



INSTITUTO UNIVERSITARIO AERONAUTICO
PROYECTO DE GRADO
Ingeniería en Sistemas de Información

**“Sistema de Gestión de Calidad de Software –
QUALITY TRACKING”**

Integrantes:

- Valverde María Virginia
- Vera Calle María Belén

Tutor:

- Ing. Ernesto Bauer.



Contenido

Agradecimientos	4
Resumen del Proyecto de Grado.....	5
Módulo 1: Introducción.....	7
Introducción	8
Justificación	11
Alcance	12
Objetivos	12
Módulo 2: Marco Teórico.....	14
Módulo 3: Modelado de Negocio	45
Introducción	46
Definición de Actores	46
Definición de Transacciones del Negocio.....	48
Módulo 4: Requerimientos	50
Introducción	51
Requerimientos del Sistema.....	52
Actores del Sistema de Información	53
Prototipos de interfaz gráfica.....	53
MODULO USUARIOS	53
MODULO PROYECTOS.....	55
MODULO REQUERIMIENTOS.....	57
MODULO CASOS DE PRUEBA	57
MODULO CICLOS DE PRUEBA	58
MODULO INCIDENCIAS	60
MÓDULO MÉTRICAS	60
Módulo 5: Análisis y Diseño del Sistema	64
Introducción	65
Diagrama de Casos de Uso del Sistema de Información.....	66
Descripción de Casos de Uso.....	68
CASO DE USO: 001. REGISTRAR USUARIO.	68
CASO DE USO: 002. ELIMINAR USUARIO.....	68
CASO DE USO: 003. MODIFICAR USUARIO.....	69
CASO DE USO: 004. MODIFICAR CONTRASEÑA USUARIO	70
CASO DE USO: 005.INICIAR SESIÓN.....	71



CASO DE USO: 006. CERRAR SESIÓN	72
CASO DE USO: 007. REGISTRAR PROYECTO	72
CASO DE USO: 008. ELIMINAR PROYECTO	73
CASO DE USO: 009. MODIFICAR PROYECTO	74
CASO DE USO: 010. REGISTRAR REQUERIMIENTO	75
CASO DE USO: 011. ELIMINAR REQUERIMIENTO	76
CASO DE USO: 012. MODIFICAR REQUERIMIENTO	76
CASO DE USO: 013. REGISTRAR CASO DE PRUEBA	77
CASO DE USO: 014. ELIMINAR CASO DE PRUEBA	78
CASO DE USO: 015. MODIFICAR CASO DE PRUEBA	79
CASO DE USO: 016. COPIAR CASO DE PRUEBA	80
CASO DE USO: 017. REGISTRAR CICLO DE PRUEBA	81
CASO DE USO: 018. ELIMINAR CICLO DE PRUEBA	82
CASO DE USO: 019. MODIFICAR CICLO DE PRUEBA	82
CASO DE USO: 020. COPIAR CICLO DE PRUEBA	83
CASO DE USO: 021. ASIGNAR PRUEBAS A CICLO	84
CASO DE USO: 022. REGISTRAR DETERMINACIÓN DE PRUEBA	84
CASO DE USO: 023. REGISTRAR INCIDENCIA	86
CASO DE USO: 024. ELIMINAR INCIDENCIA	87
CASO DE USO: 025. MODIFICAR INCIDENCIA	88
CASO DE USO: 026. GENERAR REPORTE GENERAL	89
CASO DE USO: 027. GENERAR REPORTE INCIDENCIAS	90
CASO DE USO: 028. GENERAR REPORTE DE AMPLITUD O COBERTURA DE PRUEBAS 91	
CASO DE USO: 029. GENERAR REPORTE DE MADUREZ DE PRUEBAS	92
CASO DE USO: 030. GENERAR REPORTE PERFILES DE FALLOS	93
CASO DE USO: 031. GENERAR REPORTE DE DENSIDAD DE DEFECTOS	94
CASO DE USO: 032. GENERAR REPORTE POR MÓDULOS	94
CASO DE USO: 033. REGISTRAR GRUPO DE USUARIOS.	95
CASO DE USO: 034. ELIMINAR GRUPO DE USUARIOS	96
CASO DE USO: 035. MODIFICAR GRUPO DE USUARIOS	97
Diagrama de Estados.....	98
Arquitectura de Solución.....	98
Punto de Vista de Sistemas	99
PUNTO DE VISTA DE APLICACIÓN	99
PUNTO DE VISTA DE IMPLEMENTACIÓN:	100
PUNTO DE VISTA DE DATOS	101



PUNTO DE VISTA TECNOLÓGICO	105
Diagrama Entidad – Relación	106
Módulo 6: Pruebas del Sistema	109
Conclusiones	112
Bibliografía	113



Agradecimientos

Agradecemos a nuestras familias por el apoyo brindado durante el desarrollo de nuestra carrera, y el soporte emocional durante la realización de nuestro Proyecto de Grado.

A nuestro Tutor, el Ingeniero Ernesto Bauer, quien nos guio durante este proceso de tesis, brindándonos los consejos en base a su experiencia.

Al Ing. Eliel Arriola por ayudarnos con las dudas y colaboración desde su experiencia como programador, capacitándonos y dándonos el soporte necesario para el desarrollo de nuestro sistema informático.

A la empresa Harriague y Asociados por permitirnos llevar a cabo nuestra idea, desarrollando un producto útil para sus labores diarios.

¡Muchas gracias a todos por el apoyo!
María Belén y María Virginia.



Resumen del Proyecto de Grado

El Objetivo de nuestro Proyecto de Grado, es el diseño y desarrollo de una herramienta de gestión de Calidad para los productos de Software desarrollados por la empresa Harriague y Asociados.

La aplicación desarrollada, bajo el nombre de “**QUALITY TRACKING**”, permite la gestión de las pruebas para lograr el aseguramiento de la calidad de los productos entregados.

Está compuesta por los siguientes módulos:

- Proyectos.
- Requerimientos.
- Casos de Prueba.
- Ciclos de Prueba.
- Ejecuciones.
- Incidencia.
- Métricas.

Cada módulo, permite la creación, edición y eliminación de elementos contenidos en dicho modulo. Además, permite la búsqueda de estos elementos, mediante diversos filtros.

El modulo Ejecuciones, contiene los distintos ciclos de prueba generados para cada proyecto, según criterio del Analista de Pruebas, quien lo crea. En cada uno de estos ciclos se ejecutan las pruebas definidas y se indica el avance de ejecución, así como también, de éxitos y fallas en las mismas. En caso de encontrar una falla durante una ejecución, es posible registrar una Incidencia asociada a la misma.

El modulo Incidencias, está compuesto por cada defecto u observación registrado durante las ejecuciones. En él, se indica el tipo de Incidencia, la severidad, la prioridad, el estado, el responsable y el proyecto al cual está asociada.

Cada usuario al cual se le asigna una incidencia, puede modificarla y derivarla a otro usuario para el tratamiento y/o seguimiento. Se registra la trazabilidad de cada incidencia en una base de datos.



El módulo Métricas, contiene las diversas categorías para medir la calidad de los entregables para cada proyecto registrado. Las categorías definidas son:

- Métricas Generales: indican por proyecto, el total de ciclos de prueba, mostrando para cada uno la cantidad de pruebas, las ejecuciones exitosas, las fallidas y las que restan ejecutar del ciclo. Además, muestra por Tester, las ejecuciones exitosas y las ejecuciones fallidas para cada proyecto.
- Métricas de Incidencias: indican por proyecto, mediante gráficos de torta, las incidencias por persona asignadas, por tipo, por severidad y por estado en el que se encuentra.
- Métricas de Amplitud – Cobertura: es un reporte que contiene, por proyecto y por ciclo de prueba, la información del proyecto, los recursos asignados, estadísticas por requerimiento y métricas resumen: CPE, CPR y CPRT.
- Métricas de Madurez de Pruebas: es un reporte que contiene, por proyecto y por ciclo de prueba las siguientes métricas: CPF, CPSR, CPR, CPFR.
- Métricas de Perfiles de Fallos: es un reporte grafico por proyecto y por ciclo de prueba donde se visualiza el porcentaje de defectos por tipo de severidad y el número de defectos por tipo de severidad.
- Métricas de Densidad de Defectos: es un gráfico informativo por proyecto de los estados de las Incidencias registradas en el proyecto a medida que transcurren los ciclos de prueba.

La aplicación cuenta con seguridad basada en Usuarios y Grupos de Usuarios, cada usuario pertenece a uno o más grupos, los cuales tienen habilitadas determinadas características de acción dentro de cada módulo.

Los grupos definidos son:

- ✓ Administradores
- ✓ Analistas Funcionales
- ✓ Desarrolladores
- ✓ Testers



Módulo 1: Introducción

En el primer Módulo, INTRODUCCION, se presenta a la organización HARRIAGUE Y ASOCIADOS, donde se realizó el análisis e investigación que dieron lugar al desarrollo de nuestra aplicación. Se especifica la visión y misión de la empresa, como así también la situación problemática de la misma. En cuanto a nuestro proyecto de grado, se define el alcance, los objetivos y la estructura del mismo.



Introducción

Harriague + Asociados es una empresa de servicios tecnológicos, fundada en 1994. Se especializa en brindar Soluciones Integrales en Tecnología Informática

Es una empresa de software extremadamente creativa y talentosa que diseña e implementa soluciones para tecnologías de la información.

Ayuda a las compañías a incorporar tecnologías informáticas con el objetivo de hacerlas más competitivas.

Desarrolla e integra soluciones con tecnología para el desarrollo de negocios/empresas competitivas de manera medible.

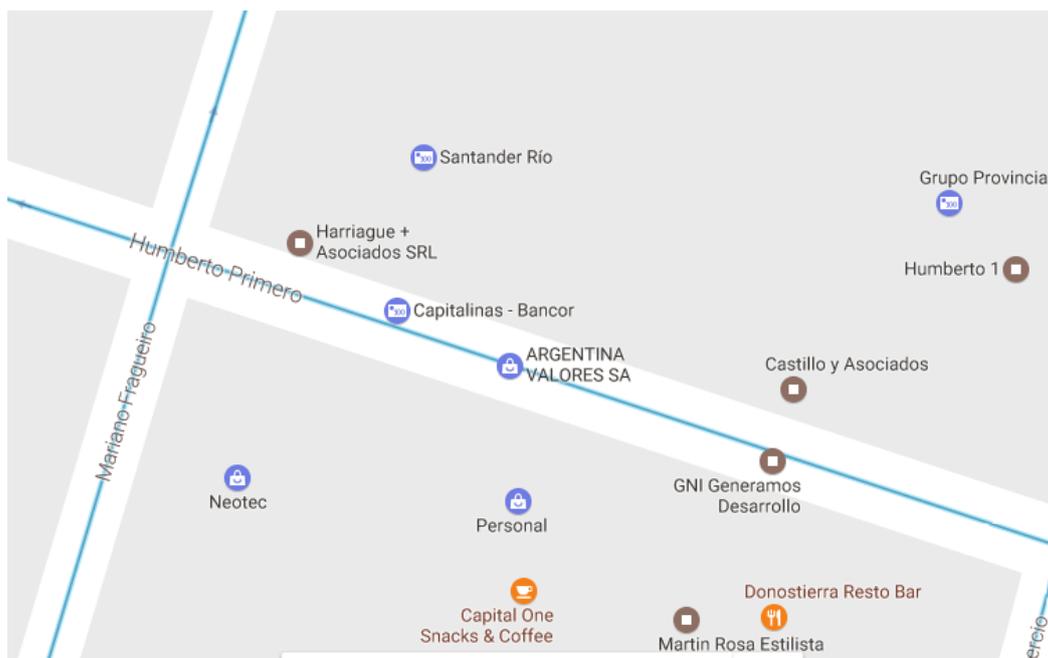
Posee 3 oficinas en Argentina, 1 en EEUU, trabajando con clientes en Latinoamérica y EEUU.

Harriague + Asociados, trabaja con metodología Agile, cuenta con certificación internacional ISO y sus colaboradores trabajan bajo el método SCRUM.

Las oficinas de Harriague +Asociado están ubicado físicamente en:

- **Buenos Aires:** Av. L.N Alem 619, Piso 9.Teléfono +54 11 51990891
- **Córdoba:** Humberto Primo 670, Torre Fragueiro, F54, Capitalinas.
Teléfono +54 351 6387560.

Ubicación oficinas de Córdoba:





MISION DE LA ORGANIZACION

La Misión de la organización es:

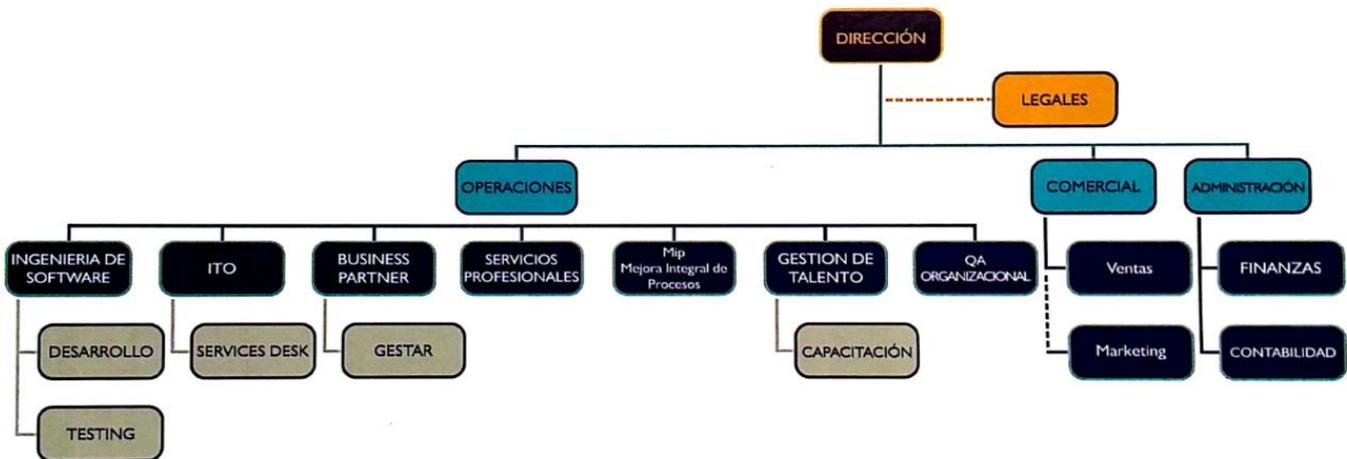
- Ser el socio más confiable para los clientes con un fuerte foco en la satisfacción de servicios globales de tecnología.
- Ser la primera opción para los clientes.
- Ser la primera opción para empresas integradoras internacionales.
- Ser la primera elección de los colaboradores por la calidad de trabajo, la posibilidad de crecimiento profesional y económico en una atmósfera de trabajo estimulante y agradable.

VISION DE LA ORGANIZACION

La Visión de esta organización es ser el socio más confiable de cada uno de sus Clientes, a través de la excelencia de los servicios tecnológicos prestados.



ORGANIGRAMA DE LA EMPRESA



SITUACION PROBLEMÁTICA.

El software se ha convertido en la actualidad, en uno de los principales objetivos estratégicos de las organizaciones debido a que, cada día, los procesos más importantes de las organizaciones y su supervivencia dependen del funcionamiento del mismo.

La calidad del software se puede observar en una característica o atributo. Cuando se habla de atributo, la calidad se refiere a características mensurables, es decir, cosas que se pueden comparar para conocer estándares, como longitud, color, propiedades y maleabilidad. Sin embargo, el software que es una entidad intelectual, tiene la complejidad de caracterizar los objetos físicos. Existen mediciones que nos permiten evaluar las características de un programa.

Cuando se examina un elemento, sus características mensurables se pueden encontrar dos tipos de calidad:

- Calidad de diseño: La calidad de diseño se refiere a las características que los diseñadores especifican para un elemento.
- Calidad de concordancia: La calidad de concordancia es el grado en el que las especificaciones de diseño se aplican durante la fabricación.

Satisfacción del usuario = producto manejable + buena calidad + entrega dentro del presupuesto y tiempo.

La implementación de un Sistema de Gestión de Calidad implica que permita lograr la calidad de los productos y servicios que se ofrecen al cliente. A esto se llega mediante



la planificación, control y mejora de aquellos elementos que influyen en satisfacción del cliente y en el logro de los resultados deseados por la organización.

Actualmente el Sistema de Gestión de Calidad de la empresa, se encuentra alineado a los requerimientos de la Norma ISO 9001:2008.

Actualmente la organización desarrolla productos de software para distintos clientes, brindando servicios a demanda o bien desarrollando proyectos completos que requieren ser entregados con calidad y cumpliendo con las expectativas del cliente.

Para medir la calidad del producto entregado se realizan pruebas manuales, las cuales se diseñan en una aplicación que constituye una solución completa para gestionar tareas en un equipo de trabajo. Es un sistema de gestión de incidencias que permite llevar un control completo de cada proyecto, donde se almacenan los casos de pruebas asociados a los casos de uso o no; y además registrar los defectos que puedan llegar a surgir de la ejecución de esas pruebas.

La implementación de este software permitió que las pruebas estuvieran organizadas en ciclos de pruebas, que puedan trabajar en forma simultánea, en el mismo proyecto, testers y desarrolladores; evitando que se pierda la trazabilidad de los defectos encontrados con los casos de pruebas y con los requerimientos.

Actualmente la empresa genera reportes manuales de métricas de calidad del producto. Dichas métricas se comparten con líderes de proyecto o referentes del lado del cliente y son requeridas por auditorías de calidad realizadas en la empresa. Estos reportes, al generarse manualmente, producen una utilización ineficiente del tiempo.

Justificación

El proyecto de Sistema de Gestión de Calidad de Software teniendo en cuenta la definición de Métricas de calidad, es factible desde el punto de vista de la importancia de realizar estos informes porque se dice que cuando se puede medir aquello de lo cual se está hablando y se puede expresar en números, se sabe realmente acerca de ello; pero cuando no puede medirse, y no puede expresarse en números, el conocimiento que se tiene de ello es escaso e insatisfactorio.

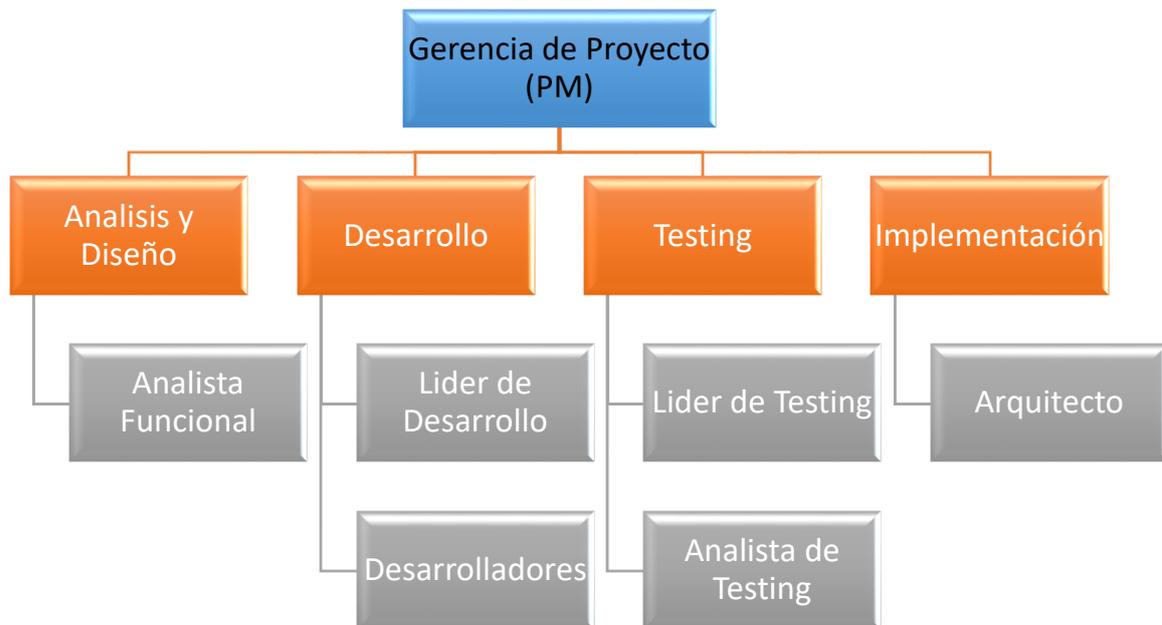
La producción de software es una actividad económica que se caracteriza por generar un alto valor agregado y aportar a la economía productos y servicios esenciales para su modernización.

El contar con datos estadísticos de métricas de software, da un panorama de situaciones reales que ayudan a aplicar y dar seguimiento a las diferentes formas de evaluar y

determinar métricas de calidad para un mejor desempeño en la calidad de software, logrando tener un control de calidad de los ciclos de pruebas que se ejecutan para adquirir así la calidad deseada para el producto final que se entregue.

Alcance

Áreas de la empresa involucradas en el Sistema de Gestión Calidad de Software.



Los proyectos de software de la empresa Harriague y Asociados se llevarán a cabo en las provincias de Buenos Aires y Córdoba.

Objetivos

OBJETIVO GENERAL DEL SISTEMA

El objetivo general del sistema es proveer de métricas para valorar la calidad de los productos que se desarrollan. Se aplica a todo ciclo de vida permitiendo descubrir y corregir aquellos problemas que sean catalogados como potenciales. Es una manera más sistemática de valorar a la calidad, ya que se basa en un conjunto de reglas que son claramente definidas.

OBJETIVOS ESPECIFICOS DEL SISTEMA

1. Proporcionar indicador que permita medir la amplitud o cobertura de las pruebas, indicando la complejidad del plan de pruebas.
2. Proporcionar indicador del buen desempeño del flujo de trabajo de pruebas. Se debe enfocar en la completitud de los casos de pruebas que se definan para



cubrir los requerimientos; y comprender casos de pruebas con resultados satisfactorios.

3. Definir categorías de fallos proporcionando la descripción del error para analizarlo estadísticamente.
4. Analizar estadísticamente la integridad y madurez del software finalizadas las pruebas.



Módulo 2: Marco Teórico

En este segundo Módulo, MARCO TEORICO, se hace una revisión de que es la Calidad, la historia de la misma, mencionando los distintos autores que han aportado para su definición. Se especifica también, lo que es la Calidad Total, Validación y verificación, el concepto de requerimientos, inspecciones, Casos de pruebas y tipo de casos de pruebas. Y por último se define y se hace hincapié, en Métricas, indicando los usos de las mismas, los tipos de métricas de acuerdo a los distintos autores y los factores q afectan directa e indirectamente a la Calidad del producto de software.



CALIDAD, ¿Qué es?

Actualmente se construyen proyectos y productos que no alcanzan los mínimos de "Calidad" esperado lo que inducen a la no finalización con éxito de los mismos.

La calidad es innegociable, es una parte inherente e inseparable del producto. No es un término fácil de definir, ya que la palabra en sí misma tiene múltiples significados:

- a) Se define a la Calidad como la idoneidad o aptitud para el uso que se determina por las características del producto que el usuario puede reconocer beneficioso para él.
- b) Es conseguir la satisfacción de las exigencias siempre en primer lugar.
- c) La calidad se define como la total y completa satisfacción de las necesidades y expectativas de los clientes-usuarios-consumidores, con los menores costos posibles.
- d) Es la totalidad de las características de una entidad (ya sea un producto o servicio), que le confiere la aptitud para satisfacer las necesidades establecidas e implícitas (ISO 9000).

Otras definiciones de organizaciones reconocidas y expertos del mundo de la Calidad son:

- Juran: Dice que la CALIDAD es la adecuación al uso del cliente.
- Crosby: Cumplimiento de una de las especificaciones.
- W. Deming: Dice que la calidad es un grado predecible de uniformidad y fiabilidad a bajo coste y adecuado a las necesidades del mercado.
- Norma ISO/IEC 8402: Dice que la calidad es la totalidad de las características de una entidad que le confiere a la aptitud para satisfacer las necesidades establecidas e implícitas.
- Por su parte Drucker, manifiesta que la calidad es lo que el cliente está dispuesto a pagar en función de lo que obtiene y valora.
- Feingembaun, define a la CALIDAD como un conjunto de características del producto de marketing, ingeniería, fabricación, mantenimiento a través del cual el producto en uso satisface las expectativas del cliente.
- SHEWART: Calidad como resultado de la interacción de dos dimensiones: dimensión subjetiva (lo que el cliente quiere) y dimensión objetiva (lo que se ofrece).

También se puede clasificar la definición de la Calidad en:



- ✓ **Trascendente:** Significa que la calidad no es tomada ni como espíritu ni materia, sino q es una tercera entidad que es independiente a ellas, se dice que la calidad no puede definirse, pero que todo el mundo sabe lo que es.
- ✓ **Basada en el producto:** Cuando existe diferencias en la calidad se dice que existen diferencias en la cantidad de alguno de los ingredientes o atributos deseados.
- ✓ **Basada en el usuario,** este "tipo de Calidad" consiste en la capacidad de satisfacer las expectativas.
- ✓ **Basada en la fabricación,** se entiende como la conformidad del producto con los requerimientos.
- ✓ **Basada en su valor,** es el grado de excelencia con el precio aceptable y un control de variabilidad a un costo aceptable.

Teniendo en cuenta las definiciones anteriores, podemos definir a la Calidad como la suma de todos los aspectos o características de un producto o servicio que influyen en su capacidad para satisfacer las necesidades del usuario, la cual está determinada por la diferencia entre calidad percibida y calidad esperada, cuando el usuario hace uso del producto o servicio.

La calidad es más que un producto funcionando. Los principales elementos del paradigma de la calidad son:

1. *Naturaleza de la calidad:* Es la orientación de los aspectos o características de un producto o servicio que influyan en su capacidad para satisfacer las necesidades dadas.
2. *Orientación de datos:* Es basada en la recolección, análisis y comparación de datos.
3. *Focalización en el cliente o usuario:* El objetivo de la calidad es la obtención de la satisfacción del cliente o usuario.
4. *Eliminación de defectos:* Es la disminución de costes de no calidad.
5. *Gestión para la Calidad:* Se debe adoptar un paradigma que requiera el compromiso de la alta dirección.

Historia de la calidad

Desde los primeros tiempos el hombre ha controlado la calidad de los productos que consumía, pudiendo discriminar entre los que podía comer y aquellos que no.



En la edad media, el mantenimiento de la calidad se lograba, a través de los prolongados periodos de capacitos que exigían los gremios a los aprendices, lo que lograba que los trabajadores se sintieran orgullosos por la obtención de productos de calidad.

La revolución industrial vio surgir el concepto de "Especialización laboral". El trabajador no tuvo a su cargo la fabricación total de un producto, sino solo una parte de éste. Este cambio trajo el deterioro

en la calidad de la mano de obra, debido a que la mayor parte de los productos que se fabricaban no eran complicados por lo que la calidad no se vio mayormente afectada. A medida que los productos se fueron complicando y las respectivas labores se fueron haciendo cada vez más especializadas, fue necesario revisar la fabricación de los productos.

En 1924 Shewhart de Bell Telephone Laboratories diseñó una gráfica de estadísticas para controlar las variables de un producto, iniciando así la era del control estadísticos de la calidad.

En el año 1950, W Edwards Deming ofreció conferencias a los ingenieros japoneses sobre los métodos estadísticos y sobre la responsabilidad de la calidad al personal gerencial. Su pensamiento se basa en 14 puntos entre los que se incluyen tres ingredientes de calidad: mejora continua, propósito constante y conocimiento profundo. Juran por su parte, en 1954, contribuyó a destacar el importante compromiso del área gerencial por el logro de la calidad, las capacitaciones del personal en la gestión para la calidad y que se mejore la calidad a un ritmo sin precedentes valiéndose de estos conceptos.

