**SISTEMA DE GESTIÓN PARA EL ANÁLISIS Y RESOLUCIÓN DE CAUSAS DE LA EMPRESA ASESOFTWARE**

**JEFFERSON RICARDO VIVAS TORRES**



**UNIVERSIDAD PEDAGÓGICA Y TECNOLÓGICA DE COLOMBIA**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE SISTEMAS Y COMPUTACIÓN**

**TUNJA-BOYACÁ**

**2017**

**SISTEMA DE GESTIÓN PARA EL ANÁLISIS Y RESOLUCIÓN DE CAUSAS DE LA EMPRESA ASESOFTWARE**

**JEFFERSON RICARDO VIVAS TORRES**

**Proyecto de grado para optar por el título de:**

**Ingeniero de Sistemas y Computación**

**Director**

**JUAN SEBASTIÁN GONZÁLEZ SANABRIA**

**Ingeniero de Sistemas y Computación**

**Especialista en Bases de Datos**



**UNIVERSIDAD PEDAGÓGICA Y TECNOLÓGICA DE COLOMBIA**

**FACULTAD DE INGENIERÍA**

**ESCUELA DE SISTEMAS Y COMPUTACIÓN**

**TUNJA-BOYACÁ**

**2017**

**Nota de aceptación**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Firma presidente del Jurado

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Firma Jurado

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Firma Jurado

Tunja, octubre de 2017

**TABLA DE CONTENIDO**

**Pág.**

INTRODUCCIÓN 14

1. PLANTEAMIENTO DEL PROBLEMA 15

1.1 ANTECEDENTES 15

1.2 DESCRIPCIÓN DEL PROBLEMA 15

1.3 FORMULACIÓN DEL PROBLEMA 16

1.4 SISTEMATIZACIÓN DEL PROBLEMA 16

2. OBJETIVOS 17

2.1 OBJETIVO GENERAL 17

2.2 OBJETIVOS ESPECÍFICOS 17

3. JUSTIFICACIÓN 18

4. MARCO REFERENCIAL 19

4.1 MARCO EMPRESARIAL 19

4.1.1 CAR. 19

4.1.2 Club CAR. 19

4.1.3 Política 033 SAM. 19

4.1.4 Rac 49 Log de CAR. 19

4.2 MARCO CONCEPTUAL 20

4.2.1 Servicio Windows. 20

4.2.2 Entity framework. 20

4.2.3 DTO 20

4.2.4 Typescript. 20

4.2.5 Ensamblados. 20

4.2.6 Patrones arquitectónicos. 21

5. METODOLOGÍA 24

5.1 DEFINICIÓN DEL PROCESO DE CICLO DE VIDA 24

5.1.1 Verificar información base. 24

5.1.2 Evaluación de la documentación del cliente. 24

5.1.3 Seleccionar el MCV. 25

5.1.4 Mapear las actividades estándar al patrón de arquitectura seleccionado. 25

5.1.5 Establecer el PCV. 25

5.2 ANÁLISIS DE REQUERIMIENTOS 25

5.2.1 Visión y alcance del proyecto. 25

5.2.2 Analizar los requerimientos. 25

5.2.3 Documentar los requerimientos. 26

5.2.4 Elaborar el modelo entidad relación. 26

5.2.5 Elaborar el modelo funcional. 26

5.3 DISEÑO 26

5.4 CONSTRUCCIÓN 27

5.4.1 Configurar el ambiente de desarrollo. 27

5.4.2 Codificar los programas de software. 27

5.4.3 Ejecución de análisis de código estático. 27

5.4.4 Diseño, construcción y ejecución de pruebas funcionales. 27

5.4.5 Ejecutar la integración de los programas de software. 27

5.4.6 Documentar el producto. 28

5.5 TRANSICIÓN 28

5.5.1 Configurar el ambiente de instalación. 28

5.5.2 Instalación del sistema. 28

5.5.3 Ejecutar migración de datos. 28

5.5.4 Capacitación. 28

6. DESARROLLO 29

6.1 DEFINICIÓN DEL PROCESO DE CICLO DE VIDA 29

6.1.1 Verificación de la información base. 29

6.1.2 Evaluación de la documentación del cliente. 29

6.1.3 Seleccionar el MVC 29

6.1.4 Establecer el MVC. 29

6.2 ANÁLISIS DE LOS REQUERIMIENTOS 31

6.2.1 Visión y alcance del proyecto. 31

6.2.2 Dilucidar y analizar los requerimientos. 33

6.2.3 Especificación de requerimientos. 34

6.2.4 Elaborar el modelo entidad relación. 35

6.2.5 Elaborar el modelo funcional. 37

6.3 Elaboración del diseño. 37

6.3.1 Definición de la arquitectura del sistema. 37

6.3.2 Diseño de la base de datos. 45

6.3.3 Estilos y tácticas de resolución de arquitectura 46

En esta sección, se presentan los escenarios identificados que pueden generar impacto en la arquitectura del sistema. 46

6.3.4 Elaboración del diseño detallado. 48

6.3.5 Diagrama de clases 51

6.3.6 Estándares de presentación y programación. 52

En esta sección se presentan los estándares de presentación que rigen el sistema CARCONTROL, así como los estándares de programación que se utilizan en la etapa de construcción. 52

6.3.7 Estrategia de implementación e integración. 55

6.4 REALIZACIÓN DE LA CONSTRUCCIÓN 56

6.4.1 Configuración del ambiente de desarrollo. 56

6.4.2 Codificar los programas de software. 56

6.4.3 Ejecución del análisis estático de código. 72

6.4.4 Diseño, construcción y ejecución de pruebas funcionales. 73

6.4.5 Ejecución de la integración de la aplicación. 78

6.4.6 Documentación del producto. 78

6.5 REALIZACIÓN DE LA TRANSICIÓN 79

6.5.1 Configuración del ambiente de producción. 79

6.5.2 Instalación del software. 79

6.5.3 Capacitación. 79

7. CONCLUSIONES 80

8. REFERENCIAS 82

**LISTA DE FIGURAS**

**Pág.**

[**Figura 1.** Patrón de diseño Proxy 22](#_Toc495578514)

[**Figura 2.** Patrón de diseño Facade 23](#_Toc495578515)

[**Figura 3**. Funcionalidades del sistema 34](#_Toc495578516)

[**Figura 4.** Modelo entidad relación 36](#_Toc495578517)

[**Figura 5.** Vista funcional del sistema 42](#_Toc495578518)

[**Figura 6**. Vista de contexto del sistema 45](#_Toc495578519)

[**Figura 7.** Diagrama de Secuencia Registrar CAR 49](#_Toc495578520)

[**Figura 8.** Diagrama de secuencia envío de alertas y notificaciones 50](#_Toc495578521)

[**Figura 17.** Diagrama de clases 51](#_Toc495578522)

[**Figura 10**. Estándar de presentación menú de navegación 52](#_Toc495578523)

[**Figura 11**. Estándar de presentación de botones 53](#_Toc495578524)

[**Figura 12**. Estándar de presentación de formularios 53](#_Toc495578525)

[**Figura 13.** Estándar de presentación mensaje de error 53](#_Toc495578526)

[**Figura 14.** Estándar de presentación mensaje de advertencia 53](#_Toc495578527)

[**Figura 15.** Estándar de presentación mensaje de éxito 54](#_Toc495578528)

[**Figura 16.** Pantalla registrar solicitud CAR 58](#_Toc495578529)

[**Figura 17.** Pantalla de aprobar CAR 59](#_Toc495578530)

[**Figura 18**. Correo aprobación CAR 59](#_Toc495578531)

[**Figura 19.** Pantalla gestionar CAR 61](#_Toc495578532)

[**Figura 20.** Pantalla consultar CAR administrador 62](#_Toc495578533)

[**Figura 21.** Pantalla consultar CAR usuario 63](#_Toc495578534)

[**Figura 22.** Pantalla historial básica de CAR 64](#_Toc495578535)

[**Figura 23.** Pantalla histórica avanzada de CAR 65](#_Toc495578536)

[**Figura 24.** Acceso a la funcionalidad de enviar link de documentación 65](#_Toc495578537)

[**Figura 25.** Pantalla advertencia envío link de documentación 66](#_Toc495578538)

[**Figura 26.** Pantalla éxito envío link de documentación 66](#_Toc495578539)

[**Figura 27.** Notificación envió de documentación 67](#_Toc495578540)

[**Figura 28**. Pantalla club CAR 68](#_Toc495578541)

[**Figura 29.** Notificación alerta 1 68](#_Toc495578542)

[**Figura 30.** Notificación alerta 2 69](#_Toc495578543)

[**Figura 31.** Notificación alerta 3 69](#_Toc495578544)

[**Figura 32**. Notificación de alerta 4 70](#_Toc495578545)

[**Figura 33.** Notificación de alerta 4.1 70](#_Toc495578546)

[**Figura 34.** Notificación de alerta 5 71](#_Toc495578547)

[**Figura 35.** Notificación de alerta 6 71](#_Toc495578548)

[**Figura 36.** Pantalla de ingreso al sistema 72](#_Toc495578549)

[**Figura 37.** Pantalla de acceso al módulo de CAR 72](#_Toc495578550)

[**Figura 38.** Análisis estático de código capa de presentación 73](#_Toc495578551)

[**Figura 39.** Priorización ejecución de pruebas funcionales 74](#_Toc495578552)

[**Figura 40.** Evidencia 1 prueba funcional REQ01 75](#_Toc495578553)

**Pág.**

[**Figura 41.** Evidencia 2 prueba funcional REQ01 75](#_Toc495578554)

[**Figura 42.** Evidencia 3 prueba funcional REQ01 76](#_Toc495578555)

[**Figura 43**. Evidencia 4 prueba funcional REQ01 76](#_Toc495578556)

[**Figura 44.** Evidencia 5 prueba funcional REQ01 77](#_Toc495578557)

[**Figura 45.** Evidencia ejecución casos de prueba TFS 78](#_Toc495578558)

**LISTA DE TABLAS**

**Pág.**

[**Tabla 1.** Actividades de diseño 26](#_Toc495533377)

[**Tabla 2.** Modelo de desarrollo 29](#_Toc495533378)

[**Tabla 3.** Características del Sistema 32](#_Toc495533379)

[**Tabla 4.** Objetivos del negocio 37](#_Toc495533380)

[**Tabla 5**. Características del sistema 37](#_Toc495533381)

[**Tabla 6.** Identificación de stakeholders 38](#_Toc495533382)

[**Tabla 7.** Criterios de evaluación frameworks de diseño 38](#_Toc495533383)

[**Tabla 8**. Evaluación framework de diseño bootstrap 39](#_Toc495533384)

[**Tabla 9.** Evaluación framework de diseño foundation 39](#_Toc495533385)

[**Tabla 10.** Evaluación framework de diseño materialize 40](#_Toc495533386)

[**Tabla 11.** Evaluación framework de diseño bootswatch 40](#_Toc495533387)

[**Tabla 12.** Elección framework de diseño 40](#_Toc495533388)

[**Tabla 13.** Criterios de evaluación manejo de notificaciones 40](#_Toc495533389)

[**Tabla 14.** Elección manejo de notificaciones 41](#_Toc495533390)

[**Tabla 15.** Manejo de excepciones 41](#_Toc495533391)

[**Tabla 16.** Descripción vista funcional del sistema 43](#_Toc495533392)

[**Tabla 17.** Vista de contexto del sistema 45](#_Toc495533393)

[**Tabla 18.** Consulta sobre interfaces externas 46](#_Toc495533394)

[**Tabla 19**. Consultas en el sistema 47](#_Toc495533395)

[**Tabla 20.** Usabilidad en el sistema 47](#_Toc495533396)

[**Tabla 21.** Decisiones envío de notificaciones 48](#_Toc495533397)

[**Tabla 22.** Reglas de uso para identificadores en el front 54](#_Toc495533398)

[**Tabla 23.** Reglas de uso para identificadores de archivos 55](#_Toc495533399)

[**Tabla 24.** Secuencia de construcción incremento 1 55](#_Toc495533400)

[**Tabla 25**. Secuencia de construcción incremento 2 55](#_Toc495533401)

**LISTA DE ANEXOS**

Anexo Digital 1. Documento de visión y alcance

Anexo Digital 2. Listado Maestro de CAR

Anexo Digital 3. Guía del club CAR

Anexo Digital 4. Política SAM-033

Anexo Digital 5. Especificación de requerimientos de software

Anexo Digital 6. Documento de casos de uso

Anexo Digital 7. Documento de arquitectura

Anexo Digital 8. Documento de diseño detallado

Anexo Digital 9. Manual técnico

Anexo Digital 10. Documento de ejecución de análisis de código estático

Anexo Digital 11. Manual de usuario

Anexo Digital 12. Funcionamiento del sistema

Anexo Digital 13. Modelo entidad relación

Anexo Digital 14. Plan de pruebas detallado

Anexo Digital 15. Casos de prueba.

**GLOSARIO**

**ADO.NET ENTITY FRAMEWORK:** “ADO.NET Entity Framework permite a los desarrolladores crear aplicaciones de acceso a datos programando con un modelo de aplicaciones conceptuales en lugar de programar directamente con un esquema de almacenamiento relacional. El objetivo es reducir la cantidad de código y el mantenimiento necesarios para las aplicaciones orientadas a datos” [1].

**ALERTAS:** es el mecanismo que se activa en el sistema cuando se vence un plazo para la realización de documentación de una solicitud CAR, SAM o QUEJA.

**ANGULAR 2: “**es un framework MVC de JavaScript para el Desarrollo Web Front End que permite crear aplicaciones SPA Single-Page Applications. Entra dentro de la familia de frameworks como BackboneJS o EmberJS” [2].

**APP WEB:** aplicación web que se ejecuta en el servidor de aplicaciones.

**BOOTSTRAP:** “es un framework de front-end gratuito para un desarrollo web más rápido y fácil el cual incluye plantillas de diseño basadas en HTML y CSS para tipografía, formularios, botones, tablas, navegación, modales, carruseles de imágenes y muchos otros, así como complementos JavaScript opcionales. Bootstrap también da la capacidad de crear fácilmente diseños de respuesta” [3].

**CAR:** proceso de análisis y resolución de causas, tiene el propósito de identificar las causas de los problemas y tomar acciones para evitar que ocurra en el futuro.

**CATEGORÍAS:** Clasificación de los niveles de Club CAR de acuerdo con el puntaje obtenido.

**CLUB CAR:** es una comunidad de socios activos que tienen en común su actitud, iniciativa, creatividad y orientación a resolver los problemas oportunamente.

**CONTROLADOR:** componente del lado del cliente que permite realizar la comunicación entre las vistas con los componentes de negocio y transaccionales (Vista y Proveedor).

**C#: “**es un lenguaje de programación que se ha diseñado para compilar diversas aplicaciones que se ejecutan en .NET Framework. C# es simple, eficaz, con seguridad de tipos y orientado a objetos. Las numerosas innovaciones de C# permiten desarrollar aplicaciones rápidamente y mantener la expresividad y elegancia de los lenguajes de estilo de C” [4].

**ESCALAMIENTO:** acción para acceder a niveles superiores en la jerarquización de la compañía.

**FRAMEWORK:** en este contexto, se refiere a un conjunto de API’s de programación que prestan un conjunto de funcionalidad bien definido.

**HTML5: “**esta especificación define la quinta revisión principal del lenguaje principal de la World Wide Web, el lenguaje de marcado de hipertexto (HTML). En esta versión, se introducen nuevas características para ayudar a los autores de aplicaciones web, se introducen nuevos elementos basados en la investigación de las prácticas de creación predominantes y se ha prestado especial atención a definir criterios claros de conformidad para los agentes usuarios en un esfuerzo por mejorar la interoperabilidad” [5].

**JSON:** JavaScript Object Notation, o Notación de Objetos de JavaScript, es un formato ligero de intercambio de datos. Leerlo y escribirlo es simple para humanos, mientras que para las máquinas es simple interpretarlo y generarlo. Está basado en un subconjunto del Lenguaje de Programación JavaScript, JSON es un formato de texto que es completamente independiente del lenguaje, pero utiliza convenciones que son ampliamente conocidos por los programadores de la familia de lenguajes C, incluyendo C, C++, C#, Java, JavaScript, Perl, Python, y muchos otros” [6]

**LOG:** hoja de registro general de información asociada a los CAR

**.NET:** “es un framework de Microsoft que hace un énfasis en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado” [7].

**NOTIFICACIONES:** es el envío de un mensaje predeterminado a través de correo electrónico al Administrador del sistema, a los responsables de un CAR y a sus jefes inmediatos en determinadas ocasiones.

**QUEJAS:** es un tipo de solicitud, con el cuál el cliente tanto interno como externo puede presentar una queja o reclamo.

