



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

**“ETAPA DE PRUEBAS DENTRO DEL PROCESO DE
DESARROLLO DE SOFTWARE DE LOS SISTEMAS EN
EL INSTITUTO NACIONAL ELECTORAL”**

TRABAJO ESCRITO

**EN LA MODALIDAD DE INFORME DEL
EJERCICIO PROFESIONAL PARA OBTENER
EL TÍTULO DE:
INGENIERO EN COMPUTACIÓN**

P R E S E N T A:

LORENA GRANJA GARCÍA



ASESOR DE TESIS: M. EN C. MARCELO PÉREZ MEDEL

MÉXICO, 2015.



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Contenido

| | |
|--|----|
| Índice de figuras | 3 |
| CAPÍTULO 1 ANTECEDENTES DEL INE | 4 |
| CAPÍTULO 2 ANTECEDENTES DE PRUEBAS DE SOFTWARE | 13 |
| Calidad de software (ISO/IEC 9126) | 14 |
| Técnicas de pruebas..... | 16 |
| Tipos de pruebas | 17 |
| Proceso de pruebas (IEEE 829) | 25 |
| CAPÍTULO 3 PROCESO DE PRUEBAS DE LOS SISTEMAS EN EL INE | 28 |
| Personal involucrado en el proceso de pruebas | 26 |
| Proceso de calidad..... | 33 |
| Elaborar el plan de pruebas | 34 |
| Especificar el ambiente de pruebas | 34 |
| Diseñar las pruebas | 35 |
| Ejecutar las pruebas | 38 |
| Informar avance del plan de pruebas..... | 39 |
| Solicitar pruebas de seguridad..... | 40 |
| Liberar a producción | 42 |
| Conclusiones | 44 |
| Bibliografía | 45 |

Índice de figuras

| | |
|--|----|
| <i>Figura 1.1</i> Consejo general del INE | 7 |
| <i>Figura 1.2</i> Organigrama general del INE | 9 |
| <i>Figura 1.3</i> Organigrama Secretaría Ejecutiva | 13 |
| <i>Figura 2.1</i> Técnicas de pruebas | 17 |
| <i>Figura 2.2</i> Pruebas de caja negra..... | 18 |
| <i>Figura 3.1</i> Proceso de calidad | 34 |
| <i>Figura 3.2</i> Plan de calidad | 35 |
| <i>Figura 3.3</i> Especificación de ambiente | 36 |
| <i>Figura 3.4</i> Caso de componente..... | 37 |
| <i>Figura 3.5</i> Caso de prueba de negocio | 37 |
| <i>Figura 3.6</i> Caso de prueba de base de datos | 38 |
| <i>Figura 3.7</i> Registro de defectos | 39 |
| <i>Figura 3.8</i> Informe del estado de plan de calidad..... | 40 |
| <i>Figura 3.9</i> Liberación de calidad..... | 41 |
| <i>Figura 3.10</i> Solicitud de pruebas de seguridad | 42 |
| <i>Figura 3.11</i> Reporte de liberación de seguridad | 43 |

INTRODUCCIÓN

En la actualidad han tomado mucha relevancia realizar pruebas a los sistemas de información más aun tener un proceso completo que abarque todas las etapas del desarrollo de software ya que se ha comprobado que cuando más pruebas se realicen desde las primeras etapas nos llevara a entregar un producto con mayor calidad, reducir los riesgos y satisfacer los requerimientos del usuario.

En el Instituto Nacional Electora (INE) el proceso de pruebas a los sistemas que en él se desarrollan ha ido evolucionando a la par en que se han desarrollado las nuevas metodologías sobre las mejores prácticas y ha ido adaptándolas de acuerdo a sus necesidades. Cuando termina cada proceso electoral se realiza un análisis y evaluación de todo el proceso de desarrollo de sistemas, las experiencias vividas en todo ese año para poder encontrar opciones de mejora que nos permita ser más eficientes y eficaces.

Después del proceso electoral del 2011-2012 y de la reestructuración a los procesos las actividades del área de calidad fueron más específicas y estuvieron más definidas en cada etapa de desarrollo, el presente trabajo pretende describir las actividades que se realizan en el proceso de pruebas dentro del desarrollo de los sistemas en el (INE), el personal involucrado la relación con las diferentes áreas y la importancia que tienen dentro del Instituto.

En el primer capítulo se describe la historia de manera breve desde el antes llamado Instituto Federal Electoral al actual el INE, su estructura y funciones principales. En el segundo capítulo se describen los conceptos más importantes de la calidad de software y que son base para las pruebas de software. En el tercer capítulo se explica cómo es el proceso de pruebas, quienes están involucrados en cada etapa, los insumos se requiere para poder desarrollar cada una y cuál es el producto al terminar de dichas etapas. Al finalizar se tienen las conclusiones personales sobre el trabajo realizado.

CAPÍTULO 1

ANTECEDENTES DEL

INSTITUTO NACIONAL

ELECTORAL

En este capítulo se hace resumen de cómo nació en Instituto Nacional Electoral (INE), su estructura y mi historia en dicha Institución.

Instituto Nacional Electoral

La historia del Instituto Nacional Electoral inicio después del proceso electoral de 1988 y los conflictos postelectorales que le siguieron la entonces Comisión Federal Electoral dio paso en

1990 como resultado de las reformas realizadas a la Constitución en materia electoral, el Congreso de la Unión expidió el Código Federal de Instituciones y Procedimientos Electorales (COFIPE) y ordena la creación del Instituto Federal Electoral (IFE), a fin de contar con una institución imparcial que dé certeza, transparencia y legalidad a las elecciones federales. INE. (2015). Historia.

<http://www.ine.mx/archivos3/portal/historico/contenido/menuitem.cdd858023b32d5b7787e6910d08600a0>

Al momento de su fundación, el Consejo General, máximo órgano de dirección del IFE, estaba compuesto por los siguientes funcionarios:

- El Presidente del Consejo General, que era el Secretario de Gobernación.
- Seis Consejeros Magistrados, personalidades sin filiación partidista con una sólida formación académica y profesional en el campo de derecho, propuestos por el Presidente de la República y aprobados por las dos terceras partes de la Cámara de Diputados.
- El Director y el Secretario General del Instituto.
- Dos diputados y dos senadores (representantes de los dos grupos parlamentarios más numerosos en cada Cámara).
- Un número variable de representantes partidistas que se fijaba de acuerdo con los resultados que obtuvieran en la última elección.

Sus principales atribuciones fueron:

- Fortalecer la confianza y la credibilidad de la ciudadanía en las elecciones federales.
- Regular el acceso de los partidos políticos y las autoridades electorales a los medios de comunicación.
- Promover la participación ciudadana en las elecciones.
- Asegurar condiciones de equidad y civilidad en las campañas electorales.

- Transparentar el proceso de organización y difusión de los resultados electorales.

La reforma constitucional en materia política-electoral, publicada el 10 de febrero de 2014 rediseñó el régimen electoral mexicano y transformó el Instituto Federal Electoral (IFE) en una autoridad de carácter nacional: el Instituto Nacional Electoral (INE), a fin de homologar los estándares con los que se organizan los procesos electorales federales y locales para garantizar altos niveles de calidad en nuestra democracia electoral.

Además de organizar los procesos electorales federales, el INE se coordina con los organismos electorales locales para la organización de los comicios en las entidades federativas.

El Consejo General del INE se compone de 11 ciudadanos elegidos por la Cámara de Diputados. Uno de ellos funge como Consejero Presidente y los 10 restantes como Consejeros Electorales. Ver Fig. 1.1

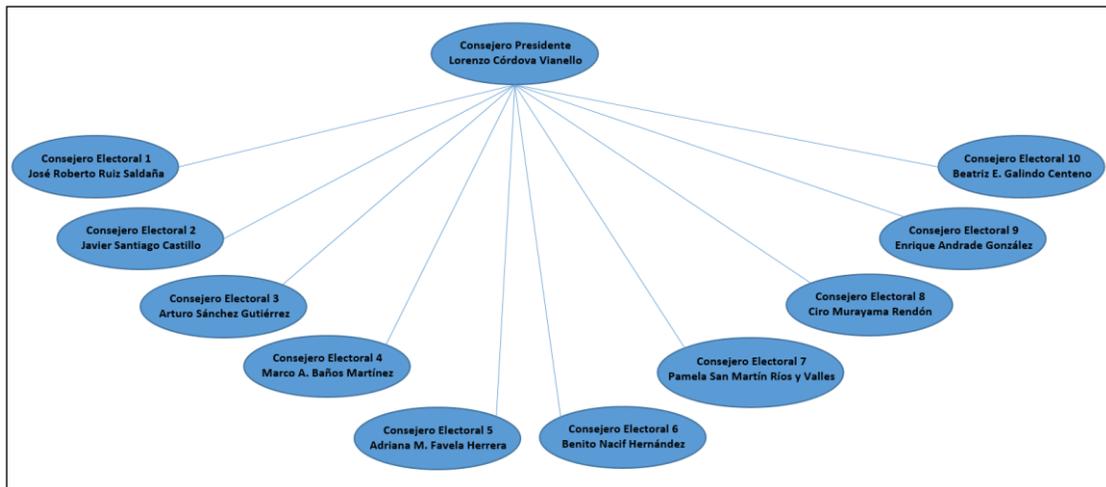


Figura 1.1 Consejo general.
Fuente: Elaboración propia (2015)

El Consejo General del INE designa a los consejeros de los organismos electorales locales y puede asumir funciones que le corresponden a dichos institutos en los casos que la Ley prevea.

De acuerdo con la reforma constitucional, entre las funciones principales del INE se encuentran las siguientes:

- Organizar la elección de los dirigentes de los partidos políticos a petición de estas organizaciones.
- Garantizar que los candidatos independientes tengan acceso a tiempos del Estado en radio y televisión, para que puedan difundir sus campañas.
- Verificar que se cumpla el requisito mínimo (2% de la lista nominal) para solicitar el ejercicio de las consultas populares y realizará las actividades necesarias para su organización, incluido el cómputo y la declaración de resultados.
- Fiscalizar los recursos de los partidos políticos nivel federal y local en forma expedita, es decir, en el transcurso de las campañas y no una vez que terminen

Organismos Delegacionales

En cada una de las entidades federativas, el Instituto contará con una delegación integrada por:

La Junta Local Ejecutiva y Juntas Distritales Ejecutivas.

El Vocal Ejecutivo

El Consejo Local o el Consejo Distrital, según corresponda, de forma temporal durante el proceso electoral federal.

Los órganos mencionados en el párrafo anterior tendrán su sede en el Distrito Federal y en cada una de las capitales de los Estados.

Juntas Distritales Ejecutivas. En cada uno de los 300 distritos electorales el Instituto contará con los siguientes órganos:

La Junta Distrital Ejecutiva.

El Vocal Ejecutivo y

El Consejo Distrital.

