



CARRERA DE ANÁLISIS DE SISTEMAS

**ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL
SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE
SEGUROS DE LA EMPRESA ITSM ECUADOR, PERÍODO
OCTUBRE 2018 – MARZO 2019.**

**TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL
TÍTULO DE TECNÓLOGO EN ANÁLISIS DE SISTEMAS**

AUTOR: CACUANGO CATUCUAMBA JESSICA ESTEFANIA

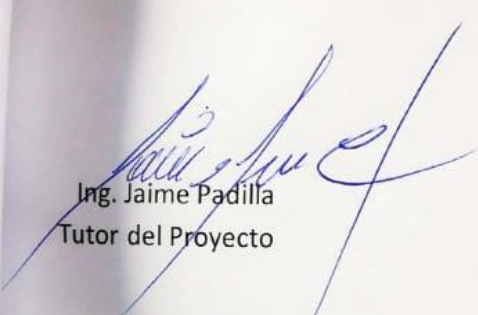
DIRECTOR: ING. PADILLA CEVALLOS JAIME MAURICIO

QUITO 2019


ACTA DE APROBACIÓN DEL TRABAJO DE TITULACIÓN

Quito, 20 de junio de 2019

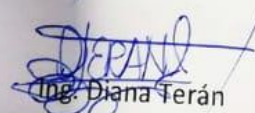
El equipo asesor del trabajo de Titulación conformado por el Ing. **JAIME MAURICIO PADILLA CEVALLOS** en calidad de tutor y por el Ing. **JOHNNY PATRICIO CORONEL ORDOÑEZ** en calidad de lector, de la Srta. **JESSICA ESTEFANÍA CACUANGO CATUCUAMBA** estudiante de la carrera de **ANÁLISIS DE SISTEMAS** cuyo tema de investigación fue: **"ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE SEGUROS DE LA EMPRESA ITSM ECUADOR, PERÍODO OCTUBRE 2018 - MARZO 2019"**. Una vez considerados los objetivos del estudio, coherencia entre los temas y metodologías desarrolladas; adecuación de la redacción, sintaxis, ortografía, puntuación con las normas vigentes sobre la presentación del escrito y el cronograma expuesto por la Unidad de Titulación, resuelve: **APROBAR** el trabajo de titulación, certificando que cumple con todos los requisitos exigidos por la institución.



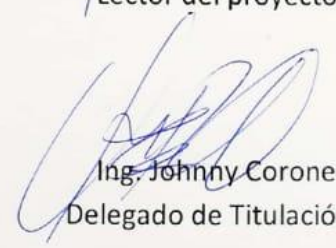
Ing. Jaime Padilla
Tutor del Proyecto



Ing. Johnny Coronel
Lector del proyecto



Ing. Diana Terán
Director de Carrera (E)



Ing. Johnny Coronel
Delegado de Titulación

CAMPUS 1 - MATRIZ

Calle N45-268 y Logroño
2255460 / 2269900
mailto:info@cordillera.edu.ec
www.cordillera.edu.ec
Quito - Ecuador

CAMPUS 2 - LOGROÑO

Calle Logroño Oe 2-84 y
Av. de la Prensa (esq.)
Edif. Cordillera
Telfs.: 2430443 / Fax: 2433649

CAMPUS 3 - BRACAMOROS

Bracamoros N15-163 y
Yacuambi (esq.)
Telf.: 2262041

CAMPUS 4 - BRASIL

Av. Brasil N46-45 y
Zamora
Telf.: 2246036

CAMPUS 5 - YACUAMBI

Yacuambi Oe2-36 y
Bracamoros
Telf.: 2249994

DECLARACIÓN DE AUTORÍA

Yo, **Jessica Estefania Cacuang Catucuamba**, declaro bajo juramento que la investigación es absolutamente original, auténtica, es de mi autoría, que se han citado las fuentes correspondientes y que en su ejecución se respetaron las disposiciones legales que protegen los derechos de autor vigentes. Las ideas, doctrinas, resultados y conclusiones a los que he llegado son de mi absoluta responsabilidad.



Jessica Estefania Cacuang Catucuamba

C.C: 172509458-3

LICENCIA DE USO NO COMERCIAL

Yo, **Jessica Estefania Cacuango Catucuamba** portador de la cédula de ciudadanía signada con el No. **172509458-3** de conformidad con lo establecido en el Artículo 110 del Código de Economía Social de los Conocimientos, la Creatividad y la Innovación (INGENIOS) que dice: “En el caso de las obras creadas en centros educativos, universidades, escuelas politécnicas, institutos superiores técnicos, tecnológicos, pedagógicos, de artes y los conservatorios superiores, e institutos públicos de investigación como resultado de su actividad académica o de investigación tales como trabajos de titulación, proyectos de investigación o innovación, artículos académicos, u otros análogos, sin perjuicio de que pueda existir relación de dependencia, la titularidad de los derechos patrimoniales corresponderá a los autores. Sin embargo, el establecimiento tendrá una licencia gratuita, intransferible y no exclusiva para el uso no comercial de la obra con fines académicos. Sin perjuicio de los derechos reconocidos en el párrafo precedente, el establecimiento podrá realizar un uso comercial de la obra previa autorización a los titulares y notificación a los autores en caso de que se traten de distintas personas. En cuyo caso corresponderá a los autores un porcentaje no inferior al cuarenta por ciento de los beneficios económicos resultantes de esta explotación. El mismo beneficio se aplicará a los autores que hayan transferido sus derechos a instituciones de educación superior o centros educativos.”, otorgo licencia gratuita, intransferible y no exclusiva para el uso no comercial del proyecto denominado **Análisis del proceso de control de calidad para el software gestión de ventas y operaciones de seguros de la empresa ITSM – Ecuador, período octubre 2018 – marzo 2019** con fines académicos al Instituto Tecnológico Superior Cordillera.



Jessica Estefania Cacuango Catucuamba

C.C: 172509458-3

Quito, 12 de junio del 2019

AGRADECIMIENTO

Al culminar mi proyecto de investigación el agradecimiento más grande es a Dios, por darme tanta fuerza, guiar mi camino y llenarme de muchas bendiciones. De la misma manera agradecer a mi familia que se mantuvo a mi lado en cada alegría tristeza siempre incondicionales con su apoyo, principalmente a mi madre que es la persona más fuerte y la que siempre ha permanecido a mi lado dándome su cariño y su motivación para seguir adelante cumpliendo cada uno de mis sueños.

A los docentes del Instituto Tecnológico Superior Cordillera que me brindaron sus conocimientos y siempre estuvieron puestos ayudarme con alguna inquietud durante los 3 años que estuve dentro de la institución.

A mis amigos que siempre me brindaron su apoyo antes y después de haber iniciado a estudiar mi carrera dentro de la institución.

A mi novio que siempre se mantiene a mi lado y juntos luchamos por cumplir nuestros sueños juntos.

DEDICATORIA

A Dios por guiar mi camino y darme una familia unida
A mi madre por nunca rendirse antes las adversidades de la vida
y siempre esta presta ayudarme en cada paso de mi vida
A mi hermano por ser al apoyo más grande
A mi docente y al mismo tiempo tutor de tesis que siempre
me apoyo incondicionalmente y no dudo de mi capacidad.

Contenido General

DECLARACIÓN DE AUTORÍA	i
LICENCIA DE USO NO COMERCIAL	iii
AGRADECIMIENTO	iv
DEDICATORIA	v
RESUMEN EJECUTIVO	xiv
ABSTRACT	xv
CAPÍTULO I	1
1. Antecedentes	1
1.01. Contexto	1
1.02. Justificación	4
1.03. Definición del problema central	4
1.03.1. Análisis de fuerzas “T”	4
CAPÍTULO II	6
2. Análisis de involucrados	6
2.01. Visión y alcance	6
2.02. Entrevistas	8
2.03. Matriz Requerimientos	9
2.03.1. Requerimientos Funcionales y No Funcionales	9
2.04. Descripción Detallada	18
2.05. Mapa de Involucrados	19
2.06. Matriz de análisis de involucrados	20
CAPÍTULO III	21
3. Problemas y objetivos	21
3.01. Árbol de Problemas	22
3.02. Árbol de objetivos	23
3.03. Casos de Uso	24
3.04. Especificación de casos de uso	25
3.05. Casos de uso de realización	27
3.06. Diagramas de secuencia	31
CAPÍTULO IV	34

4.	Análisis de alternativas.....	34
4.01.	Matriz de análisis de alternativas	34
4.02.	Matriz de análisis de impactos de los objetivos	35
4.03.	Diagramas de estrategias.....	36
4.03.1.	Diagrama de clase	37
4.03.2.	Modelo lógico y físico	38
4.04.	Matriz de marco lógico (MML)	40
4.05.	Vistas arquitectónicas	41
4.05.1.	Vista lógica.....	41
4.05.2.	Vista física.....	42
4.05.3.	Vista de desarrollo.....	43
4.05.4.	Vista de proceso	43
CAPÍTULO V		45
5.	Propuesta	45
5.01.	Antecedentes	45
5.02.	Descripción	46
5.03.	Formulación	46
5.04.	Especificación de estándares de programación.....	46
5.05.	Especificación de interfaz de usuario	48
5.06.	Especificación de pruebas de unidad	52
5.07.	Pruebas de aceptación	54
5.08.	Especificación de pruebas de carga	57
5.09.	Configuración del ambiente mínimo	59
CAPÍTULO VI		60
6.	Aspectos administrativos.....	60
6.01.	Recursos.....	60
6.01.1.	Recursos Humanos	60
6.01.2.	Recursos Materiales	60
6.01.3.	Recursos Tecnológicos.....	61
6.02.	Presupuesto	62
6.03.	Cronograma.....	62
CAPÍTULO VII		64

7. Conclusiones y recomendaciones.....	64
REFERENCIAS BIBLIOGRÁFICAS	65
ANEXOS	66

Índice de Tablas

Tabla 1	Análisis de fuerza	5
Tabla 2	Área de Administración.....	8
Tabla 3	Área de Reportes	9
Tabla 4	10
Tabla 5	Descripción detalla del requerimiento RF001	11
Tabla 6	Descripción detallada del requerimiento funcional RF002	12
Tabla 7	Descripción detallada del requerimiento funcional RF003	13
Tabla 8	Descripción detallada del requerimiento funcional RF004	14
Tabla 9	Descripción detallada del requerimiento funcional RF005	15
Tabla 10	Descripción detallada del requerimiento funcional RF006	16
Tabla 11	Descripción detallada del requerimiento funcional RF007	17
Tabla 12	Descripción detallada del requerimiento funcional RF008	18
Tabla 13	Matriz de involucrados	20
Tabla 14	Especificación de casos de uso (UC001), Ingreso del administrador	25
Tabla 15	Especificación de caso de uso (UC002), Proceso de ingreso a los test de pruebas.....	26
Tabla 16	Especificación de casos de uso (UC003), proceso del responsable de calidad	27
Tabla 17	Casos de uso de realización (CUR001) Ingreso de datos.....	28
Tabla 18	Casos de uso de realización (CUR002) Ingreso de datos a pruebas.....	29
Tabla 19	Casos de uso de realización (CUR003) ingreso de datos de proyectos.....	30
Tabla 20	Matriz de alternativas	34
Tabla 21	Matriz de análisis de impactos de los objetivos	35
Tabla 22	Matriz de marco lógico.....	40

Tabla 23	Tipos de componentes	47
Tabla 24	Tipos de controladores	47
Tabla 25	Prueba de unidad (PU001), ingreso al sistema	52
Tabla 26	Prueba de unidad (PU002), compilar código	52
Tabla 27	Prueba de unidad (PU003), generación de reportes	53
Tabla 28	Prueba de unidad (PU004) Almacenamiento en la base de datos	54
Tabla 29	Prueba de aceptación (PA001), ingreso de usuarios por cargo	55
Tabla 30	Prueba de Aceptación (PA002), crear reportes de pruebas y estados	55
Tabla 31	Prueba de unidad (PA003), ingreso de proyectos	56
Tabla 32	Prueba de Aceptación (PU004), detalle de carga de usuario.....	56
Tabla 33	Prueba de carga (PC001), ingreso de usuarios mediante cargos	57
Tabla 34	Prueba de carga (PC002), detalle de carga de varios usuarios	58
Tabla 35	Prueba de carga (PC003,)datos de pruebas de formularios	58
Tabla 36	Especificación de instalación de componentes.....	59
Tabla 37	Detalle de los recursos humanos	60
Tabla 38	Detalle de los recursos materiales	61
Tabla 39	Detalle de recursos tecnológico	61
Tabla 40	Detalle del presupuesto.....	62
Tabla 41	Requerimientos Funcionales	67

Índice de Figuras

Figura 1 Mapa de involucrados, participación directa e indirecta con el sistema	19
Figura 2 Detalle del árbol de problemas.....	22
Figura 3 Árbol de objetivos, diagrama para definir criterios de evaluación del software	23
Figura 4 Caso de uso (UC001), creación de usuario y asignación de rol.....	24
Figura 5 Caso de uso (UC002), proceso de revisión de pruebas.....	24
Figura 6 Caso de uso (UC003), proceso de registro y verificación de pruebas	25
Figura 7 Generación de ingreso de datos.....	27
Figura 8 Generación de pruebas	29
Figura 9 Generación de datos de proyectos.....	30
Figura 10 Diagrama de secuencia de ingreso de datos de usuario y proyectos.....	31
Figura 11 Diagrama de secuencia revisión de pruebas.....	32
Figura 12 Diagrama de secuencia de reportes e informes	33
Figura 13 Diagrama de estrategia.....	36
Figura 14 Detalle del diagrama de clase.....	37
Figura 15 Diagrama Lógico.....	38
Figura 16 Detalle del diagrama Físico.....	39
Figura 17 Vista lógica.....	41
Figura 18 Vista física.....	42
Figura 19 Vista de desarrollo.....	43
Figura 20 Vista de procesos funcionamiento de ingreso	43
Figura 21 Diagrama de procesos ingreso del programador al software	44
Figura 22 Diagrama de procesos funcionamiento del personal de control.....	44
Figura 23 Pantalla de ingreso al sistema	48

Figura 24 Pantalla de parámetros del menú.....	49
Figura 25 Pantalla de registro del usuario	49
Figura 26 Ingreso de Cargo	50
Figura 27 Menú de Test de pruebas.....	50
Figura 28 Test de prueba de Aceptación	51
Figura 29 Test de prueba de Validación.....	51
Figura 30 Test de prueba de Seguridad	52
Figura 31 Detalle de actividades	62
Figura 32 Cronograma de actividades especificados por sprint	63
Figura 33 Diagrama de Casos de Uso	66

Lista de Anexos

Anexo A Diagrama de Casos de Uso.....	66
Anexo B De la tabla 25. Tabla de Requerimientos.....	66
Anexo C Manual Instalación	69
Anexo D Manual de Usuario	84
Anexo E Manual Técnico	94

RESUMEN EJECUTIVO

La empresa ITSM- Ecuador ubicada en el distrito metropolitano de Quito, en la Av. Mariana de Jesús y Eloy Alfaro, dedicada al desarrollo de software se ve en la necesidad de implementar un sistema para evaluar mediante test de pruebas la calidad de sus sistemas antes de la entrega final al cliente.

La decisión se toma por parte del personal administrativo de la empresa debido a la necesidad de llevar un control automatizado para evitar pérdidas financieras debido a que el sistema no paso por un control de calidad adecuado y no cumple con las reglas del negocio.

Por lo tanto, la implementación de este sistema ayudaría a la empresa a contar con un sistema propio ajustado a las necesidades y cumpliendo todas las normas de calidad establecidas por las mismas.

ABSTRACT

The company ITSM- Ecuador located in the metropolitan district of Quito, on Av. Mariana de Jesús and Eloy Alfaro, dedicated to the development of software is seen in the need to implement a system to evaluate the quality of their systems through test tests. of the final delivery to the customer.

The decision is made by the administrative staff of the company due to the need to take an automated control to avoid financial losses because the system does not go through an adequate quality control and does not comply with the rules of the business.

Therefore, the implementation of this system would help the company to have its own system adjusted to the needs and meeting all the quality standards established by them.

CAPITULO I

1. Antecedentes

ITSM- ECUADOR empresa dedicada a la consultoría, capacitación y el desarrollo de software, constituida en el año 2013, con la misión de proporcionar servicios que faciliten a las empresas, el conocimiento que necesitan para optimizar la obtención de valor de sus tecnologías a través de la gestión, la capacitación y el desarrollo de software a la media. La empresa funciona en Quito, Ecuador y representa comercialmente la operación de Tecnofor Ibérica de España en nuestro país.

Como una de las líneas de negocio en desarrollo de software y, al capacitar en modelos de calidad de software, la empresa ha visto necesario tener un sistema que les permita administrar las pruebas de control de calidad de software y que permita fortalecer el producto que se entrega al cliente, de manera que cumpla con todos los elementos necesarios, normalizados esperados en relación a los conceptos de calidad. El principal objetivo es entregar un producto que haya pasado las marcas de aseguramiento de control de calidad y se ha visto necesario aplicar modelos basado en metodologías ágiles (SCRUM), ISO 25004, como prácticas de control de calidad de software.

1.01. Contexto

Según el autor (Deming, 2019) comenta en su libro que: “El control de calidad no significa alcanzar la perfección. Significa conseguir una eficiente producción con la calidad que espera obtener en el mercado”.

El control de calidad se relaciona con la inspección de los productos para detectar posibles errores, desde los años 80 se convirtió en un control estadístico que permite un análisis antes de que el producto llegue al usuario final, realizando gestiones, planificaciones, pruebas y creación de modelos de calidad de forma permanente de esta

manera optimizar los criterios de corrección y llevar una administración de pruebas realizadas al software. En el desarrollo de software para su entrega final debe contar con capacidades como:

Fiabilidad: Operar sin errores.

Modificable: Capacidad para realizar cambios necesarios.

Comprensible: Capacidad de comprender el funcionamiento de software, código para cambios requeridos.

Rendimiento: Velocidad del software.

Utilizable: Fácil manejo para los usuarios.

Probable: Fácil elaboración de casos de prueba.

Portable: Capacidad de movilizar el software de un lugar a otro.

Por lo cual el control de calidad se ha convertido en un objetivo estratégico para las organizaciones enfocadas al desarrollo de software, debido a que en la actualidad procedimientos que se realizaban manualmente en su mayoría se están automatizando, y las organizaciones dedicadas al desarrollo requieren entregar un producto final sin reprocesos.

ISO

Para industria del software existen normas relevantes como: La ISO 25004 proporciona requisitos, recomendaciones y guías que llevan a cabo el proceso de evaluación del software. Esta división se encuentra formada por:

ISO/IEC 25040 – “Evaluation reference model and guide: Propone un modelo de referencia general para la evaluación, que considera las entradas al proceso de evaluación, las restricciones y los recursos necesarios para obtener las correspondientes salidas” (García, 2017).

ISO/IEC 25041 – “Evaluation guide for developers, acquirers and independent evaluators: Describe los requisitos y recomendaciones para la implementación práctica de la evaluación del producto software desde el punto de vista de los desarrolladores, de los adquirentes y de los evaluadores independientes” (García, 2017).

ISO/IEC 25042 – “Evaluation modules: Define lo que la Norma considera un módulo de evaluación y la documentación, estructura y contenido que se debe utilizar a la hora de definir uno de estos módulos” (García, 2017).

ISO/IEC 25045 – “Evaluation module for recoverability: Define un módulo para la evaluación de la sub característica” (García, 2017).

SCRUM

Metodología centrada en los resultados de las exigencias del cliente, para la gestión de proyectos de software scrum se encarga de planificar proyectos en pequeños bloques llamados sprints para su revisión y corrección respectiva. El objetivo principal es trabajar en equipo a partir de los sprints para controlar y planificar proyectos de gran volumen al ser planificado por semanas al finalizar cada iteración se realiza una revisión de la semana anterior y de esta manera planificar un nuevo sprint. (Dimes, 2016)

ITSM Ecuador, como parte de su experiencia en implementación de proyectos, utiliza la metodología Light Agile denominada SCRUM, misma que ha venido usando con muy buenos resultados. Scrum es un proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado del proyecto.

1.02. Justificación

Implementar el control de calidad ha sido una preocupación para muchas empresas que realizan desarrollo de software y también para las personas encargadas de elaborar, uno de los principales objetivos para la empresa y el desarrollador es producir un sistema de alta calidad, para lo cual deben emplear métodos, parametrizaciones, medidas técnicas que evalúen la calidad con objetividad dentro del proceso y que de esta manera nos dé como resultado un sistema funcional.

ITSM Ecuador Cia. Ltda., empresa dedicada al “desarrollo de software”, tiene la necesidad de evaluar la calidad de los productos desarrollados debido a los reprocesos e inconformidad presentada por algunos clientes. Al disponer un control de calidad representa un esfuerzo importante para la organización. Por tal razón requiere iniciar las actividades para mejorar los procesos críticos en la fábrica de desarrollo en cuanto a la calidad de software, con la finalidad de minimizar los reprocesos y riesgos relacionados en sus productos y mejorar el nivel de calidad entregado a sus clientes.

1.03. Definición del problema central

Mediante la matriz de fuerza vamos a ver el estado actual en el que se encuentra la empresa antes de implementar el software

1.03.1. Análisis de fuerzas “T”

En la siguiente matriz se describe el problema que existe en la actualidad y el resultado que obtendremos después de realizar la investigación. También conoceremos las fuerzas impulsadoras y bloqueadoras que intervienen en el proyecto, las mismas que son calificadas en escala que comprende siendo 1 lo mínimo y 5 lo máximo.

Tabla 1
Análisis de fuerza

Situación Empeorada	Situación Actual				Situación Mejorada
Generar más gastos por la corrección de reprocesos y crear una mala imagen de la empresa cuando el software de errores en producción	Reprocesos del software que generan costos adicionales				Entregar un software funcional que haya pasado todas las marcas de aseguramiento de control de calidad
Fuerzas Impulsadoras	I	PC	I	PC	Fuerzas bloqueadoras
Compromiso por la dirección	3	5	5	3	Desconocimiento de la función del cargo
Mejora de los procesos para minimizar el riesgo	3	4	5	2	Resistencia al cambio
Recursos económicos necesarios para asumir el cambio	3	5	5	3	Mala distribución de recursos
Formación y capacitación del personal interno	3	4	5	2	Falta de experiencia del personal sobre el control de calidad

Nota: Matriz de fuerza T que detalla fuerza bloqueadora que impide lograr el propósito, la fuerza impulsadora nos ayuda a cumplir con nuestro objetivo

CAPÍTULO II

2. Análisis de involucrados

El análisis de involucrados investiga las personas que van a trabajar directa o indirectamente con el sistema.

ITSM Ecuador cuenta un personal encargado de sus actividades como:

Programadores: Encargados de la elaboración de los proyectos.

Administrativos: El personal administrativo es fundamental para la realización de propuestas y levantamiento de requerimientos.

Cliente: Da acceso a toda la información que se necesita para elaborar el software.

Control de Calidad: Persona encargada de realizar pruebas de control de calidad para entregar un producto final de calidad

2.01. Visión y alcance

Visión

La visión de este aplicativo es administrar las pruebas de control de calidad, para llevar a cabo una bitácora de todas las pruebas y cambios que se han realizado mediante la elaboración del software.

Alcance

El software va contar con varios módulos:

Módulo de Seguridad

Se podrá acceder al sistema con el ingreso de un usuario y contraseña, permitirá el ingreso al menú dependiendo de los roles y permisos asignado a cada usuario.

Acceso al sistema para poder ingresar nuevas empresas, cargos, cabecera de pruebas.

Módulo de Administración

Se podrá acceder al sistema con el ingreso de un usuario y contraseña, permitirá el ingreso al menú dependiendo de los roles y permisos asignado a cada usuario.

Creación de usuarios.

Creación del perfil de cada proyecto con los responsables asignados

Asignación de Personas para cada proyecto creado.

Módulo de Pruebas

Registro de pruebas elaboradas

Verificación del estado de pruebas

Cambios de estados en las pruebas

Bloqueo de campos de texto para ingreso de observaciones por parte del programador

Módulo de Reporte

En este módulo se van a generar los reportes de las pruebas que se han realizado durante la elaboración del software.

2.02. Entrevistas

Tabla 2
Área de Administración

Identificador: 001		
Preguntas	Objetivos	Análisis Posterior
¿Qué tipo de formato se utiliza para realizar las pruebas de control ?	Realizar un formato específico de las pruebas de control de calidad.	Se cuenta con formatos en los cuales se encuentran todos los parámetros dependiendo de la prueba a realizar.
¿Cómo se sabe cuántas pruebas se realizaron durante la elaboración de un sistema?	Realizar un reporte de la cantidad de pruebas que se hizo durante la elaboración de un sistema.	Mediante los registros realizados en Excel se lleva un conteo de las pruebas que se realizan a los sistemas.
¿Quiénes son las personas encargadas de realizar el control de calidad?	Asignar roles a cada usuario y asignar permisos dentro del menú.	Actualmente los programadores también realizan pruebas de control de calidad.
¿Qué pasaría si las pruebas físicas se perdieran por circunstancias naturales?	Almacenamiento de las pruebas realizadas dentro de una base de datos.	Actualmente todas las pruebas se llevan de forma manual por lo cual si existe un fenómeno natural como incendio, inundaciones no se tendría un respaldo

Nota: Para el análisis de nuestro estudio se revisó el estado actual de la elaboración de las pruebas de control de calidad

Tabla 3
Área de Reportes

Identificador: 002

Preguntas	Objetivos	Análisis Posterior
¿Qué seguridad tienen al trabajar con archivos Excel?	Automatizar los formularios para el almacenamiento de los datos de cada sistema elaborado.	Se trabajan con archivos Excel para el almacenamiento de toda la información.
¿Cómo llevan a cabo el conteo de pruebas elaboradas?	Registro de las pruebas en la base de datos.	Actualmente el conteo se realiza mediante el número físico de las pruebas y la hoja de Excel.
¿Cómo determinar si una prueba fue aceptada o rechazada ?	Verificar el tipo de estado en el que se mantiene la prueba.	Actualmente se llevan los registros en Excel y trabajan por colores los estados.

Nota: Para el análisis de nuestro estudio se revisó el estado actual de la elaboración de las pruebas de control de calidad

2.03. Matriz Requerimientos

2.03.1. Requerimientos Funcionales y No Funcionales

La tabla de requerimientos nos permite describir un conjunto de entradas, comportamientos y salidas de datos que se detalla en el (Anexo C).

Tabla 4***Requerimiento No Funcionales***

ID	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios Involucrados
RNF001	El sistema debe estar activo las 24 horas del día	Usuario	Alta	Sistema	En revisión	Administrador Usuario
RNF002	El sistema se va instalar en un sistema operativo Windows 10	Administrador	Alta	Sistema	En revisión	Administrador
RNF003	El sistema debe operar adecuadamente con hasta 100 usuarios con sesiones concurrentes	Administrador Usuario Encargado del control	Alta	Sistema	En revisión	Administración Usuario Encargado del control
RNF004	Los permisos de accesos al sistema podrán ser cambiados solamente por el administrador	Administrador	Alta	Sistema	En revisión	Administrador

Nota: Especificación de los Requerimientos No Funcionales

Tabla 5
Descripción detalla del requerimiento RF001

Descripción del requerimiento		Estado		Análisis	
Creado por	Jessica Cacuango	Actualizado	Jessica Cacuango		
Fecha de creación	25-03-2019	Fecha actualización	26-03-2019		
Identificador	RF001				
Tipo de requerimiento	Critico	Tipo de requerimiento	Funcional		
Datos de entrada	Asignación de usuario				
Descripción	Los usuarios permitirán el ingreso de un cargo al momento de registrar al usuario				
Datos de salida	Asignación de un cargo				
Origen	Matriz de requerimientos				
Dirigido a	Usuarios				
Prioridad	Alta				
Requerimientos	Ninguno				
Especificación					
Precondiciones	1. El usuario debe registrarse en el sistema 2. El Administrador asignara el cargo a cada usuario				
Post condiciones	Según el rol asignado el usuario podrá acceder al menú				
Criterios de aceptación	El usuario registrado podrá acceder al sistema.				

Nota: Detalle asignado de los cargos

Tabla 6
Descripción detallada del requerimiento funcional RF002

Descripción del requerimiento		Estado		Análisis			
Creado por		Jessica Cacuango		Actualizado		Jessica Cacuango	
Fecha de creación		25-03-2019		Fecha actualización		26-03-2019	
Identificador		RF002					
Tipo de requerimiento		Critico		Tipo de requerimiento		Funcional	
Datos de entrada		Nombre, apellido, cédula, correo, teléfono, usuario, contraseña, estado, cargo, fecha de modificación, fecha de creación.					
Descripción		El sistema permitirá el ingreso de los datos para el registro en la base de datos.					
Datos de salida		Información de los datos del usuario					
Origen		Matriz de requerimientos					
Dirigido a		Usuarios					
Prioridad		Alta					
Requerimientos		Ninguno					
Especificación							
Precondiciones		1. El usuario debe estar registrado en la base de datos					
Post condiciones		La información que el usuario da acceso debe ser verídica					
Criterios de aceptación		El usuario dependiendo de su cargo tendrá acceso a las diferentes opciones del menú para poder ingresar los datos requeridos.					

Nota: Especificación detallada del requerimiento RF002, el sistema debe registrar los usuarios en la base de datos

Tabla 7**Descripción detallada del requerimiento funcional RF003**

Descripción del requerimiento		Estado		Análisis			
Creado por		Jessica Cacuango		Actualizado		Jessica Cacuango	
Fecha de creación		25-03-2019		Fecha actualización		26-03-2019	
Identificador		RF003					
Tipo	de	Critico		Tipo	de	Funcional	
requerimiento		requerimiento					
Datos de entrada		Descripción					
Descripción		El ingreso de los cargos será realizada por el administrador					
Datos de salida		Datos de los cargos que se manejan dentro del sistema					
Origen		Matriz de requerimientos					
Dirigido a		Administrador					
Prioridad		Alta					
Requerimientos		RF003					
Especificación							
Precondiciones		1. El administrador deberá registrar los cargos para asignar a cada usuario 2. El administrador deberá registrar los cargos únicos con los que se van a trabajar dentro del sistema					
Post condiciones		Datos verídicos ingresados					
Criterios de aceptación		El administrador al crear el cargo tendrá acceso para crear usuarios.					

Nota: Especificación detallada del requerimiento RF002, el sistema podrá registrar el cargo

Tabla 8**Descripción detallada del requerimiento funcional RF004**

Descripción del requerimiento		Estado		Análisis			
Creado por		Jessica Cacuango		Actualizado		Jessica Cacuango	
Fecha de creación		27-03-2019		Fecha actualización		28-03-2019	
Identificador		RF004					
Tipo de requerimiento		Critico		Tipo de requerimiento		Funcional	
Datos de entrada		Descripción, RUC, dirección, teléfono, estado, fecha de modificación, fecha de creación.					
Descripción		El sistema va permitir el ingreso de los datos de la empresa y acceder a la creación de proyectos					
Datos de salida		Información general de la empresa					
Origen		Administrador					
Dirigido a		Administrador					
Prioridad		Alta					
Requerimientos		RF004					
Especificación							
Precondiciones		1. La empresa debe estar registrada en la base de datos 2. Los datos deben ser ingresados con datos verídicos					
Post condiciones		Si ingresa correctamente podrá acceder a visualizar los datos ingresados					
Criterios de aceptación		El ingreso de la empresa va permitir el registro del proyecto					

Nota: Especificación detallada del requerimiento RF004, el sistema debe registrar empresa

Tabla 9**Descripción detallada del requerimiento funcional RF005**

Descripción del requerimiento		Estado	Análisis
Creado por	Jessica Cacuango	Actualizado	Jessica Cacuango
Fecha de creación	27-03-2019	Fecha actualización	28-03-2019
Identificador	RF005		
Tipo de requerimiento	Critico	Tipo de requerimiento	Funcional
Datos de entrada	Proyecto: Nombre, fecha inicio, fecha final, estado, fecha creación, fecha modificación. Asignar Personas: Usuario, proyecto, fecha de creación y modificación		
Descripción	El sistema permitirá el ingreso de los datos del proyecto para poder registrar la asignación de personas y proyectos		
Datos de salida	Información general del proyecto y la asignación de personas		
Origen	Matriz de requerimientos		
Dirigido a	Administrador		
Prioridad	Alta		
Requerimientos	RF005		
Especificación			
Precondiciones	1. Se solicita llenar todos los campos son requeridos 2. Los datos se van almacenar en la base de datos		
Post condiciones	Si ingresa correctamente podrá acceder a visualizar los datos ingresados		
Criterios de aceptación	El administrador al ingresar al sistema tendrá que registrar la asignación de personas		

Nota: Especificación detallada del requerimiento RF005, Registro del proyecto y la asignación de personas

Tabla 10**Descripción detallada del requerimiento funcional RF006**

Descripción del requerimiento		Estado	Análisis
Creado por	Jessica Cacuango	Actualizado	Jessica Cacuango
Fecha de creación	27-03-2019	Fecha actualización	28-03-2019
Identificador	RF006		
Tipo de requerimiento	Critico	Tipo de requerimiento	Funcional
Datos de entrada	Fecha, módulo, requerimiento, proyecto, persona, datos principales de cada test de pruebas. Propósito, procedimiento, imagen, resultado, errores, rechas, URL, observaciones, objetivo, resultado final, etapa, elementos, fecha de creación y modificación,		
Descripción	El sistema permitirá el registro de los datos de cada uno de los test (GUI, aceptación, validación, seguridad)		
Datos de salida	Información general de cada una de las pruebas		
Origen	Matriz de requerimientos		
Dirigido a	Control de calidad		
Prioridad	Alta		
Requerimientos	RF006		
Especificación			
Precondiciones	1. Se solicita llenar todos los campos son requeridos 2. Los datos se van almacenar en la base de datos		
Post condiciones	Se ingresa correctamente podrá acceder a visualizar los datos ingresados		
Criterios de aceptación	El encargado de control de calidad debe registrar todas las pruebas.		

