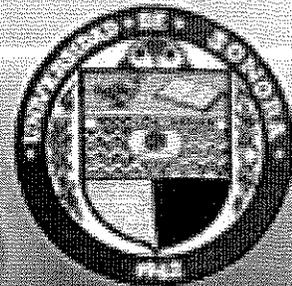


# Universidad de Sonora



**DIVISION DE INGENIERIAS  
DEPARTAMENTO DE INGENIERIA INDUSTRIAL  
INGENIERIA EN SISTEMAS DE INFORMACION**

Reporte técnico de prácticas profesionales

José David López Murillo

Empresa: Softtek Information Services S.A. De C.V.

Proyecto: Programa "Momentum"

Monterrey, Nuevo León a 06 de Diciembre de 2010

*Volvo*  
*14/12/2010*  
*[Signature]*

# Índice

1. Introducción.....	3
1.1 Pruebas de Software.....	4
1.2 Servicios de “costa cercana”.....	5
2. Pruebas de Software en Softtek.....	8
2.1 Alcance.....	9
2.2 Mapa de Implementación.....	10
3. Justificación.....	10
3.1 Aprovechamiento de la cercanía geográfica.....	10
3.2 Proximidad y zona horaria.....	10
3.3 Compatibilidad cultural y facilidad para hacer negocios.....	11
3.4 Ahorros en costos.....	11
3.5 Costo total del servicio.....	11
3.6 Ganancias en productividad.....	12
3.7 Global Nearshore™.....	12
3.8 Características.....	13
3.9 Llenar el hueco dejado por India.....	13
3.10 Sobresaliente experiencia con los clientes.....	13
3.11 Reducción de la complejidad en la administración de servicios de TI.....	13
3.12 Globalización de operaciones de TI y BPO.....	14
3.13 Optimización de costos.....	14
3.14 Apoyo a la evolución del negocio.....	15
4. Fundamento teórico de las herramientas y conocimientos aplicados.....	16
4.1 Casos de prueba.....	16
4.2 Estructura de los casos de prueba.....	17
4.3 Pruebas funcionales.....	18
4.4 Pruebas de validación.....	19
4.5 Caja negra.....	19
4.6 Pruebas de regresión.....	19

5. Objetivos.....	21
6. Limitaciones y desventajas del Testing automatizado.....	22
7. El ciclo de vida de pruebas.....	24
7.1 ¿En qué fase del ciclo de vida deben comenzar las pruebas?.....	24
7.2 Planificación de las pruebas (Fase de definición del Producto).....	24
7.3 Análisis de pruebas (Fase de Documentación).....	25
7.4 Diseño de pruebas.....	26
7.5 Ejecución de las pruebas.....	26
7.6 Ciclo de pruebas (Regresión).....	27
8. Conclusiones y recomendaciones.....	28
9. Recomendaciones.....	29
10. Bibliografía.....	30

## Introducción

El siguiente documento describe las actividades realizadas en Softtek con motivo del cumplimiento de las prácticas profesionales del Alumno José David López Murillo durante los meses de Septiembre, Octubre y Noviembre.

Softtek es un proveedor global de soluciones de TI y procesos de negocio, integrada por cerca de 6,000 colaboradores y 30 oficinas en Norteamérica, Latinoamérica, Europa y Asia. Cuenta con 9 Centros Globales de Entrega en México, China, Brasil, Argentina y España. Softtek ofrece soluciones redituables y de alta calidad a corporaciones líderes en más de 20 países, empleando modelos de entrega de servicios en sitio, *on-shore*, *offshore* o a través del modelo Global Nearshore™, marca registrada de Softtek. Creador y líder de la industria nearshore, Softtek es un proveedor de TI líder de Latinoamérica.

El programa "Momentum" de Softtek es un apoyo que permite trabajar en proyectos reales en una etapa temprana del desarrollo profesional de los alumnos de educación superior. Cada año miles de estudiantes en diferentes etapas de su educación, así como jóvenes egresados, ganan experiencia mientras se especializan en prácticas y tecnologías primordiales de Softtek; además de aprender técnicas de ingeniería de software e iniciativas de calidad como Six-Sigma.

Por motivos de cumplimiento del contrato de confidencialidad con el cliente no es posible describir de forma completa las actividades que fueron realizadas ni el nombre del cliente para el cual se trabajo.

Durante el periodo descrito se llevaron a cabo diversas tareas relacionadas a la planeación, desarrollo y ejecución de pruebas de las aplicaciones de una importante empresa de redes de almacenamiento estadounidense.

## Pruebas de Software

Las pruebas de software, en inglés *testing* son los procesos que permiten verificar y revelar la calidad de un producto software. Son utilizadas para identificar posibles fallos de implementación, calidad, o usabilidad de una aplicación computacional o videojuego.

Básicamente es una fase en el desarrollo de software consistente en probar las aplicaciones construidas.

Las pruebas de software se integran dentro de las diferentes fases del ciclo del software dentro de la Ingeniería de software. Así se ejecuta un programa y mediante técnicas experimentales se trata de descubrir que errores tiene.