Los órganos distritales tendrán su sede en la cabecera de cada uno de los distritos electorales. INE. (2015). Órganos delegacionales.

http://www.ine.mx/archivos3/portal/historico/contenido/Organos_Delegacionales/

En la Fig. 1.2 se puede ver el organigrama de general del INE, los órganos centrales que lo componen el consejo general la secretaria ejecutiva, las direcciones ejecutivas y las unidades técnicas, así como los órganos desconcentrados como son las Junta Locales y Distritales.

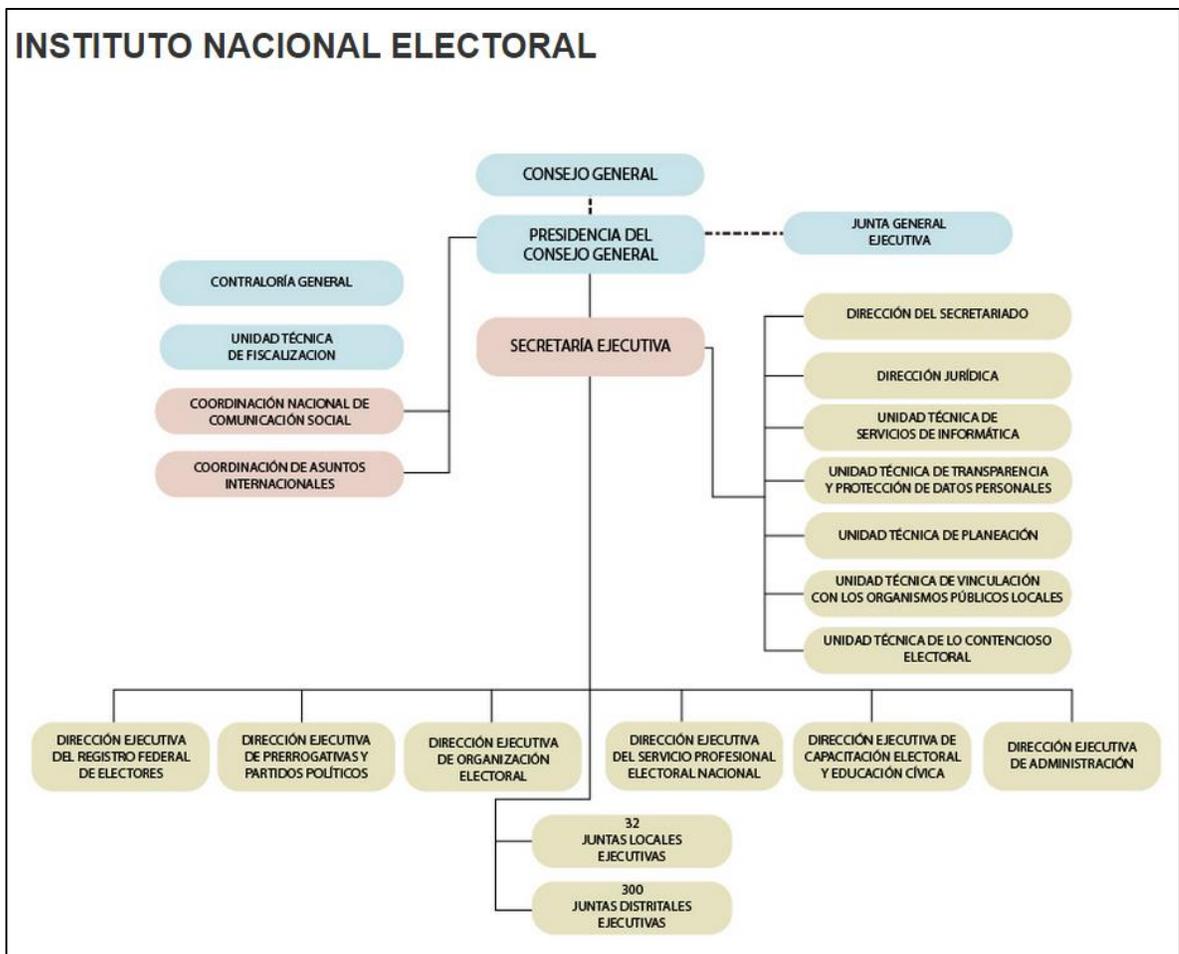


Figura 1.2 Organigrama general.

Fuente: INE (2015). Directorio Institucional.

<https://directorio.ine.mx/chartByAreaOrganigrama.ife?idArea=996>

Organismos Públicos Locales (OPL)

Son los encargados de la organización de las elecciones en su entidad federativa para la designación de:

- Gobernadores
- Diputados locales
- Presidentes municipales
- Integrantes de ayuntamientos (Síndicos y Regidores)
- Jefes delegacionales
- Jefe de gobierno
- Juntas municipales
- Gobiernos municipales

La coordinación de actividades entre el Instituto y los Organismos Públicos Locales estará a cargo de la Comisión de Vinculación con los Organismos Públicos Locales y del Consejero Presidente de cada Organismo Público Local, a través de la Unidad Técnica de Vinculación con los Organismos Públicos Locales, en los términos previstos en esta Ley.

Para la realización de las funciones electorales que directamente le corresponde ejercer al Instituto en los procesos electorales locales, de conformidad con las disposiciones establecidas en la Constitución y en esta Ley, y en concordancia con los criterios, lineamientos, acuerdos y normas que emita el Consejo General del Instituto, la Secretaría Ejecutiva del Instituto presentará a consideración del Consejo General, el proyecto de Plan Integral que contenga los mecanismos de coordinación para cada proceso electoral local.

A solicitud expresa de un Organismo Público Local, el Instituto asumirá la organización integral del proceso electoral correspondiente, con base en el convenio que celebren, en el que se establecerá de manera fehaciente las circunstancias de tiempo, modo y lugar que justifique la solicitud. LEGIPE (2015). Artículo 119, Título cuarto de la Ley General de Instituciones y Procedimientos Electorales

Secretaria Ejecutiva

Algunas de las atribuciones de la Secretaria Ejecutiva de acuerdo al artículo 51 de la Ley General de Instituciones y Procedimientos Electorales (LEGIPE):

- Aprobar la estructura de las direcciones ejecutivas, vocalías y demás órganos del Instituto conforme a las necesidades del servicio y los recursos presupuestales autorizados
- Proveer a los órganos del Instituto de los elementos necesarios para el cumplimiento de sus funciones.
- Preparar, para la aprobación del Consejo General, el proyecto de calendario integral de los procesos electorales ordinarios, así como de elecciones extraordinarias, que se sujetará a la convocatoria respectiva.

De acuerdo al artículo 41 del Reglamento Interior del Instituto Nacional Electoral:

- Analizar y, en su caso, aprobar la estructura de los órganos centrales y delegaciones, del Instituto, con excepción de aquellos que sean competencia exclusiva del Consejo, conforme a las necesidades del servicio que justifiquen los titulares de cada órgano y el dictamen de recursos presupuestales que formule la Dirección Ejecutiva de Administración.

- Coordinar el desarrollo de las actividades del Programa de Resultados Electorales Preliminares.
- Coordinar las acciones necesarias a efecto de integrar el Plan y Calendario Integrales de los procesos electorales ordinarios y, en su caso, el Calendario de las elecciones extraordinarias; así como disponer lo necesario para su aprobación por el Consejo.

De la Secretaría ejecutiva dependen las Unidades Técnicas, entre ellas la Unidad Técnica de Servicios de Informática, ver figura 1.3

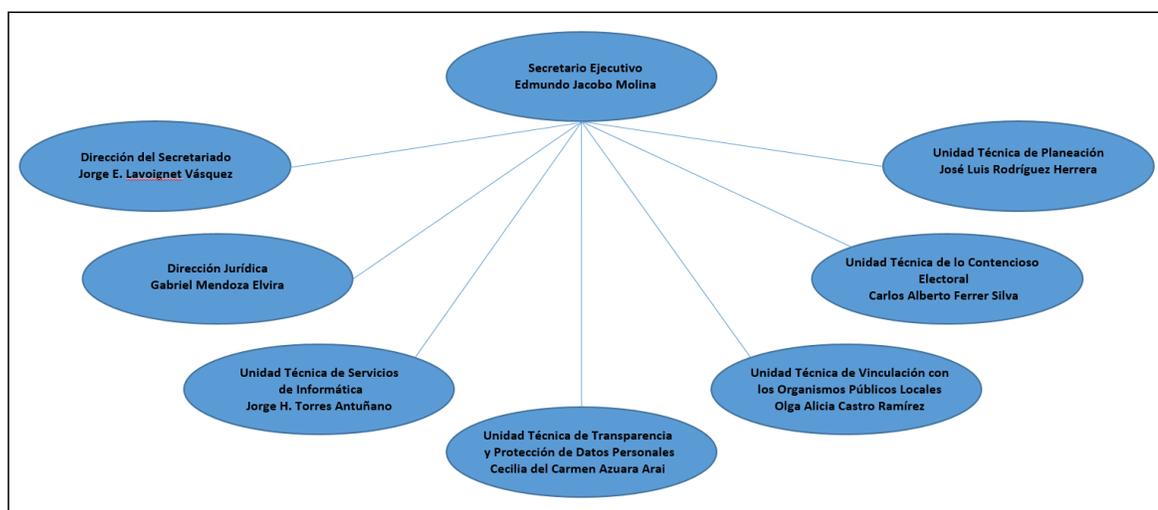


Figura 1.3 Organigrama Secretaría Ejecutiva.
Fuente: Elaboración propia (2015)

Unidad Técnica de Servicios de Informática (UNICOM)

Algunas de las atribuciones que tiene la UNICOM de acuerdo al artículo 66 del Reglamento Interior del Instituto Nacional Electoral son:

- Elaborar y proponer al Secretario Ejecutivo los proyectos estratégicos en materia de informática que coadyuven al desarrollo de las actividades del Instituto, para su presentación ante el Comité de Tecnologías de la Información y Comunicaciones, la Junta y/o el Consejo, según corresponda.
- Administrar la Red Nacional de Informática del Instituto, que interconecta sus oficinas a nivel nacional para la transmisión de voz, datos y video; estableciendo, en su caso, los mecanismos y procedimientos de

interconexión con las redes de los Organismos Públicos Locales y/o sedes de los partidos políticos.

- Establecer mejores prácticas y estándares, así como aplicar normas nacionales e internacionales a los procesos relacionados con tecnologías de la información a nivel institucional.
- Establecer y aplicar reglas, procedimientos y estándares en materia de seguridad informática, así como coordinar la aplicación de auditorías en la materia.

Departamento de Auditoria y Calidad de Sistemas (DACs)

Este departamento se encuentra dentro de la Subdirección de Análisis de Sistemas y Diseño de Bases de datos que depende de la Dirección de Sistemas de la UNICOM.

Se encarga de verificar y validar mediante revisiones de calidad, que los componentes y productos de las soluciones tecnológicas adquiridas o en desarrollo por parte de la UNICOM, cumplan con los requerimientos definidos por él área usuaria solicitante así como los estándares establecidos.