Philip Crosby, creó el movimiento cero defectos durante la década de 1960, promoviendo el concepto de hacer las cosas correctamente desde el principio.

En 1962 Kaoru Ishiwaka, constituyó los Círculos de Control de Calidad en Japón para lograr el mejoramiento de la calidad.

En noviembre del mismo año, JUSE (Unión of Japanese Scientists and Engineers) organizó la primera conferencia de control de calidad para los supervisores de nivel operativo.

El final de los años 70's y el principio de los 80's fue marcado por un empeño en la calidad en todos los aspectos de los negocios y organizaciones de servicios, incluyendo las finanzas, ventas, personal, mantenimientos, administración, fabricación y servicio. La reducción en la productividad, los altos costos, huelgas y alto desempleo hicieron que la administración se volviera hacia el mejoramiento de la calidad como medio de supervivencia organizacional.



La Organización Internacional de Normas ISO creada desde hace más de cinco décadas, desde su fundación su propósito fue mejorar la calidad, aumentar la productividad, disminuir los costos e impulsar el comercio internacional. De este organismo surgen la familia de normas ISO 9000, que están integradas por un conjunto de modelos y documentos sobre la Gestión de Calidad. En 1987 se publicaron las normas internacionales actuales sobre aseguramiento de la calidad. Por primera vez, cada una de ellas sirve como un modelo de calidad dirigido a determinada área de la industria, la manufactura o los servicios. En la actualidad cubren todas las funciones o posibilidades de desempeño, y tienen el objetivo de llevar la calidad o la productividad de los productos o servicios que se oferten al más alto nivel. Aunque los antecedentes más remotos de la existencia de la norma ISO 9000 datan de hace más de 50 años, es importante destacar que la aceptación internacional de la normalización ha tenido vigencia, sobre todo, a partir de la década de 1980.

14 Principios de Deming:

1. **Crear constancia de mejorar el producto y servicio**, con el objetivo de ser competitivos, permanecer en el negocio proporcionando puestos de trabajo.
2. **Adoptar la nueva filosofía**, es decir se necesita una nueva filosofía en donde los errores resulten inaceptable, porque estos ocultan costos que están contenidos en el reproceso y que luego se destacan en el cálculo de los costos total de la calidad.
3. **Dejar de confiar en la Inspección masiva**. Existen empresas anticuadas que inspeccionan un producto cuando éste sale de la línea de producción o en etapas importantes del camino, desechando o re elaborando los productos defectuosos. Cualquiera de estas dos prácticas, resultan innecesariamente costosas. En realidad, la empresa les está pagando a los trabajadores para que hagan defectos y luego los corrijan. La calidad NO proviene de la inspección, sino de la mejora del proceso
4. **Poner fin a la práctica de conceder negocios con base en el precio únicamente**. Las áreas de compras funcionan siguiendo la orden de buscar al proveedor de menor precio. Esto, frecuentemente conduce a provisiones de mala calidad. En lugar de ello, los compradores deberían buscar la mejor calidad en una relación de largo plazo con un solo proveedor para determinado artículo.
5. **Mejorar constantemente y por siempre el sistema de producción y servicios**. La mejora no es un esfuerzo que se realiza una sola vez, sino que la dirección debe buscar constantemente maneras de reducir el desperdicio y mejorar la calidad. Además de la mejora continua de los productos, deben



- mejorarse también los sistemas, ya que sin esa mejora difícilmente se alcanzará nuevas metas con los mismos métodos.
6. **Instituir la capacitación**, es decir, se debe capacitar permanente a trabajadores y supervisores en sus propios procesos, de manera que ese aprendizaje ayude a mejorarlos tanto de manera incremental como radicalmente.
 7. **Instituir al liderazgo**. La tarea del supervisor no es decirle a su personal qué hacer, ni amenazarlo ni castigarlo, sino dirigirlo, es decir ayudar al personal a hacer un mejor trabajo y aprender mediante métodos objetivos quién necesita ayuda individual. Los líderes establecen la unidad de propósito y la orientación de la organización. Ellos deberían crear y mantener un adecuado ambiente interno, en el cual el personal pueda llegar a implicarse totalmente en el logro de los objetivos de la organización.
 8. **Eliminar el temor**, esto es porque muchos empleados temen hacer preguntas o asumir una posición, aun cuando no comprendan cuál es su trabajo, ni qué está saliendo bien o mal. Seguirán haciendo las cosas mal o sencillamente no las harán. Las pérdidas económicas a causa del temor son terribles. Para garantizar mejor calidad y más productividad es necesario que las personas se sientan segura. No existe temor al cambio sino a la incertidumbre de lo que pasará con el cambio.
 9. **Derribar las barreras que existen entre las áreas**. Existen oportunidades donde los departamentos o las áreas de las empresas "compiten" entre sí o tienen metas que chocan entre sí. No trabajan como un equipo para resolver o prever sus problemas, esto lleva a la consecución de metas de un departamento pueden causarle problemas a otro.
 10. **Eliminar los lemas, las exhortaciones y las metas de producción para la mano de obra**. Estos lemas y demás cosas no ayudan a los empleados a realizar bien su trabajo, sino q es mejor q los propios trabajadores formulen sus propios lemas.
 11. **Eliminar los cupos para la mano de obra y los objetivos numéricos para la dirección**. Las cuotas solamente tienen en cuenta los números, no la calidad ni los métodos. Son una garantía de ineficiencia y alto costo. La persona, por conservar el empleo, cumple la cuota a cualquier costo, sin tener en cuenta el perjuicio para su empresa. Si se aceptara como una máxima de todo proceso la variabilidad implícita en todos ellos y si todo el esfuerzo se concentrara en reducirla, las metas se alcanzarían solas.
 12. **Remover las barreras que impiden el orgullo de un buen trabajo bien hecho**.



13. Instituir un programa vigoroso de educación y capacitación. Este principio es un complemento del nro. 6, pero éste es más referido a lo que se conoce como "Formación y Desarrollo de Competencias", lo cual depende de la visión de la empresa, de los objetivos para alcanzarla, de las nuevas formas de poder competir, de los nuevos procesos a desarrollar, de los nuevos comportamientos del personal a todos los niveles, en síntesis, del cambio cultural que la empresa requiere.

14. Tomar medidas para llevar a cabo la transformación, la cual no llega sola, la alta dirección debe tomar la decisión de querer hacerlo y aplicar el principio de instituir el liderazgo. Para lograr la transformación debe suceder algo más que llamar al consultor, y nombrar al "representante de la gerencia", significa comprometerse y ser ejemplo, capacitándose primero, y cumpliendo su tarea después.

Trilogía de la Calidad

Juran plantea la calidad como la integración de la planificación, el control y la mejora continua.

- a) **PLANIFICACION DE LA CALIDAD:** Se trata de crear un proceso que sea capaz de cumplir con metas establecidas, para hacerlo obviamente bajo las condiciones de operación. Se elabora el Plan Maestro de la Calidad Total que requiere la organización (para ser funcional debe durar por lo menos 3 años). Para lo cual se identifican quienes son los clientes, si son externos o internos, se determinan cuáles son sus necesidades, crean el producto o servicio que pueda responder a dichas necesidades y a optimización de los procesos que permitan obtener tales productos/servicios que satisfagan estas expectativas, así como los controles a realizar de tal manera que se asegure la producción.
- b) **CONTROL DE LA CALIDAD:** Es vigilar que el proceso opere con efectividad (eficiencia + eficacia) óptima asegurando la fabricación de acuerdo a las especificaciones, debido a que las deficiencias se originan en la operación inicial, los procesos pueden operar con un alto nivel de gasto de desperdicio.
- c) **MEJORA CONTINUA DE LA CALIDAD:** Hace referencia a las actividades de análisis a partir de los resultados de calidad y su aplicación práctica para la mejora. Comprende la identificación de causas de problemas, el establecimiento

de prioridades en relación con los problemas observados, la búsqueda de soluciones y su posterior implantación y seguimiento.

El Diagrama de la Trilogía de Juran



Principios fundamentales de CROSBY

1º Principio: Crosby está estrechamente relacionado con la idea de "Cero defectos". Para él, la calidad es conformidad con los requerimientos, lo cual se mide por el coste de la no conformidad. Al utilizar este enfoque, se llega a la meta de performance de "Cero Defectos".

Cero defectos es hacer lo acordado en el momento acordado, es hacer bien las cosas desde la primera vez, es tomar en serio los requisitos. Los errores son ocasionados por dos causas: la falta de conocimientos y la falta de atención, cosas que deben de ser primordiales que el entrenador corrija o que pueda tratar de solucionar.

2º Principio: "La calidad proviene de la prevención". Es decir, es más fácil y menos costoso prevenir las cosas que corregirlas. La prevención se basa en la comprensión de todo el proceso para llegar a un juego o evento, por lo tanto, hay que observar y determinar las posibles causas de error en dichos eventos, ya sea desde las condiciones de las instalaciones, equipos de seguridad y todo lo que intervenga en la realización del mismo para estar precavidos de posibles fallas y solucionarlas al instante o antes de que sucedan.

3º Principio: "La calidad implica cumplir con los requerimientos". Para que la calidad se vea reflejada en todo el equipo, deben estar consciente de que deben hacer las cosas bien desde la primera vez. Pero para que esto se lleve a cabo, los entrenadores tienen que establecer claramente los requisitos que deben de cumplir cada integrante del equipo; suministrar los medios necesarios para que el equipo cumpla con los requerimientos físicos y anémicos para el óptimo desempeño.



4º principio: "La medición de la calidad es el precio de la inconformidad". Es una forma de evaluar la calidad dentro del equipo costeadando las actividades que se tengan que realizar por haber hecho las cosas mal (incumplimiento de requisitos). Es un punto de retroalimentaciones y la realización de mejoras continuas.

CALIDAD TOTAL

Se basa en la satisfacción del cliente. Implica a todos los miembros de la organización y de todos los aspectos relacionados con la administración de la misma.

La Calidad Total no solo se refiere al producto o servicio en sí, sino que es la mejoría permanente del aspecto organizacional, gerencial. La organización se toma como un gran sistema, donde cada trabajador, desde el gerente, hasta el empleado del más bajo nivel jerárquico está comprometido con los objetivos de la misma.

Para que se logre la Calidad Total el empresario juega un papel fundamental, empezando por la educación previa de sus trabajadores para conseguir una población laboral más predispuesta, con mejor capacidad de asimilar los problemas de calidad, mejor criterio para sugerir cambios en provecho de la calidad, mejor capacidad de análisis y observación del proceso de manufactura en caso de productos y poder enmendar errores.

Importancia de la Calidad Total

La Calidad Total en una organización, debe ser el nervio y motor de la misma. El mensaje de la calidad total se debe comunicar a:

- Los Trabajadores.
- Los Proveedores.
- Los Clientes.

Por otra parte, los fundamentos de la calidad total son:

- El objetivo básico: la competitividad.
- El trabajo bien hecho.
- La mejora continua con la colaboración de todos: responsabilidad y compromiso individual por la calidad.
- El trabajo en equipo es fundamental para la mejora permanente
- Comunicación, información, participación y reconocimiento.
- Prevención del error y eliminación temprana del defecto.
- Fijación de objetivos de mejora.
- Seguimiento de resultados.



- Indicadores de gestión.
- Satisfacer las necesidades del cliente: calidad, precio, plazo.

Existen también obstáculos que impiden el avance de la calidad, los cuales pueden ser:

- ❖ El hecho de que la dirección no defina lo que entiende por calidad.
- ❖ No se trata de hacer bien las cosas, sino de que el cliente opine igual y esté satisfecho.
- ❖ Todos creen en su concepto, pocos en su importancia y son menos los que la practican.

ASEGURAMIENTO DE LA CALIDAD

El aseguramiento de la calidad es un sistema que pone énfasis en los productos, desde su diseño hasta el momento de enviárselo al cliente y concentra sus esfuerzos en la definición de procesos y actividades que permiten la obtención de productos conforme a unas especificaciones.

Es seguir una línea de actuación, dirigida a conseguir trabajar en base a un conjunto de acciones planificadas y sistemáticas, implantadas dentro del Sistema de Calidad de la empresa.

También se podría definir como el diseño y planificación de acciones sistemáticas que se requieren para asegurar la calidad de software.

Son procesos para garantizar que el producto, componente y calidad se mantienen o mejoran, y que los errores de fabricación/ manufactura son reducidos o eliminados. Se puede llevar a cabo de manera interna, o por un organismo independiente.

Es cumplir con las expectativas del consumidor y las regulaciones a fin de evitar defectos costosos.

Los servicios de aseguramiento de calidad y control de calidad (AC/CC) de Bureau Veritas dan soporte a tres importantes sectores: productos de consumo, mercancías (commodities) e industrial, incluyendo la fabricación/ manufactura y embarque.

Los responsables del aseguramiento de la calidad son los ingenieros del software, gerentes y líderes, clientes, y el grupo de aseguramiento de la calidad en sí.

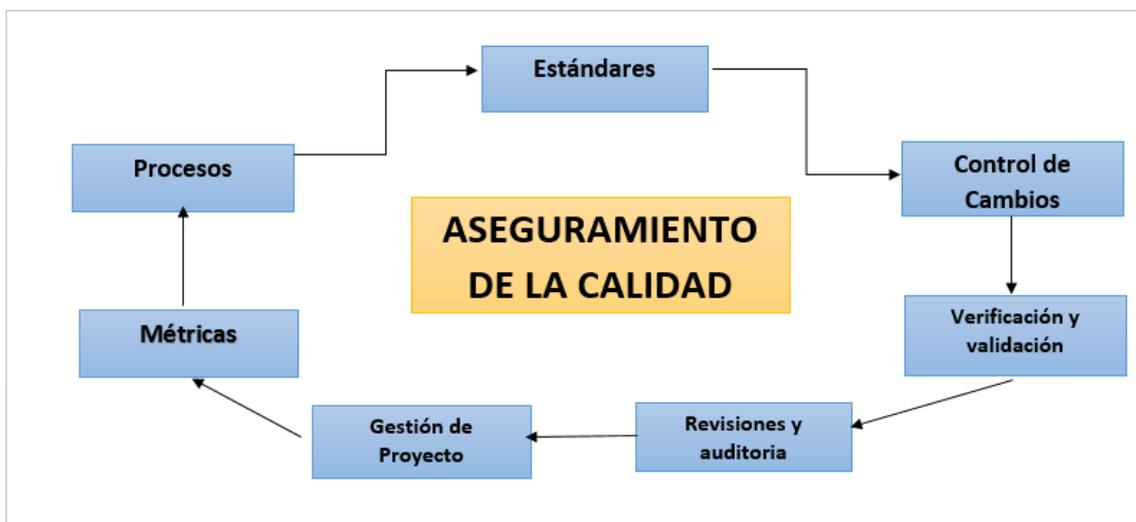
Propósito:

El propósito del aseguramiento de la calidad, es suministrar a la gerencia una visión dentro del proceso que está siendo usado para el proyecto y de los productos que se construyen.

Este aseguramiento implica la revisión y auditoria de los productos y las actividades para verificar que se cumplen con los procedimientos y estándares aplicables y suministrando al gerente de proyecto y otros del resultado de estas revisiones y auditorias.

Propiedades del Aseguramiento de la calidad

- ❖ Las visiones son las que permiten definir los modelos de Calidad, son el conjunto de características.
- ❖ Los modelos de calidad permiten definir propiedades del producto y proceso
- ❖ En base a las propiedades se definen las métricas que son los indicadores de calidad.
- ❖ En base a estas métricas se realiza la gestión de calidad asegurada.



VALIDACION Y VERIFICACION-V&V

La *verificación y validación* es el nombre que se da a los procesos de comprobación y análisis que aseguran que el software que se desarrolla está acorde a su especificación y cumple las necesidades de los clientes. La V&V es un proceso de ciclo de vida completo. Inicia con las revisiones de los requerimientos y continúa con las revisiones del diseño y las inspecciones del código hasta la prueba del producto. Existen actividades de V&V en cada etapa del proceso de desarrollo del software.

La verificación y la validación no son la misma cosa, aunque es muy fácil confundirlas.

Se define **verificación** como el proceso de evaluación de un sistema o componente para determinar si un producto de una determinada fase de desarrollo satisface las condiciones impuestas al inicio de la fase.

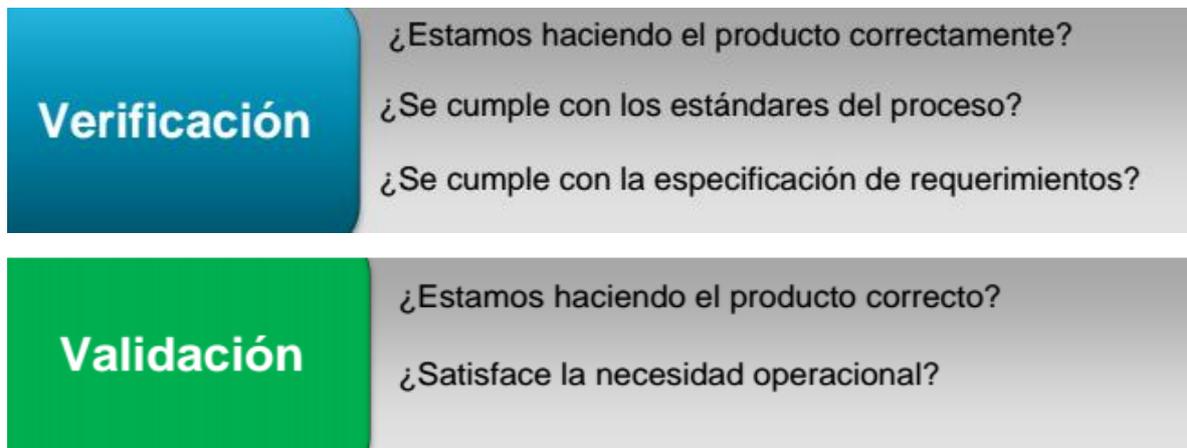


Es el proceso de evaluar, revisar, inspeccionar, y hacer controles de escritorios de productos intermedios tales como especificaciones de requerimientos especificaciones de diseño y código. En el caso de código se refiere a un análisis estático del mismo y no de su ejecución dinámica.

El propósito de la verificación, es asegurar que los productos internos seleccionados cumplen con su especificación de requerimientos funcionales y no funcionales. Los métodos de verificación pueden ser entre otros: inspecciones, revisiones por pares, auditorias, recorridas, análisis, simulaciones, pruebas y demostraciones.

La **validación** es el proceso de evaluación de un sistema o componente durante o al final del proceso de desarrollo para determinar cuándo se satisfacen los requerimientos especificados. Es un proceso más general. Se debe asegurar que el software cumple las expectativas del cliente. Va más allá de comprobar si el sistema está acorde con su especificación, para probar que el software hace lo que el usuario espera a diferencia de lo que se ha especificado.

Es importante llevar a cabo la validación de los requerimientos del sistema de forma inicial. Es fácil cometer errores y omisiones durante la fase de análisis de requerimientos del sistema y, en tales casos, el software final no cumplirá las expectativas de los clientes.



Técnicas de VERIFICACION Y VALIDACION

a) Inspecciones

Son revisiones técnicas, donde se inspecciona una parte de un componente. Son fundamentales para el aseguramiento de la calidad, ya que establecen un orden en el proceso al mismo tiempo que se establece una mejora continua.



También son denominados como un proceso de mejora incremental, debida que a medida que pasa el tiempo en el desarrollo de software los defectos deben reducirse.

Las inspecciones analizan y comprueban las representaciones del sistema como el documento de requerimientos, los diagramas de diseño y el código fuente del programa. Se aplica a todas las etapas del proceso de desarrollo. Se complementan con algún tipo de análisis automático del texto fuente o de los documentos asociados. Las inspecciones del software y los análisis automatizados son técnicas de verificación y validación estáticas puesto que no requieren que el sistema se ejecute.

Los objetivos de las inspecciones son:

- Encontrar los defectos en forma temprana.
- Prevenir el mal funcionamiento de los procesos o planes que se establecieron.
- Proporcionar mejoras en la fiabilidad, disponibilidad y factibilidad del mantenimiento del software.
- Descubrir continuamente la información técnica, asociada con funciones, formularios, actividades internas que permiten asegurar el producto.
- Establecer una igualdad de conocimiento dentro de los desarrolladores para la buena práctica de estándares y técnicas de desarrollo.

Proceso de inspección

El proceso de inspección de software involucra la interacción de cinco elementos:

- Pasos: Los pasos a seguir en las inspecciones son:
 1. Planificación
 2. Conferencia Introductoria.
 3. Preparación.
 4. Reunión.
 5. Resultados: lista de defectos encontrados y reporte para la gerencia.
 6. Retrabajo.
 7. Seguimiento.
- Roles: Las inspecciones se realizan por grupos pequeños de 3 a 6 personas. La selección de los inspectores está restringida a un grupo de desarrolladores que trabajan componentes similares o tiene interacción con el producto.
- Recopilación de datos: Permite mediante el análisis y uso de métricas específicas, mejorar la calidad del proceso de desarrollo de software.



- Producto a inspeccionar: Producto puede ser intermedio o en desarrollo, no es el producto final que será entregado luego al cliente.
- Infraestructura de soporte: Las inspecciones deben ser planificadas y apoyadas institucionalmente, designando a un responsable para que administre las mismas.

Ventajas

Los beneficios de las inspecciones pueden ser:

- *Directos*
 1. Mayor calidad, lo que permite aumentos de la productividad.
 2. Disminuyen los tiempos de prueba.
 3. Encuentra la causa directamente y no el efecto.
- *Indirectos*
 1. Capacitaciones.
 2. Mayor visibilidad del proceso
 3. Trabajo en equipo, mejor comunicación.
 4. Mejora en la calidad de estándares y métodos.

b) Pruebas de Software

Prueba es una actividad realizada para evaluar la calidad del producto y mejorarla, identificando defectos y problemas.

La palabra probar en software, significa ejecutar un programa con el objetivo de encontrar defectos en él. La prueba es considerada exitosa, si encuentran fallas. Para lograr esto se ejecutan un conjunto finito de casos de pruebas, que son seleccionados de manera adecuada desde el dominio de ejecución.

Consiste en contrastar las respuestas de una implementación del software a series de datos de prueba y examinar las respuestas del software y su comportamiento operacional, para comprobar que se desempeñe conforme a lo requerido. Llevar a cabo las pruebas es una técnica dinámica de la verificación y validación ya que requiere disponer de un prototipo ejecutable del sistema.

La prueba es tomada como parte del proceso de Verificación y Validación, para lo cual se definen:



- ✓ *Prueba para la verificación:* El comportamiento observado es revisado contra las especificaciones.
- ✓ *Prueba para la validación:* El comportamiento observado es revisado contra las expectativas del usuario.

Requerimientos y su relación con la prueba

Los requerimientos son los que el software debe hacer y/o características que el software debe poseer. Deben ser consistentes, completos, realizables y verificables.

Estos requerimientos expresan las necesidades y las restricciones puestas en un producto de software que contribuyen a la solución de un cierto problema del mundo real. La característica esencial de todos los requerimientos del software es que sean comprobables.

Se pueden distinguir entre requerimientos Funcionales que son los que describen las funciones del programa que debe realizar y los No Funcionales que son las restricciones a las posibles soluciones. Estos últimos se pueden clasificar si son requerimientos de desempeño, seguridad, confiabilidad, etc.

Estos requerimientos pueden validarse para asegurarse de que se han entendido, y es conveniente que el documento de requerimientos este conforme a los estándares de la organización y sea entendible, consistente y completo.

La validación de los requerimientos se refiere al proceso de examinar el documento de requerimientos para asegurarse de que define el software correcto, es decir, el software que los usuarios esperan. Una característica esencial de un requerimiento de software es que debe ser posible validar que el producto final lo satisface.

Elementos de la Prueba

- ❖ **Prueba:** Es una actividad en la que un sistema o componente es ejecutado bajo ciertas condiciones específicas, los resultados son observados o registrados y una evaluación es hecha de algún aspecto del sistema o componente.
- ❖ **Un caso de prueba:** Es un conjunto de valores de entrada, precondiciones de ejecución, resultados esperados y pos condiciones de ejecución, desarrollados con un objetivo particular o condición de prueba, tal como ejercita un camino de un programa particular o para verificar que se cumple un requerimiento específico.
- ❖ **Resultado Esperado:** Es el comportamiento precedido por la especificación u otra fuente, del componente o sistema a ser probado bajo condiciones específicas.
- ❖ **Ciclo de Pruebas:** Es la ejecución del proceso del testigo contra una versión identificada del producto a probar.



- ❖ **Datos de Prueba:** Son los datos que existen (por ejemplo, en una base de datos) antes de que una prueba sea ejecutada, y que afecta o es afectado por el componente o sistema a probar.
- ❖ **Ejecución de la prueba:** es el proceso de ejecutar una prueba para el componente o sistema a probar, produciendo el resultado real.

Clasificación de la prueba

Existen distintas formas de clasificar los tipos de pruebas. La clasificación se basa en el tipo de pruebas que se está haciendo: funcional o estructural y según como se realiza la prueba unitaria, de integración o del sistema.

- ✚ **Prueba estructural:** Es también llamada prueba de caja blanca o basada en el código. El objetivo de esta prueba es seleccionar los caminos del programa a ejecutar durante las pruebas.
 - Estas pruebas se seleccionan en función del conocimiento que se tiene de la implementación del módulo. Hay veces que se aplican a módulos pequeños. El tester que realiza esta prueba, analiza el código y deduce cuantos y que conjunto de valores de entrada han de probarse para que al menos se ejecute una vez cada sentencia del mismo. También es posible refinar los casos de pruebas que se identificaron con las pruebas de caja negra.
- ✚ **Prueba Funcional:** Es conocida también como basada en la especificación o de caja negra. El objetivo de estas pruebas es validar si el comportamiento observado del software probado cumple o no con sus especificaciones. La prueba funcional toma el punto de vista del usuario. Lo que trata es mostrar errores y discrepancias con la especificación y no demostrar que el programa cumple con su especificación.

En este tipo de pruebas las funciones son probadas ingresando las entradas y examinando las salidas, la estructura interna del programa raramente es considerada.

Las pruebas funcionales son una categoría que permite seleccionar las pruebas de fallos basándose en las especificaciones de los componentes y los programas y no del conocimiento de su implantación.
- ✚ **Prueba unitaria o prueba de modulo:** Es el proceso de probar los componentes individuales (subprogramas o procedimientos) de un programa. El propósito es descubrir discrepancias entre la especificación de la interface de los módulos y su comportamiento real.



- Una unidad es la pieza de software más pequeña que se puede probar, contiene cien o menos líneas de código

✚ **Prueba de Integración:** Es el proceso en el cual los componentes son agregados para crear componentes más grandes. Es el testing realizado para mostrar que, aunque los componentes hayan pasado satisfactoriamente las pruebas unitarias, la combinación de componentes es incorrecta o insatisfactoria.

- Se prueban la respuesta de grupos de módulos interconectados a fin de detectar fallos resultantes de la interacción entre los componentes.

Estas pruebas se realizan con referencia a las especificaciones del problema. La principal dificultad de las pruebas de integración es la localización de fallos. Existen interacciones complejas entre los componentes del sistema y cuando se descubre una salida anómala, es difícil encontrar la fuente de error.

Para facilitar la detección de los errores se utilizan técnicas incrementales.

Una vez que se prueban los componentes individuales del programa, se deben integrar para crear un sistema parcial o completo. En el proceso de integración se debe probar el sistema resultante con respecto a los problemas que surjan de las interacciones de los componentes.