**RESPONSABLE DEL CAR:** persona encargada de liderar el CAR.

**SAM:** es una solicitud de acción de mejora de un proceso de la compañía

**PATRONES ARQUITECTÓNICOS:** los patrones arquitectónicos se utilizan para expresar una estructura de organización base o esquema para un software. Proporcionando un conjunto de sub-sistemas predefinidos, especificando sus responsabilidades, reglas, directrices que determinan la organización, comunicación, interacción y relaciones entre ellos. [8]

# INTRODUCCIÓN

La capacidad que tienen las empresas para identificar los problemas y oportunidades de mejora dentro de sus procesos organizacionales, así como los métodos instaurados para dar solución y generar un impacto positivo a través de los mismos, son factores que influyen en el crecimiento y proyección de las empresas en el mercado, especialmente en las empresas relacionadas con el desarrollo de software y el sector TI, cuyos procesos cambian frecuentemente debido al rápido crecimiento de la tecnología.

Asesoftware es una empresa de desarrollo y mantenimiento de software cuyas soluciones de TI se destacan por su efectividad, alta calidad y servicios enfocados en desarrollar soluciones ajustadas a las necesidades particulares del cliente, optimizando su negocio y generando ventajas competitivas por medio de la tecnología. Este éxito se debe a que los resultados que tengan un gran impacto en los proyectos o procesos de Asesoftware, así como en sus objetivos estratégicos, deben seguir un proceso formal de análisis y resolución de causas. Esto con el fin de minimizar la probabilidad de que los defectos o problemas se vuelvan a presentar en un futuro y que los éxitos se puedan replicar para mejorar el futuro desempeño de los procesos.

Debido a la búsqueda del mejoramiento continuo en los procesos y servicios de la empresa, es necesario contar con un sistema que permita realizar la gestión y seguimiento del proceso de análisis y resolución de causas (CAR). Este documento expone el proceso de desarrollo del sistema de CAR de la empresa Asesoftware, el cual ha sido diseñado y desarrollado para ser utilizado por parte del área de Planeación y de todos los miembros de la compañía.

En este documento se presenta el problema identificado junto a los antecedentes, la formulación y la sistematización del mismo; se define el objetivo y la justificación del desarrollo del sistema en base al problema enunciado; posteriormente se encuentran los fundamentos teóricos que permiten describir los procesos que se ven involucrados en la metodología de análisis y resolución de causas. Seguidamente, se incluye la metodología que se divide en: la identificación de necesidades y objetivos del negocio, la realización del diseño y el desarrollo del sistema.

# PLANTEAMIENTO DEL PROBLEMA

## ANTECEDENTES

Asesoftware es una compañía con más de 25 años de experiencia brindando soluciones para empresas y entidades de todos los tamaños y sectores empresariales, comprometida con la excelencia en la prestación de servicios, certificada bajo la norma de la Organización Internacional de Normalización (ISO) 9001, lo cual certifica el compromiso con la calidad y asegura la implementación de procesos eficaces con altos niveles de productividad y rentabilidad.

Además, ha adoptado la Integración de Modelos de Madurez de las Capacidades (CMMI), esta valoración ratifica la madurez de los procesos en la empresa y su orientación hacia la innovación y resolución de problemas a través de la gestión cuantitativa[9]. Razón por la cual el área de Planeación de la compañía Asesoftware, en el marco de las funciones propias asociadas a los procesos de perfeccionamiento de la compañía, tiene a cargo un proceso de análisis y resolución de causas denominado (CAR), este describe una metodología sistemática para la detección, seguimiento y solución de problemas u oportunidades de mejora identificados por algún funcionario de la empresa.

Este proceso es adoptado por los miembros de la compañía, el cual contempla una serie de fases que deben ejecutarse sistemáticamente, iniciando por la solicitud de un CAR cuando se identifica un problema u oportunidad de mejora, luego desarrollando el tema del CAR en paralelo con el área de planeación, abordando las fases y estados contemplados en la política SAM-033 [10]; de acuerdo al cumplimiento con las actividades propuestas para cada fase y estado, el proceso se va desarrollando hasta cerrarse o no efectivamente, en la ejecución de un CAR se ve involucrado un proceso de generación de alertas y notificaciones que tiene como objetivo hacer respetar los tiempos establecidos para cada fase y estado.

## DESCRIPCIÓN DEL PROBLEMA

Actualmente el área de Planeación de la empresa Asesoftware adopta el proceso de análisis y resolución de causas como se estipula en la política SAM-033 [10] y en el Log de CAR RAC-49 [11], este último es una hoja electrónica donde se definen algunos atributos esenciales para el proceso, como lo son el identificador único de cada CAR, los estados que pueden ser: abierto, cerrado o completado y las fases que pueden ser: sin documentar, plan de acción, en verificación y completado, entre otros datos que determinan el cumplimiento de cada una de las actividades contempladas en el ciclo de vida de un CAR.

De manera mensual el área de Planeación tiene en seguimiento alrededor de un centenar de CAR, los cuales se encuentran registrados en una hoja de Excel. En la actualidad esta área realiza de manera manual el monitoreo de esos registros cada 15 días para evidenciar si alguno está incumpliendo con los plazos establecidos para realizar alguna actividad, lo cual genera demoras en la identificación de cuáles son los CAR que están irrespetando la política SAM-033 y por tal motivo se genera retardos en las notificaciones que se le deben realizar al funcionario responsable del CAR, además se ve afectado el compromiso de los empleados de la compañía por seguir el proceso adecuadamente al no existir una herramienta que pueda facilitar la visualización y seguimiento de este proceso.

## FORMULACIÓN DEL PROBLEMA

¿Cómo ayudar en la gestión del proceso de análisis y resolución de causas (CAR) en la empresa Asesoftware S.A.S?

## SISTEMATIZACIÓN DEL PROBLEMA

A través de los siguientes interrogantes se pretende abordar la problemática planteada

* ¿Cómo mejorar el proceso de levantamiento de un CAR?
* ¿Cómo facilitar la gestión de cambios en la información de un CAR?
* ¿Cómo optimizar el seguimiento diario a los CAR siguiendo los lineamientos estipulados en la política 033-SAM?

# OBJETIVOS

## OBJETIVO GENERAL

Apoyar la gestión de la metodología de análisis de resolución de causas para mejorar el seguimiento de la misma por parte del área de planeación de la compañía Asesoftware a través del desarrollo de un sistema de gestión.

## OBJETIVOS ESPECÍFICOS

* Establecer una tasa de disminución de hasta 14 días en el retraso de la generación de alertas para los CAR y SAM, realizando el seguimiento diario por parte del sistema.
* Construir el módulo de consulta de información por parte de los implicados (Administrador y responsable) en la gestión de un CAR.
* Diseñar la gestión de los cambios en la información de un CAR (Tema, equipo CAR, responsable del CAR, Fecha estimada de cierre, fase, estado, fecha y resultado de la evaluación de eficacia del CAR).
* Diseñar y Construir el módulo de registro de solicitudes de CAR.

# JUSTIFICACIÓN

El constante crecimiento de la compañía en los últimos años, abriéndose paso en el mercado internacional con presencia en países como Costa Rica, Estados Unidos, Chile y Panamá junto con el nivel de exigencia de sus clientes establecen altos estándares de calidad y compromiso en la implementación de sus procesos, buscando el mejoramiento continuo y el perfeccionamiento de los mismos en base a los problemas u oportunidades de mejora identificados por algún funcionario de la empresa.

Actualmente esta búsqueda de la excelencia se implementa a través del proceso de CAR el cual describe una metodología sistemática para la detección, seguimiento y solución de problemas, la información que genera el uso de este proceso se maneja a través de hojas electrónicas, las cuales residen en un único equipo de cómputo dentro de la compañía, lo que genera poca efectividad en el seguimiento que se realiza por parte del área de Planeación y una inadecuada gestión al proceso, brindando pocas oportunidades a los funcionarios de la compañía de poder iniciar el proceso CAR de una manera ágil, lo cual influye directamente en la baja tasa de defectos encontrados en los procesos de la compañía y pocas oportunidades de mejora dentro de los mismos.

Por este motivo se estableció la necesidad de crear un sistema de gestión que permita realizar un seguimiento efectivo al proceso CAR siguiendo los lineamientos establecidos por las políticas de la compañía, permitiéndole a los funcionarios visualizar el estado en que se encuentran los CAR asignados, facilitando al área de planeación el cambio sobre la información que genera este proceso generando notificaciones que ayuden a finalizar exitosamente cada CAR.

# MARCO REFERENCIAL

## MARCO EMPRESARIAL

Para realizar el desarrollo del sistema de gestión para el análisis y resolución de causas de la empresa Asesoftware, es necesario distinguir los procesos internos que se llevan a cabo en el área de planeación en relación con el proceso CAR, los cuales se describen a continuación.

4.1.1 CAR. Es un proceso empleado por la organización para resolver los problemas de forma sistemática. Esta resolución de problemas es el proceso a través del cual se consigue reconocer las señales que identifican la existencia de una dificultad o irregularidad de desarrollo habitual de una actividad, por esta razón se debe recolectar la información necesaria para solucionar los problemas localizados, identificar las causas raíces y se debe escoger e implementar las mejores alternativas de solución.

4.1.2 Club CAR. Es una comunidad de socios que fue creada para reconocer y premiar la lealtad de los participantes al club y su compromiso al resolver los problemas por medio de la metodología CAR. Es un club en donde los socios tienen en común su actitud, iniciativa, creatividad y orientación a resolver problemas oportunamente.

4.1.3 Política 033 SAM. Todo el personal de Asesoftware cumple con el tiempo estipulado por la organización para realizar el análisis de causas y establecimiento del plan de acción de las Solicitudes de Acción de Mejora (SAM), el cual es de 15 días a partir de la fecha de su levantamiento.

Los SAM deben cerrarse dentro del plazo definido en las fechas de ejecución del plan de acción y su eficacia debe ser evaluada de acuerdo con lo establecido en la descripción del objetivo.

Dado que los SAM son un mecanismo que permiten mejorar continuamente, su tratamiento durante el año será tenido en cuenta en la evaluación de desempeño de cada empleado, así como en todos los programas de beneficios que la empresa emprenda.

4.1.4 Rac 49 Log de CAR. Log de registro asociada a los CAR donde se establecen los requisitos mínimos que debe cumplir cada CAR para poder ser aceptado por parte del área de planeación para comenzar con la gestión del mismo, requisitos como consecutivo, fechas de detección y seguimiento, nombre del responsable, pertenencia a un área o proyecto de la compañía, fechas de cierre, fechas de evaluación de eficacia.

## MARCO CONCEPTUAL

A continuación, se hace una síntesis de los conceptos que abordan las diferentes disciplinas, elementos y principales herramientas utilizadas durante todo el ciclo de desarrollo del proyecto.

4.2.1 Servicio Windows. Un servicio de Microsoft Windows le permite al usuario crear programas que tengan el objetivo de durar ejecutándose por largos periodos de tiempo, un servicio de Windows tiene como característica principal que se inicia automáticamente ante una acción de otro sistema o de manera automática en determinada hora del día, los servicios Windows tienen las características adecuadas para ser ejecutados en un servidor donde no interfieran con las ejecuciones de los usuarios [12].

4.2.2 Entity framework. Es un conjunto de tecnologías de ADO.NET que genera grandes ventajas para los ingenieros de software al permitir trabajar con datos en forma de objetos y propiedades específicos del dominio, dejando a un lado las preocupaciones de trabajar con diferentes motores de bases de datos para almacenar y recuperar los datos, ya que cada uno puede tener sus propios protocolos de funcionamiento, con estas herramientas los desarrolladores también evitan pensar en el modelamiento de sus programas, las relaciones y la lógica que estos deben resolver ya que Entity Framework les permite trabajar a un nivel mayor de abstracción [1].

4.2.3 DTO. Los Objetos de Transferencia de Datos (DTO) se utilizan para recuperar información de una base de datos reduciendo las llamadas que se generan sobre la misma, generalmente se le conoce como un patrón de diseño que permite llevar a cabo transferencia de datos entre diferentes sistemas o aplicaciones de software, este patrón de diseño no tiene un comportamiento diferente a al almacenamiento y recuperación de datos [13]

4.2.4 Typescript. Se define como un lenguaje de código abierto derivado del lenguaje de programación JavaScript, Typescript cuenta con herramientas de programación orientada a objetos y es usado para desarrollar programas de software tanto del lado del cliente como del servidor, el código escrito en este lenguaje de programación al ser compilado se convierte en código JavaScript original [14].

4.2.5 Ensamblados. Los ensamblados son bloques de código agrupados que forman una unidad lógica para la creación de aplicaciones y son utilizados en el framework .NET, un ensamblado contiene los recursos parciales de una aplicación formando una biblioteca de código que constituye un control de versiones, además es altamente reutilizable, también funciona como la unidad lógica en la cual se solicitan y conceden los permisos en el ámbito de seguridad de una aplicación, los ensamblados pueden ser de procesos(exe) o comportarse como bibliotecas de ensamblados(dll) [15].

4.2.6 Patrones arquitectónicos. A continuación, se describen los patrones arquitectónicos que fueron utilizados durante la construcción del sistema.

4.2.6.1 MVC (modelo vista controlador). Este patrón de arquitectura separa los programas de software en tres capas las cuales son: el modelo, la vista y el controlador.

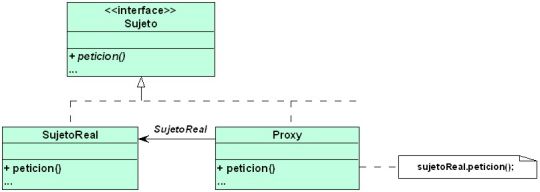
El modelo representa todos los datos de la lógica de la aplicación por ejemplo un objeto obtendrá los datos de una persona de la base de datos los cuales se manipularán e irán de vuelta a la base de datos de nuevo representando la información de esa persona.

La vista se utiliza para la interacción entre un usuario y una aplicación manipulando componentes de interfaz de usuario como botones o formularios y el controlador actúa como una interfaz entre el modelo y la vista procesando todas las peticiones entrantes manipulando los datos a través del modelo e interactuando con la vista para representar una salida final de información [16]

4.2.6.2 Patrón inyección de dependencias. La inyección de dependencias es un patrón de diseño orientado a objetos cuya filosofía principal es inyectar comportamientos a componentes de software también es llamado inyección de comportamientos y su funcionamiento se basa en colocar dentro de un objeto atributos u objetos que puedan cambiar su comportamiento sin que esto implique crear el objeto nuevamente, la inyección de dependencias consiste básicamente en hacer que los componentes de un software sean independientes comunicándose a través de la implementación de una interface, lo cual genera algunos cambios en el código como la instanciación de objetos [17].

4.2.6.3 Patrón proxy. En términos generales, un proxy es una clase que sirve de interface a un elemento computacional difícil de distribuir. En este caso, se deben generar proxies para consumir los servicios de Transferencia de Estado Representacional (REST) y se utiliza en la capa de presentación del sitio Web [18].

**Figura 1.** Patrón de diseño Proxy [22]



4.2.6.4 Patrón repositorio. El patrón de diseño repositorio busca solucionar problemas como la duplicidad del código, los errores potenciales de programación, la dificultad del almacenamiento de datos en caché y la posibilidad de probar la lógica del negocio sin involucrar dependencias externas.

Para solucionar los problemas anteriormente mencionados se utiliza el patrón de repositorio para separar la lógica que recupera los datos y los mapea al modelo, este patrón media entre la capa de origen de base de datos y las demás capas de la aplicación mapeando y persistiendo los datos en las entidades del negocio, esta separación tiene algunas ventajas como la centralización de la lógica de datos y proporciona un punto de sustitución para realizar pruebas unitarias además proporciona flexibilidad en la arquitectura de la aplicación a medida que evoluciona el diseño de la misma.

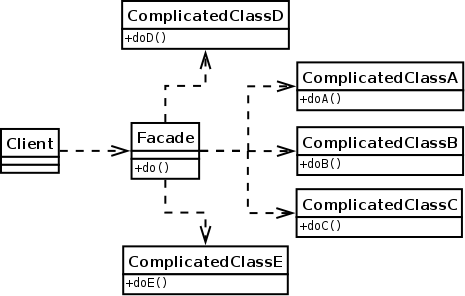
De acuerdo al planteamiento de este patrón de diseño el usuario de este no debería conocer la tecnología que se utiliza para acceder a los datos, solamente sería necesario saber las operaciones que se pueden obtener a través del repositorio [19]

4.2.6.5 Patrón Facade. El patrón de diseño Facade simplifica la comunicación entre una clase origen y una clase o sistema destino u objetivo cuya funcionalidad es muy compleja. Este sistema al tener una complejidad muy alta que requiere de la configuración de muchos parámetros para conseguir el objetivo deseado. Otro problema que se busca solucionar a través de la implementación del patrón de diseño Facade es el uso de diferentes interfaces mediante un único punto de acceso, esto es aplicable cuando el software tiene que implementar algún método o funcionalidad de un sistema complejo o cuando se necesita tener mayor legibilidad sobre una librería de clases.