## Servicios de “Costa cercana” en Softtek

*Softtek fue la primera empresa en forjar el concepto nearshore cuando comenzó a trabajar con sus primeros clientes en los Estados Unidos en 1997. Dicho concepto comenzó a ganar popularidad en 2002 cuando Softtek - junto con el gobierno mexicano – comenzó a promocionar el concepto con mayor agresividad.*

*El mercado identificó inicialmente el concepto nearshore con los servicios de TI provistos desde Canadá y México hacia clientes en Estados Unidos. Más adelante el uso del término evolucionó hasta describir todos los servicios de outsourcing provistos desde zonas cercanas geográficamente a la del cliente final (por ejemplo, describiendo los servicios desde países de Europa del Este hacia Europa Occidental). Algunos analistas lo describen actualmente como servicios que son entregados desde un país adyacente o cercano al destino final.*

*La propuesta de valor original de nearshore está basada en los siguientes beneficios al cliente:*

- *Proximidad y misma zona horaria*
- *Compatibilidad cultural y facilidad para hacer negocios*
- *Ahorros en costos*

*Estos beneficios representan la forma más elemental de la propuesta de valor nearshore, disponible para cualquier compañía que se establezca dentro de una ubicación nearshore, algo fácilmente justificable para profesionales de sourcing global experimentados. Es por ello que muchos proveedores se han “subido a la ola nearshore” validando, a su vez, la efectividad de la propuesta nearshore.*

*En Softtek creemos que estos elementos sirven como base para la construcción de una propuesta de valor sólida a usuarios de servicios globales, pero que sin metodologías y prácticas altamente maduras y de clase mundial, los beneficios distan mucho de su real potencial.*

*En 2002 Softtek introdujo otro concepto alrededor de la oferta de outsourcing, el “Costo Total de Iniciativa” o TCE (por “Total Cost of Engagement”), como una metodología para evaluar los gastos totales de las iniciativas offshore, mostrando la competitividad en costos*

de un modelo nearshore maduro, aún cuando se la compara contra modelos offshore altamente eficientes en costos.

Aunque las tarifas por hora de las ubicaciones nearshore tienden a ser ligeramente más elevadas que las de locaciones de más bajos salarios, el costo total estas ofertas termina siendo similar o menor que las de las iniciativas offshore, gracias a la eficiencia que se logra al trabajar en un destino cercano a los Estados Unidos y en la misma zona horaria. A través del uso de un proceso maduro y disciplinado, el modelo Near Shore® es mucho más eficiente en el logro de porcentajes altos de trabajo desde la ubicación económica respecto a los modelos offshore. Cabe señalar que es necesario un proceso maduro y disciplinado para lograr altos niveles de apalancamiento nearshore.

Desde el primer momento nos dimos cuenta que la competitividad de costos del modelo Near Shore® dependía de tener una gran parte del equipo trabajando en la ubicación nearshore. No obstante, para poder hacerlo posible, tuvimos que apoyarnos en fuertes procesos, así como en una rigurosa disciplina. Desde diciembre de 2007, hemos mantenido un promedio por encima del 90% de trabajo desempeñado en el sitio de menor costo, lo que representa enormes ahorros para nuestros clientes.

El arbitraje de salarios, en forma aislada, sólo permite ahorros limitados en costos a los clientes. Nuestra experiencia muestra que las compañías que realizan contratos a largo plazo regulados por SLAs (Acuerdos de Niveles de Servicio), son los que realmente aprovechan de la mejor manera las capacidades que el modelo Near Shore® puede brindar. Para 2004, Softek ya había migrado la mayoría de sus clientes bajo contratos basados en esquemas de Tiempo y Materiales hacia esquemas basados en SLAs. Esto representó una enorme ventaja para directores de IT y sus empleadores, ya que el valor de TI comenzó a ser medido por resultados más que por comparaciones de puntualidad o tarifas de hora/hombre.

En 2007, Softek bosquejó su visión de la evolución del modelo Near Shore® hacia un modelo de entrega verdaderamente global, con la publicación del white paper "Nearshore 2.0: Nearshore Goes Global", en el cual establecimos que las compañías nearshore podrán "abrir ubicaciones globales fuera de la 'zona de confort nearshore'". Esto significa la expansión no sólo dentro de los mercados primarios, sino también la apertura de centros de entrega en lugares del mundo que ofrezcan una mayor variedad de talento y mayor escala. En agosto del mismo año, Softek adquirió I.T. UNITED, proveedor líder de desarrollo de

*software y servicios de outsourcing con operaciones en Beijing, Shanghai, Xiamen y Xi'an en China. Este gran paso permitió a Softtek expandir su zona geográfica de servicios globales y complementar su cobertura en Europa y América con el componente asiático.*

*En julio de 2009, Softtek abrió un nuevo GDC en la ciudad de Wuxi, China. Esto representa el noveno GDC de Softtek en total: 4 en México, 2 en China, 1 en Brasil, 1 en Argentina y 1 en España. Estos centros permiten a la compañía proveer soluciones de Global Nearshore™ así como soluciones "follow-the-sun" completas para servicios relacionados a aplicaciones, infraestructura de TI y BPO. La nueva oficina en China tiene el objetivo de fortalecer aún más su posicionamiento de competitividad en la industria global de sourcing a medida que la industria china sigue creciendo.*

