

del año pasado”. Michael Waterson es presidente ejecutivo de Silicon Techtronics. Jane McMurdock es la Fiscal de la ciudad de Silicon Valley. El SENTINEL – OBSERVER confirmó que Waterson hizo varios grandes aportes a la campaña de reelección de McMurdock del otoño pasado.

“A Randy le están haciendo pagar los platos rotos por una empresa que tiene estándares de control de calidad laxos y no lo vamos a permitir! Grito Whitterspoon en una declaración a los periodistas. “Creemos que la política ha entrado en todo esto”.

Los documentos que fueron distribuidos por el comité por la “Justicia para Randy Samuels” eran porciones de lo que se llama un “documento de requerimientos”. Según Ruth Whitterspoon y otros miembros del comité, este documento prueba que Samuels no fue legalmente responsable de la muerte de Bart Matthews, el desafortunado operador de robot que fue muerto por un robot de Silicon Techtronics en Cybernetics, Inc. en Silicon Heights el pasado mes de abril.

El documento de requerimientos es un contrato entre Silicon Techtronics y Cybernetics, Inc. especifica con total detalle la funcionalidad del robot Robbie CX30 que Silicon Techtronics prometió entregar a Cybernetics. Según Whitterspoon, el robot Robbie CX30 fue diseñado para ser un robot “inteligente” que pudiera ser capaz de operarse en una variedad de funciones industriales. Cada cliente de la corporación necesitó de los requerimientos separados ya que Robbie CX30 no era un “robot de llave en mano” sino un robot que necesitaba ser programado de forma diferente para cada aplicación.

No obstante, todos los documentos de requerimientos que fueron acordados bajo los auspicios del proyecto Robbie CX30, incluyendo al acuerdo entre Silicon Techtronics y Cybernetics, contiene los siguientes fundamentos de importancia:

- “El robot será de operación segura y aún bajo circunstancias excepcionales (ver Sección 5.2), el robot no causará daño corporal alguno al operador humano”.
- “En el caso de condiciones excepcionales que potencialmente contengan el riesgo de daño corporal (ver Sección 5.2.4 y todas las subsecciones), el operador humano podrá ingresar una secuencia de códigos de comando, según se describe en las secciones relevantes de las especificaciones funcionales (ver Sección 3.5.2.), que detendrá el movimiento del robot mucho antes que pueda ocurrir un daño corporal”.
- Las “condiciones excepcionales” incluyen eventos inusuales tales como datos extraños desde los sensores del robot, movimiento errático o violento del robot o error del operador. Fue justamente esa condición excepcional la que llevó a la muerte a Bart Matthews.

Estos párrafos fueron extractados de las porciones del documento de requerimientos que trata sobre los “requerimientos no funcionales”. Los requerimientos no funcionales listan en detalle las restricciones bajo las cuales operaría el robot. Por ejemplo, el requerimiento de que el robot seria incapaz de dañar a su operador humano es una restricción y Silicon Techtronics, según Ruth Whitterspoon, estaba legalmente obligada a satisfacer este punto. La parte de los requerimientos funcionales del documento de requerimientos cubre (nuevamente en sumo detalle) el comportamiento del robot y su interacción con el entorno y su operador humano. En particular, los requerimientos funcionales especificaban el comportamiento del robot bajo cada una de las condiciones excepcionales esperadas. En su declaración a los periodistas en la conferencia de prensa, Whitterspoon explicó que Bart Matthews fue muerto cuando se produjo la condición excepcional 5.2.4.26. Esta involucra un movimiento del brazo del robot extremadamente violento e impredecible. Esta condición requiere de la intervención del operador, a saber el ingreso de los códigos de comandos mencionados en los documentos, pero aparentemente Bart Matthews se confundió y no pudo ingresar con éxitos los códigos.

“Si bien el programa de Randy Samuels estaba mal – él en verdad malinterpreto las formulas de la dinámica del robot, como se informó en los medios. La condición excepcional 5.2.4.26 estaba diseñada para proteger justamente este tipo de contingencia”, dijo Whitterspoon a los periodistas. “Los valores del movimiento del

robot generados por el programa de Randy identificaron correctamente a esta condición excepcional y el operador del robot recibió el debido aviso de que algo andaba mal”.

Whiterspoon dijo que poseía una declaración formada de otro operador de robot de Cybernetics en donde afirmaba que las sesiones de entrenamientos ofrecidas por Silicon Techtronics nunca mencionaron a ésta ni a tantas otras condiciones excepcionales. Según Whiterspoon, el operador del robot ha jurado que ni a él ni a ningún otro operador de robots les fue dicho que jamás el brazo del robot podía oscilar violentamente.

Whiterspoon citó esta declaración en la conferencia de prensa. “Ni yo ni Bart recibimos entrenamiento para manejar este tipo de condición excepcional. Dudo mucho que Bart Matthews tuviese idea de que se suponía que debía hacer cuando la pantalla de la computadora comenzó a mostrar mensaje de error”.

Las condiciones excepcionales que requieren de la intervención del operador causan un mensaje de error que se genera en la consola del operador. La Policía de Silicon Valley afirmó que cuando Bart Matthews fue muerto, el manual de referencia en su consola estaba abierto en la página del índice que contenía los códigos de ingreso para “errores”.

Whiterspoon luego citó, secciones del documento de requerimientos que obligan a Silicon Techtronics (el vendedor) a entrenar adecuadamente a sus operadores:

- “El vendedor suministrará cuarenta (40) horas de entrenamiento para los operadores. Este entrenamiento cubrirá todos los aspectos de la operación del robot, incluyendo una cobertura exhaustiva que contenga potencialmente el riesgo de daño corporal. El vendedor proveerá y administrará instrumentos de prueba apropiados que serán usados para certificar el suficiente entendimiento del operador de las operaciones de la consola del robot y de los procedimientos de seguridad. Solo los empleados del cliente que hayan pasado estas pruebas estarán habilitados para operar el robot Robbie CX30 en una verdadera función industrial.
- “El manual de referencia deberá suministrar instrucciones claras para la intervención del operador en todas las situaciones excepcionales, especialmente e inclusive aquellas que contengan potencialmente el riesgo de daño corporal”.

Según Whiterspoon, las declaraciones juradas de varios operadores de robots de Cybernetics Inc., aseguran que solo se destino un día laborable (aproximadamente 8 horas) al entrenamiento de operadores. Es mas, casi no dedicó tiempo alguno al tratamiento de condiciones excepcionalmente peligrosas.

“La prueba escrita desarrollada por Silicon Techtronics para habilitar a un operador están considerada por los empleados de Cybernetics como un chiste”, aseguro Whiterspoon, “Obviamente Silicon Techtronics no le dedicó mucho tiempo al entrenamiento y procedimientos de pruebas obligatorios según el documento de requerimientos según la evidencia en nuestro poder”.

Reimpreso con el permiso de ROBOTIC WORD el diario de ROBOTICS AND ROBOTICS APPLICATIONS.

1.3.6. Artículo 6 - La interfaz del “Robot Asesino”

*Dr. Horace Gritty, Departamento de Ciencias de la Computación y materias relacionadas.
Universidad de Silicon Valley, Silicon Valley, EEUU*

Resumen: El robot industrial Robbie CX30 se suponía que debía restablecer un nuevo modelo de inteligencia de robots industriales. Desgraciadamente, uno de los primeros robots Robbie CX30 mató a un obrero de la línea de montajes, y esto llevó a la acusación de uno de los que desarrollaron el robot, Randy Samuels. Este paper promueve la teoría de que quien debería estar en juicio es el desarrollador de la

interfaz robot – operador. El robot Robbie CX30 viola casi todas las reglas del diseño de interfaz. Este paper se centra en como la interfaz del Robbie CX30 violó cada una de las “Ocho reglas de oro” de Shneiderman.

1. Introducción

El 17 de mayo de 1992, un robot industrial Robbie CX30 de Silicon Techtronics mató a su operador, Bart Matthews, en Cybernetics Inc., en Silicon Heights, un suburbio de Silicon Valley. La investigación de los hechos del accidente guiaron a las autoridades a la conclusión de que un módulo de software, escrito y desarrollado por Randy Samuels, un programador de Silicon Techtronics, fue el responsable del comportamiento errático y violento que a su vez llevo a la muerte por decapitación a Bart Matthews

NOTA: Los medios de prensa manejaron la información haciendo creer que Bart Matthews había sido aplastado por el robot, pero la evidencia fotográfica dada a este autor muestra otra cosa. Tal vez, las autoridades trataban de proteger la sensibilidad pública.

Como experto en el área de interfases con el usuario (1, 2, 3), se me pidió prestar ayuda a la policía en la reconstrucción del accidente. Para poder hacer esto, se le pidió a Silicon Techtronics que me suministrara un simulador de Robbie CX30 que incluyera por completo la consola del operador del robot. Esto me permitió investigar el comportamiento del robot sin tener que en realidad arriesgarme seriamente. Debido a mi profundo entendimiento de interfaces con el usuario y factores humanos pude reconstruir el accidente con extrema precisión. Sobre la base de esta reconstrucción, llegué a la conclusión de que fue el diseño de la interfaz y no el imperfecto diseño del software lo que debería haber sido visto como el criminal en este caso.

A pesar de mi descubrimiento, la fiscal Jane McMurdock insistió en proseguir el caso en contra Randy Samuels. Pienso que cualquier computador científico competente, dado una oportunidad de interactuar con el simulador del Robbie CX30, también habría concluido que el diseñador de la interfaz y no el programador debería haber sido acusado por negligencia, si no por homicidio no premeditado.

2. “Las ocho reglas de oro” de Schneiderman

Mi evaluación de la interfaz con el usuario del Robbie CX30 está basada en las “ochos reglas de oro” de Scheiderman (4), también utilicé otras técnicas para evaluar la interfaz, pero estas serán publicadas en papers separados. En esta sección ofrezco una breve revisión de las ocho reglas de oro de Scheiderman, un tema que resultará más familiar para los expertos en interfaces de computación como yo y no hackers de robótica que leyeron este oscuro periódico.

Las ocho reglas de oro son:

1. Buscar siempre la coherencia. Tal como se puede ver mas abajo, es importante que una interfaz con el usuario sea coherente a muchos niveles. Por ejemplo, los diseños de pantalla deben ser coherentes de una pantalla a la siguiente. En un entorno en que se usa una interfaz grafica con el usuario (GUI), esto también implicará concordancia de un utilitario al siguiente.
2. Permitirle a los usuarios frecuentes el uso de shortcuts. Los usuarios frecuentes (o, ‘power users’) pueden desalentarse ante tediosos procedimientos. Permitirles a estos usuarios un procedimiento menos tedioso para completar una tarea dada.
3. Dar realimentación de información. Los usuarios necesitan ver las consecuencias de sus acciones. Si un usuario ingresa un comando pero la computadora no muestra ya lo que sea que esté procesando o lo que ha procesado, esto puede dejar al usuario confundido o desorientado.
4. Diseñar diálogos que tengan un fin. Interactuar con una computadora es algo así como dialogar o conversar. Cada tarea debe tener un inicio, un desarrollo y un fin. Es importante que el usuario sepa cuando una tarea esta finalizada. El usuario necesita tener la sensación de que la tarea ha alcanzado su término.

5. Permitir manejos simples de errores. Los errores del usuario deben estar diseñados dentro del sistema. Otro modo de decirlo es que no debe haber ninguna acción por parte del usuario que sea considerada como un error que está mas allá de la capacidad del sistema para manejarlo. Si el usuario comete un error, debe recibir información útil, clara y concisa sobre la naturaleza de tal error. Debe resultar fácil para el usuario deshacer este error.
6. Permitir deshacer las acciones con facilidad. Mas genéricamente, los usuarios deben poder deshacer lo que han hecho, sea esto o no de la naturaleza errónea.
7. Respalidar que el centro del control este internamente. La satisfacción del usuario es alta cuando el usuario tiene la sensación de que es él o ella quien tiene el control y la satisfacción del usuario es baja cuando el usuario siente que la computadora tiene el control. Diseñar interfaces para reforzar la sensación de que es en el usuario donde yace el control del ámbito de la interacción humano – computadora.
8. Reducir la carga de la memoria inmediata. La memoria inmediata del ser humano es notablemente limitada. Los psicólogos regularmente citan la ley de Miller que dice que la memoria inmediata está limitada a siete piezas discretas de información. Hacer todo lo posible para liberar la carga en la memoria del usuario. Por ejemplo, en lugar de pedirle al usuario que tipee el nombre de un archivo para abrirlo, presentar al usuario una lista de los archivos disponibles en ese momento.

3. Revisión de la consola del robot

La interfaz del operador Robbie CX30 violó casi todas y cada una de las reglas de Shneiderman. Muchas de estas violaciones fueron directamente responsables del accidente que terminó con la muerte del operador del robot.

La consola del robot era una IBM PS/2 modelo 55SX con un procesador 80386 y un monitor EGA color con una resolución de 640 X 480. La consola tenía un teclado, pero no mouse. La consola está empotrada en una estación de trabajo que tenía, además, estantes para manuales y un área para tomar notas y para leer manuales. No obstante, el área para escribir/ leer estaba a bastante distancia de la pantalla de la computadora, o sea que era bastante incómodo y cansador para el operador manejar cualquier tarea que requiera de mirar algo en el manual y luego actuar rápidamente con respecto a la posición de la consola en el área de escribir/ leer. Esto resentía mucho la espalda del operador y también causaba excesivos cansancios a la vista.

No puedo comprender como un sistema sofisticado como éste no pudo incluir un aparato de mejor diseño para los ingresos de datos. Uno solo podría concluir que Silicon Techtronics no tenía mucha experiencia en tecnología de interfaces con el usuario. El documento de requerimientos (5) especificaba un sistema manejado por menús, lo cual era una elección razonable. Sin embargo, en un utilitario en donde lo esencial era la rapidez, especialmente cuando la seguridad del operador esta en juego, el uso de un teclado para todas las tareas de selección de opción de menús fue una elección de extremada pobreza, que requería de mucho uso del teclado para lograr el mismo efecto que podía haberse conseguido casi instantáneamente mediante un mouse. (Ver el paper escrito por Foley et al. (6) En realidad, se me ocurrieron todas estas ideas antes que Foley las publicara, pero él me ganó).

El operador del robot podía interactuar con el robot y de este modo producir un impacto sobre su comportamiento al hacer las opciones en un sistema de menús. El menú principal consistía de 20 ítems, demasiados en mi opinión, y cada ítem del menú principal tenía un submenú tipo desplegable asociado a este. Algunos de los submenús tenían tanto como veinte ítems – nuevamente demasiados. Es más, parecía haber poca lógica en cuanto a por qué los ítems de los menús estaban listados en el orden en que estaban. Hubiese sido mucho mejor una organización alfabética o funcional.

Alguno de los ítems en los menús desplegables tenían hasta cuatro menús pop up relacionados a estos. Estos aparecían en secuencias a medida que se hacía una selección correspondiente en los submenús. Ocasionalmente, una elección de un submenú abriría un cuadro de dialogo en la pantalla. Un cuadro de

dialogo requiere de cierto tipo de interacción entre el operador y el sistema, por ejemplo resolver ciertos temas, como ser, ingresar cuál es el diámetro de un dispositivo dado a ser bajado en el baño de ácido.

El sistema de menús presenta una estricta jerarquía de elección de menús. El operador podría volver hacia atrás en esta jerarquía apretando la tecla de escape. Esta tecla escape también podría terminar los diálogos.

El uso del color en la interfaz fue muy poco profesional, había demasiados colores en un espacio demasiado chico. Los contrastes eran muy fuertes y el resultado, para este revisor, resultó en una severa fatiga ocular en tan solo quince minutos de usos. Hubo uso excesivo de efectos musicales tontos y falsees cuando se ingresaba opciones o códigos erróneos.

Uno debería preguntarse por qué Silicon Techtronics no intentó un enfoque mas sofisticado para el diseño de interfaz. Luego de un cuidadoso estudio del dominio de los utilitarios del Robbie CX30, he llegado a la conclusión de que una interfaz de manipulación directa, que mostrara literalmente al robot en la consola del operador, habría sido lo ideal. El entorno tan visual dentro del cual operaba el robot se prestaba naturalmente al diseño de metáforas de pantallas apropiadas para ese entorno, metáforas que el operador podría entender con facilidad. Esto permitiría que el operador manipulara el robot mediante el manejo, en la consola del robot, de la representación grafica del robot en su entorno. He solicitado a unos de mis estudiantes en el doctorado, Susan Farnsworth, que investigará un poco más esta posibilidad.