La solución que busca el patrón Facade es crear una clase fachada que proporcione la funcionalidad del sistema complejo, pero de una manera sencilla al sistema cliente, la clase fachada implementará las operaciones del sistema complejo y los expondrá a través de operaciones más simples.

En la Figura 2 se puede observar el funcionamiento del patrón de diseño, donde client representa el sistema que va a hacer uso del sistema de gran complejidad, Facade representa la clase que va a ofrecer la funcionalidad de una manera más sencilla y legible y complicatedClassX es el sistema o conjunto de clases que se desean acceder a través de la fachada. [20]

**Figura 2.** Patrón de diseño Facade [23]



4.2.6.6 Patrón TAP (Asincrónico basado en tareas). TAP usa un método para representar el inicio y la finalización de una operación asincrónica. Cuando se utiliza el patrón asíncrono en base a tareas del Framework .NET 4.5 los métodos asíncronos devuelven un objeto Task representando la tarea. Utiliza los tipos Task y Task<TResult> del espacio de nombre System.Threading.Tasks para representar operaciones asíncronas arbitrarias [21].

# METODOLOGÍA

Para el desarrollo del proyecto se realizó una investigación aplicada debido a que se encuentra definida por un marco teórico, se centra en el análisis y busca la solución de una situación problemática en la organización que requiere ser estudiada y perfeccionada. La investigación comienza con la descripción sistemática del problema encontrado, luego se establece la teoría necesaria junto con algunos conceptos y definiciones para abordar el problema, el cual se evalúa dentro del marco teórico definido y se procede a proponer acciones de solución. Se utiliza el método de investigación acción participativa donde se busca que todas las personas involucradas en la problemática asuman un rol propositivo con el objetivo de llegar a una solución que satisfaga las necesidades de los mismos.

La investigación corresponde a un tipo de diseño no experimental de tipo transeccional o transversal debido a que la información necesaria para abordar el problema se recolecta en un único momento cuyo propósito no es manipular las variables que interactúan en el sistema, solamente describir su relación en un momento dado. Dentro del diseño de la investigación se establece que esta es de tipo descriptiva debido a que se busca una interpretación correcta a través de tipos de estudio como encuestas.

A continuación, se describe la metodología definida por la empresa Asesoftware S.A.S para el desarrollo de proyectos de software. Para cada actividad que se detalla en esta sección, se realizan actividades como revisiones a cada etapa del proyecto por parte del área de calidad y a cada uno de los documentos que forman parte de los entregables definidos por la compañía.

## DEFINICIÓN DEL PROCESO DE CICLO DE VIDA

En este proceso, se busca obtener el PCV que seguirá el desarrollo, para lo cual se selecciona un patrón de diseño y se realizan las adaptaciones de acuerdo con las características del sistema y las necesidades del cliente. El PCV es una de las bases para determinar el cronograma y los hitos del proyecto, para lo cual se llevan a cabo las siguientes actividades:

5.1.1 Verificar información base. Durante esta actividad se recopila la información y documentación entregada por el cliente, en este caso por el área de Planeación de Asesoftware. Si durante la revisión de la misma se analiza que se puede suplir alguna de las etapas del ciclo de vida del software, se procede a realizar una evaluación de esta información en conjunto con el director de proyecto, el arquitecto de software asignado y el área de calidad de la compañía.

5.1.2 Evaluación de la documentación del cliente. Durante esta actividad se lleva a cabo una inspección detallada sobre la información y documentación entregada por el cliente, generalmente se trata de documentos de requerimientos, historias de usuario, documentos de visión y alcance o prototipos del sistema. A través de esa inspección se deduce si la calidad y completitud de la misma satisface los lineamientos de construcción de software de la compañía y pueden servir de base para alguno de los procesos de desarrollo de software. Si esta documentación no cumple los requisitos mínimos de la inspección realizada, se procede a realizar un control de cambios bajo criterio del director de proyecto.

5.1.3 Seleccionar el MCV. En esta actividad se debe seleccionar el modelo de Ciclo de Vida bajo el cual se va a desarrollar el sistema, se debe tener en cuenta las especificaciones técnicas estipuladas por el cliente, la frecuencia de entrega de módulos funcionales, el presupuesto y las tecnologías bajo las cuales se debe desarrollar el sistema.

5.1.4 Mapear las actividades estándar al patrón de arquitectura seleccionado. Las actividades estándar son aquellas que conforman las fases de Análisis de requerimientos, diseño, implementación, pruebas y transición. Durante el mapeo se pueden adicionar o excluir actividades del estándar. Estas decisiones deben estar debidamente justificadas.

5.1.5 Establecer el PCV. Durante esta actividad se debe determinar en cuantos incrementos se va a desarrollar el sistema y el orden en el cual se van a ejecutar las diferentes actividades. El orden de las actividades debe ser consistente con el patrón de arquitectura seleccionado y con las entradas y salidas de cada una de ellas.

## ANÁLISIS DE REQUERIMIENTOS

Cabe resaltar, que el proceso de desarrollo de software definido por la compañía se encuentra dividido en varios sub procesos. Durante esta actividad se definen junto con el cliente las características que va a tener el sistema, los requerimientos funcionales y los atributos de calidad.

5.2.1 Visión y alcance del proyecto. El objetivo principal de la visión y alcance es garantizar que el proyecto esté alineado con la estrategia del área de Planeación de Asesoftware, y dar una visión global de las funcionalidades que prestará el sistema. El documento en el cual queda plasmado puede ser empleado en fases posteriores del proyecto para delimitar el alcance, resolver conflictos y priorizar funcionalidades y actividades.

5.2.2 Analizar los requerimientos. Durante esta actividad, con la ayuda de algunos artefactos para el levantamiento de requerimientos como entrevistos con el cliente, listos de chequeo, grabaciones y observación directa del proceso involucrado se pretende estimar el sistema que se va a construir. Al finalizar esta actividad se realiza un acta de finalización de hito donde se establece un inventario de entregables, observaciones del cliente, observaciones de Asesoftware y proposiciones o reclamos en caso de existir por parte del cliente.

5.2.3 Documentar los requerimientos. En esta actividad se documentan los requerimientos una vez avalados por el cliente, estos requerimientos quedan plasmados en el documento establecido por Asesoftware denominado “RDS-02 Especificación de requerimiento de software”. Esta actividad incluye además la aprobación del documento por parte del área de calidad

5.2.4 Elaborar el modelo entidad relación. Durante esta actividad se diseña y documenta el modelo entidad relación del sistema, avalado por el arquitecto asignado al proyecto.

5.2.5 Elaborar el modelo funcional. A través de casos de uso se establecen las funcionalidades que debe tener el sistema, estas características se plasman en el documento proporcionado para tal fin denominado “RDS-20 Casos de uso”.

## DISEÑO

En esta fase se detalla el análisis de escenarios de atributos de calidad del sistema y se desacopla el sistema lógicamente en módulos de manera que estos puedan empezar a ser construidos de forma individual.

En la Tabla 1 se puede observar cada una de las actividades a realizar dentro de la etapa de diseño.

**Tabla 1.** Actividades de diseño

| **Actividad** | **Descripción** |
| --- | --- |
| Definir la arquitectura del sistema | En esta actividad se tienen en cuenta todos los aspectos que se han establecido del sistema y a partir de ellos se define la arquitectura que más se adapte de acuerdo con esas condiciones. En Asesoftware esta actividad está a cargo del arquitecto de software encargado del proyecto. |
| Diseñar la base de datos | Con base en el modelo entidad relación definido se elabora la implementación de la base de datos, se documentan cada una de las decisiones tomadas. |
| Diseñar las interfaces externas | Si el sistema a desarrollar se comunica con un sistema ya existente o con una base de datos diferente a la que se implementa, se debe especificar el mecanismo de comunicación que se tenga para cada uno. |
| Elaborar el diseño detallado | En esta actividad se establecen las decisiones de diseño y los estándares de presentación y programación, además de plasmar el flujo de información de cada uno de los módulos a construir. |
| Diseñar el diagrama de clases | Para la vista de desarrollo se elabora el diagrama de clases del sistema. |
| Diseñar el proceso de implementación e integración | En esta actividad el equipo de desarrollo y el arquitecto del proyecto establecen la secuencia de construcción del sistema, especificando el orden de desarrollo de cada una de las funcionalidades o pantallas establecidas. |
| Diseñar las pruebas de funcionalidad | Para cada funcionalidad del sistema se especifica en el documento “RDS-21 Plan de pruebas” los casos de prueba de cada una, está tarea se lleva a cabo en conjunto con los analistas de pruebas de la compañía. |

## CONSTRUCCIÓN

En esta fase del proyecto se usan los documentos realizados, la arquitectura definida y las restricciones del cliente para obtener un producto de software de calidad que exprese cada detalle especificado en el diseño del sistema.

5.4.1 Configurar el ambiente de desarrollo. En esta actividad se instalan las herramientas necesarias para llevar a cabo la construcción del sistema (lenguajes de programación, sistemas gestores de bases de datos, dependencias, etc.)

5.4.2 Codificar los programas de software. Durante esta actividad se desarrollan los módulos del sistema.

5.4.3 Ejecución de análisis de código estático. Se ejecuta el análisis de código estático para cada una de las capas definidas en la arquitectura del sistema dependiendo de la herramienta utilizada para el desarrollo.

5.4.4 Diseño, construcción y ejecución de pruebas funcionales.Para la ejecución de las pruebas de funcionalidad, se tomó como guía el documento de plan de pruebas detallado (Ver Anexo 14).

5.4.5 Ejecutar la integración de los programas de software. Esta actividad se realiza de acuerdo con la estrategia de integración definida en el documento de diseño detallado.

5.4.6 Documentar el producto. Durante esta etapa del proyecto se elabora el manual de usuario, el manual técnico, se ajustan los documentos de arquitectura, de diseño detallado, los documentos de casos de pruebas, el documento de visión y alcance, el documento de especificación de requerimientos.

## TRANSICIÓN

Esta etapa tiene como objetivo instalar el sistema en el ambiente en el cual va a ser operado por los usuarios.

5.5.1 Configurar el ambiente de instalación. Se realiza la configuración de requerimientos de software y/o hardware que sean necesarios para la operación correcta del sistema.

5.5.2 Instalación del sistema. De ser necesario, previamente se debe realizar un documento de instalación el cual debe ser ejecutado por el usuario, si la instalación se lleva a cabo de manera exitosa, existe un documento provisto por la compañía para elaborar el acta de fin de hito.

5.5.3 Ejecutar migración de datos. En caso de que aplique para el sistema a desarrollar, se deben ejecutar los procesos de migración, verificación y depuración de datos.

5.5.4 Capacitación. En caso de ser requerido por el usuario, se debe diseñar el proceso de capacitación como indica el activo de proceso “PCA-01 Capacitación” definido por la compañía.

# DESARROLLO

En el siguiente apartado se describe el desarrollo del sistema en función de la metodología descrita anteriormente.

## DEFINICIÓN DEL PROCESO DE CICLO DE VIDA

6.1.1 Verificación de la información base. A través de las reuniones iniciales que se llevaron a cabo con el área de planeación de la compañía, se pudo establecer que no existía ningún documento que pudiera ayudar a suplir alguna de las fases iniciales del proyecto, por lo cual se procede a iniciar el desarrollo del sistema desde la fase de análisis.

6.1.2 Evaluación de la documentación del cliente. Durante la actividad de verificar la información base se pudo establecer que existen documentos asociados al proceso de análisis y resolución de causas de la compañía, los cuales se encuentran establecidos en los activos de proceso de la compañía y son de uso corporativo.

6.1.3 Seleccionar el MVC**.** De acuerdo con el tiempo establecido para el desarrollo y la necesidad del cliente de tener rápidamente una funcionalidad parcial del sistema se decide utilizar el modelo incremental como modelo de desarrollo de software.

6.1.4 Establecer el MVC. El desarrollo del proyecto se llevó a cabo a través de dos incrementos los cuales estuvieron marcados con una duración de 2 meses cada uno en los cuales se llevaron a cabo todas las fases que comprenden el ciclo de vida del desarrollo de software para la compañía Asesoftware, los cuales fueron análisis, diseño, construcción, pruebas. La etapa de transición se realizó fuera de los dos meses, debido al proceso legal que conlleva la puesta en marcha del sistema.

En la Tabla 2 se puede evidenciar la distribución de tiempo para cada una de las etapas del desarrollo del sistema.

**Tabla 2.** Modelo de desarrollo

| **Incremento** | **Etapa** | **Actividad** | **Duración** |
| --- | --- | --- | --- |
| INCREMENTO 1 | ANÁLISIS | Entrevistas | 30 días |
| Documento de visión y alcance |
| Documento de análisis |
| Elaboración MER |
| Revisión CM |
| Revisión de análisis |
| Revisión MER |
| Ajustes análisis |
| Ajustes MER |
| Fin hito de análisis |
| Diseño | Diseño de arquitectura | 10 días |
| Diseño de base de datos |
| Diseño detallado |
| Ajustes revisión de diseño |
| Construcción | Configuración de ambiente | 20 días |
| Construcción |
| Pruebas unitarias |
| Inspección de código |
| Ajustes inspección de código |
| Pruebas | Planeación de pruebas | 5 días |
| Diseño casos de pruebas |
| Revisión diseño casos de prueba |
| Ajuste diseño casos de prueba |
| Ejecución y verificación pruebas funcionales |
| INCREMENTO 2 | Análisis | Entrevistas | 10 días |
| Ajustes documento de visión y alcance |
| Ajustes documento de análisis |
| Ajustes MER |
| Revisión cm |
| Revisión de análisis |
| Revisión MER |
| Fin hito de análisis |
| Diseño | Ajustes diseño de arquitectura | 5 días |
| Ajustes diseño de base de datos |
| Ajustes diseño detallado |
| Ajustes revisión de diseño |
| Construcción | Construcción | 20 días |
| Pruebas unitarias |
| Inspección de código |
| Ajustes inspección de código |
| Pruebas | Planeación de pruebas | 15 días |
| Diseño casos de pruebas |
| Revisión diseño casos de prueba |
| Ajuste diseño casos de prueba |
| Ejecución y verificación pruebas funcionales |
| Documentación | Manual de usuario | 5 días |
| Manual técnico |
| Revisión documentación |
| Ajustes de revisión de manuales |
| Transición | Despliegue | 3 días |
| Preparación entrega |
| Revisión cm |
| Ajustes y entrega |
| Fin hito construcción |

## ANÁLISIS DE LOS REQUERIMIENTOS

De acuerdo con el proceso de desarrollo de software establecido por Asesoftware, la fase de análisis de requerimientos se descompone en varias actividades las cuales se utilizan como un apoyo fundamental para establecer clara y concretamente las necesidades del cliente.

6.2.1 Visión y alcance del proyecto. El objetivo de la visión y alcance es garantizar que el proyecto esté alineado con la estrategia del área de Planeación de Asesoftware, y dar una visión global de las funcionalidades que prestará el sistema. El alcance puede ser empleado en fases posteriores del proyecto para resolver conflictos, priorizar funcionalidades y actividades (Ver Anexo digital 1).

6.2.1.1 Visión. Dirigido especialmente a las áreas de Planeación y Aseguramiento de Calidad de la compañía Asesoftware, quienes presentan dificultades en la gestión de los CAR. El sistema CARCONTROL (Control de CAR), es una plataforma de gestión en línea que permitirá tener un mayor control de las fases de un CAR, generando alertas durante la fase de “Sin documentar” para llegar a una solución eficiente, disminuyendo el esfuerzo invertido por el Administrador del área a cargo, facilitando la organización y actualización de los registros existentes, brindando a la compañía un mejor desempeño en el proceso de CAR.

6.2.1.2 Alcance de la versión inicial. La herramienta CARCONTROL permite gestionar y administrar el flujo de información correspondiente al seguimiento del proceso de CAR el proceso inicial que se implementará con la herramienta es el registro de la documentación de alguno de los tipos mencionado anteriormente, teniendo en cuenta que dicho registro puede ser realizado por cualquier funcionario de la compañía.

Se asignarán distintos estados de control, relacionados directamente al criterio del Administrador, quien es el responsable de monitorear el progreso de los CAR.