Mi historia en el INE

Al salir de la escuela estuve trabajando en Elektra, Compaq-HP entre 2000 y 2002 como soporte técnico help desk.

En febrero de 2003 entre a lo que antes era el Instituto Federal Electoral al Centro de Atención a Usuarios (CAU) donde se da soporte técnico (help desk), a los problemas relacionados con hardware y software (sistemas externos e internos del IFE) que reportan los usuarios del instituto.

En marzo de 2004 me cambie al CAU del Registro Federal de Electores como consultor de segundo nivel donde se da solución a los problemas que no se pueden resolver en el 1er nivel, dichos problemas son los que se presentan en los módulos de atención ciudadana como inconsistencias o errores en la base de datos, mal funcionamiento del Sistema Integral de Información del Registro Federal de Electores (SIIRFE-MAC), así como hardware y software externo.

A mediados de 2004 me entere de que se daba el diplomado de Desarrollo de Sistemas en Web en la FES-Aragón y me llamó la atención por lo que decidí inscribirme ya que necesitaba aprender cosas nuevas.

En septiembre de 2005 después de terminar el diplomado de Desarrollo de sistemas en web en la FES-Aragón tuve la oportunidad de cambiarme al Departamento de Auditoria y Calidad de Sistemas (DACS) del IFE donde se lleva a cabo la revisión de la calidad de los sistemas de información, para ello se realiza el análisis de requerimientos, elaboración casos de uso, definición de reglas de negocio, validaciones del sistema así como el diseño de los casos de prueba y su ejecución hasta la liberación a producción, aquí estuve hasta agosto del 2006.

Entrar al departamento de calidad del IFE hizo que me diera cuenta que me gusta esta parte del proceso de desarrollo de software y me ayudó mucho todo lo que nos enseñaron en el diplomado, por lo que desde ese año hasta el presente he estado en el área de calidad en el Instituto y en otras empresas, de septiembre 2006 a junio de 2007 en Accival Banamex Casa de Bolsa como Tester Jr, de julio de 2007 a febrero de 2008 en EFP Especialistas en Fondos de Previsión social como Tester Jr.

Al Instituto regrese de marzo de 2008 a diciembre de 2009, de marzo de 2010 a febrero de 2011 en ITERA it & business process como Ingeniero de pruebas, después de abril del 2011 y hasta el presente continuo formando parte del equipo del calidad del ahora Instituto Nacional Electoral.

Es por ello que después de estos años laborando en el área de calidad decidí presentar como informe de titulación pues considero que es una actividad muy importante dentro del proceso de desarrollo de software y me di a la tarea de investigar los conceptos más importantes acerca de las pruebas de software y que se utilizan en esta área de la ingeniería de software como antecedentes para poder entender todo el proceso y lo que ello implica.

CAPÍTULO 2

ANTECEDENTES DE LAS

PRUEBAS DE SOFTWARE

En este capítulo se describen los conceptos más importantes sobre las pruebas de software.

La calidad de software hoy en día tiene una gran importancia, es por ello que las compañías han decidido adoptar técnicas y nuevas tecnologías para mejorar sus procesos de prueba dentro del desarrollo de sistemas, para conocer más acerca de lo que significa la calidad en productos de software hay que conocer algunos conceptos como los que se describen a continuación:

1. Calidad de software

Existen diversos conceptos referente a la calidad aquí se especifican aquellos que están enfocados a la ingeniería de software.

Calidad. (IEEE Std 610)

Grado en el cual un componente, sistema o proceso satisface los requisitos especificados y/o necesidades del usuario/cliente y sus expectativas. IEEE. (1990). Sts 610.12.1990 Standar Glossary of Software Engineering Terminology

Calidad de software. (ISO/IEC 9126).

La totalidad de la funcionalidad de características de un producto de software que contribuyen a su habilidad de satisfacer necesidades especificadas o implícitas. ISO. (2005). Estándar Internacional ISO/IEC 9126 → ISO 25000:2005

De acuerdo a la norma ISO/IEC 9126 la calidad de software está constituida por:

➤ **Atributos funcionales**

Un conjunto de atributos que satisfacen las necesidades especificadas en los requisitos incluye:

- **Adecuación.** Atributos que determinan si el conjunto de funciones son apropiadas para las tareas especificadas.
- **Exactitud** o precisión que determinan que los efectos sean los correctos o los esperados.
- **Seguridad.** Atributos que miden la habilidad para prevenir accesos no autorizados, ya sea accidental o deliberado, tanto a programas como a datos.
- **Interoperabilidad.** Atributos que miden la habilidad de interactuar con sistemas especificados.

- **Cumplimiento o conformidad.** Atributos que hacen que el software adhiera a estándares relacionados con la aplicación, y convenciones o regulaciones legales.

➤ **Atributos no funcionales**

- **Fiabilidad.** Madurez, tolerancia a defectos, recuperación tras fallos. En determinadas condiciones, la aplicación mantendrá su capacidad y funcionalidad a lo largo de un período de tiempo, además de una pronta recuperación tras alguna falla.

$$\text{Fiabilidad} = \text{Calidad} / \text{Tiempo}$$

- **Usabilidad.** Fácil de aprender y usar por un conjunto de usuarios
- **Eficiencia.** Se refiere al comportamiento y nivel de desempeño de la aplicación y los recursos usados bajo las condiciones establecidas. Se miden tiempos de respuesta de procesamientos funcionales y los recursos usados durante su ejecución.
- **Mantenibilidad.** Se mide el esfuerzo requerido para realizar cambios en una aplicación.
- **Portabilidad.** Capacidad de la aplicación de ser trasferido o instalado en un nuevo entorno.

Error (IEEE610)

Acción humana que produce un resultado incorrecto

Defecto

Desperfecto en un componente o sistema que puede causar que el componente o sistema falle en desempeñar las funciones requeridas.

Fallo

Manifestación física o funcional de un defecto. Desviación de un componente o sistema respecto de la presentación, servicio o resultado esperado.

“Un error introduce un defecto, un defecto causa un fallo”

Prueba (test) (IEEE610)

Una actividad en la que un sistema o componente se ejecuta en condiciones específicas, se observan y graban los resultados, y se realiza una evaluación de algún aspecto del sistema o componente.

Caso de prueba (test case)

Conjunto de entradas y condiciones de ejecución que arrojan resultados esperados con un objetivo en particular, como el ejercicio de una ruta del programa en particular o para verificar el cumplimiento de un requerimiento específico.

2. Técnicas de pruebas

Las técnicas de pruebas facilitan el diseño de las pruebas para comprobar la lógica interna de los componentes de software y verificar las entradas y salidas de un programa para descubrir errores en la funcionalidad, comportamiento y rendimiento. Dentro del aseguramiento de la calidad se divide en técnicas estáticas y dinámicas para el caso que nos ocupa hablaremos sobre las técnicas dinámicas como se puede ver en la figura. 2.1

| | | | |
|---------------------------------------|---|---|--|
| Aseguramiento de la calidad analítico | Dinámico | Caja negra | <ul style="list-style-type: none">○ Particiones equivalentes○ Análisis de valores límite○ Pruebas de transición de estados○ Tablas de decisión○ Casos de uso |
| | | Técnicas basadas en la experiencia | |
| | Caja blanca | <ul style="list-style-type: none">○ Cobertura de sentencia○ Cobertura de rama○ Cobertura de condición○ Cobertura de camino | |
| Estático | <ul style="list-style-type: none">○ Revisiones / Revisiones guiadas○ Análisis de flujos de control○ Análisis de flujo de datos○ Métricas compilador/analizador | | |

Figura 2.1 Técnicas de pruebas.

Fuente: Testing IT Consulting S.A. de C.V. [Testing IT]. Probador Certificado (2009). Pág.90

Caja negra (black box)

Se obtiene a partir del análisis de requerimientos funcionales y no funcionales de un componente. La estructura del objeto es irrelevante o desconocida para

estas pruebas ver figura. 2.2. Se denomina prueba funcional o prueba orientada a la especificación

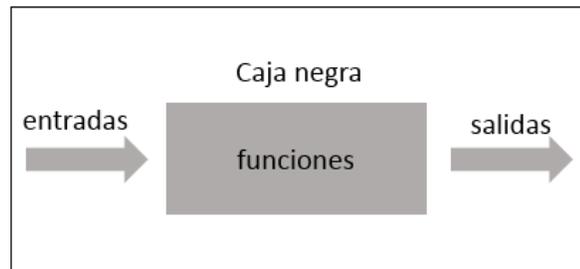


Figura 2.2 Pruebas de caja negra.
Fuente: Elaboración propia (2015)

La prueba de caja negra intenta encontrar errores de las siguientes categorías:

- Funciones incorrectas o ausentes.
- Errores de interfaz.
- Errores en estructuras de datos o en accesos a bases de datos externas.
- Errores de rendimiento.
- Errores de inicialización.

Particiones equivalentes (clases de equivalencia)

La partición en clases de equivalencia es lo que la mayoría de los probadores hacen de forma intuitiva, dividen los posibles valores en clases, mediante lo cual observan¹:

- Valores de entrada (input values) de un programa (uso habitual del método clases de equivalencia).
- Valores de salida (output values) de un programa (uso poco habitual del método clases de equivalencia)

El rango de valores definido se agrupa en clases de equivalencia para los cuales se aplican las siguientes reglas:

- Todos los valores para los cuales se espera que el programa tenga un comportamiento común se agrupan en una clase de equivalencia.
- Las clases de equivalencia pueden no superponerse y pueden no presentar ningún salto (discontinuidad)

¹ Ver. Alférez, H., (2007). Estrategias de pruebas, International Software Testing Qualifications Board, Standard glossary of terms used in Software Testing, Version 1.3. Obtenido de: <http://www.astqb.org/>

- Las clases de equivalencia pueden consistir en un rango de valores (por ejemplo: $0 < x < 10$) o en un valor aislado (por ejemplo: $x = \text{Verdadero}$).
- Para todo otro valor perteneciente a la CE se espera el mismo comportamiento que para el valor seleccionado.

Análisis de valores límite

“El análisis de valores límite amplía la técnica en clases de equivalencia introduciendo una regla para seleccionar a los representantes.” Alférez. (2007). Estrategias de pruebas (ISTQB). <http://www.astqb.org/>

Los valores frontera (valores límite) de la clase de equivalencia deben ser probados de forma intensiva:

- Frecuentemente los límites del rango de valores no están bien definidos o conducen a distintas interpretaciones,
- Comprobar si los límites han sido implementados (programados correctamente).
- La experiencia demuestra que con mucha frecuencia, los errores tienen lugar en los límites del rango de valores.

Si una clase de equivalencia está definido como un único valor numérico, por ejemplo $x = 5m$ los valores correspondientes al entorno también serán utilizados.

- Los representantes (de la clase y su entorno) son; 4,5 y 6.