4. En qué modo la interfaz del robot Robbie CX30 violó las ocho reglas de oro

La interfaz con el usuario de Robbie CX30 violó todas las reglas de oro en diferentes modos. En este paper sólo trataré unas pocas instancias de violaciones de reglas, dejando la discusión más detallada para futuros artículos y mi próximo libro.

NOTA.: CODEPENDENCIA, Cómo los Usuarios de Computadoras permiten deficientes Interfaces con el Usuario, Angst Press, Nueva York. Este libro presenta una teoría radicalmente nueva con respecto a la relación entre la persona y la máquina. Esencialmente, algunas personas necesitan una interfaz de mala calidad a los fines de evitar ciertos problemas psicológicos no resueltos en sus vidas.

Lo que haré es destacar esas violaciones que fueron relevantes en este accidente en particular:

- **Buscar siempre la coherencia:** En la interfaz del usuario de Robbie CX30 hubieron muchas incoherencias. Los mensajes de error podían aparecer en casi cualquier color y podían estar acompañados por casi cualquier tipo de efecto musical. Además, los mensajes de error podían aparecer en casi cualquier lugar de la pantalla. Cuando Bart Matthews vio el mensaje de error de la condición excepcional que ocurrió luego, la cual requería la intervención del operador, es probable que fuera esa la primera vez que veía ese mensaje en especial. Además, el mensaje de error apareció en un cuadro verde, sin ningún efecto de sonido. Este es el único mensaje de error de todo el sistema que aparece en verde y sin ningún tipo de acompañamiento de orquesta.
- **Permitir que los usuarios frecuentes utilicen shortcuts:** Este principio no aparece de ningún modo en todo el diseño de la interfaz. Por ejemplo, hubiera sido una buena idea permitir que los usuarios frecuentes pudieran ingresar la primera letra de la opción de un submenú o menú en lugar de requerírseles el uso de las teclas del cursor y luego la tecla "enter" para elegir esa opción determinada. El mecanismo de selección de menús de este mismo sistema debe haber provocado al operador bastante fatiga mental. Es más, debería haberse permitido algún tipo de sistema de tipeo anticipado que permitiera al usuario frecuente ingresar una secuencia de opciones de menú sin tener que esperar a que apareciera realmente el menú en pantalla.
- **Ofrecer realimentación de información:** En muchos casos el usuario no tiene idea de si el comando que acaba de ingresar se está procesando o no. Este problema se exagera además por las inconsistencias en el diseño de la interfaz con el usuario. En algunos casos al operador se le da una realimentación detallada de lo que el robot está ejecutando. En otros, el sistema permanece misteriosamente silencioso. En general, al usuario se lo lleva a que espere algún tipo de realimentación y por

consiguiente se queda confundido cuando está no se le da. En la pantalla, no hay una representación visual del robot y su entorno, y la visión que tiene el operador del robot a veces está obstruida.

- Diseñar diálogos que tengan fin: Hay muchos casos en los que una secuencia dada de tecleado representa una idea holística, una tarea completa, pero al operador se lo deja sin el tipo de realimentación que le confirmare que la tarea ha sido en efecto completada. Por ejemplo, hay un dialogo bastante complicado que se necesita cuando se quiere sacar un elemento de un baño de ácido. Sin embargo, luego de completar este dialogo, el usuario es llevado a otro dialogo nuevo, y no relacionado con este, sin que se les informe que el dialogo anterior ha finalizado.
- Ofrecer manejo simple de los errores: El sistema pareciera estar diseñado para que el usuario se lamentara por cualquier ingreso erróneo. No sólo el sistema permite numerosas oportunidades para el error, sino que cuando un error en realidad ocurre, es probable que se repita por algún tiempo. Ello se debe a que la interfaz con el usuario, fue diseñada en forma tal que corregir el error era una odisea tediosa, frustrante y a veces enfurecedora. Algunos de los mensajes de error eran directamente ofensivos y condescendientes.
- Permite deshacer las acciones con facilidad: Como se menciona en el párrafo anterior, la interfaz con el usuario hace muy difícil la tarea de deshacer entradas erróneas. En general, el sistema de menús permite deshacer fácilmente las acciones, pero esta filosofía no alcanza para el diseño de los cuadros de diálogos y al manejo de condiciones excepcionales. Desde un punto de vista práctico (opuesto al teórico), la mayoría de las acciones son irreversibles cuando el sistema está en un estado de condición excepcional, y esto ayudó a llegar a la tragedia del robot asesino.
- Promover que uno sea el centro del robot: Muchas de las deficiencias tratadas en los párrafos precedentes disminuyeron la sensación de “tener el control”. Por ejemplo, no recibir información, no poder concluir con las interacciones, no permitir deshacer con facilidad las acciones en el momento que surgen las excepciones, todas estas cosas actúan para disminuir la sensación de que el usuario posee el control sobre el robot. Hubieron muchas características de esta interfaz que hicieron que el operador sintiera que hay un enorme bache entre la consola del operador y el robot en sí, mientras que un buen diseño de interfaz hubiera hecho transparente la interfaz con el usuario y le hubiere dado al operador del robot la sensación de estar en contacto directo con el mismo. En un caso, le ordené al robot mover un elemento desde el baño de ácido hasta la cámara de secado y pasaron 20 segundos antes de que el robot pareció responder. De este modo, no tuve la sensación de estar controlando al robot. Tanto la repuesta demorada del robot como la falta de realimentación en la pantalla de computadora, me hicieron sentir que el robot era un agente autónomo, la verdad un sentimiento como mínimo perturbador.
- Reducir la carga de memoria de corto plazo: Un sistema que se maneja por medio de menús es generalmente bueno en términos de carga de memoria que crea a los usuarios. No obstante, hay gran variación entre implementaciones particulares de sistemas de menú en lo que hace a carga de memorias. La interfaz con el usuario de Robbie CX30 tenía menús muy grandes sin ninguna organización interna. Esto crea una gran carga al operador en términos de memoria y también en términos de tiempo de búsqueda, el tiempo que lleva al operador ubicar una opción determinada del menú.
- Muchas pantallas de dialogo requerían que el usuario ingresara con el teclado números de partes, nombre de archivos, y otra información. El sistema podría haberse diseñado fácilmente de forma de mostrarle al usuario estos números de partes, etc., sin la necesidad que el usuario recordara estas cosas de su propia memoria. Esto incrementaba la carga sobre la memoria del usuario.
- Para finalizar, y esto es realmente imperdonable, increíble como pueda parecer ¡no había ninguna instalación de ayuda en línea o sensible al contexto!. Si bien he ido a los cursos de entrenamientos ofrecidos por Silicon Techtronics, muchas veces me encontré navegando por los manuales de referencia para poder encontrar la respuesta aún a las más básicas preguntas, tales como: “¿Qué significa esta opción de menú? ¿Qué pasa si selecciono esta opción?”.

5. Una reconstrucción de “la tragedia del robot asesino”.

Las fotos policiales de la escena del accidente no son nada agradables de ver. La consola del operador estaba salpicada con bastante cantidad de sangre. No obstante, la calidad de las fotos es excepcional y

utilizando técnicas de ampliación pude descubrir los siguientes factores de importancia sobre el momento en que fue decapitado Bart Matthews:

- La luz NUM LOCK estaba encendida: El teclado de IBM contiene un tablero que se puede operar de dos modos. Cuando la luz de NUM LOCK esta encendida, esa parte se comporta como una calculadora. Del otro modo, las teclas pueden usarse para mover el cursor de la pantalla.
- Había sangre esparcida en el tablero numérico: Las huellas ensangrentadas indican que Bart Matthews estaba usando el tablero numérico en el momento en que fue golpeado y muerto.
- Se encontraba titilando en verde un mensaje de error: Esto nos dice que la situación de error vigente en el momento que ocurrió la tragedia. El mensaje de error decía “ROBOT DYNAMICS INTEGRITY ERROR – 45”.
- Había un manual de referencia apoyado y abierto sobre el área de lectura/ escritura de la estación de trabajo.
- Uno de los cuatros volúmenes del manual de referencia estaba abierto en la página del índice que contenía el ítem “ERRORES/ MENSAJES”.
- En la pantalla también había un mensaje que mostraba instrucciones al operador
- El mensaje aparecía en amarillo en la parte inferior de la pantalla. En el mensaje se leía: “POR FAVOR INGRESE INMEDIATAMENTE LA SECUENCIA DE COMANDOS PARA CANCELAR EL ERROR DINAMICO DEL ROBOT!!!”

En base a las evidencias físicas mas otras evidencias contenidas en los registros del sistema, y basándose en la naturaleza del error que ocurrió (error de integridad de dinámica del robot – 45, el error que estuvo causado por el programa de Randy Samuels), he llegado a la conclusión de que ocurrió la siguiente secuencia de eventos en la fatal mañana de la tragedia del robot asesino:

- 10:22:30 “ERROR DE INTEGRIDAD DE DINAMICA DEL ROBOT –45” aparece en la pantalla, Bart Matthews no lo ve porque no hay efecto de audio o señal sonora tal como ocurre en todas las situaciones de error. Además, el mensaje de error aparece en verde, lo que en todos los contextos significa que hay proceso completándose con normalidad.
- 10:24:00 El robot comienza a moverse lo suficientemente violento como para que Bart Samuels lo note.
- 10:24:05 Bart Matthews se da cuenta del mensaje de error, no sabe lo que significa. No sabe que hacer. Intenta con el submenú “cancelación de emergencia”, un submenú de uso genérico para apagar el robot. Este involucra SEIS opciones de menú por separado, pero el Sr. Matthews no se da cuenta de que la luz del NUM LOCK esta encendida. Por ende, las opciones del menú no estaban ingresando, dado que las teclas del cursor operaban como teclas de calculadora.
- 10:24:45 El robot gira el baño de ácido y comienza a arrastrar la consola del operador, con sus brazos dentados batiéndose con gran amplitud. Nadie espera que un operador tuviera que huir de un robot descontrolado, así que Bart Matthews queda atrapado en su área de trabajo por el robot que avanza. Mas o menos para este momento es que Bart Matthews saca el manual de referencia y empieza a buscar el error ERROR DE INTEGRIDAD DE DINAMICA DEL ROBOT –45 en el índice. Ubica con éxito la referencia a mensajes de error en el índice.
- 10:25:00 El robot ingresa al área del operador. Bart Matthews abandona la búsqueda del procedimiento del operador ante un error de integridad dinámica del robot. En su lugar, intenta una vez mas ingresar la secuencia de “cancelación de emergencia” desde el teclado numérico, momento en que es golpeado.

6. Resumen y conclusiones

Si bien el módulo de software escrito por Randy Samuels causó en verdad que el robot Robbie CX30 oscilara fuera de control y atacara a su operador humano, un buen diseño de la interfaz hubiera permitido al operador terminar con el comportamiento errático del robot. En base al análisis de la interfaz del usuario

del robot llevado a cabo utilizando las ocho reglas de oro de Schneiderman, el experto en diseño de interfaces ha arribado a la conclusión de que el diseñador de la interfaz y no el programador fue la parte mas culpable en este desafortunado evento.

7. Referencias

- Gritty, Horace (1990). The only user interface book you'll ever need. Vanity Press, Oshkosk, WI 212 pag. [El único libro sobre interfaz de usuarios que Usted necesitara].
- Gritty, Horace (1992). What we cant learn from the killer robot [Lo que podemos aprender de un robot asesino], charla dada en el Simposio internacional de la Universidad de Silicon Valley sobre Seguridad en robot e Interfaces de usuario, Marzo de 1991. también por aparecer en las Notas de los alumnos de la Universidad de Silicon Valley].
- Gritty, Horace (se espera para 1993). CODEPENDENDY: How computer users enable poor user interfaces, Angst press, New York [Como los usuarios de computadoras permiten interfaces deficientes].
- Shneiderman, Ben (1987), Designing the user interface, Addison Wesley, reading MA, 448 pag [Diseño de Interfaces].
- DOCUMENTO DE REQUERIMIENTOS DEL ROBOT INDUSTRIAL INTELIGENTE Robbie CX30: versión de Cybernetics INCS., Documento Técnico N° 91-0023XA, Silicon Techtronics Corporation Silicon Valley, USA 1245 pag.
- Foley, J. P., Wallace, V. L., y Chan, P. (1984): The human factors of computer graphics interaction techniques [Los factores humanos de las técnicas de interacción de graficas de computación] IEEE COMPUTER GRAPHICS AND APPLICATIONS, 4(11) pag, 13-48.

1.3.7. Artículo 7 - Ingeniero de software cuestiona la autenticidad de las pruebas de software del “Robot Asesino”. La indagación de un profesor de la Universidad de Silicon Valley provoca serios cuestionamientos legales y éticos.

Especial para el SENTINEL – OBSERVER de Silicon Vallley. Silicon Valley, EEUU por Mabel Muckraker

El caso del “robot asesino dio un giro significativo ayer cuando un profesor de la Universidad de Silicon Valley presentó un informe que cuestiona la autenticidad de las pruebas que fueron hechas por Silicon Techtronics al software del “robot asesino”. El profesor Wesley Silber, profesor de Ingeniería del Software, dijo en una conferencia de prensa realizada en la universidad que los resultados de las pruebas reflejados en los documentos internos de Silicon Techtronics no concordaban con los resultados de las pruebas obtenidos cuando él y sus colegas ensayaron el software real del robot.

Silicon Valley aún está reaccionando por el anuncio del Profesor Silber, que podría jugar un papel importante en el juicio a Randy Samuels, el programador de Silicon Techtronics que fue acusado por homicidio no premeditado en el ahora infame incidente del “robot asesino”. Presionada por su reacción por el informe del profesor Silber, la fiscal Jane McMurdock reiteró su confianza en que el jurado encontrara culpable a Ray Samuels. Sin embargo, la Fiscal Jane McMurdock impresionó a los periodistas cuando agregó “pero, esto en verdad promueve la posibilidad de nuevas acusaciones”.

Ruth Whirterspoon, la vocero del “Comité de justicia para Randy Samuels”, también estuvo exultante cuando habló a este periódico. “McMurdock no puede tener ambas cosas”. O el programador es el responsable por esta tragedia o se deberá hacer responsable a la gerencia por ello. Creemos que el informe del Silber exonera a nuestro amigo y colega Randy Samuels”.

El gerente Ejecutivo de Silicon Techtronics Michael Waterson hizo la siguiente tibia declaración sobre el informe de Silber:

- “Tan pronto se anunció la acusación de Randy Samuels personalmente le pedí a un estimado ingeniero del software, el Dr. Wesley Silber, que llevara a cabo una indagación objetiva sobre los procedimientos

de aseguramiento de la calidad en Silicon Techtronics. Como gerente ejecutivo de este proyecto, siempre he insistido en que la calidad es lo primero, a pesar de lo que hayan podido leer en los periódicos”.

- “Le pedí al profesor Silber que condujera una investigación objetiva de todos los aspectos de aseguramiento de la calidad de Silicon Techtronics. Prometí al profesor Silber que tendría acceso a toda la información relevante a esta infortunada situación. Le dije en una reunión frente a frente, en mi oficina, que debía proseguir la investigación hasta su final sin importar a donde terminara, sin importar las implicancias”.
- “Basándome en la información que yo recibía de mis gerentes, nunca se me hubiera ocurrido que pudiesen existir problemas de que los procedimientos de aseguramiento de la calidad fueran, ya sea débiles, o estuviesen alterados. Quiero asegurarle al público que la o las personas responsables de esta falta de aseguramiento de la calidad del software dentro de la División de Robótica de Silicon Techtronics serán exhortados a encontrar trabajo en otro lado”.

Roberta Matthews, viuda de Bart Matthews, el operador del robot que fue muerto en el incidente, habló telefónicamente desde su casa con el SENTINEL – OBSERVER. “Aún quiero ver al Sr. Samuels condenado por lo que le hizo a mi marido. No entiendo de dónde viene toda la conmoción. EL hombre que asesinó a mi esposo, debería haber probado su propio software!”.

El SENTINEL – OBSERVER entrevistó al profesor Silber justo antes de su conferencia de prensa. En las paredes de su oficina estaban colgados numerosos premios recibidos a raíz de su trabajo en el campo de Ingeniería del Software y aseguramiento de la calidad del software. Comenzamos la entrevista pidiendo al profesor Silber que explicara por qué a veces el software no es confiable. Contestó a nuestra pregunta citando la enorme complejidad del software.

“Los grandes programas de computadora son indiscutiblemente los artefactos más complejos creados por la mente humana”, explico el profesor Silber sentado frente a un monitor de grandes dimensiones. “En algún momento un programa de computación está en uno de los tantos estados posibles, y hay imposibilidad práctica de asegurar que el programa se comportará como corresponde en cada uno de esos estados. No tenemos el tiempo suficiente para hacer tal tipo de prueba exhaustiva. De modo que usamos estrategias de prueba o heurísticas que muy probablemente encontrarán los errores o bugs, si es que existe alguno”.