Las pruebas de integración se desarrollan a partir de la especificación del sistema y se inicia tan pronto como estén disponibles versiones utilizables de algún componente del sistema.

✚ **Pruebas del sistema,** esta prueba refiere al comportamiento del sistema entero. La mayoría de las faltas funcionales se deben haber identificado durante las pruebas de unidad e integración. Estas pruebas se consideran apropiadas para probar los requerimientos no funcionales del sistema, tales como seguridad, desempeño, exactitud y confiabilidad. Las interfaces externas, los dispositivos de hardware o el ambiente de funcionamiento también son evaluadas en este tipo de pruebas.

✚ **Prueba de Volumen:** Este tipo de prueba del sistema somete el programa a grandes volúmenes de datos. Su propósito es demostrar que el programa no puede manejar el volumen de datos especificados en sus objetivos. Este volumen a probar puede requerir recursos significativos, en término del tiempo del procesador y de personas.

✚ **Pruebas de estrés:** Estas pruebas son las que someten al programa a cargas pesadas. Una carga pesada es un volumen máximo de datos o de actividad en un plazo corto de tiempo. La prueba de estrés implica como elemento el tiempo.



- ✚ **Pruebas de Usabilidad:** Son las que intentan encontrar problemas de usabilidad del sistema en su interacción con humanos. El análisis de factores humanos es una cuestión altamente subjetiva.
- ✚ **Pruebas de usabilidad:** Intentan encontrar problemas de usabilidad del sistema en su interacción con humanos.
- ✚ **Pruebas de Seguridad:** Intentan subvertir los chequeos de seguridad del sistema. Una forma de pensar tales casos de pruebas es estudiar problemas conocidos de seguridad en sistemas similares.
- ✚ **Pruebas de desempeño:** Muchos programas tiene objetivos específicos de desempeño o de eficiencia, indicando características tales como tiempos de respuestas y rendimiento bajo ciertas condiciones de carga de trabajo y configuración. En las pruebas de desempeño los casos de prueba se deben diseñar para demostrar que el programa no satisface sus objetivos de desempeño.
- ✚ **Pruebas de almacenamiento:** Algunos programas tienen establecidos límites en el almacenamiento que indican, por ejemplo, la cantidad de memoria principal y secundaria que el programa utiliza. Se diseñan los casos de pruebas para demostrar que estos objetivos de almacenamiento no se han resuelto.
- ✚ **Pruebas de configuración:** Son las que prueba el programa con distintas configuraciones del hardware y software disponibles
- ✚ **Pruebas de compatibilidad/configuración/conversión:** La mayoría de los programas que son desarrollados no son totalmente nuevos, sino que son reemplazo de algún sistema existente. Algunas veces, estos programas tienen objetivos específicos referentes a su compatibilidad y procedimientos de conversión del sistema existente. En algunas oportunidades estos programas tienen objetivos específicos referentes a su compatibilidad y procedimiento de conversión del sistema existente. Los casos de pruebas son diseñados para demostrar que los objetivos de la compatibilidad no se han resuelto y que los procedimientos de conversión no funcionan.
- ✚ **Pruebas de Instalación:** La prueba de instalación es una parte importante del proceso de prueba del sistema, particularmente en un sistema empaquetado con instalación automatizada.
- ✚ **Pruebas de Confiabilidad:** Estos tipos de pruebas tiene como objetivo la mejora de la confiabilidad del programa, probando que las declaraciones específicas sobre confiabilidad se cumplen.
- ✚ **Pruebas de Recuperación:** Algunos programas tienen requerimientos específicos de recuperación que indican como el sistema se recupera de errores



e programación, de faltas del hardware y de errores en los datos. Un objetivo de la prueba del sistema es demostrar que estas funciones de recuperación no funcionan en forma correcta.

- ✚ **Pruebas de mantenimiento:** Algunos programas tienen objetivos de mantenimiento, como por ejemplo los requerimientos específicos sobre las ayudas que se proporcionarían al sistema, los procedimientos de mantenimiento y la calidad de la documentación de la lógica interna.
- ✚ **Pruebas de regresión:** El objetivo de estas pruebas es verificar que no ocurrió una regresión en la calidad del producto luego de un cambio, asegurándose que los cambios no introducen un comportamiento no deseado u errores adicionales. Implican la re ejecución de alguna o todas las pruebas realizadas anteriormente.
- ✚ **Pruebas de Humo:** Son un conjunto de pruebas aplicadas a cada nueva versión, cuyo objetivo es validar que las funciones básicas de la versión se cumplen según lo especificado. Estas pruebas buscan grandes inestabilidades o elementos clave faltantes o defectuosos, los que hacen imposible realizar la prueba como fue planificada para la versión no pasa las pruebas de humo, no se comienza la ejecución de las pruebas planificadas de la misma.

TECNICAS DE PRUEBAS

A. Técnicas de Caja Negra

Son llamadas también técnicas de prueba funcional, en donde los casos de prueba se derivan a partir de la especificación del sistema.

1. *Partición de Equivalencia:* Decimos que un buen caso de prueba es aquel que tiene una probabilidad razonable de encontrar un error y dado que la prueba exhaustiva es imposible, las pruebas se limitan a intentar con un subconjunto pequeño de todas las entradas posibles. Con esta técnica, se intenta seleccionar un subconjunto con la probabilidad más alta de encontrar la mayoría de los errores y que cumplan con las siguientes características:
 - Reducir el número de otros casos de prueba que se deben desarrollar. Esto implica que cada caso de prueba debe invocar tantas entradas como sea posible para reducir al mínimo el número total de casos de pruebas necesarios.
 - Cubrir otros casos posibles de prueba, es decir, se debe intentar repartir el dominio de la entrada del programa en un número finito de clases de equivalencias donde se pueda asumir que probar un valor representativo



de cada clase es equivalente a probar cualquier otro valor. Es decir, si un caso de prueba encuentra un error o incidente, se espera que el resto de los casos de prueba en la clase de equivalencia encontraran el mismo error.

El diseño de los casos de prueba siguiendo partición de equivalencia se divide en dos pasos:

- Identificación de las clases de equivalencia: Son identificadas tomando cada condición de la entrada (una oración, una frase en la especificación) y repartiéndola en dos o más grupos. Dos tipos de clases de equivalencia son identificados: Las clases de equivalencia validas que representan las entradas cálidas al programa y las clases de equivalencias invalidas, que representan al resto de los estados posibles (es decir son valores erróneos de entrada).
 - Definición de los casos de prueba: A partir de las clases de equivalentes se identifican los casos de pruebas. Se escriben tantos casos de pruebas que cubran cada clase de equivalencia validas como sea posible y casos de prueba que cubran clases de equivalencias invalidas.
2. *Análisis del valor límite*: Los casos de prueban que exploran condiciones limites tienen una rentabilidad mayor. Las condiciones límites son situaciones en los bordes, por arriba y por debajo de las clases de equivalencia para los valores de la entrada y de la salida.
3. *Tablas de Decisión*: Representan relaciones lógicas entre las condiciones de entradas y las acciones (salidas). Los casos de pruebas son derivados sistemáticamente considerando cada combinación posible de condiciones y acciones.

Este tipo de herramientas son buenas para capturar los requerimientos del sistema que contienen condiciones lógicas. Pueden ser utilizadas para registrar reglas de negocio complejas de un sistema. SE analiza la especificación y las condiciones y las acciones del sistema se identifican. Las condiciones y las acciones de la entrada se indican de una manera tal que puedan ser verdaderas o falsas. Cada columna de la tabla corresponde a una regla de negocio que define una combinación única de las condiciones que dan lugar a la ejecución de las acciones asociadas a esa regla. La idea es tener por lo menos una prueba por columna, que implica cubrir todas las combinaciones para accionar condiciones.



La fortaleza de estas tablas es crear combinaciones de condiciones que pueden no ejercitarse de otra manera. Se aplica cuando la acción del software depende de varias decisiones lógicas.

4. *Grafo causa Efecto*: Este tipo de técnica ayuda a seleccionar de una manera sistemática los casos de pruebas. Una causa es una condición de entrada o una clase de equivalencia de las condiciones de la entrada. Un efecto es una condición de salida o una transformación del sistema.

A partir de la especificación se identifican las causas y los efectos. El contenido semántico de la especificación es analizado y transformado en un grafo booleano el cual liga las causas con los efectos. Luego este grafo se convierte en una tabla de decisión donde cada columna en la tabla representa un caso de prueba.

5. *Prueba de Dominios*: Un dominio es un conjunto que incluyen todos los valores posibles para una función. En la prueba de dominio se identifican las variables y las funciones. Las variables pueden ser de entrada o de salida. Para cada una, se toman pocos valores representativos de la clase, se encontrarán la mayoría de los defectos que se encontrarían si se prueban con cada miembro de la clase. El elemento primario de interés es la variable más que la función, la prueba de dominio debe analizar una variable y basado en ese análisis, ejecutar las pruebas que involucren esta variable en cada función como entrada o salida.

B. Técnicas de Caja Blanca

Se llaman técnicas de Caja Blanca o de prueba estructural a aquellas donde los casos de prueba se derivan a partir de la estructura del sistema. Requieren conocer el código del programador a probar.

1. *Basadas en el flujo de control*: Son las técnicas que cubren todas las sentencias o bloques de sentencias en un programa, o las combinan específicas de ellas. Esta técnica ejecuta todas las trayectorias del flujo de control de la entrada a la salida.
2. *Basadas en el flujo de los datos*: La prueba en el flujo de datos usa la información sobre cómo se definen, se utilizan y se destruyen las variables del programa. El criterio más fuerte, requiere que, para cada variable, cada segmento de la trayectoria del flujo del control desde una definición de una variable hasta su uso ser ejecutado
3. *Mutante*: Es una versión levemente modificada del programa a probar, diferenciado de él por un cambio sintáctico pequeño. Cada caso de prueba ejercita el programa original y todos los mutantes generados, si un caso de



prueba identifica la diferencia entre el programa y un mutante, se dice que el mutante fue matado.

Puede ser usado para evaluar un conjunto de pruebas o como criterio de prueba en sí mismo.

C. Técnicas según quien hace la prueba

1. *Prueba de aceptación:* Es el proceso que permite comparar el programa contra sus requerimientos iniciales y las necesidades reales de los usuarios, es realizado por el cliente o el usuario final. Generalmente es realizado seleccionando un subconjunto de los casos de prueba del sistema que demuestra formalmente las funcionalidades claves para su aprobación final. Se realiza cuando el sistema es instalado en el ambiente del usuario.
2. *Pruebas alfa y beta:* Antes del que el software sea repartido, se distribuye a un grupo de usuarios potenciales para el uso interno (alfa) o externo (beta). Estos usuarios son los encargados de reportar problemas con el producto. El uso de las pruebas alfa o beta es a menudo descontrolado y no siempre es citado en el plan de pruebas
3. *Prueba de usuario,* esta prueba es realizada por la persona que utilizara el producto. Puede ser realizado en cualquier momento del desarrollo.
4. *Prueba en pares:* Estas pruebas consiste en que dos tester trabajan juntos para encontrar errores. comparten una computadora e intercambian el control de la misma mientras prueban.

D. Técnicas basadas en la intuición o en la experiencia

1. *Prueba Ad hoc:* En esta técnica las pruebas son derivadas confiando en la habilidad, intuición y experiencia con programas similares. Estas pruebas son útiles para identificar pruebas que no son fácilmente encontradas por técnicas más formales.
2. *Conjetura de errores:* Se trata de enumerar una lista de errores y después escribir los casos de prueba basadas en esa lista. Otra forma es identificar los casos de prueba asociados a asunciones que el programador pudo haber hecho cuando leía la especificación.

Los casos de prueba son diseñados pensando las faltas más probables de un programa dado.



MÉTRICAS DEL SOFTWARE

La calidad de un sistema es tan buena como los requisitos que detallan el problema, el diseño que modela la solución, el código que transfiere a un programa ejecutable y las pruebas que ejercita el software para detectar errores. Para lo cual se emplean mediciones que permiten evaluar a la calidad del análisis y los modelos de diseño, así como el código fuente y los casos de pruebas que se han establecidos al aplicar la ingeniería de software.

Para obtener la evaluación de la calidad se deben utilizar técnicas que permiten evaluar la calidad con objetividad, y no con subjetividad. Un buen administrador de proyectos debe evaluar la calidad objetivamente y no subjetivamente. A medida que el proyecto progresa el administrador siempre debe valorar la calidad. Se recomienda que el primer objetivo en el proyecto sea medir los errores y defectos, las métricas que provienen de estas medidas proporcionan la indicación de la efectividad de las actividades de control y de la garantía de calidad en grupos o en particulares.

Existe una gran diferencia entre producir software y producir software de calidad. Un diseño de alta calidad casi siempre es fácil de mejorar para producir una funcionalidad adicional, se puede adaptar con facilidad a requerimientos diferentes, se aplica a distintos entornos y se podría utilizar en diferentes situaciones.

Existen cuatro razones para medir: Caracterizar, Evaluar, Predecir y Mejorar.

- *Medida:* Es un valor asignado a un atributo de una entidad mediante una medición. Un ejemplo: 35.000 líneas de código.
- *Medición:* Es el acto de determinar una medida. Por ejemplo, que un recurso se encargue de medir las líneas de código de cada módulo del sistema.
- *Métrica:* Medida cuantitativa del grado en que un sistema, componente o proceso posee un atributo dado. Incluye el método de medición.
- *Indicador:* Es una métrica o combinación de métricas que proporcionan una visión profunda de software.

Nuestra investigación se centra en las Métricas del Software, por lo cual vamos a definir las:

Una Métrica de Software o Medidas de Software son una medida efectuada sobre los programas, documentación, su desarrollo y mantenimiento o sobre algún aspecto del sistema en desarrollo o del proceso empleado que permite, previa comparación con



unos valores (medidas) de referencia, obtener conclusiones sobre el aspecto medido para adoptar las decisiones necesarias.

Son medidas cuantitativas que permiten a los ingenieros de software obtener una visión de la eficacia del proceso de software y de los proyectos que se llevan a cabo utilizando el proceso como marco de trabajo

Una métrica no es un objetivo en sí mismo, sino un medio para controlar el desarrollo de un sistema de software. Pretenden mejorar los procesos de desarrollo de software y mejorar todos los aspectos de la gestión de aquellos procesos.

Estas medidas son aplicables a todo el ciclo de vida del desarrollo, desde la iniciación, cuando debemos estimar los costos, al seguimiento y control de la fiabilidad de los productos finales y a la forma en que los productos cambian a través del tiempo debido a la aplicación de mejoras.

También se define a la métrica como una medida estadística (no cuantitativa) que se aplica a todos los aspectos de calidad de software, los cuales deben ser medidos desde diferentes puntos de vistas como el análisis, construcción, funcional documentación, métodos, procesos, usuario, entre otros.

Las medidas del software y los modelos de medida son útiles para estimar y predecir costos y para medir la productividad y la calidad del producto. Un ingeniero del software recopila medidas y desarrolla métricas para obtener indicadores.

Pressman dice que las métricas de software ayudan a la evaluación de los modelos de análisis y de diseño, en donde proporcionarán una indicación de la complejidad de diseños procedimentales y de código fuente, y ayudaran en el diseño de pruebas más efectivas. Por tal motivo propone un proceso de medición, el cual se puede caracterizar por cinco actividades:

1. *Formulación*: La obtención de medidas y métricas del software apropiadas para la representación de software en cuestión.
2. *Colección*: El mecanismo empleado para acumular datos necesarios para obtener las métricas formuladas.
3. *Análisis*: El cálculo de las métricas y la aplicación de herramientas matemáticas.
4. *Interpretación*: La evaluación de los resultados de las métricas en un esfuerzo por conseguir una visión interna de la calidad de la representación.



5. *Realimentación*: Recomendaciones obtenidas de la interpretación de métricas técnicas transmitidas al equipo de software.

Realizando estas actividades se pueden responder tres preguntas fundamentales deseadas de una métrica:

1. ¿Cuánto mide? - la complejidad en la medida
2. ¿Qué tan bien mide? - la calidad en la medida
3. ¿Qué tanto tiempo mide? - la predicción.

Existen razones que muestran porque es importante medir un producto:

- a. Para indicar la calidad del producto.
- b. Para evaluar la productividad de la gente que desarrolla el producto.
- c. Para evaluar los beneficios en términos de productividad y de calidad, derivados del uso de nuevos métodos y herramientas de la ingeniería de software.
- d. Para establecer una línea de base para la estimación.
- e. Para ayudar a justificar el uso de nuevas herramientas o de formación adicional.

Ventajas del uso de métricas

- Determinar la calidad del producto.
- Evaluar la productividad de los desarrolladores.
- Conocimiento cuantitativo de las características del proceso y del producto.
- Se podrán realizar comparaciones con otros proyectos.
- Se podrá mejorar el producto ya q las métricas sirven para detectar defectos.
- Se tendrá un soporte para la estimación y la planificación.
- Evaluar los beneficios (en cuanto a calidad y productividad) derivados del uso de nuevos métodos y herramientas de ingeniería del software.
- Establecer una línea base para la estimación.
- Justificar el uso de nuevas herramientas de formación adicional.
- Comunicación efectiva basada en hechos y datos y no en opiniones.
- Identificación temprana de problemas y riesgos.
- Seguimiento de objetivos.
- Soporte a las decisiones.

Características de las métricas



La calidad de las medidas debería facilitar el desarrollo de modelos que sean capaces de predecir el comportamiento de determinados parámetros que afectan al desarrollo de productos o procesos.

Una medida ideal debería ser:

- Objetiva.
- Sencilla, definible con precisión para que puede ser evaluada.
- Fácilmente obtenible (a costo razonable).
- Valida, la métrica debería medir exactamente lo que se quiere medir y no otra cosa.
- Robusta. Debería de ser relativamente insensible a cambios poco significativos en el proceso o en el producto.
- Exactas.
- Precisas: No se debe perder información en los redondeos, ya q la información se desvirtúa.
- Consistentes: Una medición de un atributo debe dar el mismo valor independientemente de la medición.
- Comparables: Debe estar normalizada.

Uso de las métricas.

Las métricas deben ser implantadas paso a paso en cinco niveles, correspondientes al nivel de madurez del proceso de desarrollo. Este marco de nivel de madurez del proceso de desarrollo fue establecido por la SEI (Software Engineering Institute).

Proceso Inicial, nivel 1: Su objetivo es formar una base de comparación con la forma en que las mejoras se realicen y se incremente la madurez, estos incluyen:

- a. El tamaño del producto.
- b. El esfuerzo del personal (Utilidades para determinar una tasa de productividad)

Proceso Repetible, nivel 2: Las métricas en este segundo nivel incluyen como objetivos de medición:

1. La calidad de esfuerzo necesaria para desarrollar un sistema.
2. La duración del proyecto.
3. El tamaño y la volatilidad de los requerimientos.
4. Esfuerzo del trabajo personal:



- a. Esfuerzo real medido en unidades personas/mes.
 - b. Esfuerzo reportado en unidades persona/mes.
5. Volatilidad de los requerimientos (Cambios de los requerimientos).

Proceso definido, nivel 3: En este nivel de madurez se recomienda evaluar la complejidad de los requerimientos, el diseño, el código y los planes de prueba, y evaluar la calidad de los requerimientos del diseño del código y de las pruebas. En términos de complejidad, es necesario que los puntos q se mencionaran sean medidos en este nivel.

1. Complejidad de los requerimientos, (Número de distintos objetos y acciones llevadas a cabo en los requerimientos).
2. Complejidad del Diseño (Numero de módulos de diseño y complejidad ciclomatica complejidad de mc cabe).
3. Complejidad del código (Nro. de módulos de código, complejidad ciclomatica).
4. Complejidad de las pruebas (Nros. de caminos a probar, si el desarrollo es orientado a objetos debe considerarse el número de interfaces de objetos a probar). Se puede evaluar la minuciosidad de las pruebas. Algunas métricas recomendadas de calidad son:
 - a. Defectos descubiertos.
 - b. Defectos descubiertos por unidad de tamaño (Densidad de defectos).
 - c. Fallas de requerimientos descubiertos.
 - d. Fallas de diseños descubiertas.
 - e. Fallas de códigos descubiertas
 - f. Densidad de fallas por cada producto.

Proceso administrado, nivel 4: En este nivela la retroalimentación determina como son asignados los recursos pues las actividades básicas, por si mismas no combinan. Las métricas recolectadas son utilidad para encontrar y estabilizar el proceso, así la productividad y la calidad coinciden con las expectativas. Se recomienda por lo tanto recolectar información al respecto de los siguientes aspectos:

- Tipo de proceso, se refiere a que tipo de modelo se utiliza para el desarrollo de software.
- Cantidad de rehúso del consumidor, es establecido en relación a cuanto rehúsa un proyecto componentes de otros aspectos.
- Identificación de defectos, se relaciona con cómo y cuándo se descubren los defectos.
- Uso de densidad de defectos para las pruebas.



- Uso de la administración de la configuración, cuestiona acerca de que si la configuración de la administración es un esquema impuesto sobre el proceso de desarrollo.
- Terminación de módulos sobre tiempo, considera en que porcentaje los módulos está siendo debidamente terminados.

Optimización del proceso, nivel 5: Las métricas de las actividades son utilizada para mejorar el proceso. Son usadas para evaluar y controlar el proceso de desarrollo del software, de forma que permitan:

- Indicar la calidad del producto.
- Evaluar la productividad de los desarrolladores.
- Evaluar los beneficios (En cuanto a calidad y productividad).
- Derivados del uso de nuevos métodos y herramientas de ingeniería del software.
- Establecer una línea de base para la estimación.
- Justificar el uso de nuevas herramientas o de formación adicional.

Es necesario utilizar las métricas más adecuadas para conseguir el control, seguimiento y mejora de la calidad, para lo cual es necesario determinar los factores de calidad más importantes dentro del proyecto.

Tipos de Métricas:

Medida de amplitud o cobertura de las pruebas

Proporciona un indicador de cuantos requisitos se han probado del número total de ellos. Indica la complejidad del plan de pruebas.

La cobertura de las pruebas indica cómo se van cumpliendo los casos de prueba especificados, por tanto, una mayor cobertura de las pruebas indica un buen desarrollo de las pruebas.

La cobertura de las pruebas se calcula como:

- CPE: número de casos de prueba que han sido ejecutados en total.
- CPR: número de casos de prueba a ejecutar requeridos para cubrir todos los requerimientos.
- CPRT: número de casos de prueba que se han ejecutado por cada requerimiento.



Madurez de las pruebas

Indicador del buen desempeño del flujo de trabajo de pruebas, no sólo se enfoca en la completitud de los casos de prueba según los definidos para cubrir los requerimientos, sino que también comprende los casos de pruebas que han obtenido resultados satisfactorios.

Donde:

- CPS: número de casos de prueba que han dado resultados satisfactorios sobre el total de pruebas definidas.
- CPSR: número de casos de prueba que han dado resultados satisfactorios sobre el total de pruebas definidas para cada requerimiento.
- CPF: número de casos de prueba que han dado resultados fallidos sobre el total de pruebas definidas.
- CPFRR: número de casos de prueba que han dado resultados fallidos sobre el total de pruebas definidas para cada requerimiento.

Perfiles de fallos

Se emplean para categorizar y priorizar los errores. La prioridad indica la severidad del problema. Las categorías de fallos proporcionan una descripción del error para realizar análisis estadísticos de errores.

El porcentaje de defectos por tipo permite categorizar los fallos porque identifica los tipos de defectos más comunes que puedan presentarse en cualquier etapa del desarrollo del software.

El porcentaje de defectos por tipo se calcula como:

Donde:

- %DT: porcentaje de defectos por tipo de severidad.
- NDT: es el número de defectos por tipo de severidad.
- TDI: número total de defectos identificados en el proyecto.

Densidad de defectos

Esta métrica permite realizar análisis estadísticos al finalizar las pruebas para valorar la integridad y madurez del software analizado. La densidad de defectos ofrece una medida sobre la proporción de defectos con respecto a la cantidad de elementos de especificación.



La densidad de defectos se calcula como:

- TD: número total de defectos encontrados durante las pruebas.
- TDE: número de defectos encontrados sobre casos de prueba ejecutados.

El Testing nos puede ayudar a medir la Calidad del Software. La calidad es el grado en el que un conjunto de características inherentes de un producto cumple con sus requisitos previamente definidos. Esta medición se realiza en función de los Defectos encontrados y la cobertura de los mismos de acuerdo a:

- Requerimientos y características funcionales del Software (corrección y funcionalidad)
- Requerimientos y características no funcionales del Software (confiabilidad, eficiencia, integridad, facilidad de uso, facilidad de mantenimiento, flexibilidad, facilidad de prueba, portabilidad, facilidad de reutilización, interoperabilidad)

La mayoría de las métricas se concentran en el proceso y no en el producto. Debe apoyarse en las métricas del análisis y del diseño.

Las métricas basadas en la función pueden utilizarse para predecir el esfuerzo global de las pruebas.

A medida que avanzan las pruebas, hay métricas que indican la completitud de las mismas:

- Amplitud de las pruebas (cuantos requisitos se han probado).
- Perfiles de fallo (para dar prioridad y categorizar los errores encontrados).

Las métricas de calidad de software dan soporte a:

- El diseño de casos de prueba efectivos
- Evaluación de la eficacia y efectividad de las pruebas
- Medición de defectos

En muchos casos las métricas de un modelo pueden aplicarse en actividades posteriores a la ingeniería del software.

En cuanto a la medición de defectos que surgen de las pruebas, es importante clasificarlos mínimamente por Severidad y por Prioridad, este último aspecto determinara la urgencia en el tratamiento del mismo.

Son características de los defectos que van de la mano, si un defecto es altamente severo su prioridad de tratamiento es urgente.



En la aplicación desarrollada se definieron como Severidad y Prioridad las siguientes características:

En cuanto a la Severidad, la cual definimos como el grado de impacto que un defecto tiene en el desarrollo u operación de un componente del sistema, se tiene:

- Crítica: el defecto es bloqueante, produce una interrupción del servicio.
- Mayor: Un defecto en la funcionalidad principal para el cual no existe un camino alternativo o el mismo no es aceptable.
- Moderado/Medio: Un defecto en la funcionalidad principal para el cual existe un camino alternativo simple o un defecto en la funcionalidad no-principal para el cual no existe un camino alternativo o el mismo no es aceptable.
- Menor/Baja: Un defecto en la funcionalidad no-principal para el cual existe un camino alternativo simple.
- Cosmético: Un defecto que no afecta a la funcionalidad y no requiere un camino alternativo. Por ejemplo: errores ortográficos, fuente incorrecta...

En cuanto a la Prioridad, la cual describe como es de importante la corrección de un defecto, se tiene:

- Inmediata: El defecto debe resolverse inmediatamente.
- Alta: El defecto debe resolverse tan pronto como sea posible dentro del curso normal del desarrollo. Puede esperar hasta que se haga una nueva versión.
- Media: El defecto debe resolverse, pero puede esperar hasta que se hayan corregido defectos más serios.
- Baja: El defecto podría o no resolverse.



Módulo 3: Modelado de Negocio

En el tercer Módulo, MODELADO DE NEGOCIO, se hace un enfoque en la estructura y dinámicas del negocio su lógica y los problemas reales que presenta el negocio, para realizar mejoras sobre los mismos. Por tal motivo en este capítulo se definen los actores del negocio y la definición de las transacciones q ellos realizan.



Introducción

En esta etapa del proyecto, se analizarán en profundidades los procesos a través de los cuales la empresa gestiona las mejoras, soluciones y los nuevos proyectos con clientes existentes y/o nuevos. El objetivo de este modelado es entender la estructura y dinámicas del negocio, su lógica y comprender cuales son los problemas reales que se presentan para poder realizar las mejoras sobre los mismos y derivar los requerimientos del sistema.