Nota: Especificación detallada del requerimiento RF006, Registro de Test de pruebas

Tabla 11

Descripción detallada del requerimiento funcional RF007

Descripción del requerimiento		Estado		Análisis	
Creado por	Jessica Cacuango	Actualizado		Jessica Cacuango	
Fecha de creación	27-03-2019	Fecha actualización		28-03-2019	
Identificador	RF007				
Tipo de requerimiento	Critico	Tipo de requerimiento		de Funcional	
Datos de entrada	Observaciones				
Descripción	El programador va tener el acceso a todas las pruebas para verificar los estados en el caso de que la prueba este rechazada podrá ingresar una observación				
Datos de salida	Información de los estados de las pruebas				
Origen	Matriz de requerimientos				
Dirigido a	Programador				
Prioridad	Alta				
Requerimientos	RF007				
Especificación					
Precondiciones	1. Se solicita llenar el campo requerido 2. Los datos se van almacenar en la base de datos				
Post condiciones	El programador debe ingresar la observación				
Criterios de aceptación	El programador al ingresar al sistema debe registrar la observación				

Nota: Especificación detallada del requerimiento RF007, Registro de la observación de las pruebas

Tabla 12
Descripción detallada del requerimiento funcional RF008

Descripción del requerimiento		Estado	Análisis
Creado por	Jessica Cacuango	Actualizado	Jessica Cacuango
Fecha de creación	27-03-2019	Fecha actualización	28-03-2019
Identificador	RF008		
Tipo de requerimiento	Critico	Tipo de Funcional requerimiento	
Datos de entrada	Prueba, fecha, cantidad de pruebas, porcentaje de estados		
Descripción	El reporte va ayudar a la persona encargada ver el rendimiento del sistema.		
Datos de salida	Información con gráficos		
Origen	Matriz de requerimientos		
Dirigido a	Administrador		
Prioridad	Alta		
Requerimientos	RF008		
Especificación			
Precondiciones	1. El administrador deberá previamente registrar la empresa		
Post condiciones	El administrador podrá observar todas las personas ingresadas		
Criterios de aceptación	El administrador obtendrá el porcentaje de aceptación		

Nota: Especificación detallada del requerimiento RF008, Emisión de reporte

2.04.Descripción Detallada

En la empresa ITSM – Ecuador se elabora el sistema para administración de pruebas de control de calidad para mejorar su administración de pruebas durante la elaboración de software y tener control de todos los casos de pruebas elaborados para probar su funcionalidad antes de la entrega final al cliente.

El administrador es la persona encargada de generar el nuevo proyecto con las personas involucradas (Administrador, Programadores, Encargado del control).

El encargado del control es el encargado de ingresar la información en los formularios de pruebas de control de calidad del software, también realiza los reportes para enviar un informe al Administrador.

El programador trabaja bajos los parámetros indicados mediante los criterios de aceptación por parte del cliente.

2.05. Mapa de Involucrados

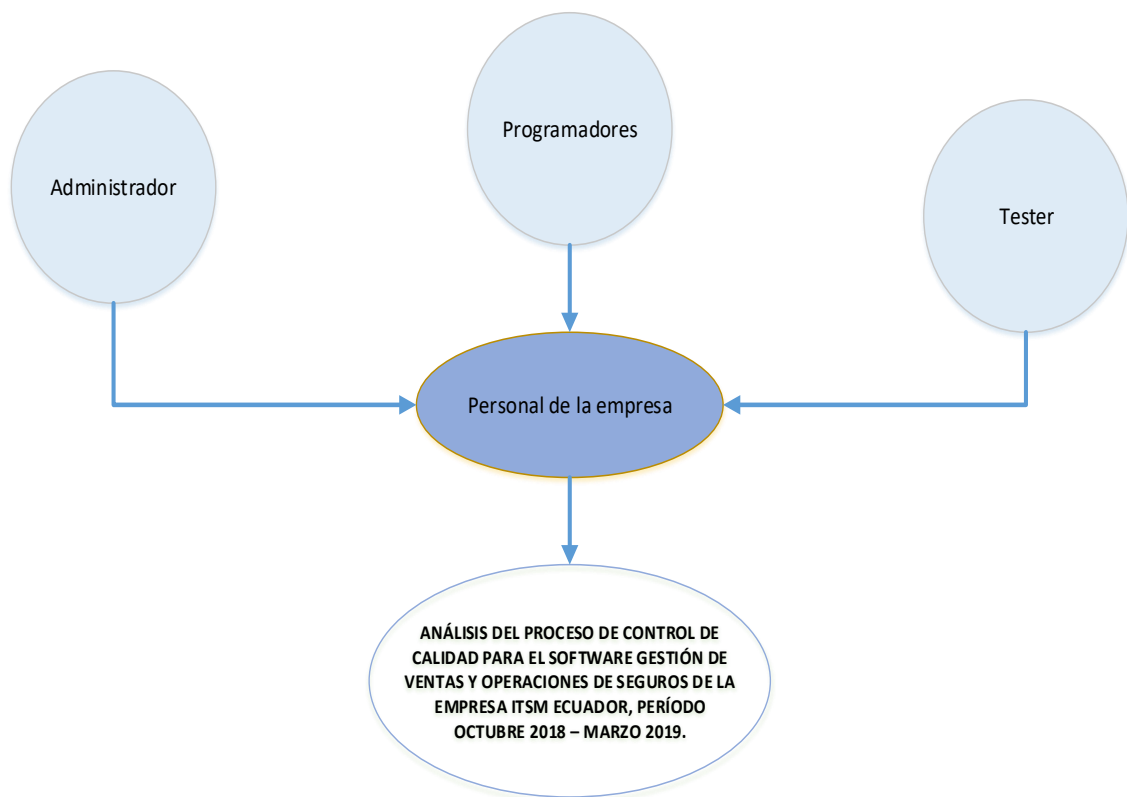


Figura 1 Mapa de involucrados, participación directa e indirecta con el sistema

2.06. Matriz de análisis de involucrados

Tabla 13
Matriz de involucrados

Actores Involucrados	Intereses sobre el problema central	Problemas Percibidos	Recursos mandatos y capacidades	Intereses sobre el proyecto	Conflictos potenciales
Administración	Entregar un software de calidad	Reprocesos en el software	Humano	Mejorar la calidad del software elaborado	Pérdidas financieras y de clientes
Desarrolladores de software	Corregir posibles errores en el software	No cumple con los requerimientos establecidos	Humano, Tecnológico	Implementar procesos estructurados para el desarrollo	Software de mala calidad
Encargado del control	Revisar el sistema elaborado y corregir todos sus errores	Mala revisión del software por parte del encargado del control	Humano	Automatizar proceso	Pérdida económica y devolución del sistema

Nota: Detalle de las personas involucradas con el sistema

CAPÍTULO III

3. Problemas y objetivos

Problema

El problema dentro de la empresa se debe por no contar con un proceso de control de calidad, que ayude a controlar, revisar, modificar posibles errores que se encuentren en base a los requerimientos levantados con el usuario y de esta manera generan pérdidas tanto en el tiempo como económicas dentro de la institución.

Por lo que se propone realizar un software de administración de pruebas de control de calidad, donde mediante el análisis y registro de información en los test de pruebas realizadas al sistema se ira corrigiendo posibles errores que podrían afectar durante su ejecución.

Objetivo

1. El objetivo principal es realizar revisiones técnicas durante el proceso de elaboración y de esta manera corregir problemas a tiempo, para que no afecten a módulos siguientes.
2. Guardar un registro de todas las pruebas realizadas al sistema elaborado
3. Realizar reportes de la cantidad de pruebas realizadas y emitir un informe al Administrador.

3.01.Árbol de Problemas

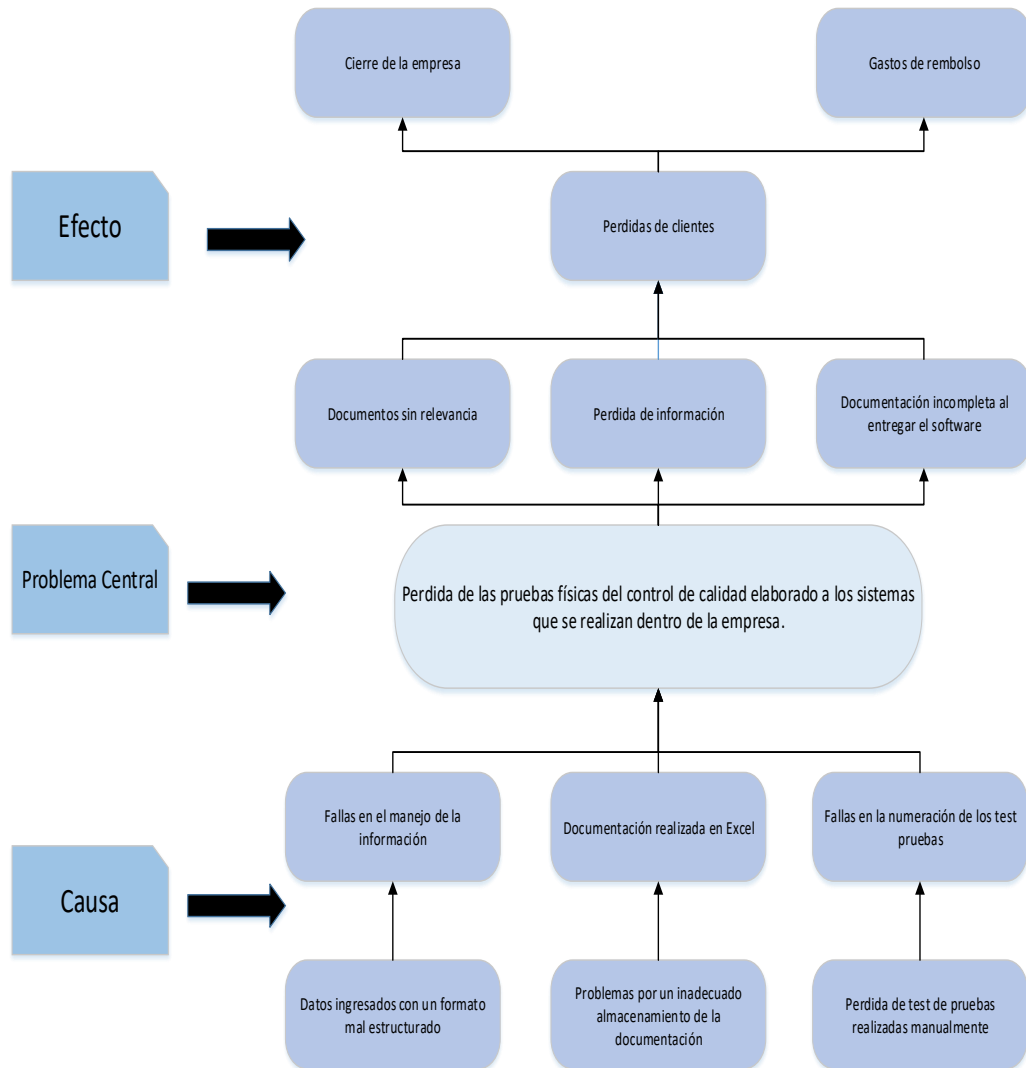


Figura 2 Detalle del árbol de problemas

3.02.Árbol de objetivos

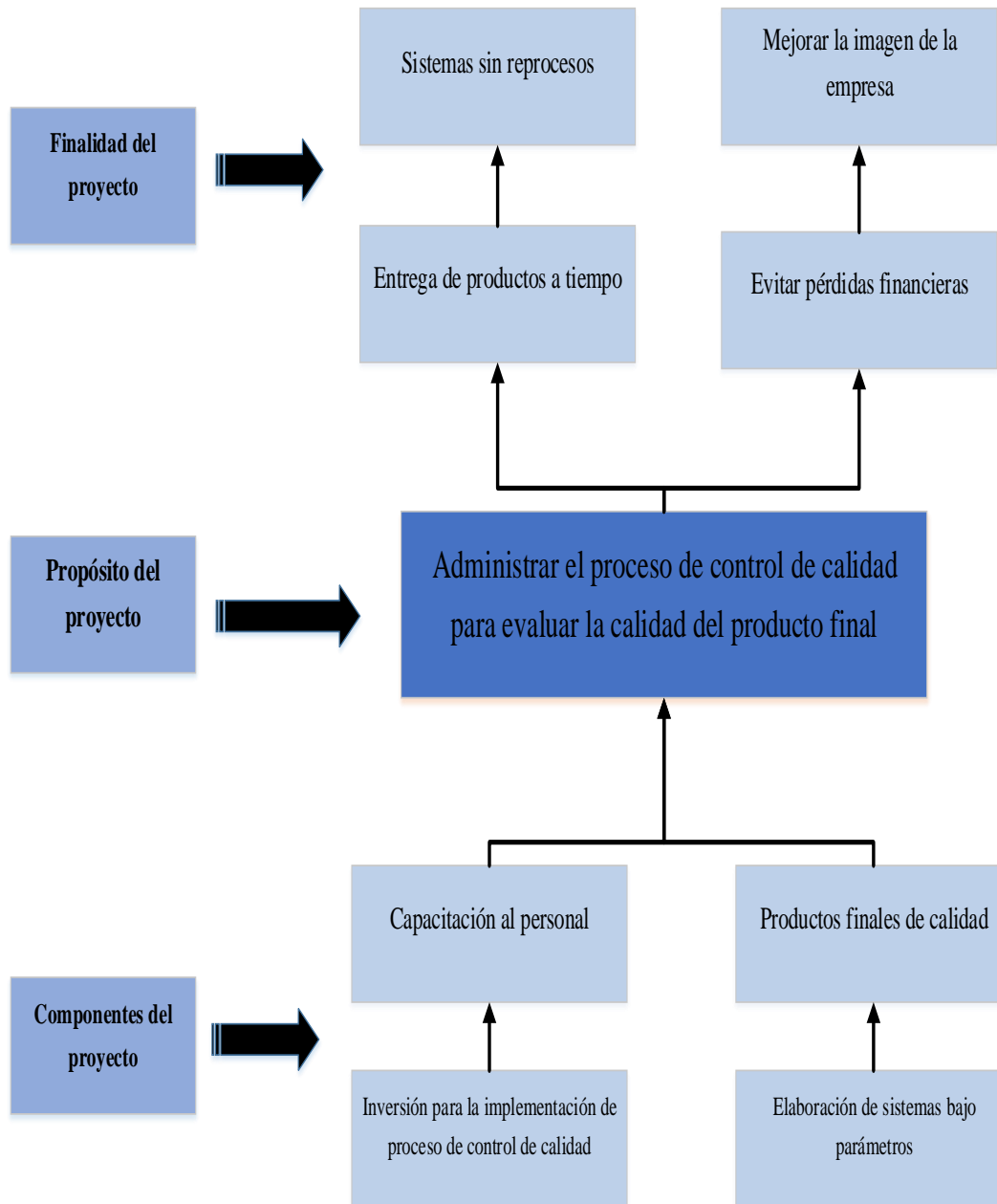


Figura 3 Árbol de objetivos, diagrama para definir criterios de evaluación del software

3.03.Casos de Uso

En el diagrama de casos de uso que se observa en el (Anexo A), describe el funcionamiento basado en el levantamiento de requerimientos, actores que intervienen en cada módulo. Se especifica las actividades que van a realizar cada uno de los actores dentro del software al momento de manipularlo.

Caso de uso UC001

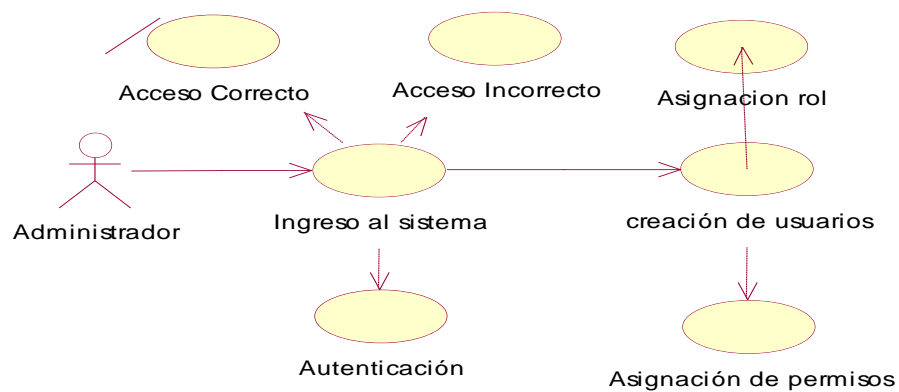


Figura 4 Caso de uso (UC001), creación de usuario y asignación de rol

Caso de uso UC002

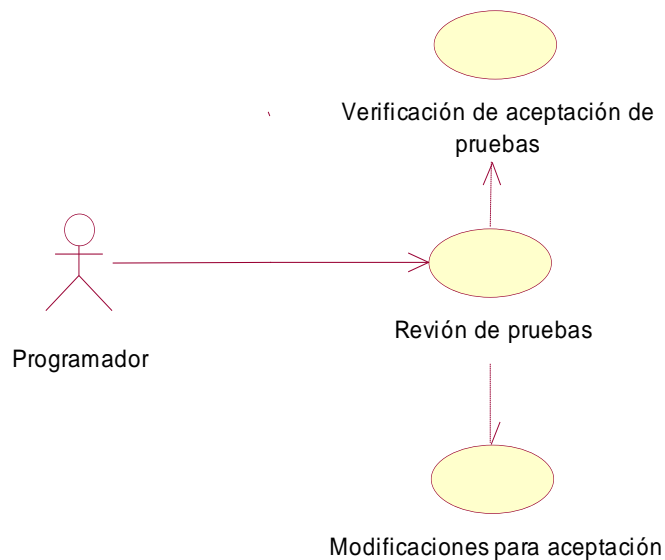


Figura 5 Caso de uso (UC002), proceso de revisión de pruebas

Caso de uso UC003

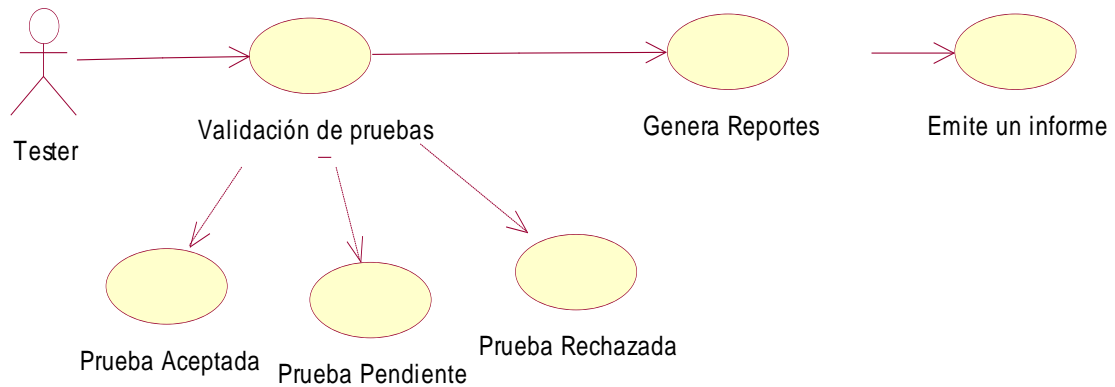


Figura 6 Caso de uso (UC003), proceso de registro y verificación de pruebas

3.04. Especificación de casos de uso

Tabla 14

Especificación de casos de uso (UC001), Ingreso del administrador

Casos de uso	Ingreso
Identificador	UC001
Curso típico de evento	
Usuario	Sistemas
Ingreso de operador	<p>El usuario ingresará a la ventana de acceso del sistema.</p> <p>El usuario ingresa el usuario y contraseña asignado para su verificación.</p>
Cursos alternativos	
	<ol style="list-style-type: none"> 1. Error de autenticación. 2. Falla en la conexión con la base de datos. 3. Datos erróneos solicitados.

Nota: Ingreso del Administrador

Tabla 15*Especificación de caso de uso (UC002), Proceso de ingreso a los test de pruebas*

Casos de uso	Ingreso
Identificador	UC002
Curso típico de evento	
Usuario	Sistemas
Ingreso de operador	<p>El usuario ingresara a la ventana de acceso del sistema.</p> <p>El usuario ingresa el usuario y contraseña asignado para su verificación.</p> <p>El usuario ingresa al módulo de pruebas dependiendo si es responsable de calidad genera test de pruebas.</p> <p>En el caso del ingreso de un usuario que sea programador se va encargar de revisar si la prueba fue rechazada</p>
Cursos alternativos	
<ol style="list-style-type: none">1. Error al ingresar al sistema2. Falla en la conexión con la base de datos.3. Pruebas sin registrar	

Nota: Proceso de verificación de pruebas

Tabla 16

Especificación de casos de uso (UC003), proceso del responsable de calidad

Casos de uso	Ingreso
Identificador	UC003
Curso típico de evento	
Usuario	Sistemas
Ingreso de operador	El usuario ingresa al módulo de pruebas, en el caso del responsable del control crea nuevos casos de pruebas para analizar.
Cursos alternativos	
1. Errores de estado de prueba 2. Pruebas sin registrar	

Nota: Ingreso del Responsable del Control de Calidad

3.05.Casos de uso de realización

CUR001

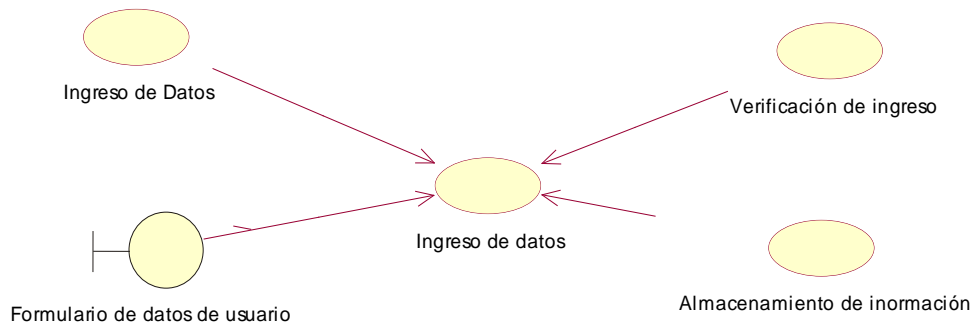


Figura 7 Generación de ingreso de datos

Tabla 17***Casos de uso de realización (CUR001) Ingreso de datos***

Nombre	Ingreso y creación
Identificación	CUR001
Responsabilidad del caso de uso	El administrador es el responsable del ingreso de la información verídica del usuario
Referencia del caso de uso	CU001
Referencia de requerimientos	Requisitos previos
Precondición	En el formulario de datos se va registrar la información del usuario, permitiendo que ingrese el usuario al sistema sin ningún inconveniente.
Relación	Relación directa con la base de datos
Post condiciones	Orden en el almacenamiento de información, anulación de ingreso de datos nulos.
Salidas de pantalla	Registro Guardado

Nota: Descripción del ingreso de datos

CUR002

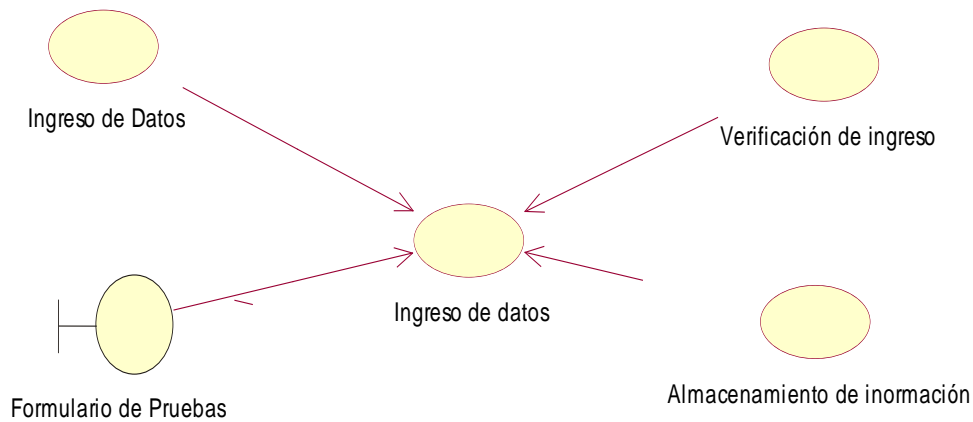


Figura 8 Generación de pruebas

Tabla 18

Casos de uso de realización (CUR002) Ingreso de datos a pruebas

Nombre	Ingreso y creación
Identificación	CUR002
Responsabilidad del caso de uso	El responsable de calidad es el encargado de llenar los datos de las pruebas
Referencia del caso de uso	CU002
Referencia de requerimientos	Requisitos previos
Precondición	El responsable de calidad ingresa datos verídicos mediante el control de calidad para dar un estado a la prueba
Relación	Relación directa con la base de datos
Post condiciones	Orden en el almacenamiento de información, anulación de ingreso de datos nulos.
Salidas de pantalla	Registro Guardado

Nota: Ingreso de datos en los test de pruebas

CR003

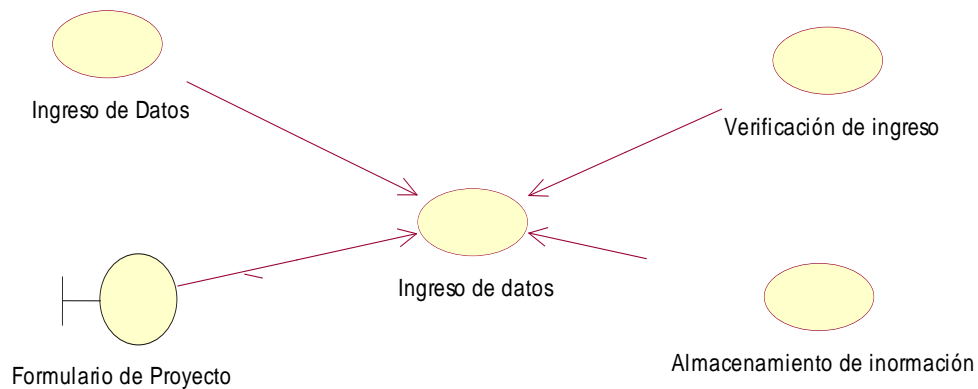


Figura 9 Generación de datos de proyectos

Tabla 19

Casos de uso de realización (CUR003) ingreso de datos de proyectos

Nombre	Ingreso y creación
Identificación	CR003
Responsabilidad del caso de uso	El Administrador es el encargado de llenar el formulario con los datos del proyecto para realizar las pruebas
Referencia del caso de uso	C003
Referencia de requerimientos	Requisitos previos
Precondición	El administrador se va encargar de llenar los campos con la información del proyecto y asignar los responsables
Relación	Relación directa con la base de datos
Post condiciones	Orden en el almacenamiento de información, anulación de ingreso de datos nulos.
Salidas de pantalla	Registro Guardado

Nota: Ingreso de datos del proyecto y asignación de personas

3.06. Diagramas de secuencia

Diagrama de secuencia de Ingreso, creación de usuario y proyecto.

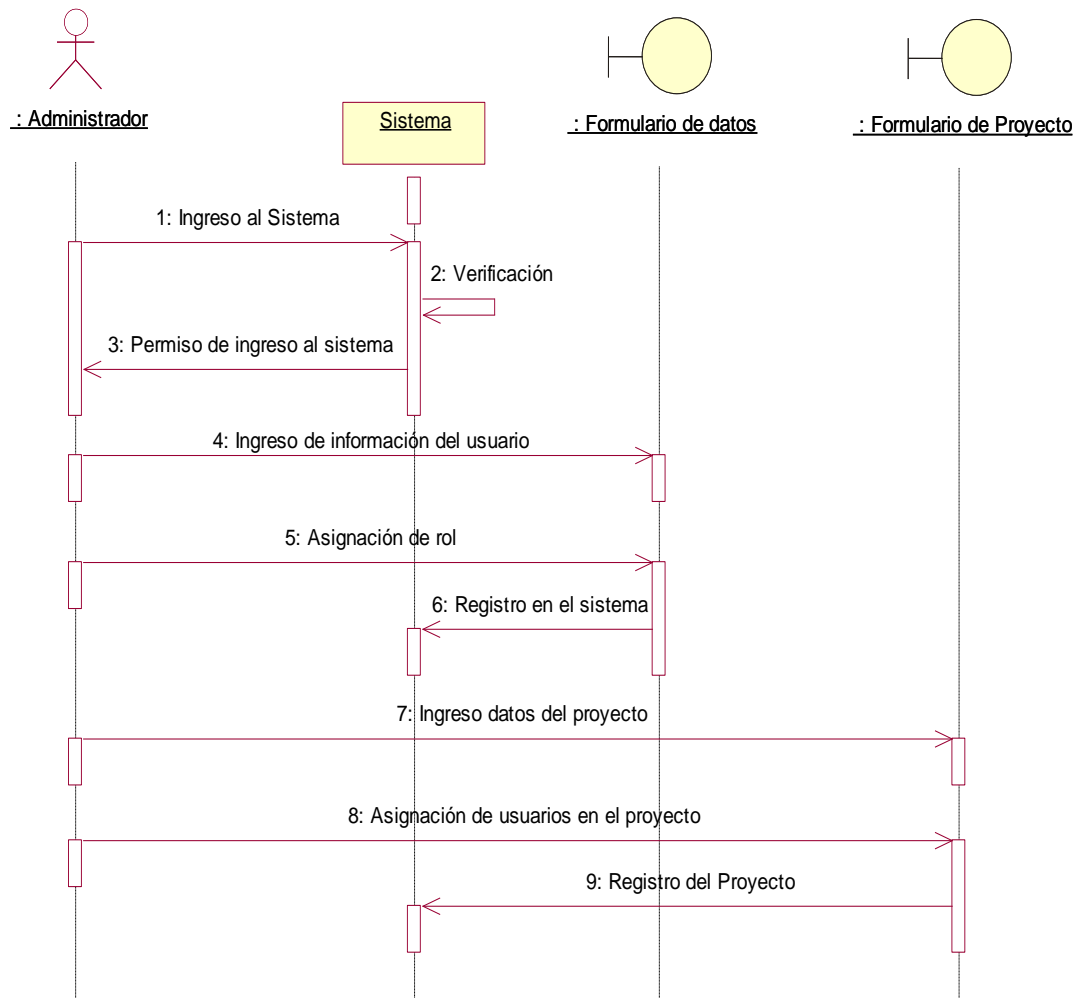


Figura 10 Diagrama de secuencia de ingreso de datos de usuario y proyectos

Diagrama de Secuencia para la revisión de pruebas por el encargado del control y el

programador

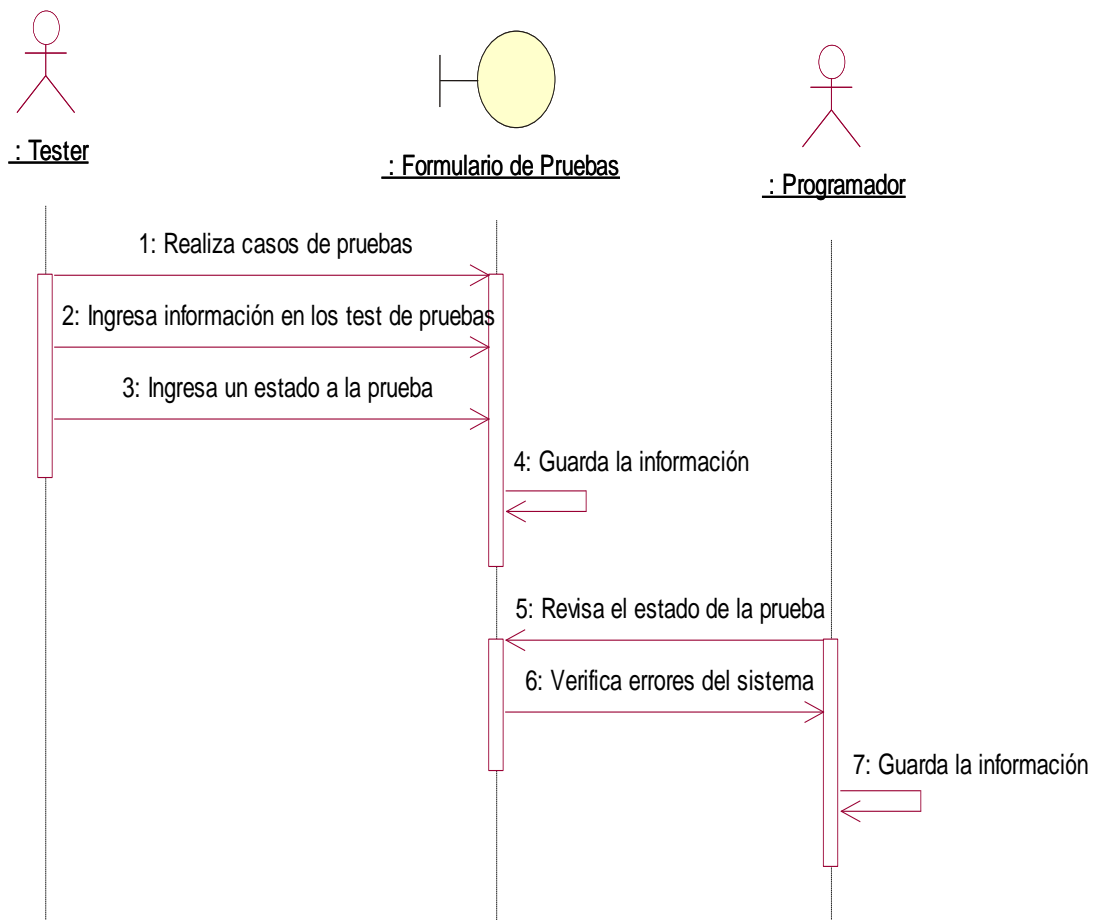


Figura 11 Diagrama de secuencia revisión de pruebas

Diagrama de secuencia de reportes e informes que será emitido al gerente para verificación de los avances del software