El profesor Silber ha publicado numerosos papers sobre Ingeniería del Software. Estuvo en la primera plana cuando el año pasado publicó su lista de “Aerolíneas a evitar si su vida dependiera de ello”. En esa lista se enumeraban las aerolíneas de cabotaje que él consideraba irresponsables por su compra de aviones que están controladas casi por completo por software de computación.

Poco tiempo después de los cargos contra Randy Samuels en el caso del “robot asesino”, el gerente ejecutivo de Silicon Techtronics, Michael Waterson, pidió al profesor Silber que condujera una revisión objetiva de los procedimientos de aseguramiento de la calidad de Silicon Techtronics. La intención de Waterson era contrarrestar la mala publicidad de su empresa luego de las acusaciones de Samuels.

“El aseguramiento de la calidad” se refiere a aquellos métodos que usa un especialista de desarrollo de software para asegurar que el software es confiable, correcto y robusto. Estos métodos se aplican a todo lo largo del ciclo de vida de desarrollo del producto de software. En cada etapa se aplican los métodos de aseguramiento de calidad adecuados. Por ejemplo, cuando un programador escribe código, una medida de aseguramiento de la calidad es probar el código confrontándolo en verdad con los datos de prueba. Otro método sería correr programas especiales, llamados analizadores estáticos, confrontándolos con el nuevo código. Un analizador estático es un programa que busca patrones sospechosos en los programas, patrones que podrían indicar errores o bug.

Estas dos formas de aseguramiento de la calidad son denominadas pruebas dinámicas y pruebas estáticas, respectivamente.

El software consiste de componentes discretos o unidades que eventualmente se combinan para crear un sistema mas grande. Las unidades mismas deben ser probadas, y este proceso de prueba individual de las unidades es llamado prueba unitaria. Cuando las unidades se combinan, se deben probar los subsistemas integrados y este proceso se llama prueba de integración.

El profesor Silber comentó al SENTINEL – OBSERVER sobre su trabajo en Silicon Techtronics: “Mike (Waterson) me dijo de ir allí (a la compañía) y conducir una revisión de procedimientos de pruebas de software y de hacer públicos mis hallazgos. Mike parecía confiado, tal vez debido a lo que le habían dicho sus gerentes, en el sentido de que no encontraría nada malo en los procedimientos de aseguramiento de calidad de Silicon Techtronics”.

Luego de arribar a Silicon Techtronics, el profesor Silber centró su atención en los procedimientos para ensayo dinámico de software en la compañía.

Ayudado por un grupo de graduados, el profesor Silber describió una discrepancia entre el comportamiento real de la sección del código del programa (escrito por Randy Samuels) que causó que el robot Robbie CX30 matara a su operador, y el comportamiento según se lo registro en la documentación de pruebas de Silicon Techtronics. Este descubrimiento en realidad fue hecho por Sandra Henderson, una estudiante graduada en Ingeniería del Software que está contemplando su doctorado con el profesor Silber. Entrevistamos a la Sra. Henderson en uno de sus laboratorios de computación para egresados en la Universidad de Silicon Valley. “Encontramos un problema en la prueba de unidad”, explicó la Sra. Henderson, “Acá están los resultados de la prueba que nos dio el Sr. Waterson en Silicon Techtronics, que se suponen están hechos para código C (lenguaje de programación) que Randy Samuels escribió y que causó el incidente del robot asesino. Como puede ver, todo está claramente documentado y organizado. Hay dos juegos de pruebas: Uno basado en una prueba de caja blanca y otro en una prueba de caja negra. Basándonos en nuestros propios estándares para probar software, estos juegos de prueba están bien diseñados, completos y rigurosos. “La prueba de caja negra implica ver la unidad de software (o sus componentes) como una caja negra que tiene comportamientos predecibles de input y output. Si en el juego de pruebas el componente demuestra los comportamientos esperados para los inputs, entonces pasa la prueba. Los juegos de prueba están diseñados para cubrir todos los comportamientos “interesantes” que una unidad podría mostrar pero sin tener conocimiento alguno sobre la estructura o naturaleza del código en realidad. La prueba de caja blanca implica cubrir todos los pasos posibles a través de la unidad. Así, la prueba de caja blanca se hace con vasto conocimiento de la estructura de la unidad. En la prueba de caja blanca, el juego debe causar que cada sentencia del programa se ejecute por lo menos una vez de modo que ninguna quede sin ser ejecutada.

Sandra Henderson prosiguió explicando el significado de la prueba del software. “ Ni la prueba de caja blanca ni de caja negra “prueban” que un programa esté correcto. No obstante, los probadores de software, tales como se emplean en Silicon Techtronics, pueden volverse bastantes expertos en el diseño de los casos de prueba para descubrir nuevos bugs en el software. La actitud apropiada es que una prueba es exitosa cuando se encuentra un bug”. Básicamente, al probador le dan un juego de especificaciones y hace lo mejor de sí para demostrar que el código que esta probando no satisface sus especificaciones”, explicó la Sra. Henderson.

La Sra. Henderson luego mostró a este reportero los resultados de las prueba que ella en verdad obtuvo cuando corrió el código critico del “robot asesino” usando los juegos de prueba de la compañía, tanto para ensayo de caja blanca como de caja negra. En muchos casos, los resultados registrados en los documentos de prueba de la compañía no fueron los mismos que los generados por el verdadero código del robot asesino.

Durante su entrevista de ayer con el SENTINEL – OBSERVER, el profesor Silber discutió la discrepancia. “Verá, el software que en verdad fue entregado junto con el robot Robbie CX30 no fue el mismo que

supuestamente fue probado, ¡por lo menos de acuerdo con estos documentos!. Hemos podido determinar que el verdadero “código asesino”, tal como lo llamamos, fue escrito después de que se condujeron supuestamente las pruebas de software. Esto sugiere varias posibilidades. Primero, el proceso de prueba del software, por lo menos para esta parte crítica del software, fue falseado deliberadamente. Todos sabemos que hubo una enorme presión para tener listo a este robot en una fecha determinada. Otra posibilidad es que hubo una cierta dificultad en la versión de la gerencia en Silicon Techtronics, en cuanto a que el código correcto fue verdaderamente escrito, y probado con éxito, pero en el producto entregado se insertó el código equivocado”.

Solicitamos al profesor Silber que explicara qué quería decir con “versión de la gerencia”. “En un proyecto dado, un componente dado de software puede tener varias versiones, versión 1, versión 2, etc.”.

Esto refleja la evolución de ese componente a medida que avanza el proyecto. Se necesita tener algún tipo de mecanismo para tener control de las versiones de los componentes de software en un proyecto tan complejo como este. Tal vez el probador de software probó una versión correcta del código de dinámica del robot, pero en realidad se entregó una versión equivocada del mismo. No obstante, esto trae a colación una pregunta en cuanto a qué pasó con el código correcto.”

El profesor Silber se reclinó en su sillón. “Realmente esto es una gran tragedia. Si el código asesino hubiese sido pasado por el proceso de prueba de modo honesto, el robot nunca hubiese asesinado a Bart Matthews. Entonces, la pregunta es, ¿qué pasaba en Silicon Techtronics que no permitió una prueba honesta del código crítico?”

El SENTINEL – OBSERVER preguntó al profesor Silber si estaba de acuerdo con el concepto de que la interfaz del usuario fue la primordial culpable en este caso. “No creo en el argumento que esgrime mi colega, el profesor Gritty, que toda la culpabilidad en este caso pertenece al diseñador o diseñadores de la interfaz. Conuerdo con ciertas cosas que dice, pero no con todo. Debo preguntarme a mí mismo si Silicon Techtronics estaba poniendo mucho énfasis en la interfaz del usuario como la última línea de defensa contra el desastre. Esto es, ellos sabían que había un problema con la dinámica del robot, pero pensaron que la interfaz podría permitirle al operador manejarlo.”

El SENTINEL – OBSERVER preguntó entonces al profesor Silber sobre los cargos que se hacían en cuanto que nunca deberían haber aceptado la designación de Waterson para conducir una investigación objetiva del accidente. Las críticas señalan que la Universidad de Silicon Valley, y en particular el profesor Silber, tenían muchos intereses comunes con Silicon Techtronics, y de ese modo no podía ser elegido para conducir una investigación objetiva.

“Pienso que mi informe habla por mí mismo, “ replicó el profesor Silber, visiblemente molesto por nuestra pregunta. “Ya les he dicho a Ustedes los periodistas una y otra vez que no se trata de una investigación gubernamental sino de una interna de la corporación, de modo que creo que Silicon Techtronics tenía derecho a elegir a quien se le ocurriera, creo que yo les resultaba una persona con integridad.”

Ayer tarde, Sam Reynolds, el gerente del proyecto del CX30 contrató una abogada, Valerie Thomas. La Sra. Thomas hizo estas declaraciones a favor de su cliente: “ Mi cliente está escandalizado de que alguien de Silicon Techtronics haya podido engañar al profesor Silber en lo que concierne a las pruebas de software del robot Robbie CX30. El Sr. Reynolds asegura que el software fue probado y que él y otros sabían muy bien el hecho de que había algo que no funcionaba en el software de dinámica del robot. Sin embargo, el Sr. Ray Jonson, el superior inmediato de mi cliente en Silicon Valley, decidió que el robot fuera entregado a Cybernetics, Inc., basándose en la teoría del Sr. Johnson: “Nada es tan blanco como la nieve”.

Conforme con esta teoría, el software estaba casi libre de bugs y por ende podía ser liberado. Según el Sr. Johnson, el riesgo de falla era muy pequeño y el costo por demorar más la entrega del robot era muy alto. “Según mi cliente, el Sr. Johnson creyó que las condiciones del medio ambiente que podría llegar a disparar un comportamiento errático y violento del robot eran extremadamente improbables de ocurrir. Aún más, el Sr. Jonson creyó que el operador del robot no podría estar en peligro debido a que la interfaz del usuario

fue diseñada de modo de permitir al operador detener el robot fijo en sus guías en el caso de un movimiento del robot que comprometiera la vida del operador.”

El Sr. Jonson, jefe de la División de Robótica de Silicon Techtronics, no pudo ser ubicado para obtener sus comentarios. Randy Samuels será juzgado el mes entrante en la Corte de Silicon Valley. Cuando se lo contactó por teléfono, Samuels derivó todas las preguntas a su abogado, Alex Allendale.

Allendale tenía esto para decir con respecto a los descubrimientos del profesor Silber: “Mi cliente remitió el software en cuestión del modo usual junto con la documentación usual y con la esperanza de que su código fuera probado exhaustivamente. Desconocía hasta el momento de que saliera a la luz el informe del profesor Silber, que el código involucrado en esta terrible tragedia no había sido probado adecuadamente o que los resultados de prueba pudieran haber sido falsificados”.

“El Sr. Samuels nuevamente quiere expresar su gran pesar por este accidente. Él, más que nadie, quiere que se haga justicia en este caso. El Sr. Samuels nuevamente extiende sus mas sentidas condolencias a la Sra. Matthews y a sus hijos”.

1.3.8. Artículo 8 - Empleado de Silicon Techtronics admite falsificación de las pruebas de software.

Mensajes tomados del correo electrónico revelan nuevos detalles en el caso del “Robot Asesino”. Una asociación de computadores científicos lanza una investigación sobre violaciones al código de ética

Especial para el SENTINEL – OBSERVER de Silicon Vallley. Silicon Valley, EEUU por Mabel Muckraker

Cindy Yardley, una probadora de Silicon Techtronics admitió hoy que ella fue la persona que creó las pruebas de software fraudulentas del “robot asesino”. Las pruebas fueron reveladas a principios de la semana por el profesor Wesley Silber de la Universidad de Silicon Valley, con lo que se ha dado en llamar “El informe Silber”.

Se cuestionan los procedimientos de aseguramiento de la calidad que fueron realizados en el código del programa escrito por Randy Samuels, el programa acusado por asesinato no premeditado en el incidente del robot asesino. El Informe Silber afirma que los resultados de las pruebas reflejados en documentos internos de Silicon Techtronics son inconsistentes con respecto a los resultados de las pruebas obtenidas cuando fue probado el verdadero código del robot asesino.

Ayer al mediodía, en un acontecer inesperado, anunció su renuncia al cargo de Jefe de Seguridad de Silicon Techtronics, el Sr. Max Worthington, en una conferencia de prensa que fue transmitida en vivo por la CNN y otros informativos.

Worthington sacudió a los periodistas cuando comenzó su conferencia de prensa con el anuncio “Yo soy Marta”.

Worthington describió de este modo sus responsabilidades en Silicon Techtronics: “Básicamente, mi trabajo era proteger a Silicon Techtronics de todos los enemigos, locales y extranjeros. Por extranjeros quiero significar adversarios de otras corporaciones. Mi papel era más que nada de dirección. Aquellos que trabajaban bajo mi supervisión tenían muchas responsabilidades, incluyendo la de proteger la planta en sí, estar alertas por espionaje industrial e incluso sabotaje. También yo era responsable de vigilar a los empleados que pudiesen estar abusando de drogas o que de algún modo estuviesen siendo desleales con Silicon Techtronics.” Luego Worthington apuntó a una pila de volúmenes que había en una mesa a su izquierda. “Estos volúmenes representan tan solo algunos de los relevamientos electrónicos de empleados que yo hice a lo largo de los años para mi superior, el Sr. Waterson. Estas son impresiones de mensajes por e-mail que los empleados de Silicon Techtronics se enviaron entre si y a personas de otros sitios. Puedo decir con gran certeza que nunca jamás se le dijo a ningún empleado que se hacía este tipo de requisa

electrónica. No obstante, creo que la evidencia muestra que algunos empleados sospechaban que esto podía estar pasando.”

Varios periodistas preguntaron a los gritos quien en Silicon Techtronics estaba al tanto de esta requisa.

Worthington respondió. “Nadie sabía de esto a excepción del Sr. Waterson y yo, y uno de mis asistentes que era el responsable de conducir el monitoreo. Mi asistente producía un informe especial, resumiendo toda la actividad por e-mail de la semana, y ese informe era para que lo viera Waterson y yo solamente. Si se lo solicitaba, mi asistente podía dar un recuento más detallado de las comunicaciones electrónicas”. Worthington explicó que estaba poniendo a disposición de la prensa las transcripciones del correo electrónico porque quería que saliera a la luz toda la verdad sobre Silicon Techtronics y el incidente del robot asesino.

Los mensajes de e-mail entre empleados de Silicon Techtronics en verdad revelaron nuevas facetas del caso. Un mensaje de Cindy Yardley al Jefe de División de Robótica, Ray Jonson, indica que ella falsificó a su pedido los resultados de las pruebas. Acá esta el texto del mensaje:

A: Ray Jonson

De: Cindy Yardley

Asunto: Software de Samuels

Terminé de crear los resultados de las pruebas de software para ese software problemático, según tu idea de usar una simulación en vez del software propiamente dicho. Adjunto encontrarás el documento de prueba modificado, mostrando la simulación exitosa.

¿Le deberíamos decir a Randy sobre esto?

Cindy

La respuesta de Johnson al mensaje de Yardley sugiere que él sospechaba que el correo electrónico podía no ser seguro.

A: Cindy Yardley

De: Ray Jonson

Asunto: Re: Software de Samuels

Sabía que podía contar contigo. Estoy seguro de que tu dedicación a Silicon Techtronics te será pagada con creces. Por favor, en el futuro usa un medio de comunicación más seguro cuando discutimos este tema. Te aseguro que el modo en que manejamos esto fue completamente transparente, pero yo tengo mis enemigos acá mismo en la propia SiliTech.

Ray

Estas comunicaciones fueron intercambiadas justo días antes que se enviara al robot Robbie CX30 a Cybernetics Inc.. Este hecho es importante porque las pruebas de software falsificadas no fueron parte de un encubrimiento en el incidente del robot asesino. Estos hechos parecen indicar que el propósito de falsificar las pruebas de software era asegurarse de que el robot Robbie CX30 fuera entregado a Cybernetics, Inc. en la fecha que fue negociada entre Silicon Techtronics y Cybernetics.

Las transcripciones del correo electrónico revelan que hubieron repetidos mensajes de Ray Johnson a diferentes personas en el sentido de que la División de Robótica iba a ser cerrada definitivamente si el proyecto Robbie CX30 no está completado en término. En uno de los mensajes, diserta con su líder de proyecto, Sam Reynolds, acerca de la “teoría Ivory Snow”.