Durante el modelado del negocio se examinarán cuáles son las entidades internas y externas que interactúan en la empresa, el rol que juegan y en qué proceso están involucradas. Estos procesos son los que constituyen los casos de uso del sistema del negocio.

Las relaciones y dependencias entre los actores y casos de usos se representarán a través de diagramas de casos de uso. Posteriormente se definirá una especificación en trazo grueso de los casos de uso donde se describirá el flujo de eventos y datos que se intercambian entre actor y caso de uso.

Como último paso, se construirá el modelo de objetos del negocio; para lo cual se realizará el diagrama de clases identificando las entidades del negocio y la relación entre ellas.

En conclusión, el objetivo de este modelado de negocio, es describir los procesos existentes u observados con el propósito de comprenderlos y hacer que todas las personas involucradas en el proyecto tengan un entendimiento común de la empresa.

Definición de Actores

En esta sección se describirán los actores del negocio conjuntamente con la descripción del rol que desempeñan de acuerdo a los procesos de negocios en los que intervienen.

Nombre del Actor	Descripción del Rol
CLIENTE	Es el que se contacta con la empresa para consultar sobre algún proyecto nuevo en base a lo que necesita, o solicitar alguna mejora, o solucionar algún defecto.
PROJECT MANAGER	Es la persona encargada de hablar con el cliente, negociar con él, y recolectar nuevos clientes. Se encarga de definir el proyecto, presentar las etapas que lo componen y el plazo a cada cliente. Tiene en cuenta la planificación, define fechas, plazos, responsables, cargos, recursos y calcula el costo del mismo.
	Es el encargado de relevar y gestionar las necesidades funcionales del cliente, realiza el análisis del caso. Es el



ANALISTA FUNCIONAL	<p>que cumple con los estándares y la documentación requerida por las normas de calidad de la empresa. Especifica los requerimientos y las funcionalidades de una posible solución.</p> <p>Sabe detectar lo que realmente el cliente necesita y quiere, esto lo hace a través de las entrevistas que tiene con los usuarios. También guía al líder de testing y a los analistas en la elaboración de los casos de pruebas.</p> <p>Valida y obtiene la aprobación de las definiciones del usuario.</p>
LIDER DE DESARROLO	<p>Es el responsable de guiar al equipo de desarrolladores cuando se presenta un nuevo proyecto o una mejora, evaluando. Define como se van a integrar las partes implementadas y es el que asigna los recursos de desarrollo que van a estar involucrados en el proyecto.</p>
DESARROLLADOR	<p>Es la persona que, a través de una serie de requisitos, diseña y da forma a la solución técnica de lo que se tiene q implementar. Diseña los datos y la estructura del código, los algoritmos funcionales y el flujo de la aplicación que engloba la lógica del negocio y satisface los casos de uso del software.</p> <p>Define y ejecuta pruebas unitarias, es decir pruebas de pequeñas unidades de código que permite asegurar que funcionen bien, permitiendo depurar bugs desde las fases iniciales del desarrollo del software y permiten que sean más fáciles las pruebas de regresión posteriores cuando haya q editar o actualizar el software.</p>
LIDER DE TESTING	<p>Es el responsable de coordinar y dirigir las actividades de testing. Es el que revisa y analiza los requisitos del proyecto. Define el alcance de las pruebas que se definirán, así como también planifica, organiza y dirige las reuniones de lanzamientos de las pruebas.</p> <p>Crea el cronograma de pruebas (tareas, dependencias y asignación de recursos).</p> <p>Define el plan de pruebas de software, para que sea revisado, aprobado y firmado por las partes interesadas.</p> <p>Asigna los recursos necesarios para el proyecto. Estima los esfuerzos de las prueba y del equipo involucrado (en cuanto a su tamaño, habilidades, actitudes y calendario).</p>
ANALISTA DE TESTING	<p>Es el recurso encargado de definir y/o ejecutar los casos de pruebas en base a la ERS que recibe dependiendo del proyecto al cual esté involucrado. Trata de detectar la mayor cantidad de fallas severas (incidentes de alto impacto) con el mínimo esfuerzo, antes de que el software salga a producción. Confecciona los documentos de pruebas y reporta, si es que existen, los defectos encontrados.</p>
ARQUITECTO	<p>Es el personal encargado de diseñar la arquitectura del sistema. Planifica los elementos individuales del sistema, los implementa y también tienen la responsabilidad de mejorarlos.</p>



Definición de Transacciones del Negocio

Nro.	NOMBRE	OBJETIVO
001	Atender requerimiento del cliente	Establecer los procedimientos necesarios para atender las solicitudes de los clientes referidas a una nueva característica, una mejora o un fallo del sistema.
002	Reunir a los involucrados	Realizar reuniones con las personas involucradas en el proyecto, como los clientes, testers, desarrolladores, el PM, el arquitecto, donde se habla sobre las solicitudes expuestas por el cliente, como así también, los avances del equipo involucrado en el desarrollo del sistema.
003	Definir requerimiento	Definirlas características que debe poseer un software, para cumplir con las necesidades y deseos que fueron pedidos por el cliente y las personas que están involucradas en el sistema.
004	Planificar proyecto	Definición de los objetivos, el alcance del y las personas que van a estar involucradas al proyecto y la división de tareas entre ellos, como así también, la confección de cronogramas con las fechas de inicio y fin del proyecto.
005	Confeccionar cronogramas de fechas	Ordenar por fechas las acciones o actividades que se van a desarrollar en el proyecto.
006	Realizar minutas diarias	Realizar reuniones diarias con las personas
007	Confeccionar documentación de minuta	Realizar reuniones diarias con los recursos involucrados en el proyecto, donde se establecen las tareas que se han realizado y las próximas a realizarse, si surgió algún defecto o problema, etc.
008	Definir casos de pruebas	Definir un conjunto de condiciones o variables bajo las cuales se valida que el sistema funcione correctamente, se trata de encontrar los defectos.
009	Crear ciclo de pruebas	Confeccionar un conjunto de pruebas para ser probadas.
010	Documentar casos de pruebas	Dejar asentado en forma escrita, las evidencias de cada prueba realizada.
011	Confeccionar planilla de casos de pruebas	Confeccionar una planilla de Excel con los casos de pruebas del proyecto solicitado, indicando id del caso de prueba, nombre, objetivo, descripción, precondiciones, pasos y resultado esperado.
012	Confeccionar planilla de ciclos de pruebas	Confeccionar una planilla de Excel con los ciclos de pruebas creados para el proyecto, indicando id de ciclo, nombre, proyecto, fecha de inicio y fecha de finalización.
013	Ejecutar ciclos de pruebas	Ejecutar cada una de las pruebas que forman el ciclo de pruebas indicando si la ejecución fue exitosa o fallida.
014	Confeccionar plan de pruebas	Especificar las funciones a probar, como serán ejecutadas esas pruebas, quienes serán los responsables y el cronograma para su ejecución.
015	Confeccionar planillas de defectos	Confeccionar una planilla Excel con los defectos encontrados en las corridas de los ciclos de pruebas, indicando id, nombre, descripción, Caso de prueba defectuoso, a quien se asignara, y el estado del defecto.
016	Documentar defectos	Dejar asentado en forma escrita, como se detectó el defecto, adjuntando imágenes, y toda evidencia necesaria para que el desarrollador pueda solucionarlo.



017	Solucionar incidencias	Desarrollador soluciona el defecto que se le asigno y comunica que ya está solucionado en forma verbal o vía mail.
018	Realizar el pase a usuarios	Exponer al usuario el sistema con las nuevas características o mejoras funcionando en forma correcta.
019	Solicitar aprobación de usuario	Solicitar la aprobación del sistema por parte del usuario.
020	Implementar desarrollo	Enviar las evidencias de las pruebas que se realizaron en el sistema y solicitar confirmación del buen funcionamiento del mismo vía mail.
021	Realizar el pase a producción	Enviar los planes de despliegue del paso a paso a las personas responsables de servidores del cliente. Esperar la respuesta de dichas personas, donde se comunica que se implementó lo solicitado.
022	Confeccionar planillas con resultados	Confeccionar una planilla de Excel con los resultados de las pruebas que se ejecutaron, los resultados de las incidencias registradas y el resultado del pase a usuario y a producción.
0243	Generar reportes manuales de métricas	Generar reportes de métricas manuales, utilizando los datos que se obtuvieron de la ejecución de los ciclos de pruebas y del registro de incidencias, para verificar si se están alcanzando los objetivos que se propusieron y también para orientar las acciones futuras para mejorar la estrategia y los resultados.



Módulo 4: Requerimientos

En el Módulo cuatro, REQUERIMIENTOS, se realiza una visión de los requerimientos que van a ser parte de nuestro sistema, como así también los actores del sistema de información. Se muestran los distintos prototipos de interfaz gráfica que tendrá la aplicación, especificando por modelo de usuarios, proyecto, requerimiento, casos de pruebas, ciclos de pruebas, ejecuciones/Determinaciones, incidencias y reportes.



Introducción

Partiendo del modelado de negocio, en el flujo de trabajo de requerimientos pretendemos captar los servicios o funcionalidades que proveerá el sistema.

- **Requerimientos Funcionales:** Son declaraciones de los servicios que proveerá el sistema, de la manera en que este reaccionará a entradas particulares y de cómo se comportará en situaciones particulares. En algunos casos estos requerimientos también declaran explícitamente lo que el sistema no debe hacer.
- **Requerimientos No Funcionales:** son restricciones de los servicios o funciones ofrecidas por el sistema. Incluyen restricciones de tiempo sobre el proceso de desarrollo, estándares, etc. No se refieren directamente a las funciones específicas que entrega el sistema sino a las propiedades emergentes de este.

Tomando como base el Proceso Unificado y UML, construiremos un modelo utilizando casos de uso e identificaremos los actores del sistema de información o trabajadores del negocio que colaboran con los casos de uso.

Los casos de uso se especificarán en trazo fino describiendo su flujo de sucesos en detalle, incluyendo cómo comienza, termina e interactúan con los actores. Los casos de uso proporcionan un medio intuitivo y sistemático para capturar los requisitos funcionales con un énfasis especial en el valor añadido para cada usuario individual o para cada sistema externo y serán la entrada fundamental para el análisis, el diseño y las pruebas.

Finalmente, se desarrollarán prototipos de interfaz gráfica que permitan a los usuarios del sistema llevar a cabo los casos de uso de manera eficiente.



Requerimientos del Sistema

Nro.	Requerimiento	Objetivo
1	Registrar Usuario	Registrar los datos de un nuevo usuario con su rol correspondiente en la BD.
2	Eliminar Usuario	Eliminar un usuario registrado la base de datos.
3	Modificar Usuario	Actualizar los datos de un usuario registrado en la BD.
4	Modificar Contraseña de Usuario	Registrar la actualización de la clave personal del usuario en la base de datos del sistema.
5	Iniciar Sesión	Validar las credenciales del usuario y registrar su inicio de sesión en el sistema.
6	Cerrar Sesión	Finalizar la sesión del usuario en el sistema.
7	Registrar Proyecto	Registrar los datos de un nuevo proyecto en la base de datos.
8	Eliminar Proyecto	Eliminar un proyecto y sus datos asociados de la base de datos.
9	Modificar Proyecto	Actualizar los datos de un proyecto registrado en la BD.
10	Registrar Requerimiento	Registrar los datos de un nuevo requerimiento en la base de datos.
11	Eliminar Requerimiento	Eliminar un requerimiento y sus datos asociados de la base de datos.
12	Modificar Requerimiento	Actualizar los datos de un requerimiento registrado en la BD.
13	Registrar Caso de Prueba	Registrar los datos de un nuevo caso de prueba en la base de datos.
14	Eliminar Caso de Prueba	Eliminar un caso de prueba y sus datos asociados de la base de datos.
15	Modificar Caso de Prueba	Actualizar los datos de un caso de prueba registrado en la BD.
16	Copiar Caso de Prueba	Duplicar los datos de un caso de prueba ya registrado en la base de datos, bajo un nuevo nombre.
17	Registrar Ciclo de Prueba	Registrar los datos de un nuevo ciclo de prueba en la base de datos.
18	Eliminar Ciclo de Prueba	Eliminar un ciclo de prueba y sus datos asociados de la base de datos.
19	Modificar Ciclo de Prueba	Actualizar los datos de un ciclo de prueba registrado en la BD.
20	Copiar Ciclo de Prueba	Duplicar los datos de un ciclo de prueba ya registrado en la base de datos, bajo un nuevo nombre.
21	Asignar Pruebas a Ciclo	Asociar casos de pruebas registrados a un ciclo de prueba creado en la base de datos.
22	Registrar determinación de prueba	Registrar el estado de la ejecución de una prueba dentro del ciclo de prueba.
23	Registrar Incidencia	Registrar los datos de una nueva incidencia en la base de datos.
24	Eliminar Incidencia	Eliminar una incidencia y sus datos asociados de la base de datos.
25	Modificar Incidencia	Actualizar los datos de una incidencia registrada en la BD.
26	Generar Reporte Métricas Generales	Generar reporte de métricas generales por proyecto.
27	Generar Reporte de Incidencias	Generar reporte de incidencias registradas para un proyecto.
28	Generar Reporte de Amplitud o Cobertura de pruebas	Generar reporte de amplitud y cobertura de pruebas definidas y realizadas por proyecto.
29	Generar Reporte de Madurez de Pruebas	Generar reporte de la madurez alcanzada por las pruebas realizadas en un proyecto.
30	Generar Reporte Perfiles de Fallos	Generar reporte de perfiles de las fallas registradas para un proyecto.
31	Generar Reporte de Densidad de Defectos	Generar reporte grafico de densidad de defectos para un proyecto específico.



32	Generar reporte de módulos	Generar listado de los elementos registrados en el módulo seleccionado.
33	Registrar Grupo de Usuario	Registrar los datos de un nuevo grupo de usuarios en la base de datos.
34	Eliminar Grupo de Usuarios	Eliminar un grupo de usuarios de la base de datos.
35	Modificar Grupo de Usuarios	Actualizar los datos de un grupo de usuarios registrados en la base de datos.

Actores del Sistema de Información

Nombre del Actor	Descripción del Rol
ADMINISTRADOR	Es el usuario encargado de la gestión de usuarios y proyectos del sistema.
ANALISTA FUNCIONAL	Es el usuario encargado de la gestión de requerimientos dentro del sistema.
DESARROLLADOR	Es el usuario encargado de la gestión de incidencias dentro del sistema.
TESTER	Es el usuario encargado de la gestión de casos de prueba, ciclos de prueba, determinaciones e incidencias dentro del sistema.

Prototipos de interfaz gráfica

MODULO USUARIOS

Registrar y Editar Usuarios

Prototipo de interfaz gráfica para el formulario "Nuevo Usuario". El formulario tiene un título "Nuevo Usuario" y tres pestañas: "Usuario", "Grupo de Usuario" (seleccionada) y "Proyectos". Contiene los siguientes campos de texto:

- Apellido:
- Nombre:
- Email:
- Password:

En la parte inferior del formulario hay tres botones: "Guardar y Salir", "Guardar" y "Cancelar".



Nuevo Usuario

Usuario Grupo de Usuario Proyectos

Search

Administradores
 Desarrolladores
 Analistas Funcionales
 Testers

Guardar y Salir Guardar Cancelar

Nuevo Usuario

Usuario Grupo de Usuario Proyectos

Search

TESIS
 PREIN
 DVAL
 OPJ

Guardar y Salir Guardar Cancelar

Modificar Contraseña de Usuario

Cambiar Contraseña

Contraseña Actual:
text

Contraseña Nueva:
text

Repetir Contraseña:
text

Guardar Cancelar



Iniciar Sesión

QUALITY TRACKING

Email:

Contraseña:

[Olvido su contraseña?](#)

MODULO PROYECTOS

Proyectos

Estado: Creado Desde: Creado Hasta:

Nombre	Descripción	Fecha Inicio	Fecha Fin	Estado	Acciones
DVAL	Deposito de Valores	03/03/2014	23/12/2016	Cerrado	
PREIN	Precalificacion e Ingresos	07/03/2016	15/10/2017	Ejecucion	

Registrar Proyecto



Nuevo Proyecto

Id:
Estado:

Nombre del Proyecto:

Descripcion del Proyecto:

Fecha Inicio: **Fecha Fin:**

Lider Proyecto:

Observaciones:

Modificar Proyecto

Editar Proyecto

Id: **Estado:**

Nombre del Proyecto:

Descripcion del Proyecto:

Fecha Inicio: **Fecha Fin:**

Lider Proyecto:

Observaciones:



MODULO REQUERIMIENTOS

Requerimientos

Clave: Nombre: Tipo: Proyecto:

Clave	Nombre	Tipo Requerimiento	Proyecto	Documento	Acciones
RFPIN1	Visualizar Ingreso Cliente	Funcional	PREIN		
RFDV02	Colores de Pantallas	No Funcional	DVAL		

Registrar Requerimiento

Nuevo Requerimiento

Proyecto: Clave:

Tipo de Requerimiento: Nombre del Requerimiento:

Objetivo del Requerimiento:

Documento:

MODULO CASOS DE PRUEBA

Casos de Prueba

Clave: Nombre: Proyecto: Requerimiento:

Clave	Requerimiento	Caso de Prueba	Proyecto	Acciones
CP_PREIN_1	Visualizar Ingreso Cliente	Ingreso Vacio	PREIN	
CP_TESIS_2	Registrar Usuario	Registro Exitoso Usuario Grupo Administrador	TESIS	



Registrar Caso de Prueba

Nuevo Caso de Prueba

Proyecto: ▼ Requerimiento: ▼

Clave: Nombre del Requerimiento:

Categoría: ▼

Objetivo:

Pre-Condiciones:

Pasos de Ejecucion:

Resultado Esperado:

MODULO CICLOS DE PRUEBA

Ciclos de Prueba

Clave: Proyecto: ▼ Desde: Hasta:

Proyecto	Nombre	Ambiente	Fecha Inicio	Fecha Fin	Acciones
PREIN	PRE_Ciclo1	Desarrollo	15/03/2017	17/03/2017	
TESIS	TES_Ciclo1	Testing	15/03/2017	17/03/2017	



Registrar Ciclo de Prueba

Nuevo Ciclo de Prueba [Minimizar] [Maximizar] [Cerrar]

Proyecto:

Nombre:

Ambiente:

Fecha Inicio: Fecha Fin:

Observaciones:

Modificar Ciclo de Prueba

Editar Ciclo de Prueba [Minimizar] [Maximizar] [Cerrar]

Proyecto:

Nombre:

Ambiente:

Fecha Inicio: Fecha Fin:

Observaciones:



MODULO INCIDENCIAS

Incidencias

Proyecto: Estado: Tipo:

Severidad: Prioridad: Asignado a:

Clave	Nombre	Tipo	Severidad	Prioridad	Estado	Asignado	Proyecto	Acciones
IDVAL001	Error en Alta de Cheque	Bug	Critico	Alta	En Tratamiento	Arriola Eliel	DVAL	
IOPJ001	Falla proceso contable	Bug	Medio	Alta	Solucionado	Carle Claudio	OPJ	

Registrar Incidencia

Nueva Incidencia

Nombre/Resumen: Proyecto:

Requerimiento: Caso de Prueba:

Clave: Estado:

Tipo: Severidad:

Prioridad: Asignado a:

Evidencia:

Observaciones:

MÓDULO MÉTRICAS

Amplitud o Cobertura



Métricas - Amplitud o Cobertura de Pruebas

Reporte: Amplitud o Cobertura de Pruebas

Información de Proyecto

Proyecto: DVAL - Declaración de Valores
Sprint: N/A
Ciclo: 1
Entorno: Desarrollo

Recursos asignados

Lider de Proyecto: Marcelo Perez
Desarrollador/es: Juan Sosa / Claudio Ñañez
Analistas Testing: Noelia Peralta / Federico Valdez

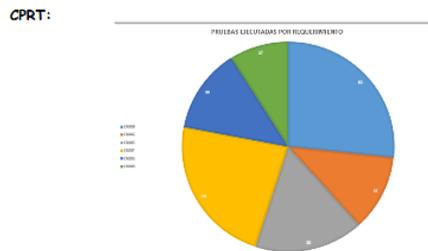
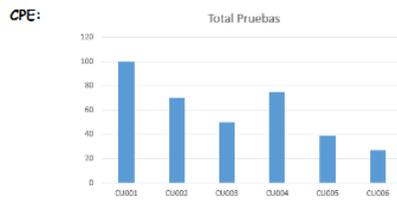
Estadísticas

Requerimiento	Total Pruebas	Ejecutadas	Exitosas	% Exitosas	Con Fallos	% ConFallos	No Ejecutadas	% No Ejecutadas
CU001	100	80	80	100%	0	0%	20	20%
CU002	70	35	30	85%	5	15%	0	0%

Métricas Resumen

CPE: 80 Casos de Prueba Ejecutados
CPR: 80 Casos de Prueba Requeridos

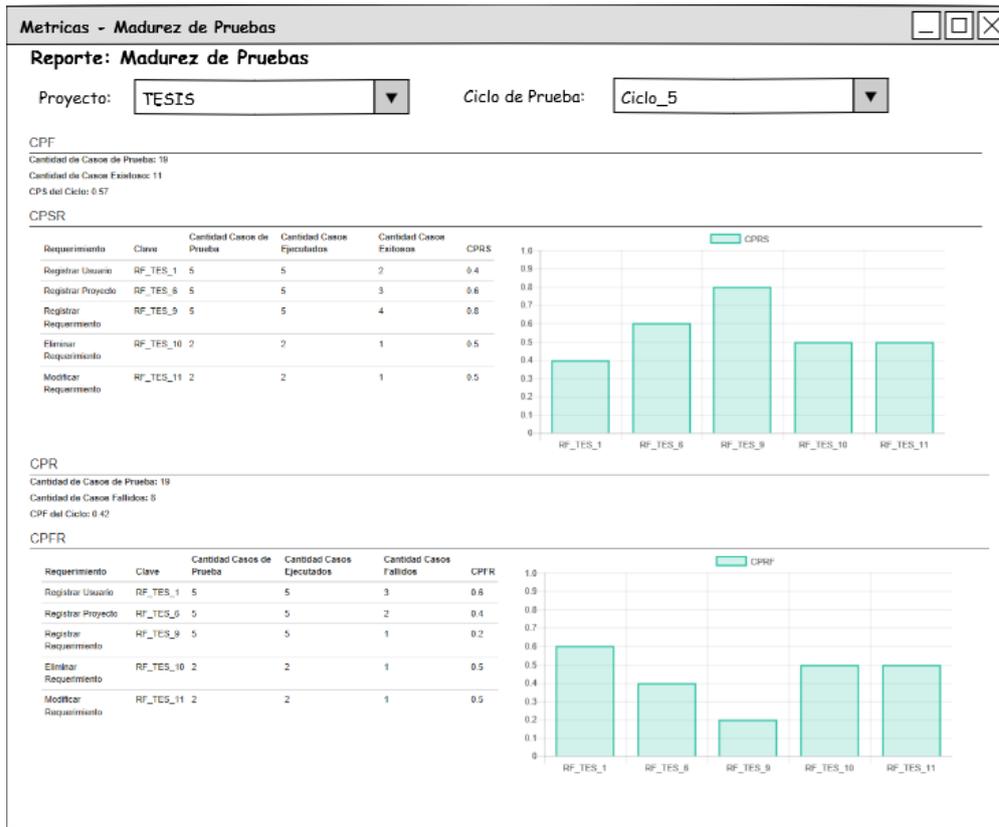
Gráficos



Imprimir Guardar Salir

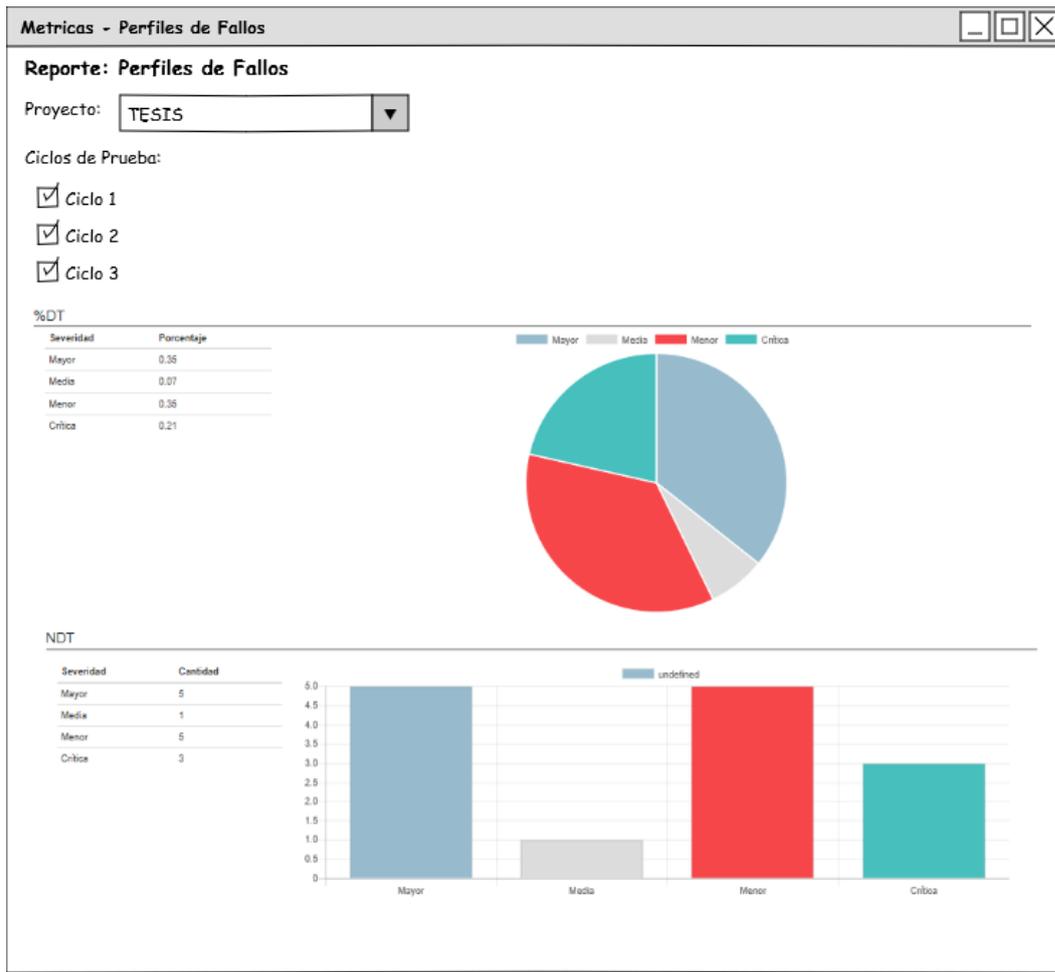


Madurez de Pruebas

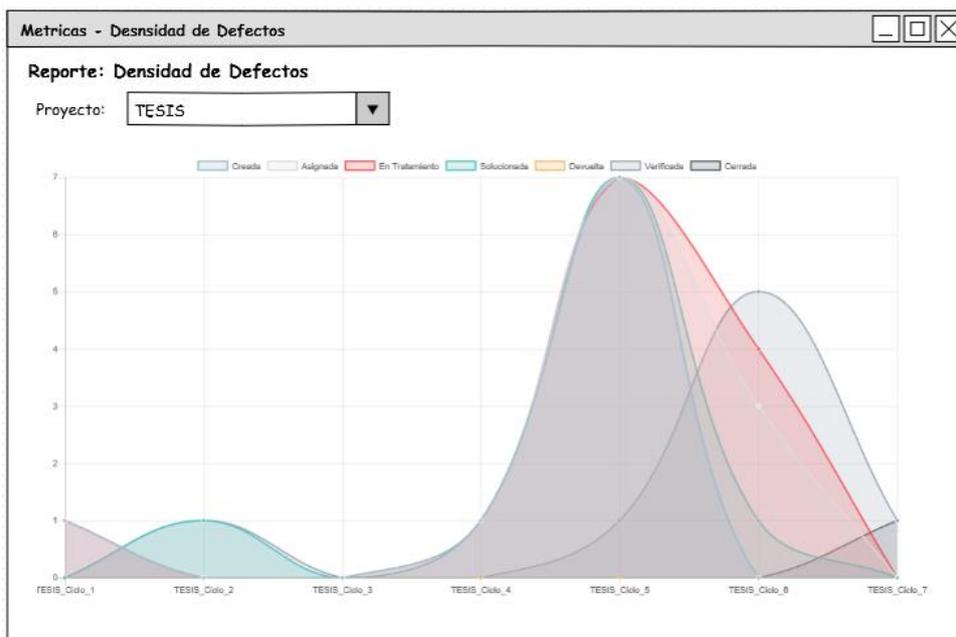




Perfiles de Fallos



Densidad de Defectos





Módulo 5: Análisis y Diseño del Sistema

En el Módulo cinco, ANALISIS Y DISEÑO DEL SISTEMA, Se definen los Casos de usos del sistema de información, el diagrama de estado y el modelado de datos, a través del diagrama de Entidad-Relación, describiendo en forma gráfica las entidades y las relaciones que existen entre ellas.