CARCONTROL contará con la funcionalidad de enviar alertas a los responsables de los CAR (ver política 033), a través de mensajes de correo electrónico, hasta que se alcance la fase de “plan de acción”, así como alertar sobre los cambios de estado y fase.

CARCONTROL permitirá a los responsables de cada tipo de solicitud visualizar el historial de los CAR, de los cuales están a cargo y el proceso por el cual ha pasado cada uno.

CARCONTROL permitirá consultar un listado de todos los miembros de la compañía, quienes han levantado un CAR, indicando el puntaje acumulado de todos los CAR realizados por cada usuario responsable del mismo, permitiendo al Administrador del sistema visualizar los puntajes según lo establecido en la guía GAC-10 Guía Club CAR (Ver Anexo Digital 3).

Finalmente se realizará una capacitación técnica para la instalación y administración de la herramienta, y una capacitación funcional para sus usuarios.

6.2.1.3 Características principales. Las características principales de la aplicación se describen en la Tabla 3.

**Tabla 3.** Características del Sistema

| **Identificador** | **Descripción** |
| --- | --- |
| **C01** | Permitir el levantamiento de cada CAR por parte de la persona que lo identificó, registrando los datos pertinentes que puedan agilizar el trámite |
| **C02** | Aseguramiento de la información según el tipo de usuarios involucrados en el proceso, protegiendo la consulta y edición de los datos de cada CAR de acuerdo con las responsabilidades de dichos usuarios en la compañía |
| **C03** | Registrar las observaciones de los cambios de cada CAR por parte del Administrador, luego de realizar el seguimiento periódico. |
| **C04** | Generar notificaciones dando aviso de alertas a los usuarios y a los jefes inmediatos del responsable de cada CAR de acuerdo con la política 033. |
| **C05** | Visualizar cada uno de los registros de los CAR ordenados por fecha (del más antiguo al más reciente), de acuerdo con cada uno de los estados en los que se encuentren, así como permitir la búsqueda de un CAR por su consecutivo. |
| **C06** | Permitir la visualización del historial de los estados por los cuales ha pasado cada uno de los CAR en seguimiento. |
| **C07** | Permitir al Administrador del sistema exportar el log de registros (establecido en el RAC-49 Log de CAR), a formato Excel, el cual contiene la información de todos los CAR existentes en su último estado. |
| **C08** | Permitir al Administrador del sistema exportar el registro de puntuación de Club CAR (establecido en la guía GAC-10 Guía Club CAR), a formato Excel. |

6.2.1.4 Limitaciones y exclusiones. Inicialmente la funcionalidad de la aplicación CARCONTROL estará alineada con las responsabilidades de la persona encargada del seguimiento de los CAR (Administrador, el cual pertenecerá al área de Planeación), por lo que no se contemplarán las actividades realizadas por otros usuarios pertenecientes a áreas diferentes a Planeación; por lo tanto, se establece que en el momento en que un CAR pase a estado de “Plan de Acción” se finaliza el proceso de generación de alertas a los mismos a través de CARCONTROL.

Se aclara que CARCONTROL no permitirá almacenar documentos, dado que la compañía Asesoftware ya cuenta con un repositorio de documentos (SharePoint).

6.2.2 Dilucidar y analizar los requerimientos. Con el objetivo de realizar el análisis y especificación de requerimientos para el proyecto, se identificaron los documentos de procesos que intervienen en el proceso CAR en la compañía.

La documentación existente relacionada al negocio del proyecto está ligada al ciclo de vida del proceso CAR, entre la documentación se encuentra el listado maestro de CAR donde se tiene el consolidado de todos los CAR en sus diferentes fases y etapas (Ver anexo digital 2), la guía del Club CAR donde se establecen los criterios de calificación y premiación a los empleados por su lealtad a la metodología CAR (Ver anexo digital 3) y la política 033-SAM donde se establecen los lineamientos organizacionales para dar tratamiento a los SAM y CAR (Ver anexo digital 4).

6.2.3 Especificación de requerimientos. Durante la fase de especificación de requerimientos se llegó a un acuerdo junto con el cliente (área de planeación) sobre las funcionalidades con las cuales debía contar el sistema de gestión, estos requerimientos y los detalles de esta fase se encuentran documentados en el activo de proceso de especificación de requerimientos de software (Ver anexo digital 5).

En la Figura 3 se aprecian las funcionalidades establecidas para el sistema de gestión de CAR.

**Figura 3**. Funcionalidades del sistema

DIAGRAMA_FUNCIONAL

6.2.3.1 Registro de información CAR. El sistema permitirá a los miembros de la compañía registrar la solicitud de un CAR definiendo los parámetros que se establecen en el log de CAR, además permitirá al administrador del sistema (área de planeación) aprobar o rechazar las solicitudes creadas.

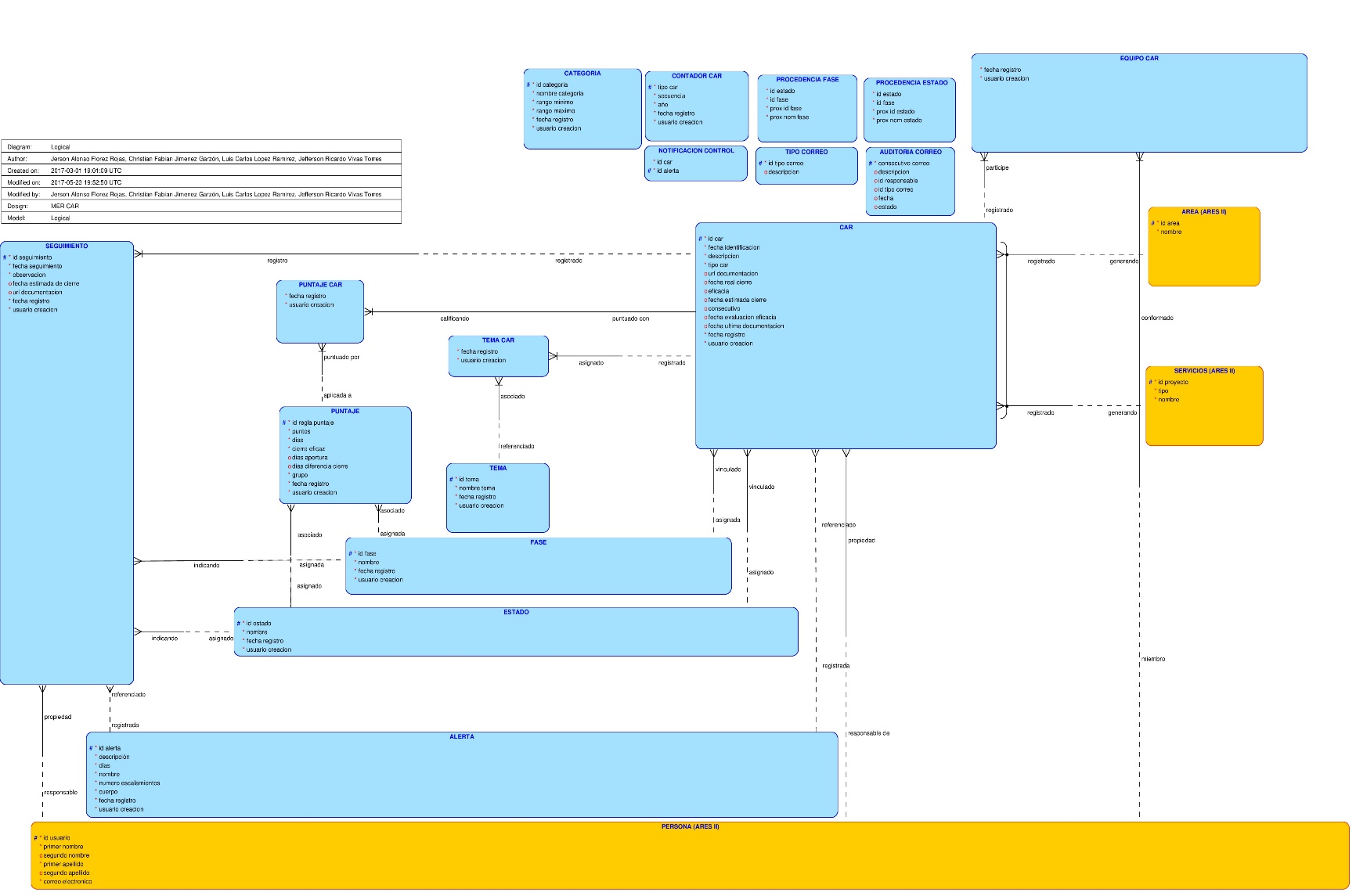
6.2.3.2 Gestionar CAR. El sistema permitirá a los miembros de la compañía enviar la documentación de sus CAR al área de planeación para su posterior evaluación y definir si el CAR avanza o no en su ciclo de vida normal a través de la asignación de estados hasta llegar al estado de completado y de fase terminado, además el sistema permitirá la generación de alertas y de notificaciones vía correo electrónico al responsable de un CAR especificando el estado en el que se encuentra y la posible repercusión que tenga de acuerdo a los plazos establecidos en las políticas de la compañía, además de brindar la funcionalidad de poder asignar un nuevo responsable y un nuevo equipo de trabajo a un CAR.

6.2.3.3 Consultar CAR. El sistema permitirá realizar consultas sobre la información de los CAR a diferentes niveles de detalle y según sea el tipo de usuario que acceda a la aplicación, el administrador tendrá la posibilidad de tener diferentes criterios de búsqueda de información, los cuales van desde los estados y fases de un CAR, hasta la fecha de identificación, responsable, área o proyecto asociado, esto debido a la cantidad de registros que pueden existir en un determinado momento.

El usuario tendrá un nivel de detalle menor debido a que la cantidad máxima de registros que puede llegar a manejar en un determinado momento es reducida. El sistema permitirá al administrador del sistema y a los usuarios en general visualizar el Club CAR donde se mostrarán los puntajes parciales y totales para cada empleado de la compañía según su desempeño mensual.

6.2.4 Elaborar el modelo entidad relación. La elaboración del modelo entidad relación se realizó teniendo en cuenta los sistemas externos con los cuales debe tener comunicación la aplicación, en este caso con el sistema ARES II, el cual es una herramienta de Asesoftware que se utiliza para el seguimiento de proyectos, personas y actividades. En el modelo se identifican las entidades que pertenecen a la base de datos de ARES II como se puede ver en la Figura 4 (Ver Anexo Digital 13).

**Figura 4.** Modelo entidad relación



6.2.5 Elaborar el modelo funcional. Para la elaboración del modelo funcional se escogió la elaboración del modelo de descomposición funcional, así como la elaboración de casos de uso para cada uno de los requerimientos planteados para el sistema, los cuales se pueden encontrar en detalle en el documento de casos de uso (Ver Anexo digital 6).

## Elaboración del diseño.

6.3.1 Definición de la arquitectura del sistema. La arquitectura del sistema se define en base a los atributos de calidad, como características que debe cumplir el producto para poder ser aceptado, además de tener en cuenta los sistemas con los cuales debe comunicarse la aplicación y las restricciones puestas por el cliente (Ver Anexo digital 7).

6.3.1.1 Objetivos del negocio. En la Tabla 4 se observan los objetivos del sistema orientados al negocio, los cuales están alineados con la misión 20/20 de la compañía, la cual busca potenciar cada una de las áreas de la empresa en pro de mejorar sus procesos organizacionales.

**Tabla 4.** Objetivos del negocio

|  |  |
| --- | --- |
| **ID** | **Descripción del objetivo de negocio** |
| OBJ0 | Disminuir hasta 14 días el retraso en la generación de alertas para los CAR, realizando el seguimiento diario de las alertas por parte del sistema. |
| OBJ1 | Permitir la consulta de información por parte de los implicados (Administrador y responsable de CAR) en la gestión de un CAR. |
| OBJ2 | Facilitar el proceso de asignación de los estados y fases de un CAR. |
| OBJ3 | Descentralizar el levantamiento de los CAR, permitiendo a distintos funcionarios de la compañía generar los mismos. |
| OBJ4 | Facilitar la gestión de cambios en la información de un CAR (Tema, Equipo CAR, Responsable del CAR, Fecha estimada de cierre, Fecha y resultado de la evaluación de la eficacia del CAR). |

6.3.1.2 Características del sistema. En la Tabla 5 se puede observar cada una de las características implementadas en el sistema junto con la prioridad que tiene dentro del proyecto y el objetivo con el cual está alineado.

**Tabla 5**. Características del sistema

|  |  |  |  |
| --- | --- | --- | --- |
| **ID** | **Descripción** | **Prioridad** | **Objetivo de Negocio Asociado** |
| F01 | Registro Información CAR |  |  |
| F01.1 | Realizar registro de solicitud del CAR | Alta | OBJ3 |
| F01.2 | Aprobar Solicitud de CAR | Media | OBJ2 – OBJ3 |
| F02 | Gestión de CAR |  |  |
| F02.1 | Envío URL Documentación | Alta | OBJ2 |
| F02.2 | Asignación de estados | Alta | OBJ2-OBJ4 |
| F02.3 | Generación de alertas | Alta | OBJ0 |
| F02.4 | Cambio de responsable | Media | OBJ2 – OBJ1-OBJ4 |
| F03 | Consulta de CAR |  |  |
| F03.1 | Consulta general CAR | Alta | OBJ1 |
| F03.2 | Consulta detallada CAR | Alta | OBJ1 |
| F03.3 | Consulta histórico | Media | OBJ1 |

6.3.1.3 Identificación de los Stakeholders. En la Tabla 6 se observa la descripción de cada uno de los interesados del sistema:

**Tabla 6.** Identificación de stakeholders

|  |  |  |
| --- | --- | --- |
| **Stakeholder** | **Perfil** | **Intereses** |
| Usuario solicitante del CAR | Miembro de Asesoftware | Solicitar el levantamiento y llevar a cabo la consulta del seguimiento de un CAR. |
| Administrador del sistema. | Miembro del área de Planeación. | Aprobar el levantamiento y llevar acabo el seguimiento de un CAR. De igual manera consultar el listado del Club CAR. |

6.3.1.4 Decisiones de arquitectura.

* Framework de diseño de estilos: se hace necesario el uso de un framework que permita gestionar los estilos de las pantallas de la aplicación. En la Tabla 7 se puede observar los criterios de evaluación para la elección del framework de diseño.

**Tabla 7.** Criterios de evaluación frameworks de diseño

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Análisis** | **Puntuación** |
| CR1 | El framework utilizado debe ser compatible con la mayor cantidad de navegadores web. | 3 |
| CR2 | El framework debe contar con gran variedad de componentes disponibles y adaptables a los colores institucionales de Asesoftware | 4 |
| CR3 | El framework debe ser fácil de implementar y adaptable a las restricciones de uso de Angular y .NET | 5 |
| CR4 | El framework debe tener soporte y documentación para su uso. | 3 |

Alternativas de solución diseño de estilos: para el diseño de los estilos del sistema se tienen los siguientes frameworks: Bootstrap, Foundation, Materialize, Bootswach.

**Tabla 8**. Evaluación framework de diseño bootstrap

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Análisis** | **Puntuación** |
| CR1 | Algunas funcionalidades de Bootstrap no son soportadas en Internet Explorer 8.  Es compatible con el navegador Chrome en Android, iOS, Mac OS X, Windows y Linux.  Es compatible con Safari en iOS y Mac OS X.  Es compatible con Firefox en Mac OS X y Windows. | 4 |
| CR2 | Cuenta con 21 tipos de componentes, sin embargo, los colores son limitados y no es posible realizar modificaciones. | 2 |
| CR3 | Para utilizar Bootstrap se debe descargar una carpeta que contiene compilados y CSS, se incluye en el proyecto y así se podrán usar sus funcionalidades. | 4 |
| CR4 | En la página web de Bootstrap se encuentra documentación de cada una de las funcionalidades que se pueden implementar con ejemplos y plantillas. | 5 |

**Tabla 9.** Evaluación framework de diseño foundation

| **Criterio** | **Análisis** | **Puntuación** |
| --- | --- | --- |
| CR1 | Es compatible con el navegador Chrome en Android, iOS, Mac OS X, Windows y Linux.  Es compatible con Safari en iOS y Mac OS X.  Es compatible con Firefox en Mac OS X y Windows.  Es compatible con Internet Explorer 9 o superiores. | 3 |
| CR2 | Tiene 30 tipos de componentes y es posible cambiar la paleta de colores. | 5 |
| CR3 | Para su uso se incluye un archivo plano y no son necesarias clases adicionales. | 5 |
| CR4 | En la página web de Foundation existe documentación para cada componente, sin embargo, no hay ejemplos. | 3 |

**Tabla 10.** Evaluación framework de diseño materialize

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Análisis** | **Puntuación** |
| CR1 | No se encontró información acerca de los navegadores. | 1 |
| CR2 | Tiene 26 tipos de componentes y es posible cambiar la paleta de colores. | 5 |
| CR3 | Se debe descargar una carpeta que contiene compilados y CSS, se incluye en el proyecto y así se podrán usar sus funcionalidades. | 4 |
| CR4 | En la página WEB hay documentación de los componentes con plantillas y ejemplos | 5 |

**Tabla 11.** Evaluación framework de diseño bootswatch

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Análisis** | **Puntuación** |
| CR1 | Algunas funcionalidades de Bootstrap no son soportadas en Internet Explorer 8.  Es compatible con el navegador Chrome en Android, iOS, Mac OS X, Windows y Linux.  Es compatible con Safari en iOS y Mac OS X.  Es compatible con Firefox en Mac OS X y Windows. | 4 |
| CR2 | Cuenta con 21 tipos de componentes de Bootstrap y es posible realizar modificaciones. | 5 |
| CR3 | Para utilizar Bootswatch se debe descargar un archivo CSS y reemplazar el de Bootstrap | 4 |
| CR4 | En la página web se encuentra documentación de cada una de las funcionalidades que se pueden implementar con ejemplos y plantillas. | 5 |

* Evaluación diseño de estilos: De acuerdo con la puntuación anterior, se toma la decisión de hacer uso de Bootswatch como framework de diseño Web como se puede observar en la Tabla 12.