*Tras más de una década de historia de Near Shore®, Softtek ha evolucionado su propuesta de valor: partiendo de la entrega de servicios desde ubicaciones cercanas a las de los grandes usuarios de servicios, la propuesta evolucionó hasta conformar una oferta que ayuda a clientes grandes y medianos a globalizar sus operaciones TI, a reducir la complejidad de la administración de servicios, a optimizar los costos de operación y a evolucionar los negocios de los clientes.*

Fuente: [www.softtek.com.mx](http://www.softtek.com.mx)

## Pruebas de Software en Softtek

Las aplicaciones de software actuales son cada vez más complejas: a menudo operan en ambientes multi-capas, operan bajo múltiples plataformas; son diseñadas en condiciones rápidas y ágiles, y con requerimientos más ambiciosos que nunca. En este escenario, el software testing se ha tornado un elemento crítico para asegurar la calidad y minimizar el riesgo, al mismo tiempo que se atienden las exigencias de la organización y del mercado. En este contexto, Softtek cuenta con una práctica de testing amplia y robusta, que permite identificar la exactitud, integridad y calidad de los productos de software. Softtek eleva la productividad e incrementa la comunicación con los equipos de desarrollo, a la vez que logra ahorros significativos.

Cumplir los objetivos de negocio:

- Incremente la satisfacción de sus usuarios.
- Mejore las aplicaciones y disminuya el re-trabajo de sus equipos.
- Optimice el rendimiento de su equipo de calidad, en procesos de negocio enfocados al cliente.
- Reduzca los tiempos de entrega y optimice recursos.
- Disminuya el ciclo de vida del proyecto, interactuando en tiempo real.
- Realice esfuerzos de testing paralelos (manual, manual/automático, perfil de aplicación).
  - Cuente con un modelo integral para el ciclo de vida de testing.

## **Alcance**

Ya sea que requiera testing a la medida, automático, o tan solo definir el perfil de una aplicación, nuestro modelo de implementación cumplirá con cada requerimiento de su ciclo de vida de testing:

- Test de Planeación: Pruebe estrategias, objetivos y enfoques.
- Test de Análisis & Diseño: analice requerimientos funcionales, tome decisiones automatizadas, diseñe casos y ambientes de prueba.
- Test del Entorno de Negocios: Instale hardware, software, herramientas de prueba y realice un “Smoke Testing”.
- Test de Implementación: Desarrolle scripts y cree datos de prueba
- Test de Ejecución & Reportes: Ejecute casos y scripts de prueba. Evalúe reportes y métricas. Realice una administración de defectos.
- Test de Conclusión de Proyecto: aceptación del proyecto, entrega de testware (software testing), análisis postmortem.
- Administración de Proyecto: control y monitoreo de los procesos de prueba, administración general de las iniciativas de prueba
- Configuración & Administración del Cambio: control de versiones, manejo de código fuente, administración del cambio y configuración de los elementos.
- Aseguramiento de la calidad: proyectos Six Sigma.

## **Mapa de implementación**

El proceso Softtek Software Testing Process es una metodología integral, que funciona tanto para el ciclo de vida completo de pruebas, como para proyectos multi-fase que requieren de pruebas específicas. Nuestro servicio se cimienta en una cultura de cero-defectos y una actitud de constancia exhaustiva hasta alcanzar el resultado deseado. Esto es posible porque nos impulsan dos factores determinantes: la calidad y el generar un producto listo para salir al mercado.

## **Justificación**

### **Aprovechamiento de la cercanía geográfica:**

Esta fue nuestra primera onda; la idea de este concepto fue llenar el hueco dejado por los servicios de entrega global centrados en India. Este concepto permanece como un sólido diferenciador de Softtek, aunque esto es sólo el inicio de nuestra aventura en los servicios globales.

Este componente de la propuesta de valor comprende lo siguiente:

### **Proximidad y Zona Horaria**

La proximidad geográfica y operar en la misma zona-horaria del cliente permitieron a las compañías incrementar la interacción cara-a-cara, permitiendo la externalización de proyectos más complejos a través del modelo Nearshore. Esto permitió llenar el hueco creado por los servicios prestados desde ubicaciones de offshore lejanas.

## **Compatibilidad Cultural y Facilidad para Hacer Negocios**

Debido a la cercanía, la mayoría de las ubicaciones Nearshore tienen una mayor compatibilidad cultural con sus mercados primarios que las ubicaciones offshore. Los mexicanos, por ejemplo, están más familiarizados con los estilos de vida, costumbres y estilos de comunicación de EE.UU. Además, debido al TLC, México no sólo ha sido un fabricante y proveedor fundamental de servicios para los EE.UU., sino que es un mercado importante para los negocios de dicho país. Este intercambio comercial ha crecido más allá de la familiaridad entre dos culturas, ayudando a minimizar problemas de comunicación causados por diferencias culturales. Además, gracias al TLC, los problemas con visas son virtualmente inexistentes, ya que los mexicanos pueden obtener TN visas (renovables anualmente) fácilmente.

## **Ahorros en Costos**

Dependiendo de la ubicación, los ahorros de costos pueden ser iguales al los de la ubicación offshore. Por ejemplo, México puede ofrecer ahorros de costos considerables a clientes estadounidenses, comparables con los de India, ya que los costos indirectos - tales como costos de contratos, de auditoría, de comunicación y de viajes - son más económicos.