Pruebas de transición de estado

Muchos métodos solo tienen en cuenta el comportamiento del sistema en términos de datos de entrada y datos de salida. No se tienen en cuenta los diferentes estados que puede tomar el objeto de prueba.

Por ejemplo el resultado de acciones que hubieran ocurrido en el pasado, acciones que hubieran causado que el objeto de prueba adquiera un determinado estado interno.

Los distintos estados que puede tomar un objeto de prueba se modelan a través del diagrama de transición de estado.

Las Pruebas de transición de estado son útiles para el desarrollo y de procedimiento orientado a objetos. Se basa en los conceptos de los estados y las máquinas de estados finitos.

- Una configuración de estado interno de un sistema o componente. Se define en términos de los valores asumidos en un particular para las variables que caracterizan el sistema o componente.
- Una máquina de estado finito es una máquina abstracta que puede ser representado por un diagrama de estados que tiene un número finito de estados y un número finito de transiciones entre los estados. Bursteun. (2003). p. 78.

Tablas de decisión

La partición de clases de equivalencia y el análisis de valores límite tratan entradas en condiciones aisladas. Sin embargo una condición de entrada puede tener efectos solo en combinación con otras condiciones de entrada y no tienen en cuenta el efecto de dependencias y combinaciones.

El uso del conjunto completo de las combinaciones de todas las clases de equivalencia de entrada conduce normalmente a un número muy alto de casos de prueba, con la ayuda de diagramas causa-efecto y tablas de decisiones obtenidas a partir de aquellas, la cantidad de combinaciones posibles se puede reducir de forma sistemática a un subconjunto de las mismas. Alférez. (2007). Estrategias de pruebas (ISTQB). <http://www.astqb.org/>

Casos de uso

Los casos de prueba se obtienen directamente a partir de los casos de uso del objeto de prueba. Un caso de uso describe la interacción de todos los actores involucrados conduciendo a un resultado final por parte del sistema, tiene precondiciones que deben ser cumplidas y pos condiciones que describen el sistema tras la ejecución del caso de prueba.

Caja Blanca (White box)

La prueba de caja blanca, denominada a veces prueba de caja de cristal es un método de diseño de casos de prueba que usa la estructura interna de un

programa. Mediante los métodos de prueba de caja blanca, el ingeniero del software puede obtener casos de prueba que:

- Garanticen que se ejercita por lo menos una vez todos los caminos independientes de cada módulo.
- Ejerciten todas las decisiones lógicas en sus vertientes verdadera y falsa.
- Ejecuten todos los bucles en sus límites y con sus límites operacionales.
- Ejerciten las estructuras internas de datos para asegurar su validez.

Durante las prueba de caja blanca, el programa objeto de las pruebas es ejecutado de la misma forma que las pruebas de caja negra. Ambas categorías conforman las pruebas dinámicas.

Las técnicas de caja blanca requieren el apoyo de herramientas en muchas áreas, a saber:

- Especificación de casos de prueba. Generación automática del diagrama del flujo de control a partir del código fuente del programa.
- Ejecución de prueba. Herramienta para monitorear y controlar el flujo del programa dentro del objeto de prueba.

El soporte de herramientas asegura la calidad de las pruebas e incrementa su eficiencia. Dada la complejidad de las mediciones necesarias para las pruebas de caja blanca, la ejecución manual de pruebas implica:

- Consumo de tiempo, consumo de recursos
- Dificultades en la implementación y propensión de errores.

Principales tipos de cobertura en las pruebas de caja blanca son:

- Cobertura de sentencia.
- Cobertura de decisión.
- Cobertura de condición.
- Cobertura de camino (“path coverage”), Porcentaje de caminos de ejecución que han sido practicados por casos de prueba

Técnicas basadas en la experiencia

Los casos de prueba están basados en la intuición y experiencia, también se denominan pruebas intuitivas incluyen predicción de errores y pruebas

exploratorias. Se aplican para complementar otros casos de pruebas, muchas veces no cumplen los criterios de un proceso de pruebas sistemático. Son útiles cuando el tiempo de ejecución de pruebas es mínimo.

3. Tipos de pruebas

Existen diferentes tipos de pruebas que se pueden realizar a un sistema, estas se pueden ejecutar en alguna etapa específica del proceso de.

Unitarias o componente

Dadas las convenciones para cada lenguaje de programación para la asignación de nombres a sus respectivos componentes se podrá hacer referencia a un componente como:

- Prueba de módulo (por ejemplo en C)
- Prueba de clase (Java o C++)
- Prueba de unidad (en Pascal)

Los componentes son referidos como módulos, clases o unidades. La prueba de unidad, centra el proceso de verificación en la menor unidad del diseño del software: el componente o módulo. Suelen ser llevadas a cabo por el equipo de desarrollo y específicamente, por el programador que codifica la unidad, la persona responsable de probar y ejecutar una serie de pruebas para asegurarse de que la unidad responde a sus exigencias

Alcance:

- Solo se prueban los componentes individuales.
- Un componente puede estar constituido por un conjunto de unidades más pequeñas.
- Los objetos de prueba no siempre pueden ser probados en solitario (de forma autónoma).
- Cada componente de forma independiente.
- Descubriendo defectos internos.
- Los efectos cruzados entre componentes quedan fuera del alcance de estas pruebas.

La ejecución pruebas de los componentes requiere, frecuentemente de “drivers” y “stubs”:

- Un “driver” procesa la interfaz de un componente, simulan estados de entrada, registran datos de salida y aportan una armadura de pruebas (test harness).
- Un “strub” reemplaza o simula un componente que aún no se encuentra disponible.

Integración

La prueba de integración es una técnica sistemática para construir la estructura del programa mientras que, al mismo tiempo, se llevan a cabo pruebas para detectar errores asociados con la interacción. El objetivo es coger los módulos probados mediante la prueba de unidad y construir una estructura de programa que esté de acuerdo con lo que dicta el diseño.

A menudo hay una tendencia a intentar una integración no incremental; es decir, a construir el programa mediante un enfoque de (“big bang”). Se combinan todos los módulos por anticipado. Se prueba todo el programa en conjunto. ¡Normalmente se llega al caos! Se encuentran un gran conjunto de errores. La corrección se hace difícil, puesto que es complicado aislar las causas al tener delante el programa entero en toda su extensión. Una vez que se corrigen esos errores aparecen otros nuevos y el proceso continúa en lo que parece ser un ciclo sin fin.

La integración incremental es la antítesis del enfoque del «big bang». El programa se construye y se prueba en pequeños segmentos en los que los errores son más fáciles de aislar y de corregir, es más probable que se puedan probar completamente las interfaces y se puede aplicar un enfoque de prueba sistemática. En las siguientes secciones se tratan varias estrategias de integración incremental diferentes.

- **Integración descendente**

La prueba de integración descendente es un planteamiento incremental a la construcción de la estructura de programas. Se integran los módulos moviéndose hacia abajo por la jerarquía de control, comenzando por el módulo de control principal (programa principal). Los módulos subordinados (subordinados de cualquier modo) al módulo de control principal se van incorporando en la estructura, bien de forma primero-en-profundidad, o bien de forma primero-en-anchura.

- **Integración ascendente**

La prueba de la integración ascendente, como su nombre indica, empieza la construcción y la prueba con los módulos atómicos (es decir, módulos de los niveles más bajos de la estructura del programa). Dado que los módulos se integran de abajo hacia arriba, el proceso requerido de los módulos subordinados siempre está disponible y se elimina la necesidad de resguardos.

Sistema

Las pruebas del sistema de software integrado tienen el objetivo de comprobar la conformidad con los requisitos especificados. La calidad de software es observada desde el punto de vista del usuario.

Las pruebas del sistema se refieren a (de acuerdo a ISO 9126):

- **Requisitos funcionales**
 - Funcionalidad. 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.
- **Requisitos no funcionales.** Se trata de definir criterios de aceptación tolerantes para el correcto funcionamiento de la aplicación
 - Fiabilidad
 - Usabilidad
 - Eficiencia
 - Mantenibilidad
 - Portabilidad

El entorno de pruebas debería coincidir con el entorno real.

- No son necesarios “stubs” o “drivers”.
- Todas las interfaces externas son probadas en condiciones reales.
- Emulación próxima al futuro entorno real del sistema.

No se realizan pruebas en el entorno real

- Los defectos inducidos podrían dañar el entorno real

- Un software objeto de despliegue se encuentra en un estado de cambio constante. Muchas pruebas no serán reproducibles.

Existen tres enfoques para probar requisitos funcionales:

1. Pruebas basadas en requisitos. Los casos de prueba se derivan de la especificación de requisitos o requerimientos. El número de casos de prueba variará en función del tipo/profundidad de las pruebas basadas en la especificación de los requisitos.
2. Pruebas basadas en procesos de negocio. Cada proceso de negocio sirve como fuente para derivar/generar pruebas. El orden de relevancia de los procesos de negocio pueden ser aplicados para asignar prioridades a los casos de prueba.
3. Pruebas basadas en casos de uso. Los casos de uso se derivan/generan a partir de las secuencias de usos esperados o razonables. Las secuencias utilizadas con mayor frecuencia reciben una prioridad más alta.

Aceptación de usuario

El objetivo de las pruebas de aceptación es validar que un sistema cumple con el funcionamiento esperado y permitir al usuario de dicho sistema que determine su aceptación, desde el punto de vista de su funcionalidad y rendimiento.

Las pruebas de aceptación son definidas por el usuario del sistema y preparadas por el equipo de desarrollo, aunque la ejecución y aprobación final corresponden al usuario.

La validación del sistema se consigue mediante la realización de pruebas de caja negra que demuestran la conformidad con los requisitos y que se recogen en el plan de pruebas, el cual define las verificaciones a realizar y los casos de prueba asociados. Dicho plan está diseñado para asegurar que se satisfacen todos los requisitos funcionales especificados por el usuario teniendo en cuenta también los requisitos no funcionales relacionados con el rendimiento, seguridad de acceso al sistema, a los datos y procesos, así como a los distintos recursos del sistema.

- **Alpha**

Cuando se construye software a medida para un cliente, se lleva a cabo una serie de pruebas de aceptación para permitir que el cliente valide todos los requisitos. La mayoría de los desarrolladores de productos de software llevan a cabo un proceso denominado pruebas alfa y beta para descubrir errores que parezca que sólo el usuario final puede descubrir.

Se lleva a cabo, por un cliente, en el lugar de desarrollo. Se usa el software de forma natural con el desarrollador como observador del usuario y registrando los errores y problemas de uso. Las pruebas alfa se llevan a cabo en un entorno controlado. Pressman. (2003). p. 316

- **Beta**

Se llevan a cabo por los usuarios finales del software en los lugares de trabajo de los clientes. A diferencia de la prueba alfa, el desarrollador no está presente normalmente. "Así, la prueba beta es una aplicación en vivo del software en un entorno que no puede ser controlado por el desarrollador. El cliente registra todos los problemas que encuentra durante la prueba beta e informa a intervalos regulares al desarrollador". Pressman. (2003). p. 316.