A: Sam Reynolds

De: Ray Jonson

Asunto: Re: ¡no seas perfeccionista!

Sam:

Tu y yo hemos tenido diferencias, pero debo decirte que personalmente me caes bien. Por favor entiende que todo lo que hago es con el propósito de SALVAGUARDAR TU TRABAJO Y EL TRABAJO DE TODOS EN ESTA DIVISIÓN. Yo te veo a ti y a toda la gente que trabajan para mi en la División de Robótica como mi familia.

Waterson fue claro: quiere tener el proyecto del robot completado a término. Y punto. Entonces, no tenemos otro recurso mas que el de "Ivory Snow". Sabes lo que quiero decir con eso. No tiene que ser perfecto. La interfaz del usuario es nuestro respaldo si esta versión del software para el robot tiene algunas fallas. El operador del robot va a estar seguro porque podrá cancelar cualquier movimiento del robot en cualquier momento. Concuerdo contigo en cuanto a que los requerimientos no funcionales son en algunas partes demasiados vagos. Lo ideal sería que si estos no fueran tiempos de apuro, cuantificáramos el tiempo que le llevaría al operador detener el robot en un caso de accidente.

Sin embargo no podemos negociar esto ahora. Como tampoco tenemos tiempo para diseñar requerimientos no funcionales nuevos y mas precisos.

No puedo enfatizar suficientemente de que estos son tiempos de apurarse. A Waterson no le cuesta nada deshacerse de toda la División Robótica. Sus amigos del Wall Street sólo le van a decir ¡Felicitaciones!. Veras, para Waterson nosotros somos tan solo del montón.

Ray

En este mensaje Ray Johnson parecía estar menos preocupado por la seguridad de comunicarse por correo electrónico.

El SENTINEL – OBSERVER entrevistó ayer por la tarde a Cindy Yardley en su propia casa. No se pudieron contar ni a Ray Jhonson ni a Sam Reynolds.

La Srta. Yardley estaba notoriamente ofuscada porque sus mensajes por e-mail fueran dados a conocer a la prensa. "De alguna forma me siento aliviada. Sentí una enorme culpa cuando ese hombre fue muerto por un robot que yo ayude a construir. Una tremenda culpa."

El SENTINEL – OBSERVER pregunto a la Srta. Yardley si es que ella había hecho una elección ética al acceder a falsear los resultados de las pruebas de software., respondió con gran emoción. "Nada, pero nada a lo largo de mi experiencia y background me preparó para algo como lo que pasó. Estudie ciencias de la computación en una universidad de gran nivel y allí me enseñaron sobre pruebas del software, ¡pero jamás me dijeron que alguien con poder sobre mi me pediría generar una prueba falseada!".

"Cuando Jonson me pidió que lo hiciera, me llamó a su oficina, como para mostrarme las trampas del poder; verá, algún día me gustaría estar en un puesto gerencial. Me senté en su oficina y vino directamente y me dijo: "Quiero que falsifiques los resultados de las pruebas de Samuels. No quiero que Reynolds se entere de nada de esto".

Yardley contuvo las lagrimas. "Me aseguró que probablemente nadie vería jamás los resultado de las pruebas dado que el robot está perfectamente seguro. Era tan solo una cuestión interna, un tema de prolijidad, en caso de que alguien en Cybernetics o de un puesto alto dentro de la corporación le diera curiosidad de ver los resultados de las pruebas. Le pregunte si estaba seguro de que el robot era seguro y todo eso y me dijo: "¡Es seguro!". La interfaz del usuario es nuestra línea de defensa. En alrededor de seis meses podemos enviar una segunda versión del software del robot y para entonces este problema de Samuels estará resuelto."

Yardley se reclinó en su asiento como si lo que dijera a continuación necesitara de un énfasis especial. "Entonces me dijo que si yo no falsificaba las pruebas, todos los de la División de Robótica perderían sus trabajos. Sobre esa base decidí falsificar las pruebas, trataba de proteger mi trabajo y el de mis compañeros."

La Srta. Yardley está al presente cursando un grado de maestría en Administración de Empresas en la Universidad de Silicon Valley.

Luego el SENTINEL – OBSERVER preguntó a la Srta. Yardley si aún sentía que había tomado una decisión ética, en vista de la muerte de Bart Matthews. “Creo que fui manipulada por Ray Jonson. El me dijo que el robot era seguro.”

Otra revelación, contenida en las transcripciones del correo electrónico dadas a conocer, fue el hecho de que Randy Samuels hurto parte del software que usó en el proyecto del robot asesino. Este hecho se reveló en un mensaje que Samuels envió a Yardley cuando ella probó por primera vez su software y dió resultados erróneos:

A: Cindy Yardley

De: Randy Samuels

Asunto: Re: Maldito si lo sé

Por mi vida, no puedo entender que es lo que anda mal en esta función balancear_brazo(). Verifique la fórmula de dinámica del robot una y otra vez y pareciera estar implementada correctamente. Como sabes, la función balancear_brazo() invoca a 14 funciones diferentes. A cinco de ellas las tomé tal cual del paquete estadístico PACKSTAT 1-2-3. ¡Por favor no se lo digas a nadie! No son éstas las que causarían el problema, ¿o sí?

Randy

Los expertos le dijeron al SENTINEL – OBSERVER que tomar software de paquetes comerciales de software como el PACKSTAT 1-2-3 es una violación a la ley. El software tal como el inmensamente popular PACKSTAT 1-2-3 esta protegido por el mismo copyright que protege al material impreso.

Mike Waterson, Presidente ejecutivo del Silicon Techtronics, emitió una enojosa declaración porque Max Worthington había dado a conocer las transcripciones del correo electrónico “confidencial”. Las declaraciones de Waterson decían, en parte, que “Yo le pedí a nuestros abogados que intervinieran en este tema. Consideramos que esas transcripciones son propiedad exclusiva de Silicon Techtronics. Nuestra intención es efectuar cargos ya sea civiles o criminales contra el Sr. Worthington.”

Como reacción a lo ocurrió ayer en el caso del robot asesino, la ACM o Association for Computer Machinery anunció su intención de investigar si algún miembro de la ACM de Silicon Techtronics ha violado el código de ética de la ACM. La ACM es asociación internacional de computadores científicos con 85.000 miembros.

La Dra. Turina Babbage, presidente de la ACM, hizo una declaración en la Conferencia de Ciencias de la Computación de ACM que se lleva a cabo cada invierno y que esta temporada se hará en Suluth, Minnesota.

Un extracto de las declaraciones de la Dra. Babbage sigue a continuación:

“Todos los miembros de la ACM están ligados por el código de ética y Conducta Profesional de la ACM (Nota al pie: Un borrador de este código fue dado a conocer en Comunicaciones de la ACM, Mayo 1992. Por favor nótese que las declaraciones hechas por la ficticia Dra. Babbage contienen citas del verdadero código de ACM). Este código establece, en parte, que los miembros de ACM tiene el imperativo moral de contribuir con el bienestar de la sociedad y los hombres, evitar daños a terceros, ser honestos y confiables, dar crédito adecuado a la propiedad intelectual, acceder a los recursos de comunicación y de computación solo cuando así lo estén autorizados, respetar la privacidad de terceros y honrar la confidencialidad.

Más allá de eso, existen responsabilidades profesionales tales como la obligación de cumplir los contratos, acuerdos y responsabilidades asignadas, y de dar evaluaciones profundas y completas de los sistemas de computación y de sus impactos, poniendo especial énfasis en los riesgos potenciales.

Varias de las personas involucradas en el caso del robot asesino son miembros de la ACM y hay causas para creer que han incurrido en violación de código de ética de nuestra asociación. Por lo tanto, estoy

solicitando al directorio de la ACM designar una Fuerza de Tareas para investigar a los miembros de la ACM que puedan haber violado groseramente el código.

No tomamos este paso a la ligera. Esta sanción ha sido aplicada sólo rara vez, pero el incidente del robot asesino no sólo ha costado una vida humana, sino que ha causado mucho daño a la reputación de la profesión de computación.

1.3.9. Artículo 9 - La revista dominical del Sentinel Observer

*Una conversación con el Dr. Harry Yoder
Por Robert Franklin*

Harry Yoder es una figura bien conocida en el campo universitario de Silicon Valley. El profesor de Tecnología y Ética de la Computación de Samuel Southerland ha escrito numerosos artículos y textos sobre ética y el impacto social de las computadoras. Sus clases son muy famosas, y muchos de sus cursos están completos mucho antes de que finalice el período de inscripción. El Dr. Yoder se ha recibido del Doctorado en Ingeniería Eléctrica del Instituto de Tecnología de Georgia en 1958. En 1976 recibió un grado en “Maestría en Divinidad” del Harvard Divinity School. En 1983 recibió un Master en Ciencias de la Computación de la Universidad de Washington. Ingresó en la facultad de la Universidad de Valley en 1988. Entrevisté al Dr. Yoder en su oficina del campus. Mi intención era obtener su reacción con respecto al caso del robot asesino, y “leer su pensamiento” sobre los temas éticos que involucra el caso.

Aquí la entrevista:

SENTINEL – OBSERVER: Ir de la Ingeniería eléctrica al estudio de la religión parece un gran salto.

YODER: Yo era un Ingeniero electricista por profesión, pero todos los seres humanos tienen una vida interior, ¿no lo cree así?

SENTINEL – OBSERVER: Sí.

YODER: ¿De qué se trata su vida interior?

SENTINEL – OBSERVER: Tratar de hacer lo correcto. También se trata de lograr la excelencia en lo que hago. ¿Es eso lo que lo llevó a la Escuela de Divinidad de Harvard? ¿Usted quería clarificar su vida interior?

YODER: Sucedían muchas cosas en la Escuela de Divinidad, y muchas de ellas eran muy poderosas. Sin embargo, más que nada quería comprender la diferencia entre lo que estaba bien y lo que estaba mal.

SENTINEL – OBSERVER: ¿Y qué hay de Dios?

YODER: Si, estudié mi propia religión cristiana y a la mayoría de las religiones del mundo, y todas ellas tenían cosas interesantes que decir acerca de Dios. No obstante, cuando yo discuto sobre ética en un foro tal como este, que es secular, o cuando discuto de ética en mis cursos de ética de la computación, no coloco a esa discusión en un contexto religioso. Pienso que la fe religiosa puede ayudarle a una persona a abrazar la ética, pero por otra parte, todos sabemos que ciertas personalidades notorias que se han autopronunciado religiosas han sido altamente no éticas. De este modo, cuando yo discuto sobre ética de la computación, el punto de partida no es la religión, sino mas bien un acuerdo común entre mis estudiante y yo de que queremos ser gente ética, que luchar por la excelencia ética es una tarea humana que vale la pena. Por lo menos, lo que no queremos es herir a otros, no queremos mentir, robar, hacer trampas, asesinar, etc.

SENTINEL – OBSERVER: ¿Quién es el responsable de la muerte de Bart Matthews?

YODER: Por favor discúlpeme si lo remito nuevamente a la Escuela de la Divinidad de Harvard, pero creo que uno de mis profesores de allí tiene la respuesta correcta a su pregunta. Este profesor era un hombre mayor, tal vez de 70 años, de la Escuela Oriental, un rabino. Este rabino dijo que de acuerdo al Talmud, una tradición antigua de la ley judía, si se derrama sangre inocente en un pueblo, entonces los líderes de ese pueblo deben ir a los límites del mismo y realizar un acto de penitencia. Esto es además de la justicia que se aplicará a la persona o personas que cometieron el asesinato.

SENTINEL – OBSERVER: Ese es un concepto interesante.

YODER: ¡Y uno de verdad! Un pueblo, una ciudad, una corporación son sistemas en que la parte esta ligada al todo y el todo a la parte.

SENTINEL – OBSERVER: Usted quiere decir que los lideres de Silicon Techtronics, tales como Mike Waterson y Ray Jonson, deberían haber asumido la responsabilidad por este incidente desde el vamos. Además, tal vez otros individuos, como ser Randy Samuels y Cindy Yardley, comparten una carga especial de responsabilidad.

YODER: Si, responsabilidad, no culpabilidad. La culpabilidad es un concepto legal y la culpabilidad o la inocencia de las partes involucradas, sean ya en lo criminal o lo civil, será decidida en la corte. Estimo que una persona es responsable por la muerte de Bath Matthews si su acción ha ayudado a causar el incidente – es una cuestión de causalidad, independientemente de los juicios éticos y legales. Las cuestiones de responsabilidad podrían serle de interés a los Ingenieros de software y gerentes, quienes tal vez querrían analizar que es lo anduvo mal, de modo de evitar que similares problemas ocurrieran en el futuro.

Mucho de lo que salió de los medios con respecto a este caso indica que Silicon Techtronics era una organización enferma. Esa enfermedad creó el accidente. ¿Quién creó la enfermedad? La gerencia creó esa enfermedad, pero también los empleados que no tomaron las decisiones éticas correctas contribuyeron con la misma.

Tanto Randy Samuels como Cindy Yardley eran recién egresados. Se graduaron en ciencias de la computación y su primera experiencia en el mundo laboral fue en Silicon Techtronics. Uno debería preguntarse si recibieron alguna enseñanza sobre ética. Relacionado a esto esta la cuestión de si alguno de ellos tenia con anterioridad experiencia en trabajo en grupo. En el momento en que se los asigno al desarrollo del robot asesino, ¿Ellos vieron la necesidad de ser personas éticas? ¿Vieron que el éxito como profesional requiere de un comportamiento ético? Hay mucho mas para ser científico en computación o ingeniero de software que tan solo la habilidad y el conocimiento de la ética.

SENTINEL – OBSERVER: Sé con seguridad que ninguno de los dos tomó cursos sobre ética o ética de computación.

YODER: Lo sospechaba. Veamos a Randy Samuels. Basándome en lo que leí en su periódico y en otros lados, era básicamente de los del tipo “hacker”. Amaba la computación y la programación. Comenzó a programar en los primeros años de la secundaria y continuó a lo largo de toda su carrera universitaria. El punto importante es que Samuels aún era un “hacker” cuanto entro a Silicon Techtronics y ellos permitieron que el siguiera siendo así. Estoy usando el termino “hacker” de un modo peyorativo y tal vez no del todo justo. El punto que estoy tratando de remarcar es que Samuels nunca maduró mas allá de su angosto enfoque como hacker. En Silicon Techtronics, Samuels aún mantuvo esta actitud en lo que hacia a sus funciones de programador, la misma que tenía cuando estaba en la secundaria. Su percepción de la vida y de sus responsabilidades no creció. El no maduró. No hay en evidencia de que tratara de desarrollarse y convertirse en una persona ética.

SENTINEL – OBSERVER: Una dificultad, en lo que hace a enseñar ética, es que en general a los estudiantes no les gusta que se les diga “esto esta bien y aquello esta mal”.

YODER: Los alumnos necesitan entender que el tocar temas de ética es parte de computadores científicos o ingenieros de software profesionales.

Una cosa que me ha fascinado acerca de la situación en Silicon Techtronics es que a veces es difícil ver los límites entre lo legal, lo técnico y lo ético. Los temas técnicos involucran temas de gerencia y de computación. He llegado a la conclusión de que este desvanecimiento de los límites resulta del hecho de que la industrial del software aún se encuentra en pañales. Los temas éticos surgen abruptamente en parte porque hay una ausencia de lineamientos técnicos y legales.

En particular, no existen prácticas normalizadas para desarrollar o probar software. Hay estándares, pero no lo son realmente. Una broma muy común entre los computadores científicos es que lo bueno de los estándares es que hay mucho para elegir.

Ante la ausencia de prácticas normalizadas aceptadas universalmente para ingeniería del software, surgen muchos juicios de valor, probablemente más que cualquier otra forma de producción. Por ejemplo, en el caso del robot asesino, hubo una controversia con respecto al uso del modelo de cascada versus el de prototipo. Debido a que no había un proceso de desarrollo de software estandarizado, esto se transformó en una controversia, y los temas éticos surgen por el modo en que se resuelve la controversia. Usted recordará que el modelo de cascada no fue elegido por sus méritos sino porque el gerente del proyecto tenía experiencia en este.

SENTINEL – OBSERVER: ¿Usted cree que Cindy Yardley actuó éticamente?

YODER: Al principio su argumento parece más poderoso: ella, efectivamente mintió, para así salvar los puestos de trabajo de sus compañeros, y por supuesto, el de ella. Pero, ¿siempre es correcto mentir, para crear una falsedad, en un marco profesional?.

Un libro que usado en mis cursos de ética de la computación es el “Ethical Decision Making and Information Technology” (Toma de Decisión Ética y Tecnología de la Información) de Kallman y Grillo.