## **Costo Total del Servicio:**

El Costo Total del Servicio, o TCE (por sus siglas en inglés), es una metodología utilizada para evaluar los gastos totales de los servicios offshore, ilustrando la competitividad en costos de los modelos nearshore maduros, incluso cuando se los compara contra modelos offshore altamente eficientes en costos.

Aunque las tarifas por hora nearshore tienden a ser más altas, el costo total de los servicios Near Shore® de Softtek termina siendo equivalente o inferior al de las alternativas offshore, gracias a la eficiencia lograda a través de la proximidad con Estados Unidos y a la operación en zonas horarias similares.

Cabe señalar que un proceso maduro y disciplinado es necesario para lograr tales niveles de apalancamiento nearshore.

Usando un modelo de alta calidad, construimos la infraestructura necesaria para apoyar el trabajo remoto, que incluso contando los beneficios de una ubicación nearshore, representa un gran reto. El modelo de calidad soporta nuestra capacidad de lograr un muy alto apalancamiento del componente nearshore (trabajo desempeñado en la instalación más económica).

### **Ganancias en Productividad:**

Al combinar el maduro modelo de calidad, procesos digitalizados y la experiencia tecnológica y vertical, somos capaces de ofrecer a nuestros clientes servicios basados en SLA (Acuerdos de Niveles de Servicio). Esto representa una enorme ventaja para líderes de reclutamiento y sus empleadores puesto que el valor pasa a ser medido por resultados más que por horarios de entrada o por la comparación de tarifas hora/hombre.

Con el uso de la metodología Six-Sigma, ofrecemos a nuestros clientes SLAs evolutivos, que se traduce en el compromiso de nuestra parte de hacer más con menos, además de incrementar la productividad, reducir costos, minimizar defectos, dar un mejor soporte, entregar resultados más rápido, o cualquier otra combinación de beneficios a nuestros clientes, en relaciones a largo plazo.

### **Global Nearshore™**

Nuestro compromiso de hoy es reducir la complejidad de la administración de servicios de TI, optimizar costos y apoyar la globalización de las operaciones de TI y BPO. De esta manera, contribuimos a la evolución de los negocios de nuestros clientes.

Lo anterior es posible gracias a nuestra capacidad de proporcionar servicios de clase mundial eficientes en costos, a nuestro enfoque en entender las necesidades del mercado - muchas veces ignorado por los proveedores tradicionales - y a nuestro compromiso por proporcionar una excelente experiencia a todos nuestros clientes.

## **Características**

Servicios eficientes en costos de clase mundial

Desempeño medible y en constante evolución; el Costo Total del Servicio más económico.

## **Llenar el hueco dejado por India**

Creemos que un proveedor de soluciones debe encontrar formas de hacer más fácil la vida de sus clientes, mientras se continúa proporcionando los mejores servicios del mercado.

Nuestros clientes necesitan soluciones personalizadas específicas y servicios de calidad de un proveedor de alto desempeño. Por esto creamos el Modelo Near Shore®, como una forma de llenar el hueco que deja el modelo de entrega global de servicios basado en India. Al apalancar el Modelo Global Nearshore™ de Softtek, las corporaciones pueden satisfacer sus necesidades de alta interactividad, rápida respuesta al cambio, diversificación de riesgos, baja rotación de personal y apoyo multilingüe; elementos críticos que el modelo con base en la India no puede resolver en forma adecuada.

## **Sobresaliente experiencia de nuestros clientes**

El valor de negocio se mide por el precio, la calidad y los plazos de entrega de los productos finales. El valor de la satisfacción del cliente, por otro lado, se enfoca en resolver las necesidades, requerimientos e intereses de la gente – individuos – que interactúan con el proveedor. Ambos elementos juegan un papel importante en el éxito del sourcing global. Nuestro compromiso es incrementar el valor de negocio y de la experiencia en forma simultánea.

## **Reducción de la Complejidad en la Administración de Servicios de TI**

- Consolidar el portafolio de aplicaciones nuevas y legadas en único equipo de trabajo
- Asegurar que el comportamiento de las aplicaciones críticas sea predecible y su operación, ininterrumpida
- Reducir la multiplicidad de proveedores y apalancar las fortalezas de un socio de negocios sólido y flexible, con alcance global
- Monitorear los acuerdos de niveles de servicios, y no los horarios de entrada y salida

## **Globalización de Operaciones de TI y BPO**

- Incrementar la capacidad de competir en la economía mundial, apoyándose en un socio de negocios global
- Operar bajo estándares de clase mundial a nivel local, regional y global, sin importar la naturaleza de sus necesidades
- Recibir el servicio desde la ubicación geográfica que haga más sentido para su negocio, no para sus vendedores
- Apoyarse en el exhaustivo conocimiento de Softek de las peculiaridades de los mercados de Norteamérica y Latinoamérica. Soportar las operaciones europeas y asiáticas de su empresa a través del modelo Near Shore®
- Aprovechar la “cercanía global”. Administrar las operaciones del día, de día, sin importar en qué parte del mundo se requiera esa modalidad
- Mejorar el tiempo de respuesta a sus necesidades teniendo siempre cerca a su equipo de trabajo