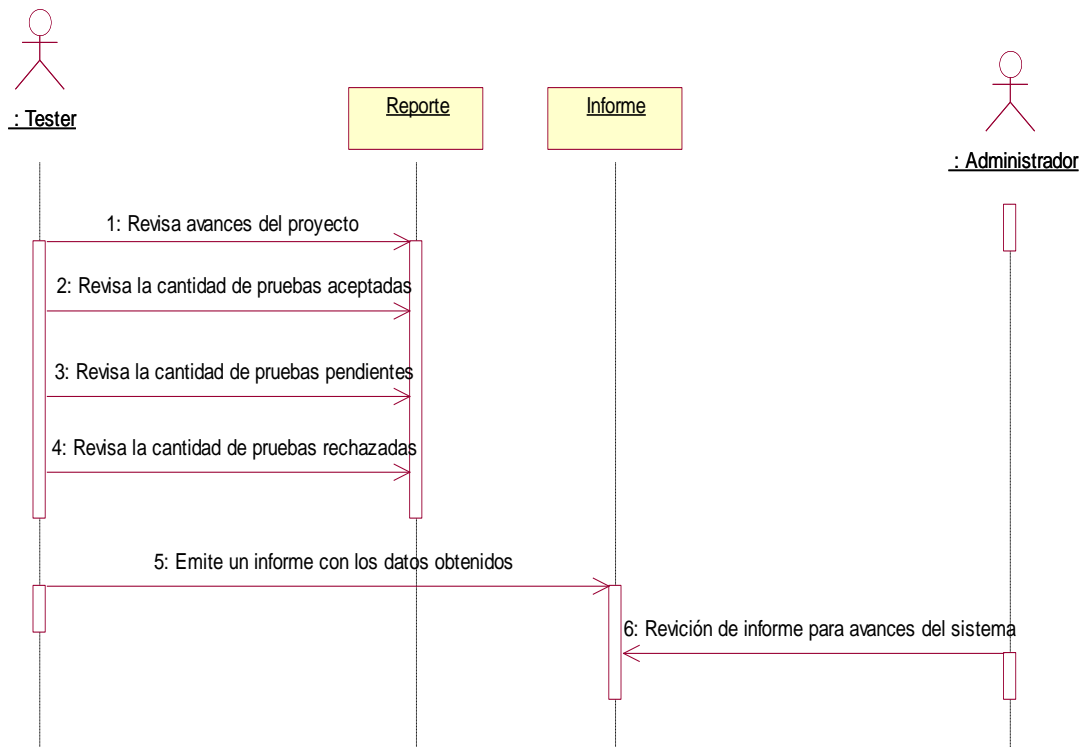


Figura 12 Diagrama de secuencia de reportes e informes

CAPÍTULO IV

4. Análisis de alternativas

4.01. Matriz de análisis de alternativas

Tabla 20

Matriz de alternativas

Objetivo	Impacto sobre el propósito	Factibilidad Técnica	Factibilidad financiera	Factibilidad Social	Factibilidad Política	Total	Categoría
Evitar reprocesos del sistema en su entrega final	4	5	5	2	2	18	Medio alta
Controlar el tiempo de trabajo definido por un cronograma	5	5	3	4	3	20	Medio alta
Establecer parámetros durante el desarrollo	4	5	5	3	2	19	Medio alta
Realizar revisiones técnicas para evitar posibles errores	5	4	4	3	3	18	Medio alta
Total	18	19	17	12	10	75	

Nota: Análisis de alternativas

4.02. Matriz de análisis de impactos de los objetivos

Tabla 21

Matriz de análisis de impactos de los objetivos

Objetivos	Factibilidad de lograrse	Impacto en genero	Impacto Ambiental	Relevancia	Sostenibilidad
Evitar reprocesos del sistema en su entrega final	Entregar sistemas sin ningún inconveniente	Capacitaciones del control de calidad para hombre y mujeres	Entregar sistemas en el cronograma establecido	Beneficia a la empresa y al cliente	Mejora la imagen de la empresa
Sistema regido en base al levantamiento de requerimientos	Sistema funcional	Facilidad de manejo para hombres y mujeres	Menos utilización de material de papelería	Satisfacción del cliente con el sistema	Proporcionar seguridad del sistema
Evitar pérdidas financieras y de tiempo	Organizando cronogramas de actividades	Organización y trabajo en equipo de hombres y mujeres	Gastos innecesarios	Ganancia para la empresa y el cliente	Mayor control dentro de la empresa

Nota: Análisis del impacto de objetivos

4.03. Diagramas de estrategias

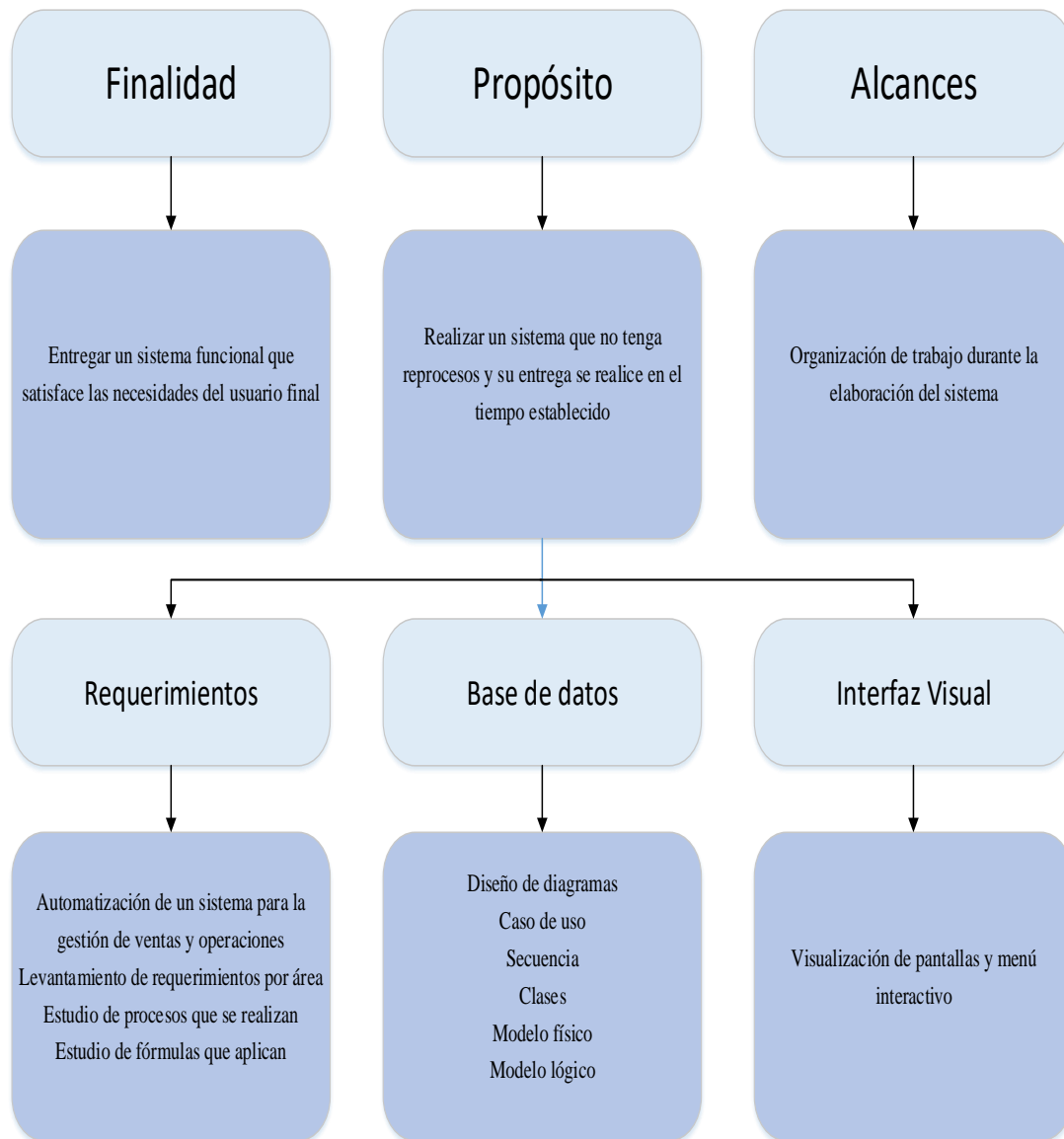


Figura 13 Diagrama de estrategia

4.03.1. Diagrama de clase

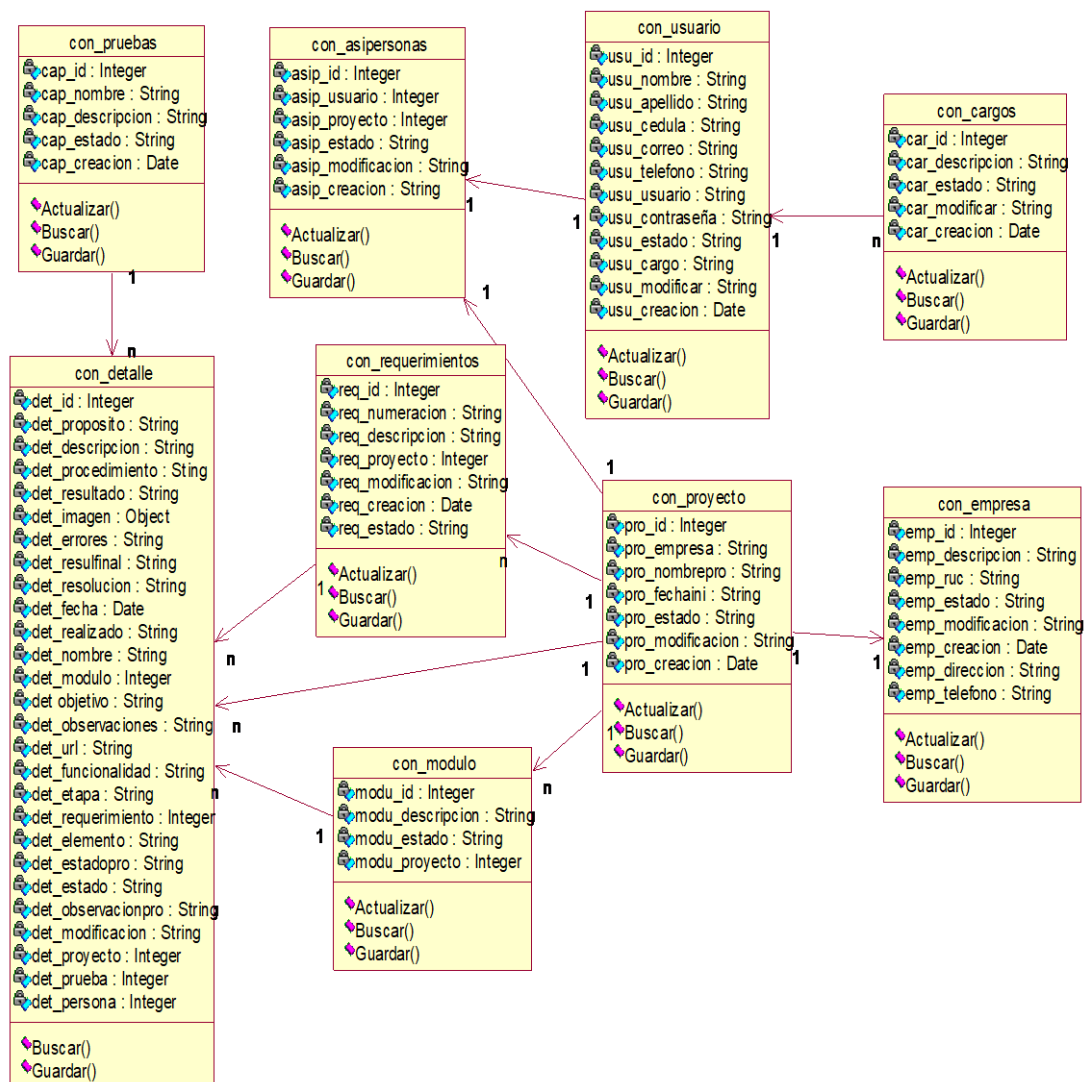


Figura 14 Detalle del diagrama de clase

Diagrama Físico

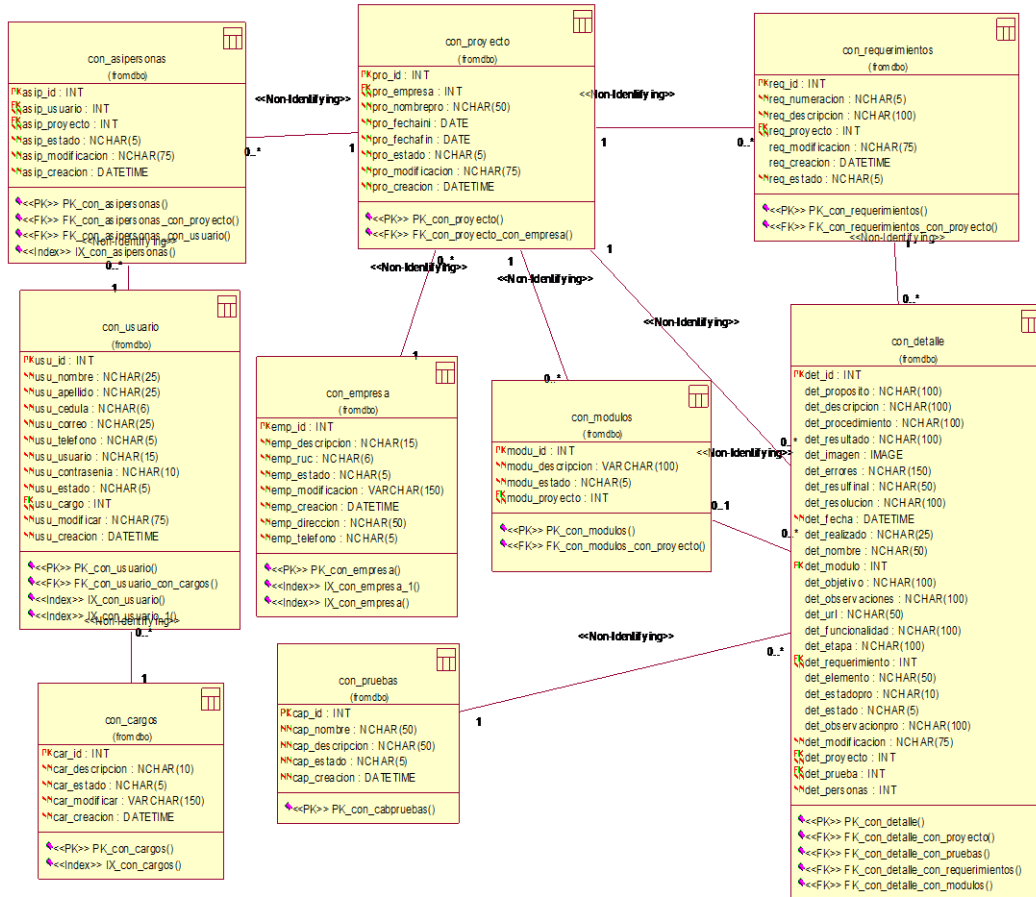


Figura 16 Detalle del diagrama Físico

4.04. Matriz de marco lógico (MML)

Tabla 22
Matriz de marco lógico

Resumen Narrativo	Indicadores	Medios de verificación	Supuestos
Fin de proyecto Automatizar un administrador de pruebas para poder almacenar con sus estados y emitir un informe de avance	Permite interactuar el usuario con el sistema para registrar las actividades	Verifica la información	Verificación de la información
Propósito del Proyecto Almacenar todas las pruebas que se realizan al software generar reportes de avances	Control de la información y manejo adecuado de los registros	Verificación de datos al momento de realizar una consulta	Seguridad de información
Componentes del Proyecto El sistema tiene una interfaz accesible y fácil de usar	Acogida favorable a los usuarios	Buena interacción con el sistema	El uso del sistema sea bueno
Actividad: Implementación del sistema que facilite el almacenamiento de pruebas Manejo de estados Generación de reportes	Ayuda a la empresa a reducir costos Mejora la calidad de los productos evaluados	Identifica y analiza los resultados obtenidos del sistema	Facilidad del sistema

Nota: Descripción de la matriz del marco lógico

4.05. Vistas arquitectónicas

4.05.1. Vista lógica

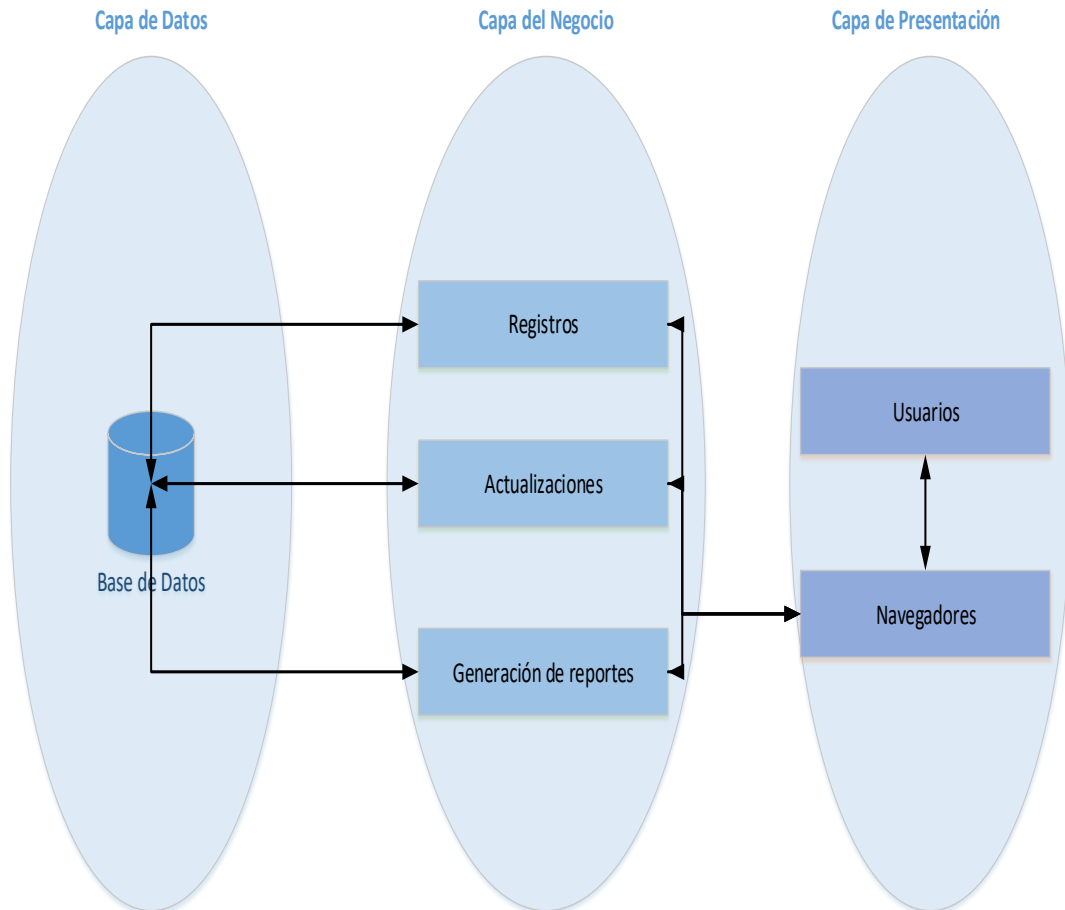


Figura 17 Vista lógica

4.05.2. Vista física

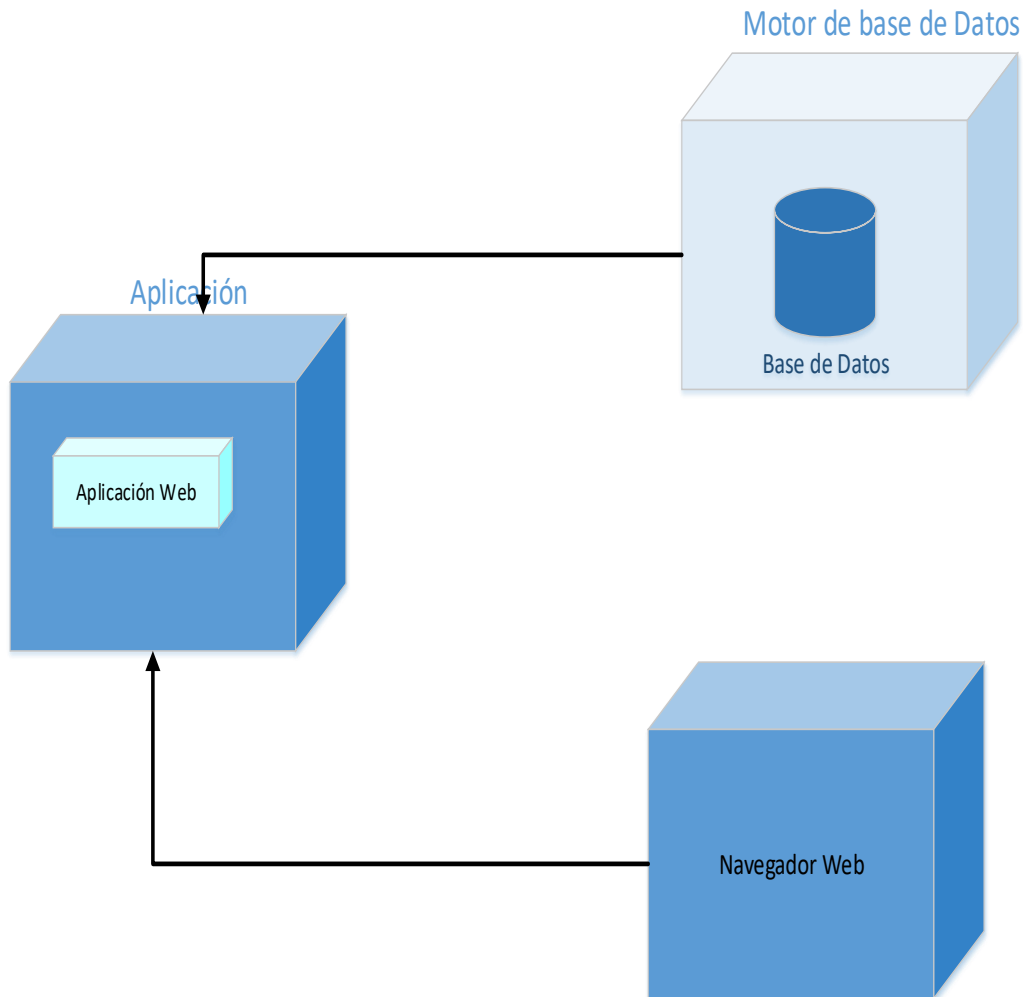


Figura 18 Vista física

4.05.3. Vista de desarrollo

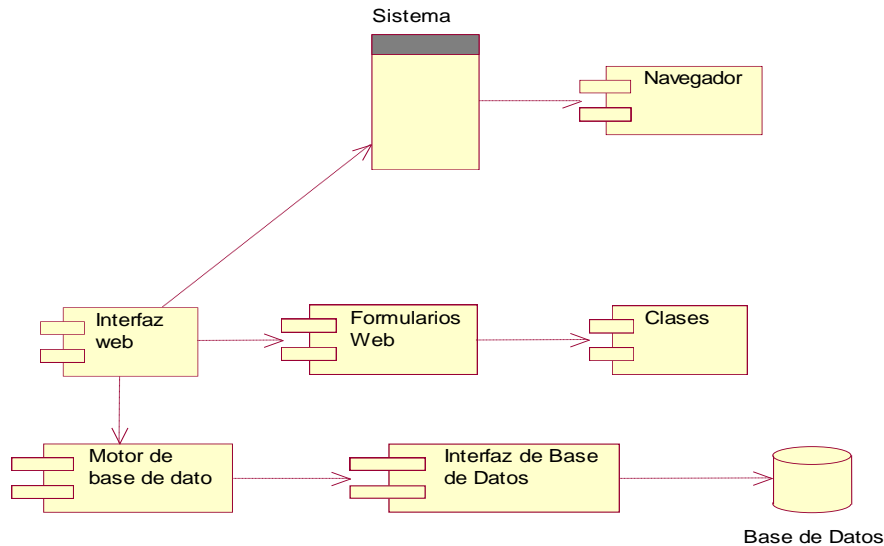


Figura 19 Vista de desarrollo

4.05.4. Vista de proceso

Se detalla el funcionamiento de ingresos al sistema, validación de los datos ingresados

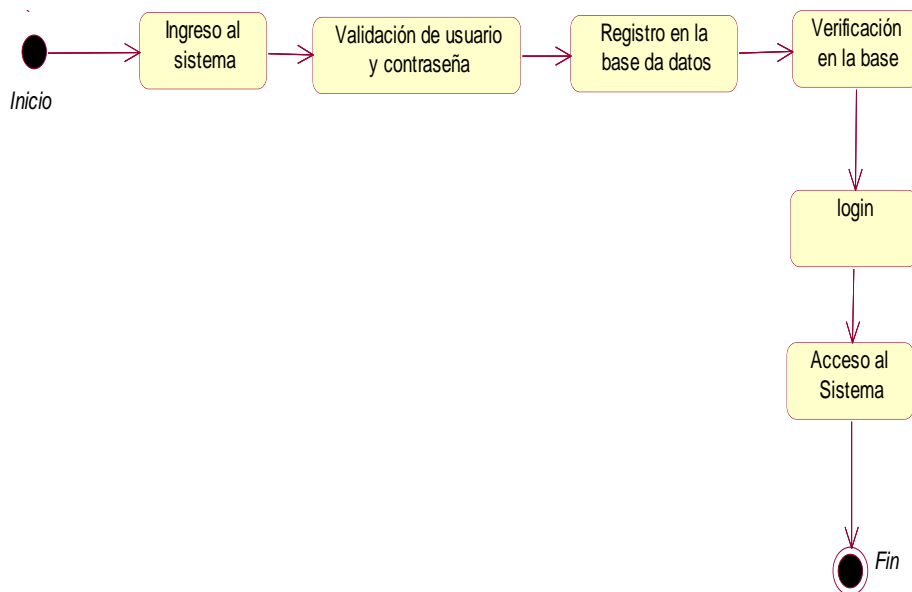


Figura 20 Vista de procesos funcionamiento de ingreso

Se detalla el funcionamiento que realiza el programador al ingresar a ver los estados de las pruebas

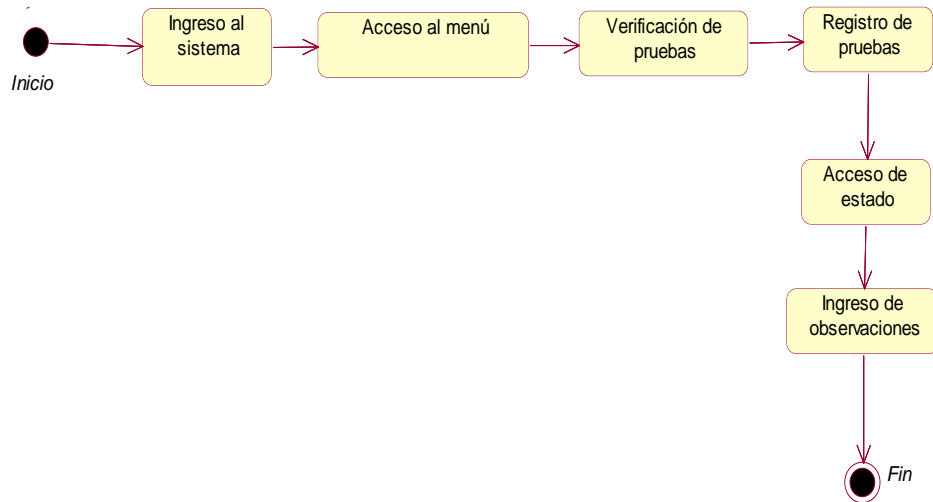


Figura 21 Diagrama de procesos ingreso del programador al software

Se detalla el funcionamiento que realiza la persona encargada del control de calidad al ingresar al sistema

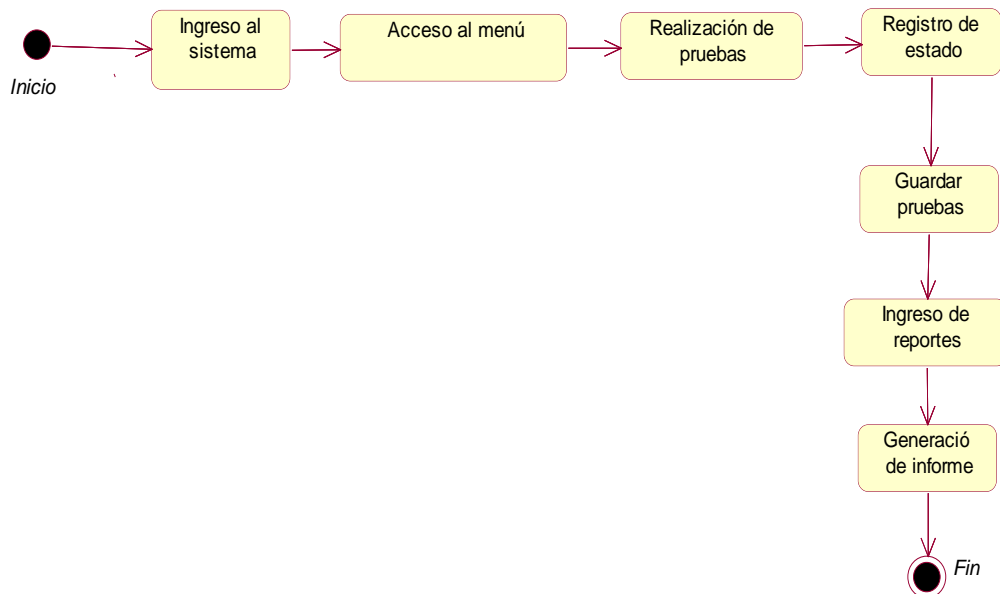


Figura 22 Diagrama de procesos funcionamiento del personal de control

CAPÍTULO V

5. Propuesta

5.01. Antecedentes

Una de las metodologías que utiliza ITSM Ecuador, como parte de su experiencia en implementación de proyectos, es la SCRUM, misma que ha utilizado durante mucho tiempo con excelentes resultados, scrum proceso en el que se aplican de manera regular un conjunto de buenas prácticas para trabajar colaborativamente, en equipo, y obtener el mejor resultado del proyecto. Estas prácticas se apoyan unas a otras y su selección tiene origen en un estudio de la manera de trabajar de equipos altamente productivos.

Roles Funcionales

Product Owner: Analizar, entender y elaborar conjuntamente con el personal los requerimientos. Gestionar el paso a producción de los productos liberados.

Jefe de Proyecto: Coordinar y ejecutar la reunión de planificación con el equipo de trabajo (todos los roles) en el cual explicara los requerimientos acordados con el usuario, además coordinar y ejecutar la reunión de seguimiento, reunión de presentación y reunión de retrospectiva para demostrar el avance de elaboración del software

Desarrolladores y Responsable del control: Ejecutar todas las tareas necesarias para cumplir el objetivo.

Proceso

En Scrum un proyecto se ejecuta en bloques cortos y fijos (iteraciones de un mes y hasta de dos semanas). Cada iteración proporciona un resultado completo, para que al entregar un producto final no tenga errores de gran magnitud y los cambios que se requiera sean mínimos. La planificación de la iteración debe ser de 4 horas máximo. El

equipo elabora la lista de tareas de la iteración necesarias para desarrollar los requisitos que se ha comprometido.

Cada día el equipo realiza una reunión de sincronización 15 minutos máximos. Cada miembro del equipo inspecciona el trabajo que el resto está realizando (dependencias entre tareas, progreso hacia el objetivo de la iteración, obstáculos que pueden impedir este objetivo) para poder hacer las adaptaciones necesarias que permitan cumplir con el compromiso adquirido. Durante la iteración el Facilitador Scrum Master se encarga de que el equipo pueda cumplir con su compromiso y que no disminuya su productividad. (Albaladejo, 2019)

5.02.Descripción

La Empresa ITSM – Ecuador no cuenta con un sistema que permita almacenar los casos de pruebas que se realizan a los sistemas que se encuentran en elaboración, la realización de los test de pruebas se trabaja de manera física impreso y en archivos Excel para su almacenamiento

5.03.Formulación

Como parte de la empresa ITSM – Ecuador se vio el tipo de fallas que se daba por llevar la documentación de cada proyecto de manera física y las pérdidas de archivos de las pruebas que se realizaban a cada uno de los proyectos por lo que se planteó realizar un software en donde almacene cada una de las pruebas y se pueda ver el avance de cada uno de los proyectos y generar informes con el porcentaje de avance.