Introducción

Durante el flujo de trabajo de análisis, se analizan los requisitos descritos en el flujo de trabajo de requerimientos, refinándolos y estructurándolos. El objetivo de hacerlo es conseguir una comprensión más precisa de los requisitos y una descripción de los mismos que sea fácil de mantener y que ayude a estructurar el sistema entero, incluyendo su arquitectura.

El modelo de análisis no sólo es útil para el mantenimiento de los requisitos como tales, sino que también se utiliza como entrada en las actividades de diseño y de implementación. A través del análisis se consigue una separación de intereses que prepara y simplifica las subsiguientes actividades, delimitando los temas que deben resolverse y las decisiones que deben tomarse en esas actividades.

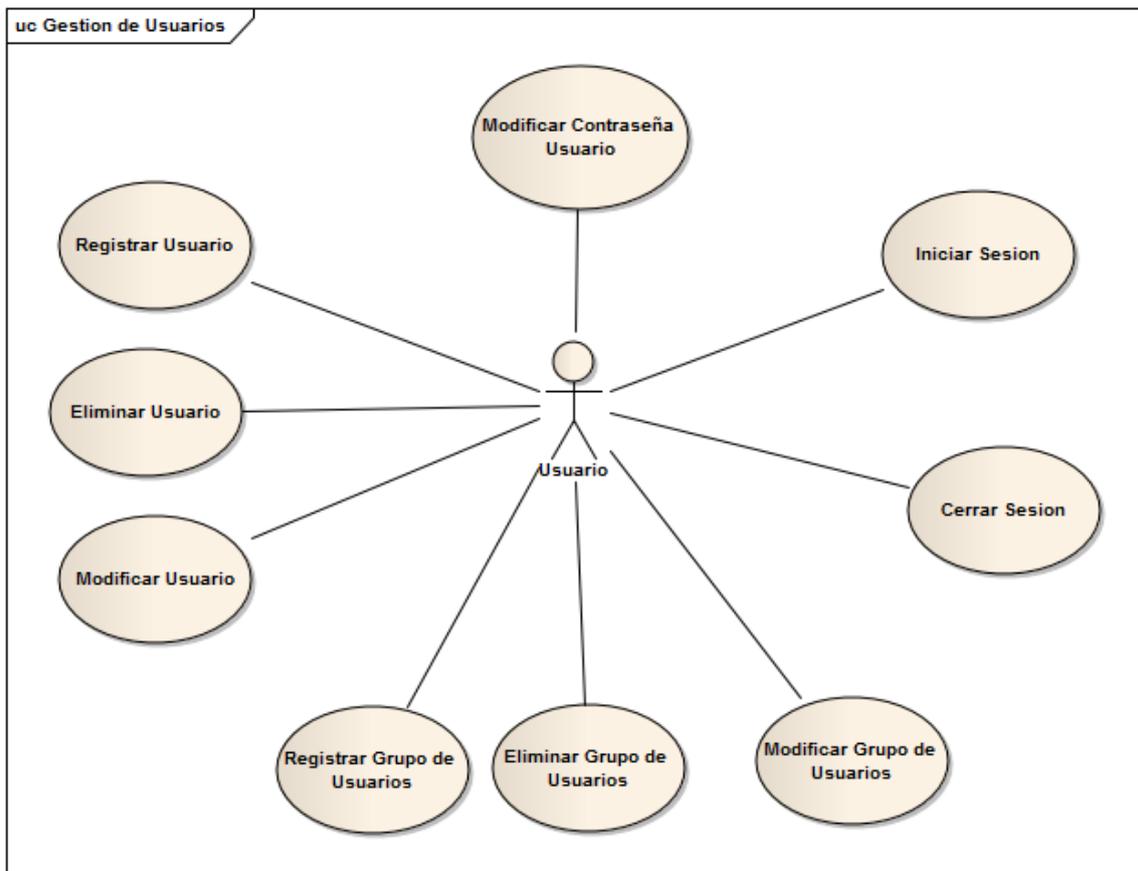
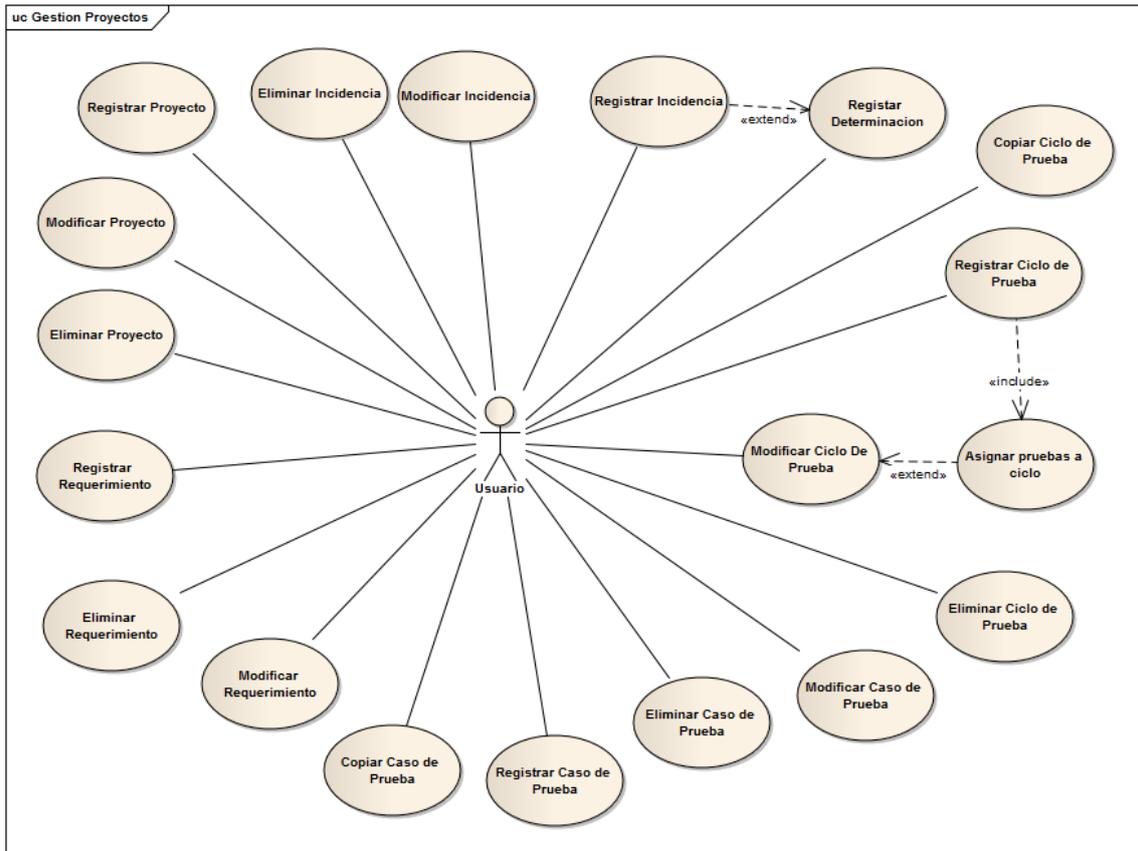
El objetivo de la etapa de diseño es modelar el sistema y encontrar su forma, incluida su arquitectura, para que soporte tanto los requisitos funcionales como los no funcionales junto con otras restricciones relacionadas con el entorno de implementación.

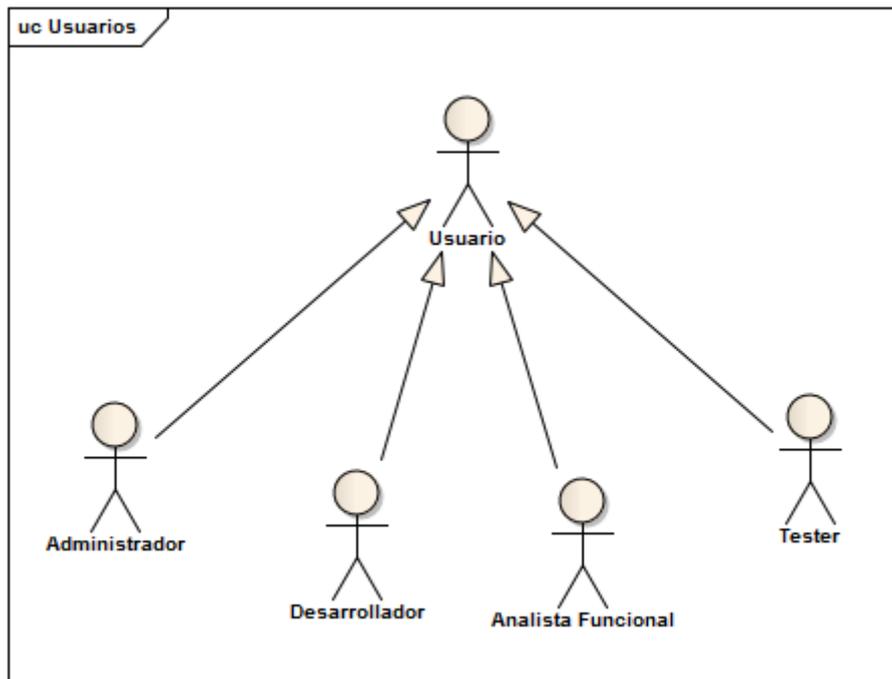
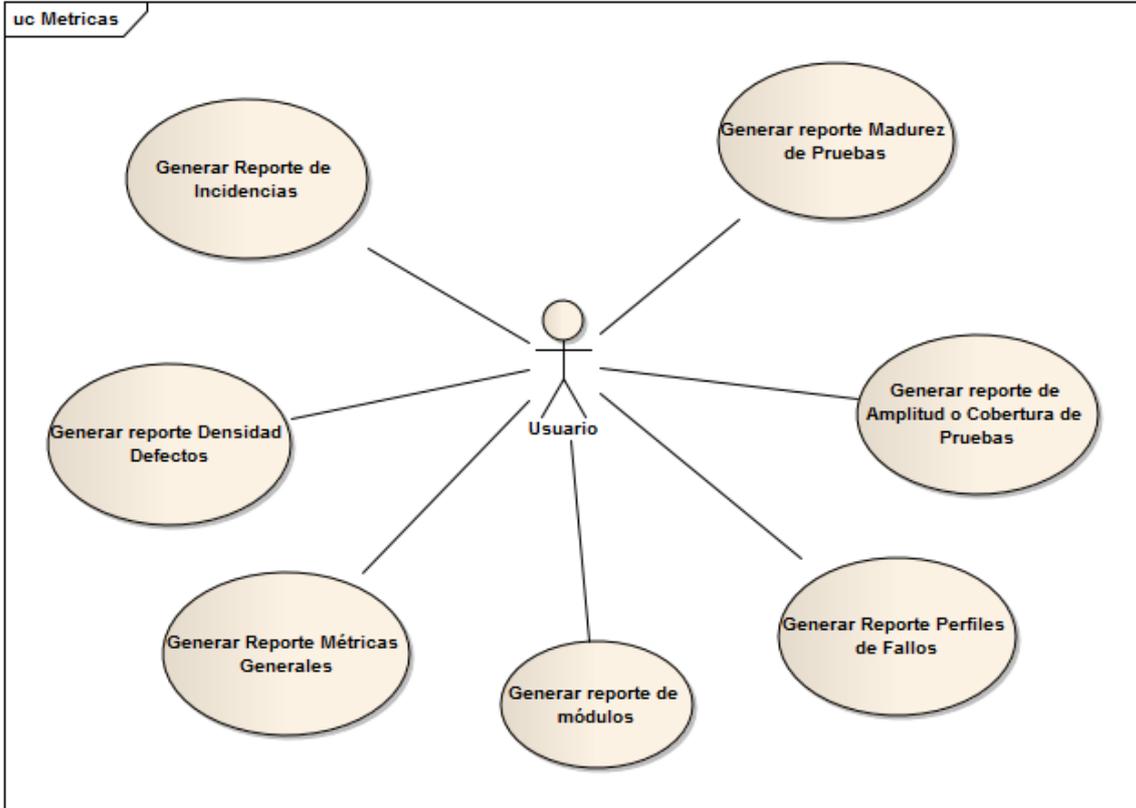
Durante el diseño se buscará preservar la estructura definida en el análisis buscando una arquitectura estable y sólida y creando un plano del modelo de implementación. Para cumplir dichos objetivos se definirá la tecnología de información que soportará el sistema y se confeccionará un diagrama de despliegue que describa la distribución física del sistema en términos de cómo se distribuye la funcionalidad entre los nodos de cómputo.

Se construirá el diagrama de casos de uso, diagrama de estados y modelo de datos a través de un diagrama de Entidad Relación que describa en forma gráfica las entidades y las relaciones entre ellas.



Diagrama de Casos de Uso del Sistema de Información







Descripción de Casos de Uso

CASO DE USO: 001. REGISTRAR USUARIO.

Descripción

Nombre del Caso de Uso: Registrar usuario		Número: 001
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Registrar los datos de un nuevo usuario con su rol correspondiente en la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación, con permisos para realizar la acción.		
Post-Condicion:	Éxito 1: Usuario registrado exitosamente.	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el Usuario selecciona dentro del menú principal, la opción Usuarios y luego "Agregar usuario".		
2. El sistema muestra un formulario llamado "Nuevo usuario".		
3. El usuario completa/selecciona los siguientes campos solicitados obligatorios y presiona "Guardar y Salir": <ul style="list-style-type: none"> • Apellido. • Nombre. • Email. • Contraseña. 		3.A. El usuario no completa todos los campos requeridos. Presiona la opción "Guardar y Salir". 3.A.1. El sistema informa la situación. 3.A.2. Continúa en el paso 3 del curso normal. 3.B. El usuario completa todos los datos de la pestaña Usuario y selecciona la pestaña "Grupos de Usuario" 3.B.1. El sistema muestra los grupos de usuarios existentes y la opción de buscar el grupo deseado. 3.B.2. El usuario selecciona el/los grupos a los que quiera que participe el usuario y presiona "Guardar y Salir". 3.C. El usuario completa todos los datos de las pestañas Usuario y Grupos de Usuario y selecciona la pestaña "Proyectos". 3.C.1. El sistema muestra todos los proyectos registrados. 3.C.2. El usuario selecciona el/los proyectos que quiere asociar al nuevo usuario y presiona "Guardar y Salir".
4. El sistema registra el nuevo usuario en la base de datos.		
5. Fin del caso de uso.		
Observaciones: El usuario puede cancelar en cualquier momento el registro del nuevo usuario.		
Relaciones entre CU:		
Asociación de Inclusión: no aplica		
Asociación de Extensión: no aplica		
Caso de uso donde se incluye: no aplica		
Caso de uso al que extiende: no aplica		

CASO DE USO: 002. ELIMINAR USUARIO.



Descripción

Nombre del Caso de Uso: Eliminar usuario		Número: 002
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Eliminar un usuario registrado la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post- Condiciones:	Éxito 1: Usuario eliminado exitosamente.	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el usuario selecciona del menú, la opción “Usuarios”.		
2. El sistema muestra un filtro de búsqueda para encontrar el usuario que desea eliminar.		
3. El usuario ingresa el filtro de búsqueda.		
4. El sistema devuelve los usuarios que coincidan con el filtro ingresado.		
5. El usuario presiona “Eliminar” sobre el usuario que desea quitar de la base de datos.		
6. El sistema consulta si está seguro que desea eliminar al usuario.		
7. El usuario confirma la eliminación.		7.A. El usuario no confirma la eliminación 7.A.1. El sistema no elimina el usuario. 7.A.2. Fin del Caso de Uso.
8. El sistema elimina al usuario de la base de datos.		
9. Fin del caso de uso.		
Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: no aplica Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica		

CASO DE USO: 003. MODIFICAR USUARIO.

Descripción

Nombre del Caso de Uso: Modificar usuario		Número: 003
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Actualizar los datos de un usuario registrado la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post- Condiciones:	Éxito 1: Propiedades de usuario modificadas exitosamente.	
	Fracaso :	
Curso Normal		Alternativas



1. El caso de uso comienza cuando el usuario selecciona del menú, la opción “Usuarios”.	
2. El sistema muestra un filtro de búsqueda para encontrar el usuario que desea editar.	
3. El usuario ingresa el filtro de búsqueda.	
4. El sistema devuelve los usuarios que coincidan con el filtro ingresado.	
5. El usuario presiona “Editar” sobre el usuario que desea modificar.	
6. El sistema muestra las tres pestañas de propiedades del usuario en modo edición (Usuario, Grupos de Usuario y Proyectos).	
7. El usuario modifica los datos que desea actualizar del usuario y presiona “Guardar y Salir”.	7.A. El usuario no completa todos los campos requeridos. Presiona la opción “Guardar y Salir”. 7.A.1. El sistema informa la situación. 7.A.2. Continúa en el paso 7 del curso normal.
8. El sistema actualiza la información del usuario seleccionado exitosamente.	
9. Fin del caso de uso.	

Observaciones:

Obs.1: Los datos que pueden ser editados son:

- Nombre.
- Apellido.
- Email.
- Contraseña.
- Grupo de Usuario.
- Proyectos.

Obs.2: El usuario puede cancelar en cualquier momento la edición del nuevo usuario, presionando Cancelar

Relaciones entre CU:

Asociación de Inclusión: no aplica

Asociación de Extensión: no aplica

Caso de uso donde se incluye: no aplica

Caso de uso al que extiende: no aplica

CASO DE USO: 004. MODIFICAR CONTRASEÑA USUARIO

Descripción

Nombre del Caso de Uso: Modificar contraseña de Usuario		Número: 004
Actor Principal: Usuario		Actor Secundario: Administrador
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Registrar la actualización de la clave personal del usuario en la Base de Datos del sistema.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post-Condicion:	Éxito 1: Actualización de la clave personal del usuario exitosamente.	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el usuario presiona sobre su nombre en la parte superior derecha de la ventana y selecciona la opción “Perfil” y luego “Cambiar contraseña”		1.A. El caso de uso comienza cuando el Usuario se encuentra en la pantalla de inicio de sesión, ingresa su email y presiona la opción “¿Olvido su contraseña?”



	1.A.1. El sistema envía un email a la dirección proporcionada con la nueva contraseña generada automáticamente. 1.A.2. Fin del caso de uso.
2. El sistema muestra un pop up y solicita que se ingrese: <ul style="list-style-type: none"> • Clave Actual. • Nueva Clave. • Reingresar la nueva clave. 	
3. El usuario ingresa: <ul style="list-style-type: none"> • Clave Actual. • Nueva Clave. • Reingresar la nueva clave. Y presiona “Actualizar”.	
4. El sistema valida la información, actualiza la contraseña del usuario e informa la operación exitosa.	4.A. El sistema valida los datos ingresados y uno o más datos son incorrectos. 4.A.1. El sistema informa la situación. 4.A.2. Continúa en el paso 2 del curso normal.
5. Fin del caso de uso.	
Reglas de Negocio Asociadas:	
Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: no aplica Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica	

CASO DE USO: 005.INICIAR SESIÓN
Descripción

Nombre del Caso de Uso: Iniciar Sesión		Número: 005
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input type="checkbox"/> Simple <input type="checkbox"/> Mediano <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Validar las credenciales del usuario y registrar su inicio de sesión en el sistema.		
Pre-condiciones: El Usuario debe estar registrado en la aplicación.		
Post-Condicion:	Éxito 1: Inicio de Sesión de usuario exitoso.	
	Fracaso:	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el usuario ingresa a la pantalla de "Inicio de Sesión".		
2. El sistema solicita se ingrese: <ul style="list-style-type: none"> • Mail • Contraseña 		
3. El usuario ingresa los datos solicitados válidos, y presiona “Iniciar Sesión”.		3.A. El usuario ingresa uno o más datos inválidos. 3.A.1. El sistema informa que los datos no son correctos. 3.A.2. Continúa en el paso 2 del curso normal. 3.B. El usuario presiona la opción <u>¿Olvido su contraseña?</u> 3.B.1. El sistema llama al Caso de Uso: Modificar Contraseña de Usuario.



4. El Sistema verifica los datos ingresados por el usuario, e inicia la sesión acorde a los permisos que posee el usuario registrado.	
5. Fin del caso de uso.	
Observaciones:	
Reglas de Negocio Asociadas:	
Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: <i>Modificar Contraseña de Usuario.</i> Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica	

CASO DE USO: 006. CERRAR SESIÓN
Descripción

Nombre del Caso de Uso: Cerrar Sesión		Número: 006
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		
Complejidad: <input type="checkbox"/> Simple <input type="checkbox"/> Mediano <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Finalizar la sesión del usuario en el sistema.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación.		
Post-Condición:	Éxito 1: Registro cierre de sesión satisfactoriamente.	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el usuario selecciona desde la parte superior de la ventana, dentro del menú desplegable en el nombre del usuario logueado, la opción "Cerrar Sesión".		
2. El sistema solicita confirmación para el cierre de sesión.		
3. El usuario confirma el cierre de la sesión.		
4. El sistema informa cierre de sesión.		
5. Fin del caso de uso.		
Observaciones:		
Reglas de Negocio Asociadas:		
Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: no aplica Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica		

CASO DE USO: 007. REGISTRAR PROYECTO
Descripción

Nombre del Caso de Uso: Registrar Proyecto		Número: 007
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		



Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo	
Objetivo: Registrar los datos de un nuevo proyecto en la base de datos.	
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.	
Post- Condiciones:	Éxito 1: Proyecto almacenado en la base de datos.
	Fracaso :
Curso Normal	
Alternativas	
1. El caso de uso comienza cuando el Usuario ingresa al módulo "Proyectos" y presiona "Agregar Proyecto".	
2. El sistema muestra el formulario de alta de proyecto con estado seleccionado en "Creado" y solicita que se ingrese un Nombre al Proyecto, una descripción del mismo, seleccionar una fecha de inicio y una fecha de fin, ingresar observaciones y seleccionar un "Líder de Proyecto".	
3. El Usuario ingresa/selecciona los datos requeridos y presiona "Guardar".	3.A. El Usuario no ingresa/selecciona alguno de los datos requeridos y presiona "Guardar". 3.A.1. El sistema informa que faltan datos obligatorios. 3.A.2. Continúa en el paso 3 del curso normal.
4. El sistema valida que los datos ingresados son correctos, asigna un identificador y almacena el nuevo proyecto en la base de datos con Estado "Creado".	
5. Fin del caso de uso.	
Observaciones:	
Relaciones entre CU:	
Asociación de Inclusión: no aplica	
Asociación de Extensión: no aplica	
Caso de uso donde se incluye: no aplica	
Caso de uso al que extiende: no aplica	

CASO DE USO: 008. ELIMINAR PROYECTO
Descripción

Nombre del Caso de Uso: Eliminar Proyecto		Número: 008	
Actor Principal: Usuario		Actor Secundario:	
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto			
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja			
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo			
Objetivo: Eliminar un proyecto y sus datos asociados de la base de datos.			
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.			
Post- Condiciones:	Éxito 1: Proyecto eliminado exitosamente de la base de datos		
	Fracaso :		
Curso Normal		Alternativas	
1. El caso de uso comienza cuando el Usuario ingresa al módulo "Proyectos".			
2. El sistema muestra los filtros de búsqueda disponibles para la consulta de proyectos.			
3. El usuario ingresa/selecciona los filtros de búsqueda deseados y presiona Buscar.			
4. El sistema busca y muestra los proyectos que coinciden con el filtro de búsqueda.			



5. El usuario selecciona “Eliminar” sobre el Proyecto que desea borrar de la base de datos.	
6. El sistema consulta si está seguro que desea eliminar el proyecto.	
7. El usuario confirma la eliminación del proyecto.	7.A. El usuario no confirma la eliminación del proyecto 7.A.1. Fin del caso de uso.
8. El sistema elimina el proyecto de la base de datos y todos sus elementos asociados.	
9. Fin del Caso de Uso	
Observaciones:	
Relaciones entre CU:	
Asociación de Inclusión: no aplica	
Asociación de Extensión: no aplica	
Caso de uso donde se incluye: no aplica	
Caso de uso al que extiende: no aplica	

CASO DE USO: 009. MODIFICAR PROYECTO
Descripción

Nombre del Caso de Uso: Modificar Proyecto		Número: 009
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Actualizar los datos de un proyecto registrado en la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post-Condicion:	Éxito 1: Proyecto actualizado exitosamente.	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Proyectos”.		
2. El sistema muestra los filtros de búsqueda disponibles para la consulta de proyectos.		
3. El usuario ingresa/selecciona los filtros de búsqueda deseados y presiona Buscar.		
4. El sistema busca y muestra los proyectos que coinciden con el filtro de búsqueda.		
5. El usuario selecciona “Editar” sobre el Proyecto al cual desea actualizar su información.		5.A. El usuario selecciona “Detalle” sobre el Proyecto que desea visualizar. 5.A.1. El sistema muestra las características del proyecto. 5.A.2. Fin del Caso de Uso.
6. El sistema muestra las propiedades del proyecto en modo edición.		
7. El usuario modifica los datos deseados y presiona Guardar.		
8. El sistema actualiza la información del proyecto en la base de datos.		
9. Fin del Caso de Uso		
Observaciones:		



Relaciones entre CU:

Asociación de Inclusión: no aplica
Asociación de Extensión: no aplica
Caso de uso donde se incluye: no aplica
Caso de uso al que extiende: no aplica

CASO DE USO: 010. REGISTRAR REQUERIMIENTO
Descripción

Nombre del Caso de Uso: Registrar Requerimiento		Número: 010
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Registrar los datos de un nuevo requerimiento en la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post- Condiciones:	Éxito 1: Requerimiento registrado exitosamente y asignado a un proyecto.	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo "Requerimientos" y presiona "Agregar Requerimiento".		
2. El sistema solicita se seleccione un proyecto para el cual se va a crear el requerimiento.		
3. El Usuario selecciona el proyecto para el cual desea registrar el nuevo requerimiento.		
4. El sistema solicita para el nuevo requerimiento, se ingrese un nombre, un objetivo, se seleccione un tipo y opcionalmente adjuntar un archivo al requerimiento (Obs.1) .		
5. El usuario ingresa los datos requeridos y presiona "Guardar y Salir".		5.A. El usuario ingresa los datos requeridos y presiona "Guardar". 5.A.1. El sistema agrega el requerimiento al proyecto asignándole un código identificador; y continua en el paso 4 del curso normal. (Obs.2)
6. El sistema agrega el requerimiento a la lista, asignándole un identificador generado automáticamente.		
7. Fin del caso de uso.		