## **Optimización de Costos**

- Reducir el Costo Total del Servicio, reducir costos indirectos y apalancar ubicaciones con menores costos
- Optimizar procesos, consolidar funciones, reducir defectos y evolucionar de forma constante los Acuerdos de Nivel de Servicio
- Habilitar a su gente para que alcance su más alto potencial; realizar la tarea correcta desde el lugar apropiado
- Optimizar el uso de la tecnología; acelerar nuevas implementaciones de tecnología asegurando su efectividad

## **Apoyar la Evolución del Negocio**

- Apoyar iniciativas de globalización
- Acelerar la implementación, personalización y puesta en marcha de las aplicaciones de su empresa con el mayor ROI posible
- Construir una plataforma para el futuro. Implementar aplicaciones sólidas y los procesos necesarios para soportarlas
- Maximizar el conocimiento del negocio. Asegurar los recursos apropiados para las tareas de negocio de valor. Extraer la lógica de negocios contenida en sus aplicaciones legadas

# Fundamento teórico de las herramientas y conocimientos aplicados

## Casos de prueba

En la Ingeniería del software, los casos de PICO o Test Case son un conjunto de condiciones o variables bajo las cuáles el analista determinará si el requisito de una aplicación es parcial o completamente satisfactorio.

Se pueden realizar muchos casos de prueba para determinar que un requisito es completamente satisfactorio. Con el propósito de comprobar que todos los requisitos de una aplicación son revisados, debe haber al menos un caso de prueba para cada requisito a menos que un requisito tenga requisitos secundarios. En ese caso, cada requisito secundario deberá tener por lo menos un caso de prueba. Algunas metodologías como RUP recomiendan el crear por lo menos dos casos de prueba para cada requisito. Uno de ellos debe realizar la prueba positiva de los requisitos y el otro debe realizar la prueba negativa.

Si la aplicación es creada sin requisitos formales, entonces los casos de prueba se escriben basados en la operación normal de programas de una clase similar.

Lo que caracteriza un escrito formal de caso de prueba es que hay una *entrada conocida* y una *salida esperada*, los cuales son formulados antes de que se ejecute la prueba. La *entrada conocida* debe probar una precondición y la *salida esperada* debe probar una poscondición.

Bajo circunstancias especiales, podría haber la necesidad de ejecutar la prueba, producir resultados, y luego un equipo de expertos evaluaría si los resultados se pueden considerar como "Correctos". Esto sucede a menudo en la determinación del número del rendimiento de productos nuevos. La primera prueba se toma como línea base para los subsecuentes ciclos de pruebas/lanzamiento del producto.

Los casos de prueba escritos, incluyen una descripción de la funcionalidad que se probará, la cuál es tomada ya sea de los requisitos o de los casos de uso, y la preparación requerida para asegurarse de que la prueba pueda ser dirigida.

Los casos de prueba escritos se recogen generalmente en una suite de pruebas.

Las variaciones de los casos de prueba son comúnmente utilizadas en pruebas de aceptación. La prueba de aceptación es realizada por un grupo de usuarios finales o los clientes del sistema, para asegurarse que el sistema desarrollado cumple sus requisitos. La prueba de aceptación de usuario se distingue generalmente por la incorporación de un trayecto feliz o casos de prueba positivos.

### **Estructura de los casos de prueba**

Formalmente, los casos de prueba escritos consisten principalmente en tres partes con subdivisiones:

- Introducción/visión general contiene información general acerca de los Casos de Prueba.
  - Identificador es un identificador único para futuras referencias, por ejemplo, mientras se describe un defecto encontrado.
  - Caso de prueba dueño/creador es el nombre del analista o diseñador de pruebas, quien ha desarrollado pruebas o es responsable de su desarrollo.
  - Versión la actual definición del caso de prueba.
  - Nombre el caso de prueba debe ser un título entendible por personas, para la fácil comprensión del propósito del caso de prueba y su campo de aplicación.
  - Identificador de requerimientos el cual está incluido por el caso de prueba. También aquí puede ser identificador de casos de uso o especificación funcional.
  - Propósito contiene una breve descripción del propósito de la prueba, y la funcionalidad que chequea.
  - Dependencias

- Actividades de los casos de prueba
  - Ambiente de prueba/configuración contiene información acerca de la configuración del hardware o software en el cuál se ejecutará el caso de prueba.
  - Inicialización describe acciones, que deben ser ejecutadas antes de que los casos de prueba se hayan inicializado. Por ejemplo, debemos abrir algún archivo.
  - Finalización describe acciones, que deben ser ejecutadas después de realizado el caso de prueba. Por ejemplo si el caso de prueba estropea la base de datos, el analista debe restaurarla antes de que otro caso de prueba sea ejecutado.
  - Acciones pasos a realizar para completar la prueba.
  - Descripción de los datos de entrada
- Resultados
  - Resultados esperados contiene una descripción de lo que el analista debería ver tras haber completado todos los pasos de la prueba
  - Resultados reales contienen una breve descripción de lo que el analista encuentra después de que los pasos de prueba se hayan completado. Esto se sustituye a menudo con un Correcto/Fallido. Si un caso de prueba falla, frecuentemente la referencia al defecto implicado se debe enumerar en esta columna.

## **Pruebas funcionales**

Una prueba funcional es una prueba basada en la ejecución, revisión y retroalimentación de las funcionalidades previamente diseñadas para el software. Las pruebas funcionales se hacen mediante el diseño de modelos de prueba que buscan evaluar cada una de las opciones con las que cuenta el paquete informático.