5.04.Especificación de estándares de programación

El sistema se ha desarrollado bajo los estándares designados por el programador, esto ayuda a mantener orden y organización en el código fuente que va permitir entender a

otros desarrolladores de mejor manera. Para realizar el software se utilizó el entorno de desarrollo integrado de Visual Studio que es compatible con varios lenguajes de programación como C# y .Net creados para realizar aplicaciones de todo tipo.

Se utilizó la siguiente sintaxis para los controles del código y programación en Visual Studio 2017

Tabla 23
Tipos de componentes

Tipo de componente	Descripción
Variables	Espacio reservado en la memoria que pueden cambiar de contenido en la ejecución
Métodos	Bloque de código que contiene una serie de instrucciones
Clases	Propiedades y comportamiento de un tipo de objeto
Objeto	Entidad provista de un conjunto de propiedades o atributos
Connection	Permite la conexión con la base de datos
Command	Realiza una consulta a la base de datos y devuelve un valor
Atributos	Especificación que define una propiedad de un objeto, elemento o archivo

Nota: Especificaremos los componentes y su definición para entender cuál es su funcionalidad

Tabla 24
Tipos de controladores

Tipo de Controlador	Abreviación	Ejemplo
Table	Tbl	Tbl_Pruebas
Label	Lbl	Lbl_Nombre
TextBox	Txt	Txt_Lista

Button	Btn	Btn_Registro
List	Lst	Lst_Pruebas
Form	Frm	Frm_Usuarios
CheckBox	Chk	Chk_Pruebas
GridView	Gdv	Gdv_Cargos

Nota: Definición de los controladores utilizados en la programación del software

5.05.Especificación de interfaz de usuario

La interfaz debe ser fácil de entender para el usuario final

El sistema se encarga de ingresar pruebas que se realizan a un software determinado y administra las pruebas que se aplican dando un estado de aceptación o rechazo al proyecto creado.

El sistema cuenta con una pantalla de ingreso al sistema

Figura 23 Pantalla de ingreso al sistema

El sistema va tener cuatro módulos para las diferentes actividades que van a realizar cada usuario.

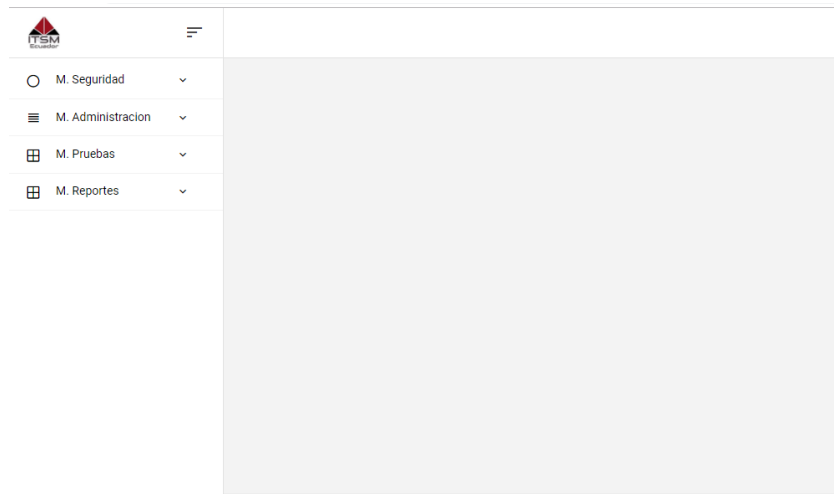
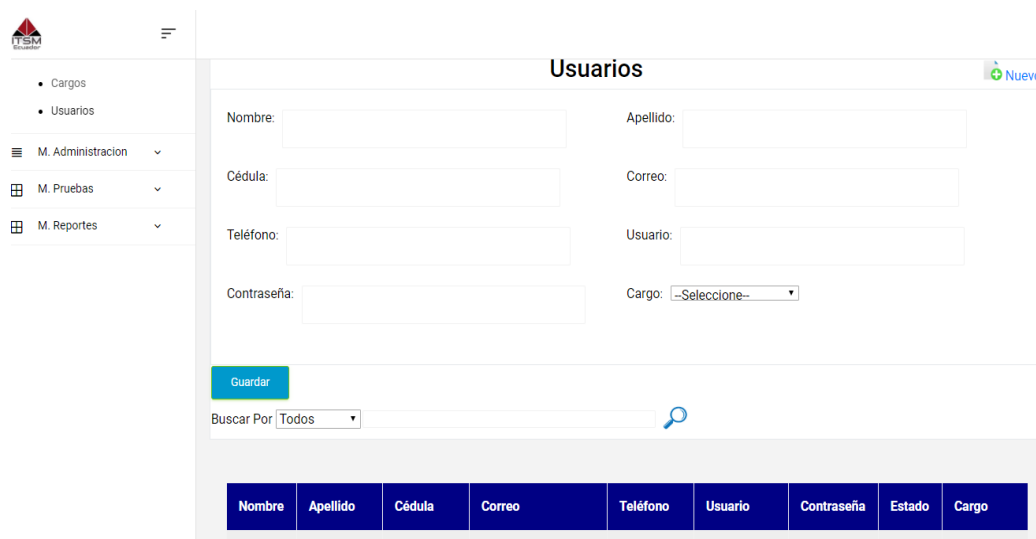


Figura 24 Pantalla de parámetros del menú

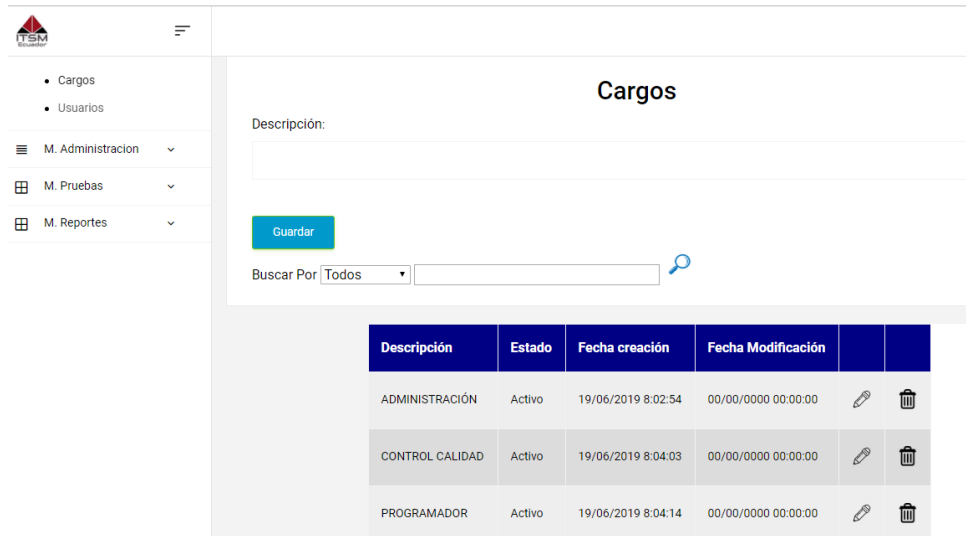
En la siguiente pantalla nos permite ingresar los datos de cada uno de los usuarios que se van a registrar.



Nombre	Apellido	Cédula	Correo	Teléfono	Usuario	Contraseña	Estado	Cargo
--------	----------	--------	--------	----------	---------	------------	--------	-------

Figura 25 Pantalla de registro del usuario

En esta pantalla nos va permitir registrar el cargo



Descripción	Estado	Fecha creación	Fecha Modificación		
ADMINISTRACIÓN	Activo	19/06/2019 8:02:54	00/00/0000 00:00:00		
CONTROL CALIDAD	Activo	19/06/2019 8:04:03	00/00/0000 00:00:00		
PROGRAMADOR	Activo	19/06/2019 8:04:14	00/00/0000 00:00:00		

Figura 26 Ingreso de Cargo

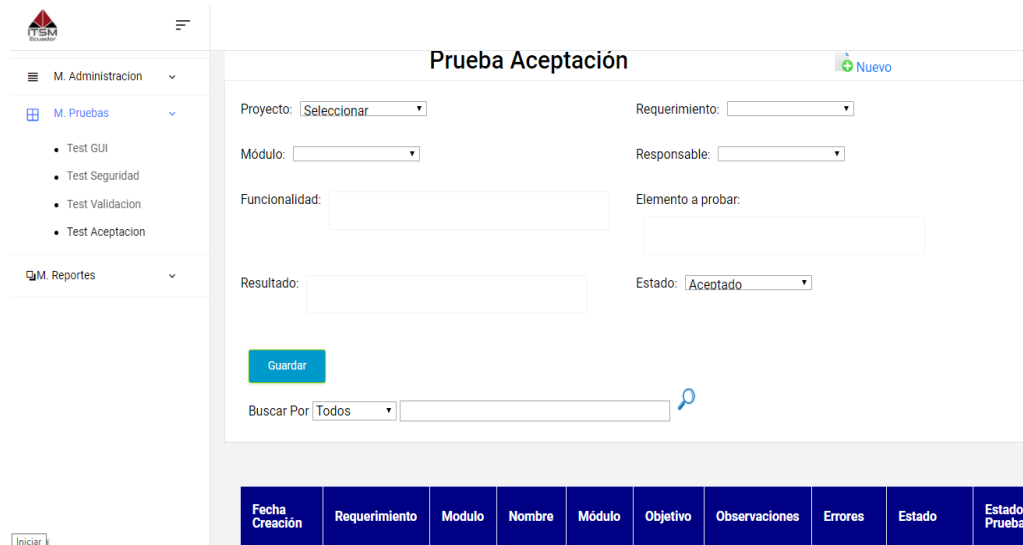
Pantalla de menú para ingreso a los Test de pruebas



Fecha Creación	Requerimiento	Modulo	Nombre	Módulo	Objetivo	Observaciones	Errores	Estado	Estado Prueba
----------------	---------------	--------	--------	--------	----------	---------------	---------	--------	---------------

Figura 27 Menú de Test de pruebas

Pantalla del Test para el registro de la prueba de aceptación



Prueba Aceptación [Nuevo](#)

Proyecto: Requerimiento:

Módulo: Responsable:

Funcionalidad: Elemento a probar:

Resultado: Estado:

[Guardar](#)

Buscar Por: [🔍](#)

Fecha Creación	Requerimiento	Módulo	Nombre	Módulo	Objetivo	Observaciones	Errores	Estado	Estado Prueba
----------------	---------------	--------	--------	--------	----------	---------------	---------	--------	---------------

Figura 28 Test de prueba de Aceptación

Pantalla del Test para el registro de la prueba de validación



Prueba Validación [Nuevo](#)

Proyecto: Requerimiento:

Módulo: Responsable:

Propósito: Descripción:

Procedimiento: Resultados:

Imagen: Ningún ...ionado Errores:

Resultado Fin:

Resolución:

Estado:

Figura 29 Test de prueba de Validación

Pantalla para Test para el registro de la prueba de



Figura 30 Test de prueba de Seguridad

5.06. Especificación de pruebas de unidad

Se realizará pruebas a cada módulo establecido en el sistema para su aprobación

Tabla 25

Prueba de unidad (PU001), ingreso al sistema

Prueba de unidad Ingreso al sistema	
Identificador de pruebas	PU001
Métodos a probar	Ingreso al sistema
Objetivo de la prueba	Permite el acceso al sistema sin problemas
Datos de Entrada	
Usuario, Contraseña	
Resultados Esperados	
Los datos ingresados van a ser verificados y no permitir el acceso de un usuario que no se encuentre registrado en la base de datos	
Comentario	
Tener un control de los accesos de los usuarios que ingresan al sistema	
Nota: Descripción del ingreso al sistema	

Tabla 26***Prueba de unidad (PU002), compilar código***

Prueba de unidad Compilación de código	
Identificador de pruebas	PU002
Métodos a probar	Compilación del código
Objetivo de la prueba	Verificar errores al compilar
Datos de Entrada	
Compilación de cada uno de los módulos para verificación de funcionamiento	
Resultados Esperados	
El sistema debe estar compilado hasta el final del proceso para evitar errores	
Comentario	
Las pruebas funcionan correctamente	

Nota: Descripción para compilar código**Tabla 27*****Prueba de unidad (PU003), generación de reportes***

Prueba de unidad Generación de reportes	
Identificador de pruebas	PU003
Métodos a probar	Reportes
Objetivo de la prueba	Verificar la información que sea verídica
Datos de Entrada	
Ingreso a los diferentes módulos disponibles en la base de datos	
Resultados Esperados	
Obtener la información ingresada en los módulos de la base de datos	
Comentario	
Prueba realizada con éxito	

Nota: Descripción del ingreso al sistema

Tabla 28***Prueba de unidad (PU004) Almacenamiento en la base de datos***

Prueba de unidad Almacenamiento en la base de datos	
Identificador de pruebas	PU004
Métodos a probar	Almacenamiento en la base de datos
Objetivo de la prueba	Verificar el ingreso y almacenamiento en la base de datos
Datos de Entrada	
Información ingresada y manipulada por el usuario	
Resultados Esperados	
Verificar si la información ingresada es la misma que esta almacenada en la base de datos	
Comentario	
Prueba realizada con éxito	

Nota: Descripción del almacenamiento en la base de datos

5.07.Pruebas de aceptación

Mediante las pruebas de aceptación se establece el grado de confianza de un sistema para su correcto funcionamiento durante la ejecución del sistema y de esta manera evitar reprocesos.

Tabla 29***Prueba de aceptación (PA001), Ingreso de usuarios por cargo***

Ingreso de usuarios mediante perfiles	
Identificador de pruebas	PA001
Tipo de usuario	Administrador Personal de control Programador
Objetivo de la prueba	Establecer perfiles mediante su cargo
Secuencia de evento	Acceso al sistema ingresar nombre y contraseña validar
Resultados Esperados	No permitir el acceso a usuarios que no estén registrados en la base de datos
Comentario	Prueba realizada con éxito
Estado	Aceptado

Nota: Descripción del ingreso de usuario por su cargo

Tabla 30***Prueba de Aceptación (PA002), Crear reportes de pruebas y estados***

Crear reportes de pruebas y estados	
Identificador de pruebas	PA002
Tipo de usuario	Personal de control, Administrador
Objetivo de la prueba	Crear un reporte de pruebas estados
Secuencia de evento	Se ingresa al módulo de reportes y se elige que tipo de reportes necesita
Resultados Esperados	Visualización de un reporte que será analizada para emitir un informe
Comentario	Prueba realizada con éxito
Estado	Aceptado

Nota: Descripción de ingresos de proyectos

Tabla 31
Prueba de unidad (PA003), ingreso de proyectos

Ingreso de proyectos	
Identificador de pruebas	PA003
Tipo de usuario	Administrador
Objetivo de la prueba	Ingreso de datos por proyecto
Secuencia de evento	
Ingresar al sistema y llenar el formulario con los datos pedidos por el sistema	
Resultados Esperados	
Almacenar la información del proyecto a evaluar	
Comentario	
Prueba realizada con éxito	
Estado	
Aceptado	

Nota: Descripción de ingresos de proyectos

Tabla 32
Prueba de Aceptación (PU004), detalle de carga de usuario

Detalle de carga de usuario	
Identificador de pruebas	PA004
Tipo de usuario	Administrador, Encargado del control, Programador
Objetivo de la prueba	Establecer el tiempo de respuesta cuando los usuarios ingresen
Secuencia de evento	
Ingresar y entrar a cada módulo habilitado para ejecutar las pruebas	
Resultados Esperados	
Visualización de respuesta del sistema al procesar la información ingresada	
Comentario	
Prueba realizada con éxito	
Estado	
Aceptado	

Nota: Descripción de carga de usuarios

5.08. Especificación de pruebas de carga

Las pruebas se realizan para evidenciar el rendimiento del sistema, la capacidad máxima que tiene un servidor web (hardware y software).

La prueba en este caso consiste en simular una carga de trabajo similar o superior a la que tendrá cuando ya este funcionando, con el fin de ver que el sistema cumpla con los requerimientos establecidos.

Tabla 33

Prueba de carga (PC001), ingreso de usuarios mediante cargos

Ingreso de usuarios mediante perfiles	
Identificador de pruebas	PC001
Tipo de prueba	Prueba normal
Objetivo de la prueba	Verificación del tiempo de respuesta cuando los usuarios ingresen al sistema
Descripción	
Permite verificar el funcionamiento del sistemas después de su aplicación	
Resultados Esperados	
Respuesta inmediata del sistema	
Comentario	
Prueba realizada con éxito	

Nota: Descripción del ingreso al sistema mediante cargos

Tabla 34*Prueba de carga (PC002), detalle de carga de varios usuarios*

Detalle de carga de varios usuarios	
Identificador de pruebas	PC002
Tipo de prueba	Prueba con varios usuarios
Objetivo de la prueba	Establecer el tiempo de carga con varios usuarios ejecutando al mismo tiempo
Descripción	
Permite verificar el funcionamiento con varios usuarios conectados	
Resultados Esperados	
Trabajo simultaneo sin novedad	
Comentario	
Prueba realizada con éxito	

Nota: Descripción del ingreso con varios usuarios

Tabla 35*Prueba de carga (PC003,)datos de pruebas de formularios*

Datos de pruebas de formularios	
Identificador de pruebas	PC003
Tipo de prueba	Encargado del control, Administrador
Objetivo de la prueba	Verificación de los datos ingresados en cada uno de los formularios de pruebas
Descripción	
Permite verificar el ingreso de las pruebas que se realizan dependiendo de cada proyecto	
Resultados Esperados	
Almacenamiento de la información ingresada	
Comentario	
Prueba realizada con éxito	

Nota: Descripción de datos de pruebas de formularios

5.09. Configuración del ambiente mínimo

La aplicación está diseñada por capas, y cumpliendo los estándares de programación y metodologías, permitiendo el funcionamiento correcto y reducir tiempos para el análisis de pruebas.

Tabla 36
Especificación de instalación de componentes

Especificación de instalación	
Base de datos	SQL Server 2016
Hardware servidor	Procesador de 3GHZ Disco Duro 120 GB Memoria RAM 2GB
Equipo Cliente	Procesador de 2GHZ Disco Duro 80 GB Memoria RAM 2GB

Nota: Especificación de componentes

CAPÍTULO VI

6. Aspectos administrativos

6.01. Recursos

El presente proyecto fue intervenido por 3 factores importantes

6.01.1. Recursos Humanos

Personal que se dedicó en la realización del proyecto y ayudaron a terminar el desarrollo del sistema.

Tabla 37
Detalle de los recursos humanos

Recursos Humanos			
Humano	Nombre	Actividad	Responsabilidad
Tutor	Ing. Jaime Padilla	Tutor y guía del proyecto de tesis	Revisar los avances del documento escrito Revisar el desarrollo del sistema
Lector	Ing. Johnny Coronel	Revisar el desarrollo del sistema y el escrito	Informar si el documento escrito se encuentra con las normas indicadas Revisar el funcionamiento del sistema
Empresa	ITSM-Ecuador	Autorización para entrevistar a los usuarios	Permite la recopilación de la información

Nota: Descripción de los recursos humanos

6.01.2. Recursos Materiales

Herramientas de trabajo utilizadas para la creación del sistema en el desarrollo y en el escrito de la tesis

Tabla 38
Detalle de los recursos materiales

Recursos Materiales	
Nombre	Descripción
Laptop	Realización del proyecto dentro y fuera de casa
Computadora de escritorio	Implementación de avances realizados

Nota: Descripción de los recursos materiales

6.01.3. Recursos Tecnológicos

Recursos que permitieron realizar las investigaciones para realizar el desarrollo de la programación

Tabla 39
Detalle de recursos tecnológico

Nombre	Descripción
Hosting	Permite visualizar el sistema en cualquier parte
Internet	Servicio utilizado para las investigaciones necesarias

Nota: Descripción de los recursos tecnológicos

6.02.Presupuesto

Tabla 40
Detalle del presupuesto

Presupuesto del proyecto			
Concepto			
1	Servicios Básicos	Agua luz, teléfono, Internet	300.00 USD
2	Mano de Obra	Elaboración del sistema	500.00 USD
3	Materiales	Impresiones	50.00 USD
4	Alimentación	Almuerzos	70.00
5	Otros	Varios	100.00 USD
Total			1020 USD

Nota: Descripción del presupuesto

6.03.Cronograma

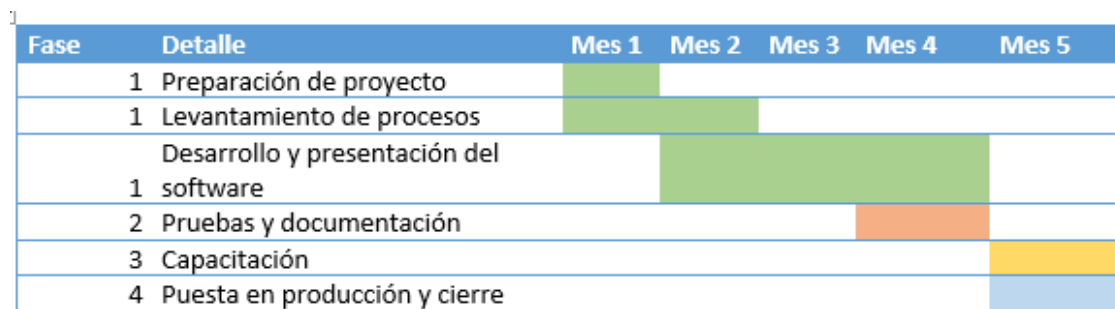


Figura 31 *Detalle de actividades*

Columna1	Columna2	Columna3	Columna4
Nombre de tarea	Duración	Comienzo	Fin
Proyecto Control	100 días	01-dic	19-abr
Sprint 1	20 días	3-dic	28-dic
Presentación	3 días	3-dic	5-dic
Levantamiento de procesos	10 días	6-dic	19-dic
Generación de pantallas	6 días	20-dic	27-dic
Presentación sprint 1	1 día	27-dic	28-dic
Sprint 2	30 días	1-ene	8-feb
Diseño DB	5 días	1-ene	7-ene
Módulo de Administración	10 días	8-ene	22-ene
Usuarios	3 días	23-ene	25-ene
Perfil Roles	5 días	28-ene	1-feb
Presentación sprint 2	4 días	4-feb	8-feb
Sprint 3	20 días	11-feb	8-mar
Módulo de Pruebas	5 días	11-feb	15-feb
Pruebas	5 días	18-feb	22-feb
mantenimientos	5 días	25-feb	1-mar
Presentación sprint 3	5 días	4-mar	8-mar
Sprint 4	20 días	11-mar	5-abr
Módulo de reportes	10 días	11-mar	22-mar
designación de reportes	5 días	25-mar	29-mar
Presentación sprint 4	5 días	1-mar	5-mar
Sprint 5	10 días	8-abr	19-abr
Realización de documentación	8 días	8-abr	17-abr
Fin del Proyecto	2 días	19-abr	19-abr

Figura 32 Cronograma de actividades especificados por sprint

CAPÍTULO VII

7. Conclusiones y recomendaciones

Conclusiones

Con la automatización e implementación del software “Administración de pruebas” en la empresa ITSM-Ecuador ayudo a llevar un orden establecido de los test de pruebas para la verificación del avance de los módulos de cada proyecto que se está elaborando internamente. El desarrollo del aplicativo mejoro notablemente el control y desempeño de las actividades para cada uno de los usuarios que manipulan la aplicación.

Con la emisión de informes se verifica los avances mediante los estados de los test de pruebas y se puede establecer el nivel de trabajo de los involucrados en cada proyecto.

Recomendaciones

Dentro de la estructura organizacional es recomendable dar capacitaciones a los trabajadores, de los avances tecnológicos ya que cada día se dan nuevas maneras de mejorar el trabajo dentro de las organizaciones

Es recomendable actualizar el aplicativo con nuevos test de pruebas con el fin de mejorar el control de calidad dentro de la empresa y entregar productos de alta calidad a los clientes.

REFERENCIAS BIBLIOGRÁFICAS

- Albaladejo, X. (15 de 02 de 2019). *Proyectos Ágiles.org*. Obtenido de Proyectos Ágiles.org: <https://proyectosagiles.org/que-es-scrum/>
- Deming, E. (20 de 01 de 2019). Introducción a la calidad de Software. *Introducción a la calidad de Software*. Quito, Pichincha, Ecuador: Redalyc.org.
- Dimes, T. (2016). *Conceptos básicos de SCRUM*. España: Babelcube Inc.
- García, R. (12 de 05 de 2017). *ISO 25000 calidad del producto software*. Obtenido de ISO 25000 calidad del producto software:
<https://iso25000.com/index.php/normas-iso-25000?limit=4&start=4>

ANEXOS

Anexo A Diagrama de Casos de Uso

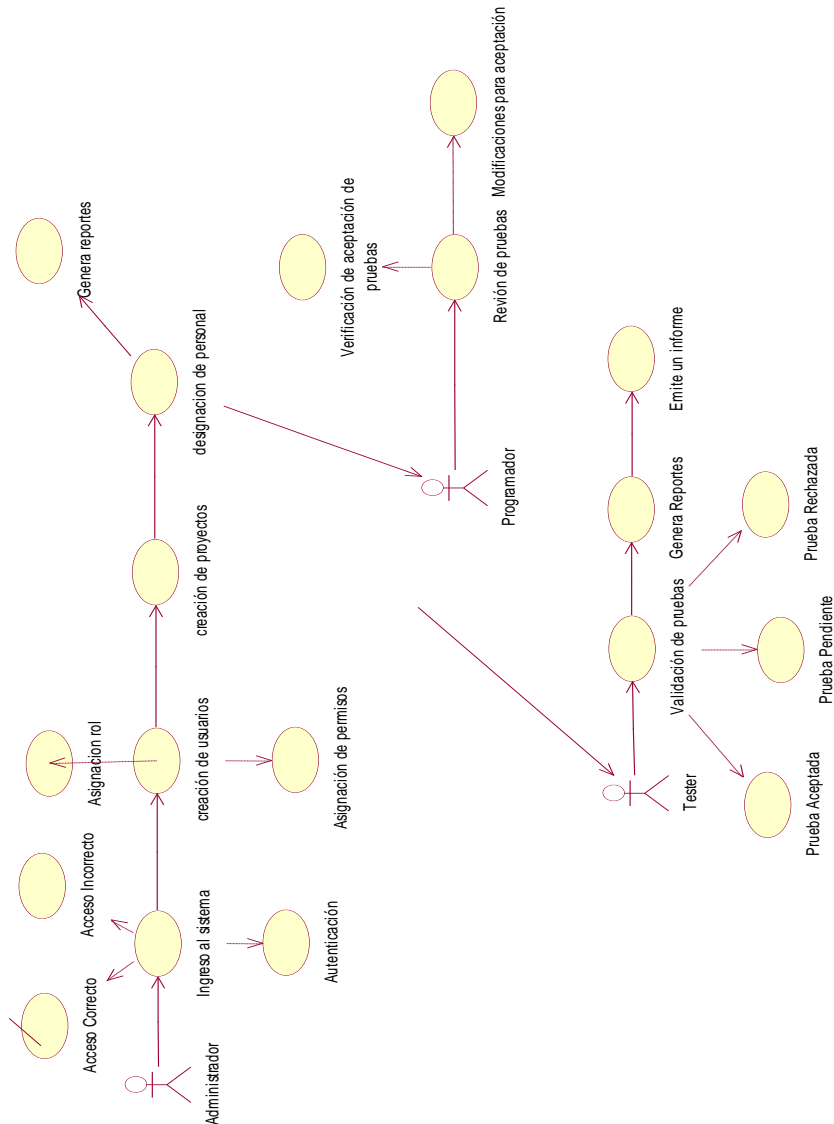


Figura 33 Diagrama de Casos de Uso

Anexo B Tabla de Requerimientos

Tabla 41
Requerimientos Funcionales

ID	Descripción	Fuente	Prioridad	Tipo	Estado	Usuarios Involucrados
RF001	El sistema debe tener registrado 3 cargos principales para su uso	Administrador Usuario Control de calidad	Alta	Sistema	En revisión	Administrador Usuario Control de calidad
RF002	El sistema debe registrar los usuarios en la base de datos	Usuario	Alta	Sistema	En revisión	Administrador Usuario
RF003	El sistema debe registrar los cargos en la base de datos	Administrador	Alta	Sistema	En revisión	Administración
RF004	El Sistema debe registrar las empresas en la base de datos	Administrador	Alta	Sistema	En revisión	Administración
RF005	El sistema debe registrar el proyecto y la asignación de personas en la base de datos	Administrador Encargado del control de calidad	Alta	Sistema	En revisión	Administración
RF006	El sistema debe registrar los Test de Pruebas en la base de datos	Encargado del control de calidad	Alta	Sistema	En revisión	Encargado del control

RF007	El sistema debe registrar la observación del programador	Programador	Alta	Sistema	En revisión	Programador
RF008	El sistema podrá realizar actualizaciones en las tablas de usuarios, cargos, empresas, proyectos	Administrador	Alta	Sistema	En revisión	Administrador

Anexo C Manual Instalación



CARRERA DE ANÁLISIS DE SISTEMAS

**ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL
SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE
SEGUROS DE LA EMPRESA ITSM ECUADOR, PERÍODO
OCTUBRE 2018 – MARZO 2019.**

MANUAL DE INSTALACIÓN

AUTOR: CACUANGO CATUCUAMBA JESSICA ESTEFANIA

QUITO 2018 - 2019

Índice

Manual de Instalación.....	74
1.01. Introducción	74
1.02. Objetivos	74
1.03. Requisitos Mínimos.....	74
Instalación de SQL SERVER 2016	75
Instalación de Visual Studio 2017	83

Tabla de Figuras

Figura 1 Ejecutables de instalación	74
Figura 2 Carga de la instalación	74
Figura 3 Instalación del SQL	75
Figura 4 Selección de instalación	75
Figura 5 Instalación del SQL EXPRESS	76
Figura 6 Aceptación de licencia.....	76
Figura 7 Carga de productos	77
Figura 8 Selección de características de SQL server	77
Figura 9 Pantalla de configuración de Instancia.....	78
Figura 10 Configuración del Servidor	78
Figura 11 Establecer la contraseña y el modo de autenticación	79
Figura 12 Instalación de la configuración.....	79
Figura 13 Aceptar términos de la configuración	80
Figura 14 Comprobación de las características a instalar	80
Figura 15 Pantalla de progreso de instalación	81
Figura 16 Pantalla de finalización	81
Figura 17 Pagina de descarga	82
Figura 18 Carga de instalación y descarga	82
Figura 19 Características de Visual Studio.....	83
Figura 20 Proceso de carga de instalación.....	83

Manual de Instalación

1.01.Introducción

El manual de instalación se realiza con la finalidad de proporcionar la información necesaria para las configuraciones de cada programa utilizado en su ejecución.

1.02.Objetivos

Proporcionar al usuario toda la información necesaria para la implementación del sistema en cualquier ordenador.