NOTA: Este texto es un texto real y esta publicado por McGraw Hill.

En este libro se dan algunos de los principios y teorías que están detrás de la toma de decisiones. Yo uso este y otros libros para ayudar a que los alumnos desarrollen sus apreciaciones sobre la naturaleza de dilemas éticos y toma ética de las decisiones.

Kallman y Grillo presentan un método para la toma de decisión éticas y parte de su método consiste en el uso de cinco pruebas: la prueba de la mamá: ¿Le diría Ud. a su mamá lo que hizo?; la prueba de la TV: ¿Le diría Ud. a una audiencia nacional de TV lo que hizo?; la prueba del olfato: ¿lo que Ud. hizo tiene mal olor?; la prueba de ponerse los zapatos del otro: ¿le gustaría que el otro le haga lo que Ud. hizo?; y la prueba del mercado: ¿Sería su acción una buena estrategia de venta?

Lo que hizo Yardley reprobó todas estas pruebas, pienso que todos concuerdan conmigo. Por ejemplo, ¿puede imaginar a Silicon Techtronics usando una campaña publicitaria que diga algo como?: “En Silicon Techtronics el software que Usted recibirá de nosotros esta libre de bugs, porque aún cuando haya uno, distorsionaremos los resultados de las pruebas para esconderlo, Usted nunca se enterará. La ignorancia es la felicidad.”

Esto demuestra que el altruismo aparente no es un indicador suficiente de un comportamiento ético. Uno podría preguntarse que otros movimientos no declarados tenía la Srta. Yardley. ¿Podría ser que la ambición personal la llevara a aceptar la explicación que le dió Ray Jhonson y su afirmación que el robot era seguro?

SENTINEL – OBSERVER: ¿Usted cree que Randy Samuels actuó éticamente?

YODER: Robar software en el modo que lo hizo es tanto ilegal como no ético.

Pienso que el punto más importante con Randy Samuels nunca fue discutido en los medios de prensa. Honestamente dudo que Samuels tuviera el conocimiento necesario para su puesto. Este tipo de conocimiento se lo llama conocimiento de la especialidad. Samuels tenía conocimiento de computación y programación pero no tenía un sólido conocimiento de la física en especial de la mecánica clásica. Su falta de conocimiento en el dominio de la aplicación fue una causa directa del horrible accidente. Si alguien con conocimientos de matemáticas, estadística y física hubiera programado el robot en lugar de Samuels, probablemente hoy Bart Matthews estuviera vivo. No tengo dudas de ello. Samuels malentendió la fórmula física porque no entendió su significado e importancia en la aplicación en el robot. Puede ser que la gerencia sea en parte responsable de esta situación. Puede que Samuels les haya dicho acerca de sus limitaciones y la gerencia habrá dicho. “Y bueno, que importa”.

Samuels tenía dificultades en trabajar en equipo, hacer revisiones en conjunto, y programar sin egoísmo. ¿Es posible que estuviera intentando esconder su falta de experiencia en el área?

SENTINEL – OBSERVER: ¿Cree que Ray Johnson actuó éticamente?

YODER: ¡Este tema de “Ivory Snow”! El problema con la teoría de Ivory Snow fue tan solo una racionalización para sacarse de encima a software fallado y entregarlo en término al cliente. Esta teoría solo es válida, ética y profesionalmente, si al cliente se le informa de los bugs de lo que se tiene conocimiento, o de impurezas, utilizando la jerga. En el caso de Silicon Techtronics, la teoría de Ivory Snow funcionó así: ¡sabemos que no lo es, pero al cliente hay que decirle que sí lo es!

Desde luego, presionar a Cindy Yardley como lo hizo Ray Jonson tampoco es ético. ¿El creía en lo que le dijo a la Srta. Yardley, es decir, que el robot era seguro, o fue eso una mentira del momento? Si el creía que el robot era seguro, entonces ¿por qué cubrirse con pruebas falsa? Si la interfaz con el usuario era tan importante como la última línea de defensa, entonces ¿por qué evitar pruebas más rigurosas de la interfaz?.

SENTINEL – OBSERVER: ¿Qué piensa de Mike Waterson en todo esto?

YODER: Si Jonson es el padre de la teoría de Ivory Snow, Waterson es el abuelo. Su exigencia de que el robot estuviera completado para una fecha determinada o de lo contrario “rodarían cabezas”, puede haber causado que Jonson formulara la teoría de Ivory Snow. Verá, es evidente que Jonson pensaba que era imposible entregar en Cybernetics Inc. el robot CX30 para una fecha determinada, a menos que el software fuera con bugs.

En muchos sentidos pienso que Waterson actuó sin ética e irresponsablemente. Pone a Sam Reynolds a cargo del proyecto del robot, cuando aún él, Reynolds, carecía de experiencia en robot e interfaces con el usuario modernas, Reynolds rechazó la idea de desarrollar un prototipo, lo que podría haber permitido el desarrollo de una mejor interfaz.

Waterson creó una atmósfera opresiva entre sus empleados, que en sí mismo es falto de ética. No solo amenazó con despedir a todos los de la División de Robótica si el robot no se terminaba a tiempo, sino que hurgó en comunicaciones por correo electrónico privadas de toda la corporación, un derecho controvertido que algunas empresas alegan tener.

Mi creencia personal es que este tipo de investigaciones es falto de ética. La naturaleza del e-mail es algo así como un híbrido de correspondencia común y conversación telefónica. Monitorear o espiar correspondencia ajena está considerado no ético, tal como lo es interferir el teléfono. Por cierto, estas actividades también son ilegales bajo la mayoría de las circunstancias. O sea, creo que monitorear a los empleados del modo que lo hizo Waterson es un abuso de poder.

SENTINEL – OBSERVER: ¿Usted cree que en esto el fiscal tiene un caso?

YODER: ¿Contra Randy Samuels?

SENTINEL – OBSERVER: Sí.

YODER: Lo dudo, a menos que ella tenga información que hasta ahora no se ha hecho pública. El asesinato no premeditado, a mi entender, implica un tipo de acto irresponsable y negligente, que causa la muerte de un tercero. ¿Se aplica esta descripción a Samuels? Pienso que la mejor apuesta de la fiscal es hacer hincapié en su falta de conocimiento en el área de aplicaciones, si puede mostrarse que Samuels se involucró deliberadamente en un fraude.

La semana pasada leí que el 79% de la gente esta a favor de la absolución. La gente es proclive a acusar a la compañía y a sus gerentes. La otra noche, uno de los noticieros dijo: “Samuels no es un asesino, es un producto de lo que lo rodea”.

SENTINEL – OBSERVER: ¿podría nuevamente decir su posición sobre el tema de la responsabilidad final en el caso del robot asesino?

YODER: En mi mente, el tema de la responsabilidad de un individuo versus la responsabilidad de la corporación, es un tema muy importante. La corporación creó un entorno en el que podían ocurrir este tipo de accidentes. Aún así, los individuos, dentro de este sistema, actuaron sin ética e irresponsablemente, y fueron los que de hecho causaron el accidente. Una compañía puede crear un entorno que saca a flote lo peor de sus empleados, pero cada empleado también puede contribuir a empeorar ese ambiente corporativo. Este es un plazo cerrado que se alimenta a si mismo, un sistema en el sentido clásico. Entonces, hay cierta responsabilidad de la corporación y cierta responsabilidad de los individuos en el caso del robot asesino.

SENTINEL – OBSERVER: Muchas gracias Profesor Yoder.

Guía de Trabajo Práctico 2: Diagramas de Flujo de Datos (DFDs)

2.1. Ejercicio 1

TÍTULOS

Objetivos: Analizar el siguiente texto y realizar el diagrama de contexto, tabla de eventos y DFD correspondientes.

Cuando un alumno rinde su último final, se presenta en la oficina de Títulos y presenta el "Formulario KOOZ1-", una fotocopia del documento y la libreta universitaria (con toda esta documentación se genera un expediente). Se controla que el formulario se haya completado correctamente, se verifica que los datos personales y académicos coincidan con el archivo 'Datos Actualizados' que quincenalmente envía Secretaría Académica. Si todo es correcto, se genera un comprobante de aceptación, por duplicado, donde consta la fecha de inicio del trámite, la documentación presentada y el tipo de trámite que se inicia (Extensión de título, Revalidación del título, etc.). El original se envía al Secretario de Decanato junto con el expediente y el original se le entrega al alumno.

En Decanato, cuando termina el periodo lectivo, el Secretario envía los expedientes recibidos a la supervisora de Títulos, Ana Martínez, quien se encarga de distribuirlo entre todos los analistas del sector. Primero se verifica que no adeude ningún final del plan que corresponda. Si adeuda alguna materia se genera una nota de Rechazo de expediente que se envía junto con el expediente a la oficina de Títulos. Si no adeuda materias, se controla que haya cumplido con el régimen de correlatividades. Si hay alguna infracción de correlativas se genera una nota de Documentación Incorrecta que se le envía al alumno, En la nota se detalla la/s materias en las que se cometió la infracción.

Cuando el alumno recibe la nota de Documentación Incorrecta controla las materias contra el plan de correlativas y si constata que no cumplió con el régimen de correlativas genera una nota de Excepción de correlativas por triplicado, el original lo presenta en Mesa de Entradas, el duplicado lo envía a la oficina de Títulos y archiva el triplicado.

Si en Decanato se constata que no cometió ninguna infracción, se aprueba el expediente y se genera un pedido de Extensión de Título por duplicado y un memo por triplicado. El original del pedido de Extensión de Título se envía junto con el expediente al Ministerio de Educación. El original del memo, que acredita que el expediente fue enviado al Ministerio, se envía a la oficina de Títulos, el duplicado al Secretario de Decanato y el triplicado se archiva.

2.2. Ejercicio 2

GESTORÍA "PLACENTE, GANCEDO Y CÍA."

Objetivos: Armar el Diagrama de Contexto Tabla de Eventos y DFD.

La gestoría "Placente, Gancedo y Cía." se dedica a realizar trámites referentes al pago de impuestos y moratorias. Además gestiona habilitaciones municipales.

Cuando un interesado se acerca para iniciar un trámite, se verifica en el archivo 'Clientes' que haya realizado un trámite anteriormente a través de la gestoría. Si es así, se le realizará un descuento en los gastos administrativos. Se controla que la gestoría pueda realizar el trámite. Si lo puede realizar, se consultan las distintas tasas de pago enviadas semanalmente por cada municipalidad y se genera el informe preliminar de presentación por triplicado. El duplicado se le entrega al cliente, el triplicado se archiva en la carpeta de 'Trámites Presentados' y, el original se envía a la municipalidad. El informe contiene el número de trámite asignado, el gestor que lo realiza, el monto de impuesto y el porcentaje que cobrará la gestoría por el trámite. Si la gestoría no está en condiciones de realizar ese tipo de trámite, se le da un folleto al cliente con las gestorías habilitadas. En este caso se genera un memo con el tipo de servicio que no se pudo prestar para el dueño.

El descuento que se realiza en los gastos administrativos se determina de la siguiente manera: si el cliente ha realizado más de 3 tramites de tipo municipal el descuento será del 5%. Si realizó sólo un trámite el descuento es del 2%. Si gestionó alguna habilitación municipal se le realizará un descuento del 1% más.

Cuando en la municipalidad se recibe el informe, el asistente controla contra los libros de deudas generados por zonas y, si corresponde, emite un formulario completando la fecha, quien lo recibió, el número de folio, rubrica del libro y el nombre del inspector municipal que corresponde a la zona. Al final del día genera un listado de estadísticas con los informes recibidos y lo envía al inspector zonal correspondiente.

El señor Ángel, encargado de las finanzas de la gestoría, cada miércoles genera estadísticas sobre los trámites presentados. Emite un listado por duplicado (original para el dueño de la gestoría y duplicado para el Estudio Contable que los asesora) donde consta el número de trámite, el nombre del gestor, el código de zona municipal (asignado según un nomenclador provincial), la dirección de la delegación municipal (cada municipalidad envía a las gestorías cada seis meses el padrón de delegaciones municipales), la tasa cobrada en concepto de impuesto y, si es moratoria, también se incluye el interés correspondiente.

2.3. Ejercicio 3

DISTRIBUIDORA "SU REMEDIO"

Objetivos: Armar el Diagrama de Contexto, Tabla de Eventos, DFD y Tabla de decisión

La distribuidora de medicamentos "Su Remedio" es intermediaria entre distintos laboratorios que elaboran medicamentos y las farmacias que los venden. Cada farmacia se identifica con un número de cliente.

Cuando una farmacia quiere realizar un pedido se comunica telefónicamente con el departamento de ventas y solicita los medicamentos. La telemarketer carga el pedido en el sistema y corta la comunicación con la farmacia. En ese momento verifica que la cuenta corriente del cliente se encuentre al día, para ello consulta el archivo "Cuentas Corrientes" que diariamente genera el Jefe de Créditos con el número de cliente y el crédito disponible. Si el crédito disponible no es suficiente, se le avisa al Jefe de Créditos que el pedido no fue aprobado, informando el número de cliente y el número de pedido. Además, se genera un nuevo registro en el archivo Pedidos Rechazados donde se guarda la fecha, el número de pedido, el número de cliente, el estado crediticio, la diferencia del rechazo y los medicamentos solicitados (código de medicamento, cantidad solicitada). Si el crédito está bien, se controla el stock en el listado de stock proveniente de Depósito que contiene el código de medicamento y la cantidad. Si el stock es suficiente se genera un nuevo registro en el archivo Pedidos Aprobados donde se graba la fecha, el número de pedido, el número de cliente, y la información de cada medicamento solicitado: código de medicamento, laboratorio, cantidad comprada, precio, descuento, bonificaciones e importe neto del medicamento. Además se graba el importe total del pedido. Para determinar las bonificaciones se consulta la información que diariamente envía el Jefe de Ventas sobre Bonificaciones. La telemarketer cuenta con una base de datos de Medicamentos (código de medicamento, nombre, laboratorio, presentación y acción terapéutica) y sus Precios (Laboratorio, código de medicamento, precio); ambos son enviados semanalmente desde cada laboratorio.

Una vez generado el pedido aprobado, se emite una nota de remito, por duplicado, donde consta el número de cliente, por cada medicamento su código y cantidad a enviar, la calificación de entrega y si corresponde alguna promoción. Para ello tiene en cuenta que los medicamentos se clasifican según su tipo en: oncológico, quirúrgico y no vital. Los medicamentos pueden ser solicitados por la farmacia para mantener el stock o porque fueron pedidos especialmente por un paciente.

Si el medicamento es oncológico y fue pedido por un paciente se lo califica como Entrega Urgente. En caso contrario (para actualizar stock) se lo califica como Entrega Normal. En ambos casos si la cantidad pedida es mayor a 3, se le regala una unidad a la farmacia como promoción.

Si el medicamento es quirúrgico y fue pedido por un paciente, se lo califica como Entrega Urgente. En caso contrario (para actualizar stock) se lo califica como Entrega Normal. En ambos casos, si la cantidad pedida es mayor a 2 se le regala un "kit quirúrgico" a la farmacia como promoción.

Si el medicamento es No vital se lo califica como Entrega Normal. Si la cantidad pedida es mayor a 10 se le regala una unidad a la farmacia como promoción.

El original de la nota de remito se envía a Depósito y el duplicado se archiva. En Depósito se prepara el pedido y se le avisa al Jefe de Repartos de Correo Expreso S.A. que es la encargada de la distribución de los pedidos.

El responsable de Correo Expreso S.A, cuando recibe el aviso, genera una orden de reparto por triplicado. El original se lo envía al cartero seleccionado según la zona de destino del pedido, el duplicado lo pasa al sector Administración para que actualice el archivo "Repartos Realizados por Empresa" donde se graba el código de la empresa, el cartero seleccionado y el tipo de reparto. El triplicado lo archiva.

Todos los días entre las 14hs. y las 15hs. el Jefe de Créditos toma la información del archivo Pedidos Rechazados y la analiza. Teniendo en cuenta el historial de pagos enviado diariamente por Cobranzas, aprueba los pedidos que tengan buen comportamiento histórico de pago. Genera un nuevo registro en el archivo Pedidos Aprobados grabando el número de pedido, el número de cliente, el código de medicamento, laboratorio, cantidad comprada, precio, importe neto del medicamento e importe total del pedido. En este caso no se realizarán descuentos ni bonificaciones sobre el pedido. Además emite una planilla con el pedido aprobado donde consta el número de pedido, el número de cliente, la fecha de aprobación y el detalle de medicamentos. Al final del día envía todas las planillas al Jefe de Ventas.