## Pruebas de validación

Las pruebas de validación en la ingeniería de software son el proceso de revisión que el sistema de software producido cumple con las especificaciones y que cumple su cometido. Es normalmente una parte del proceso de pruebas de software de un proyecto, que también utiliza técnicas tales como evaluaciones, inspecciones, y tutoriales. La validación es el proceso de comprobar lo que se ha especificado es lo que el usuario realmente quería.

Se trata de evaluar el sistema o parte de este durante o al final del desarrollo para determinar si satisface los requisitos iniciales. La pregunta a realizarse es: ¿Es esto lo que el cliente quiere?.

## Caja negra

En teoría de sistemas y física, se denomina caja negra a aquel elemento que es estudiado desde el punto de vista de las entradas que recibe y las salidas o respuestas que produce, sin tener en cuenta su funcionamiento interno. En otras palabras, de una *caja negra* nos interesará su forma de interactuar con el medio que le rodea (en ocasiones, otros elementos que también podrían ser *cajas negras*) entendiendo qué es lo que hace, pero sin dar importancia a cómo lo hace. Por tanto, de una *caja negra* deben estar muy bien definidas sus entradas y salidas, es decir, su interfaz; en cambio, no se precisa definir ni conocer los detalles internos de su funcionamiento.

## Pruebas de regresión

Se denominan Pruebas de regresión a cualquier tipo de pruebas de software que intentan descubrir las causas de nuevos errores (bugs), carencias de funcionalidad, o divergencias funcionales con respecto al comportamiento esperado del software, inducidos por cambios recientemente realizados en partes de la aplicación que anteriormente al citado cambio no eran propensas a este tipo de error. Esto implica que el error tratado se reproduce como consecuencia inesperada del citado cambio en el programa.

Este tipo de cambio puede ser debido a prácticas no adecuadas de control de versiones, falta de consideración acerca del ámbito o contexto de producción final y extensibilidad del error que fue corregido (fragilidad de la corrección), o simplemente una consecuencia del rediseño de la aplicación.

Por lo tanto, en la mayoría de las situaciones del desarrollo de software se considera una buena práctica que cuando se localiza y corrige un bug, se grabe una prueba que exponga el bug y se vuelvan a probar regularmente después de los cambios subsiguientes que experimente el programa.

Existen herramientas de software que permiten detectar este tipo de errores de manera parcial o totalmente automatizada, la práctica habitual en programación extrema es que este tipo de pruebas se ejecuten en cada uno de los pasos del *ciclo de vida del desarrollo del software*.

## Objetivos

Si bien podemos pensar en que los objetivos del testing son triviales pueden también diferir mucho de una organización a otra.

Estos son algunos de los objetivos:

1. Testing consistente y repetible
2. Correr casos de prueba desatendidos
3. Encontrar errores de regresión
4. Correr casos de prueba más seguidos
5. Mejorar la calidad del software
6. Testear más profundo
7. Mejorar la confianza en la calidad del software
8. Testear más software
9. Incrementar la confianza de los usuarios
10. Medir performance
11. Reducir el costo del testing
12. Encontrar más errores
13. Testear en diferentes sistemas operativos
14. Disminuir el tiempo de salida al mercado/correr las pruebas más rápido
15. Mejorar la moral en los testers
16. Testear en diferentes bases de datos

## **Limitaciones y desventajas del Testing automatizado**

El testing automatizado lejos de ser una fórmula perfecta es una actividad de constantes mejora y constantes problemáticas por el manejo humano de la configuración de las mismas, a continuación se mencionan los principales problemas del testing automatico:

### **Falta de lineamientos de desarrollo**

El concepto se refiere a el software desarrollado con normas de calidad pero sin lineamientos orientados a las pruebas del mismo. Sin embargo ay que tener en cuenta que no se debe desarrollar funcionalidades extras o incluir requerimientos no solicitados por el cliente, solo se requiere formar una base de lineamientos que permitan hacer más rápido y eficiente el desarrollo de los scripts automatizados, como por ejemplo, no incluir elementos personalizados que no sean detectados por las herramientas de automatización o en el caso del desarrollo web, que todos los principales elementos de la pantalla contengan un ID que los identifique de los demás.

### **Falta de conocimientos de los integrantes del equipo en codificación o en la herramienta**

Al igual que al contratar desarrolladores juniors el riesgo de que la aplicación contenga errores o no se entregue a tiempo aumenta, con las pruebas automatizadas sucede lo mismo, ya que hay que tener en claro que las pruebas automatizadas son un desarrollo y para ello se necesita de personal capacitado. Este factor acarrea a otro factor, el que desarrolladores con pobres conocimientos generen scripts con código pobre, lo cual lo hace difícil de mantener y de muy baja performance. Pensar que cualquiera puede automatizar una prueba es un error que puede resultar muy costoso ya que las pruebas automatizadas requieren de habilidades de desarrollo de software. El esfuerzo de automatización sólo tiene éxito si se aplican mediante el una metodología comprobada.

## **Falta de conocimiento del equipo en el concepto de Pruebas Automatizadas**

Que el equipo desconozca la diferencias entre el testing manual y el testing automatizado conlleva a que se toman malas decisiones (intentar automatizar el 100% de las pruebas manuales es la primera) y la relación costo/beneficio se vuelva en contra del equipo.