**Tabla 12.** Elección framework de diseño

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **CR1(3)** | **CR2(4)** | **CR3(5)** | **CR4(3)** | **Total** |
| **Bootstrap** | 4 | 2 | 4 | 5 | 55 |
| **Foundation** | 3 | 5 | 5 | 3 | 63 |
| **Materialize** | 1 | 5 | 4 | 5 | 58 |
| **Bootswatch** | 4 | 5 | 4 | 5 | 67 |

* Notificaciones: Se necesita notificar a los usuarios sobre algunos procesos o actividades que no han sido completados en las fechas establecidas. En la Tabla 13 se puede observar los criterios de evaluación para el manejo de notificaciones del sistema.

**Tabla 13.** Criterios de evaluación manejo de notificaciones

|  |  |
| --- | --- |
| **Variable** | **Valor** |
| Costo (C) | 5 |
| Efectividad para los usuarios (EU) | 5 |
| Tiempo de respuesta (TR) | 3 |
| Facilidad de implementación (FI) | 4 |
| Multiplataforma (M) | 3 |

* + Alternativas de evaluación manejo de notificaciones: Dentro de las alternativas de solución definidas para informar a los usuarios, se encuentran las siguientes: Correo electrónico, mensajes de texto.
  + Evaluación y selección de manejo de notificaciones: Dentro de las alternativas propuestas, el mayor puntaje de acuerdo con los criterios de calificación lo obtuvo el envío de correos electrónicos, por lo cual este medio se utilizará en las funcionalidades correspondientes del sistema.

**Tabla 14.** Elección manejo de notificaciones

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **criterio** | **C(5)** | **EU(5)** | **TR(3)** | **FI(4)** | **M(3)** | **Total** |
| Mensajes de texto | 2 | 4 | 4 | 3 | 2 | 60 |
| Correo electrónico | 4 | 2 | 3 | 4 | 4 | 67 |

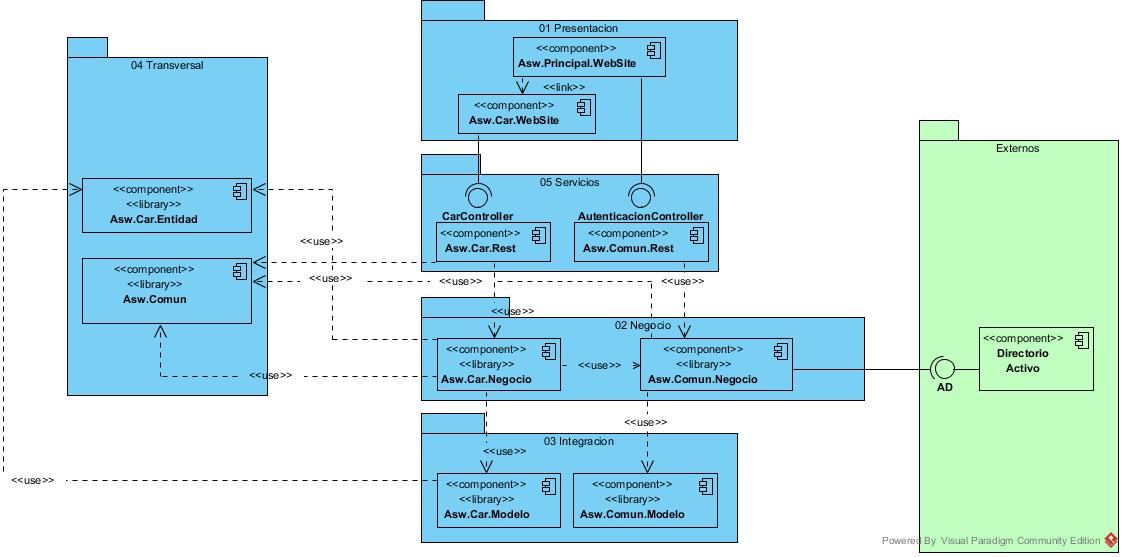
* Riesgos de la solución: Se identifican los siguientes riesgos para la solución planteada:
  + Que las personas no revisen estos correos de notificación, y que los tomen como spam.
  + Que no se establezca de forma clara la política de envíos y se transforme en un bombardeo de correos electrónicos enviados saturando las cuentas.
* Manejo de excepciones: el manejo de excepciones se realizará bajo los siguientes parámetros:
  + Los mensajes de fallos o éxito de una operación al consumir los servicios rest desde los aplicativos rest se deben presentar de forma amigable al usuario.
  + Los mensajes se enviarán a través de una clase Mensaje que contendrá los atributos que se plantean en la Tabla 15.

**Tabla 15.** Manejo de excepciones

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Tipo** | **Valores** |
| Código | Entero | 200 “ejecuto bien” |
| 500 “Error en la ejecución” |
| Descripción | String | Operación Exitosa |
| Operación Errónea |
| Dato | object | Puede almacenar cualquier tipo de dato |

* Framework de persistencia de datos: La implementación de la capa de persistencia se basará en ADO.Net Entity Framework 6.0, y como es con Oracle, se debe instalar el conector ODT de Oracle. Además, se debe tener en cuenta las siguientes recomendaciones:
  + Las operaciones de consulta se realizan por procedimientos almacenados a menos que solo sea una entidad que no utilice *Include* y que no se busque por su llave primaria.
  + Las operaciones de inserción, actualización y eliminación se implementan con procedimientos almacenados.
  + En las entidades se remueven las propiedades de navegación
* Autenticación y autorización: La autenticación y autorización se van a manejar bajo los siguientes parámetros:
  + La autenticación se realizará mediante el directorio activo con el cual cuenta la compañía.
  + Dentro del alcance de la aplicación no se va a realizar el aseguramiento en los servicios rest pero es necesario en implementaciones futuras trabajar con JWT (JsonWebToken).
  + Para definir los permisos de acceso a las funcionalidades, se realizará mediante el acceso a páginas y/o a nivel de controles.
  + Se va a contar con roles fijos dependiendo de la funcionalidad.
* Vistas de arquitectura: para dar una visión de la estructura del proyecto, se presentan las siguientes vistas de arquitectura.
  + Vista funcional.

**Figura 5.** Vista funcional del sistema

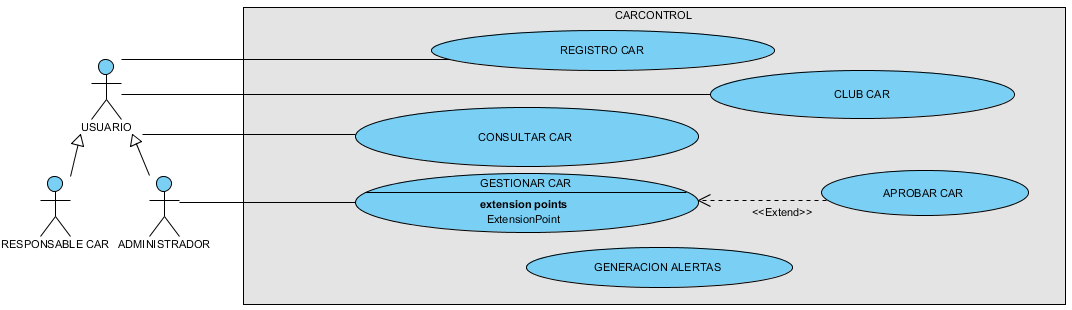


**Tabla 16.** Descripción vista funcional del sistema

| **Elemento** | **Función o Responsabilidades** |
| --- | --- |
| Asw.Principal.WebSite | Página web inicial donde se encuentra el módulo de autenticación, autorización, y el menú inicial de todas las aplicaciones internas de Asesoftware.  Para enviar y recibir los datos esta página web consumirá el componente **Asw.Comun.Rest.**  Accede mediante un link al componente **Asw.Car.WebSite.** |
| Asw.Car.WebSite | Página web desde donde se expondrán todas las funcionalidades para realizar la gestión de los CAR.  Para enviar y recibir los datos esta página web consumirá el componente **Asw.Car.Rest.** |
| Asw.Car.Rest | Servicio rest que permitirá exponer todas las operaciones de actualización inserción, eliminación y consulta que se realizarán sobre la gestión de los CAR.  Este componente consumirá estas operaciones a través del componente de negocio **Asw.Car.Negocio**. |
| Asw.Comun.Rest | Servicio rest que permitirá exponer todas las operaciones comunes a los proyectos internos de Asesoftware, para este caso para los proyectos:   * CAR   Entre las operaciones expuestas, inicialmente se desarrollará la autenticación, y la autorización de los usuarios.  Este componente consumirá estas operaciones a través del componente de negocio **Asw.Comun.Negocio**. |
| Asw.Car.Negocio | Librería de clases donde se encontrarán las reglas del negocio y operaciones sobre la aplicación CAR, y que funciona como fachada a las solicitudes realizadas desde los servicios rest.  Puede consumir operaciones comunes del componente **Asw.Comun.Negocio.**  Puede realizar operaciones sobre la base de datos a través del componente **Asw.Car.Modelo.**  Para realizar el transporte de información y usar métodos utilitarios se hace uso de los componente **Asw.Car.Entidad** y **Asw.Comun**. |
| Asw.Comun.Negocio | Librería de clases donde se encontrarán las reglas del negocio y operaciones comunes a las aplicaciones internas de Asesoftware (CAR).  Puede realizar operaciones sobre la base de datos a través del componente **Asw.Comun.Modelo.**  Para realizar el transporte de información y usar métodos utilitarios se hace uso del componente **Asw.Comun.** |
| Asw.Car.Modelo | Librería de clase donde se encuentra el modelo y el contexto de la base de datos. Mediante entity framework, esta librería permitirá acceder a la base de datos (**Esquema CAR y Ares2**) y realizar las operaciones de consulta, inserción, actualización y eliminación.  Esta librería hace uso de componente **Asw.Car.Entidad,** donde se encuentran las entidades de la aplicación. |
| Asw.Comun.Modelo | Librería de clase donde se encuentra el modelo y el contexto de la base de datos. Mediante entity framework, esta librería permitirá acceder a la base de datos (**Esquema Comun**) y realizar las operaciones de consulta, inserción, actualización y eliminación. |
| Directorio Activo | Directorio donde se encuentran almacenados los usuarios de la organización con sus roles, este se va a consumir para realizar la autenticación a través del componente Asw.Comun.Negocio.  Este componente ya se encuentra implementado. |
| Asw.Car.Entidad | Librería de clase donde se encontrarán las entidades del proyecto CAR, este componente es transversal ya que se utilizará en varios componentes del proyecto |
| Asw.Comun | Librería de clase donde se encontrarán las entidades y métodos comunes a los proyectos internos (CAR, Asignación y Mejoras), este componente es transversal debido a que se utilizará en varios componentes del proyecto |

* + Vista de contexto.

**Figura 6**. Vista de contexto del sistema



**Tabla 17.** Vista de contexto del sistema

| **Elemento** | **Función o Responsabilidades** |
| --- | --- |
| Registro CAR | Permitir el levantamiento de cada CAR por parte de la persona que lo identificó, registrando los datos pertinentes que puedan agilizar el trámite. |
| Consultar CAR | Permitir al administrador y a los responsables asignados a uno o más CAR visualizar los cambios generados durante la gestión de este proceso. |
| Gestionar CAR | Permite al administrador del sistema modificar los cambios de estados de cada una de las etapas durante el seguimiento de cada CAR. |
| Club CAR | Este proceso permite a los usuarios del sistema generar y exportar a Excel una tabla en la que se indican los puntos asignados a los CAR de cada persona, validando plazos de cierre, documentación y eficacia. |
| Aprobar CAR | Permitir al Administrador visualizar las nuevas solicitudes de CAR, para poder aceptarla o rechazarla. Si una solicitud es aprobada, el sistema permitirá la gestión del CAR, para ejecutar la asignación necesaria de fase y estado iniciales. |
| Generación Alertas | Generar notificaciones dando aviso de alertas a los usuarios y a los jefes inmediatos del responsable de cada CAR de acuerdo a la política 033. |

6.3.2 Diseño de la base de datos. El diseño de la base de datos se realizó teniendo en cuenta el modelo entidad relación y los estándares de nombramientos definidos por la compañía para cualquier desarrollo de software que involucre un modelo de base de datos relacional.

A continuación, se presentan los estándares definidos por Asesoftware.

* Tablas y columnas: el tamaño no está definido, este debería ser significativo a la información que se desea almacenar, el recomendado es de 30 caracteres.
* Restricción de tipo check: se debe nombrar chk\_<validación>.
* Llave primaria: PK\_<tabla>\_<columnas>.
* Llave foránea: FK\_<detalle>\_<maestro>\_#. Se usará el mismo nombre de la tabla sobre la cual se está definiendo, \_# solo se usa cuando entre las dos tablas existe más de una relación.

En cuanto al uso de arcos, se establecen dos estrategias diferentes para resolverlos

* Explícito: se emplea cuando las llaves foráneas no tienen el mismo dominio. Se crea una columna de llave foránea por cada relación en el arco.
* Genérico: se emplea cuando las llaves foráneas tienen el mismo dominio. Se crea una sola columna que generaliza la integridad referencial respetando la opcionalidad y se crea una columna para indicar cuál es la relación valida en cada instancia.

6.3.3 Estilos y tácticas de resolución de arquitectura.

En esta sección, se presentan los escenarios identificados que pueden generar impacto en la arquitectura del sistema.