1.03.Requisitos Mínimos

Hardware

- **Memoria RAM:** 4GB
- **Procesador:** Intel Core i3 3.40GHz
- **Disco Duro:** 500GB

Software

- **Sistema Operativo:** Windows 10
- **Navegador:** Google Chrome, Mozilla Firefox

Instalación de SQL SERVER 2016

Una vez descargado los ejecutables de SQL Server 2016, se debe empezar la instalación

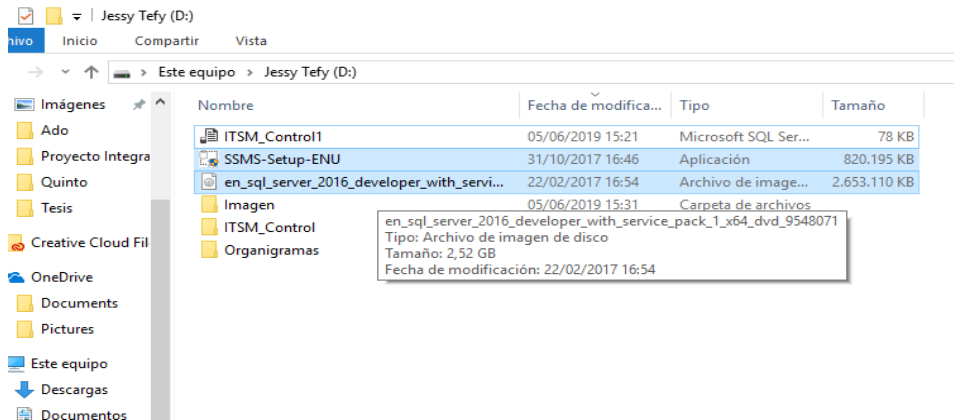


Figura 1 Ejecutables de instalación

Ejecutar el instalador y dejar que se cargué

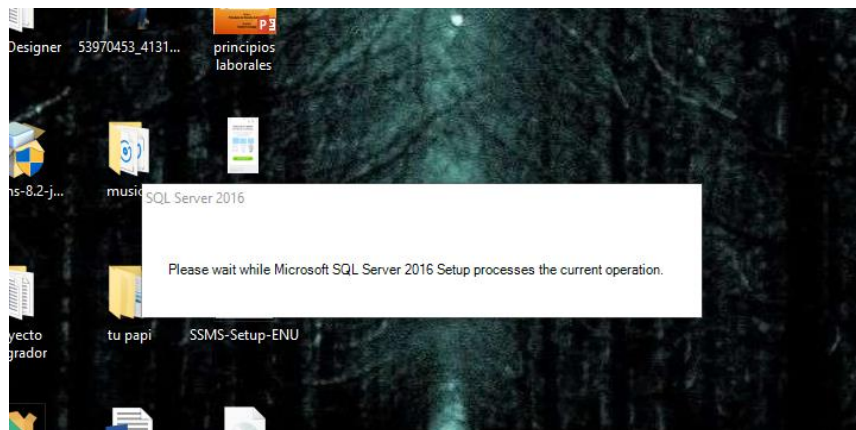


Figura 2 Carga de la instalación

Una vez ejecutado inicia el programa de instalación

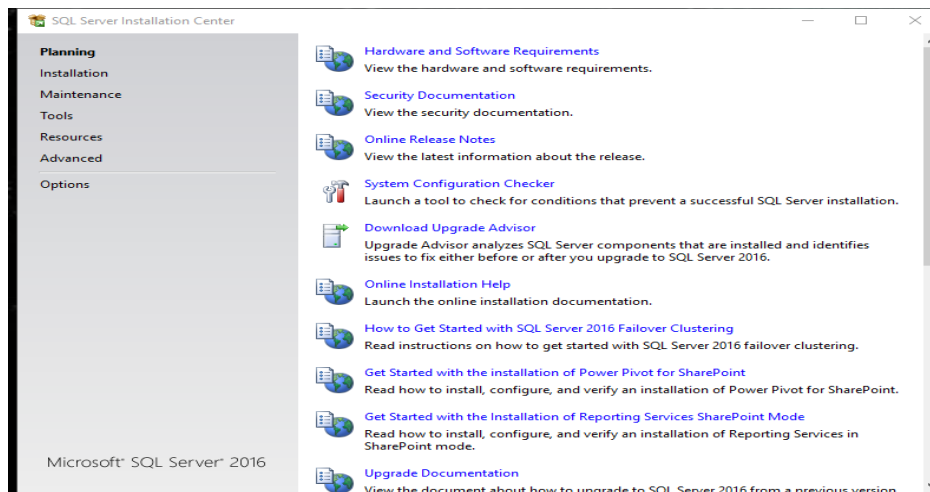


Figura 3 *Instalación del SQL*

Escoger la opción New SQL para empezar con la configuración

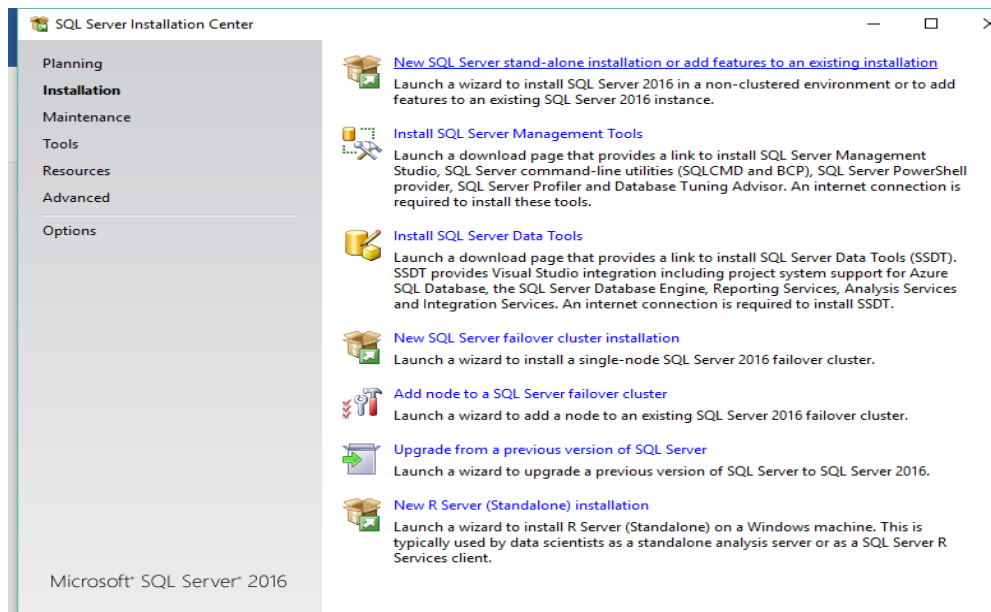


Figura 4 *Selección de instalación*

Seleccionar SQL Server para y poner en “Next”

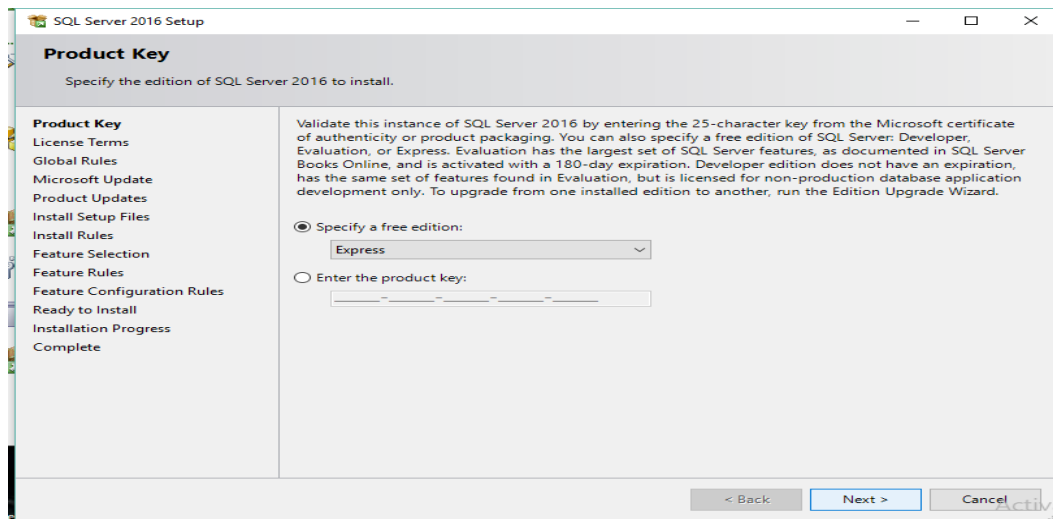


Figura 5 *Instalación del SQL EXPRESS*

Aceptar los términos necesarios para empezar la instalación

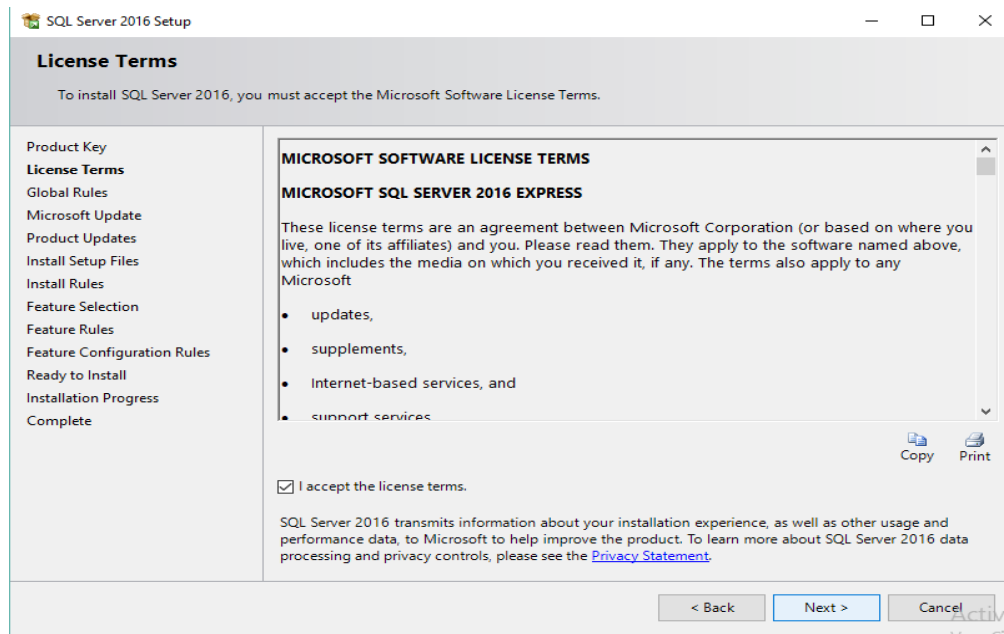


Figura 6 *Aceptación de licencia*

Instalación de complementos al terminar con su ejecución dar en “Next”

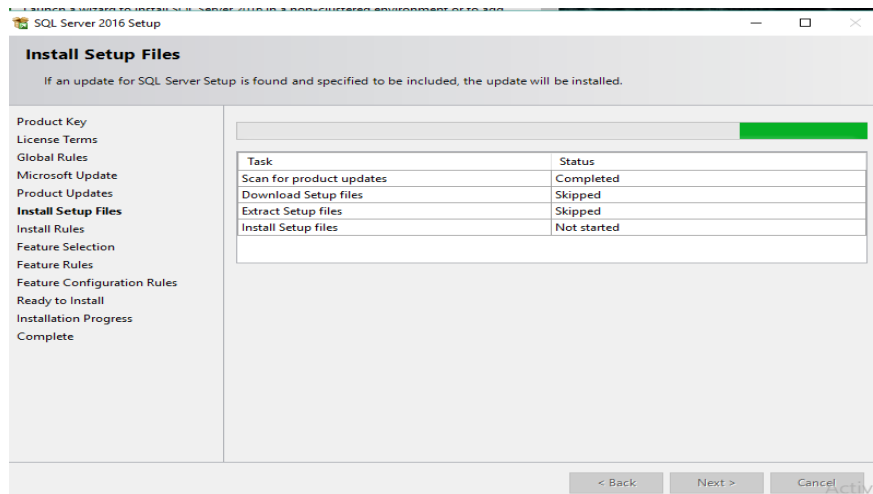


Figura 7 Carga de productos

Seleccionar las características “Servicios de Motor de base de datos”. Dar clic en “Next”

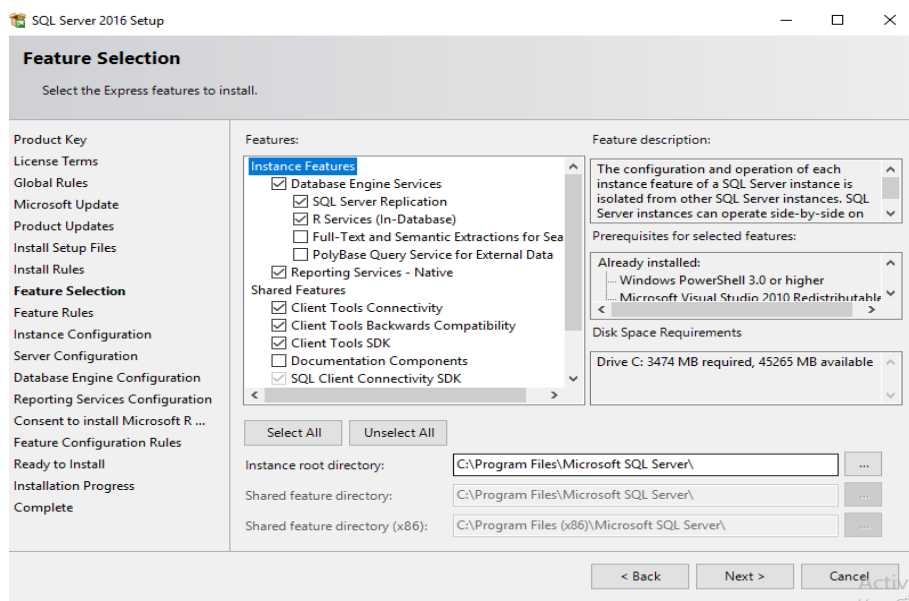


Figura 8 Selección de características de SQL server

Configuración de la instancia si se va trabajar con usuarios se debe ingresar el ID la dirección y agregar y dar en “Next”

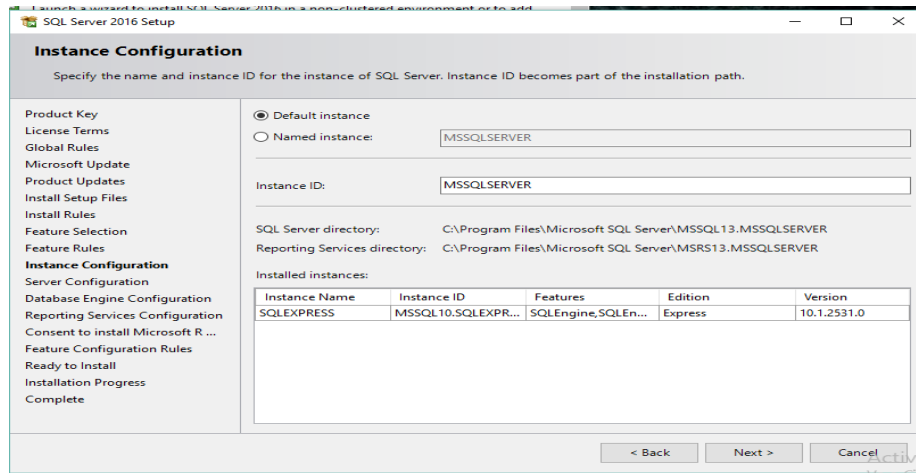


Figura 9 Pantalla de configuración de Instancia

Se mostrata una ventana de “Configuración de Servidor”. Dar en “Next”

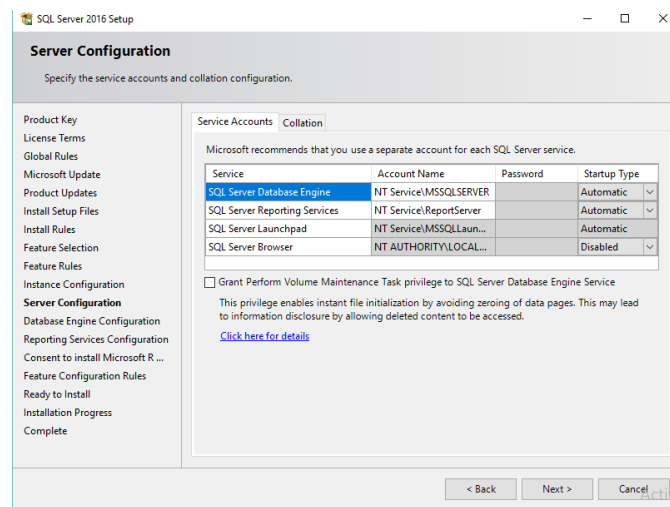


Figura 10 Configuración del Servidor

En la configuración del motor e base de datos se debe seleccionar “Autenticación de Windows”, agregar y dar en “Next”.

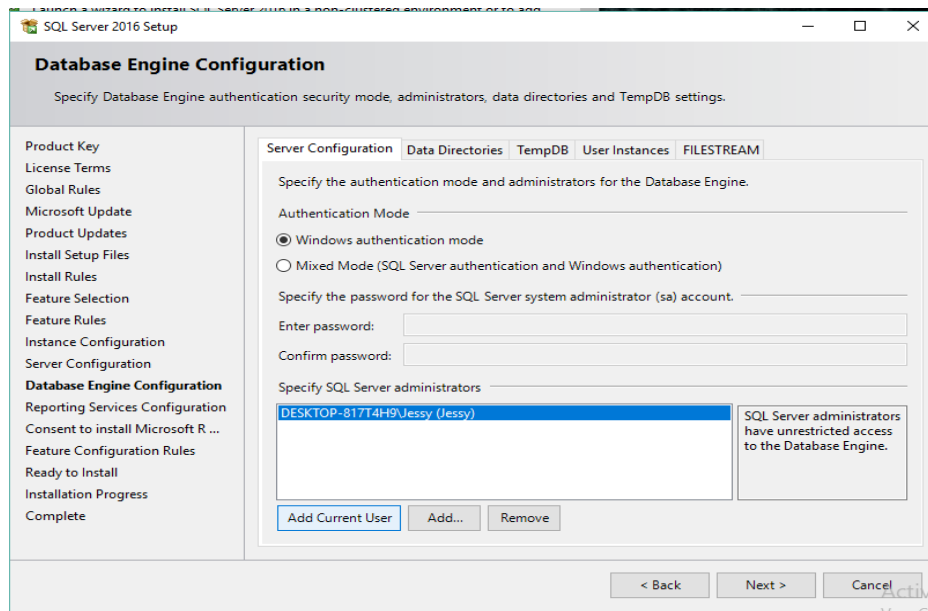


Figura 11 Establecer la contraseña y el modo de autenticación

Instalar la configuración realizada y dar en “Next”

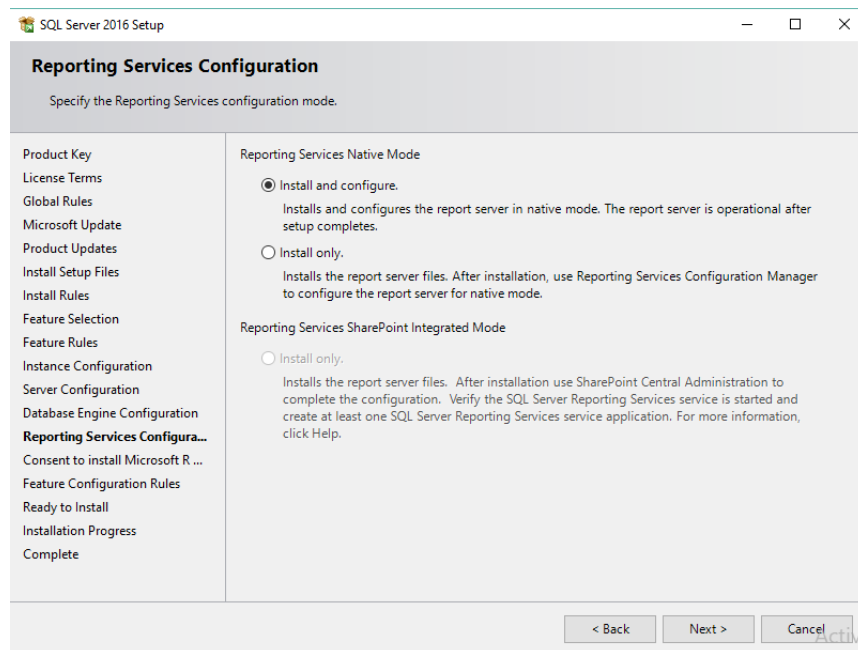


Figura 12 Instalación de la configuración

Aceptar los términos de la configuración y dar “Next”

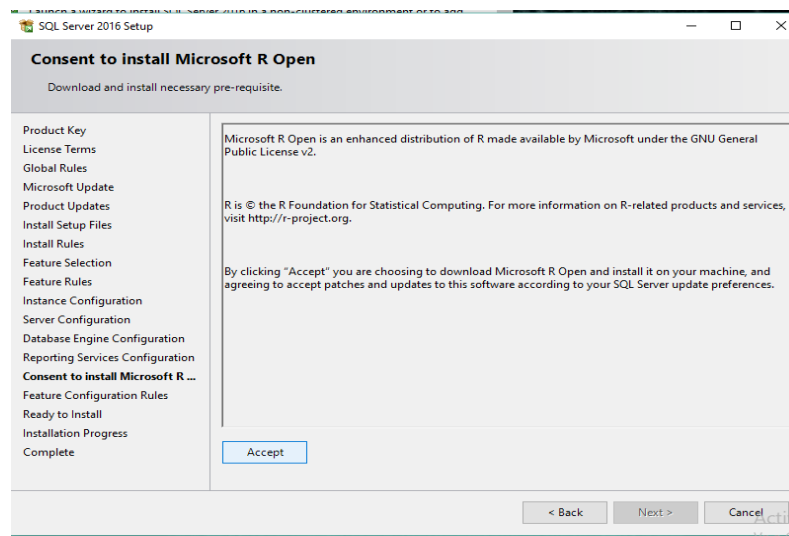


Figura 13 Aceptar términos de la configuración

Aparecerá una pantalla donde se muestra todas las características que se van a instalar.

Y dar “Next”

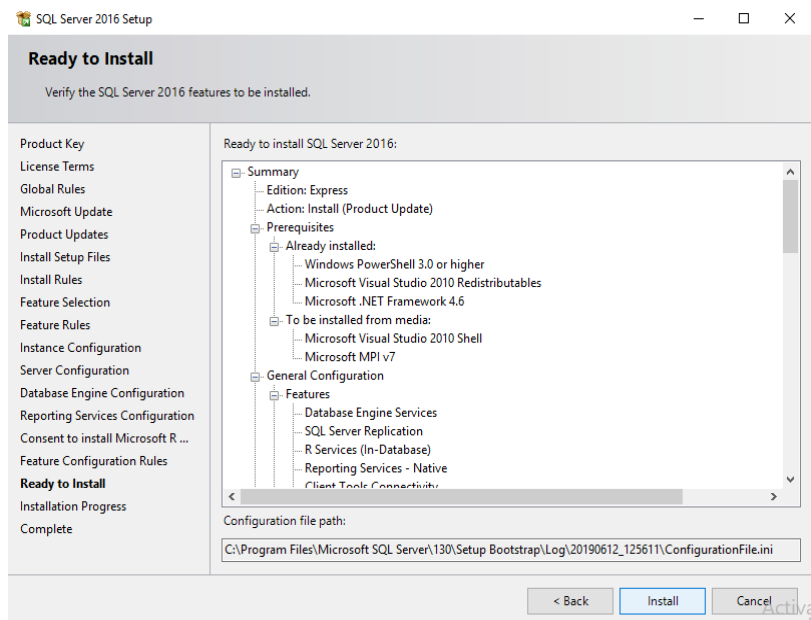


Figura 14 Comprobación de las características a instalar

Se visualizará el progreso de la instalación

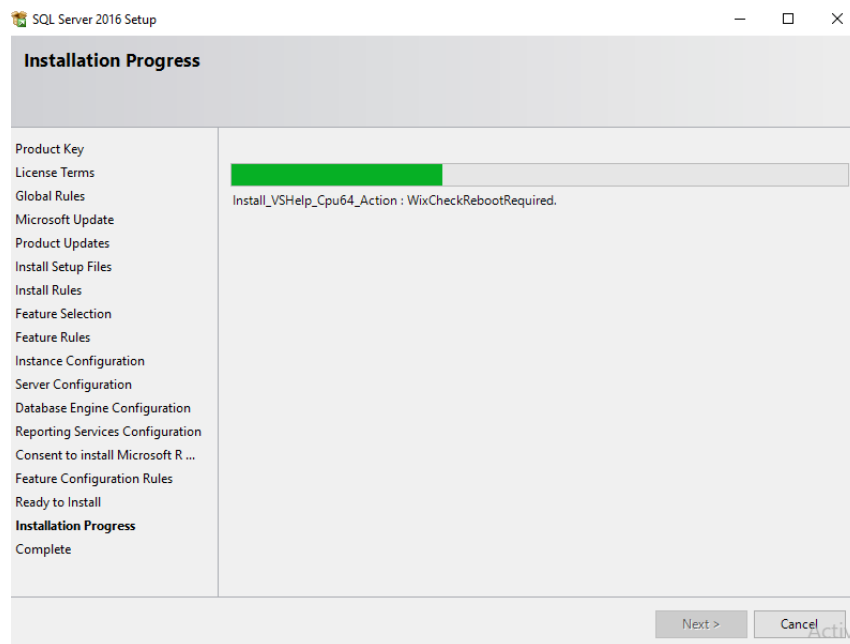


Figura 15 Pantalla de progreso de instalación

Pantalla de la información que la operación se completó satisfactoriamente y dar en “Close”

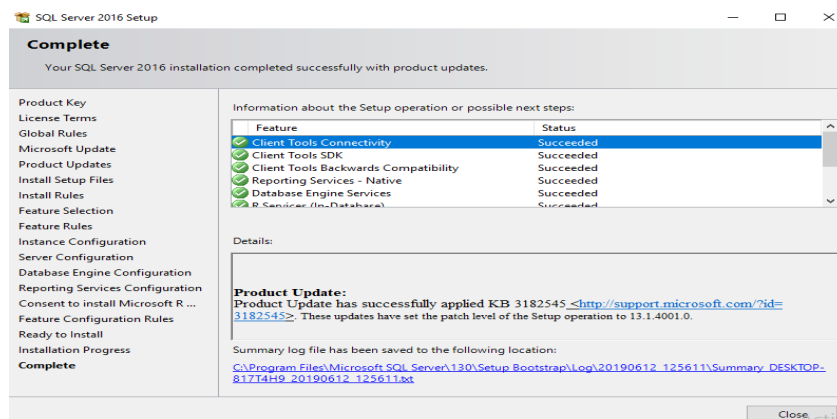


Figura 16 Pantalla de finalización

Instalación de Visual Studio 2017

Una vez descargado el instalador



Figura 17 Pagina de descarga

Ejecutar el instalador como administrador

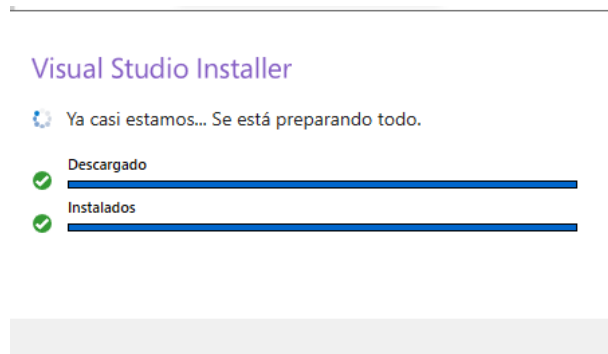


Figura 18 Carga de instalación y descarga

Características para la instalación de Visual Studio 2017

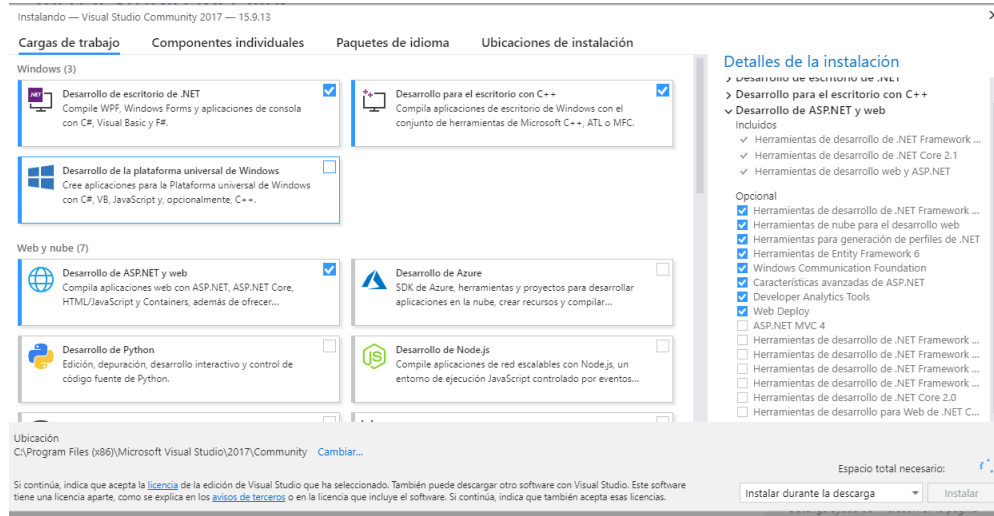


Figura 19 Características de Visual Studio

Visual Studio Installer

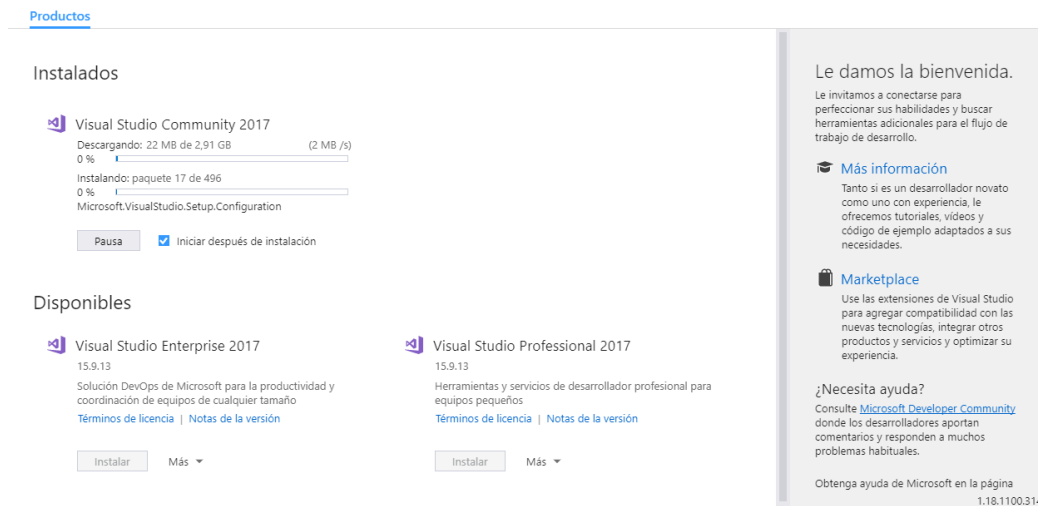


Figura 20 Proceso de carga de instalación

Anexo D Manual de Usuario



CARRERA DE ANÁLISIS DE SISTEMAS

**ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL
SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE
SEGUROS DE LA EMPRESA ITSM ECUADOR, PERÍODO
OCTUBRE 2018 – MARZO 2019.**

MANUAL DE USUARIO

**AUTOR: CACUANGO CATUCUAMBA JESSICA ESTEFANIA
QUITO 2018 - 2019**

ÍNDICE

Manual de Usuario.....	89
1.01. Introducción.....	89
1.02. Objetivos.....	89
Instrucciones Manual de Usuario	89

Índice de Figuras

Figura 1 Ingreso al Sistema	88
Figura 2 Menú principal	89
Figura 3 Registro para cargos	89
Figura 4 Registro para usuarios	90
Figura 5 Registro para empresa	90
Figura 6 Pantalla de registro de Proyectos.....	91
Figura 7 Pantalla de asignación de proyectos y personas	91

Manual de Usuario

1.04.Introducción

El manual de usuario se realiza con la finalidad de orientar a los usuarios en el manejo del sistema, guiándolos por cada uno de los módulos que contiene y sus diferentes funciones para entienda el proceso requerido.

1.05.Objetivos

Proporcionar al usuario toda la información necesaria para el manejo adecuando del sistema

Instrucciones Manual de Usuario

2.01.Módulo de Seguridad

Se digita el usuario y contraseña para el ingreso al sistema



The screenshot shows a web application interface for 'ITSMCONTROL'. At the top, there is a dark header with the text 'ITSMCONTROL' in white. Below the header, the word 'LOGIN' is centered in a light gray font. In the center of the page, there is a white rectangular box containing three input fields: 'Nombre Usuario', 'Contraseña', and a button labeled 'Ingresar'.