El anteúltimo día hábil de cada mes, en el sector Cobranzas se realiza la facturación de los pedidos basándose en los Pedidos Aprobados del mes. Se tienen en cuenta el informe de Impuestos Vigentes (fecha, motivo de impuesto, porcentaje) que recibió el sector y las constancias de Pedidos Entregados que semanalmente envía Depósito. Se emite la factura por triplicado, enviándose el original a la farmacia y el duplicado a Tesorería. El triplicado se archiva.

2.4. Ejercicio 4

COMPRADOR FRECUENTE

Objetivos: Armar el Diagrama de Contexto, Tabla de Eventos, DFD y Tabla de decisión

La empresa Comprador Frecuente S.A. se dedica a entregar tarjetas de fidelización a clientes que mediante compras a comercios específicos, reciben puntos que luego pueden canjear por premios.

Cuando el interesado en la tarjeta quiere suscribirse completa el formulario de afiliación con su nombre, apellido, tipo de documento, número de documento, fecha de nacimiento y número de teléfono.

Una vez por semana el analista de suscripciones verifica los pedidos de afiliación recibidos. Para ello, se conecta al sistema de Deudas S.A (empresa que brinda información crediticia sobre personas) y realiza una consulta online. Si el solicitante es deudor, automáticamente se rechaza la petición y se le envía una nota al solicitante informándole la situación, Una copia de esta nota se envía a Auditoría (quien realiza un informe con todos los rechazos para el Director), Si no hay problemas crediticios, se genera un formulario de Alta de Cliente por duplicado, El original se envía a la empresa Tarjetas S.A. para que imprima la tarjeta y, el duplicado, se envía a Atención al Cliente. Además, genera un nuevo registro en el archivo Tarjetas Activas, donde se define el nombre y apellido, el número de cliente, el saldo en puntos (se crea con 0), la fecha de alta y la fecha de vencimiento de la tarjeta.

Cuando el dueño de Tarjetas S.A, recibe el formulario de Alta de Cliente, verifica si la compañía que hace el pedido no tiene saldos pendientes de pago. Para ello verifica en el archivo Saldos Pendientes (que recibe diariamente de Cuentas Corrientes). Si la compañía tiene deudas pendientes genera una nota de Rechazo por Deuda por duplicado. Envía el original a la empresa que solicitó la impresión y el duplicado lo archiva. Si no hay deuda pasa el pedido al Sector Impresiones.

Cuando el cliente realiza una compra y presenta su tarjeta en un negocio adherido automáticamente el sistema verifica la cantidad de puntos a bonificar. Para esta asignación tiene en cuenta: si el importe de compra está entre \$10 y \$100 y el pago es en efectivo, se le asignan 10 puntos (además se genera un vale por un CD a elección). En caso de que el pago sea con tarjeta también se le asignan 10 puntos (si la cantidad de cuotas es menor o igual a 3), Si la cantidad de cuotas es mayor a 3 sólo se le asignan 8 puntos.

Cuando el importe de compra está entre \$101 y \$500 se le asignan 20 puntos (si el pago fue en efectivo o con tarjeta en 3 cuotas o menos). Si se pagó en más de 3 cuotas, se le asignan 15 puntos. En el caso de pago en efectivo, se genera un vale para un bolso.

Si el importe de compra es mayor a \$500, se le asignan 50 puntos y se imprime un vale por un día de spa.

La información con los vales de regalos es cargada diariamente por el responsable del sector Promociones. Se genera un comprobante de puntaje que se le entrega al cliente junto con el vale en los casos que corresponda. Se actualiza en el archivo Tarjetas Activas el saldo de puntos.

Cuando el cliente se presenta para canjear sus puntos por un premio, en Atención al Cliente, se verifica en el listado de Premios (que semanalmente envía Marketing), la cantidad de puntos necesarios para el canje. Luego se verifica el saldo de puntos del cliente. Si la cantidad de puntos le alcanza para el canje, se consulta al encargado del Depósito si hay existencia. Si hay existencia, se le entrega el premio al cliente junto con un comprobante con el nuevo saldo. Se actualiza el saldo. Además, se actualiza (si ya realizó un canje anterior) o se da de alta (si es el primer canje) un registro en Historial de Canjes donde se detalla el número de cliente, el código de premio y la fecha de canje.

Todos los viernes, el supervisor verifica según el historial de canjes y el informe de disponibilidad que diariamente envía el Supervisor del Depósito, si se deben reponer los artículos que se otorgan como premio. Si algún premio está en el límite de stock consulta a Marketing si se realiza el pedido de reposición o, si ese premio será reemplazado por otro. Si hay que reponerlo, se genera una orden de reposición/ que se envía al Proveedor. Si el premio será reemplazado por otro, se le envía a Compras un formulario de Producto Nuevo. En ambos casos se genera un Aviso de Reposición/Compra a Tesorería y al Supervisor del Depósito. Tesorería genera y devuelve un Comprobante de Recepción de la solicitud de Reposición/Compra.

2.5. Ejercicio 5

LABORATORIUM

Objetivos: Armar el Diagrama de Contexto, Tabla de Eventos y DFD

Laboratorium S.R.L. se dedica a la fabricación y distribución de medicamentos. La gerencia de marketing informa que en los últimos meses se detectó un aumento considerable en la venta de tranquilizantes especialmente en el nicho docente universitario, los especialistas lo atribuyen a los resultados de las encuestas docentes realizadas en los cursos de Análisis de Sistemas.

Todos los viernes los APM (Agentes de Propaganda Médica) visitan a los médicos programados para esa semana. El APM, desde su notebook ingresa, en forma remota, al sistema de Promociones con la especialidad del médico. Obtiene el detalle de los nuevos medicamentos de promoción para esa especialidad. Busca por apellido del médico en la planilla almacenada en su disco C (Medicos_por_Programa.xls) si ese médico ya está asociado al programa de beneficios con alguno de los medicamentos nuevos. Si no es así, le pregunta al médico si desea que sus pacientes ingresen al Programa de Beneficios. Si el médico está de acuerdo, el APM le entrega 5 tarjetas por cada medicamento seleccionado por el médico. Por cada tarjeta generada, se confecciona el comprobante Tarjetas Entregadas, donde figura el nombre del APM, su legajo, el medicamento (código y descripción), nombre y especialidad del médico. Este comprobante es firmado por el médico confirmando la recepción de la tarjeta, Además se genera una nueva línea en el archivo Excel con el médico, el detalle de ese medicamento, la fecha y el programa al que se asoció.

Cuando el paciente visita al médico, le explica que con la tarjeta (firmada y sellada) que le entrega, el 50 o/o de su tratamiento estará a cargo del laboratorio (por cada caja que el paciente compre, el laboratorio le entrega otra caja sin cargo). En ese momento el médico confecciona la receta por duplicado. El original se lo entrega al paciente junto a la tarjeta y, el duplicado lo archiva para, a fin de mes, enviarlos al laboratorio para participar en los sorteos bimestrales de Laboratorio.

Al terminar la visita al médico, el APM envía un mail al supervisor informando el resultado de la misma: fecha, material entregado, tarjetas entregadas, consultas a responder.

Cuando el paciente compra el medicamento llama al laboratorio. A través del IVR selecciona la opción Programa de Beneficios y una operadora le pide sus datos y los datos del médico. La operadora tiene que verificar que el médico esté en el programa (consultando Médicos por_Programa.xls). Si es correcto, se controla si el paciente ya fue dado de alta contra la base "Pacientes en Programa" (actualizada diariamente por Admisión). Si el paciente existe (ya sea por ese medicamento o por otro) le informan la fecha en que le entregarán el medicamento a cambio de la caja comprada. Si el paciente no existe, además de informarle la fecha, se le solicitan los datos identificatorios (nombre y apellido completo, número de documento, fecha de nacimiento, dirección y teléfono).

El último día hábil del mes, se asignan los premios a los APM por las ventas realizadas. Para ello, se toman los comprobantes de tarjetas entregadas y se comparan contra las ventas del mes (año, mes, producto y cantidad vendida), enviadas por el sector de ventas. Si se cumplen los objetivos mensuales de ventas, determinados por el director, se completo un formulario de asignación por triplicado. El original se envía al APM, el duplicado se entrega a comisiones y el triplicado se archiva.

2.6. Ejercicio 6

TICKETRONN

Objetivos: Armar el diagrama de contexto, definir los eventos y funciones principales del sistema y armar el DFD.

La venta de entradas en el sistema Ticketronn se realiza de la siguiente forma:

El cliente puede adquirir las entradas de distintas formas: en boneterías donde se lleva a cabo el espectáculo, en boleterías localizadas en sitios remotos, comercios autorizados...

Cuando se hace una solicitud, el sistema determina lugar y fecha del espectáculo (algunos clientes tienen información errónea sobre el mismo), Para esto se lleva un calendario maestro por la Gerencia de Promoción. Si corresponde, se provee al cliente de información, sobre diversos espectáculos en la misma fecha o en otra diferente.

Una vez que el cliente selecciona el evento y la fecha, el sistema le asigna los asientos, de acuerdo con un plano maestro. A medida que se otorgan los asientos, el plano maestro se actualiza para indicar la disponibilidad de los lugares.

Cuando el cliente confirma su acuerdo, el sistema habilita la cobranza. El precio de la entrada está en función del evento, de la hora, del lugar y de la localidad en la que se realiza el espectáculo. También varían de acuerdo al patrocinador del evento. Por ejemplo, es común que el precio de las entradas sea menor, en caso de que el cliente sea jubilado; de acuerdo al tipo de espectáculo también se realizan descuentos a estudiantes. Algunas entradas son distribuidas sin costo por la administración o los patrocinadores del evento, aún en estos casos solicitan las entradas a través del sistema. Las entradas cobradas y entregadas se registran en el Archivo Diario de Movimientos de Entradas.

Se registra información de cada una de las ventas y cobros para que Marketing lleve el control y la liquidación para los administradores y patrocinadores.

2.7. Ejercicio 7

COMUNICACIÓN

Objetivos: Armar el diagrama de contexto definir los eventos y funciones principales del sistema. Armar el DFD.

Diariamente, se verifica, en base al archivo FACTUMS del Sistema de Gestión Comercial, que existan comprobantes pendientes de cobro que hayan superado la fecha de segundo vencimiento (estipulada en un plazo de días posterior a la fecha del primer vencimiento).

Para aquellos clientes, a los cuales se les detectaron facturas en la condición mencionada, o que posean un monto total de la deuda acumulada, superior al límite de crédito asignado (indicado en el archivo maestro de CLIENTES), se dispara el proceso de morosidad.

La primera etapa del proceso de morosidad, es el aviso de incomunicación al cliente, que consiste en la emisión de cartas a los domicilios particulares de los clientes impagos.

En el caso de que no se reciba una respuesta en una semana, se pasa a la segunda etapa en la que se genera la orden de incomunicación. Dicha etapa, tiene como objetivo generar el archivo INCOMUNICA, con el detalle de la situación del cliente para que el sistema Técnico lleve a cabo la acción correspondiente. Se deben indicar los números de servicio del cliente afectados por la deuda, según el archivo SERVICIOS CLIENTE, también del Sistema Comercial.

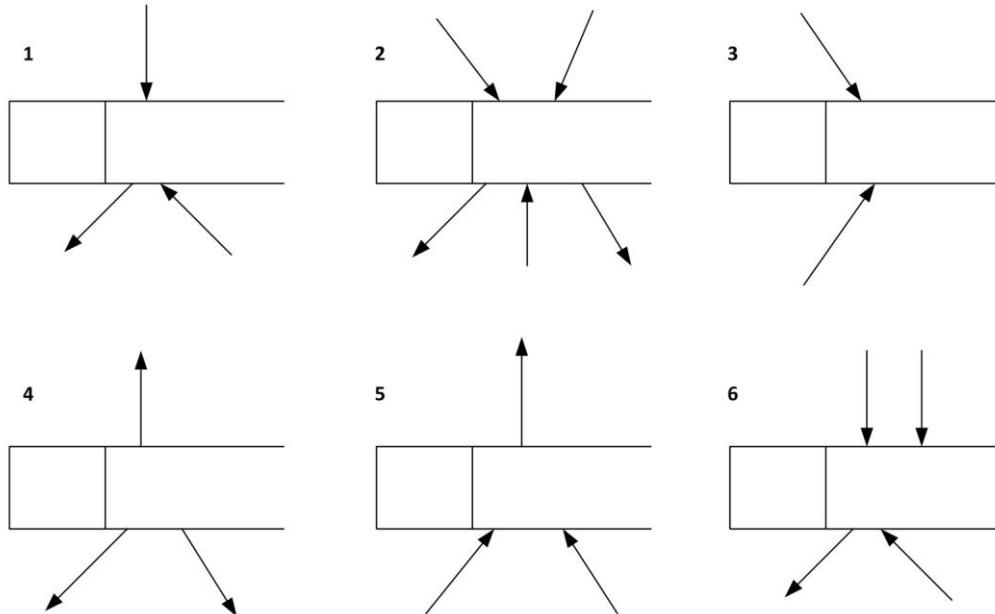
Una vez resuelta o cumplimentada la incomunicación, el Sistema Técnico le informará al Departamento de Clientes Morosos acerca de la ejecución de la misma y se guardará la información recibida en los archivos correspondientes.

Al recibirse la confirmación de la incomunicación realizada, se emite una carta para el cliente, informándole la futura baja del servicio, a menos que regularice su situación. Se envía una copia de la misma, al Departamento Legales para que quede como antecedente de las acciones tomadas.

2.8. Ejercicio 8

DEMORAS

Objetivo: Indique si las siguientes situaciones son factibles o no. En caso de ser factibles, indique bajo qué condiciones.



2.9. Ejercicio 9

MACMICKEY

Objetivos: Documente la información mediante DC, TE y DFD

La cadena de comidas rápidas "MacMickey" posee un sistema que se encarga de la atención de los clientes que lleguen con sus autos y quieran comer dentro de ellos. Se les entregará su pedido previo pago. Para poder acceder al servicio, el cliente se presenta con su auto en la primer ventanilla del local y es atendido por una persona (puede ser hombre si es de noche o mujer si es de día). Allí el cliente hace su pedido. puede pasar que el cliente, ya sea porque es su primera vez o porque quiera probar otros menús, le pida al empleado que lo atiende, recomendaciones o información de los menús que se ofrecen.

Cuando el cliente se decide y realiza el pedido, el empleado se comunica mediante un micrófono con la cocina (en la cual trabaja mucha gente) e informa el pedido, para que se realice (a veces lo tienen pre cocido y lo re-calientan). Una vez que el pedido lo tienen listo, se lo entregan al empleado de la ventanilla 3 para que lo entregue al cliente.

Una vez que el cliente finalizó su pedido en caja 1, el empleado le informa que se presente en la ventanilla siguiente y que allí realice su pago. El cliente avanza con su auto (a veces rápido, otras lento...pero eso no viene al caso) y al llegar a la ventanilla, el empleado le dice el importe total. El cliente saca de su billetera o de otro lado el dinero para pagar y se lo entrega al empleado. Este, lo registra en la máquina, actualiza los ingresos en un archivo que posee la empresa, imprime un ticket y se lo da al cliente, informándole también que se dirija a la tercera (y ultima ventanilla) para retirar el pedido.

Puede que el cliente sea un SuperCliente (ya que visita el local seguido). En ese caso, se le hace un descuento. Para ello, al momento de pagar, el cliente le deberá presentar una tarjeta (posee sus datos y puntos acumulados por sus compras). El empleado, consulta con un archivo de descuentos y efectúa el pago.