6.3.3.1 Consultas sobre la tabla Ares II.

**Tabla 18.** Consulta sobre interfaces externas

|  |  |
| --- | --- |
| **Escenarios analizados** | Escenarios de integración con ares 2 |
| **Directrices arquitectónicas** | Usar un método de conexión que permita tener la integración con las tablas de ares 2. |
| **Estilos, tácticas y raciocinio** | Crear los esquemas de base de datos de los aplicativos sobre la misma plataforma de base de datos de ares 2 (Oracle), para así permitir accesar de forma más nativa a la información |
| **Riesgos** | Ninguno |
| **Ventajas y desventajas** | (+) Mejor rendimiento en los tiempos de respuesta de las consultas.  (+) Se trabaja sobre plataforma de base de datos por defecto de la compañía.  (-) No se cumpla con los conocimientos básicos en sql server que es la plataforma definida inicialmente (Aprendizaje Semilla). |

6.3.3.2 Carga de consultas en el aplicativo.

**Tabla 19**. Consultas en el sistema

|  |  |
| --- | --- |
| **Escenarios analizados** | Los escenarios de desempeño que tengan que ver con consultas |
| **Directrices arquitectónicas** | Definir un método de carga de consultas cuando tengan muchos registros. |
| **Estilos, tácticas y raciocinio** | En primera medida definir los índices necesarios para optimizar las consultas.  Si los registros a devolver de una consulta superan los 200 registros se debe realizar la paginación mediante procedimiento almacenado. |
| **Riesgos** | Definir muchos índices que disminuyen el rendimiento en las inserciones. |
| **Ventajas y desventajas** | (+) Tiempos de carga más rápidos.  (-) Toma más tiempo la implementación. |

6.3.3.3 Usabilidad en las pantallas.

**Tabla 20.** Usabilidad en el sistema

|  |  |
| --- | --- |
| **Escenarios analizados** | Los escenarios de usabilidad correspondiente |
| **Directrices arquitectónicas** | Separar la interfaz de usuario del resto de la aplicación. |
| **Estilos, tácticas y raciocinio** | Se realiza la separación de la interfaz (Aplicación web) de todo el negocio, se accede mediante servicios rest.  Se utiliza Angular 2 que es un framework MVC basado en componentes (web components), y que permite el diseño limpio con html 5 y CSS3.  Los web components, que son recomendación de la W3C desde Julio 2014 y permiten extender el HTML a nuestro antojo; y la nueva especificación de JavaScript que desde Junio 2015 cuenta con la especificación ECMAScript 6, de la que se nutre TypeScript para añadirle tipos de datos, interfaces y mejoras en la programación orientada a objetos. |
| **Riesgos** | Demoras en el desarrollo por inexperiencia en Angular 2. |
| **Ventajas y desventajas** | (+) Aprovecha las últimas mejoras en estándares front-end (web components).  (+) Rendimiento superior (teniendo en cuenta otros frameworks como Blaze, React, Angular 1.5)  (+) Fácil integración.  (+) Soporte a multiples navegadores  (-) Aprender a manejar las promesas (Debido a que javascript es asíncrono se debe aprender a controlar la espera de una respuesta cuando se realiza peticiones al servidor) |

6.3.3.4 Envío masivo de notificaciones.

**Tabla 21.** Decisiones envío de notificaciones

|  |  |
| --- | --- |
| **Escenarios analizados** | Los escenarios de rendimiento o escalabilidad correspondiente |
| **Directrices arquitectónicas** | Envío de los correos de sin colapsar el servidor, y sin sobrepasar las 12 horas de atraso |
| **Estilos, tácticas y raciocinio** | Se van a realizar envíos de correos seleccionando grupos de a 10 (mediante consulta de procedimiento almacenado)  Se debe contar con una tabla donde se lleve el control de las notificaciones enviadas.  Este proceso se debe realizar de forma asíncrona para el envío de los 10 correos y cuando se complete si debe enviar el siguiente grupo.  Para esto vamos a utilizar el patrón “patrón asincrónico basado en tareas” TAP (Async y Await) |
| **Riesgos** | . |
| **Ventajas y desventajas** | (+) Es fácil de implementar  (-) Los desarrolladores deben dominar el llamado de métodos con nuevas sentencias (Async y Await) |

6.3.4 Elaboración del diseño detallado. A continuación, se presenta un ejemplo de los diagramas de secuencia diseñados para cada una de las funcionalidades establecidas para el sistema, las cuales permiten ver en mayor detalle la implementación a nivel de Backend y Frontend de acuerdo a la arquitectura establecida.

En el documento de diseño detallado se puede observar de manera más específica los diagramas de secuencia de todas las funcionalidades (Ver Anexo digital 8).

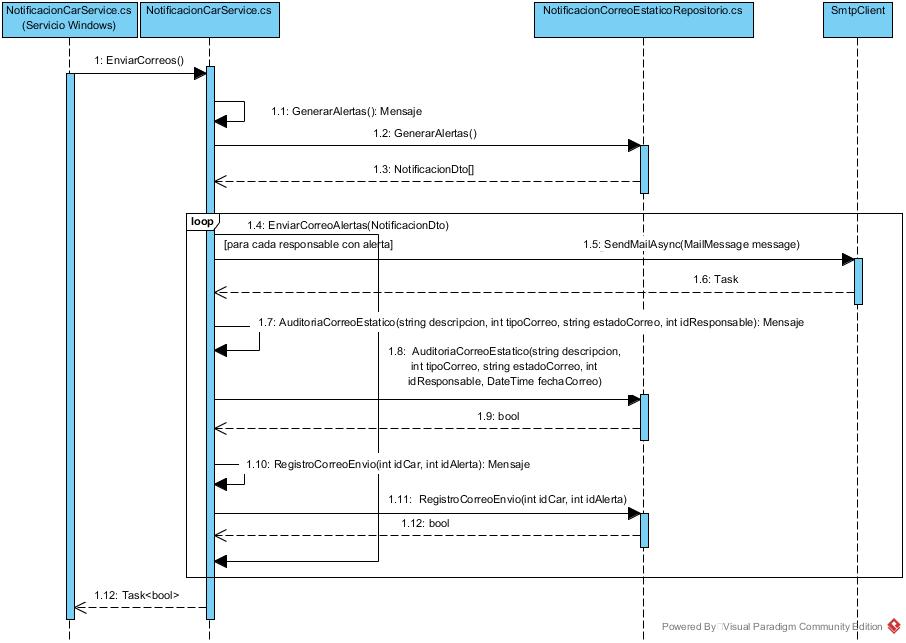
* Diagrama de secuencia (Registro de solicitud CAR - Fragmento)

**Figura 7.** Diagrama de Secuencia Registrar CAR



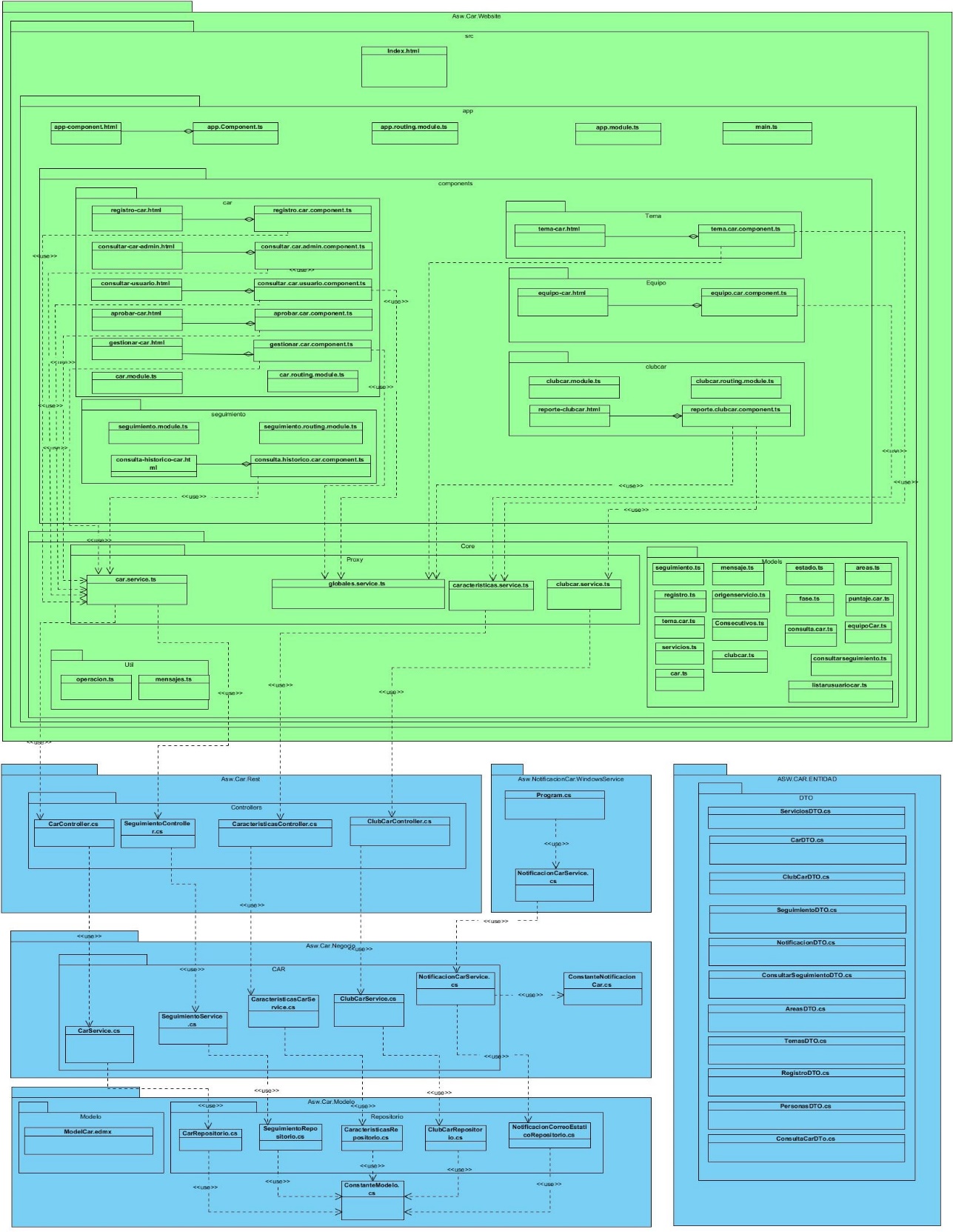
* Diagrama de secuencia (Generación de alertas y envío de notificaciones)

**Figura 8.** Diagrama de secuencia envío de alertas y notificaciones

****

6.3.5 Diagrama de clases.

**Figura 9.** Diagrama de clases



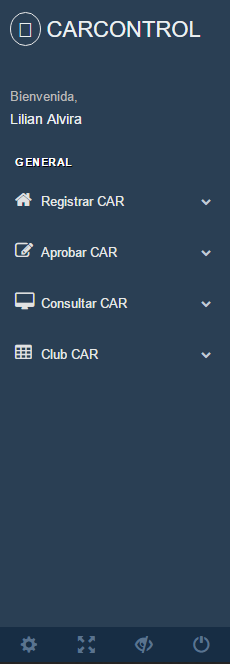
6.3.6 Estándares de presentación y programación.

En esta sección se presentan los estándares de presentación que rigen el sistema CARCONTROL, así como los estándares de programación que se utilizan en la etapa de construcción.

6.3.6.1 Estándares de presentación.

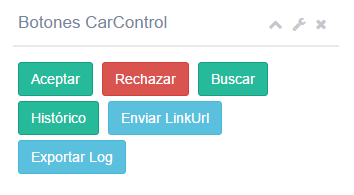
* El menú de navegación principal estará ubicado verticalmente en la parte izquierda de la pantalla, con color de fondo RGB (42, 63, 84) como se muestra en la Figura 10.

**Figura 10**. Estándar de presentación menú de navegación



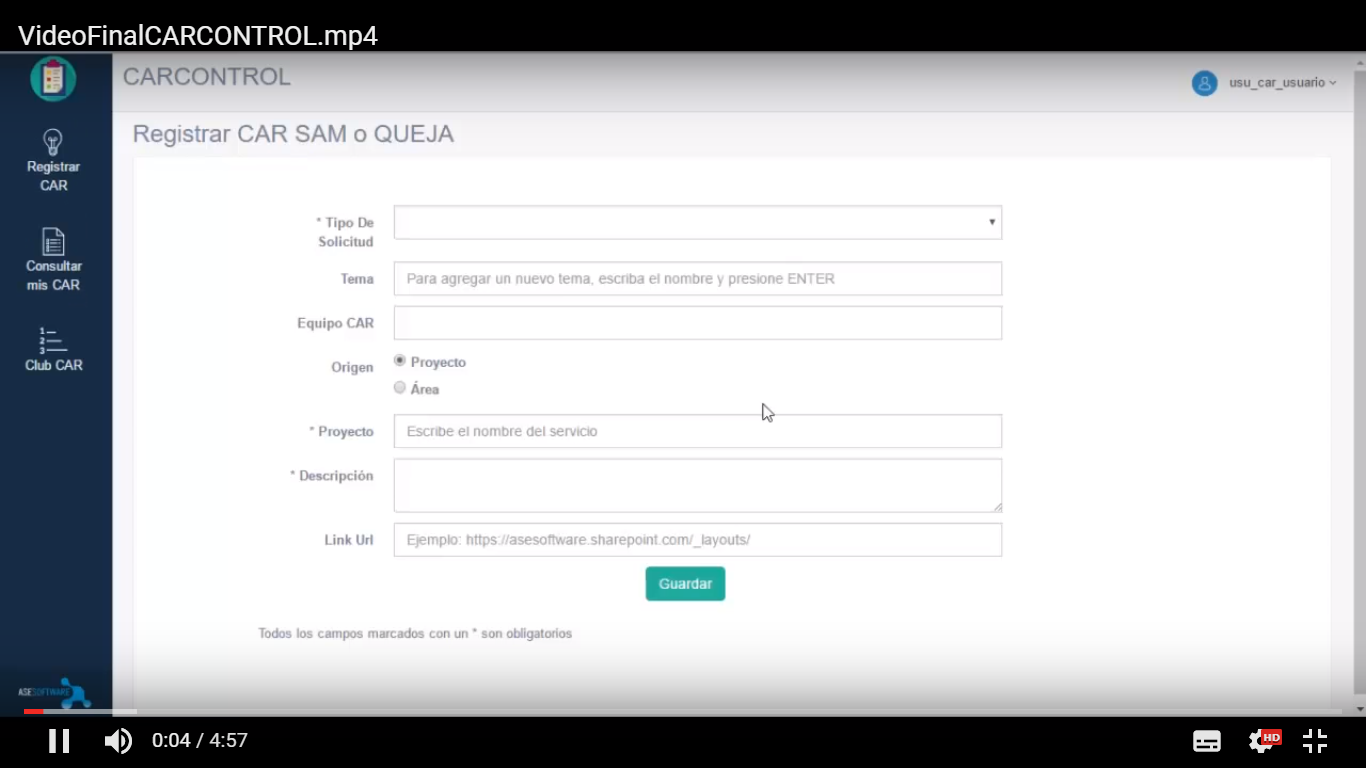
* Los botones que se van a manejar en el sistema CAR Control son: Aceptar, Rechazar, Buscar, Histórico, Enviar LinkUrl, Exportar Log y se van a manejar como se muestran en la Figura 11.

**Figura 11**. Estándar de presentación de botones



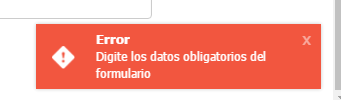
* Los formularios para registrar y actualizar un CAR se van a manejar con el diseño que se muestra en la Figura 12.

**Figura 12**. Estándar de presentación de formularios

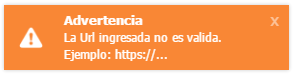


* Los mensajes de error, de confirmación y de advertencia se van a mostrar en la esquina inferior derecha de la pantalla con el diseño que se muestra en las Figuras 13, 14 y 15, respectivamente.

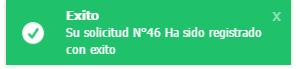
**Figura 13.** Estándar de presentación mensaje de error



**Figura 14.** Estándar de presentación mensaje de advertencia



**Figura 15.** Estándar de presentación mensaje de éxito



6.3.6.2 Estándares de programación.

* Nomenclatura.
  + Pascal: La primera letra del identificador y la primera letra de las siguientes palabras concatenadas están en mayúsculas. El estilo de mayúsculas y minúsculas Pascal se puede utilizar en identificadores de tres o más caracteres. Ejemplo: BuscarPersona.
  + Camel: La primera letra del identificador está en minúscula y la primera letra de las siguientes palabras concatenadas en mayúscula. Ejemplo: personaSeleccionada.
* Estándares en el front.
  + Reglas de uso para los identificadores: Cuando un identificador está compuesto de varias palabras, no se utilizan separadores, como guiones de subrayado ("\_"), guiones ("-") ni puntos (".") entre las palabras. En su lugar, se utiliza la nomenclatura Pascal para señalar el principio de cada palabra.

En la Tabla 22 se resumen las reglas de uso de mayúsculas y minúsculas para los identificadores y se proporcionan ejemplos de los diferentes tipos de identificadores.

**Tabla 22.** Reglas de uso para identificadores en el front

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Nomenclatura** | **Ejemplo** |
| Modelo (Clase) | Pascal | **EquipoCar** |
| Método | Pascal | **PersonaSeleccionada** |
| Variable | Camel | **equipoCar** |
| Propiedad Clase | Pascal | **IdPersona** |

* + Reglas de uso para identificadores de archivos: Cuando el nombre de un archivo está compuesto de varias palabras, se utilizan separadores según el tipo de archivo del que se trata.

En la Tabla 23 se resumen las reglas de uso de mayúsculas y minúsculas para los identificadores y se proporcionan ejemplos de los diferentes tipos de identificadores.

**Tabla 23.** Reglas de uso para identificadores de archivos

|  |  |  |
| --- | --- | --- |
| **Identificador** | **Nomenclatura** | **Ejemplo** |
| Modelo | Camel | **ClubCar** |
| HTML | Minúscula | **gestionar-car.html** |
| Componente | Minúscula | **gestionar.car.component.ts** |
| Servicios | Minúscula | **car.service.ts** |
| Módulo | Minúscula | **car.module.ts** |
| Ruta | Minúscula | **car.routing.module.ts** |

6.3.7 Estrategia de implementación e integración.

En esta sección se describe la estrategia definida para la implementación del sistema utilizando dos incrementos.