Figura 1 *Ingreso al Sistema*

Se va contar con 3 tipos de cargos para los usuarios

- Administrador
- Control de Calidad
- Programador

Dependiendo de su cargo nos va permitir en las diferentes opciones del menú

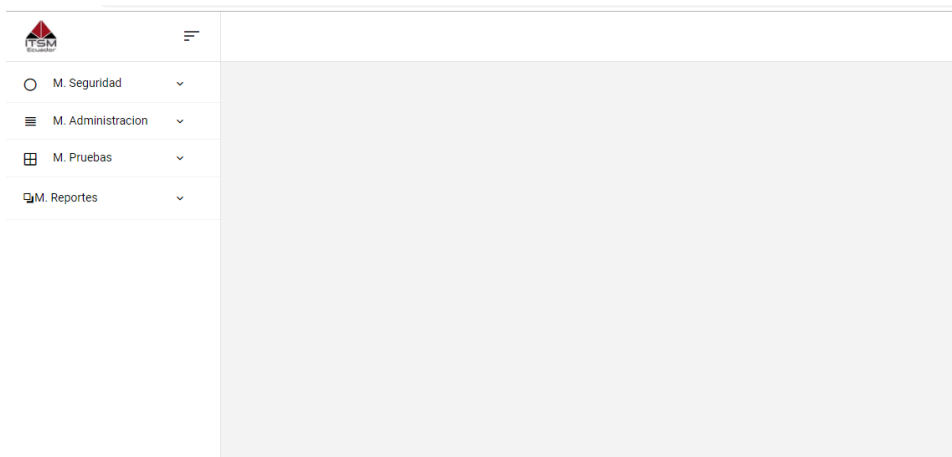


Figura 2 Menú principal

En el módulo de Seguridad se tendrá acceso para la creación de cargos, empresas y pruebas

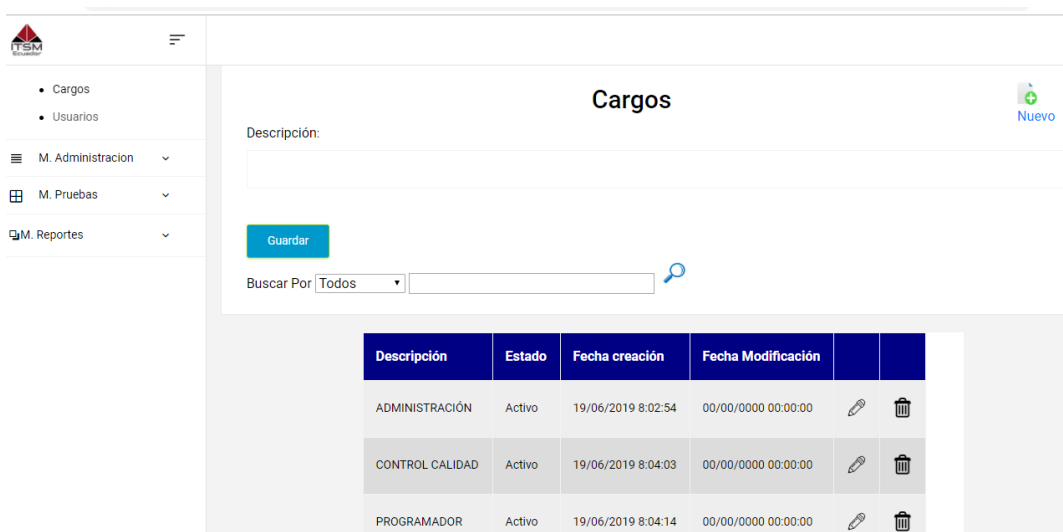
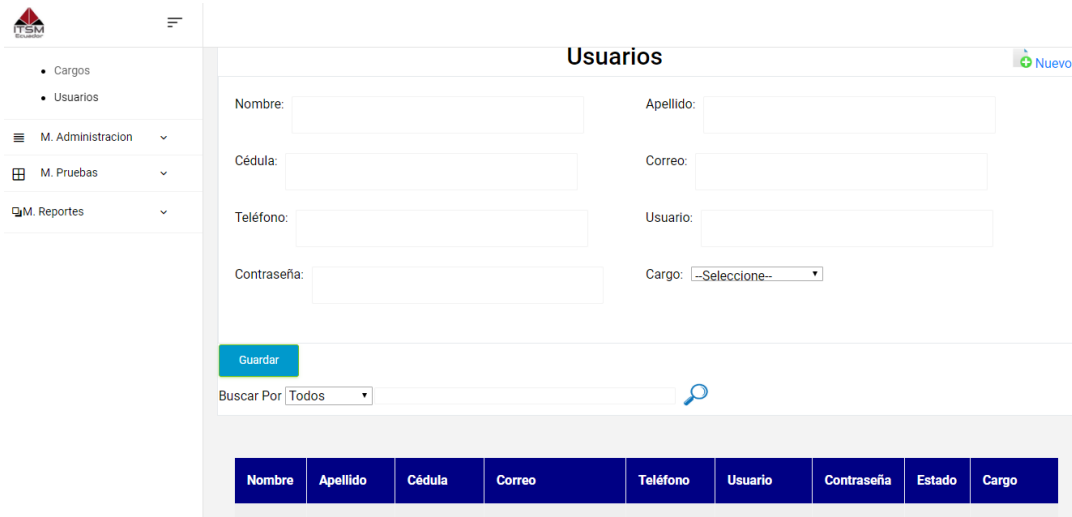


Figura 3 Registro para cargos

La pantalla de Usuario nos va permitir ingresar datos de la persona que va manipular el sistema



Usuarios Nuevo

Nombre: Apellido:

Cédula: Correo:

Teléfono: Usuario:

Contraseña: Cargo:

Guardar

Buscar Por:

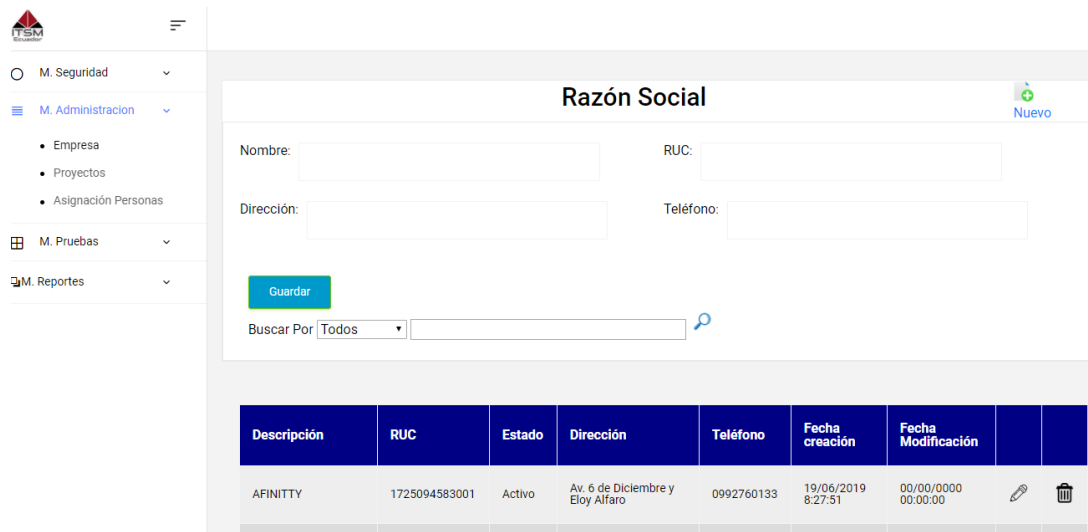
Nombre	Apellido	Cédula	Correo	Teléfono	Usuario	Contraseña	Estado	Cargo
--------	----------	--------	--------	----------	---------	------------	--------	-------

Figura 4 Registro para usuarios

2.02. Módulo de Administración

Dentro de este módulo se podrá registrar el proyecto, la asignación de personas al proyecto, la empresa.

La pantalla de empresas nos permite registrar los datos



Razón Social Nuevo

Nombre: RUC:

Dirección: Teléfono:

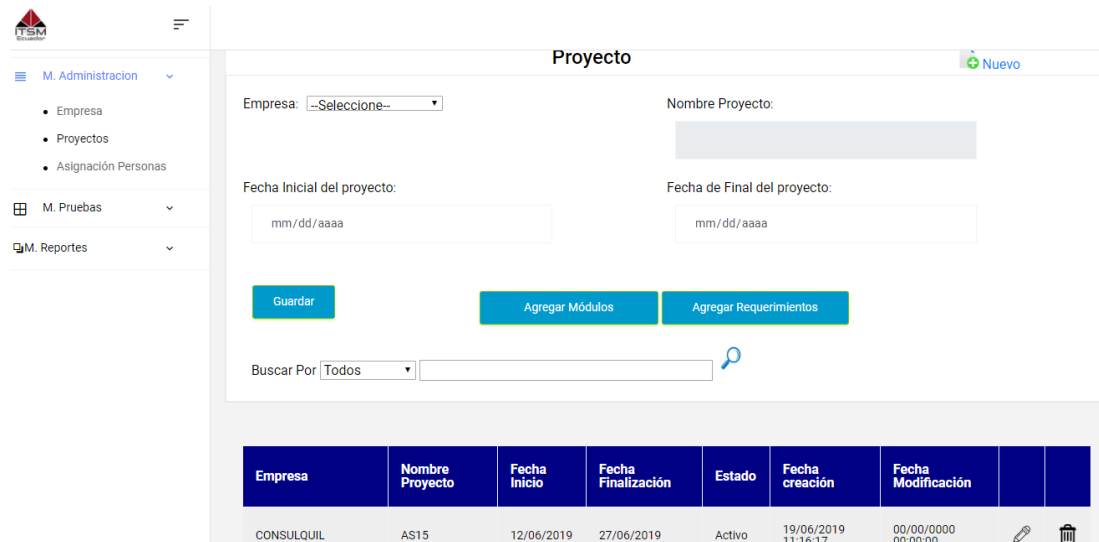
Guardar

Buscar Por:

Descripción	RUC	Estado	Dirección	Teléfono	Fecha creación	Fecha Modificación		
AFINITTY	1725094583001	Activo	Av. 6 de Diciembre y Eloy Alfaro	0992760133	19/06/2019 8:27:51	00/00/0000 00:00:00		

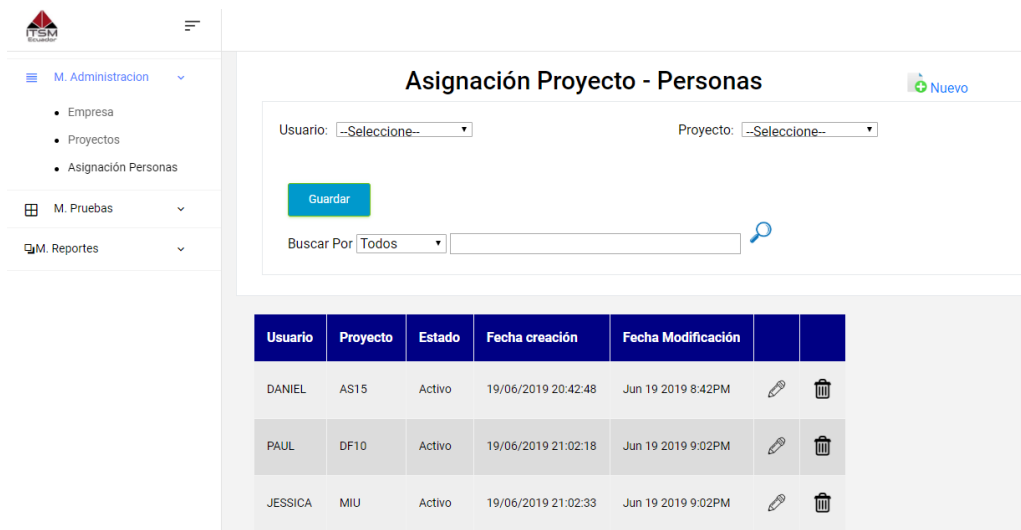
Figura 5 Registro para empresa

En esta pantalla podremos crear un proyecto agregar módulos y requerimientos para analizar.



Empresa	Nombre Proyecto	Fecha Inicio	Fecha Finalización	Estado	Fecha creación	Fecha Modificación		
CONSULQUIL	AS15	12/06/2019	27/06/2019	Activo	19/06/2019 11:16:17	00/00/0000 00:00:00		

Figura 6 Pantalla de registro de Proyectos



Usuario	Proyecto	Estado	Fecha creación	Fecha Modificación		
DANIEL	AS15	Activo	19/06/2019 20:42:48	Jun 19 2019 8:42PM		
PAUL	DF10	Activo	19/06/2019 21:02:18	Jun 19 2019 9:02PM		
JESSICA	MIU	Activo	19/06/2019 21:02:33	Jun 19 2019 9:02PM		

Figura 7 Pantalla de asignación de proyectos y personas

Test de pruebas para registrar

Test GUI



Prueba GUI [Nuevo](#)

Proyecto: Requerimiento:

Módulo: Responsable:

Objetivo: Observaciones:

Errores: Estado:

[Guardar](#)

Buscar Por: [🔍](#)

Fecha Creación	Requerimiento	Módulo	Nombre	Módulo	Objetivo	Observaciones	Errores	Estado	Estado Prueba
----------------	---------------	--------	--------	--------	----------	---------------	---------	--------	---------------

Test Seguridad



Prueba Seguridad [Nuevo](#)

Proyecto: Requerimiento:

Módulo: Responsable:

Objetivo: URL:

Observaciones: Errores:

Estado:

[Guardar](#)

Buscar Por: [🔍](#)

Fecha	Requerimiento	Módulo	Nombre	Módulo	Objetivo	Observaciones	Errores	Observación
-------	---------------	--------	--------	--------	----------	---------------	---------	-------------

Test Validación



- M. Administración
- M. Pruebas
 - Test GUI
 - Test Seguridad
 - Test Validación
 - Test Aceptación
- M. Reportes

Prueba Validación

[Nuevo](#)

Proyecto:

Requerimiento:

Módulo:

Responsable:

Propósito:

Descripción:

Procedimiento:

Resultados:

Imagen: Ningún ...ionado


Errores:

Resultado Fin:

Resolución:

Estado:

Test Aceptación



- M. Administración
- M. Pruebas
 - Test GUI
 - Test Seguridad
 - Test Validación
 - Test Aceptación
- M. Reportes

Prueba Aceptación

[Nuevo](#)

Proyecto:

Requerimiento:

Módulo:

Responsable:

Funcionalidad:

Elemento a probar:

Resultado:

Estado:

[Guardar](#)

Buscar Por: [🔍](#)

Fecha Creación	Requerimiento	Modulo	Nombre	Módulo	Objetivo	Observaciones	Errores	Estado	Estado Prueba
----------------	---------------	--------	--------	--------	----------	---------------	---------	--------	---------------

Anexo E Manual Técnico



CARRERA DE ANÁLISIS DE SISTEMAS

**ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL
SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE
SEGUROS DE LA EMPRESA ITSM ECUADOR, PERÍODO
OCTUBRE 2018 – MARZO 2019.**

MANUAL TÉCNICO

AUTOR: CACUANGO CATUCUAMBA JESSICA ESTEFANIA

QUITO 2018 - 2019

Índice

Manual Técnico	97
1.01. Introducción.....	97
1.02. Objetivos.....	97
Contenido.....	97
2.01. Script Base de datos.....	97
2.02. Código del Sistema ITSM- CONTROL	113
2.03 Diccionario de Datos.....	151

Manual Técnico

1.06.Introducción

El manual dirigido a personas con conocimientos técnicos, con el fin de proporcionar el código de la base de datos y el código de aplicativo.

1.07.Objetivos

Mostrar funcionalidad técnica del diseño, estructura del sistema.

Contenido

2.03. Script Base de datos

```
/****** Object: Database [ITSM_Control]   Script Date: 6/20/2019 11:28:08 AM  
*****/
```

```
CREATE DATABASE [ITSM_Control]  
GO  
ALTER DATABASE [ITSM_Control] SET COMPATIBILITY_LEVEL = 130  
GO  
IF (1 = FULLTEXTSERVICEPROPERTY('IsFullTextInstalled'))  
begin  
EXEC [ITSM_Control].[dbo].[sp_fulltext_database] @action = 'enable'  
end  
GO  
ALTER DATABASE [ITSM_Control] SET ANSI_NULL_DEFAULT OFF  
GO  
ALTER DATABASE [ITSM_Control] SET ANSI_NULLS OFF  
GO  
ALTER DATABASE [ITSM_Control] SET ANSI_PADDING OFF  
GO  
ALTER DATABASE [ITSM_Control] SET ANSI_WARNINGS OFF  
GO  
ALTER DATABASE [ITSM_Control] SET ARITHABORT OFF  
GO  
ALTER DATABASE [ITSM_Control] SET AUTO_CLOSE OFF  
GO  
ALTER DATABASE [ITSM_Control] SET AUTO_SHRINK OFF  
GO  
ALTER DATABASE [ITSM_Control] SET AUTO_UPDATE_STATISTICS ON  
GO  
ALTER DATABASE [ITSM_Control] SET CURSOR_CLOSE_ON_COMMIT OFF  
GO  
ALTER DATABASE [ITSM_Control] SET CURSOR_DEFAULT GLOBAL
```

```
GO
ALTER DATABASE [ITSM_Control] SET CONCAT_NULL_YIELDS_NULL
OFF
GO
ALTER DATABASE [ITSM_Control] SET NUMERIC_ROUNDABORT OFF
GO
ALTER DATABASE [ITSM_Control] SET QUOTED_IDENTIFIER OFF
GO
ALTER DATABASE [ITSM_Control] SET RECURSIVE_TRIGGERS OFF
GO
ALTER DATABASE [ITSM_Control] SET DISABLE_BROKER
GO
ALTER          DATABASE          [ITSM_Control]          SET
AUTO_UPDATE_STATISTICS_ASYNC OFF
GO
ALTER          DATABASE          [ITSM_Control]          SET
DATE_CORRELATION_OPTIMIZATION OFF
GO
ALTER DATABASE [ITSM_Control] SET TRUSTWORTHY OFF
GO
ALTER DATABASE [ITSM_Control] SET ALLOW_SNAPSHOT_ISOLATION
OFF
GO
ALTER DATABASE [ITSM_Control] SET PARAMETERIZATION SIMPLE
GO
ALTER DATABASE [ITSM_Control] SET READ_COMMITTED_SNAPSHOT
OFF
GO
ALTER DATABASE [ITSM_Control] SET HONOR_BROKER_PRIORITY OFF
GO
ALTER DATABASE [ITSM_Control] SET RECOVERY SIMPLE
GO
ALTER DATABASE [ITSM_Control] SET MULTI_USER
GO
ALTER DATABASE [ITSM_Control] SET PAGE_VERIFY CHECKSUM
GO
ALTER DATABASE [ITSM_Control] SET DB_CHAINING OFF
GO
ALTER          DATABASE          [ITSM_Control]          SET          FILESTREAM(
NON_TRANSACTED_ACCESS = OFF )
GO
ALTER DATABASE [ITSM_Control] SET TARGET_RECOVERY_TIME = 60
SECONDS
GO
ALTER DATABASE [ITSM_Control] SET DELAYED_DURABILITY =
DISABLED
GO
ALTER DATABASE [ITSM_Control] SET QUERY_STORE = OFF
```

```
GO
USE [ITSM_Control]
GO
ALTER DATABASE SCOPED CONFIGURATION SET
LEGACY_CARDINALITY_ESTIMATION = OFF;
GO
ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET
LEGACY_CARDINALITY_ESTIMATION = PRIMARY;
GO
ALTER DATABASE SCOPED CONFIGURATION SET MAXDOP = 0;
GO
ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET
MAXDOP = PRIMARY;
GO
ALTER DATABASE SCOPED CONFIGURATION SET
PARAMETER_SNIFFING = ON;
GO
ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET
PARAMETER_SNIFFING = PRIMARY;
GO
ALTER DATABASE SCOPED CONFIGURATION SET
QUERY_OPTIMIZER_HOTFIXES = OFF;
GO
ALTER DATABASE SCOPED CONFIGURATION FOR SECONDARY SET
QUERY_OPTIMIZER_HOTFIXES = PRIMARY;
GO
USE [ITSM_Control]
GO
/***** Object: Table [dbo].[con_asipersonas]    Script Date: 6/20/2019 11:28:08
AM *****/
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Estructura de la tabla “con_asipersonas”

```
CREATE TABLE [dbo].[con_asipersonas](
    [asip_id] [int] IDENTITY(1,1) NOT NULL,
    [asip_usuario] [int] NOT NULL,
    [asip_proyecto] [int] NOT NULL,
    [asip_estado] [nchar](10) NOT NULL,
    [asip_modificacion] [nchar](150) NOT NULL,
    [asip_creacion] [datetime] NOT NULL,
    CONSTRAINT [PK_con_asipersonas] PRIMARY KEY CLUSTERED
(
    [asip_id] ASC
)
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]  
GO  
/***** Object: Table [dbo].[con_cargos] Script Date: 6/20/2019 11:28:09 AM  
*****/  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO
```

Estructura de la tabla “con_cargos”

```
CREATE TABLE [dbo].[con_cargos](  
    [car_id] [int] IDENTITY(1,1) NOT NULL,  
    [car_descripcion] [nchar](20) NOT NULL,  
    [car_estado] [nchar](10) NOT NULL,  
    [car_modificar] [varchar](150) NOT NULL,  
    [car_creacion] [datetime] NOT NULL,  
    CONSTRAINT [PK_con_cargos] PRIMARY KEY CLUSTERED  
    (  
        [car_id] ASC  
    )WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
    IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
    ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]  
GO  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO
```

Estructura de la tabla “con_detalle”

```
CREATE TABLE [dbo].[con_detalle](  
    [det_id] [int] IDENTITY(1,1) NOT NULL,  
    [det_proposito] [nchar](200) NULL,  
    [det_descripcion] [nchar](200) NULL,  
    [det_procedimiento] [nchar](200) NULL,  
    [det_resultado] [nchar](200) NULL,  
    [det_imagen] [image] NULL,  
    [det_errores] [nchar](300) NULL,  
    [det_resulfinal] [nchar](100) NULL,  
    [det_resolucion] [nchar](200) NULL,  
    [det_fecha] [datetime] NOT NULL,  
    [det_realizado] [nchar](50) NULL,
```

```
[det_nombre] [nchar](100) NULL,  
[det_modulo] [int] NULL,  
[det_objetivo] [nchar](200) NULL,  
[det_observaciones] [nchar](200) NULL,  
[det_url] [nchar](100) NULL,  
[det_funcionalidad] [nchar](200) NULL,  
[det_etapa] [nchar](200) NULL,  
[det_requerimiento] [int] NOT NULL,  
[det_elemento] [nchar](100) NULL,  
[det_estadopro] [nchar](20) NULL,  
[det_estado] [nchar](10) NULL,  
[det_observacionpro] [nchar](200) NULL,  
[det_modificacion] [nchar](150) NOT NULL,  
[det_proyecto] [int] NOT NULL,  
[det_prueba] [int] NOT NULL,  
[det_personas] [int] NOT NULL,  
CONSTRAINT [PK_con_detalle] PRIMARY KEY CLUSTERED  
(  
    [det_id] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY] TEXTIMAGE_ON [PRIMARY]  
GO  
SET ANSI_NULLS ON  
GO  
SET QUOTED_IDENTIFIER ON  
GO
```

Estructura de la tabla “con_empresa”

```
CREATE TABLE [dbo].[con_empresa](  
    [emp_id] [int] IDENTITY(1,1) NOT NULL,  
    [emp_descripcion] [nchar](30) NOT NULL,  
    [emp_ruc] [nchar](13) NOT NULL,  
    [emp_estado] [nchar](10) NOT NULL,  
    [emp_modificacion] [varchar](150) NOT NULL,  
    [emp_creacion] [datetime] NOT NULL,  
    [emp_direccion] [nchar](100) NOT NULL,  
    [emp_telefono] [nchar](10) NOT NULL,  
    CONSTRAINT [PK_con_empresa] PRIMARY KEY CLUSTERED  
(  
    [emp_id] ASC  
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]  
GO
```

```
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Estructura de la tabla “con_modulos”

```
CREATE TABLE [dbo].[con_modulos](
    [modu_id] [int] IDENTITY(1,1) NOT NULL,
    [modu_descripcion] [varchar](100) NOT NULL,
    [modu_estado] [nchar](10) NOT NULL,
    [modu_proyecto] [int] NOT NULL,
    CONSTRAINT [PK_con_modulos] PRIMARY KEY CLUSTERED
(
    [modu_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Estructura de la tabla “con_proyecto”

```
CREATE TABLE [dbo].[con_proyecto](
    [pro_id] [int] IDENTITY(1,1) NOT NULL,
    [pro_empresa] [int] NOT NULL,
    [pro_nombrepro] [nchar](100) NOT NULL,
    [pro_fechaini] [date] NOT NULL,
    [pro_fechafin] [date] NOT NULL,
    [pro_estado] [nchar](10) NOT NULL,
    [pro_modificacion] [nchar](150) NOT NULL,
    [pro_creacion] [datetime] NOT NULL,
    CONSTRAINT [PK_con_proyecto] PRIMARY KEY CLUSTERED
(
    [pro_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Estructura de la tabla “con_pruebas”

```
CREATE TABLE [dbo].[con_pruebas](
    [cap_id] [int] IDENTITY(1,1) NOT NULL,
    [cap_nombre] [nchar](100) NOT NULL,
    [cap_descripcion] [nchar](100) NOT NULL,
    [cap_estado] [nchar](10) NOT NULL,
    [cap_creacion] [datetime] NOT NULL,
    CONSTRAINT [PK_con_cabpruebas] PRIMARY KEY CLUSTERED
(
    [cap_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Estructura de la tabla “con_requerimientos”

```
CREATE TABLE [dbo].[con_requerimientos](
    [req_id] [int] IDENTITY(1,1) NOT NULL,
    [req_numeracion] [nchar](10) NOT NULL,
    [req_descripcion] [nchar](200) NOT NULL,
    [req_proyecto] [int] NOT NULL,
    [req_modificacion] [nchar](150) NULL,
    [req_creacion] [datetime] NULL,
    [req_estado] [nchar](10) NOT NULL,
    CONSTRAINT [PK_con_requerimientos] PRIMARY KEY CLUSTERED
(
    [req_id] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]
) ON [PRIMARY]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Estructura de la tabla “con_usuario”

```
CREATE TABLE [dbo].[con_usuario](
```

```
[usu_id] [int] IDENTITY(1,1) NOT NULL,  
[usu_nombre] [nvarchar](50) NOT NULL,  
[usu_apellido] [nvarchar](50) NOT NULL,  
[usu_cedula] [nvarchar](13) NOT NULL,  
[usu_correo] [nvarchar](50) NOT NULL,  
[usu_telefono] [nvarchar](10) NOT NULL,  
[usu_usuario] [nvarchar](30) NOT NULL,  
[usu_contraseña] [nvarchar](20) NOT NULL,  
[usu_estado] [nvarchar](10) NOT NULL,  
[usu_cargo] [int] NOT NULL,  
[usu_modificar] [nvarchar](150) NOT NULL,  
[usu_creacion] [datetime] NOT NULL,  
  
CONSTRAINT [PK_con_usuario] PRIMARY KEY CLUSTERED  
(  
    [usu_id] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
IGNORE_DUP_KEY = OFF, ALLOW_ROW_LOCKS = ON,  
ALLOW_PAGE_LOCKS = ON) ON [PRIMARY]  
) ON [PRIMARY]  
GO  
SET IDENTITY_INSERT [dbo].[con_asipersonas] ON
```

Asignación de campos únicos

```
CREATE UNIQUE NONCLUSTERED INDEX [IX_con_asipersonas] ON  
[dbo].[con_asipersonas]  
(  
    [asip_usuario] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,  
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON  
[PRIMARY]  
GO  
SET ANSI_PADDING ON  
GO  
/***** Object: Index [IX_con_cargos]    Script Date: 6/20/2019 11:28:09 AM  
*****/  
CREATE UNIQUE NONCLUSTERED INDEX [IX_con_cargos] ON  
[dbo].[con_cargos]  
(  
    [car_descripcion] ASC  
) WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,  
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON  
[PRIMARY]
```



```
GO
SET ANSI_PADDING ON
GO
/***** Object: Index [IX_con_empresa]    Script Date: 6/20/2019 11:28:09 AM
*****/
CREATE UNIQUE NONCLUSTERED INDEX [IX_con_empresa] ON
[dbo].[con_empresa]
(
    [emp_descripcion] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
SET ANSI_PADDING ON
GO
/***** Object: Index [IX_con_empresa_1]    Script Date: 6/20/2019 11:28:09 AM
*****/
CREATE UNIQUE NONCLUSTERED INDEX [IX_con_empresa_1] ON
[dbo].[con_empresa]
(
    [emp_ruc] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
SET ANSI_PADDING ON
GO
/***** Object: Index [IX_con_usuario]    Script Date: 6/20/2019 11:28:09 AM
*****/
CREATE UNIQUE NONCLUSTERED INDEX [IX_con_usuario] ON
[dbo].[con_usuario]
(
    [usu_cedula] ASC
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON
[PRIMARY]
GO
SET ANSI_PADDING ON
GO
/***** Object: Index [IX_con_usuario_1]    Script Date: 6/20/2019 11:28:09 AM
*****/
CREATE UNIQUE NONCLUSTERED INDEX [IX_con_usuario_1] ON
[dbo].[con_usuario]
(
    [usu_usuario] ASC
```

```
)WITH (PAD_INDEX = OFF, STATISTICS_NORECOMPUTE = OFF,  
SORT_IN_TEMPDB = OFF, IGNORE_DUP_KEY = OFF, DROP_EXISTING = OFF,  
ONLINE = OFF, ALLOW_ROW_LOCKS = ON, ALLOW_PAGE_LOCKS = ON) ON  
[PRIMARY]  
GO
```

Filtros para la tabla con "asipersonas"

```
ALTER TABLE [dbo].[con_asipersonas] WITH CHECK ADD CONSTRAINT  
[FK_con_asipersonas_con_proyecto] FOREIGN KEY([asip_proyecto])  
REFERENCES [dbo].[con_proyecto] ([pro_id])  
GO
```

Filtros para la tabla "con_asipersonas"

```
ALTER TABLE [dbo].[con_asipersonas] CHECK CONSTRAINT  
[FK_con_asipersonas_con_proyecto]  
GO  
ALTER TABLE [dbo].[con_asipersonas] WITH CHECK ADD CONSTRAINT  
[FK_con_asipersonas_con_usuario] FOREIGN KEY([asip_usuario])  
REFERENCES [dbo].[con_usuario] ([usu_id])  
GO  
ALTER TABLE [dbo].[con_asipersonas] CHECK CONSTRAINT  
[FK_con_asipersonas_con_usuario]  
GO
```

Filtros para la tabla "con_detalle"

```
ALTER TABLE [dbo].[con_detalle] WITH CHECK ADD CONSTRAINT  
[FK_con_detalle_con_modulos] FOREIGN KEY([det_modulo])  
REFERENCES [dbo].[con_modulos] ([modu_id])  
GO  
ALTER TABLE [dbo].[con_detalle] CHECK CONSTRAINT  
[FK_con_detalle_con_modulos]  
GO  
ALTER TABLE [dbo].[con_detalle] WITH CHECK ADD CONSTRAINT  
[FK_con_detalle_con_proyecto] FOREIGN KEY([det_proyecto])  
REFERENCES [dbo].[con_proyecto] ([pro_id])  
GO  
ALTER TABLE [dbo].[con_detalle] CHECK CONSTRAINT  
[FK_con_detalle_con_proyecto]  
GO  
ALTER TABLE [dbo].[con_detalle] WITH CHECK ADD CONSTRAINT  
[FK_con_detalle_con_pruebas] FOREIGN KEY([det_prueba])  
REFERENCES [dbo].[con_pruebas] ([cap_id])  
GO  
ALTER TABLE [dbo].[con_detalle] CHECK CONSTRAINT  
[FK_con_detalle_con_pruebas]
```

```
GO
ALTER TABLE [dbo].[con_detalle] WITH CHECK ADD CONSTRAINT
[FK_con_detalle_con_requerimientos] FOREIGN KEY([det_requerimiento])
REFERENCES [dbo].[con_requerimientos] ([req_id])
GO
ALTER TABLE [dbo].[con_detalle] CHECK CONSTRAINT
[FK_con_detalle_con_requerimientos]
GO
```

Filtros para la tabla “con_modulos”

```
ALTER TABLE [dbo].[con_modulos] WITH CHECK ADD CONSTRAINT
[FK_con_modulos_con_proyecto] FOREIGN KEY([modu_proyecto])
REFERENCES [dbo].[con_proyecto] ([pro_id])
GO
ALTER TABLE [dbo].[con_modulos] CHECK CONSTRAINT
[FK_con_modulos_con_proyecto]
GO
```

Filtros para la tabla “con_proyecto”

```
ALTER TABLE [dbo].[con_proyecto] WITH CHECK ADD CONSTRAINT
[FK_con_proyecto_con_empresa] FOREIGN KEY([pro_empresa])
REFERENCES [dbo].[con_empresa] ([emp_id])
GO
ALTER TABLE [dbo].[con_proyecto] CHECK CONSTRAINT
[FK_con_proyecto_con_empresa]
GO
```

Filtros para la tabla “con_requerimiento”

```
ALTER TABLE [dbo].[con_requerimientos] WITH CHECK ADD CONSTRAINT
[FK_con_requerimientos_con_proyecto] FOREIGN KEY([req_proyecto])
REFERENCES [dbo].[con_proyecto] ([pro_id])
GO
ALTER TABLE [dbo].[con_requerimientos] CHECK CONSTRAINT
[FK_con_requerimientos_con_proyecto]
GO
```

Filtros para la tabla “con_usuario”

```
ALTER TABLE [dbo].[con_usuario] WITH CHECK ADD CONSTRAINT
[FK_con_usuario_con_cargos] FOREIGN KEY([usu_cargo])
REFERENCES [dbo].[con_cargos] ([car_id])
GO
```

```
ALTER TABLE [dbo].[con_usuario] CHECK CONSTRAINT
[FK_con_usuario_con_cargos]
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Procedimiento almacenado Cargos

```
CREATE PROCEDURE [dbo].[updateCargos]
(@id int, @descripcion nchar (20), @estado nchar(10), @modificacion
varchar(150))

AS
BEGIN
SET NOCOUNT ON;

UPDATE [dbo].[con_cargos]
SET car_descripcion = @descripcion
,car_estado = @estado
,car_modificar = @modificacion
where car_id=@id
END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Procedimiento almacenado Detalles

```
CREATE PROCEDURE [dbo].[updateDetalle]
(@id int, @proposito nchar (200), @descripcion nchar (200),@procedimiento
nchar (200),
@resultado nchar (200), @imagen image, @errores nchar (300), @resulfinal nchar
(100),@resolucion nchar (200),
@fecha Datetime, @realizado nchar (50), @nombre nchar (100), @modulo
int,@objetivo nchar (200),
@observaciones nchar (200), @url nchar (100), @funcionalidad nchar (200),
@etapa nchar (200),@requerimiento int,
@elemento nchar (100), @estadopro nchar(20), @estado nchar (10),
@observacionpro nchar (200), @modificacion nchar(150),
@proyecto int, @prueba int,@personas int )

AS
BEGIN
SET NOCOUNT ON;
```

```
UPDATE [dbo].[con_detalle]
SET
    det_proposito = @proposito
    ,det_descripcion = @descripcion
    ,det_procedimiento=@procedimiento
    ,det_resultado = @resultado
    ,det_imagen = @imagen
    ,det_errores = @errores
    ,det_resulfinal = @resulfinal
    ,det_resolucion=@resolucion
    ,det_fecha = @fecha
    ,det_realizado = @realizado
    ,det_nombre = @nombre
    ,det_modulo = @modulo
    ,det_objetivo=@objetivo
    ,det_observaciones = @observaciones
    ,det_url = @url
    ,det_funcionalidad = @funcionalidad
    ,det_etapa = @etapa
    ,det_requerimiento=@requerimiento
    ,det_elemento = @elemento
    ,det_estadopro = @estadopro
    ,det_estado = @estado
    ,det_observacionpro = @observacionpro
    ,det_modificacion =@modificacion
    where det_id=@id

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Procedimiento almacenado Empresa