Observaciones:

(Obs.1): El archivo adjunto no es un campo obligatorio para el registro del requerimiento.

(Obs.2): La opción "Guardar" permite la carga de requerimientos para el mismo Proyecto seleccionado, sin necesidad de volver a ingresar.

Relaciones entre CU:

Asociación de Inclusión: no aplica
Asociación de Extensión: no aplica
Caso de uso donde se incluye: no aplica
Caso de uso al que extiende: no aplica



CASO DE USO: 011. ELIMINAR REQUERIMIENTO
Descripción

Nombre del Caso de Uso: Eliminar Requerimiento		Número: 011
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Eliminar un requerimiento y sus datos asociados de la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post-Condicion:	Éxito 1: Requerimiento eliminado exitosamente de la base de datos.	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo "Requerimientos".		
2. El sistema muestra los filtros de búsqueda disponibles para la consulta de requerimientos.		
3. El usuario ingresa/selecciona los filtros de búsqueda deseados y presiona Buscar.		
4. El sistema busca y muestra los requerimientos que coinciden con el filtro de búsqueda.		
5. El usuario selecciona "Eliminar" sobre el Requerimiento que desea borrar de la base de datos.		
6. El sistema consulta si está seguro que desea eliminar el requerimiento.		
7. El usuario confirma la eliminación del requerimiento.		7.A. El usuario no confirma la eliminación del requerimiento 7.A.1. Fin del caso de uso.
8. El sistema elimina el requerimiento de la base de datos y todos sus elementos asociados.		
9. Fin del Caso de Uso		
Observaciones:		
Relaciones entre CU:		
Asociación de Inclusión: no aplica		
Asociación de Extensión: no aplica		
Caso de uso donde se incluye: no aplica		
Caso de uso al que extiende: no aplica		

CASO DE USO: 012. MODIFICAR REQUERIMIENTO
Descripción

Nombre del Caso de Uso: Modificar Requerimiento		Número: 012
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Actualizar los datos de un requerimiento registrado en la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		



Post- Condiciones:	Éxito 1:
	Fracaso :
Curso Normal	
Alternativas	
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Requerimientos”.	
2. El sistema muestra los filtros de búsqueda disponibles para la consulta de requerimientos.	
3. El usuario ingresa/selecciona los filtros de búsqueda deseados y presiona Buscar.	
4. El sistema busca y muestra los requerimientos que coinciden con el filtro de búsqueda.	
5. El usuario selecciona “Editar” sobre el Requerimiento al cual desea actualizar su información.	5.A. El usuario selecciona “Detalle” sobre el Requerimiento que desea visualizar. 5.A.1. El sistema muestra las características del requerimiento. 5.A.2. Fin del Caso de Uso.
6. El sistema muestra las propiedades del requerimiento en modo edición.	
7. El usuario modifica los datos que desea y presiona Guardar.	
8. El sistema actualiza la información del requerimiento en la base de datos.	
9. Fin del Caso de Uso	
Observaciones:	
Relaciones entre CU:	
Asociación de Inclusión: no aplica	
Asociación de Extensión: no aplica	
Caso de uso donde se incluye: no aplica	
Caso de uso al que extiende: no aplica	

CASO DE USO: 013. REGISTRAR CASO DE PRUEBA
Descripción

Nombre del Caso de Uso: Registrar Caso de Prueba		Número: 013
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Registrar los datos de un nuevo caso de pruebas en la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post- Condiciones:	Éxito 1: Caso de Prueba registrado exitosamente	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Casos de Prueba” y presiona “Agregar Caso”.		
2. El sistema solicita se seleccione un proyecto para el cual se va a crear el caso de prueba.		
3. El Usuario selecciona el proyecto para el cual desea registrar el nuevo caso de prueba.		



4. El sistema busca los requerimientos asociados al proyecto seleccionado y solicita se seleccione el requerimiento para el cual desea crear el caso de prueba.	
5. El usuario selecciona el requerimiento.	
6. El sistema solicita se ingrese un nombre al caso de prueba.	
7. El usuario ingresa un nombre para el caso de prueba.	
8. El sistema solicita se seleccione una Categoría de Prueba.	
9. El usuario selecciona la Categoría “Funcional”	<p>9.A. El usuario selecciona otra Categoría.</p> <p>9.A.1. El sistema busca y muestra la información asociada a la categoría seleccionada y solicita se complete la información faltante.</p> <p>9.A.2. El usuario ingresa/selecciona la información restante y presiona “Guardar y Salir”.</p> <p>9.A.3. El sistema almacena el Caso de prueba en la base de datos.</p> <p>9.A.4. Fin del caso de uso.</p>
10. El sistema solicita se ingrese: <ul style="list-style-type: none"> • Objetivo. • Pre-Condiciones • Pasos requeridos para la ejecución. • Resultado Esperado. 	
11. El usuario ingresa los datos requeridos y presiona “Guardar y Salir”.	<p>11.A. El usuario ingresa los datos requeridos y presiona “Guardar”.</p> <p>11.A.1. El sistema agrega el caso de prueba al requerimiento y continúa en el paso 6 del curso normal.</p>
12. El sistema almacena el caso de prueba en la base de datos.	
13. Fin del caso de uso.	
Observaciones: El nuevo caso de uso se genera con un identificador único generado automáticamente por el sistema.	
<p>Relaciones entre CU:</p> <p>Asociación de Inclusión: no aplica</p> <p>Asociación de Extensión: no aplica</p> <p>Caso de uso donde se incluye: no aplica</p> <p>Caso de uso al que extiende: no aplica</p>	

CASO DE USO: 014. ELIMINAR CASO DE PRUEBA
Descripción

Nombre del Caso de Uso: Eliminar Caso de Prueba	Número: 014
Actor Principal: Usuario	Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte	Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo	
Objetivo: Eliminar un caso de prueba y sus datos asociados de la base de datos.	
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.	
Éxito 1: Caso de Prueba eliminado exitosamente de la base de datos	



Post-Condiciones:	Fracaso :
Curso Normal	Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Casos de Prueba”.	
2. El sistema muestra los filtros de búsqueda disponibles para la consulta de casos de prueba.	
3. El usuario ingresa/selecciona los filtros de búsqueda deseados y presiona Buscar.	
4. El sistema busca y muestra los casos de prueba que coinciden con el filtro de búsqueda.	
5. El usuario selecciona “Eliminar” sobre el Caso de Prueba que desea borrar de la base de datos.	
6. El sistema consulta si está seguro que desea eliminar el caso de prueba.	
7. El usuario confirma la eliminación del caso de prueba.	7.A. El usuario no confirma la eliminación del caso de prueba 7.A.1. Fin del caso de uso.
8. El sistema elimina el caso de prueba de la base de datos y todos sus elementos asociados.	
9. Fin del Caso de Uso	
Observaciones:	
Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: no aplica Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica	

CASO DE USO: 015. MODIFICAR CASO DE PRUEBA
Descripción

Nombre del Caso de Uso: Modificar Caso de Prueba		Número: 015
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	
Categoría:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte	Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad:	<input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo	
Objetivo: Editar los datos de un caso de prueba registrado en la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post-Condiciones:	Éxito 1: Caso de Prueba actualizado exitosamente	
	Fracaso :	
Curso Normal	Alternativas	
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Casos de Prueba”.		
2. El sistema muestra los filtros de búsqueda disponibles para la consulta de casos de prueba.		
3. El usuario ingresa/selecciona los filtros de búsqueda deseados y presiona Buscar.		
4. El sistema busca y muestra los casos de prueba que coinciden con el filtro de búsqueda.		



5. El usuario selecciona “Editar” sobre el caso de prueba al cual desea actualizar su información.	5.A. El usuario selecciona “Detalle” sobre el caso de prueba que desea visualizar. 5.A.1. El sistema muestra las características del caso de prueba. 5.A.2. Fin del Caso de Uso.
6. El sistema muestra las propiedades del caso de prueba en modo edición.	
7. El usuario modifica los datos que desea y presiona Guardar.	
8. El sistema actualiza la información del caso de prueba en la base de datos.	
9. Fin del Caso de Uso	

Observaciones:

Relaciones entre CU:

Asociación de Inclusión: no aplica
Asociación de Extensión: no aplica
Caso de uso donde se incluye: no aplica
Caso de uso al que extiende: no aplica

CASO DE USO: 016. COPIAR CASO DE PRUEBA
Descripción

Nombre del Caso de Uso: Copiar Caso de Prueba		Número: 016
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media <input type="checkbox"/> Baja
Categoría:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Soporte
Complejidad:	<input checked="" type="checkbox"/> Simple	<input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo
Objetivo:	Duplicar los datos de un caso de prueba ya registrado en la base de datos, bajo un nuevo nombre.	
Pre-condiciones:	El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.	
Post-Condición:	Éxito 1: Caso de prueba copiado exitosamente	
	Fracaso:	

Curso Normal	Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Casos de Prueba”.	
2. El sistema muestra los filtros de búsqueda.	
3. El usuario ingresa los filtros deseados y presiona buscar.	
4. El sistema muestra los ciclos de prueba que coinciden con los filtros ingresados.	
5. El Usuario selecciona “Copiar” sobre el caso de prueba que desea duplicar.	
6. El sistema solicita que se ingrese un nombre para el caso de prueba nuevo.	
7. El usuario ingresa el nombre del nuevo caso de prueba y presiona “Copiar”.	7.A. El usuario presiona “Cancelar”. 7.A.1. El sistema retorna a la ventana de búsqueda del módulo Casos de Prueba, sin generar la copia del caso. 7.A.2. Fin del Caso de Uso.
8. El sistema almacena el nuevo caso de prueba en la base de datos con el nombre ingresado y las características del caso copiado.	



9. Fin del caso de uso.

Relaciones entre CU:

Asociación de Inclusión: no aplica
Asociación de Extensión: no aplica
Caso de uso donde se incluye: no aplica
Caso de uso al que extiende: no aplica

CASO DE USO: 017. REGISTRAR CICLO DE PRUEBA
Descripción

Nombre del Caso de Uso: Registrar Ciclo de Prueba		Número: 017
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Registrar los datos de un nuevo ciclo de pruebas en la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post- Condiciones:	Éxito 1: Ciclo de prueba registrado exitosamente en la base de datos	
	Fracaso :	

Curso Normal	Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Ciclos de Prueba” y presiona “Agregar Ciclo”.	
2. El sistema muestra la ventana de “Nuevo Ciclo de Prueba” y solicita se seleccione un proyecto para el cual se va a crear el ciclo de prueba.	
3. El Usuario selecciona el proyecto para el cual desea registrar el nuevo ciclo de prueba.	
4. El sistema solicita se ingrese el nombre, ambiente, fecha inicio, fecha fin y observaciones del ciclo de prueba.	
5. El usuario ingresa el nombre, ambiente, fecha inicio, fecha fin, observaciones del ciclo de prueba y presiona la opción “Casos de Prueba”	
6. El sistema llama al Caso de Uso “Asignar pruebas a ciclo”.	
7. El usuario presiona “Guardar y Salir”	7.A. El usuario presiona “Guardar”. 7.A.1. El sistema almacena el ciclo de prueba en la base de datos con los casos de pruebas asignados y permanece en la ventana de Registro de Ciclo de Prueba. 7.B. El usuario presiona “Cancelar”. 7.B.1. El sistema cancela la registración del ciclo de prueba y regresa a la ventana principal de “Ciclos de Prueba” 7.B.2. Fin del Caso de Uso.
8. El sistema almacena el ciclo de prueba en la base de datos con los casos de prueba asignados al mismo y cierra la ventana de Registro de Ciclo de prueba.	
9. Fin del caso de uso.	

Relaciones entre CU:

Asociación de Inclusión: Asignar pruebas a ciclo
Asociación de Extensión: no aplica
Caso de uso donde se incluye: no aplica
Caso de uso al que extiende: no aplica



CASO DE USO: 018. ELIMINAR CICLO DE PRUEBA
Descripción

Nombre del Caso de Uso: Eliminar Ciclo de Prueba		Número: 018
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Eliminar un ciclo de prueba y sus datos asociados de la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post- Condiciones:	Éxito 1: Ciclo de Prueba eliminado exitosamente	
	Fracaso:	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo "Ciclos de Prueba".		
2. El sistema muestra los filtros de búsqueda.		
3. El usuario ingresa los filtros deseados y presiona buscar.		
4. El sistema muestra los ciclos de prueba que coinciden con los filtros ingresados.		
5. El Usuario selecciona "Eliminar" sobre el ciclo de prueba que desea eliminar de la base de datos.		
10. El sistema consulta si está seguro que desea eliminar el ciclo de prueba.		
11. El usuario confirma la eliminación del ciclo de prueba.		11.A. El usuario no confirma la eliminación del ciclo de prueba 11.A.1. Fin del caso de uso.
12. El sistema elimina el ciclo de prueba de la base de datos y todos sus elementos asociados.		
6. Fin del caso de uso.		
Relaciones entre CU:		
Asociación de Inclusión:		
Asociación de Extensión: no aplica		
Caso de uso donde se incluye: no aplica		
Caso de uso al que extiende: no aplica		

CASO DE USO: 019. MODIFICAR CICLO DE PRUEBA
Descripción

Nombre del Caso de Uso: Modificar Ciclo de Prueba		Número: 019
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Actualizar los datos de un ciclo de prueba registrado en la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post- Condiciones:	Éxito 1: Ciclo de prueba actualizado exitosamente	
	Fracaso:	



Curso Normal	Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Ciclos de Prueba”.	
2. El sistema muestra los filtros de búsqueda.	
3. El usuario ingresa los filtros deseados y presiona buscar.	
4. El sistema muestra los ciclos de prueba que coinciden con los filtros ingresados.	
5. El Usuario selecciona “Editar” sobre el ciclo de prueba que desea actualizar la información.	
6. El sistema muestra el ciclo de prueba en modo edición.	
7. El usuario modifica los datos deseados y presiona “Guardar y Salir”.	7.A. El usuario modifica los datos deseados y/o presiona “Casos de Pruebas”. 7.A.1. El sistema llama al caso de uso “Asignar pruebas a ciclo”. 7.A.2. El usuario presiona “guardar y Salir”. 7.B. El usuario presiona “Guardar”. 7.B.1. El sistema almacena el ciclo de prueba en la base de datos”. 7.B.2. Continúa en el paso 4 del curso normal.
8. El sistema almacena el ciclo de prueba en la base de datos.	
9. Fin del caso de uso.	
Relaciones entre CU: Asociación de Inclusión: <i>Caso de Uso: Asignar Pruebas a Ciclo</i> Asociación de Extensión: no aplica Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica	

CASO DE USO: 020. COPIAR CICLO DE PRUEBA
Descripción

Nombre del Caso de Uso: Copiar Ciclo de Prueba		Número: 020
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media <input type="checkbox"/> Baja
Categoría:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Soporte
Complejidad:	<input checked="" type="checkbox"/> Simple	<input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo
Objetivo: Duplicar los datos de un ciclo de prueba ya registrado en la base de datos, bajo un nuevo nombre.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post-Condición:	Éxito 1: Ciclo de prueba copiado exitosamente	
Post-Condición:	Fracaso:	
Curso Normal	Alternativas	
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Ciclos de Prueba”.		
2. El sistema muestra los filtros de búsqueda.		
3. El usuario ingresa los filtros deseados y presiona buscar.		
4. El sistema muestra los ciclos de prueba que coinciden con los filtros ingresados.		
5. El Usuario selecciona “Copiar” sobre el ciclo de prueba que desea duplicar.		



6. El sistema solicita que se ingrese un nombre para el ciclo de prueba nuevo.	
7. El usuario ingresa el nombre del nuevo ciclo de prueba y presiona “Copiar”.	7.A. El usuario presiona “Cancelar”. 7.A.1. El sistema retorna a la ventana de búsqueda del módulo Ciclo de Prueba, sin generar la copia del ciclo. 7.A.2. Fin del Caso de Uso.
8. El sistema almacena el nuevo ciclo de prueba en la base de datos con el nombre ingresado y las características del ciclo copiado.	
9. Fin del caso de uso.	
Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: no aplica Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica	

CASO DE USO: 021. ASIGNAR PRUEBAS A CICLO

Nombre del Caso de Uso: Asignar pruebas a ciclo.		Número: 021
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	
Categoría:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte	Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad:	<input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo	
Objetivo: Asociar casos de pruebas registrados a un ciclo de pruebas creado en la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post-Condición:	Éxito 1: Casos de pruebas asignados exitosamente a un ciclo	
	Fracaso:	
	Curso Normal	Alternativas
1. El caso de uso comienza cuando el Usuario ha seleccionado la opción “Casos de Prueba” dentro de un ciclo de pruebas		
2. El sistema muestra el árbol de requerimientos y casos de prueba asociados al proyecto para el cual se creó el ciclo.		
3. El usuario selecciona los Casos de Prueba que desea agregar al ciclo de prueba y presiona “Volver”.		3.A. El usuario no selecciona ningún caso de Prueba y presiona “Cancelar”.
4. Fin del caso de uso.		
Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: no aplica Caso de uso donde se incluye: <i>Registrar Ciclo de Prueba / Modificar Ciclo de Prueba</i> Caso de uso al que extiende: no aplica		

CASO DE USO: 022. REGISTRAR DETERMINACIÓN DE PRUEBA
Descripción

Nombre del Caso de Uso: Registrar Determinación de Prueba	Número: 022
Actor Principal: Usuario	Actor Secundario:



Tipo de Caso de Uso:		<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Prioridad:		<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media <input type="checkbox"/> Baja
Categoría:		<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Soporte
		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Complejidad: <input type="checkbox"/> Simple <input type="checkbox"/> Mediano <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo			
Objetivo: Registrar el estado de la ejecución de una prueba dentro de un ciclo de prueba.			
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.			
Post- Condiciones:	Éxito 1: Determinación de prueba registrada exitosamente		
	Fracaso:		
Curso Normal		Alternativas	
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Ejecuciones”.			
2. El sistema muestra los filtros de búsqueda.			
3. El usuario ingresa los filtros deseados y presiona buscar.			
4. El sistema muestra los ciclos de prueba que coinciden con los filtros ingresados. (Obs.1)			
5. El Usuario selecciona “Ejecutar” sobre el ciclo de prueba que desea ejecutar.		5.A. El usuario selecciona la opción “Ejecutar” y luego “Detalle” sobre el caso que desea visualizar el detalle. 5.A.1. El sistema muestra el detalle de la ejecución del caso de prueba seleccionado. (Obs.2)	
6. El sistema abre una nueva ventana y solicita se seleccione un caso de prueba.			
7. El usuario selecciona el caso de prueba que desea ejecutar.			
8. El sistema muestra para el caso de prueba seleccionado, la descripción de pasos para la ejecución, el resultado esperado, la opción para seleccionar un estado de la ejecución (Obs.3) , la opción para adjuntar archivos de evidencia de la ejecución y un campo para ingresar Observaciones a la ejecución.			
9. El usuario selecciona un estado para el caso de prueba ejecutado, adjunta evidencias a la ejecución e ingresa una observación de la ejecución realizada. Luego presiona “Guardar”		9.A. El usuario selecciona un estado para el caso de prueba ejecutado, adjunta evidencias a la ejecución e ingresa una observación de la ejecución realizada. Luego presiona “Guardar y Salir” 9.A.1. El sistema guarda los datos ingresados de la ejecución y retorna a la pantalla de “Ejecuciones” 9.B. El usuario selecciona un estado para el caso de prueba ejecutado, adjunta evidencias de la ejecución, ingresa una observación de la ejecución realizada y presiona “Registrar Incidencia”. 9.B.1. El sistema llama al caso de uso “ Registrar Incidencia ”. 9.B.2. El usuario presiona “Guardar”/”Guardar y Salir”.	
10. El sistema guarda los datos ingresados de la ejecución, actualiza el estado de la ejecución y el porcentaje de ejecución del ciclo.			



11. Fin del caso de uso.
<p>Observaciones: (Obs.1): Para cada ciclo se informa:</p> <ul style="list-style-type: none"> • Nombre • Proyecto • Fecha Inicio • Fecha de Fin • Porcentaje de Ejecución • Acciones → Ejecutar <p>(Obs.2): Los posibles estados de un caso de prueba son:</p> <ul style="list-style-type: none"> • Exitoso • Fallido • No Ejecutado
<p>Relaciones entre CU: Asociación de Inclusión: Asociación de Extensión: Registrar Incidencia Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica</p>

CASO DE USO: 023. REGISTRAR INCIDENCIA
Descripción

Nombre del Caso de Uso: Registrar Incidencia		Número: 023
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media <input type="checkbox"/> Baja
Categoría:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Soporte Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad:	<input type="checkbox"/> Simple <input type="checkbox"/> Mediano <input checked="" type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo	
Objetivo: Registrar los datos de una nueva incidencia en la base de datos.		
Pre-condiciones: El usuario debe estar logueado en la aplicación. Usuario debe poseer permisos para registrar incidencia.		
Post-Condición:	Éxito 1: Incidencia registrada exitosamente.	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el usuario ingresa al módulo "incidencias" y presiona "Agregar Incidencia"		1.A. El caso de uso comienza cuando el usuario dentro de la ejecución de un caso de prueba presiona "Registrar Incidencia".
2. El sistema muestra el formulario de registro de la incidencia con los siguientes campos: (Obs.1)		
<ul style="list-style-type: none"> • Nombre/Resumen • Proyecto • Ciclo de Prueba • Requerimiento • Caso de Prueba • Clave (Generado automáticamente) • Tipo • Severidad • Prioridad • Asignada a • Estado (por defecto al crearla en "CREADA").(Obs.2) • Evidencias • Observaciones 		



<p>3. El usuario completa los campos obligatorios (sin asignarla a ningún usuario) para el registro de la incidencia, y presiona “Guardar y Salir”.</p>	<p>3.A. El usuario no completa todos los datos obligatorios y presiona Guardar. 3.A.1. El sistema informa que debe completar los campos obligatorios faltantes. 3.A.2. Continúa en el paso 3 del Curso Normal. 3.B. El usuario completa los campos obligatorios para la incidencia, selecciona un usuario a quien asignar la incidencia y presiona “Guardar”. 3.B.1. El sistema genera la incidencia en estado “ASIGNADA” y notifica a la persona a quien se asignó mediante un mail. 3.B.2. Fin del Caso de Uso.</p>
<p>4. El sistema genera la incidencia en estado “CREADA”.</p>	
<p>5. Fin del caso de uso.</p>	
<p>Observaciones: (Obs.1): Los datos obligatorios para registrar la incidencia son:</p> <ul style="list-style-type: none"> • Proyecto • Ciclo de Prueba • Clave • Tipo • Severidad • Nombre/Resumen • Estado <p>Obs.2: Los estados posibles de una incidencia son:</p> <ul style="list-style-type: none"> • Creada • Asignada • En Tratamiento • Solucionada • Verificada • Devuelta • Cerrada 	
<p>Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: no aplica Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica</p>	

CASO DE USO: 024. ELIMINAR INCIDENCIA
Descripción

<p>Nombre del Caso de Uso: Eliminar Incidencia</p>		<p>Número: 024</p>
<p>Actor Principal: Usuario</p>		<p>Actor Secundario:</p>
<p>Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto</p>		
<p>Prioridad: <input type="checkbox"/> Alta <input checked="" type="checkbox"/> Media <input type="checkbox"/> Baja</p>		
<p>Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No</p>		
<p>Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo</p>		
<p>Objetivo: Eliminar una incidencia y sus datos asociados de la base de datos.</p>		
<p>Pre-condiciones: Usuario logueado con permisos para llevar a cabo esta acción.</p>		
<p>Post-Condiciones: Éxito 1: Incidencia eliminada de la base de datos.</p>		
<p>Fracaso :</p>		
<p>Curso Normal</p>		<p>Alternativas</p>
<p>1. El caso de uso comienza cuando el Usuario ingresa al módulo “Incidencias”.</p>		



2. El sistema muestra los filtros de búsqueda disponibles para la consulta de incidencias.	
3. El usuario ingresa/selecciona los filtros de búsqueda deseados y presiona Buscar.	
4. El sistema busca y muestra las incidencias que coinciden con el filtro de búsqueda.	
5. El usuario selecciona “Eliminar” sobre la incidencia que desea eliminar de la base de datos.	
6. El sistema consulta si está seguro que desea eliminar la incidencia seleccionada.	
7. El usuario confirma la eliminación de la incidencia.	7.A. El usuario No confirma la eliminación. 7.A.1. Se cancela el Caso de Uso.
8. El sistema elimina la incidencia de la base de datos.	
9. Fin del Caso de Uso	
Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: no aplica Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica	

CASO DE USO: 025. MODIFICAR INCIDENCIA
Descripción

Nombre del Caso de Uso: Modificar Incidencia		Número: 025
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto	<input type="checkbox"/> Abstracto
Prioridad:	<input checked="" type="checkbox"/> Alta	<input type="checkbox"/> Media <input type="checkbox"/> Baja
Categoría:	<input checked="" type="checkbox"/> Esencial	<input type="checkbox"/> Soporte Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad:	<input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo	
Objetivo: Editar los datos de una incidencia registrada en la base de datos.		
Pre-condiciones: Usuario logueado con permisos para llevar a cabo esta acción.		
Post-Condición:	Éxito 1: Incidencia actualizada exitosamente	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Incidencias”.		
2. El sistema muestra los filtros de búsqueda disponibles para la consulta de incidencias.		
3. El usuario ingresa/selecciona los filtros de búsqueda deseados y presiona Buscar.		
4. El sistema busca y muestra las incidencias que coinciden con el filtro de búsqueda.		
5. El usuario selecciona “Editar” sobre la incidencia a la cual desea actualizar su información.		5.A. El usuario selecciona “Detalle” sobre la incidencia que desea visualizar. 5.A.1. El sistema muestra las características de la incidencia. 5.A.2. Fin del Caso de Uso.
6. El sistema muestra las propiedades de la incidencia en modo edición.		
7. El usuario modifica los datos que desea y presiona Guardar. (Obs.1)		7.A. El usuario modifica el estado de la incidencia y asigna la misma a un usuario y presiona Guardar.



	7.A.1. El sistema actualiza la información de la incidencia y envía la notificación a la persona a quien fue asignada. 7.A.2. Fin del Caso de Uso.
8. El sistema actualiza la información de la incidencia.	
9. Fin del Caso de Uso	
Observaciones: Obs.1: Los campos que pueden editarse de la incidencia son: <ul style="list-style-type: none"> • Tipo • Severidad • Prioridad • Nombre/Resumen • Descripción • Archivo Adjunto • Asignada a • Estado (por defecto al crearla en “CREADA”). Obs.2: Los estados posibles de una incidencia son: <ul style="list-style-type: none"> • Creada • Asignada • En Tratamiento • Solucionada • Verificada • Devuelta • Cerrada 	
Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: no aplica Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica	

CASO DE USO: 026. GENERAR REPORTE GENERAL
Descripción

Nombre del Caso de Uso: Generar reporte general		Número: 026
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	
Categoría:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte	Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad:	<input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo	
Objetivo: Generar reporte de métricas generales por proyecto		
Pre-condiciones: Usuario logueado con permisos para llevar a cabo esta acción.		
Post-Condicion:	Éxito 1: Reporte generado exitosamente	
	Fracaso :	
	Curso Normal	Alternativas
1.	El caso de uso comienza cuando el Usuario ingresa al módulo “Métricas”.	
2.	El sistema despliega un menú para seleccionar la métrica deseada.	
3.	El usuario selecciona la opción “Generales”.	
4.	El sistema solicita que el usuario seleccione el proyecto para el cual desea obtener las métricas.	
5.	El usuario selecciona el proyecto para el cual desea obtener las métricas.	