6.3.7.1 Secuencia de construcción. El desarrollo del sistema se lleva a cabo en dos incrementos, los cuales permiten establecer bases funcionales del sistema:

**Tabla 24.** Secuencia de construcción incremento 1

|  |  |
| --- | --- |
| **Funcionalidad** | **Componente** |
| F2 | Pantalla de navegación |
| F3 | Pantalla registro CAR |
| F4 | Pantalla de aprobación de CAR |
| F5 | Pantalla gestión CAR |
| F6 | Pantalla consultar CAR Administrador |
| F12 | Pantalla Tema |
| F13 | Pantalla Equipo |

**Tabla 25**. Secuencia de construcción incremento 2

|  |  |
| --- | --- |
| **Funcionalidad** | **Componente** |
| F1 | Pantalla de ingreso |
| F8 | Pantalla consultar CAR usuario |
| F7 | Pantalla consultas histórico de CAR |
| F9 | Pantalla de envío URL |
| F10 | Pantalla reporte Club CAR |
| F11 | Generación de alertas |
| F11 | Envío de correos electrónicos estáticos (aprobación y documentación de un CAR) y Envío de correos electrónicos a través de Servicio Windows (alertas) |

6.3.7.2 Secuencia de integración del producto. Para poderse llevar a cabo el registro de solicitudes de un CAR y la gestión de los mismos debe permitirse tener acceso a los temas de la base local, personas de la compañía, Áreas y Proyectos de ARESII, así como la generación de alertas; por lo tanto, se establece los entregables necesarios para que se pueda llevar a cabo dicho proceso, los cuales son análisis, construcción, implementación, estandarización, cierre y envío de correos.

## REALIZACIÓN DE LA CONSTRUCCIÓN

6.4.1 Configuración del ambiente de desarrollo. En el documento técnico de la aplicación (Ver anexo digital 9) se puede encontrar el procedimiento necesario para realizar la configuración del ambiente de desarrollo, el cual está diseñado para dar soporte a futuros desarrollos o para realizar mantenimientos, allí se pretende describir la composición del sistema a nivel técnico, se encuentra la descripción de los pasos de instalación y un listado de los archivos fuente y una matriz para identificar que fuentes corresponden a cada requerimiento.

6.4.2 Codificar los programas de software. Luego de configurar el ambiente de desarrollo se procede a construir las funcionalidades del sistema las cuales están descritas en el numeral 6.3.7.1 de este documento, la construcción del total de funcionalidades las cuales están ligadas a los requerimientos definidos para el sistema se encuentra distribuida en dos incrementos, a continuación, se describe cada funcionalidad construida.

Incremento 1.Para el primer incremento de construcción, se desarrollaron las funcionalidades de la pantalla de navegación (F2), la pantalla de registrar un CAR (F3), la pantalla de aprobación de CAR (F4), la pantalla de gestión de CAR (F5), la pantalla de consultar CAR por administrador (F6) y las pantallas de tema CAR (F12) y equipo CAR (F13), estas dos últimas se integran dentro de las pantallas de registro y gestión pero debido a la complejidad técnica necesaria para su construcción se decidió implementarlas a través de componentes independientes.

La primera funcionalidad en ser construida fue la pantalla de navegación, la cual está compuesta por un menú horizontal fijado en la parte superior de la pantalla, en el costado derecho del menú horizontal se encuentra el nombre del usuario que esté en el sistema y la opción de salir del sistema. Esta pantalla también está compuesta por un menú vertical fijado en la parte izquierda de la pantalla donde aparecen las funcionalidades asignadas a cada rol del sistema, administrador o usuario. En la parte central de esta pantalla se encuentra el marco de trabajo donde se ubican cada una de las otras pantallas desarrolladas (Ver Figura 16).

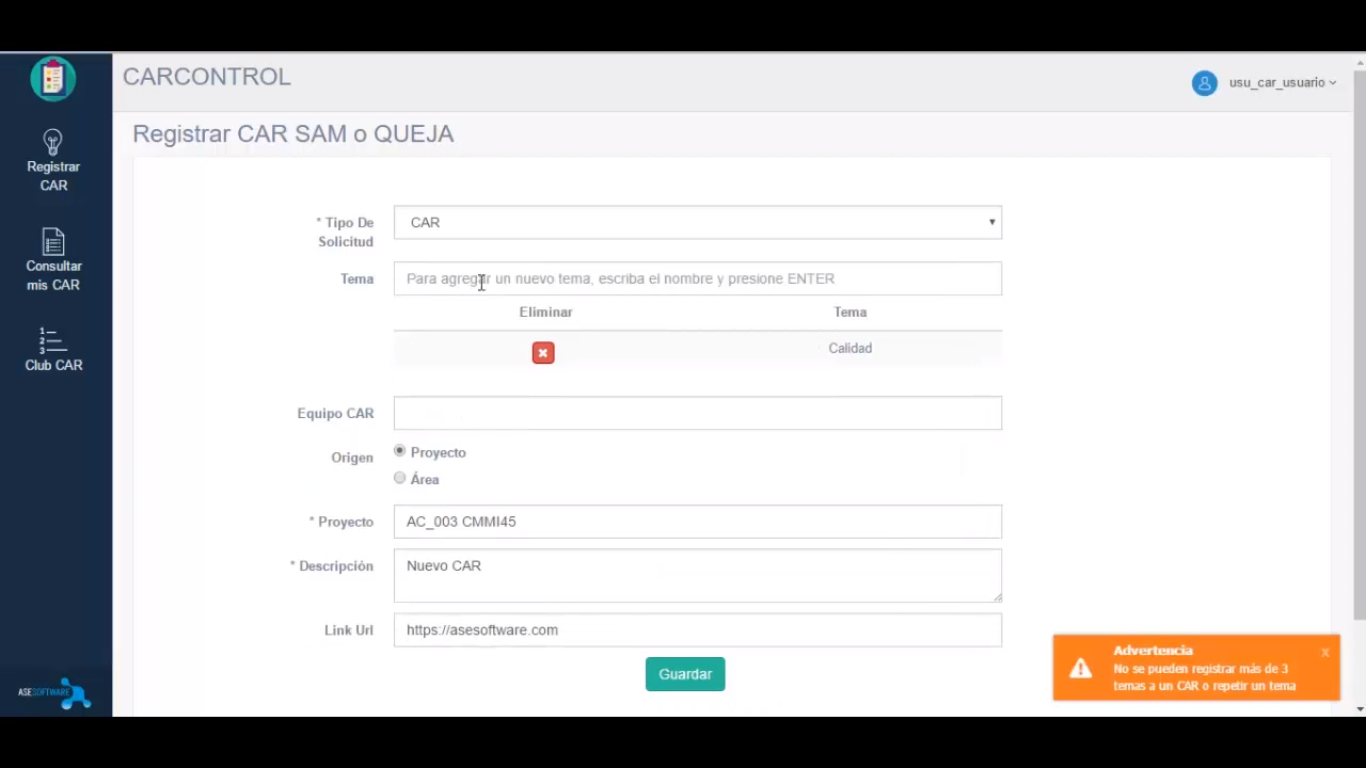
La primera funcionalidad que se ubica dentro de la pantalla de navegación es la de registrar un CAR, la cual está conformada por el formulario de registro, en esta pantalla se implementan las pantallas de tema CAR y Equipo CAR como se explica a continuación. El formulario de registro está conformado por los campos de *Tipo de solicitud* el cual es un componente desplegable donde se puede seleccionar si la solicitud está relacionada a un CAR un SAM o una Queja.

A continuación, se encuentran implementadas las pantallas de Tema y Equipo CAR cuyo manejo funciona de la misma manera, son campos de texto donde el usuario debe ingresar el nombre de un tema o una persona que va a estar asociada a la solicitud que se desea registrar, a medida que se escriba el nombre, el campo va a ir mostrando sugerencias al usuario para diligenciar de manera más rápida estos campos. Para el caso del campo tema, el sistema permite agregar un máximo de tres (3) a la solicitud y para el caso del equipo, el sistema permite agregar un máximo de 5 personas a la solicitud a registrar, de la misma manera antes de enviar el formulario el sistema le permite al usuario corregir alguno de los campos si se registra un ítem de manera incorrecta.

Este formulario cuenta con la opción de elegir si la solicitud está ligada a un proyecto de la compañía o un área, el usuario puede seleccionar solo alguna de las dos opciones ya que debido a las reglas del negocio una solicitud nunca puede estar asociada a un proyecto y un área; dependiendo la elección del usuario a medida que se digite el nombre aparece una lista con sugerencias de coincidencia para facilitar la diligencia de este campo. En la parte final del formulario aparece un campo de texto destinado a agregar una descripción de la solicitud la cual se pide de manera opcional y un campo de texto para ingresar el link del repositorio de la compañía en el caso que la solicitud cuente en ese momento con una documentación asociada, el campo tiene las validaciones necesarias para guiar al usuario a identificar que el link solo puede ser del repositorio de la empresa (SharePoint) y el formato con el cual debe ingresarlo.

En la parte inferior se encuentra el botón de Guardar la solicitud y enviarla al administrador del sistema, al momento de presionar el botón el sistema valida la información del formulario, en caso de tener datos ingresados incorrectamente se muestra un mensaje de advertencia en la parte inferior derecha de la pantalla informando el error o en caso de que la información sea correcta se muestra un mensaje de éxito informando al usuario que la solicitud fue registrada correctamente adjuntando el número de la misma, las pantallas de navegación y de registro de solicitud de CAR como se describen anteriormente se pueden observar en la Figura 16.

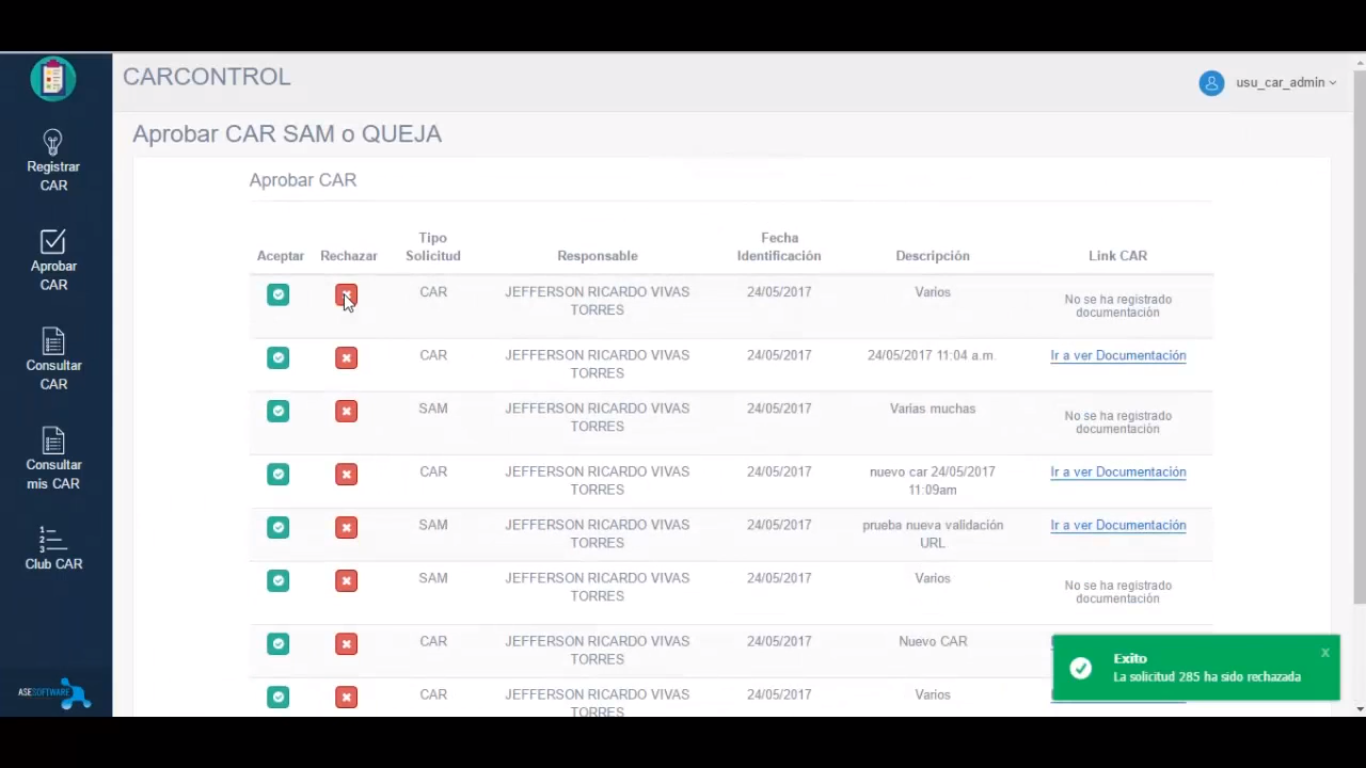
**Figura 16.** Pantalla registrar solicitud CAR



La siguiente funcionalidad en ser implementada es la aprobación de un CAR (F4), en esta pantalla se muestra el listado completo de las solicitudes que están pendientes por aprobar o rechazar, la información que se muestra para cada una de las solicitudes es el tipo de solicitud de acuerdo a lo ingresado en el registro, el responsable, la fecha de identificación que se toma directamente del sistema, la descripción de la misma y el link de documentación asociada en caso de que esta haya sido ingresada en el registro como se puede observar en la Figura 17.

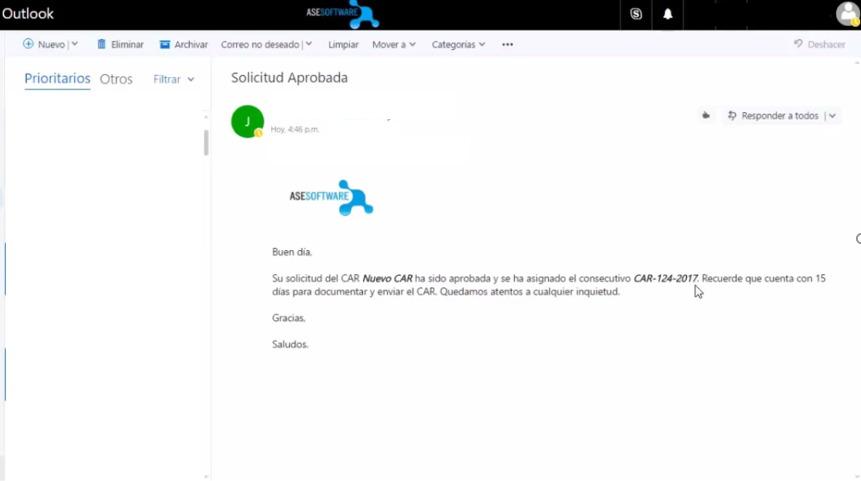
Adicionalmente aparecen los botones de rechazar o aceptar la solicitud, decisión que es tomada por el administrador del sistema de acuerdo a la veracidad y completitud de la información que se ha registrado.

**Figura 17.** Pantalla de aprobar CAR



En el momento de aceptar una solicitud el sistema le asigna un número consecutivo el cual será el identificador único con el cual se va a tratar durante todo el ciclo de vida de un CAR y luego dirige al administrador a la pantalla de gestión, al momento de rechazar una solicitud, se muestra un mensaje en la parte inferior derecha indicando que la solicitud fue rechazada y se envía un mensaje al responsable indicando la decisión tomada como se puede ver en la Figura 18.

**Figura 18**. Correo aprobación CAR



De acuerdo a la secuencia de construcción del primer incremento, la siguiente funcionalidad en ser implementada es la gestión de las solicitudes que han sido aprobadas por el administrador.

A la pantalla de gestión se accede automáticamente al momento de aprobar una solicitud o desde la pantalla de consulta que será explicada más adelante en este documento. La pantalla de gestión cuenta con la siguiente información: el número consecutivo que fue asignado en el momento de la aprobación, este número consecutivo está formado por tres secciones, la primera indica el tipo de solicitud creada ya sea CAR, SAM o Queja, la segunda parte es un número incremental y la tercera parte el año en el que se crea la solicitud. Adicionalmente se muestra información como la fecha de identificación, la fecha real de cierre la cual se genera automáticamente cuando el CAR llega al estado cerrado y a la fase completado; en la parte superior de la pantalla se muestra también el proyecto o el área asociados al CAR.