4. Proceso de pruebas (IEEE 829)

Actividades del proceso de pruebas:

- Planificación y control de pruebas (plan de pruebas)
- Especificación de pruebas (análisis y diseño de pruebas)
- Ejecución de pruebas (registro de defectos)
- Evaluación y generación de informes de pruebas
- Actividades de cierre de pruebas

Cada fase del proceso de pruebas tiene lugar de forma concurrente con las fases del proceso de desarrollo de software

Planeación

La planificación de pruebas es una parte importantes del aseguramiento de calidad. Se desarrolla un plan de pruebas que documenta la estrategia que se utilizará para comprobar y garantizar que un producto o sistema cumple con las

especificaciones de diseño y otros requisitos. El estándar IEEE 829. IEEE. (2008). Estándar 829 Standard for Software and System Test Documentation. aporta una estructura de plan de pruebas:

- Introducción. Indicar el contenido del plan de pruebas.
- Suposiciones u objetivos. Indicar el objetivo del plan de pruebas.
- Elementos de pruebas. Tipos de pruebas a realizar.
- Características sujetas a pruebas. Requerimientos de usuario.
- Características no sujetas a pruebas. Requerimientos no funcionales.
- Enfoque. Dirección que tomaran las pruebas a realizar.
- Criterios de paso/fallo. Lo que se tomara en cuenta para indicar que una prueba es exitosa o no.
- Criterios de suspensión o reanudación. En qué momento se pueden suspender una prueba o reanudar la ejecución de las mismas.
- Entregables de pruebas. Documentos soporte de la ejecución de las pruebas.
- Tareas de pruebas. Actividades que realizará cada tester en el diseño y ejecución de las pruebas.
- Necesidades relativas al entorno. Requerimientos del ambiente de pruebas
- Responsabilidades. Cada área y rol involucrado en durante la etapa de pruebas tiene unas responsabilidades muy específicas para que éxito del proceso.
- Calendario. Fechas en que se realizarán las actividades de pruebas (cronograma)
- Riesgos y contingencias. Elaborar plan de riesgos de no contar con recursos humanos y económicos, de infraestructura entre otros así como el plan de contingencia de cada uno.
- Aprobación. Aprobación por parte del usuario del producto entregado (pruebas de aceptación de usuario).

Especificación de pruebas

Diseño de verificación o pruebas de cumplimiento a realizar durante las etapas de desarrollo o aprobación del producto, por lo general, en una muestra pequeña de unidades. Elaboración de casos de prueba utilizando las diferentes técnicas de pruebas.

Implementación y ejecución de pruebas

Implementación del ambiente de pruebas para la ejecución de los casos de prueba así como el registro de defectos, generalmente de usa un sistema de gestión de incidencias.

Evaluación y generación de informes de pruebas

Después de la ejecución de las pruebas se debe entregar la información respecto a las actividades de pruebas realizadas en el periodo de tiempo especificado en el plan de pruebas, de acuerdo con el IEEE 829. IEEE. (2008). Estándar 829 Standard for Software and System Test Documentation. un informe del estado de pruebas puede contener:

- Objeto u objetos de pruebas. Que se va a probar
- Nivel de prueba, ciclo de prueba, periodo del informe. Si se realizarán pruebas exhaustivas o exploratorias, fechas de pruebas y entrega de informe.
- Avance de pruebas (pueden utilizarse métricas). Número de casos de pruebas ejecutados y cumplimiento de los requerimientos del usuario.
- Recursos utilizados. Recursos humanos, de infraestructura y económicos utilizados para realizar las pruebas.
- Metas alcanzadas (objetos probados). Ejecución de las pruebas que abarquen la totalidad de los requerimientos del usuario.
- Informe de defectos (número de defectos encontrados, corregidos)
- Evaluación del riesgo
- Pronóstico, actividades planificadas para el próximo periodo del informe
- Evaluación general y estado de las pruebas.

CAPÍTULO 3

PROCESO DE PRUEBAS

DE LOS SISTEMAS EN EL

INE

Descripción de las etapas del proceso de calidad, actividades, roles involucrados, formatos de entrada y salida de cada etapa.

El desarrollo y la calidad de soluciones tecnológicas se apoyan en metodologías como el Proceso Unificado, SCRUM, están dentro de la Administración de Proyectos de Tecnologías de Información (APTI) del Sistema de Gestión de Tecnologías de la Información (SIGETIC) del Instituto cuyo objetivo es:

Obtener los resultados esperados de los proyectos de TIC, mediante la administración efectiva y aplicación de conocimientos, habilidades, herramientas técnicas y recursos en el desarrollo de las actividades de los proyectos, para cumplir los objetivos de las iniciativas y programas de proyectos conforme a la normatividad vigente, así como a las directrices establecidas por la Unidad Técnica de Planeación. INE. (2015). Manual de procedimientos del sistema de gestión de tecnologías de la información y comunicaciones (sigetic). http://norma.ine.mx/documents/27912/286168/2013_Manual_Procedimientos_SIGETIC_JGE140_13/c552739a-2dfb-42c4-81b7-5c939829c41e.

Las pruebas de software pueden aportar fiabilidad a la calidad de software en caso de detectar pocos o ningún defecto. Pasar una prueba diseñada correctamente reduce el nivel general de riesgo de un sistema. Dentro del proceso de pruebas que se ejecuta en el instituto intervienen diferentes actores a saber:

Personal involucrado en el proceso de pruebas

Dentro del proceso de calidad de soluciones tecnológicas participan diferentes personas con actividades específicas y cada uno de ellos es parte fundamental.

- **Administrador de proyecto**
Administrar el proyecto, definir y cumplir los objetivos que se han de alcanzar, integrar a los roles que participan en el proyecto de las diferentes áreas.
- **Líder de desarrollo**
Dar seguimiento al equipo durante el desarrollo de la aplicación, en la ejecución de pruebas y resolución de defectos de la solución tecnológica
- **Líder de calidad**
Se encarga de determinar el enfoque de las pruebas, las técnicas que se utilizarán así como la cobertura de las pruebas. Es responsable del plan de pruebas y que las pruebas se ejecuten de acuerdo a lo establecido en el plan. También se encarga de apoyar al equipo de pruebas del proyecto y entregar informes al líder y administrador del proyecto.

- **Analista de pruebas de software**
Se encargará de analizar los requerimientos del usuario, para diseñar y en algunos casos ejecutar los casos de prueba apoyará para la creación de las historias de usuario o casos de uso generados
- **Probador (Tester)**
Se encargará de ejecutar los casos de prueba, revisará o en su caso apoyara para la creación de las historias de usuario o casos de uso generados, dará seguimiento a los defectos encontrados y será responsable de que estos queden cerrados.
- **Probador de seguridad**
Diseñar y ejecutar pruebas de seguridad con la finalidad de detectar vulnerabilidades en la solución tecnológica.
- **Desarrollador**
Desarrollar la aplicación con la tecnología y herramientas definidas para el proyecto, realizar pruebas unitarias y corregir los errores identificados durante las pruebas. Desarrollador de base de dato generar la base de datos y la documentación correspondiente.
- **Diseñador.**
Elaborar el prototipo de la solución tecnológica.

Una persona puede tener más de un rol en las diferentes etapas del desarrollo de software.

Inicio

Personal involucrado:

- Administrador del proyecto
- Líder de desarrollo
- Líder de calidad
- Área usuaria

El área usuaria entrega el requerimiento del servicio.

Los líderes de desarrollo y calidad se encargarán de revisar el requerimiento del usuario, identificar todos los paquetes de trabajo, así como realizar y actualizar el cronograma del proyecto.

El administrador del proyecto dará el vo.bo del cronograma, junto con el usuario formalizará el requerimiento y apoyará a la solución tecnológica.

Análisis

Personal involucrado:

- Líder de desarrollo
- Líder de calidad
- Analistas

En esta etapa los tester y desarrolladores se encargaran de la revisión de los requerimientos identificar las reglas de negocio, entidades y atributos, realizar historias de usuario.

El líder de calidad se encargara de identificar los roles que tendrá el sistema, apoyar en el análisis de requerimientos y especificación de reglas de negocio

El líder de desarrollo especificara la arquitectura de la solución tecnológica.

Diseño

Personal involucrado:

- Líder de desarrollo
- Líder de calidad
- Analista de pruebas de software
- Desarrollador
- Desarrollador de base de datos
- Diseñador

El analista diseña las pruebas de componente del paquete asignado, agregará o modificará las reglas de negocio e identificará los mensajes y validaciones del sistema.

El desarrollador diseña las pruebas unitarias que realizará al paquete asignado, realizará diagramas de clases, secuencias, estados entre otros si es que aplica.

El líder de calidad realizará el plan de pruebas y diseñara las pruebas de negocio del sistema así como diagramas de proceso y flujo de operación del sistema y cómo interactúan los roles que ya se identificaron

El líder de desarrollo se encargará de realizar diagramas de transición de estado, actividades.

El desarrollador de base de datos se encargará de realizar la documentación de la base de datos, diagramas entidad relación, diccionario de datos, dimensionamiento de la base de datos.

El diseñador se encarga de realizar el prototipo del sistema.

Codificación

Personal involucrado:

- Líder de desarrollo
- Líder de calidad
- Tester
- Desarrollador
- Desarrollador de base de datos

El tester se encargará de complementar las pruebas de componente del paquete asignado así como la preparación de los datos de prueba.

El desarrollador iniciará con la implementación del paquete asignado así como la ejecución de las pruebas unitarias diseñadas

El desarrollador de base de datos se encargara de desarrollar los script para la creación de la base de datos y las cargas iniciales si es que se requieren.

El líder de análisis y calidad se encargará de preparar el ambiente de pruebas, asignar usuarios de pruebas y apoyar a los tester en dudas sobre los casos de pruebas y preparación de datos de prueba.

El líder de desarrollo se encargará de hacer las solicitudes de pruebas, resolver las dudas técnicas o de implementación, integración de los paquetes de trabajo y generar el war de la aplicación y el manual de instalación.

Pruebas

Personal involucrado:

- Líder de desarrollo
- Líder de calidad
- Desarrollador
- Tester
- Probador de seguridad
- Área usuaria

El tester se encargará de la ejecución de las pruebas de caja negra, reportar defectos y dar seguimiento. Ejecutará las pruebas de regresión correspondientes.

El desarrollador se encargará de corregir los defectos encontrados y documentar la solución de los mismos.

El líder de calidad llevará un control de las pruebas ejecutadas así como un estatus de las mismas, solicitar pruebas de área usuaria y de seguridad.

El líder de desarrollo realizará la integración de los paquetes de trabajo corregidos, generar el war de la aplicación y actualizar el manual de instalación.

Resolverá los hallazgos reportador por el probador de seguridad

Probador de seguridad se encargará de ejecutar las pruebas de seguridad necesaria para verificar que el sistema no sea vulnerable cuando entre en operación.