```
CREATE PROCEDURE [dbo].[updateEmpresa]
    (@id int, @descripcion nchar (30), @ruc nchar (13), @estado
nchar(10),@modificar nchar(150), @direccion nchar(100),@telefono nchar(10))

AS
BEGIN

    SET NOCOUNT ON;

    UPDATE [dbo].[con_empresa]
    SET emp_descripcion = @descripcion
    ,emp_ruc = @ruc,
    emp_estado = @estado,
```

```
emp_modificacion=@modificar
,emp_direccion=@direccion
,emp_telefono=@telefono

where emp_id=@id

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Procedimiento almacenado Persona

```
CREATE PROCEDURE [dbo].[updatePersona]
(@id int, @usuario int, @proyecto int, @estado nchar(10), @modificar nchar(150))
AS
BEGIN
SET NOCOUNT ON;

UPDATE [dbo].[con_asipersonas]
SET asip_usuario = @usuario
,asip_proyecto = @proyecto,
asip_estado = @estado
,asip_modificacion =@modificar

where asip_id=@id

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Procedimiento Almacenado Proyecto

```
CREATE PROCEDURE [dbo].[updateProyecto]
(@id int, @empresa int, @nombrepro nchar (100), @fechaini datetime, @fechafin
datetime, @estado nchar (10), @modificacion nchar(150))

AS
BEGIN

SET NOCOUNT ON;
```

```
UPDATE [dbo].[con_proyecto]
SET pro_empresa = @empresa
  ,pro_nombrepro = @nombrepro
  ,pro_fechaini = @fechaini
  ,pro_fechafin = @fechafin
  ,pro_estado = @estado
  ,pro_modificacion= @modificacion

where pro_id=@id

END
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
```

Procedimiento Almacenado Usuario

```
CREATE PROCEDURE [dbo].[updateUsuarios]
  (@id int, @nombre nchar (50), @apellido nchar (50), @cedula nchar(13), @correo
nchar (50), @telefono nchar (10), @usuario nchar (30), @contrasenia nchar (20),
@estado nchar (10),@cargo int, @modificar nchar(150))

AS
BEGIN

SET NOCOUNT ON;

  UPDATE [dbo].[con_usuario]
SET usu_nombre = @nombre
  ,usu_apellido = @apellido
  ,usu_cedula = @cedula
  ,usu_correo = @correo
  ,usu_telefono = @telefono
  ,usu_usuario = @usuario
  ,usu_contrasenia = @contrasenia
  ,usu_estado = @estado
  ,usu_cargo = @cargo
  ,usu_modificar= @modificar

  where usu_id=@id

END
GO
USE [master]
GO
```

```
ALTER DATABASE [ITSM_Control] SET READ_WRITE  
GO
```

2.04.Código del Sistema ITSM- CONTROL

Módulo de Seguridad

Cargo

Controlador

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Web;  
using System.Web.UI;  
using System.Web.UI.WebControls;  
using Controlador;  
using AccesoDatos;  
  
namespace ITSM_Control.Administracion  
{  
    public partial class AdmCargo : System.Web.UI.Page  
    {  
        protected void Page_Load(object sender, EventArgs e)  
        {  
            if (!IsPostBack)  
            {  
                if (Request["cod"] != null)  
                {  
                    string codigo = Convert.ToString(Request["cod"]);  
                    if (!string.IsNullOrEmpty(codigo))  
                    {  
                        loadTipoR(codigo);  
                    }  
                }  
                cargarTipoReserva();  
            }  
        }  
  
        protected void btnAgregar_Click(object sender, EventArgs e)  
        {  
            if (Convert.ToString(Request["cod"]) != null)  
            {
```



```
        GuardarDatos(int.Parse(Convert.ToString(Request["cod"])));
    }
    else
    {
        GuardarDatos(0);
    }
}
private void loadTipoR(string codigo)
{
    con_cargos _infoTipoR = new con_cargos();
    _infoTipoR = logicaCargos.obtenerCargosXID(int.Parse(codigo));

    if (_infoTipoR != null)
    {
        lblCodigo.Text = _infoTipoR.car_id.ToString();
        txtDescripcion.Text = _infoTipoR.car_descripcion;
    }
}

private void GuardarDatos(int codigo)
{
    if (codigo > 0)
    {
        update();
    }
    else
    {
        Save();
    }
}

private void update()
{
    if (Request["cod"] != null)
    {
        string codigo = Convert.ToString(Request["cod"]);
        if (!string.IsNullOrEmpty(codigo))
        {
            try
            {
                con_cargos _infoRepresentante = new con_cargos();
                _infoRepresentante.car_id = int.Parse(lblCodigo.Text);
                _infoRepresentante.car_descripcion = txtDescripcion.Text.ToUpper();
                logicaCargos.updateCargos(_infoRepresentante);
                Response.Redirect("/Administracion/AdmCargo.aspx");
            }
            catch (Exception ex)
            {
            }
        }
    }
}
```

```
        {
            lblMensaje.Text = ex.Message;
        }
    }
}

private void Save()
{
    try
    {
        var messageError = "";
        if (string.IsNullOrEmpty(txtDescripcion.Text))
        {
            messageError += "Nombre campo obligatorio \n";
        }

        if (!string.IsNullOrEmpty(messageError))
        {
            lblMensaje.Text = messageError;
            return;
        }

        con_cargos inst = new con_cargos();
        inst.car_descripcion = txtDescripcion.Text.ToUpper();

        logicaCargos.saveCargos(inst);
        lblMensaje.Text = "Registro Guardado Correctamente";
        Response.Redirect("/Administracion/AdmCargo.aspx");

        //lblMensaje.Text = "Registro Repetido";

    }
    catch (Exception ex)
    {
        lblMensaje.Text = ex.Message;
    }
}

//datos formulario
private void cargarTipoReserva()
{
    List<con_cargos> listaUsuario = new List<con_cargos>();
```

```
listaUsuario = logicaCargos.obtenerCargos();
if (listaUsuario.Count > 0 && listaUsuario != null)
{
    GdvInstituciones.DataSource = listaUsuario;
    GdvInstituciones.DataBind();
}
}
protected void GdvInstituciones_RowCommand(object sender,
GridViewCommandEventArgs e)
{
    string codigo = Convert.ToString(e.CommandArgument);

    if (e.CommandName == "Modificar")
    {
        //string codigoen =
FormsAuthentication.HashPasswordForStoringInConfigFile(codigo, "MD5");
        //codigo = Encriptar(codigo, llave);
        codigo = codigo;
        Response.Redirect("/Administracion/AdmCargo.aspx?cod=" + codigo,
true);
    }
    if (e.CommandName == "Eliminar")
    {
        if (!string.IsNullOrEmpty(codigo))
        {
            con_cargos _infoUsuario = new con_cargos();
            _infoUsuario = logicaCargos.obtenerCargosXID(int.Parse(codigo));
            if (_infoUsuario != null)
            {
                if (logicaCargos.deleteCargo(_infoUsuario))
                {
                    Response.Write("<script>window.alert('Registro
Borrado');</script>");

                    //Aqui borras
                }
                else
                {
                    Response.Write("<script>window.alert('El registro no se puede
eliminar');</script>");

                    //mensaje que nno se borro
                }
                cargarTipoReserva();
            }
        }
    }
}
```

```
    }  
    }  
}  
  
protected void GdvInstituciones_SelectedIndexChanged(object sender,  
EventArgs e)  
{  
  
}  
private void nuevo()  
{  
    Response.Redirect("/Administracion/AdmCargo.aspx");  
}  
protected void lnkNuevo_Click(object sender, EventArgs e)  
{  
    nuevo();  
}  
  
protected void imgNuevo_Click(object sender, ImageClickEventArgs e)  
{  
    nuevo();  
}  
  
protected void imgBuscar_Click(object sender, ImageClickEventArgs e)  
{  
    buscar(ddlBuscar.SelectedValue);  
}  
private void buscar(string op)  
{  
    switch (op)  
    {  
        case "T":  
            cargarTipoReserva();  
            break;  
        case "C":  
            GdvInstituciones.DataSource =  
logicaCargos.obtenerCargosxID(txtBuscar.Text);  
            GdvInstituciones.DataBind();  
            break;  
        case "N":  
            GdvInstituciones.DataSource =  
logicaCargos.obtenerCargosXNombre(txtBuscar.Text);  
            GdvInstituciones.DataBind();  
            break;  
    }  
}  
  
protected void btnRegresar_Click(object sender, EventArgs e)
```

```
        {  
            Response.Redirect("/Menu.aspx");  
        }  
    }  
}
```

Clase

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using AccesoDatos;  
  
namespace Controlador  
{  
    public class logicaCargos  
    {  
        private static DCTesisDataContext dc { get; set; }  
  
        public static List<con_cargos> obtenerCargos()  
        {  
            try  
            {  
                dc = new DCTesisDataContext();  
                var lista = dc.con_cargos.Where(usu => usu.car_estado ==  
"Activo").OrderBy(usu => usu.car_descripcion).ToList();  
                return lista;  
            }  
            catch (Exception ex)  
            {  
                throw new ArgumentException(ex.Message);  
            }  
        }  
  
        public static con_cargos obtenerCargosXID(int codigo)  
        {  
            try  
            {  
                dc = new DCTesisDataContext();  
                var rol = dc.con_cargos.FirstOrDefault(usu => usu.car_estado == "Activo"  
                    && usu.car_id.Equals(codigo));  
  
                return rol;  
            }  
            catch (Exception ex)  
            {  
                throw new ArgumentException(ex.Message);  
            }  
        }  
    }  
}
```

```
    }  
}  
  
public static List<con_cargos> obtenerCargosxID(string codigo)  
{  
    try  
    {  
        dc = new DCTesisDataContext();  
        var rol = dc.con_cargos.Where(usu => usu.car_estado == "Activo"  
                                     && usu.car_id.Equals(codigo));  
        return rol.ToList();  
    }  
    catch (Exception ex)  
    {  
        throw new ArgumentException(ex.Message);  
    }  
}  
  
public static List<con_cargos> obtenerCargosXNombre(string nombreRol)  
{  
    try  
    {  
        dc = new DCTesisDataContext();  
        var lista = dc.con_cargos.Where(usu => usu.car_estado == "Activo"  
                                       &&  
usu.car_descripcion.StartsWith(nombreRol)).OrderBy(usu  
usu.car_descripcion).ToList();  
        return lista;  
    }  
    catch (Exception ex)  
    {  
        throw new ArgumentException(ex.Message);  
    }  
}  
  
public static void saveCargos(con_cargos _infoRol)  
{  
    try  
    {  
        dc = new DCTesisDataContext();  
        _infoRol.car_estado = "Activo";  
        _infoRol.car_creacion = DateTime.Now;  
        _infoRol.car_modificar = "00/00/0000 00:00:00";  
  
        dc.con_cargos.InsertOnSubmit(_infoRol);  
        dc.SubmitChanges();  
    }  
}
```

//te amo muchisisisismo siempre vas ha tener mi apoyo en cualquier
circunstacia que estemos pasando

//me duele mucho verte y en este momento escucharte asi

```
    }  
    catch (Exception ex)  
    {  
        throw new ArgumentException(ex.Message);  
    }  
}  
  
public static bool deleteCargo(con_cargos _infoRol)  
{  
    try  
    {  
        int idRol = _infoRol.car_id;  
        logicaUsuarios logicausuario = new logicaUsuarios();  
  
        List<con_usuario> lista = logicaUsuarios.obtenerUsuariosXCargo(idRol);  
        if (lista.Count > 0)  
        {  
            return false;  
        }  
        else  
        {  
            //dc = new DCHelpDeskDataContext();  
            _infoRol.car_estado = "Inactivo";  
            // _infoRol.ROL_ADD = DateTime.Now;  
            // dc.TBL_DEPARTAMENTOS.InsertOnSubmit(_infoDepartamento);  
            dc.SubmitChanges();  
            return true;  
        }  
    }  
    catch (Exception ex)  
    {  
        throw new ArgumentException(ex.Message);  
    }  
}  
  
public static void updateCargos(con_cargos _infoRol)  
{  
    try  
    {  
        dc = new DCTesisDataContext();  
        _infoRol.car_estado = "Activo";  
        _infoRol.car_modificar = Convert.ToString(DateTime.Now);  
        dc.updateCargos(_infoRol.car_id, _infoRol.car_descripcion,  
            _infoRol.car_estado, _infoRol.car_modificar);  
    }  
    catch (Exception ex)  
    {  
        throw new ArgumentException(ex.Message);  
    }  
}
```

```
        dc.SubmitChanges();

    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}
}
```

Usuario

Controlador

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Controlador;
using AccesoDatos;

namespace ITSM_Control.Administracion
{
    public partial class AdmUsuario : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                if (Request["cod"] != null)
                {
                    string codigo = Convert.ToString(Request["cod"]);
                    if (!string.IsNullOrEmpty(codigo))
                    {
                        loadTipoR(codigo);
                    }
                }
                cargarUsuarios();
                cargarTUsuarios();
                //cargarEmpresa();
            }
        }
    }
}
```



```
    }  
}  
  
protected void btnAgregar_Click(object sender, EventArgs e)  
{  
    if (Convert.ToString(Request["cod"]) != null)  
    {  
        GuardarDatos(int.Parse(Convert.ToString(Request["cod"])));  
    }  
    else  
    {  
        GuardarDatos(0);  
    }  
}  
  
private void loadTipoR(string codigo)  
{  
    con_usuario _infoTipoR = new con_usuario();  
    _infoTipoR = logicaUsuarios.obtenerUsuariosXID(int.Parse(codigo));  
  
    if (_infoTipoR != null)  
    {  
        lblCodigo.Text = _infoTipoR.usu_id.ToString();  
        txtNombre.Text = _infoTipoR.usu_nombre;  
        txtApellido.Text = _infoTipoR.usu_apellido;  
        txtCedula.Text = _infoTipoR.usu_cedula;  
        txtCorreo.Text = _infoTipoR.usu_correo;  
        txtTelefono.Text = _infoTipoR.usu_telefono;  
        txtUsuario.Text = _infoTipoR.usu_usuario;  
        txtContrasenia.Text = _infoTipoR.usu_contrasenia;  
        ddlcarga.SelectedValue = _infoTipoR.usu_cargo.ToString();  
    }  
}  
  
private void GuardarDatos(int codigo)  
{  
    if (codigo > 0)  
    {  
        update();  
    }  
    else  
    {  
  
        Save();  
    }  
}
```

```
    }  
  
    }  
  
    private void nuevo()  
    {  
        Response.Redirect("/Administracion/AdmUsuario.aspx");  
    }  
    protected void InkNuevo_Click(object sender, EventArgs e)  
    {  
        nuevo();  
    }  
  
    protected void imgNuevo_Click(object sender, ImageClickEventArgs e)  
    {  
        nuevo();  
    }  
  
    private void update()  
    {  
        if (Request["cod"] != null)  
        {  
            string codigo = Convert.ToString(Request["cod"]);  
            if (!string.IsNullOrEmpty(codigo))  
            {  
                try  
                {  
                    con_usuario _infoRepresentante = new con_usuario();  
                    // _infoDepartamento =  
logicaDepartamento.obtenerDepartamentoID(int.Parse(codigo));  
                    _infoRepresentante.usu_id = int.Parse(lblCodigo.Text);  
                    _infoRepresentante.usu_nombre = txtNombre.Text.ToUpper();  
                    _infoRepresentante.usu_apellido = txtApellido.Text.ToUpper();  
                    _infoRepresentante.usu_cedula = txtCedula.Text;  
                    _infoRepresentante.usu_correo = txtCorreo.Text.ToLower();  
                    _infoRepresentante.usu_telefono = txtTelefono.Text.ToUpper();  
                    _infoRepresentante.usu_usuario = txtUsuario.Text.ToUpper();  
                    _infoRepresentante.usu_contrasenia = txtContrasenia.Text;  
                    _infoRepresentante.usu_cargo = int.Parse(ddlcargo.SelectedValue);  
                    logicaUsuarios.updateUsuarios(_infoRepresentante);  
                    Response.Redirect("/Administracion/AdmUsuario.aspx");  
                }  
                catch (Exception ex)  
                {  
                    lblMensaje.Text = ex.Message;  
                }  
            }  
        }  
    }  
}
```

```
}

private void Save()
{
    try
    {
        var messageError = "";
        if (string.IsNullOrEmpty(txtNombre.Text))
        {
            messageError += "Nombre campo obligatorio \n";
        }

        if (!string.IsNullOrEmpty(messageError))
        {
            lblMensaje.Text = messageError;
            return;
        }

        con_usuario inst = new con_usuario();
        inst.usu_nombre = txtNombre.Text.ToUpper();
        inst.usu_apellido = txtApellido.Text.ToUpper();
        inst.usu_cedula = txtCedula.Text;
        inst.usu_correo = txtCorreo.Text.ToLower();
        inst.usu_telefono = txtTelefono.Text.ToUpper();
        inst.usu_usuario = txtUsuario.Text.ToUpper();
        inst.usu_contrasenia = txtContrasenia.Text;
        inst.usu_cargo = int.Parse(ddlcargo.SelectedValue);

        logicaUsuarios.saveUsuarios(inst);
        lblMensaje.Text = "Registro Guardado Correctamente";
        Response.Redirect("/Administracion/AdmUsuario.aspx");
    }
    catch (Exception ex)
    {
        lblMensaje.Text = ex.Message;
    }
}

private void cargarUsuarios()
{
    List<UsuarioCargo> listaUsuario = new List<UsuarioCargo>();
    listaUsuario = logicaUsuarioReporte.ObtenerUsuarios();
    if (listaUsuario.Count > 0 && listaUsuario != null)
```

```
{
    GdvUsuarios.DataSource = listaUsuario;
    GdvUsuarios.DataBind();
}

protected void GdvUsuarios_RowCommand(object sender,
GridViewCommandEventArgs e)
{
    string codigo = Convert.ToString(e.CommandArgument);

    if (e.CommandName == "Modificar")
    {
        //string codigoen =
FormsAuthentication.HashPasswordForStoringInConfigFile(codigo, "MD5");
        //codigo = Encriptar(codigo, llave);
        codigo = codigo;
        Response.Redirect("/Administracion/AdmUsuario.aspx?cod=" + codigo,
true);
    }
    if (e.CommandName == "Eliminar")
    {
        if (!string.IsNullOrEmpty(codigo))
        {
            con_usuario _infoUsuario = new con_usuario();
            _infoUsuario = logicaUsuarios.obtenerUsuariosXID(int.Parse(codigo));
            if (_infoUsuario != null)
            {
                logicaUsuarios.deleteUsuarios(_infoUsuario);
                cargarUsuarios();
            }
        }
    }
}

private void cargarTUsuarios()
{
    List<con_cargos> _listaTh = new List<con_cargos>();
    _listaTh = logicaCargos.obtenerCargos();
    if (_listaTh.Count > 0 && _listaTh != null)
    {
        _listaTh.Insert(0, new con_cargos { car_id = 0, car_descripcion = "--
Seleccione--" });
        ddlcarga.DataSource = _listaTh;
        ddlcarga.DataTextField = "car_descripcion";
    }
}
```

```
        ddlcarga.DataValueField = "car_id";
        ddlcarga.DataBind();

    }
}
protected void imgBuscar_Click(object sender, ImageClickEventArgs e)
{
    buscar(ddlBuscar.SelectedValue);
}
private void buscar(string op)
{
    switch (op)
    {
        case "T":
            cargarUsuarios();
            break;
        case "C":
            GdvUsuarios.DataSource =
logicaUsuarios.obtenerUsuariosxID(txtBuscar.Text);
            GdvUsuarios.DataBind();
            break;
        case "N":
            GdvUsuarios.DataSource =
logicaUsuarioReporte.ObtenerUsuariosNombre(txtBuscar.Text);
            GdvUsuarios.DataBind();
            break;
    }
}

protected void txtApellido_TextChanged(object sender, EventArgs e)
{
}

protected void btnRegresar_Click(object sender, EventArgs e)
{
    Response.Redirect("Menu.aspx");
}
}
```

Clase

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
```

```
using AccesoDatos;

namespace Controlador
{
    public class logicaUsuarios
    {
        private static DCTesisDataContext dc { get; set; }

        public static List<con_usuario> obtenerUsuarios()
        {
            try
            {
                dc = new DCTesisDataContext();
                var lista = dc.con_usuario.Where(usu => usu.usu_estado ==
"Activo").OrderBy(usu => usu.usu_nombre).ToList();
                return lista;
            }
            catch (Exception ex)
            {
                throw new ArgumentException(ex.Message);
            }
        }

        public static con_usuario obtenerUsuariosXID(int codigo)
        {
            try
            {
                dc = new DCTesisDataContext();
                var rol = dc.con_usuario.FirstOrDefault(usu => usu.usu_estado ==
"Activo"
                                && usu.usu_id.Equals(codigo));

                return rol;
            }
            catch (Exception ex)
            {
                throw new ArgumentException(ex.Message);
            }
        }

        public static List<con_usuario> obtenerUsuariosxID(string codigo)
        {
            try
            {
                dc = new DCTesisDataContext();
                var rol = dc.con_usuario.Where(usu => usu.usu_estado == "Activo"
                                && usu.usu_id.Equals(codigo));

                return rol.ToList();
            }
        }
    }
}
```

```
        catch (Exception ex)
        {
            throw new ArgumentException(ex.Message);
        }
    }

    public static List<con_usuario> obtenerUsuariosXNombre(string nombreRol)
    {
        try
        {
            dc = new DCTesisDataContext();
            var lista = dc.con_usuario.Where(usu => usu.usu_estado == "Activo"
                                            &&
            usu.usu_nombre.StartsWith(nombreRol)).OrderBy(usu => usu.usu_apellido).ToList();
            return lista;
        }
        catch (Exception ex)
        {
            throw new ArgumentException(ex.Message);
        }
    }

    public static List<con_usuario> obtenerUsuariosXCargo(int cargoId)
    {
        try
        {
            dc = new DCTesisDataContext();
            var lista = dc.con_usuario.Where(usu => usu.usu_estado == "Activo"
                                            && usu.usu_cargo == cargoId).ToList();
            return lista;
        }
        catch (Exception ex)
        {
            throw new ArgumentException(ex.Message);
        }
    }

    public static void saveUsuarios(con_usuario _infoRol)
    {
        try
        {
            dc = new DCTesisDataContext();
```

```
_infoRol.usu_estado = "Activo";

_infoRol.usu_creacion = DateTime.Now;
_infoRol.usu_modificar = "00/00/0000 00:00:00";
dc.con_usuario.InsertOnSubmit(_infoRol);
dc.SubmitChanges();

}
catch (Exception ex)
{
    throw new ArgumentException(ex.Message);
}
}

public static void deleteUsuarios(con_usuario _infoRol)
{
    try
    {
        dc.SubmitChanges();
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static void updateUsuarios(con_usuario _infoRol)
{
    try
    {
        dc = new DCTesisDataContext();
        _infoRol.usu_estado = "Activo";
        _infoRol.usu_modificar = Convert.ToString(DateTime.Now);
        dc.updateUsuarios(_infoRol.usu_id, _infoRol.usu_nombre,
_infoRol.usu_apellido, _infoRol.usu_cedula, _infoRol.usu_correo,
_infoRol.usu_telefono, _infoRol.usu_usuario, _infoRol.usu_contrasenia,
_infoRol.usu_estado, _infoRol.usu_cargo, _infoRol.usu_modificar);
        dc.SubmitChanges();

    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}
}
}
```


Módulo de Administración

Empresas

Controlador

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Controlador;
using AccesosDatos;

namespace ITSM_Control.Administracion
{
    public partial class AdmEmpresas : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                if (Request["cod"] != null)
                {
                    string codigo = Convert.ToString(Request["cod"]);
                    if (!string.IsNullOrEmpty(codigo))
                    {
                        loadTipoR(codigo);
                    }
                }
                cargarEmpresa();
            }
        }

        protected void btnAgregar_Click(object sender, EventArgs e)
        {
            if (Convert.ToString(Request["cod"]) != null)
            {
                GuardarDatos(int.Parse(Convert.ToString(Request["cod"])));
            }
            else
            {
            }
        }
    }
}
```

```
        GuardarDatos(0);
    }
}
private void loadTipoR(string codigo)
{
    con_empresa _infoTipoR = new con_empresa();
    _infoTipoR = logicaEmpresa.obtenerEmpresaXID(int.Parse(codigo));

    if (_infoTipoR != null)
    {
        lblCodigo.Text = _infoTipoR.emp_id.ToString();
        txtDescripcion.Text = _infoTipoR.emp_descripcion;
        txtRuc.Text = _infoTipoR.emp_ruc;
        txtDireccion.Text = _infoTipoR.emp_direccion;
        txtTelefono.Text = _infoTipoR.emp_telefono;

    }
}

private void GuardarDatos(int codigo)
{
    if (codigo > 0)
    {
        update();
    }
    else
    {
        Save();
    }
}

private void update()
{
    if (Request["cod"] != null)
    {
        string codigo = Convert.ToString(Request["cod"]);
        if (!string.IsNullOrEmpty(codigo))
        {
            try
            {
                con_empresa _infoRepresentante = new con_empresa();
                // _infoDepartamento
logicaDepartamento.obtenerDepartamentoXID(int.Parse(codigo));
                _infoRepresentante.emp_id = int.Parse(lblCodigo.Text);
                _infoRepresentante.emp_descripcion
txtDescripcion.Text.ToUpper();
            }
            catch { }
        }
    }
}
```

```
        _infoRepresentante.emp_ruc = txtRuc.Text;
        _infoRepresentante.emp_direccion = txtDireccion.Text;
        _infoRepresentante.emp_telefono = txtTelefono.Text;
        logicaEmpresa.updateEmpresa(_infoRepresentante);
        Response.Redirect("/Administracion/AdmEmpresas.aspx");
    }
    catch (Exception ex)
    {
        lblMensaje.Text = ex.Message;
    }
}
}

private void Save()
{
    try
    {
        var messageError = "";
        if (string.IsNullOrEmpty(txtDescripcion.Text))
        {
            messageError += "Nombre campo obligatorio \n";
        }

        if (!string.IsNullOrEmpty(messageError))
        {
            lblMensaje.Text = messageError;
            return;
        }

        con_empresa inst = new con_empresa();
        inst.emp_descripcion = txtDescripcion.Text.ToUpper();
        inst.emp_ruc = txtRuc.Text;
        inst.emp_direccion = txtDireccion.Text;
        inst.emp_telefono = txtTelefono.Text;

        if (inst.emp_descripcion != (txtDescripcion.Text))
        {
            logicaEmpresa.saveEmpresa(inst);
            lblMensaje.Text = "Registro Guardado Correctamente";
            Response.Redirect("/Administracion/AdmEmpresas.aspx");
        }
        else
        {
            lblMensaje.Text = "Registro Repetido";
        }
    }
}
```

```
}
catch (Exception ex)
{
    lblMensaje.Text = ex.Message;

}
}
//datos formulario
private void cargarEmpresa()
{
    List<con_empresa> listaUsuario = new List<con_empresa>();
    listaUsuario = logicaEmpresa.obtenerEmpresa();
    if (listaUsuario.Count > 0 && listaUsuario != null)
    {
        GdvEmpresa.DataSource = listaUsuario;
        GdvEmpresa.DataBind();
    }
}
protected void GdvEmpresa_RowCommand(object sender,
GridViewCommandEventArgs e)
{
    string codigo = Convert.ToString(e.CommandArgument);

    if (e.CommandName == "Modificar")
    {
        //string codigoen =
FormsAuthentication.HashPasswordForStoringInConfigFile(codigo, "MD5");
        //codigo = Encriptar(codigo, llave);
        codigo = codigo;
        Response.Redirect("/Administracion/AdmEmpresas.aspx?cod=" + codigo,
true);
    }
    if (e.CommandName == "Eliminar")
    {
        if (!string.IsNullOrEmpty(codigo))
        {
            con_empresa _infoUsuario = new con_empresa();
            _infoUsuario = logicaEmpresa.obtenerEmpresaXID(int.Parse(codigo));
            if (_infoUsuario != null)
            {
                logicaEmpresa.deleteEmpresa(_infoUsuario);
                cargarEmpresa();
            }
        }
    }
}
```

```
}

protected void GdvEmpresa_SelectedIndexChanged(object sender, EventArgs
e)
{
}
private void nuevo()
{
    Response.Redirect("/Administracion/AdmEmpresas.aspx");
}
protected void lnkNuevo_Click(object sender, EventArgs e)
{
    nuevo();
}

protected void imgNuevo_Click(object sender, ImageClickEventArgs e)
{
    nuevo();
}

protected void imgBuscar_Click(object sender, ImageClickEventArgs e)
{
    buscar(ddlBuscar.SelectedValue);
}
private void buscar(string op)
{
    switch (op)
    {
        case "T":
            cargarEmpresa();
            break;
        case "C":
            GdvEmpresa.DataSource
logicaEmpresa.obtenerEmpresaxID(txtBuscar.Text);
            GdvEmpresa.DataBind();
            break;
        case "N":
            GdvEmpresa.DataSource
logicaEmpresa.obtenerEmpresaXNombre(txtBuscar.Text);
            GdvEmpresa.DataBind();
            break;
    }
}

protected void btnRegresar_Click(object sender, EventArgs e)
{
    Response.Redirect("/Menu.aspx");
}
```

```
    }  
  }  
}
```

Clase

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using AccesosDatos;  
  
namespace Controlador  
{  
    public class logicaEmpresa  
    {  
        private static DCTesisDataContext dc { get; set; }  
  
        public static List<con_empresa> obtenerEmpresa()  
        {  
            try  
            {  
                dc = new DCTesisDataContext();  
                var lista = dc.con_empresa.Where(usu => usu.emp_estado ==  
"Activo").OrderBy(usu => usu.emp_descripcion).ToList();  
                return lista;  
            }  
            catch (Exception ex)  
            {  
                throw new ArgumentException(ex.Message);  
            }  
        }  
  
        public static con_empresa obtenerEmpresaXID(int codigo)  
        {  
            try  
            {  
                dc = new DCTesisDataContext();  
                var rol = dc.con_empresa.FirstOrDefault(usu => usu.emp_estado ==  
"Activo"  
                && usu.emp_id.Equals(codigo));  
                return rol;  
            }  
            catch (Exception ex)  
            {  
                throw new ArgumentException(ex.Message);  
            }  
        }  
    }  
}
```

```
public static List<con_empresa> obtenerEmpresaxID(string codigo)
{
    try
    {
        dc = new DCTesisDataContext();
        var rol = dc.con_empresa.Where(usu => usu.emp_estado == "Activo"
                                     && usu.emp_id.Equals(codigo));

        return rol.ToList();
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static List<con_empresa> obtenerEmpresaXNombre(string nombreRol)
{
    try
    {
        dc = new DCTesisDataContext();
        var lista = dc.con_empresa.Where(usu => usu.emp_estado == "Activo"
                                     &&
usu.emp_descripcion.StartsWith(nombreRol)).OrderBy(usu                =>
usu.emp_descripcion).ToList();
        return lista;
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static void saveEmpresa(con_empresa _infoRol)
{
    try
    {
        dc = new DCTesisDataContext();
        _infoRol.emp_estado = "Activo";

        _infoRol.emp_creacion = DateTime.Now;

        _infoRol.emp_modificacion = "00/00/0000 00:00:00";

        dc.con_empresa.InsertOnSubmit(_infoRol);
        dc.SubmitChanges();
    }
}
```

```
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static void deleteEmpresa(con_empresa _infoRol)
{
    try
    {
        //dc = new DCHelpDeskDataContext();
        _infoRol.emp_estado = "Inactivo";
        //_infoRol.ROL_ADD = DateTime.Now;
        // dc.TBL_DEPARTAMENTOS.InsertOnSubmit(_infoDepartamento);
        dc.SubmitChanges();

    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static void updateEmpresa(con_empresa _infoRol)
{
    try
    {
        dc = new DCTesisDataContext();
        _infoRol.emp_estado = "Activo";
        _infoRol.emp_modificacion = Convert.ToString(DateTime.Now);
        dc.updateEmpresa(_infoRol.emp_id, _infoRol.emp_descripcion,
        _infoRol.emp_ruc, _infoRol.emp_estado, _infoRol.emp_modificacion,
        _infoRol.emp_direccion, _infoRol.emp_telefono);
        dc.SubmitChanges();