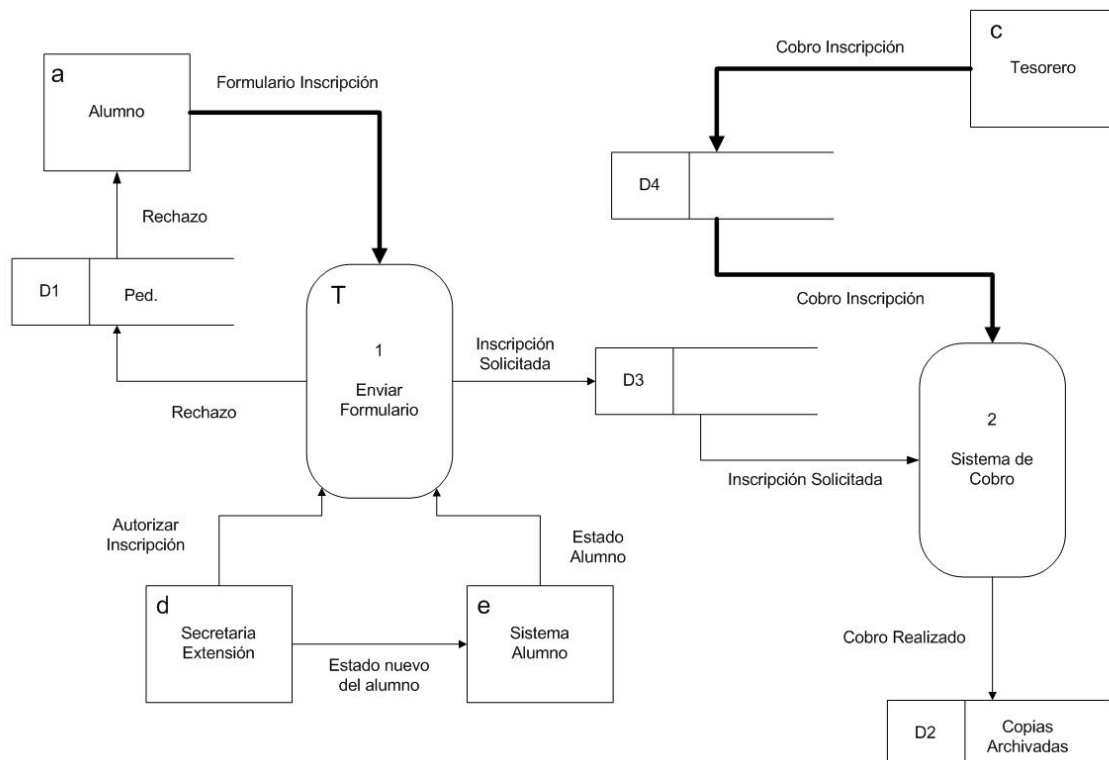
El cliente avanza nuevamente con su auto y al llegar, luego de que el cliente le entregara el ticket del pedido, el empleado nro3 (a veces puede que atienda el mismo que atendió en la ventanilla 2) entrega el pedido deseándole suerte en el viaje.

Mensualmente el local realiza estadísticas de los SuperClientes, y se las envía al Jefe del Local para que luego se comunique con el Jefe de todos los locales del país.

2.10. Ejercicio 10

ADMINISTRACIÓN DE ALUMNOS

Objetivos: Detectar al menos 5 errores en el siguiente DS.



2.11. Ejercicio 11

IMPORTAC

Objetivos: Armar el diagrama de contexto, definir los eventos y funciones principales del sistema y armar el DFD.

La tramitación necesaria para la comercialización de mercaderías importadas, es la siguiente:

La mercadería que entra al país es registrada por la Aduana, quien se ocupa de documentar sus condiciones de ingreso.

La DGI hace el control de su estadía en el país.

Cuando el importador decide comercializarla realiza el siguiente procedimiento: confecciona una solicitud de comercialización por triplicado. El original lo presenta en la Oficina de Comercio Internacional, el duplicado lo entrega en la Aduana y el triplicado lo guarda para él.

No bien puede la Oficina de Comercio Internacional le da curso, consulta los antecedentes que recibió de la DGI. Si son malos rechaza la solicitud, si son buenos consulta los de entrada al país que posee la Aduana. Si son malos rechaza la solicitud, si son buenos entrega al interesado un formulario de aranceles para que abone los derechos de comercialización.

Los derechos de comercialización se cobran en función de un porcentaje de acuerdo al tipo y cantidad de mercadería que se trate. El importador presenta el formulario de aranceles en Caja, el cajero completa la cifra a pagar, controla que el resto de los datos se haya completado, recibe el pago, sella y devuelve el original como constancia de pago y archiva el duplicado por fecha

Con las solicitudes aceptadas y rechazadas la Oficina de Comercio Internacional elabora trimestralmente un informe de resultados que envía al INDEC para hacer estadísticas respecto de la importación de mercaderías y poder establecer el déficit de la balanza comercial

2.12. Ejercicio 12

CIRC_EVENT

Objetivos: Armar el diagrama de contexto, definir los eventos y funciones principales del sistema y armar el DFD.

El Círculo del Personal Civil de la Fuerza Aérea organiza frecuentes eventos, ya sea por fechas patrias o por fechas específicas del Círculo o de la Fuerza Aérea.

Cada vez que se decide realizar un evento el Administrador envía un memorándum a la Comisión de Eventos. Según el tipo de evento, se verifica cual es su categoría y se prepara la lista de invitados según la categoría del evento, de acuerdo con lo dispuesto en uno de los Estatutos del Reglamento Interno del Círculo. En caso de duda respecto a algún invitado se consulta al Administrador y se espera su respuesta.

De la lista de invitados se verifica si la cuota de afiliado está al día o presenta signos de irregularidad, en cuyo caso no será invitado. Si todo está en regla se procede a emitir las participaciones que serán enviadas por correo al domicilio particular del socio.

Unos diez días antes del evento con las respuestas de los socios que fueron participados, se realiza una selección por grado militar, si se tratase de un evento exclusivo para personal civil se hace por categoría para determinar en qué sectores deberán ser ubicados y que protocolo de ceremonia se le asignará al evento. Se informará de la selección, la ubicación y el protocolo al Comandante de Personal.

Cada vez que un evento termina, el Comandante de Personal envía una lista de los asistentes reales, con esta lista se procede a realizar un informe para el Administrador, considerando entre otros, todas las invitaciones enviadas, las respuestas a las invitaciones, y quienes fueron seleccionados.

2.13. Ejercicio 13

RASCACIE

Objetivos: Armar el diagrama de contexto, definir los eventos y funciones principales del sistema y armar el DFD.

La empresa constructora “El Rascacielos” está construyendo edificios de lujo con departamentos de uno, dos, tres ambientes y semipisos.

Se desea estudiar el Sistema de Atención al Cliente.

Cuando un interesado llega hasta la obra, el encargado de la oficina de atención al público le comenta las características de los departamentos (de acuerdo a lo informado por el Departamento de Construcciones), los antecedentes de la empresa, y los requisitos para efectuar la compra; evitando mencionar inicialmente las características de aquellos departamentos que se encuentran reservados por cualquier motivo.

Si el comprador solicita algún departamento específico se le piden en ese momento los comprobantes de ingresos mensuales y si tiene alguna propiedad.

De acuerdo a los datos anteriores, el empleado decide el trámite que deberá seguirse, según corresponda. Para ello cuenta con una tabla cuyos valores actualiza semanalmente el Departamento de Ventas.

INGRESO MENSUAL	PROPIEDADES	TRAMITE A SEGUIR
menor a \$1000	NO	Rechazar solicitud
\$1000 a \$2000	SI	Ofrecer financiación
\$1000 a \$2000	NO	Rechazar solicitud
más de \$2000	SI	Ofrecer financiación
más de \$2000	NO	Pedir una garantía

Si se trata de pedir una garantía pasa el trámite al Departamento de Investigación de la empresa, quien verificará la garantía. En el caso de planes de financiación, completa un formulario y se pasa el trámite al Departamento de Finanzas quien informa a los interesados acerca de los planes posibles teniendo en cuenta sus ingresos, las características de mercado a esa fecha y un estudio a largo plazo de la economía del país.

En algún caso no previsto, el encargado consulta al Gerente de Ventas, quien resuelve en forma particular cual de las alternativas conviene seguir en ese caso, le informa al encargado que proceda según corresponda. En todos los casos se deja marcado con lápiz en la ficha correspondiente “reservado para tramitación”.

2.14. Ejercicio 14

SARGO

Objetivos: Armar el diagrama de contexto, definir los eventos y funciones principales del sistema y armar el DFD.

La empresa SARGO S.A. es líder en la comercialización de artículos de pirotecnia. Trabaja con clientes mayoristas y minoristas a los cuales brinda una atención diferenciada.

Los pedidos mayoristas se analizan apenas se reciben en la empresa. Se verifican primero los datos del cliente y si existen pedidos pendientes según la ficha enviada por Compras. Se suma el total del pedido y se controla que el crédito establecido por la División Cuentas Mayoristas no esté excedido; en ese caso se rechaza, caso contrario se revisa el stock actual (informado una vez por día desde Compras vía listado). Si el stock es insuficiente se llama telefónicamente al importador que responde en el instante.

El pedido se acepta solo si se puede cumplir con la entrega solicitada por el cliente. Se le responde siempre al cliente, se avisa de los pedidos aceptados a Compras y el total del pedido a Cuentas Mayoristas para que actualice el crédito del cliente.

Los pedidos minoristas se analizan una vez por día, controlando los datos del cliente, su situación crediticia informada por Cuentas Minoristas y el stock actual (solo se acepta si existe stock). El pedido puede ser en cuenta corriente o al contado. Si es en cuenta corriente y no alcanza su crédito se rechaza. Si es al contado y tiene deuda pendiente se consulta en el momento con Cuentas Minoristas para que tome una decisión, si no tiene deuda se avisa a Tesorería para que efectúe la cobranza.

Todos los pedidos aceptados van al fin del día a Compras y se envía por correo la respuesta a los clientes.

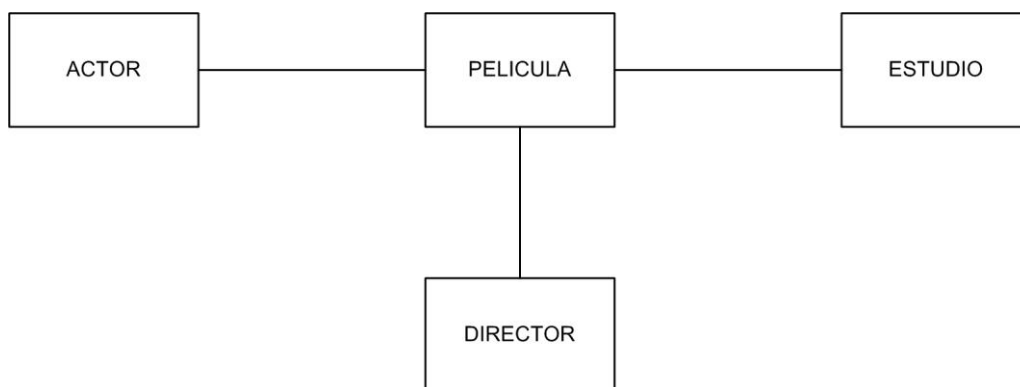
Guía de Trabajo práctico 3: Diagramas de Entidad Relación (DER)

3.15. Ejercicio 15

PELÍCULAS PARA COMPLETAR

Objetivos: Responda el siguiente cuestionario.

- 1) Defina Cardinalidad y Modalidad.
- 2) En función de las siguientes reglas, complete la Cardinalidad y modalidad del diagrama DER



- 3) Un actor puede o no estar filmando una película.
- 4) El Estudio siempre tiene en ejecución proyectos de películas
- 5) Las películas pueden ser realizadas por varios estudios
- 6) Los directores del estudio filman continuamente.

3.16. Ejercicio 16

PELUQUERÍA 1

Objetivos: Arme el DER que represente el siguiente enunciado.

Una de las más prestigiosas peluquerías de Bs. As, mantiene un registro de sus clientes identificando a cada uno por tipo y Nro. de documento. Entre los datos que se guardan figura el peluquero que lo atiende.

Los empleados de la peluquería son identificados por un código asignado por la misma peluquería, guardando además el nombre, la dirección y el teléfono entre otros datos. Existen distintos tipos de empleados: administrativos, peluqueros y auxiliares, guardando de cada uno de ellos datos particulares. De los auxiliares interesa saber qué cursos realizaron.

Cada vez que un cliente es atendido se registra en una ficha: Nro. de pedido, fecha de atención, horario del turno y el empleado administrativo que lo atendió.

Por cada pedido interesa saber, qué servicios se han prestado al cliente. La peluquería tiene un catálogo con los distintos servicios que presta; un servicio es, por ejemplo lavado de cabello, corte, peinado, permanente, etc.

De un servicio de un determinado pedido interesa saber los productos empleados, siendo un dato importante la cantidad utilizada de cada uno; la peluquería tiene un catálogo de los productos con que trabaja.

De un servicio de un determinado pedido puede participar, en algunas ocasiones, un auxiliar.

3.17. Ejercicio 17

BANCO

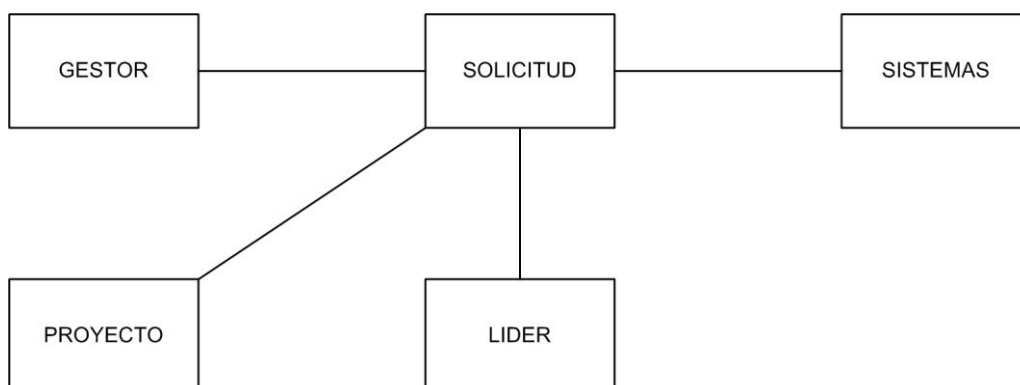
Objetivo: Armar el diagrama DER en función de las siguientes reglas. Como ayuda, se indican las entidades en negrita.

- El **banco** tiene distintas **sucursales** que se identifican por un código.
- Cada **sucursal** tiene una serie de **cuentas corrientes** asignadas a ella que se identifican por un código distinto para cada cuenta.
- Una **cuenta corriente** tiene asociados a ella a uno o varios **clientes**.
- Cada **cliente**, que se identifica por su DNI, puede tener varias cuentas y, por supuesto, unos **privilegios** distintos en cada una de ellas.
- Los **clientes** pueden tener otorgados **préstamos**, sin que estén asociados a ninguna **cuenta corriente**. Cada **préstamo** se otorga a nombre de un solo cliente, y a un **cliente** se le puede asignar más de un **préstamo**.

3.18. Ejercicio 18

PROYECTO

Objetivos: En función de las siguientes reglas complete la Cardinalidad y Modalidad de un diagrama DER



- a. Cada solicitud tiene dos gestores, uno de servicio y otro de demanda
- b. Todas las solicitudes tienen un único líder, pero cada líder puede liderar varias solicitudes
- c. Las solicitudes afectan o pueden afectar a uno o varios sistemas.
- d. Hay muchas solicitudes por cada sistema
- e. Las solicitudes pueden o no ser tratadas como un proyecto.

3.19. Ejercicio 19

RAYO

Objetivos: Grafique la siguiente observación mediante un DER

Transportes Rayo del Sur posee una flota de más de 300 micros de larga distancia. Realiza servicios diarios a distintas ciudades del país a cargo de su calificada dotación de conductores y azafatas. Todo **servicio** tiene asignado una pareja de **conductores**, y en el caso de los servicios diferenciales una **azafata**.

Los conductores intercambian el mando de la unidad cuando arriban a las ciudades cabeceras de cada recorrido, ya que un servicio puede recorrer varias ciudades, se guarda **registro** de la fecha y hora de llegada a la ciudad y del conductor asignado.

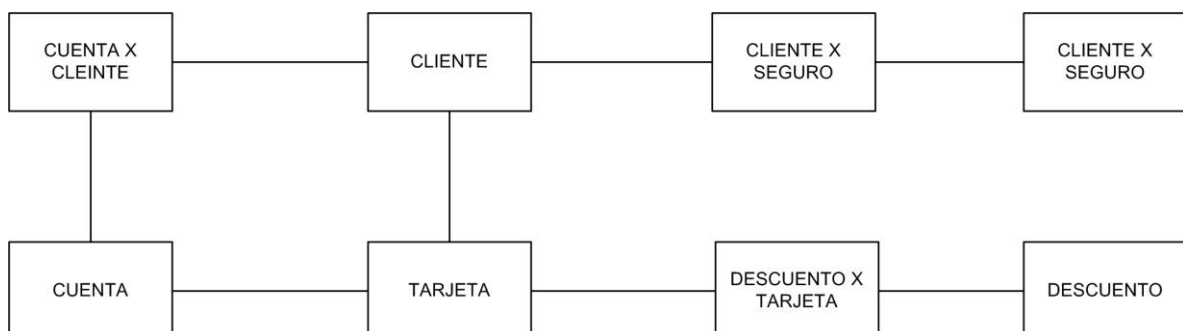
Los **micros** se clasifican en clase A (diferenciales), B (comunes) y C (coches camas), los primeros y los últimos realizan únicamente servicios diferenciales, mientras que las clases B sólo pueden cubrir servicios comunes ya que no cuentan con baño, ni TV, aire acondicionado ni bar.