El área usuaria se encargará de entregar observaciones o la aprobación del sistema.

Liberación

Personal involucrado:

- Administrador del proyecto.
- Líder de desarrollo
- Líder de calidad
- Área solicitante

El líder de desarrollo se encargará de preparar el paquete de instalación del proyecto y apoyar en el check list de instalación.

El líder de calidad se encargará de enviar la solicitud de liberación a producción, el war de la aplicación así como la solicitud de ejecución de script para la creación de la base de datos, validar la instalación de acuerdo al check list generado.

El administrador aprobará el check list de instalación y coordinará a las áreas involucradas para liberación, enviará la carta de la liberación a producción a todos los involucrados con el proyecto.

El área usuario realizará pruebas y aprobará la liberación de la aplicación.

Proceso de calidad

Verificar mediante revisiones de calidad, que los componentes y productos de las soluciones tecnológicas adquiridas o en desarrollo por parte de la UNICOM cumplan con los estándares establecidos y los requerimientos definidos de cada área con el fin de dar una mayor satisfacción al usuario, ver figura 3.1

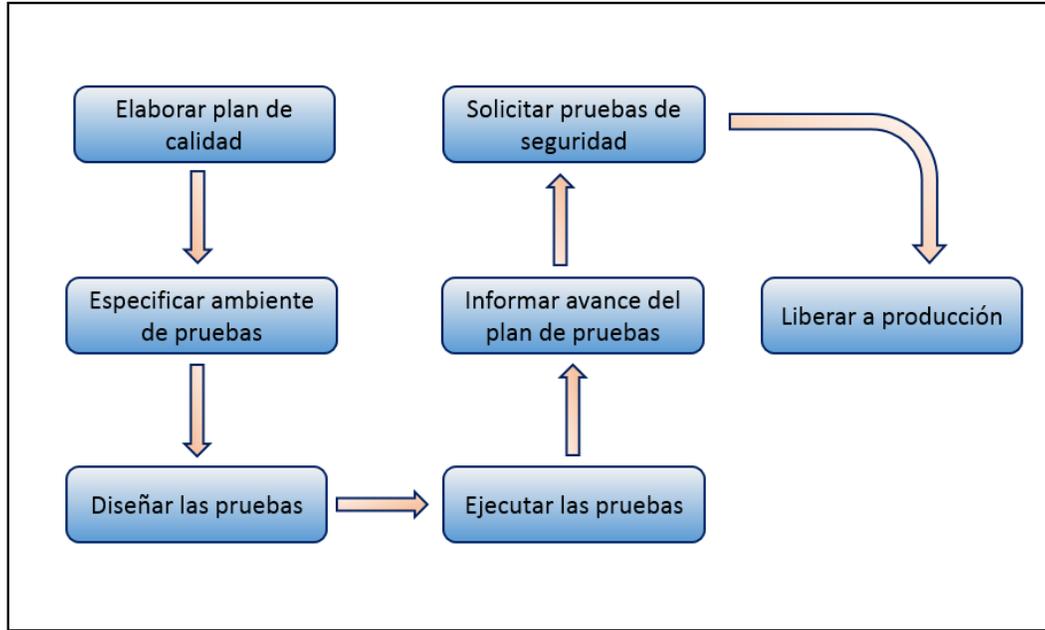


Figura 3.1 Proceso de calidad.
Fuente: Elaboración propia (2015)

Elaborar el plan de pruebas

En el plan de pruebas se describe el alcance, enfoque, recursos y calendario de las actividades de las pruebas. En él se identifican entre otras cosas los elementos de prueba, las características a probar, las tareas de prueba, quién realizará cada tarea, el ambiente de pruebas, las técnicas de diseño de las pruebas, los criterios de entrada y de salida que se utilizarán y la justificación de la elección, así como los riesgos y los planes de contingencia requeridos, ver figura 3.2

Para elaborar el plan de pruebas se requieren varios insumos como son:

- Acta constitutiva del proyecto
- Acta constitutiva del grupo de trabajo
- Cronograma
- Paquete de diseño de servicios

| TÍTULO | | PLAN DE CALIDAD |
|------------------|---|-----------------|
| CONTENIDO | | |
| 1. | INTRODUCCIÓN | 5 |
| 2. | REFERENCIAS | 5 |
| 3. | ALCANCE | 5 |
| 4. | ELEMENTOS DE PRUEBA | 9 |
| 5. | ESTRATEGIA PARA EL DISEÑO Y EJECUCIÓN DE PRUEBAS | 10 |
| 6. | PROCEDIMIENTOS DE USUARIO | 13 |
| 7. | CRITERIOS DE ACEPTACIÓN PARA LAS REVISIONES Y EJECUCIÓN PRUEBAS | 13 |
| 7.1 | CRITERIOS DE ACEPTACIÓN PARA LAS REVISIONES | 14 |
| 7.2 | CRITERIOS DE ACEPTACIÓN PARA LAS PRUEBAS | 14 |
| 8. | CRITERIOS DE SUSPENSIÓN Y REQUERIMIENTOS DE REANUDACIÓN | 14 |
| 8.1 | CRITERIO DE SUSPENSIÓN | 14 |
| 8.2 | REQUERIMIENTO DE REANUDACIÓN | 15 |
| 9. | MÉTRICAS | 15 |
| 10. | RIESGOS | 16 |
| 11. | ANEXO 1. DESCRIPCIÓN DE LOS TIPOS DE PRUEBA | 17 |

Figura 3.2 Plan de calidad.

Fuente: INE (2015). SIGETIC.

https://colabora.ife.org.mx/sites/SIGETIC/Formatos/_layouts/15/start.aspx#/CST%20Calidad%20de%20soluciones%20tecnolgicas/Forms/AllItems.aspx

Especificar el ambiente de pruebas

En esta etapa se describe el hardware y software asignados para poder ejecutar las pruebas. Los usuarios que utilizaran las diferentes áreas como son calidad, área usuaria y área de capacitación, esto de acuerdo a los roles definidos para la aplicación, ver figura 3.3

También se indicaran las cargas necesarias para poder realizar las pruebas a todos los módulos que conforman el sistema si es que aplica.

Los insumos que se requieren para poder realizar esta etapa son:

- Documento de visión
- Descripción de actores
- Trazabilidad de requerimientos.

En esta etapa se tiene como salida los documentos de:

- Especificación de ambiente
- Lista de verificación de la etapa de análisis

| TÍTULO | | ESPECIFICACIÓN DE AMBIENTE |
|------------------|---|----------------------------|
| CONTENIDO | | |
| 1. | Necesidades de ambiente..... | 5 |
| 1.1 | Hardware | 5 |
| 1.1.1 | PC necesarias para realizar las pruebas | 5 |
| 1.1.2 | LAP TOP necesarias para realizar las pruebas..... | 5 |
| 1.1.3 | Otro tipo de Hardware..... | 6 |
| 1.2 | Software..... | 6 |
| 1.2.1 | Software para la PC asignadas para las pruebas | 6 |
| 2. | Especificación de los usuarios de prueba para las diferentes áreas | 8 |
| 2.1 | Asignación de usuarios para el área encargada de realizar las pruebas | 8 |
| 2.2 | Asignación de usuarios para las pruebas que llevará acabo el Área usuaria | 11 |
| 2.3 | Asignación de usuarios para el Área de Capacitación..... | 12 |
| 3. | Cargas para realizar pruebas..... | 13 |

Figura 3.3 Especificación de ambiente.

Fuente: INE (2015). SIGETIC.

https://colabora.ife.org.mx/sites/SIGETIC/Formatos/_layouts/15/start.aspx#/CST%20Calidad%20de%20soluciones%20tecnicas/Forms/AllItems.aspx

Diseñar las pruebas

Antes de hacer la preparación de las pruebas se realiza el análisis de los requerimientos, aclaración de dudas, identificación de las reglas de negocio, revisión del diccionario de datos así como el diagrama entidad relación, esto permitirá poder diseñar los casos de pruebas de componente necesarios para validar toda la funcionalidad de estos, ver figura 3.4.

| TÍTULO: | | [Nombre del Proyecto] | | | | | | |
|--|--|---|---|--|-----------------------------------|--|--|--|
| Fecha de inicio de la creación del | dd/mm/aaaa | Fecha de término de | dd/mm/aaaa | | | | | |
| Fecha de inicio de las pruebas: | dd/mm/aaaa | Fecha de término de | dd/mm/aaaa | | | | | |
| Clasificación de la prueba: | Pruebas internas | | | | | | | |
| Líder de proyecto: | [Nombre(s) Apellido Paterno Apellido Materno] | | | | | | | |
| Ejecutor de pruebas: | [Nombre(s) Apellido Paterno Apellido Materno] | | | | | | | |
| Desarrollador: | [Nombre(s) Apellido Paterno Apellido Materno] | | | | | | | |
| Versión del War: | [Lista de los WAR probados] | | | | | | | |
| [Nombre de la historia de usuario - Nombre del módulo] | | | | | | | | |
| I. Precondiciones de acceso: | | | | | | | | |
| a) URL: | | | | | | | | |
| b) Usuario: | | | | | | | | |
| c) Password: | | | | | | | | |
| II. Precondiciones de negocio: | | | | | | | | |
| a) El usuario deberá estar registrado en la tabla USUARIOS_CAMPUS | | | | | | | | |
| b) Precarga en las tablas C_TIPO_MODELO, C_TEMATICA_MODELO, C_NIVEL_MODELO, C_TIPO_MODALIDAD, C_TIPO_PRESENCIA | | | | | | | | |
| Clave CP | Descripción | Pasos a seguir | Estatus CP | Datos de prueba | Resultados esperados | Resultados obtenidos | Resultado del CP | Comentarios |
| Flujo básico [Son las pruebas del camino feliz con valores válidos, pueden incluir validaciones de datos negativos] | | | | | | | | |
| [Se asigna un identificador consecutivo] | [Descripción generar del objetivo de la prueba a ejecutar] | [Pasos requeridos para ejecutar la prueba descrita en la columna "Descripción"] | [Estatus de ejecución del caso de prueba: No probado Probado] | [Datos necesarios para realizar los pasos capturados en la columna "Pasos a seguir". | [Detalle del resultado esperado.] | [Descripción detallada del defecto en caso de que se haya encontrado]. | [Estatus de la ejecución del caso de prueba] | [Se pueden agregar comentarios relacionados con el caso de prueba ejecutado] |

Figura 3.4 Caso de componente.