## **Demostración del ROI**

Este puede ser el más discutible de los factores y aplica principalmente para aquellas organizaciones que no realizan pruebas automatizadas y necesitan demostrar los beneficios de la misma en relación con la inversión al momento de agregar este entre sus servicios. Como toda nueva práctica, su aplicación inicial va a requerir una inversión que no se verá en el corto plazo pero si al mediano plazo, esto puede poner muy nerviosos a los managers y comerciales atentando contra las pruebas con sus pedidos de achicar tiempos y gastos.

## **Falta de administración de los datos de prueba**

Al igual que para el testing manual, los datos de prueba son un pilar importante en las pruebas automatizadas. Hay que tener en cuenta que una buena administración de datos de prueba es la base de la automatización con éxito, ya que al no hacer un seguimiento de los datos con los que corremos las pruebas o los datos que estas generan puede provocar un gran número de reportes de errores los cuales se produzcan por datos corruptos en la base de datos y no por la aplicación bajo pruebas.

## **Tratar las Pruebas Automatizadas como una actividad secundaria**

Es importante que las pruebas automatizadas no se traten como una actividad secundaria, es decir, pedir a un tester automatizar determinados casos de prueba cada vez que tiene tiempo libre. Los testers rara vez tienen tiempo libre y los plazos son siempre inminentes. Las pruebas automatizadas requieren un ciclo de vida de mini-desarrollo con los requisitos de pruebas, diseño de las pruebas, la implementación y verificación de las pruebas.

## El ciclo de vida de pruebas

Normalmente, el testing está considerado como parte del Ciclo de Vida de Desarrollo de cualquier aplicación (Requisitos, análisis & diseño, implementación, testing y desarrollo), pero el testing, a su vez, tiene su propio ciclo de vida. Dependiendo de la organización, este ciclo puede tener más o menos fases, pero a continuación comentaré las fases que son siempre comunes.

A grandes rasgos, podemos decir que el ciclo de vida del Software Testing incluye las siguientes fases:

1. Planificación.
2. Análisis
3. Diseño
4. Ejecución
5. Ciclos
6. Pruebas Finales e Implementación
7. Producción

### ¿En qué fase del ciclo de vida deben comenzar las pruebas?

Lo antes posible. Si dividimos el ciclo de desarrollo en cuatro etapas: Inicio, Elaboración, Construcción y Transición... el equipo de Quality Assurance debería comenzar sus labores entre la fase de Inicio y Elaboración. Es recomendable que al menos el responsable del equipo de calidad asista a las reuniones de toma de requisitos.

### 1. Planificación de las pruebas (Fase de definición del Producto):

La fase de planificación de pruebas significa redactar el Test Plan ó Plan de Pruebas. El Test Plan es un documento a alto nivel que describe las estrategias a seguir durante el desarrollo de las pruebas.

Contenidos de un Test Plan:

- + Alcance de las pruebas.
- + Comienzo y fin (planificación)
- + Estrategias (Black Box, White Box, etc.)
- + Niveles de Pruebas (Integration testing, Regression testing, etc.)
- + Limitaciones
- + Riesgos
- + Revisiones y entregables.
- + Técnicas de Pruebas (Boundary Value Analysis, Equivalence Partitioning, etc.)

Herramientas para realizar las pruebas (Automáticas, carga, integración co

- + Reportes
- + Hitos
- + Métricas
- + Recursos y planes de formación
- + Tareas y responsabilidades

Estas son sólo algunos de los contenidos de un Plan de pruebas aunque muchas veces varía dependiendo de la empresa. Hay algunas plantillas interesantes en internet, como la de la IEEE.

## **2. Análisis de pruebas (Fase de Documentación)**

La fase de análisis es una extensión de la fase de la planificación. Mientras que la fase de planificación es a más alto nivel, la fase de análisis es donde se comienza a documentar los planes detallados. Se comienza a analizar los casos de prueba:

- + Revisión de los Inputs: Se consideran, el documento de especificación de requisito y otros documentos de planificación del proyecto...mientras, el plan de pruebas se rompe en pequeñas partes que serán los futuros casos de prueba.
- + Formatos: Generalmente en esta fase se crea una matriz de validación basada en los requisitos. Estas matrices nos ayudarán a la hora de ejecutar los test cases. También, usando alguna de las herramientas de planificación de proyectos que hay en el mercado (phpcollab, MS project, gantt...) creamos la planificación de las pruebas basándonos en cada

uno de los hitos del plan de proyecto. Las métricas a utilizar, también se diseñan en esta fase.

+ Test Cases: ó casos de prueba. Siguiendo la matriz de validación y otros documentos de entrada (inputs), se escriben los test cases. Comenzamos también a vincular los requisitos con cada test case.

+ Automatización: Mientras se crean test cases, se identifican los casos que deben ser automatizados. También se identifican las áreas para las pruebas de rendimiento, de carga y de stress.

+ Plan de Regresión: Los ciclos de pruebas, es decir, número de veces que se probará de nuevo la aplicación para verificar que los defectos encontrados no han introducido nuevos errores.