3.20. Ejercicio 20

BANCO MAR

Objetivos: Complete Cardinalidad y Modalidad en el siguiente DER

Un cliente de banco MAR puede obtener tarjetas de débito, adquirir seguros (de vivienda, de accidentes, de autos y de vida) y abrir una cuenta, ya sea corriente en pesos o en dólares y caja de ahorro. Las tarjetas están a nombre de un solo titular, y cada tarjeta sirve para movilizar saldos de una cuenta en particular. También, con las tarjetas de banco MAR se obtienen descuentos en distintos comercios, dependiendo del tipo de tarjeta elegido. Las cuentas pueden estar a nombre de más de una persona, al igual que los seguros.



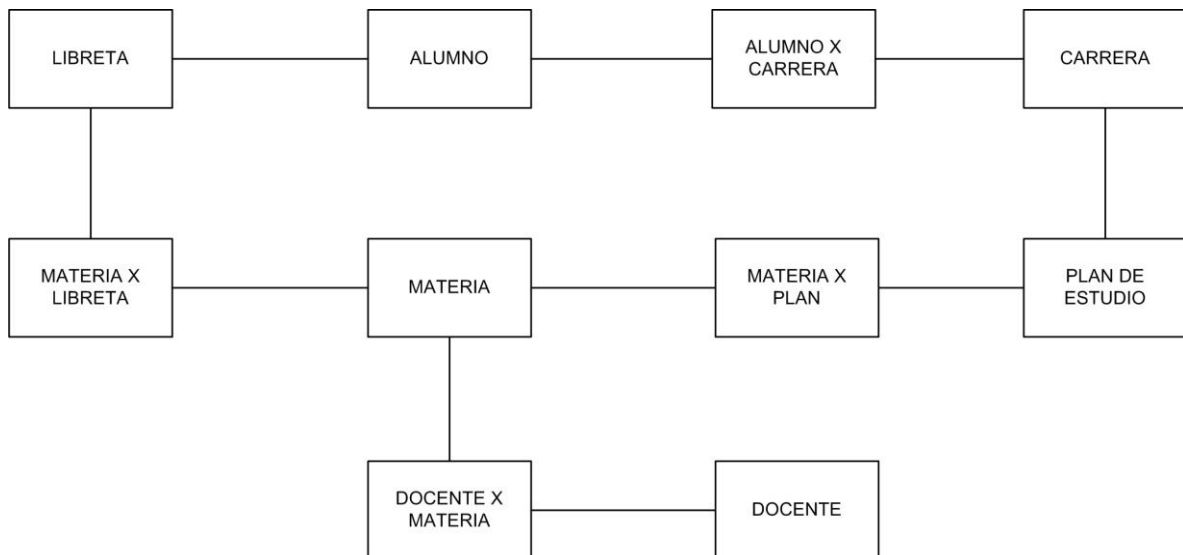
3.21. Ejercicio 21

ADMINISTRACIÓN DE ACTAS DE FINALES

Objetivos: Complete Cardinalidad y Modalidad en el siguiente DER. Realice aclaraciones si lo considera.

Si deseáramos realizar un sistema para la administración de actas de finales y de materias cursadas no podríamos dejar de lado que:

Un alumno de la facultad puede estar cursando varias carreras, y por eso puede tener varias libretas, una por cada carrera que cursa. Desde que es alumno de la facultad, posee libreta, ya sea transitoria o definitiva. Los planes de estudio de las carreras se identifican con un código único, y puede haber varios planes vigentes para la misma carrera. Cada plan se compone de sus materias correspondientes y docentes que las dictan. La libreta lleva un registro de las materias aprobadas y firmadas del alumno.



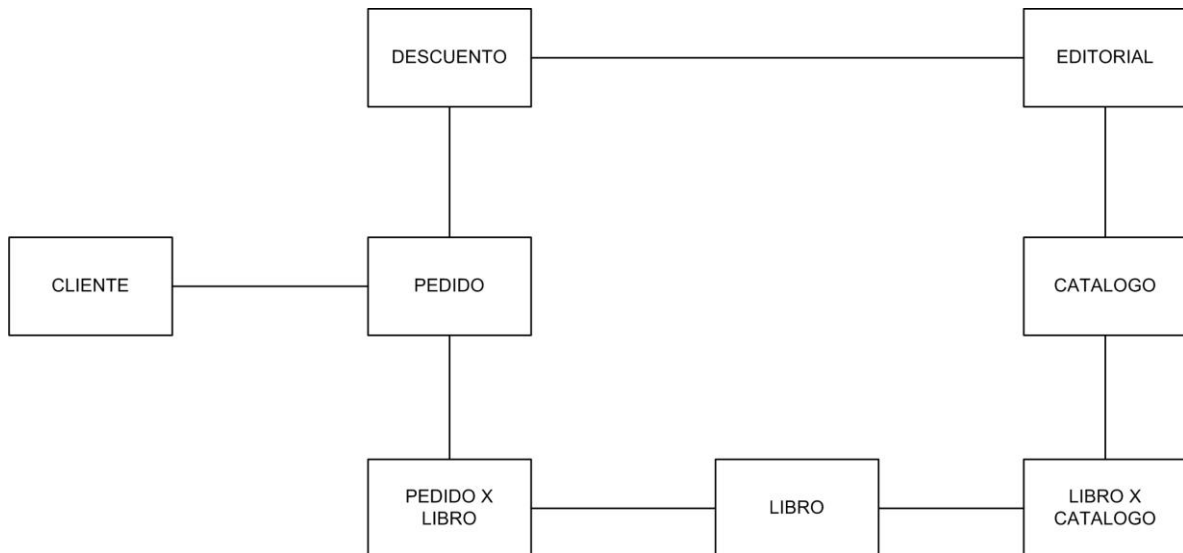
3.22. Ejercicio 22

GEROX

Objetivos: Complete Cardinalidad y Modalidad en el siguiente DER. Realice aclaraciones si lo considera necesario.

La empresa Gerox es la intermediaria en la distribución de libros de las distintas editoriales.

Generalmente los clientes son de escuelas o facultades que realizan un pedido de varios libros. Las editoriales tienen varios catálogos donde figuran los libros, y proponen distintos descuentos, que luego la empresa se encargará de decidir si se lo aplica o no al pedido correspondiente. Un pedido tiene un solo descuento. Un mismo libro puede aparecer en más de un catálogo.



3.23. Ejercicio 23

DER COMPLETO

Parte A

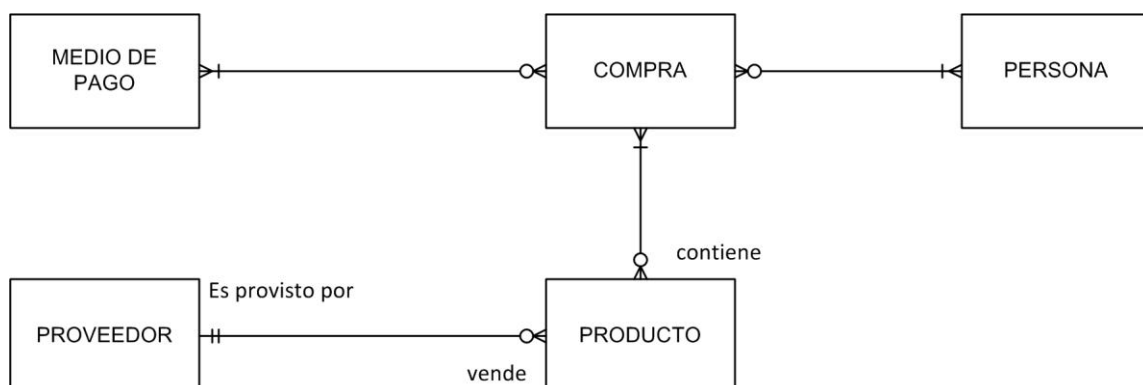
Objetivo: *Elabore el DER relacionado al siguiente enunciado. Algunas de las entidades necesarias se encuentran remarcadas en negrita*

Una empresa vende **productos** a varios clientes. Se necesitan conocer los datos personales de los clientes (nombre, apellidos, DNI, dirección y fecha de nacimiento), Cada producto tiene un nombre y un código, así como un precio unitario. Un cliente puede comprar varios productos a la empresa, y un mismo producto puede ser comprado por varios clientes.

Los productos son suministrados por diferentes **proveedores**. Se debe tener en cuenta que un producto sólo puede ser suministrado por un proveedor, y Que un proveedor puede suministrar diferentes productos. De cada proveedor se desea conocer el CUIL, nombre de la compañía, dirección y nombre del contacto.

Parte B

Objetivo: *Indique al menos 5 reglas que se pueden deducir del siguiente modelo DER y rearme el modelo para partir las relaciones Muchos a Muchos. Complete los nombres faltantes en las relaciones.*



3.24. Ejercicio 24

EDITORIALES

Objetivos:

Identificar las entidades componentes del sistema
Identificar las relaciones entre entidades
Identificar los atributos de las entidades
Armar el diagrama de Entidad-Relación

Ediciones CPU trabaja con muchos autores diferentes que escriben libros para la editora. Algunos autores escribieron un solo libro, mientras que muchos escribieron más de uno; además algunos libros han sido escritos por más de un autor.

La empresa trabaja con múltiples imprentas, algunas están asociadas por lo que reciben un porcentaje de las ventas de cada libro mientras que otras son independientes y cobran honorarios por cada trabajo realizado. Se maneja información distinta para cada tipo de imprenta aún antes de que impriman su primer libro, lo que le permite a la empresa contar con información de proveedores posibles y no sólo con los que ya imprimieron algún libro. Cada libro es asignado a una sola imprenta.

Un editor que trabaja para la compañía coordina varios proyectos a la vez, aunque cada proyecto tiene asignado un solo editor. En cada proyecto participan uno o varios autores y puede suceder en algún momento que un autor no esté trabajando en ningún proyecto.

3.25. Ejercicio 25

BUQUES

Objetivos:

Identificar las entidades componentes del sistema
Identificar las relaciones entre entidades
Identificar los atributos de las entidades
Armar el diagrama de Entidad-Relación

Una importante compañía naviera, la decimoquinta a nivel mundial, tiene una flota de más de 500 buques, todos identificados por un número de patente, nombre, la bandera del país, dimensiones (eslora, calado, tonelaje), velocidad crucero y una velocidad máxima.

Cada barco tiene un capitán que debe estar matriculado, se conoce su apellido y nombre y su nacionalidad.

La compañía naviera trabaja con dos tipos de clientes: empresas de combustible y líneas comerciales. De las primeras existe un catálogo internacional donde figuran la identificación, la razón social, sus teléfonos y el número de fax. De las segundas, cada país codifica sus líneas e informa sus nombres y números de fax.

Los buques pueden ser de varios tipos (cargueros de contenedores, tanques, cerealeros o de pasajeros). Los cargueros pueden o no estar equipados con un tipo de grúa (codificación internacional) y tienen una cantidad máxima de contenedores.

Los tanques son clasificados según el tipo de combustible que transportan (codificación de la compañía) y se conoce su capacidad en litros. Realizan viajes transportando combustible para distintas empresas, aunque en cada viaje lo hacen para una sola.

Los viajes se numeran secuencialmente y pueden incluir más de una parada en cada puerto, pagando en cada uno de ellos un costo por tonelada y por día, conociéndose además la fecha desde y hasta de cada parada.

La compañía numera los servicios que realizan sus cargueros. Cada servicio se factura por un total y se registra la línea comercial, el buque carguero y la cantidad de contenedores transportados.

Cada país numera sus puertos.

3.26. Ejercicio 26

BECAS RCI

Objetivos:

Identificar las entidades componentes del sistema
Identificar las relaciones entre entidades
Identificar los atributos de las entidades
Armar el diagrama de Entidad-Relación

Las Universidades extranjeras conceden un número de becas para cada uno de los cursos que dictan, por ello el Instituto de Orientación al Estudiante guarda registro de las becas disponibles en todo el mundo.

Cada beca tiene un código de identificación único dado por cada universidad extranjera y puede incluir varios cursos.

Cada universidad tiene un código de identificación único de carácter internacional, sus otros datos son: el país al que pertenece, el nombre de la universidad, los teléfonos y el domicilio de la universidad.

De los cursos interesa su código de identificación (asignado por cada universidad), el nombre del curso, la fecha de dictado, la beca ofrecida y el número de becas disponibles, el responsable del curso, el domicilio del responsable y todos sus títulos.

De cada beca se desea conocer el tipo de beca (total, parcial, individual, grupal), el nombre de la beca, el monto de la beca y las características que pueden variar entre 1 y 15 (están definidas a través de un código único y de una descripción).

3.27. Ejercicio 27

TALLER

Objetivos:

Identificar las entidades componentes del sistema
Identificar las relaciones entre entidades
Identificar los atributos de las entidades
Armar el diagrama de Entidad-Relación

El taller “Libertad” se dedica a la atención integral del automotor. Cuenta entre sus clientes a varias empresas de rubros como: reparto de soda, alimentos, correos privados y clearing bancario. Además atiende a clientes particulares, de los cuales sólo algunos pueden operar en cuenta corriente. Todos ellos deben dejar sus datos identificatorios (razón social para empresas y apellido y nombre para particulares, además, en todos los casos, dirección y hasta un máximo de tres teléfonos) y fundamentalmente impositivos (CUIT y situación ante el IVA). Las empresas informan un contacto y los días de pago.

Uno de los socios del taller es el encargado de la administración de las cuentas corrientes, para lo cual asigna el límite de crédito y la cantidad de días de plazo para calcular la fecha de vencimiento de las facturas. Se conoce en todo momento el saldo actualizado de cada cuenta corriente.

Por cada pedido que se atiende abren una ficha prenumerada donde anotan la fecha, el número de cliente, la patente del vehículo, el recepcionista y las observaciones que formula el cliente.

A medida que los distintos sectores participan en la atención del vehículo, anotan en la ficha su código de sector, la fecha y hora de entrada, la fecha y hora de salida, el responsable de la atención, el servicio realizado (código único que permite conocer la descripción) y los repuestos utilizados (están codificados, puede haber ninguno). Se puede pasar más de una vez por el mismo sector durante la atención de un pedido.

Con toda esta información, Marcela confecciona la factura cumpliendo con todas las normas de la DGI, agregando a la letra y número de factura: la fecha de emisión y de vencimiento, la forma de pago (contado o cuenta corriente) y el total de la factura (sumando los precios de los artículos). Ella decide como facturar los servicios y repuestos, ayudada por una tabla de artículos de facturación que incluye los precios. Los códigos de artículo no guardan relación directa con los servicios o repuestos. La factura siempre hace referencia a un único pedido.

3.28. Ejercicio 28

PELUQUERIA 2

Objetivos:

- Identificar las entidades componentes del sistema
- Identificar las relaciones entre entidades
- Identificar los atributos de las entidades
- Armar el diagrama de Entidad-Relación

Una de las más prestigiosas peluquerías de Bs. As., tiene las siguientes características: mantiene un registro de sus clientes identificando a cada uno por tipo y Nro. de documento; entre los datos que se guardan figura el peluquero que lo atiende.

Los empleados de la peluquería son identificados por un código asignado por la misma peluquería, guardando además el nombre, la dirección y el teléfono entre otros datos. Existen distintos tipos de empleados: administrativos, peluqueros y auxiliares, guardando de cada uno de ellos datos particulares. Por ejemplo, de los auxiliares interesa saber que cursos realizaron.

Cada vez que un cliente es atendido se registra en una ficha: Nro de pedido, fecha de atención, horario del turno y el empleado administrativo que lo atendió.

Por cada pedido interesa saber, qué servicios se han prestado al cliente. La peluquería tiene un catálogo con los distintos servicios que presta; un servicio es, por ejemplo: lavado de cabello, o corte, o peinado, o permanente, etc.

De un servicio de un determinado pedido interesa saber los productos empleados, siendo un dato importante la cantidad utilizada de cada uno; la peluquería tiene un catálogo de los productos con que trabaja. Además puede participar, en algunas ocasiones, un auxiliar.

Guía de Trabajo práctico 4: Diseño de Sistemas OO

4.29. Ejercicio 29

Para cada uno de los enunciados de los ejercicios 2.1 - 2.2 - 2.3 - 2.4 - 2.5 - 2.6 de la “guía de trabajo práctico 2: diagramas de flujo de datos (DFDs)”, armar: casos de uso, diagrama de casos de uso, diagramas de actividades (con calles), diagrama de secuencias, diagrama de clases y escenarios.