Fuente: INE (2015). SIGETIC.

https://colabora.ife.org.mx/sites/SIGETIC/Formatos/_layouts/15/start.aspx#/CST%20Calidad%20de%20soluciones%20tecnologicas/Forms/AllItems.aspx

También se diseñan los casos de prueba de negocio o integración, donde se considera la relación de diferentes módulos de acuerdo a un rol o varios, dependiendo de los procesos o flujos que integran el sistema, ver figura 3.5

| TÍTULO CASO DE PRUEBA DE NEGOCIO | | |
|--|-------------------------------------|-----------------------------|
| <p><i>[Nombre del proyecto]</i></p> <p><i>[Nombre de la prueba de integración a ejecutar]</i></p> | | |
| + | | |
| CLAVE_CP: | ESTATUS DEL CP: | RESULTADO DEL CP: |
| | <i><Probado / No probado></i> | <i><Paso / Falló></i> |
| PRECONDICIONES DE ACCESO: | | |
| PRECONDICIONES DE PRUEBAS: | | |
| <i>[Especificar las precondiciones necesarias para la ejecución de pruebas del presente caso de negocio]</i> | | |
| PRECONDICIONES DE NEGOCIO: | | |
| <i>[Especificar las reglas de negocio que son necesarias para la revisión del flujo de negocio]</i> | | |
| REFERENCIA HISTORIA DE USUARIO: | | |
| <i>[Listar todos las historias de usuario relacionados con el flujo de negocio que se probará]</i> | | |
| REFERENCIA REGLAS DE NEGOCIO: | | |
| <i>[Se puede hacer referencia a los documentos que incluyen las Reglas de negocio del proyecto o en su caso describir de manera detallada las reglas de negocio que aplican para el presente caso de pruebas de negocio]</i> | | |
| DESCRIPCIÓN: | | |
| <i>[Describir en una línea brevemente las funcionalidades que se probarán, es decir, los componentes o módulos que interactúan entre sí para obtener la funcionalidad completa de negocio.]</i> | | |
| PASOS A SEGUIR: | | |

Figura 3.5 Caso de prueba de negocio.

Fuente: INE (2015). SIGETIC.

https://colabora.ife.org.mx/sites/SIGETIC/Formatos/_layouts/15/start.aspx#/CST%20Calidad%20de%20soluciones%20tecnologicas/Forms/AllItems.aspx

Adicional se diseñan dos casos de prueba que se van ejecutando conforme se van liberando los módulos, ver figura 3.6

- **Casos de prueba de acceso**, estos se diseñan a partir de una matriz de roles donde se indican los tipos de usuarios y el acceso que tendrá cada uno de acuerdo al nivel operativo que se maneja en el INE como son Junta Distrital, Junta Local y Oficinas Centrales.

- **Casos de prueba de base de datos**, estos se diseñan de acuerdo al diccionario de datos y la finalidad es verificar que se haya creado la base de datos, tablas, campos, relaciones y restricciones de acuerdo a lo establecido en dicho documento.

| TÍTULO: CASO DE PRUEBA | | | | | | | | |
|---|---|--|------------------|---|---|----------------------|------------------|-------------|
| Reclutamiento y Seguimiento a Supervisores Electorales y Capacitadores-Asistentes Electorales V 5.0 | | | | | | | | |
| Fecha de inicio de la creación del documento: | | | 02/09/2014 | del documento: | | dd/mm/aaaa | | |
| Fecha de inicio de las pruebas: | | | dd/mm/aaaa | Fecha de término de las pruebas: | | dd/mm/aaaa | | |
| Clasificación de la prueba: | | | Pruebas internas | | | | | |
| Líder de proyecto: | | | | | | | | |
| Ejecutor de pruebas: | | | | | | | | |
| Desarrollador: | | | | | | | | |
| Versión del War: [Lista de los WAR probados] | | | | | | | | |
| HisUs_31_Registro_Aspirantes_SUPYCAP_5.0.docx - Registro de aspirantes | | | | | | | | |
| Clave CP | Descripción | Pasos a seguir | Estatus CP | Datos de prueba | Resultados esperados | Resultados obtenidos | Resultado del CP | Comentarios |
| Base de datos vacía | | | | | | | | |
| CP01 | Verificar el nombre, tipo de dato, longitud, constraints (PK, FK) y descripción en cada tabla vs el DD | 1. Abrir las tablas: - ASPIRANTES - EVALUACION_CURRICULAR 2. Seleccionar la pestaña de Columns y comparar el nombre, tipo de dato, longitud, constraints (PK, FK) y descripción de los campos en cada tabla con el diccionario de datos | | Bajar el diccionario de datos DicDa_SUPYCAP_140902.xlsx del SVN: /home/repositorio/DIS-UNICOM/Sistemas/Electorales/Proceso electoral 2014-2015/SUPYCAP | La información que se muestra en la base de datos y el diccionario de datos para cada tabla y campo coincide en cuanto a: - nombre - tipo de dato - longitud - constraints (PK, FK) | | | |
| Integración de negocio (entradas) | | | | | | | | |
| CP07 | Verificar que no se muestra la pantalla de captura de aspirantes cuando no se tienen inf. de los módulos dependientes | 1. Entrar al sistema 2. Seleccionar del menú Aspirantes la opción "Registro de aspirantes" | | | El sistema muestra el mensaje: No hay convocatorias registradas | | | |
| Flujo básico general | | | | | | | | |
| Acceso | | | | | | | | |
| Escenarios alternos | | | | | | | | |
| Bitácora base de datos | | | | | | | | |
| Concurrencia | | | | | | | | |
| Seguridad | | | | | | | | |
| Integración (prueba sobre objetos en la pantalla) | | | | | | | | |
| Interfaz | | | | | | | | |

Figura 3.6 Caso de prueba de base de datos.

Fuente: INE (2015). SIGETIC.

https://colabora.ife.org.mx/sites/SIGETIC/Formatos/_layouts/15/start.aspx#/CST%20Calidad%20de%20soluciones%20tecnológicas/Forms/AllItems.aspx

Los insumos que se requieren para poder realizar esta etapa son:

- Requerimientos de usuario
- Historias de usuario
- Reglas de negocio
- Diccionario de datos
- Diagrama entidad relación
- Mensajes y validaciones.

En esta etapa se tiene como salida los documentos de:

- Caso de prueba de componente
- Caso de prueba de negocio
- Caso de prueba de acceso
- Caso de prueba de base de datos

Ejecutar las pruebas

Antes de iniciar esta etapa se debe asegurar que esté lista la aplicación y la base de datos para que los tester ejecuten las pruebas.

Los primeros casos de pruebas en ejecutarse son los referentes a la validación la de base de datos y verificación de acceso, en seguida los de componente para finalizar con los de negocio, esto para los módulos o reportes liberados.

Al encontrar un defecto este se debe avisar al desarrollador y registrar en el documento específico donde se indica la versión de la aplicación, ver figura 3.7, la fecha en que se encontró el defecto, la clave del caso de prueba, el nombre del tester, el tipo de defecto, el impacto que tiene en el sistema, la prioridad en su solución, la descripción breve y los pasos a seguir para replicarlo, el responsable de desarrollo y comentarios adicionales.

| TÍTULO: REGISTRO DE DEFECTOS | | | | | | | | | | | | | | | |
|--|--|--------------------------------|--------------------------|---------------------------|-----------------|---------------------|---------|-----------|-----------|--|--------------------------|-----------------------------------|----------------------|---------------------------|-----------------------------|
| [Se debe generar una pestaña por cada módulo del sistema y se deben registrar los defectos detectados durante la ejecución de los casos de prueba] | | | | | | | | | | | | | | | |
| No. | Versión del WAR o ejecutable del Sistema | Fecha de detección del defecto | Clave del caso de prueba | Persona que lo desarrolló | Tipo de defecto | Estatus del defecto | Impacto | Prioridad | Artefacto | Descripción breve y pasos a seguir | Detectado en la fase de: | Motivos que originaron el defecto | Solución o Respuesta | Persona que lo identificó | Observaciones / Comentarios |
| 1 | Ver. 0.21 20140611_1315 | 12/06/2014 | CP04 | Luis Alberto del Campo | Funcionalidad | Nuevo | Alto | Alto | Sistema | 1. Entrar al sistema 2. Seleccionar del menú Difusión de la convocatoria la opción "Verificación de gabinete" 3. Seleccionar del menú hamburguesa la opción Modificar: El sistema marca error: Error: L_spring_security_scp_... ... org.springframework.web.servlet.DispatcherServlet.THEME_SOURCE=WebApplicationContext for namespace 'appServlet-servlet': startup date [Wed Jun | Codificación | | | Lorena Granja García | |

Figura 3.7 Registro de defectos.

Fuente: INE (2015). SIGETIC.

https://colabora.ife.org.mx/sites/SIGETIC/Formatos/_layouts/15/start.aspx#/CST%20Calidad%20de%20soluciones%20tecnol%C3%93gicas/Forms/AllItems.aspx

Se realizan pruebas de regresión hasta que son corregidos todos los defectos o que existan defectos conocidos que no afecten la funcionalidad del sistema previa autorización del administrador del proyecto.

Los insumos que se requieren para poder realizar esta etapa son:

- Caso de prueba de componente
- Caso de prueba de negocio
- Caso de prueba de acceso
- Caso de prueba de base de datos

En esta etapa se tiene como salida los documentos de:

- Registro de defectos
- Lista de verificación de la etapa de diseño

Informar avance del plan de pruebas.

Al finalizar la primera entrega, se realiza un informe del estado del plan de calidad sobre el seguimiento y resolución de los defectos encontrados, ver figura 3.8. Este informe sirve para que todos los involucrados en el desarrollo del proyecto estén enterados del avance en las pruebas y de ser posible aprobar la liberación a seguridad.

| TÍTULO INFORME DEL ESTADO DEL PLAN DE CALIDAD ESPECÍFICO | |
|---|---|
| CONTENIDO | |
| 1. OBJETIVO..... | 5 |
| 2. FECHA DE INFORME..... | 5 |
| 3. COBERTURA DE CASOS DE PRUEBA EJECUTADOS..... | 5 |
| 4. NÚMERO DE DEFECTOS SOBRE MÓDULOS DESARROLLADOS, CLASIFICADOS POR TIPO DE DEFECTO | 6 |
| 5. NÚMERO DE DEFECTOS POR ESTATUS | 6 |
| 6. ANEXO 1. DEFINICIÓN DE ESTATUS DE UN DEFECTO | 7 |
| 7. GLOSARIO DE TÉRMINOS..... | 7 |

Figura 3.8 Informe del estado de plan de calidad.