    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}
}
```


Mantenimiento Personas

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using Controlador;
using AccesoDatos;

namespace ITSM_Control.Administracion
{
    public partial class AdmPersonas : System.Web.UI.Page
    {
        protected void Page_Load(object sender, EventArgs e)
        {
            if (!IsPostBack)
            {
                if (Request["cod"] != null)
                {
                    string codigo = Convert.ToString(Request["cod"]);
                    if (!string.IsNullOrEmpty(codigo))
                    {
                        loadTipoR(codigo);
                    }
                }
                cargarPersona();
                cargarTUsuarios();
                cargarProyecto();
            }
        }

        protected void btnAgregar_Click(object sender, EventArgs e)
        {
            if (Convert.ToString(Request["cod"]) != null)
            {
                GuardarDatos(int.Parse(Convert.ToString(Request["cod"])));
            }
            else
            {
                GuardarDatos(0);
            }
        }
    }
}
```

```
private void loadTipoR(string codigo)

{
    con_asipersonas _infoTipoR = new con_asipersonas();
    _infoTipoR = logicaPersonas.obtenerPersonasXID(int.Parse(codigo));

    if (_infoTipoR != null)
    {
        lblCodigo.Text = _infoTipoR.asip_id.ToString();
        ddlusuario.SelectedVale = _infoTipoR.asip_usuario.ToString();
        ddlproyecto.SelectedVale = _infoTipoR.asip_proyecto.ToString();
    }
}

private void GuardarDatos(int codigo)
{
    if (codigo > 0)
    {
        update();
    }
    else
    {
        Save();
    }
}

private void nuevo()
{
    Response.Redirect("/Administracion/AdmPersonas.aspx");
}
protected void lnkNuevo_Click(object sender, EventArgs e)
{
    nuevo();
}

protected void imgNuevo_Click(object sender, ImageClickEventArgs e)
{
    nuevo();
}

private void update()
{
    if (Request["cod"] != null)
    {
        string codigo = Convert.ToString(Request["cod"]);
        if (!string.IsNullOrEmpty(codigo))
        {

```

```
try
{
    con_asipersonas _infoRepresentante = new con_asipersonas();
    // _infoDepartamento
logicaDepartamento.obtenerDepartamentoID(int.Parse(codigo));
    _infoRepresentante.asip_id = int.Parse(lblCodigo.Text);
    _infoRepresentante.asip_usuario
int.Parse(ddlusuario.Selected Value);
    _infoRepresentante.asip_proyecto
int.Parse(ddlproyecto.Selected Value);
    logicaPersonas.updatePersonas(_infoRepresentante);
    Response.Redirect("/Administracion/AdmPersonas.aspx");
}
catch (Exception ex)
{
    lblMensaje.Text = ex.Message;
}
}
}
```

```
private void Save()
{
```

```
try
{
    var messageError = "";
    //if (string.IsNullOrEmpty(txtNombre.Text))
    //{
    //    messageError += "Nombre campo obligatorio \n";
    //}
}
```

```
if (!string.IsNullOrEmpty(messageError))
{
    lblMensaje.Text = messageError;
    return;
}
```

```
con_asipersonas inst = new con_asipersonas();
```

```
inst.asip_usuario = int.Parse(ddlusuario.Selected Value);
inst.asip_proyecto = int.Parse(ddlproyecto.Selected Value);
```

```
logicaPersonas.savePersona(inst);
lblMensaje.Text = "Registro Guardado Correctamente";
Response.Redirect("/Administracion/AdmPersonas.aspx");
```

```
}
catch (Exception ex)
{
    lblMensaje.Text = ex.Message;

}
}

private void cargarPersona()
{
    List<PersonaProyecto> listaUsuario = new List<PersonaProyecto>();
    listaUsuario = logicaPersonaReporte.ObtenerPersona();
    if (listaUsuario.Count > 0 && listaUsuario != null)
    {
        GdvPersonas.DataSource = listaUsuario;
        GdvPersonas.DataBind();
    }
}

protected void GdvPersonas_RowCommand(object sender,
GridViewCommandEventArgs e)
{
    string codigo = Convert.ToString(e.CommandArgument);

    if (e.CommandName == "Modificar")
    {
        //string codigoen =
FormsAuthentication.HashPasswordForStoringInConfigFile(codigo, "MD5");
        //codigo = Encriptar(codigo, llave);
        codigo = codigo;
        Response.Redirect("/Administracion/AdmPersonas.aspx?cod=" + codigo,
true);
    }
    if (e.CommandName == "Eliminar")
    {
        if (!string.IsNullOrEmpty(codigo))
        {
            con_asipersonas _infoUsuario = new con_asipersonas();
            _infoUsuario = logicaPersonas.obtenerPersonasXID(int.Parse(codigo));
            if (_infoUsuario != null)
            {
                logicaPersonas.deletePersona(_infoUsuario);
                cargarPersona();
            }
        }
    }
}
```

```
        }
    }
}
private void cargarTUsuarios()
{
    List<con_usuario> _listaTh = new List<con_usuario>();
    _listaTh = logicaUsuarios.obtenerUsuarios();
    if (_listaTh.Count > 0 && _listaTh != null)
    {
        Seleccione--" });
        _listaTh.Insert(0, new con_usuario { usu_id = 0, usu_nombre = "--
        ddlusuario.DataSource = _listaTh;
        ddlusuario.DataTextField = "usu_nombre";
        ddlusuario.DataValueField = "usu_id";
        ddlusuario.DataBind();
    }
}
private void cargarProyecto()
{
    List<con_proyecto> _listaTh = new List<con_proyecto>();
    _listaTh = logicaProyecto.obtenerProyecto();
    if (_listaTh.Count > 0 && _listaTh != null)
    {
        Seleccione--" });
        _listaTh.Insert(0, new con_proyecto { pro_id = 0, pro_nombrepro = "--
        ddlproyecto.DataSource = _listaTh;
        ddlproyecto.DataTextField = "pro_nombrepro";
        ddlproyecto.DataValueField = "pro_id";
        ddlproyecto.DataBind();
    }
}

protected void imgBuscar_Click(object sender, ImageClickEventArgs e)
{
    buscar(ddlBuscar.Selected.Value);
}
private void buscar(string op)
{
    switch (op)
    {
        case "T":
            cargarPersona();
            break;
        case "C":
    }
```

```
                GdvPersonas.DataSource =  
logicaPersonas.obtenerPersonasxID(txtBuscar.Text);  
                GdvPersonas.DataBind();  
                break;  
            case "N":  
                GdvPersonas.DataSource =  
logicaPersonas.obtenerPersonasXNombre(txtBuscar.Text);  
                GdvPersonas.DataBind();  
                break;  
            }  
        }  
    }  
  
    protected void btnRegresar_Click(object sender, EventArgs e)  
    {  
        Response.Redirect("/Menu.aspx");  
    }  
}
```

Mantenimiento Proyecto

```
using System;  
using System.Collections.Generic;  
using System.Linq;  
using System.Text;  
using System.Threading.Tasks;  
using AccesoDatos;  
  
namespace Controlador  
{  
    public class logicaEmpresa  
    {  
        private static DCTesisDataContext dc { get; set; }  
  
        public static List<con_empresa> obtenerEmpresa()  
        {  
            try  
            {  
                dc = new DCTesisDataContext();  
                var lista = dc.con_empresa.Where(usu => usu.emp_estado ==  
"Activo").OrderBy(usu => usu.emp_descripcion).ToList();  
                return lista;  
            }  
            catch (Exception ex)  
            {  
                throw new ArgumentException(ex.Message);  
            }  
        }  
    }  
}
```

```
public static con_empresa obtenerEmpresaXID(int codigo)
{
    try
    {
        dc = new DCTesisDataContext();
        var rol = dc.con_empresa.FirstOrDefault(usu => usu.emp_estado ==
"Activo"
                                                && usu.emp_id.Equals(codigo));

        return rol;
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static List<con_empresa> obtenerEmpresaxID(string codigo)
{
    try
    {
        dc = new DCTesisDataContext();
        var rol = dc.con_empresa.Where(usu => usu.emp_estado == "Activo"
                                                && usu.emp_id.Equals(codigo));

        return rol.ToList();
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static List<con_empresa> obtenerEmpresaXNombre(string nombreRol)
{
    try
    {
        dc = new DCTesisDataContext();
        var lista = dc.con_empresa.Where(usu => usu.emp_estado == "Activo"
                                                &&
usu.emp_descripcion.StartsWith(nombreRol)).OrderBy(usu
usu.emp_descripcion).ToList();
        return lista;
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}
```

```
public static void saveEmpresa(con_empresa _infoRol)
{
    try
    {
        dc = new DCTesisDataContext();
        _infoRol.emp_estado = "Activo";

        _infoRol.emp_creacion = DateTime.Now;

        _infoRol.emp_modificacion = "00/00/0000 00:00:00";

        dc.con_empresa.InsertOnSubmit(_infoRol);
        dc.SubmitChanges();

    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static void deleteEmpresa(con_empresa _infoRol)
{
    try
    {
        //dc = new DCHelpDeskDataContext();
        _infoRol.emp_estado = "Inactivo";
        //_infoRol.ROL_ADD = DateTime.Now;
        // dc.TBL_DEPARTAMENTOS.InsertOnSubmit(_infoDepartamento);
        dc.SubmitChanges();

    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static void updateEmpresa(con_empresa _infoRol)
{
    try
    {
        dc = new DCTesisDataContext();
        _infoRol.emp_estado = "Activo";
```



```
        _infoRol.emp_modificacion = Convert.ToString(DateTime.Now);
        dc.updateEmpresa(_infoRol.emp_id, _infoRol.emp_descripcion,
        _infoRol.emp_ruc, _infoRol.emp_estado, _infoRol.emp_modificacion,
        _infoRol.emp_direccion, _infoRol.emp_telefono);
        dc.SubmitChanges();

    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}
}
```

Mantenimiento Test Pruebas

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using AccesosDatos;

namespace Controlador
{
    public class logicaGui
    {
        private static DCTesisDataContext dc { get; set; }

        public static List<con_detalle> obtenerDetalleGui()
        {
            try
            {
                dc = new DCTesisDataContext();
                var lista = dc.con_detalle.Where(usu => usu.det_estado ==
"Activo").OrderBy(usu => usu.det_descripcion).ToList();
                return lista;
            }
            catch (Exception ex)
            {
                throw new ArgumentException(ex.Message);
            }
        }
    }
}
```

```
public static List<con_detalle> obtenerDetalleAceptacion()
```

```
{
    try
    {
        dc = new DCTesisDataContext();
        var lista = dc.con_detalle.Where(usu => usu.det_estado == "Activo"
            & usu.det_prueba.Equals('6')).ToList();
        return lista;
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static List<con_detalle> obtenerDetalleSeguridad()
{
    try
    {
        dc = new DCTesisDataContext();
        var lista = dc.con_detalle.Where(usu => usu.det_estado == "Activo"
            & usu.det_prueba.Equals(4)).OrderBy(usu => usu.det_nombre).ToList();
        return lista;
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static List<con_detalle> obtenerDetalleValidacion()
{
    try
    {
        dc = new DCTesisDataContext();
        var lista = dc.con_detalle.Where(usu => usu.det_estado == "Activo"
            & usu.det_prueba.Equals(5)).OrderBy(usu => usu.det_nombre).ToList();
        return lista;
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static con_detalle obtenerDetalleXID(int codigo)
{
    try
```

```
{
    dc = new DCTesisDataContext();
    var rol = dc.con_detalle.FirstOrDefault(usu => usu.det_estado == "Activo"
        && usu.det_id.Equals(codigo));

    return rol;
}
catch (Exception ex)
{
    throw new ArgumentException(ex.Message);
}
}

public static List<con_detalle> obtenerDetallesID(string codigo)
{
    try
    {
        dc = new DCTesisDataContext();
        var rol = dc.con_detalle.Where(usu => usu.det_estado == "Activo"
            && usu.det_id.Equals(codigo));

        return rol.ToList();
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static List<con_detalle> obtenerDetalleXNombre(string nombreRol)
{
    try
    {
        dc = new DCTesisDataContext();
        var lista = dc.con_detalle.Where(usu => usu.det_estado == "Activo"
            &&
usu.det_nombre.StartsWith(nombreRol)).OrderBy(usu => usu.det_nombre).ToList();
        return lista;
    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static void saveDetalle(con_detalle _infoRol)
{
    try
```

```
{

    dc = new DCTesisDataContext();

    _infoRol.det_estado = "Activo";
    _infoRol.det_modificacion = "00/00/0000 00:00:00";
    dc.con_detalle.InsertOnSubmit(_infoRol);
    dc.SubmitChanges();

}
catch (Exception ex)
{
    throw new ArgumentException(ex.Message);
}
}

public static void deleteDetalle(con_detalle _infoRol)
{
    try
    {

    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}

public static void updateDetalle(con_detalle _infoRol)
{
    try
    {

    }
    catch (Exception ex)
    {
        throw new ArgumentException(ex.Message);
    }
}
}
```

Diccionario de Datos

ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE SEGUROS DE LA EMPRESA ITSM – ECUADOR, PERÍODO OCTUBRE 2018 – MARZO 2019.

Descripción Tabla Asignación personas con_asipersonas

Nombre del campo	Tipo de dato	Integridad	Longitud	Descripción
asip_id	Int	PrimaryKey autoincrementnt	11	Código de asignación
asip_proyecto	varchar	Not null	11	Código de proyecto
asip_estado	varchar	Not null	11	Estado de la asignación
asip_modificacion	varchar	Not null	150	Fecha de modificación
asip_creacion	datetime	Not null	11	Fecha de creación

ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE SEGUROS DE LA EMPRESA ITSM – ECUADOR, PERÍODO OCTUBRE 2018 – MARZO 2019.

Descripción Tabla Asignación personas con_asipersonas

Nombre del campo	Tipo de dato	Integridad	Longitud	Descripción
car_id	Int	PrimaryKey autoincrementnt	11	Código de asignación
car_descripcion	varchar	Not null	11	Descripción de cargo
car_estado	varchar	Not null	11	Estado de la asignación
car_modificacion	varchar	Not null	150	Fecha de modificación
car_creacion	datetime	Not null	11	Fecha de creación

ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE SEGUROS DE LA EMPRESA ITSM – ECUADOR, PERÍODO OCTUBRE 2018 – MARZO 2019.

Descripción Tabla Asignación personas con_asipersonas

Nombre del campo	Tipo de dato	Integridad	Longitud	Descripción
------------------	--------------	------------	----------	-------------

det_id	Int	PrimaryKey autoincrementnt	11	Código de asignación
det_proposito	varchar	Not null	200	Descripción de propósito
det_descripcion	varchar	Not null	200	Descripción
det_procedimiento	varchar	Not null	200	Descripción del procedimiento
det_resultado	varchar	Not null	200	Descripción del resultado
det_imagen	image	Not null	100	Registro de imagen
det_errores	varchar	Not null	300	Descripción de errores
det_resulfinal	varchar	Not null	100	Descripción de errores
det_resolucion	varchar	Not null	200	Descripción de resolución
det_fecha	datetime	Not null	10	Descripción de la fecha
det_realizado	varchar	Not null	50	Descripción de realizado
det_módulo	int	Not null foreingkey	11	Código
det_objetivo	varchar	Not null	200	Descripción de objetivo
det_observaciones	varchar	Not null	200	Descripción de observaciones
det_url	varchar	Not nul	100	Descripción de url
det_funcionalidad	varchar	Not null	200	Descripción de funcionalidad
det_etapa	varchar	Not null	200	Descripción de etapa
det_requerimiento	int	Not null Foreingkey	11	Código
det_elemento	varchar	Not null	100	Descripción del elemento
det_estadopro	varchar	Not null	20	Descripción del estado
det_estado	varchar	Not null	10	Registro de estado
det_observacionpro	varchar	Not null	200	Descripción de observación

det_modificacion	varchar	Not null	150	Fecha de modificación
det_proyecto	int	Not null foreignkey	11	Código
det_prueba	int	Not null foreignkey	11	Código
det_persona	int	Not null foreignkey	11	Código

ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE SEGUROS DE LA EMPRESA ITSM – ECUADOR, PERÍODO OCTUBRE 2018 – MARZO 2019.

Descripción Tabla Asignación personas con_asipersonas

Nombre del campo	Tipo de dato	Integridad	Longitud	Descripción
emp_id	Int	PrimaryKey autoincrementnt	11	Código de asignación
emp_descripcion	varchar	Not null	11	Descripción de cargo
emp_ruc	varchar	Not null	13	Descripción de ruc
emp_estado	varchar	Not null	150	Descripción de estado
emp_modificacion	datetime	Not null	11	Fecha de modificación
emp_creación	datetime	Not null	11	Fecha de creación
emp_dirección	varchar	Not null	100	Descripción de dirección
emp_telefono	varchar	Not null	10	Descripción de teléfono

ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE SEGUROS DE LA EMPRESA ITSM – ECUADOR, PERÍODO OCTUBRE 2018 – MARZO 2019.

Descripción Tabla Asignación personas con_asipersonas

Nombre del campo	Tipo de dato	Integridad	Longitud	Descripción
modu_id	Int	PrimaryKey	11	Código de

		autoincrementnt		asignación
modu_descripcion	varchar	Not null	11	Descripción de modulo
modu_estado	varchar	Not null	150	Descripción de estado
modu_proyecto	int	Not null	11	Código

ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE SEGUROS DE LA EMPRESA ITSM – ECUADOR, PERÍODO OCTUBRE 2018 – MARZO 2019.

Descripción Tabla Asignación personas con_asipersonas

Nombre del campo	Tipo de dato	Integridad	Longuitud	Descripción
pro_id	Int	PrimaryKey autoincrementnt	11	Código de asignación
pro_empresa	int	Not null foreingkey	11	Código
pro_nombrepro	varchar	Not null	100	Nombre del Proyecto
pro_fechaini	date	Not null	11	Fecha de inicio
pro_fechafin	date	Not null	11	Fecha de finalización
pro_estado	varchar	Not null	10	Estado
pro_modificacion	varchar	Not null	10	Fecha de modificación
pro_creacion	datetime	Not null	10	Fecha de creación

ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE SEGUROS DE LA EMPRESA ITSM – ECUADOR, PERÍODO OCTUBRE 2018 – MARZO 2019.

Descripción Tabla Asignación personas con_asipersonas

Nombre del campo	Tipo de dato	Integridad	Longuitud	Descripción
cap_id	Int	PrimaryKey autoincrementnt	11	Código de asignación
cap_nombre	varchar	Not null	100	Descripción del nombre

cap_descripcion	varchar	Not null	100	Descripción
cap_estado	varchar	Not null	10	Estado
cap_creacion	datetime	Not null	11	Fecha de creación

ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE SEGUROS DE LA EMPRESA ITSM – ECUADOR, PERÍODO OCTUBRE 2018 – MARZO 2019.

Descripción Tabla Asignación personas con_asipersonas

Nombre del campo	Tipo de dato	Integridad	Longitud	Descripción
req_id	Int	PrimaryKey autoincrementnt	11	Código de asignación
req_numeracion	Varchar	Not null	10	Descripción de la numeración
req_descripcion	Varchar	Not null	200	Descripción
req_proyecto	int	Not null	11	Código
req_modificacion	datetime	Not null	11	Fecha de modificación
req_creacion	datetime	Not null	11	Fecha de creación
req_estado	varchar	Not null	10	Estado

ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE SEGUROS DE LA EMPRESA ITSM – ECUADOR, PERÍODO OCTUBRE 2018 – MARZO 2019.

Descripción Tabla Asignación personas con_asipersonas

Nombre del campo	Tipo de dato	Integridad	Longitud	Descripción
usu_id	Int	PrimaryKey autoincrementnt	11	Código de asignación
usu_nombre	Varchar	Not null	10	Descripción de la numeración
usu_apellido	Varchar	Not null	200	Descripción
usu_cedula	int	Not null	11	Código
usu_correo	datetime	Not null	11	Fecha de modificación

usu_telefono	datetime	Not null	11	Fecha de creación
usu_usuario	varchar	Not null	10	Estado
usu_contrasenia	Varchar	Not null	20	Contraseña
usu_estado	varchar	Not null	10	Estado
usu_cargo	int	Not null	11	Código
		foreingkey		
usu_modificar	varchar	Not null	150	Fecha de modificación
usu_creación	datetime	Not null	11	Fecha de creación

Quito 06 de junio del 2019

Señores

INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

Presente.

De mi consideración

Me permito emitir el presente certificado, correspondiente a la entrega e implementación del Software Desarrollado en el Instituto Tecnológico Superior Cordillera, dicha herramienta se encuentra en pre- producción y se está personalizando, ya que ha cumplido con los requisitos solicitados por parte de nuestra institución (ITSM- Ecuador) se ha dispuesto implementar el sistema de Administración de Pruebas de Control de Calidad, de manera profesional a la Sita. Jessica Estefanía Cacuango Catcuamba con el ID 172509458-3.

El trabajo sobre ANÁLISIS DEL PROCESO DE CONTROL DE CALIDAD PARA EL SOFTWARE GESTIÓN DE VENTAS Y OPERACIONES DE SEGUROS DE LA EMPRESA ITSM ECUADOR, PERÍODO OCTUBRE 2018 – MARZO 2019, se encuentra implementado satisfactoriamente en la institución desde el 03 de junio del 2019.

Es todo cuanto puedo decir en honor a la verdad.

Atentamente



Ing. Karina Campaña

Gerente de Proyectos y Negocios



ITSM ECUADOR Cia. Ltda.
SOLUCIONES EN SERVICIOS
DE GESTIÓN TECNOLÓGICA

Urkund Analysis Result

Analysed Document: Tesis_Jessica_Estefania_Cacuango_Catucuamba.pdf (D53299858)
Submitted: 6/3/2019 3:43:00 AM
Submitted By: jessytesy25@gmail.com
Significance: 8 %

Sources included in the report:

HENRY GUSTAVO PULUPA PUETATE 2017 TESIS.pdf (D30551285)
Alex Ajila-Trabajo Titulacion.docx (D43472020)
Valle_Edgar.pdf (D15723494)
URKUND_RODOLFO_SANTIAGO_VITERI_CUERO_SISTEMAS.pdf (D26604232)
<https://proyectosagiles.org/que-es-scrum/>

Instances where selected sources appear:

16


03-06-2019
M.J.D.

INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA									
BETACORA PARA EL CONTROL DE PROYECTOS DE TITULACIÓN									
FAMILIA DEVALLOS (AMÉLIE MARI SUZANNE)									
CARRERA: CAMBANGO CATALUÑA, BISSICA, ESTIPANIA									
ANÁLISIS DE SISTEMAS									
SISTEMATIZACIÓN DEL PROCESO DE CONTROL DE CALIDAD PARA EL SOFTWARE GESTION DE VENTAS Y OBLIGACIONES DE SEGUROS DE LA EMPRESA TRISA ELUADOR, PERIODO OCTUBRE 2013 - MARZO 2014									
Quito, 31 de junio del 2019 por 19-27									
ACUMULATIVO									
MODALIDAD: INVESTIGACIÓN DESARROLLO INNOVACIÓN									
Nº. CÓDIGO	FECHA TUTORÍA	TIPO ASIGNATURA	HORA INICIO	TEMA TRATADO	HORA FIN	HORAS	OBSERVACION	ESTADO SC	OCT 2018 MAR 2019
1	150014	2018-02-06	INSITU	2019-02-08 16:00:00	ANTECEDENTES / CONTEXTO	2019-02-08 18:00:00	2.00	EL ALUMNO TUVO UNA REASIGNACIÓN DE TUTORÍA PRESENTANDO EL CAPÍTULO I. TIENE CONOCIMIENTO DE LA PROBLEMÁTICA, YA QUE NO ESPECIFICA UNA DESCRIPCIÓN CONCRETA DE LA MATRIZ T. TIENE REVISAR LA ORTOGRAFÍA Y LA CONSTRUCCIÓN DE LAS FRASES. DEBE REVISAR LOS ANTECEDENTES. EL PR DE PAGINA NO ES EL CORRECTO. EL ALUMNO DEBE REALIZAR LAS CORRECCIONES INDICADAS.	PROCESADO
2	150019	2018-02-06	AUTONOMA	2019-02-09 13:00:00	ANTECEDENTES / CONTEXTO	2019-02-09 17:00:00	4.00	EL ALUMNO PRESENTA LAS CORRECCIONES. FALTA ABRIGULAR LA MATRIZ T.	PROCESADO
3	150020	2018-02-11	INSITU	2019-02-13 16:00:00	ANTECEDENTES / CONTEXTO	2019-02-13 19:00:00	3.00	EL ALUMNO DEBE REALIZAR LAS CORRECCIONES INDICADAS Y PRESENTAR EL CAPÍTULO II.	PROCESADO
4	150021	2018-02-16	AUTONOMA	2019-02-16 06:00:00	ANTECEDENTES / CONTEXTO	2019-02-16 13:00:00	4.00	EL ALUMNO PRESENTA LAS CORRECCIONES SOLICITADAS. BIEN POR FINALIZADO EL CAPÍTULO II.	PROCESADO
5	150022	2018-02-23	INSITU	2019-02-23 16:00:00	ANTECEDENTES / CONTEXTO	2019-02-23 12:00:00	2.00	EL ALUMNO TUVO UNA REASIGNACIÓN DE TUTORÍA PRESENTANDO EL CAPÍTULO I. TIENE QUE REVISAR Y CORREGIR LA PROBLEMÁTICA, YA QUE NO ESPECIFICA CON CLARIDAD CUAL ES EL PROBLEMA CENTRAL. TIENE REVISAR LA ORTOGRAFÍA Y LA CONSTRUCCIÓN DE LAS FRASES. DEBE REVISAR LOS ANTECEDENTES. EL PR DE PAGINA NO ES EL CORRECTO. EL ALUMNO DEBE REALIZAR LAS CORRECCIONES INDICADAS.	PROCESADO
6	150015	2018-02-06	INSITU	2019-02-08 16:00:00	ANTECEDENTES / JUSTIFICACION	2019-02-08 18:00:00	2.00	EL ALUMNO PRESENTA LAS CORRECCIONES INDICADAS. FALTA ABRIGULAR LA MATRIZ T.	PROCESADO
7	150024	2018-02-06	AUTONOMA	2019-02-09 13:00:00	ANTECEDENTES / JUSTIFICACION	2019-02-09 17:00:00	4.00	EL ALUMNO PRESENTA LAS CORRECCIONES INDICADAS. FALTA ABRIGULAR LA MATRIZ T.	PROCESADO
8	150025	2018-02-11	INSITU	2019-02-13 16:00:00	ANTECEDENTES / JUSTIFICACION	2019-02-13 19:00:00	3.00	EL ALUMNO DEBE REALIZAR LAS CORRECCIONES INDICADAS Y PRESENTAR EL CAPÍTULO II.	PROCESADO
9	150025	2018-02-16	AUTONOMA	2019-02-16 06:00:00	ANTECEDENTES / JUSTIFICACION	2019-02-16 13:00:00	4.00	EL ALUMNO PRESENTA LAS CORRECCIONES SOLICITADAS. BIEN POR FINALIZADO EL CAPÍTULO II.	PROCESADO
10	150017	2018-02-23	INSITU	2019-02-23 16:00:00	ANTECEDENTES / JUSTIFICACION	2019-02-23 12:00:00	2.00	EL ALUMNO PRESENTA LAS CORRECCIONES INDICADAS. FALTA ABRIGULAR LA MATRIZ T.	PROCESADO
11	150028	2018-02-06	INSITU	2019-02-08 16:00:00	ANTECEDENTES / DEFINICION DEL PROBLEMA CENTRAL (MATRIZ T)	2019-02-08 18:00:00	2.00	EL ALUMNO TUVO UNA REASIGNACIÓN DE TUTORÍA PRESENTANDO EL CAPÍTULO I. TIENE QUE REVISAR Y CORREGIR LA PROBLEMÁTICA, YA QUE NO ESPECIFICA CON CLARIDAD CUAL ES EL PROBLEMA CENTRAL. TIENE REVISAR LA ORTOGRAFÍA Y LA CONSTRUCCIÓN DE LAS FRASES. DEBE REVISAR LOS ANTECEDENTES. EL PR DE PAGINA NO ES EL CORRECTO. EL ALUMNO DEBE REALIZAR LAS CORRECCIONES INDICADAS.	PROCESADO
12	150039	2018-02-06	AUTONOMA	2019-02-09 13:00:00	ANTECEDENTES / DEFINICION DEL PROBLEMA CENTRAL (MATRIZ T)	2019-02-09 17:00:00	4.00	EL ALUMNO PRESENTA LAS CORRECCIONES INDICADAS. FALTA ABRIGULAR LA MATRIZ T.	PROCESADO
13	150000	2018-02-11	INSITU	2019-02-13 16:00:00	ANTECEDENTES / DEFINICION DEL PROBLEMA CENTRAL (MATRIZ T)	2019-02-13 19:00:00	3.00	EL ALUMNO DEBE REALIZAR LAS CORRECCIONES INDICADAS Y PRESENTAR EL CAPÍTULO II.	PROCESADO
14	150001	2018-02-16	AUTONOMA	2019-02-16 06:00:00	ANTECEDENTES / DEFINICION DEL PROBLEMA CENTRAL (MATRIZ T)	2019-02-16 13:00:00	4.00	EL ALUMNO PRESENTA LAS CORRECCIONES SOLICITADAS. BIEN POR FINALIZADO EL CAPÍTULO II.	PROCESADO
15	150002	2018-02-23	INSITU	2019-02-23 16:00:00	ANTECEDENTES / DEFINICION DEL PROBLEMA CENTRAL (MATRIZ T)	2019-02-23 12:00:00	2.00	EL ALUMNO PRESENTA LAS CORRECCIONES INDICADAS. FALTA ABRIGULAR LA MATRIZ T.	PROCESADO

36	162185	2019-04-22	INSTITU	2019-04-22 16:00:00	CONCLUSIONES Y RECOMENDACIONES	2019-04-22 17:00:00	1.00	REVISAR LA ORTOGRAFIA LAS RECOMENDACIONES DEBEN CONTER LAS ACTIVIDADES A REALIZARSE CON LA FINALIDAD DE DAR CUMPLIMIENTO AL OBJETIVO PROPUES TO LAS RECOMENDACIONES SON MUY SIMPLES, NO INTRICAR CLARAMENTE QUE ES LO QUE SE DEBE HACER	PROCESADO
37	162186	2019-04-25	AUTONOMA	2019-04-25 17:00:00	CONCLUSIONES Y RECOMENDACIONES	2019-04-25 18:00:00	2.00	EL ALUMNO PRESENTA EL DOCUMENTO CON LAS SIGUIENTES SOLICITADAS	PROCESADO
38	162219	2019-04-27	AUTONOMA	2019-04-27 16:00:00	CONCLUSIONES Y RECOMENDACIONES	2019-04-27 16:00:00	3.00	MODIFICAR LA REPORTERIA INCLUTANDO FILTROS DE BUSQUEDA SE DEBE INCLUIR EN EL SISTEMA, CONTROLIS PARA EVITAR DUPLICIDAD	PROCESADO
39	162221	2019-04-28	AUTONOMA	2019-04-28 16:00:00	CONCLUSIONES Y RECOMENDACIONES	2019-04-28 16:00:00	3.00	REALIZAR LAS PRUEBAS DE INTEGRIDAD CON LA BASE DE DATOS	PROCESADO
40	162227	2019-04-28	AUTONOMA	2019-04-28 16:00:00	CONCLUSIONES Y RECOMENDACIONES	2019-04-28 16:00:00	3.00	SE DEBE REALIZAR LA DOCUMENTACIÓN INTERNA EN EL SISTEMA INCLUIR LA DOCUMENTACIÓN EN LAS NUEVAS TABLAS Y CAMPOS ESTANDARIZAR LAS INTERFACES	PROCESADO
41	162238	2019-04-29	AUTONOMA	2019-04-29 16:00:00	CONCLUSIONES Y RECOMENDACIONES	2019-04-29 17:00:00	7.00	BOLEAR LOS DATOS PRUEBA E INGRESAR DATOS VALIDOS PARAMETRIZAR EL SISTEMA CON INFORMACION REAL REALIZAR LAS PRUEBAS DE STRESS INCLUIR LA DOCUMENTACIÓN ADICIONAL AL DOCUMENTO CONE LA DEDICATORIA	PROCESADO
TOTAL HORAS:							246		
PABLO CESVALLOS GARCIA				C. ACANISO CATALAN BRISA ESTEFANIA			CORONEL GONZALEZ RUBEN PATRICIO		
TU TOR							DELEGADO		
CI: 706070179				CI: 17594583			CI: 18229433		
							FECHA:		



TECNOLÓGICO SUPERIOR
CORDILLERA

INSTITUTO TECNOLÓGICO SUPERIOR CORDILLERA

ANÁLISIS DE SISTEMAS

ORDEN DE EMPASTADO

Una vez verificado el cumplimiento de los requisitos establecidos para el proceso de Titulación, se **AUTORIZA** realizar el empastado del trabajo de titulación, del alumno(a) **CACUANGO CATUCUAMBA JESSICA ESTEFANIA**, portador de la cédula de identidad N° 172509458-3, previa validación por parte de los departamentos facultados.

Quito, 11 de Junio del 2019.

11 JUN 2019
Mariela Balseca
VISTO FINANCIERO

Sra. Mariela Balseca
CAJA

INSTITUTO TECNOLÓGICO SUPERIOR
CORDILLERA

13 JUN 2019
9,3 BS
COORDINACIÓN

Ing. Samira Villalba
PRÁCTICAS PREPROFESIONALES

INSTITUTO TECNOLÓGICO SUPERIOR
CORDILLERA

Ing. Johnny Coronel
CONSEJO DE CARRERA
DELEGADO DE LA UNIDAD DE TITULACIÓN

Análisis de Sistemas

BIBLIOTECA
INSTITUTO TECNOLÓGICO SUPERIOR
CORDILLERA

Ing. William Parra
BIBLIOTECA

INSTITUTO TECNOLÓGICO SUPERIOR
CORDILLERA

Ing. Diana Terán
DIRECTOR DE CARRERA

INSTITUTO TECNOLÓGICO SUPERIOR
CORDILLERA

13 JUN 2019
Carolina Guerra

Tgla. Carolina Guerra
SECRETARIA ACADÉMICA