### **3. Diseño de pruebas:**

Tenemos que darnos cuenta que el ciclo de vida de pruebas transcurre en paralelo al de desarrollo. Así, una vez que hayamos llegado a esta fase (el equipo de desarrollo probablemente habrá comenzado a escribir las primeras líneas de código) comenzaremos el diseño.

Durante esta fase todos los planes, test cases, etc... de la fase de análisis son revisados y finalizados. E

Nuevos test cases pueden ser añadidos y se realiza el documento de gestión de riesgos. También es el momento para hacer los scripts de pruebas automáticas (nuevos ó updates de los ya existentes). Finalmente, los informes de pruebas, especialmente los informes sobre pruebas unitarias.

### **4. Ejecución de las pruebas:**

En este momento el equipo de desarrollo tiene una versión estable y lista para testear. Lo más recomendable es hacer un test de cualificación de la versión para evaluar que la aplicación es lo suficientemente estable para comenzar la ejecución de las pruebas. De nada sirve comenzar una ronda de test si al cabo de cinco golpes de ratón la aplicación rompe. Tiene que tener un mínimo de estabilidad.

Una vez terminado el reporte de resultados, los ingenieros de pruebas ya pueden comenzar a ejecutar el plan de test. Así como los test automáticos (si los hubiera), pruebas unitarias, revisiones de código, performance....etc.

Para llevar el seguimiento de la ejecución de los test cases podemos utilizar alguna de las herramientas comerciales que hay en el mercado ó utilizar una matriz de excel donde iremos apuntado nuestros PASSED o FAILED (OK or NOT).

### **5. Ciclo de pruebas (Regresión):**

En este momento del ciclo al menos alguna ronda de test ya se habrá ejecutado y algunos problemas se habrán reportado. Una vez que el equipo de desarrollo ha arreglado los defectos, la segunda ronda de pruebas comienza. Esta nueva ronda de pruebas podría solamente centrarse en las áreas ó funcionalidades que han sido arregladas. Pero también podría ser regresión testing, donde la aplicación es testeada nuevamente para verificar su correcto funcionamiento y que el arreglo de los defectos no ha afectado otras partes del código.

Pruebas → Reporte de Defectos → Arreglo de Defectos (y mejoras) → Nueva ronda de Pruebas

Aquí es donde pruebas automáticas pueden ser de gran ayuda para repetir el mismo test case una y otra vez.

Durante esta fase es importante revisar el documento de requisitos (CRD) y el Project Plan por si ha habido modificaciones.

## **Conclusiones y recomendaciones**

Sofftek es una empresa cuyos esfuerzos estratégicos han demostrado ser los mejores del área tecnológica en toda Latinoamérica y cuyas metodologías han sido aprobadas y usadas por un gran número de organizaciones con los mismos fines.

Las técnicas utilizadas no solo son parte de estándares internacionales sino que muchas de ellas empiezan a crear tendencias en el rubro tecnológicas y que nacieron en esta organización.

La oportunidad de realizar el trabajo de prácticas profesionales en esta organización es una gran oportunidad para cualquier estudiante de TI, ya que en Sofftek se manejan prácticamente todas las áreas de las tecnologías de la información, tanto en su forma técnica como estratégica.

Mi recomendación a la organización es únicamente seguir con los esfuerzos de mejora continua en todos los ámbitos y una contante alineación de sus objetivos estratégicos con sus prácticas reales.

## Retroalimentación

Definitivamente la formación de la Universidad de Sonora fue una fuerte herramienta para el buen desempeño en la organización, básicamente la mejor arma que recibí de la Unisón fue la habilidad de análisis que desarrolle en gran parte gracias a las asignaturas formativas de los últimos semestres.

Los conocimientos técnicos si bien son importantes no son un diferencial considerable a tomar en cuenta por empresas que contratan a recién egresados de nivel superior. Las capacidades analíticas por otra parte son fundamentales en la práctica de desarrollo de pruebas de software.

En softtek se imparte una academia de "testing" para recién egresados en la cual se ahonda en temas referentes a este tópico, temas que en la carrera de Ingeniería en Sistemas de Información de la Universidad de Sonora se ven pobremente y sin la opción de ampliar el conocimiento mediante una asignatura optativa especializante.

Mi recomendación a la Unisón y particularmente a la carrera de ISI sería el dar más opciones especializantes en otros rubros como son las pruebas de software.

## Bibliografía

SofttekSTAFF <En World Wide Web:  
<http://www.softtek.com/en/mexico>> [Nov 2010]

Usuario WIKI <En World Wide Web:  
[http://es.wikipedia.org/wiki/Pruebas de software](http://es.wikipedia.org/wiki/Pruebas_de_software)> [4 Oct 2010]

Atom <En World Wide Web:  
<http://www2.gxtechnical.com/portal/hqxpp001.aspx?15,1,53,O,S,0,PAG;CONC;3;19;D;25342;1;PAG;MNU;E;25;2;13;10;MNU;,>> [24 Ene 2010]

Floría Cortés <En World Wide Web:  
<http://www.sidar.org/recur/desdi/traduc/es/visitable/testing.htm>> [Feb 2010]

Usuario WIKI <En World Wide Web:  
[http://es.wikipedia.org/wiki/Red de %C3%A1rea de almacenamiento](http://es.wikipedia.org/wiki/Red_de_%C3%A1rea_de_almacenamiento)> [7 Dic 2010]