6. El sistema presenta un informe con las estadísticas de pruebas diferenciadas por “Totales por Ciclos de Prueba” y “Totales por Tester”	
7. El usuario presiona la opción “Ver Gráfico” dentro de las métricas “Totales por Ciclos de Prueba”.	7.A. El usuario presiona la opción “Ver Gráfico” dentro de las métricas “Totales por Tester”. 7.A.1. El sistema muestra el grafico correspondiente a la métrica seleccionada. 7.A.2. Fin del Caso de Uso.
8. El sistema muestra el grafico correspondiente a la métrica seleccionada.	
9. Fin del Caso de Uso	
Observaciones: No Aplica	
Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: no aplica Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica	

CASO DE USO: 027. GENERAR REPORTE INCIDENCIAS
Descripción

Nombre del Caso de Uso: Generar reporte incidencias		Número: 027
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso:	<input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto	
Prioridad:	<input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja	
Categoría:	<input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte	Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad:	<input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo	
Objetivo: Generar reporte grafico de incidencias registradas para un proyecto.		
Pre-condiciones: Usuario logueado con permisos para llevar a cabo esta acción.		
Post-Condicion:	Éxito 1: Reporte generado exitosamente.	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Métricas”.		
2. El sistema despliega un menú para seleccionar la métrica deseada.		
3. El usuario selecciona la opción “Incidencias”.		
4. El sistema solicita que el usuario seleccione el proyecto para el cual desea obtener las métricas.		
5. El usuario selecciona el proyecto para el cual desea obtener las métricas.		
6. El sistema presenta gráficos por cada una de las siguientes categorías para las incidencias registradas en el proyecto: <ul style="list-style-type: none"> • Incidencias por persona asignada. • Incidencias por Tipo. • Incidencias por Severidad. • Incidencias por Estado. 		
7. Fin del Caso de Uso		
Observaciones: No Aplica		



Relaciones entre CU:

Asociación de Inclusión: no aplica
Asociación de Extensión: no aplica
Caso de uso donde se incluye: no aplica
Caso de uso al que extiende: no aplica

CASO DE USO: 028. GENERAR REPORTE DE AMPLITUD O COBERTURA DE PRUEBAS
Descripción

Nombre del Caso de Uso: Generar Reporte de Amplitud o Cobertura de Pruebas		Número: 028
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Generar reporte de amplitud y cobertura de pruebas definidas y realizadas por proyecto.		
Pre-condiciones: Usuario logueado con permisos para llevar a cabo esta acción.		
Post-Condición: Éxito 1: Reporte generado exitosamente.		
Fracaso :		
Curso Normal		Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Métricas”.		
2. El sistema despliega un menú para seleccionar la métrica deseada.		
3. El usuario selecciona la opción “Amplitud-Cobertura”.		
4. El sistema solicita que el usuario seleccione el proyecto para el cual desea obtener las métricas.		
5. El usuario selecciona el proyecto para el cual desea obtener las métricas.		
6. El sistema busca los ciclos de prueba asociados al proyecto y solicita al usuario que seleccione un ciclo.		
7. El usuario selecciona el ciclo de prueba para el cual desea obtener las métricas.		
8. El sistema muestra para el ciclo de prueba seleccionado, la siguiente información en forma tabular y gráfica: <ul style="list-style-type: none"> • Información del proyecto. • Recursos Asignados al proyecto. • Estadísticas por requerimiento. • Métricas Resumen. 		8.A. El sistema muestra para el ciclo de prueba seleccionado, la siguiente información: <ul style="list-style-type: none"> • Información del proyecto. • Recursos Asignados al proyecto. • Estadísticas por requerimiento. • Métricas Resumen. 8.A.1. El usuario presiona Imprimir 8.A.2. El sistema muestra la vista preliminar de impresión del Reporte.
9. Fin del Caso de Uso		

Observaciones: Como métricas resumen se establece:

- CPE: Número de casos de prueba que han sido ejecutados en total.
- CPR: Número de casos de prueba a ejecutar requeridos para cubrir todos los requerimientos.
- CPRT: Número de casos de prueba que se han ejecutado por cada requerimiento.

Relaciones entre CU:

Asociación de Inclusión: no aplica
Asociación de Extensión: no aplica



Caso de uso donde se incluye: no aplica
Caso de uso al que extiende: no aplica

CASO DE USO: 029. GENERAR REPORTE DE MADUREZ DE PRUEBAS
Descripción

Nombre del Caso de Uso: Generar Reporte de Madurez de Pruebas		Número: 029
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Generar reporte de la madurez alcanzada por las pruebas realizadas en un proyecto.		
Pre-condiciones: Usuario logueado con permisos para llevar a cabo esta acción.		
Post- Condiciones:	Éxito 1: Reporte generado exitosamente.	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Métricas”.		
2. El sistema despliega un menú para seleccionar la métrica deseada.		
3. El usuario selecciona la opción “Madurez de Pruebas”.		
4. El sistema solicita que el usuario seleccione el proyecto para el cual desea obtener las métricas.		
5. El usuario selecciona el proyecto para el cual desea obtener las métricas.		
6. El sistema busca los ciclos de prueba asociados al proyecto y solicita al usuario que seleccione un ciclo.		
7. El usuario selecciona el ciclo de prueba para el cual desea obtener las métricas.		
8. El sistema muestra para el ciclo de prueba seleccionado, la siguiente información en forma tabular y gráfica: <ul style="list-style-type: none"> • CPS: número de casos de prueba que han dado resultados satisfactorios sobre el total de pruebas definidas. • CPSR: número de casos de prueba que han dado resultados satisfactorios sobre el total de pruebas definidas para cada requerimiento. • CPF: número de casos de prueba que han dado resultados fallidos sobre el total de pruebas definidas. • CPFR: número de casos de prueba que han dado resultados fallidos sobre el total de pruebas definidas para cada requerimiento. 		8.A. El sistema muestra para el ciclo de prueba seleccionado, la siguiente información en forma tabular y gráfica: <ul style="list-style-type: none"> •CPS: número de casos de prueba que han dado resultados satisfactorios sobre el total de pruebas definidas. •CPSR: número de casos de prueba que han dado resultados satisfactorios sobre el total de pruebas definidas para cada requerimiento. •CPF: número de casos de prueba que han dado resultados fallidos sobre el total de pruebas definidas. •CPFR: número de casos de prueba que han dado resultados fallidos sobre el total de pruebas definidas para cada requerimiento. 8.A.1. El usuario presiona Imprimir 8.A.2. El sistema muestra la vista preliminar de impresión del Reporte.
9. Fin del Caso de Uso		
Observaciones:		
Relaciones entre CU:		



Asociación de Inclusión: no aplica
Asociación de Extensión: no aplica
Caso de uso donde se incluye: no aplica
Caso de uso al que extiende: no aplica

CASO DE USO: 030. GENERAR REPORTE PERFILES DE FALLOS

Descripción

Nombre del Caso de Uso: Generar Reporte de perfiles de fallos		Número: 030
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Generar reporte de perfiles de las fallas registradas para un proyecto		
Pre-condiciones: Usuario logueado con permisos para llevar a cabo esta acción.		
Post-Condicion:		Éxito 1: Reporte generado exitosamente.
		Fracaso :

Curso Normal	Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Métricas”.	
2. El sistema despliega un menú para seleccionar la métrica deseada.	
3. El usuario selecciona la opción “Perfiles de Fallos”.	
4. El sistema solicita que el usuario seleccione el proyecto para el cual desea obtener las métricas.	
5. El usuario selecciona el proyecto para el cual desea obtener las métricas.	5.A. El usuario selecciona el proyecto para el cual desea obtener las métricas y selecciona el o los ciclos de prueba de los cuales desea visualizar información.
6. El sistema muestra para el proyecto seleccionado, la siguiente información en forma tabular y gráfica: <ul style="list-style-type: none"> • %DT: porcentaje de defectos por tipo de severidad. • NDT: es el número de defectos por tipo de severidad. 	6.A. El sistema muestra para el ciclo de prueba seleccionado, la siguiente información en forma tabular y gráfica: <ul style="list-style-type: none"> • %DT: porcentaje de defectos por tipo de severidad. • NDT: es el número de defectos por tipo de severidad. 6.A.1. El usuario presiona Imprimir 6.A.2. El sistema muestra la vista preliminar de impresión del Reporte.
7. Fin del Caso de Uso	

Observaciones:

Relaciones entre CU:

Asociación de Inclusión: no aplica
Asociación de Extensión: no aplica
Caso de uso donde se incluye: no aplica
Caso de uso al que extiende: no aplica



CASO DE USO: 031. GENERAR REPORTE DE DENSIDAD DE DEFECTOS
Descripción

Nombre del Caso de Uso: Generar Reporte de densidad de defectos		Número: 031
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Generar reporte grafico de densidad de defectos para un proyecto específico.		
Pre-condiciones: Usuario logueado con permisos para llevar a cabo esta acción.		
Post-Condicion:	Éxito 1: Reporte generado exitosamente.	
	Fracaso:	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el Usuario ingresa al módulo “Métricas”.		
2. El sistema despliega un menú para seleccionar la métrica deseada.		
3. El usuario selecciona la opción “Densidad Defectos”.		
4. El sistema solicita que el usuario seleccione el proyecto para el cual desea obtener las métricas.		
5. El usuario selecciona el proyecto para el cual desea obtener las métricas.		
6. El sistema busca los ciclos de prueba asociados al proyecto y solicita al usuario que seleccione un ciclo.		
7. El usuario selecciona el ciclo de prueba para el cual desea obtener las métricas.		
8. El sistema muestra para el ciclo de prueba seleccionado, la siguiente información en forma gráfica: <ul style="list-style-type: none"> Cantidad de Incidencias por Estado y por ciclo. 		8.A. El sistema muestra para el ciclo de prueba seleccionado, la siguiente información en forma gráfica: <ul style="list-style-type: none"> Cantidad de Incidencias por Estado y por ciclo. 8.A.1. El usuario presiona Imprimir 8.A.2. El sistema muestra la vista preliminar de impresión del Reporte.
9. Fin del Caso de Uso		
Observaciones:		
Relaciones entre CU:		
Asociación de Inclusión: no aplica		
Asociación de Extensión: no aplica		
Caso de uso donde se incluye: no aplica		
Caso de uso al que extiende: no aplica		

CASO DE USO: 032. GENERAR REPORTE POR MÓDULOS
Descripción

Nombre del Caso de Uso: Generar Reporte por módulos		Número: 032
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Generar listado de los elementos registrados en el módulo seleccionado.		



Pre-condiciones: Usuario logueado con permisos para llevar a cabo esta acción.															
Post- Condiciones:	Éxito 1: Reporte generado exitosamente.														
	Fracaso:														
<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Curso Normal</th> <th style="width: 50%;">Alternativas</th> </tr> </thead> <tbody> <tr> <td>1. El caso de uso comienza cuando el Usuario ingresa a algún módulo de los siguientes dentro del sistema: <ul style="list-style-type: none"> • Proyectos • Requerimientos • Casos de Prueba • Ciclos de Prueba • Ejecuciones • Incidencias </td> <td></td> </tr> <tr> <td>2. El sistema muestra un listado de los elementos que contiene el modulo.</td> <td></td> </tr> <tr> <td>3. El usuario selecciona “Exportar a Excel”.</td> <td></td> </tr> <tr> <td>4. El sistema genera un archivo .xls con la información del módulo.</td> <td></td> </tr> <tr> <td>5. El usuario abre el archivo generado donde se visualizan todos los elementos registrados en la base de datos de la aplicación para el modulo seleccionado mostrando campos definidos según el modulo (Obs.1).</td> <td></td> </tr> <tr> <td>6. Fin del Caso de Uso</td> <td></td> </tr> </tbody> </table>		Curso Normal	Alternativas	1. El caso de uso comienza cuando el Usuario ingresa a algún módulo de los siguientes dentro del sistema: <ul style="list-style-type: none"> • Proyectos • Requerimientos • Casos de Prueba • Ciclos de Prueba • Ejecuciones • Incidencias 		2. El sistema muestra un listado de los elementos que contiene el modulo.		3. El usuario selecciona “Exportar a Excel”.		4. El sistema genera un archivo .xls con la información del módulo.		5. El usuario abre el archivo generado donde se visualizan todos los elementos registrados en la base de datos de la aplicación para el modulo seleccionado mostrando campos definidos según el modulo (Obs.1).		6. Fin del Caso de Uso	
Curso Normal	Alternativas														
1. El caso de uso comienza cuando el Usuario ingresa a algún módulo de los siguientes dentro del sistema: <ul style="list-style-type: none"> • Proyectos • Requerimientos • Casos de Prueba • Ciclos de Prueba • Ejecuciones • Incidencias 															
2. El sistema muestra un listado de los elementos que contiene el modulo.															
3. El usuario selecciona “Exportar a Excel”.															
4. El sistema genera un archivo .xls con la información del módulo.															
5. El usuario abre el archivo generado donde se visualizan todos los elementos registrados en la base de datos de la aplicación para el modulo seleccionado mostrando campos definidos según el modulo (Obs.1).															
6. Fin del Caso de Uso															
Observaciones:															
Obs.1: Los campos que se visualizan dentro del archivo según el modulo son:															
<ul style="list-style-type: none"> • Modulo Proyectos: Nombre del Proyecto, Descripción, Fecha de Inicio, Fecha de Fin, Observaciones, Estado • Modulo Requerimientos: Clave, Nombre del Requerimiento, Tipo de Requerimiento, Objetivo, Proyecto • Modulo Casos de Prueba: Clave, Nombre del Caso, Requerimiento, Proyecto, Objetivo, Precondiciones, Pasos Ejecución, Resultado Esperado • Modulo Ciclos de Prueba: Nombre, Proyecto, Fecha Inicio, Fecha Fin, Ambiente de Pruebas, Observaciones • Modulo Ejecuciones: Ciclo De Prueba, Proyecto, Fecha Inicio, Fecha Fin, Porcentaje Ejecución • Modulo Incidencias: Clave, Nombre, Tipo, Severidad, Prioridad, Estado, Asignado, Proyecto 															
Relaciones entre CU:															
Asociación de Inclusión: no aplica															
Asociación de Extensión: no aplica															
Caso de uso donde se incluye: no aplica															
Caso de uso al que extiende: no aplica															

CASO DE USO: 033. REGISTRAR GRUPO DE USUARIOS.
Descripción

Nombre del Caso de Uso: Registrar grupo de usuarios		Número: 033	
Actor Principal: Usuario		Actor Secundario:	
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto			
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja			
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No	
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo			
Objetivo: Registrar los datos de un nuevo grupo de usuarios en la base de datos.			
Pre-condiciones: El Usuario debe estar logueado en la aplicación, con permisos para realizar la acción.			
Post- Condiciones:	Éxito 1: Grupo de Usuarios registrado exitosamente.		
	Fracaso :		
<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Curso Normal</th> <th style="width: 50%;">Alternativas</th> </tr> </thead> <tbody> </tbody> </table>		Curso Normal	Alternativas
Curso Normal	Alternativas		



1. El caso de uso comienza cuando el Usuario selecciona dentro del menú principal, la opción Grupos de Usuarios y luego “Agregar grupo”.	
2. El sistema muestra un formulario llamado "Nuevo Grupo de Usuarios".	
3. El usuario completa los siguientes campos solicitados obligatorios y presiona “Guardar y Salir”: <ul style="list-style-type: none"> • Nombre del grupo • Descripción 	3.A. El usuario no completa todos los campos requeridos. Presiona la opción “Guardar y Salir”. 3.A.1. El sistema informa la situación. 3.A.2. Continúa en el paso 3 del curso normal. 3.B. El usuario completa todos los datos de la pestaña Usuario y selecciona la pestaña “Permisos” 3.B.1. El sistema muestra un listado de los módulos del sistema con las posibilidades de acciones sobre cada uno de ellos. 3.B.2. El usuario selecciona las acciones que desea habilitar para el grupo y presiona “Guardar y Salir”. 3.C. El usuario presiona Cancelar 3.C.1. Se cancela el caso de uso.
4. El sistema registra el nuevo usuario en la base de datos.	
5. Fin del caso de uso.	
Observaciones:	
Relaciones entre CU:	
Asociación de Inclusión: no aplica	
Asociación de Extensión: no aplica	
Caso de uso donde se incluye: no aplica	
Caso de uso al que extiende: no aplica	

CASO DE USO: 034. ELIMINAR GRUPO DE USUARIOS.

Descripción

Nombre del Caso de Uso: Eliminar grupo de usuarios		Número: 034
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte		Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Eliminar un grupo de usuarios registrado la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post- Condiciones:	Éxito 1: Usuario eliminado exitosamente.	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el usuario selecciona del menú, la opción “Grupo de Usuarios”.		
2. El sistema muestra un filtro de búsqueda para encontrar el grupo de usuarios que desea eliminar.		
3. El usuario ingresa el filtro de búsqueda.		
4. El sistema devuelve los grupos de usuarios que coincidan con el filtro ingresado.		
5. El usuario presiona “Eliminar” sobre el grupo de usuarios que desea quitar de la base de datos.		



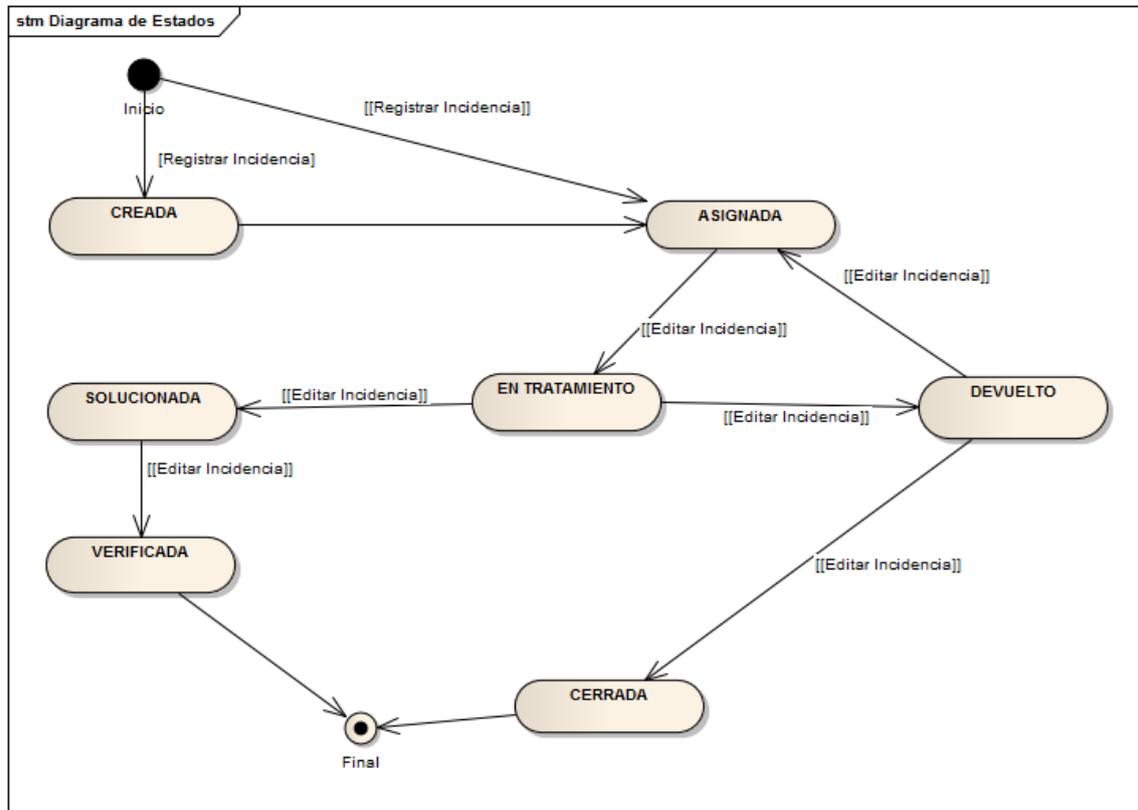
6. El sistema consulta si está seguro que desea eliminar el grupo.	
7. El usuario confirma la eliminación.	7.A. El usuario no confirma la eliminación 7.A.1. El sistema no elimina el grupo. 7.A.2. Fin del Caso de Uso.
8. El sistema elimina el grupo de usuarios de la base de datos.	
9. Fin del caso de uso.	
Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: no aplica Caso de uso donde se incluye: no aplica Caso de uso al que extiende: no aplica	

CASO DE USO: 035. MODIFICAR GRUPO DE USUARIOS.
Descripción

Nombre del Caso de Uso: Modificar grupo de usuario		Número: 035
Actor Principal: Usuario		Actor Secundario:
Tipo de Caso de Uso: <input checked="" type="checkbox"/> Concreto <input type="checkbox"/> Abstracto		
Prioridad: <input checked="" type="checkbox"/> Alta <input type="checkbox"/> Media <input type="checkbox"/> Baja		
Categoría: <input checked="" type="checkbox"/> Esencial <input type="checkbox"/> Soporte Significativo para la Arquitectura: <input checked="" type="checkbox"/> Si <input type="checkbox"/> No		
Complejidad: <input checked="" type="checkbox"/> Simple <input type="checkbox"/> Mediano <input type="checkbox"/> Complejo <input type="checkbox"/> Muy Complejo <input type="checkbox"/> Extremadamente Complejo		
Objetivo: Actualizar los datos de un grupo de usuarios registrado la base de datos.		
Pre-condiciones: El Usuario debe estar logueado en la aplicación con permisos para realizar la acción.		
Post-Condicion:	Éxito 1: Propiedades de grupo de usuarios modificadas exitosamente.	
	Fracaso :	
Curso Normal		Alternativas
1. El caso de uso comienza cuando el usuario selecciona del menú, la opción “ Grupo de Usuarios”.		
2. El sistema muestra un filtro de búsqueda para encontrar el usuario que desea editar.		
3. El usuario ingresa el filtro de búsqueda.		
4. El sistema devuelve los grupos de usuarios que coincidan con el filtro ingresado.		
5. El usuario presiona “Editar” sobre el grupo de usuarios que desea modificar.		
6. El sistema muestra las dos pestañas de propiedades del grupo de usuario en modo edición (Grupos de Usuario y Permisos).		
7. El usuario modifica los datos que desea actualizar del grupo de usuarios y presiona “Guardar y Salir”.		7.A. El usuario no completa todos los campos requeridos. Presiona la opción “Guardar y Salir”. 7.A.1. El sistema informa la situación. 7.A.2. Continúa en el paso 7 del curso normal.
8. El sistema actualiza la información del grupo de usuarios seleccionado exitosamente.		
9. Fin del caso de uso.		
Relaciones entre CU: Asociación de Inclusión: no aplica Asociación de Extensión: no aplica Caso de uso donde se incluye: no aplica		

Caso de uso al que extiende: no aplica

Diagrama de Estados



Arquitectura de Solución

El sistema será desarrollado en base a una arquitectura de 3 capas, capa de presentación, capa de negocio y capa de datos.

La arquitectura basada en capas se enfoca en la distribución de roles y responsabilidades de forma jerárquica proveyendo una forma efectiva de separación de responsabilidades.

El objetivo es obtener un sistema modular, desacoplable, reutilizable y capaz de escalar en el tiempo.

Punto de Vista de Sistemas

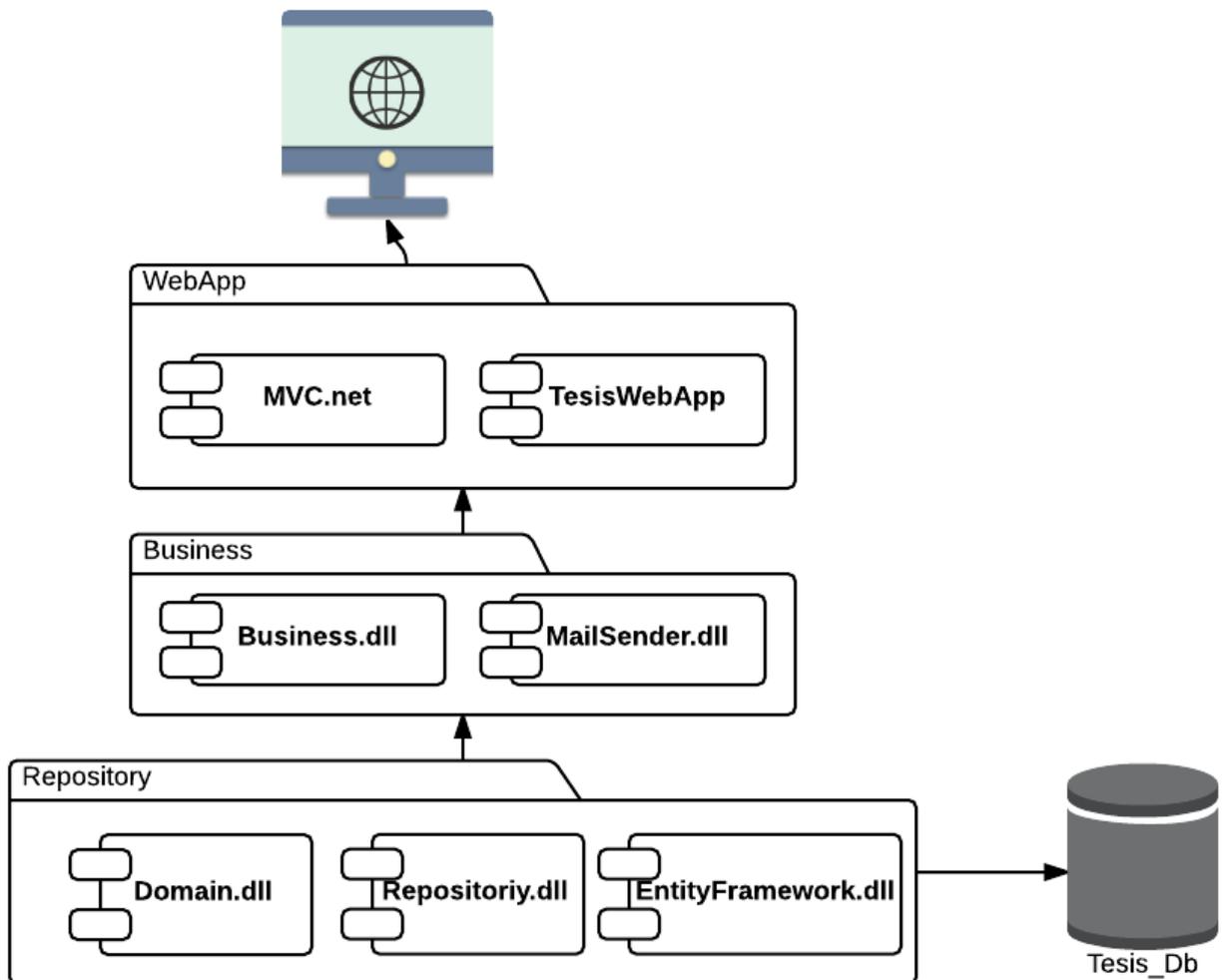
La aplicación estará desarrollada bajo una arquitectura escalable y flexible, que permita de manera intuitiva y lógica el agregado de nuevas funcionalidades y la detección clara de la localización de errores para una rápida corrección de los mismos.

Se busca lograr el mejor aprovechamiento de las tecnologías disponibles actualmente para obtener una experiencia de usuario satisfactoria al momento de usar las aplicaciones.