Fuente: INE (2015). SIGETIC.

https://colabora.ife.org.mx/sites/SIGETIC/Formatos/_layouts/15/start.aspx#/CST%20Calidad%20de%20soluciones%20tecnologicas/Forms/AllItems.aspx

Los insumos que se requieren para poder realizar esta etapa son:

- Registro de defectos
- Lista de verificación de la etapa de diseño

En esta etapa se tiene como salida los documentos de:

- Informe del estado del plan de calidad específico
- Liberación de calidad ver figura 3.9

DETALLE DE LA LIBERACIÓN

| | |
|---|---|
| TIPO DE REVISIÓN: | [Sistema, módulo de registro, portal, herramienta, servidor, documentación etc.] |
| NOMBRE DEL SISTEMA: | [Nombre completo del sistema] |
| VERSIÓN DEL SISTEMA: | [Versión del sistema] |
| DOCUMENTO DE LA SOLICITUD DE PRUEBAS DE SEGURIDAD: | [Ruta y nombre de la ubicación del archivo de la solicitud de pruebas de seguridad] |
| SITUACIÓN ACTUAL: | [Describir a grandes rasgos las características con las que se libera el sistema/aplicación.] |
| DESCRIPCIÓN DE LA LIBERACIÓN: | <i>Este sistema ha pasado la prueba de control de <u>calidad</u>, siendo las pruebas satisfactorias, se concluye la primera revisión del sistema en el ambiente de pruebas.</i> |
| FECHA DE LIBERACIÓN: | [DD del Mes del AAAA] |
| SHA1SUM DEL ARCHIVO: | [Versión del <u>war</u> o paquete liberado por seguridad] |
| NOMBRE DEL WAR O APLICACIÓN: | [<u>nombre aplicacion war</u>] |
| VERSIÓN DEL WAR O APLICACIÓN: | [X.XX] |
| NÚMERO DE REVISIONES: | [Describir el número de versiones que fueron liberadas para el ambiente de Pruebas] |
| CARACTERÍSTICAS DE LA REVISIÓN: | [Especificar los tipos de prueba que se ejecutaron, considerados en el Plan de Calidad] |
| DOCUMENTO DEL REGISTRO DE DEFECTOS DURANTE LA REVISIÓN: | [Ubicación y nombre en el repositorio o herramienta donde se registraron los defectos] |
| LISTADO DE MÓDULO (S) LIBERADOS: | [Listar todos los módulos y reportes liberados] |
| PROCEDIMIENTO (S) DE USUARIO LIBERADOS: | [Listar todos los procedimientos de usuario y/o materiales de capacitación] |

Figura 3.9 Liberación de calidad.

Fuente: INE (2015). SIGETIC.

https://colabora.ife.org.mx/sites/SIGETIC/Formatos/_layouts/15/start.aspx#/CST%20Calidad%20de%20soluciones%20tecnologicas/Forms/AllItems.aspx

Solicitar pruebas de seguridad

Una vez que se ha autorizado la liberación de se realiza la solicitud de pruebas de seguridad al departamento correspondiente ver figura 3.10

Es importante señalar que a la par que se realizan estas pruebas de seguridad se entrega el sistema al área usuaria para que haga la revisión correspondiente.

Además se explica el funcionamiento del sistema al área de capacitación para que pueda elaborar las guías de usuario, capsulas informativas, diagramas y demás instrumentos necesarios para capacitar al usuario sobre la operación del sistema.

| TÍTULO SOLICITUD DE PRUEBAS DE SEGURIDAD | | | |
|--|------------|-----|-----------------|
| 1. Datos del Sistema en el ambiente de pruebas | | | |
| ID | Nombre | URL | IP del servidor |
| | | | |
| | | | |
| 2. Cuentas de usuarios con los máximos privilegios y su respectiva contraseña | | | |
| Usuario | Contraseña | Rol | |
| | | | |
| | | | |
| 3. Cuentas de usuarios si el sistema usa diferentes roles y sus respectivas contraseñas | | | |
| Usuario | Contraseña | Rol | |
| | | | |
| | | | |
| | | | |
| 4. Códigos de seguridad | | | |

Figura 3.10 Solicitud de pruebas de seguridad.

Fuente: INE (2015). SIGETIC.

https://colabora.ife.org.mx/sites/SIGETIC/Formatos/_layouts/15/start.aspx#/CST%20Calidad%20de%20soluciones%20tecnologicas/Forms/AllItems.aspx

Si se tienen observaciones por parte del área usuaria o incluso nuevos requerimientos, estos se revisan y analizan para ver el impacto que tendrían y los tiempos necesarios para poder realizar los ajustes necesarios.

Si se aprueban dichos cambios se realiza un ajuste a los casos de pruebas por parte del área de calidad, el área de desarrollo se encarga de la implementación y de nueva cuenta se realiza la ejecución de las pruebas, información del avance de las pruebas para entregar al área de seguridad una nueva versión donde se incluyen los cambios solicitados por el área usuaria, la corrección a los hallazgos encontrados por parte del departamento de seguridad para que se realicen de nuevo las pruebas de esta área y se libere la aplicación.

| TÍTULO | AUDITORÍA DE SEGURIDAD DE APLICACIONES |
|---------------------------------|--|
| Contenido | |
| Introducción..... | 3 |
| Fechas de la Auditoría..... | 3 |
| Objetivo | 3 |
| Área responsable | 3 |
| Áreas involucradas..... | 3 |
| Sistema..... | 3 |
| Descripción del Hallazgo N..... | 4 |
| Estrategia de mitigación | 5 |
| Conclusiones | 13 |
| Firmas | 14 |
| Evidencias | 15 |
| Códigos de integridad | 19 |

Figura 3.11 Reporte de liberación de seguridad.

Fuente: INE (2015). SIGETIC.

https://colabora.ife.org.mx/sites/SIGETIC/Formatos/_layouts/15/start.aspx#/CST%20Calidad%20de%20soluciones%20tecnologicas/Forms/AllItems.aspx

Los insumos que se requieren para poder realizar esta etapa son:

- Solicitud de pruebas de seguridad
- Versión final de la aplicación
- Manual de instalación
- Liberación de calidad

En esta etapa se tiene como salida los documentos de:

- Reporte de hallazgos, propuesta de acciones para mitigación de riesgos y liberación, ver figura 3.11

Liberar a producción

Una vez liberada la aplicación en el ambiente de seguridad y realizados los ajustes solicitados por el área usuaria, se pide instalar en el ambiente de producción.

Los insumos que se requieren para poder realizar esta etapa son:

- Manual de instalación de la aplicación y base de datos.
- Diagrama E-R
- Diccionario de datos
- Dimensionamiento de la base de datos

- Script para creación de la base de datos, cargas y procesos especiales.
- Aplicación
- Reporte de liberación de seguridad.

En esta etapa se tiene como salida los documentos de:

- Solicitud de instalación a producción
- Oficio de liberación de la aplicación al área usuaria

En algunas ocasiones dependiendo de la aplicación se solicita realizar ejercicios a nivel nacional para que los usuarios se familiaricen con la operación del sistema, en este punto si se llegasen a encontrar defectos o el usuario solicitara algún ajuste, se analiza y si se aprueba se hace la implementación, pruebas de calidad, pruebas de seguridad para que se libere nuevamente por ambas áreas. Con esto ya se hace la liberación oficial para su operación en producción.

Conclusiones

Sin duda es muy importante hacer estas verificaciones al software que se desarrolla desde las primeras etapas del proceso ya que esto al final nos ahorrara tiempo, dinero y sobre todo esfuerzo.

Antes de entrar al departamento de calidad del instituto no sabía todo el esfuerzo que implicaba para que un sistema fuera liberado a producción, que existen metodologías de pruebas y herramientas que nos pueden ayudar a la revisión. Estas metodologías y herramientas han ido evolucionando a lo largo de tiempo que he estado en el INE y en otras empresas, dichas empresas hacen la implementación de las diferentes metodologías para crear sus estándares de desarrollo así como las buenas prácticas que en ellas existen.

No somos infalibles y el hecho de que no se encuentren efectos antes de liberar una aplicación, no garantiza que no puedan ocurrir mientras esta en operación, sin embargo las pruebas pueden aportar fiabilidad a la calidad de software, ejecutar una prueba diseñada correctamente reduce el nivel de riesgo de un sistema.

Es sabido que en cualquier empresa dedicada al desarrollo de software el tiempo que más se sacrifica es el de pruebas, esto debido a nuevos requerimientos, mal análisis, rotación de personal, poca infraestructura o tiempo para el desarrollo pero al final es muy importante dedicar tiempo a las pruebas para poder garantizar en mayor medida el cumplimiento de los requerimientos solicitados por el usuario.

En el INE ha sido muy satisfactorio el trabajo realizado a lo largo de todos estos años no solo por mí, si no todos los compañeros involucrados en todo el proceso de desarrollo de los sistemas que se han utilizado en los procesos electorales del 2006, 2009, 2012 y 2015, aun con las implicaciones políticas que puede llegar a tener un error en producción, al final todo nos ha permitido ir creciendo y cada día encontrar opciones de mejora para poder entregar productos confiables y acorde a las necesidades de los usuarios.

Es por ello que pienso que a los que inician la carrera de Ingeniería en Computación y que se sientan atraídos por esta área en particular tengan en cuenta que como tester que comúnmente nos llaman, es algo más que probar un sistema y detrás de todo esto existe una metodología de pruebas incluso varias certificaciones; además al final del proceso somos los que mejor conocemos las reglas del negocio, la base de datos, la funcionalidad del sistema a veces incluso más que el propio usuario.

BIBLIOGRAFÍA

Binder R. V., (2000). *Testing Object-Oriented Systems: Models, Patterns, and Tools*. Addison-Wesley.

Kshirasagar, N., Priyadarshi T. (2008). *Software Testing and Quality Assurance: Theory and Practice*. Wiley.

Perry E. W., (2006). *Effective Methods for Software Testing Third Edition*. Wiley.

Pressman, R. (2003) *Ingeniería del Software: Un Enfoque Práctico*. McGraw-Hill

Godbole, S. N., (2004). *Software Quality Assurance*. Alpha Science.

Burnstein, I. (2003). *Practical Software Testing*. Springer Professional Computing

Alarcos. (2002). *Metodología MÉTRICA versión 3. Técnicas y Prácticas*. Ministerio de Administraciones Públicas

Alfárez, H., (2007). Estrategias de pruebas, International Software Testing Qualifications Board, Standard glossary of terms used in Software Testing, Version 1.3. Obtenido de: <http://www.astqb.org/>

Collado, M. (2003). Pruebas de software. Obtenido de: www.lml.ls.fi.upm.es/ftp/ed2/0203/Apuntes/pruebas.ppt

Acuña, J.A., (2009). Pruebas de software. Obtenido de: <http://www.slideshare.net/aracelij/pruebas-de-software>

Tuya, J. (2009). Hacia el nuevo estándar de pruebas ISO / IEC 29119. Obtenido de <http://in2test.lsi.uniovi.es/gt26/presentations/ISO-29119-Javier-Tuya-JICS2009.pdf>

Sanz, M.L., (2009). Ingeniería del software de gestión. Obtenido de: http://www.kybele.etsii.urjc.es/docencia/IS_LADE/2009-2010/Material/%5BIS-LADE-2009-10%5DTema2.Ciclo%20de%20vida%20del%20SW.pdf

INE. (2015). Estructura e historia. Obtenido de: <http://www.ine.mx/portal/>

INE. (2015). Sistema de Tecnologías de la Información y Comunicaciones. Obtenido de: http://colabora.ife.org.mx/_layouts/15/start.aspx#/SitePages/Inicio.aspx