Se clarifica la arquitectura desde los siguientes puntos de vista:

PUNTO DE VISTA DE APLICACIÓN

El esquema arquitectónico se muestra a continuación:





La capa inferior de la arquitectura contendrá principalmente 3 módulos bien definidos:

- “Domain”: Donde se encontrará la definición de la estructura de las clases de negocio necesarias para el proyecto.
- “Repository”: Este módulo tendrá la responsabilidad de interactuar con la base de datos (A través del Entity Framework) para realizar las operaciones CRUD. (Alta, Modificación, Borrado y consulta de datos).
- “EntityFramework”: Es el ORM (Object – Relational Mapping) provisto por el framework .Net. Se encarga de interactuar directamente con la base de datos mapeando el modelo de tablas de la base de datos relacional a un modelo de objetos del Framework .Net.

La capa de negocios, contendrá 2 módulos:

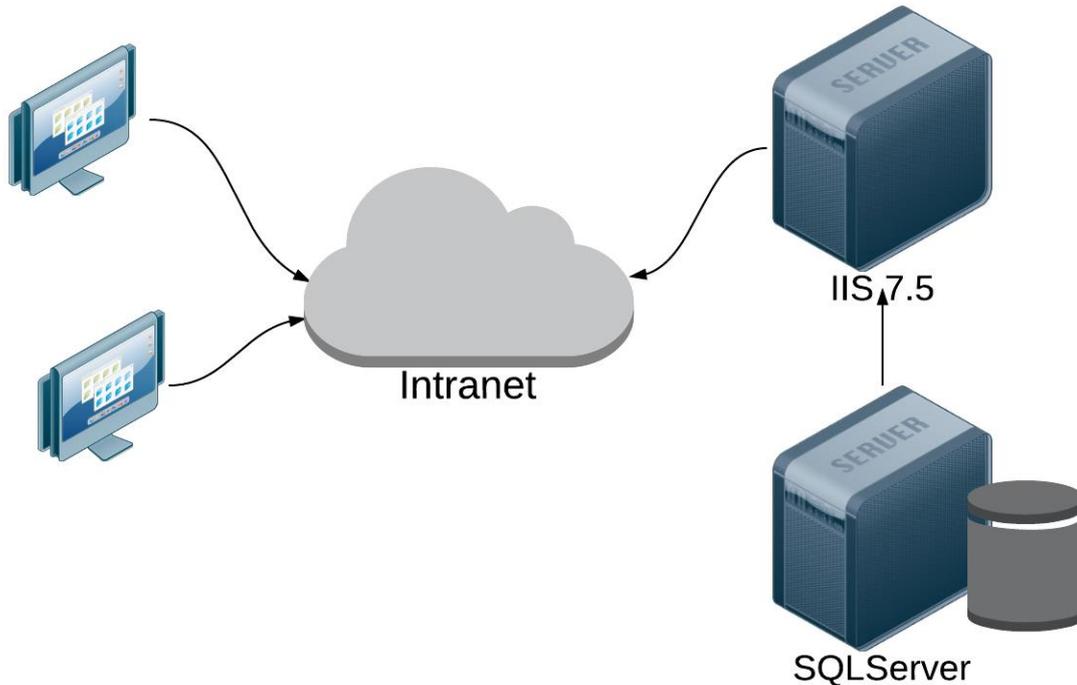
- **Business**: contendrá las reglas de negocio necesarias para poder procesar las respuestas obtenidas de las consultas a la base de datos. Todas las reglas de negocio necesarias para la realización de cualquier transacción se ubicarán en este módulo.
- **MailSender**: Este módulo tiene la responsabilidad de enviar las notificaciones generadas por la aplicación a los usuarios a través de mails.

TesisApp, será el front-end visible para los usuarios. Esta capa está compuesta básicamente por 2 módulos:

- **TesisWebApp**: La estructura de la aplicación web estará basada en el patrón de diseño MVC. Esta capa se encargará de la presentación de la información al usuario y de la interacción de los mismos con el sistema. En esta capa se realizará la validación primaria de la estructura de los datos que sean ingresados.
- **MVC.net** es el modulo del framework .net que da soporte al desarrollo de aplicaciones MVC.

PUNTO DE VISTA DE IMPLEMENTACIÓN:

Para la implementación final del producto se contempla la siguiente infraestructura:



Se deberán disponer de 2 servidores (físicos o virtuales):

- Servidor de base de datos: Este servidor será el encargado de correr el servicio de base de datos. Se deberá instalar el motor “MS SQL 2008 R2” y la base de datos “SGCS_DB”.
- Servidor de aplicaciones: Este servidor será el encargado de correr la aplicación web. Se deberá instalar el servidor de aplicaciones “Internet Information Server 7.5”.

El servidor de aplicaciones se conectará mediante el protocolo TCP/IP al servidor de aplicaciones para que la aplicación Web pueda disponer de estos datos.

El servidor deberá estar conectado a la intranet corporativa para poder brindar el servicio a todos los usuarios de la red.

PUNTO DE VISTA DE DATOS

A continuación, se presentan diagramas representativos de la estructura de datos para las aplicaciones del proyecto.

Usuarios	
Campo	Tipo Dato
IdUsuario	Numérico
Email	Texto
Password	Alfanumérico
Apellido	Texto



Nombre	Texto
--------	-------

FuncionesModuloGrupoUsuarios	
Campo	Tipo Dato
FuncionesModuloGrupoidUsuario	Numérico
GrupoUsuarioid	Numérico
Moduloid	Numérico
FuncionId	Numérico

Moduloes	
Campo	Tipo Dato
Moduloid	Numérico
NombreModulo	Texto
ClassLink	Alfanumérico
URL	Alfanumérico
Orden	Numérico
Modulo_Moduloid	Numérico

FuncionPorModulo	
Campo	Tipo Dato
Moduloid	Numérico
FuncionId	Numérico

Funcions	
Campo	Tipo Dato
FuncionId	Numérico
NombreFuncion	Texto

GrupoUsuarios	
Campo	Tipo Dato
GrupoUsuarioid	Numérico
NombreGrupo	Texto
Descripcion	Texto

UsuariosPorGrupoUsuario	
Campo	Tipo Dato
IdUsuario	Texto
GrupoUsuarioid	Numérico

IncidenciaSeveridad	
Campo	Tipo Dato
IncidenciaSeveridadId	Numérico
Nombre	Texto
Orden	Numérico



IncidenciaTipos	
Campo	Tipo Dato
IncidenciaTipold	Numérico
Nombre	Texto
Orden	Numérico

Incidencias	
Campo	Tipo Dato
Incidenciald	Numérico
Nombre	Texto
Adjuntold	Numérico
Estadold	Numérico
Asignadold	Numérico
Tipold	Numérico
Severidadld	Numérico
Prioridadld	Numérico
Determinacionld	Numérico
Descripcion	Alfanumérico
Clave	Alfanumérico
ReportadoPorld	Numérico
Proyectold	Numérico
CasoDePruebald	Numérico
Requerimientold	Numérico
CicloDePruebald	Numérico
FechaCreacion	Fecha

IncidenciaPrioridades	
Campo	Tipo Dato
IncidenciaPrioridadld	Numérico
Nombre	Texto
Orden	Numérico

CicloDePrueba	
Campo	Tipo Dato
CicloDePruebald	Numérico
Proyectold	Numérico
Nombre	Texto
Ambiente	Texto
FechaInicio	Fecha
FechaFin	Fecha
Observaciones	Alfanumérico



Archivos	
Campo	Tipo Dato
Archivold	Numérico
NombreArchivo	Texto
Path	Texto
Extension	Texto
FullPath	Texto

Determinaciones	
Campo	Tipo Dato
DeterminacionId	Numérico
CasoDePruebald	Numérico
CicloDePruebald	Numérico
Archivold	Numérico
Observaciones	Alfanumérico
Estadold	Numérico
EjecucionId	Numérico

Estados	
Campo	Tipo Dato
Estadold	Numérico
Descripción	Alfanumérico
TipoEstado	Texto

TransicionesDeEstado	
Campo	Tipo Dato
TransicionEstadold	Numérico
EstadoOrigenId	Numérico
EstadoDestinold	Numérico

HistorialDeEstado	
Campo	Tipo Dato
Id	Numérico
EstadoAnteriorId	Numérico
EstadoActualId	Numérico
FechaCambio	Fecha
UsuarioCambiald	Numérico
Incidenciald	Numérico

TiposRequerimientos	
Campo	Tipo Dato
TipoRequerimientold	Numérico
NombreTipo	Texto
Activo	Binario



CasoDePruebaPlantillas	
Campo	Tipo Dato
CasoDePruebaPlantillaId	Numérico
Objetivo	Texto
Precondiciones	Alfanumérico
ResultadoEsperado	Alfanumérico
PasosDeEjecucion	Alfanumérico
CategoriaId	Numérico
NombrePlantilla	Texto

CasoDePruebaCategorias	
Campo	Tipo Dato
CasoDePruebaCategoriald	Numérico
NombreCategoria	Texto

CasosCiclosDePrueba	
Campo	Tipo Dato
CicloDePruebald	Numérico
CasoDePruebald	Numérico

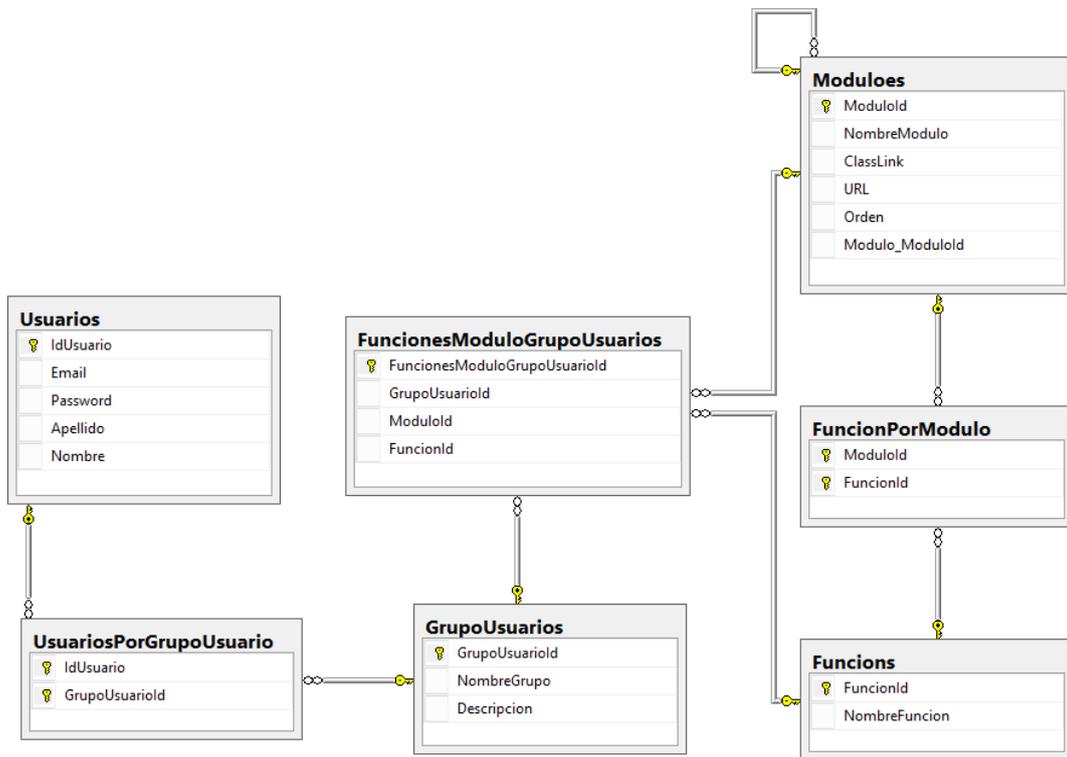
PUNTO DE VISTA TECNOLÓGICO

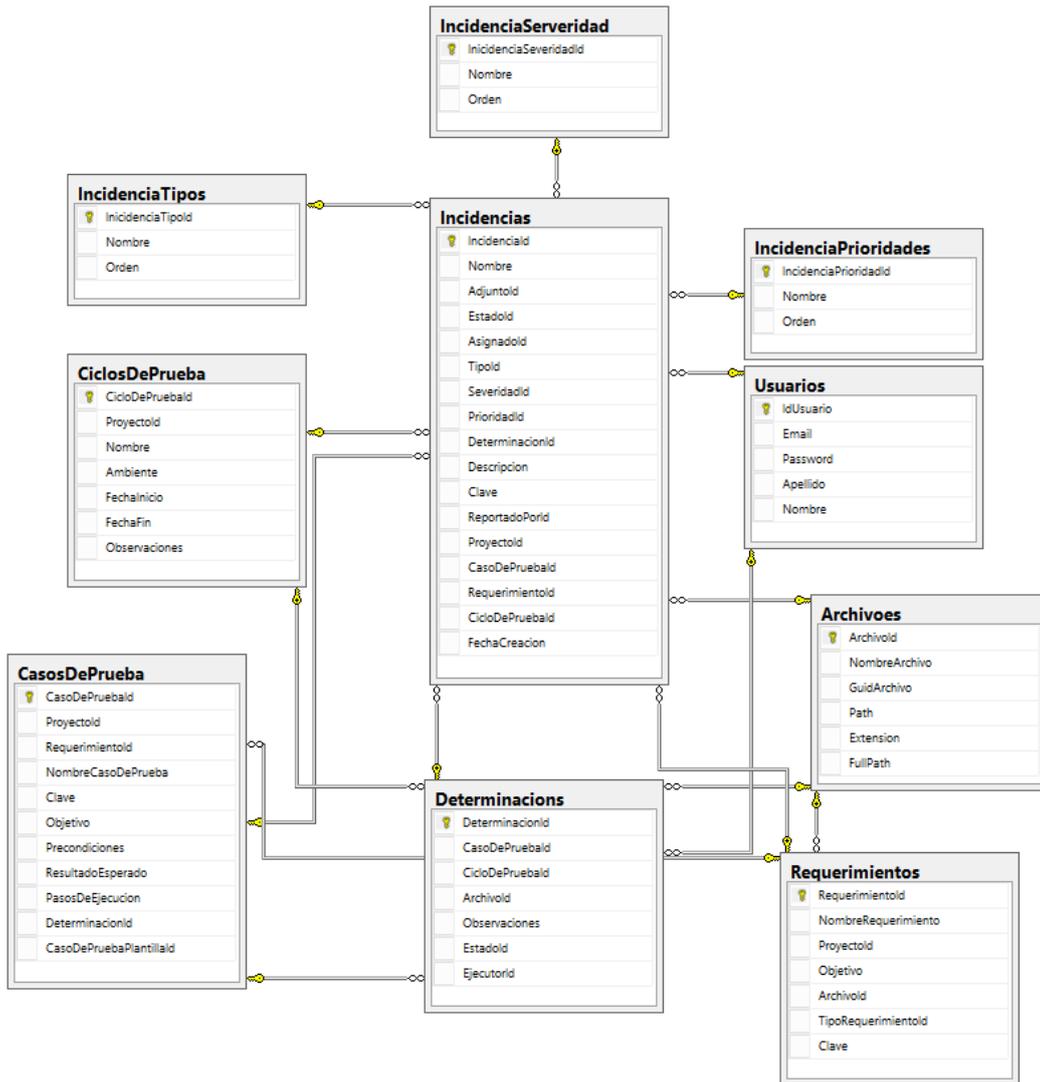
Para el desarrollo de la solución se hará uso de las siguientes tecnologías/herramientas:

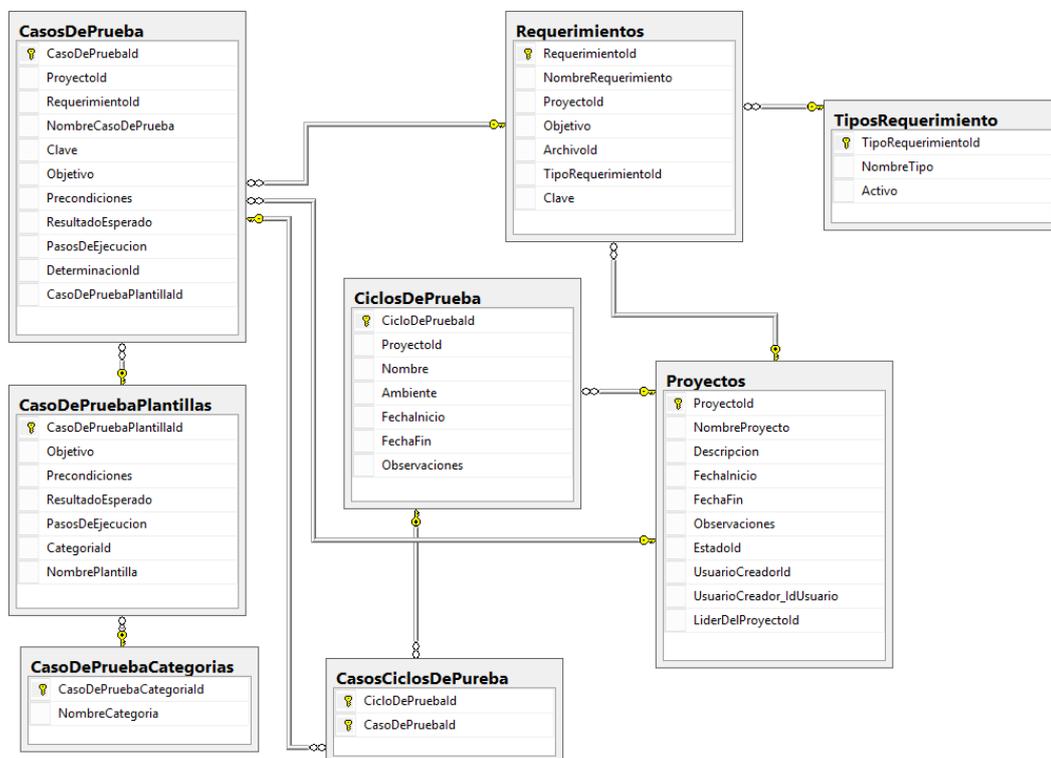
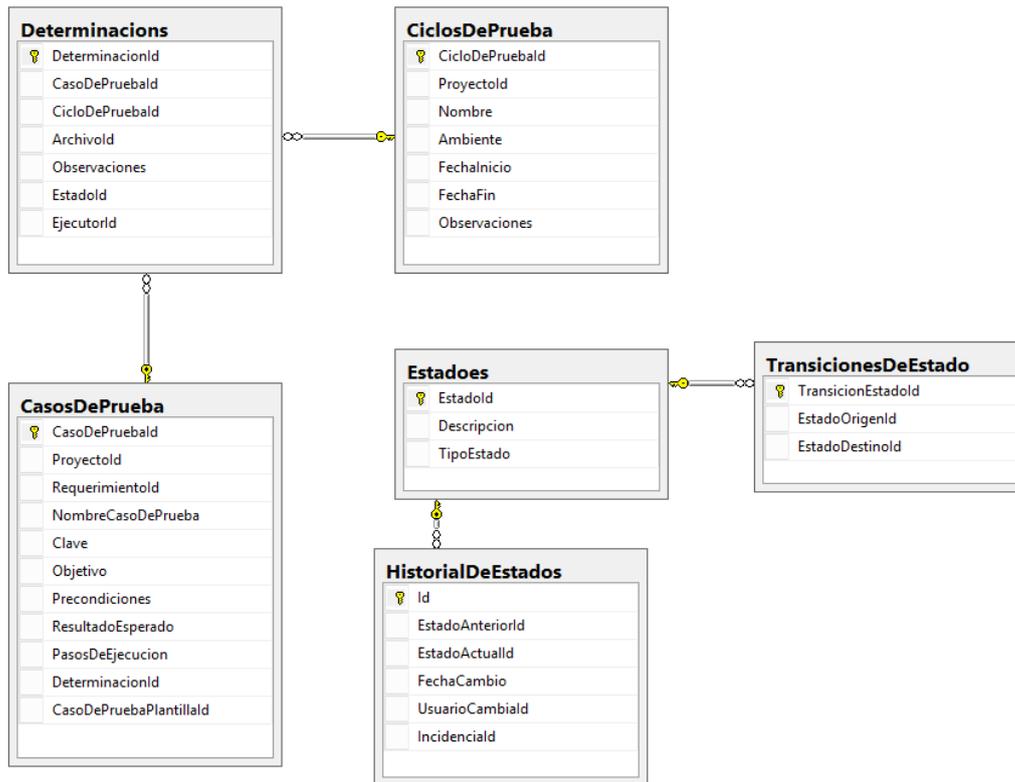
- .NET Framework 4.0 (en todas las capas de la aplicación).
- MVC 4.0 (para front-end de aplicaciones).
- Entity Framework 6.0.
- MS SQL Server 2008 R2.
- AngularJS (script para front-end).
- Bootstrap (hojas de estilo – responsive)



Diagrama Entidad – Relación









Módulo 6: Pruebas del Sistema

En el Módulo seis, PRUEBAS DEL SISTEMA, Se definen los Casos de prueba del sistema de información, basados en técnicas de diseño de pruebas mencionadas anteriormente.



La fase de pruebas es una de las más costosas del ciclo de vida software. En sentido estricto, deben realizarse pruebas de todos los artefactos generados durante la construcción de un producto software, lo que incluye las especificaciones de requisitos, casos de uso, diagramas de diversos tipos y, por supuesto, el código fuente y el resto de elementos como por ejemplo la base de datos.

El objetivo de esta etapa es verificar el resultado de la implementación, probando cada construcción desarrollada. Es necesario probar tanto construcciones intermedias, como las versiones finales del sistema que serán entregadas al cliente.

Para ello se debe planificar las pruebas requeridas en cada iteración, esto incluye tanto las pruebas de integración como las pruebas del sistema completo. El propósito de las pruebas de integración es determinar si las distintas unidades que hemos desarrollado por separado funcionan adecuadamente juntas. Luego se probará el sistema completo ejecutando casos de usos en paralelo y sometiéndolo a diferentes cargas.

A continuación, se deben diseñar e implementar las pruebas creando casos de prueba y los procedimientos de pruebas utilizando alguna de las técnicas que se mencionaron en el Modulo Nro. II "Marco Teórico".

El objetivo de realizar estas pruebas en nuestro sistema es encontrar los defectos en las construcciones y poder resolverlas lo antes posible.

DISEÑO DE CASOS DE PRUEBA

Las pruebas a realizar son funcionales. Se parte de los requerimientos funcionales de alto nivel y se diseñan pruebas que se aplican al sistema sin necesidad de conocer como está construido por dentro. Las pruebas se aplican sobre el sistema empleando un determinado conjunto de datos de entradas y observando las salidas que se producen, para determinar si la función se está desempeñando en forma correcta por el sistema bajo prueba.

Para lo cual utilizamos el método de prueba basado en **Grafos**, tal como lo definimos en la unidad nro. 2 "Marco Teórico".

Este método posee los siguientes beneficios:

- ✓ Nos ayuda a determinar las causas fundamentales de un problema o de calidad usando un enfoque estructurado.
- ✓ Utiliza un formato ordenado, fácil de leer el diagrama las relaciones de causa y efecto.



- ✓ Indica posibles causas de variación en un proceso.
- ✓ Identifica áreas, donde los datos deben recogerse para su posterior estudio.
- ✓ Se fomenta la participación del equipo y utiliza el conocimiento del equipo del proceso.
- ✓ Aumenta el conocimiento del proceso, ayudando a todos a aprender más acerca de los factores en el trabajo y cómo se relacionan.

Otra de las técnicas utilizadas para el diseño de las pruebas de nuestra aplicación fueron:

- **Análisis del valor límite:** Los casos de prueba que exploran condiciones límites tienen una rentabilidad mayor. Las condiciones límites son situaciones en los bordes, por arriba y por debajo de las clases de equivalencia para los valores de la entrada y de la salida.
- **Prueba Ad hoc:** En esta técnica las pruebas son derivadas confiando en la habilidad, intuición y experiencia con programas similares. Estas pruebas son útiles para identificar pruebas que no son fácilmente encontradas por técnicas más formales.
- **Conjetura de errores:** Se trata de enumerar una lista de errores y después escribir los casos de prueba basadas en esa lista. Otra forma es identificar los casos de prueba asociados a asunciones que el programador pudo haber hecho cuando leía la especificación. Los casos de prueba son diseñados pensando las faltas más probables de un programa dado.



Conclusiones

El Testing nos puede ayudar a medir la Calidad del Software. La calidad se define como el grado en el que un conjunto de características inherentes de un producto cumple con sus requisitos previamente definidos.

La Calidad se incorpora al software en todo el Proceso de Ingeniería. La aplicación apropiada de métodos y herramientas, las revisiones técnicas formales y efectivas, junto con una administración y una medición sólidas aportan la Calidad, que se confirma durante las Pruebas.

Cuando se pregunta, la gente usualmente piensa en el Testing en términos de ejecución del Software. El Testing es una actividad que durante muchos años se encontró menospreciada y académicamente dejada de lado. Hoy en día, no se concibe el Desarrollo de Software sin Testing. Involucra como mínimo entre el 30 y el 50% de los recursos asignados a los Proyectos de Desarrollo de Software y su importancia se encuentra en alza. Testing: El proceso que consiste en todas las actividades del ciclo de vida, tanto estáticas como dinámicas, concernientes a la planificación, preparación y evaluación de productos de software y productos de trabajo relacionados para determinar que satisfacen sus requerimientos, demostrar que concuerdan con su propósito y detectar defectos.

Las actividades de Testing emplean recursos altamente capacitados y con conocimientos específicos, que no son proporcionados por las carreras de Sistemas. El mercado actual demanda un perfil de profesional, que no está siendo tenido en cuenta: aquella persona que le interesa analizar las cosas, que tiene una naturaleza inquisitiva, observador, curioso, creativo y que le gusta “desarmar o romper cosas”.

El éxito en la producción de software se obtiene logrando hacerlo con calidad y demostrando el grado de ésta, calificando como buena. Esto sólo es posible con la implantación de un Sistema para el Aseguramiento de la Calidad del Software directamente relacionado con la política establecida para su elaboración y que esté en correspondencia con la definición internacional ISO de calidad, ampliamente aceptada, y por los estándares del grupo ISO 9000.

El presente trabajo basado en los conocimientos aprendidos durante la carrera, exhibió que existen métricas e indicadores que permiten evaluar y analizar distintos atributos y características de los productos y procesos, asegurando la calidad del software producido y aplicándolo a una herramienta de gestión de las pruebas “QUALITY TRACKING”.



Dicha herramienta permitió a la empresa Harriague y Asociados, obtener información de sus productos previo a una entrega a un cliente, a través de indicadores o métricas que permiten garantizar la calidad de los entregables y facilitar la toma de decisiones. Además, ofreció a la empresa la posibilidad de organizar todos sus proyectos, agilizar los tiempos de los recursos, simplificar el seguimiento de los defectos encontrados, y el resguardo de toda la información concerniente a los proyectos.

Bibliografía

- Guía de Estudios de Calidad de Software IUA
- El Proceso Unificado de Desarrollo de Software – Ivar Jacobson, Grady Booch, James Rumbaugh – ISBN: 84-7829-036-2 – Addison Wesley
- El Lenguaje Unificado de Modelado – Ivar Jacobson, Grady Booch, James Rumbaugh – ISBN: 8478290370 – Addison Wesley
- Material obtenido mediante el cursado de “Diplomatura Superior en Testing de Softare” dictado por la Universidad Tecnológica Nacional.
- Material obtenido de curso “Analista de Testing” dictado por CESSI (Cámara de Empresas de Software y servicios Informáticos) – Harriague + Asociados.