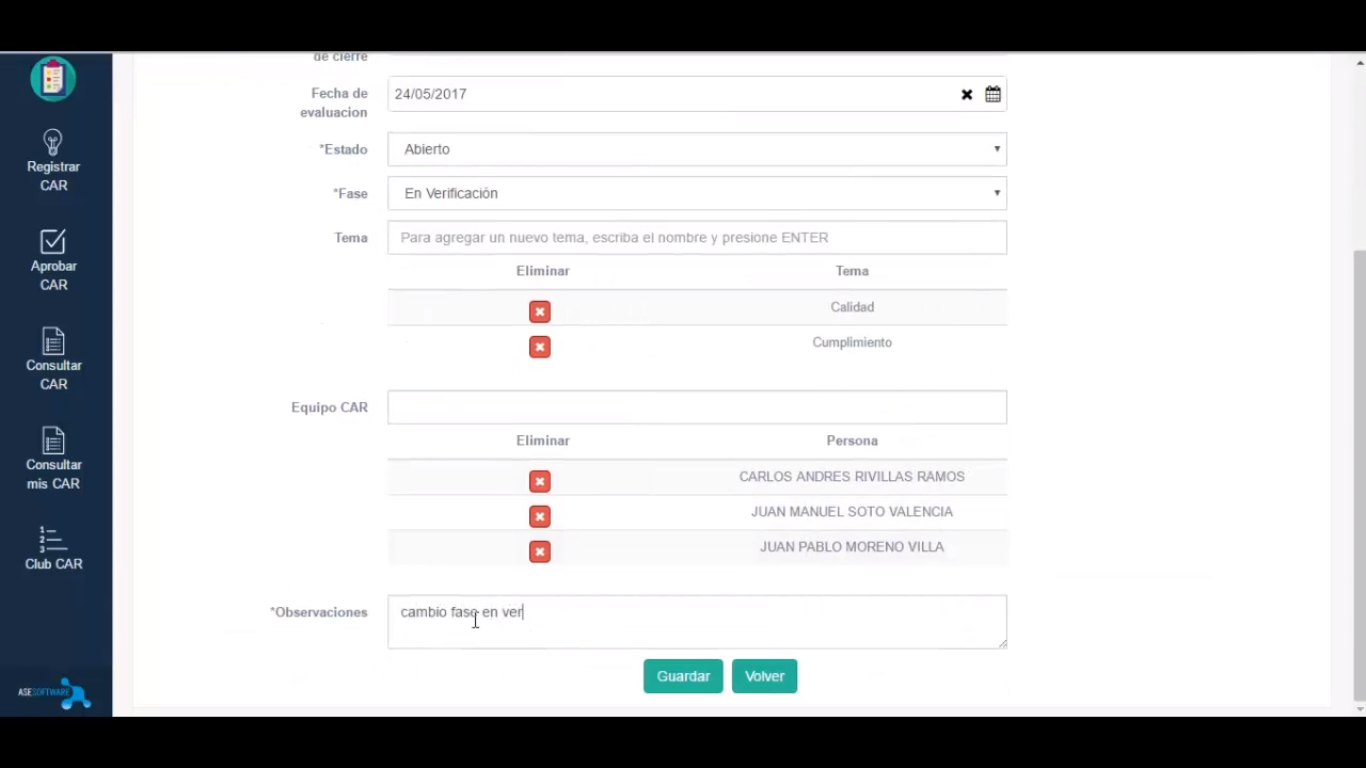
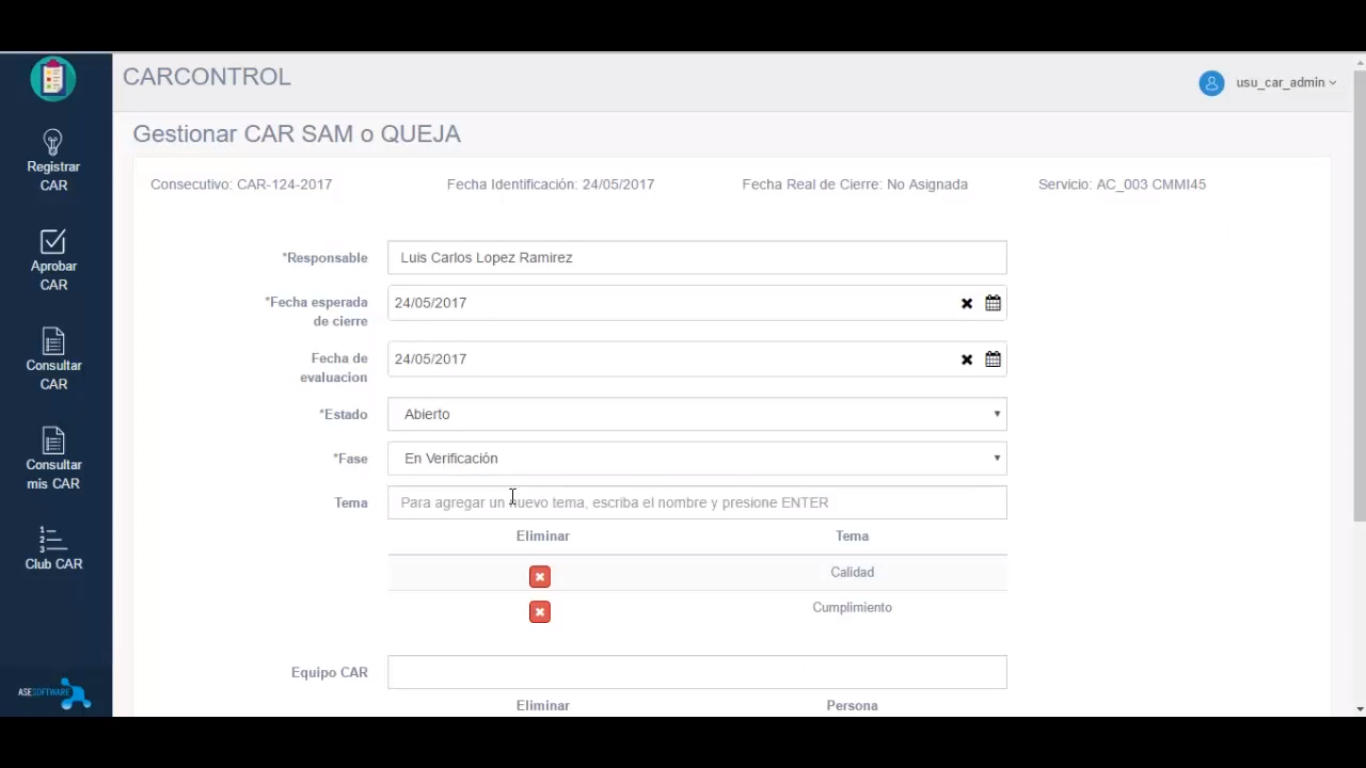
En la parte central de la pantalla de gestión aparece un formulario con los datos que pueden ser gestionados por el administrador del sistema, en el primer caso está el nombre del responsable, de acuerdo a la política de la compañía, un CAR no tiene que tener un único responsable a lo largo de su ciclo de vida, por lo contrario pueden existir situaciones en las que se tenga que cambiar, también aparece la fecha esperada de cierre la cual se asigna bajo criterio del administrador de la misma manera que asigna la fecha de evaluación, esta última se puede asignar únicamente cuando el CAR está en estado cerrado y fase completado.

Los siguientes campos que se pueden gestionar por parte del administrador son los estados y las fases, se pueden asignar 3 estados posibles: abierto, cerrado y cancelado y 4 fases que dependiendo las reglas del negocio se pueden asociar o no a cada uno de los estados, las fases para elegir son: sin documentar, plan de acción, en verificación y completado.

De la misma forma que sucede en la pantalla de registro, en la pantalla de gestión están implementadas las pantallas de tema y equipo CAR, en esta funcionalidad se pueden editar los temas y las personas asociadas a la solicitud.

En la parte inferior del formulario se encuentra un campo destinado para realizar alguna observación sobre los cambios ejecutados y los botones para aceptar o volver como se observa en la Figura 19.

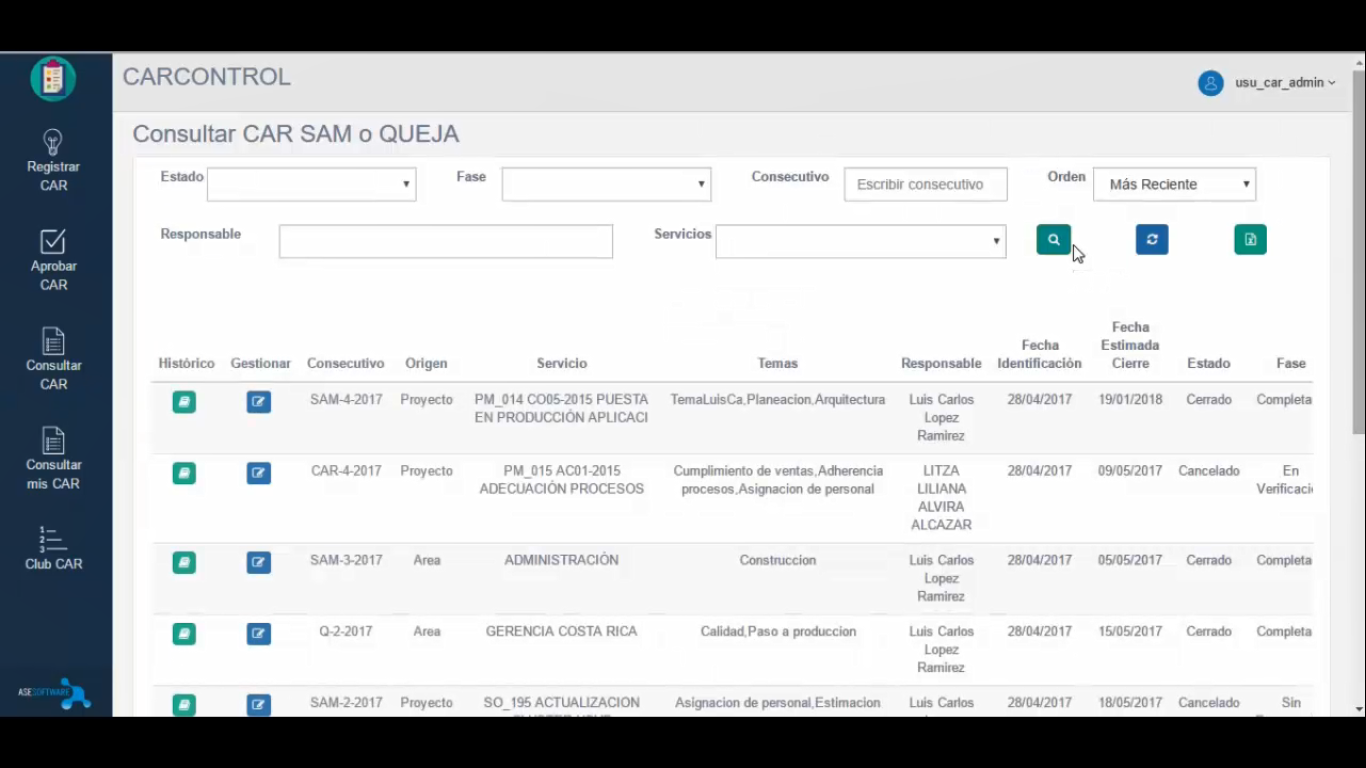
**Figura 19.** Pantalla gestionar CAR



La ultima funcionalidad desarrollada durante el primer incremento es la consulta de CAR por parte del administrador, el usuario también tiene está funcionalidad, pero implementada de manera más sencilla. Debido a la magnitud de información que se puede manejar en esta pantalla por parte del administrador, se cuenta con una serie de filtros de consulta para realizar la búsqueda de un CAR de manera más eficiente.

En la pantalla de consultar CAR por parte del administrador se tienen los filtros de fase y estado los cuales son listas desplegables donde aparecen las fases y estados establecidos para el proceso de CAR, en la parte superior de la pantalla se encuentra también un campo para buscar la información por número de consecutivo, el cual al ir ingresando caracteres va mostrando una lista con posibles resultados de coincidencia; el administrador tiene la posibilidad de filtrar los resultados de esta pantalla por el nombre del responsable, este campo de texto muestra resultados de coincidencia a medida que se van digitando caracteres, este funcionamiento aplica también para otro criterio de búsqueda en la pantalla que es buscar por servicio o área, en estos campos para que las consultas sean más optimas, solo aparece la información que ya ha sido asociada en los registros de los CAR; en la parte izquierda de la pantalla aparecen los botones que llevan a ver el historial de cada registro o ingresar a gestionar cada uno de ellos como se puede ver en la Figura 20.

**Figura 20.** Pantalla consultar CAR administrador



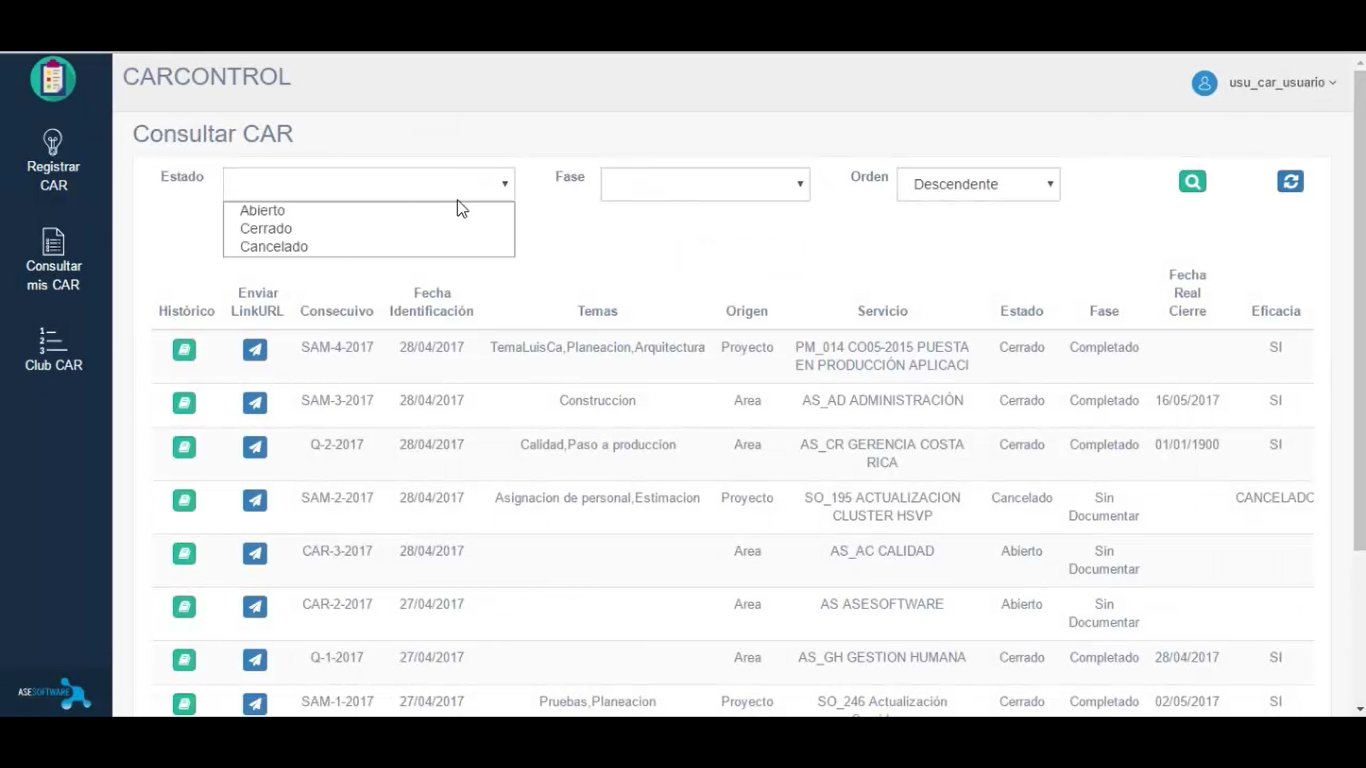
Esta pantalla tiene la funcionalidad de exportar los registros a formato Excel para realizar procesos de toma de decisiones por parte del área de planeación.

**Incremento 2.** En la segunda iteración de la construcción del proyecto se realizaron las siguientes funcionalidades: Ingreso al sistema (F1), consultar histórico de CAR (F7), consultar Car por parte del usuario (F8), envío del link de documentación (F9), la consulta de reporte del club CAR (F10), generación de alertas, envío de correos electrónicos cuando se aprueba, se rechaza o se registra documentación de un CAR y el envío de correos electrónicos para la notificación de alertas.

De las funcionalidades implementadas durante el segundo incremento, la primera en ser desarrollada es la consulta de los CAR por parte del usuario, esta funcionalidad es similar a la consulta que realiza el administrador sobre todos los CAR, pero tiene menos criterios de búsqueda debido a que la magnitud de información que maneja el usuario es mucho menor.

Los criterios de búsqueda disponibles para el usuario son: el estado, la fase y tiene la posibilidad de ordenar la información en orden ascendente o descendente de acuerdo con la fecha de identificación, en la parte izquierda de la pantalla se encuentran los botones que dirigen a las funcionalidades de consultar el histórico de cada registro y el envío del link de la documentación para ser evaluada por el administrador del sistema como se observa en la Figura 21.

**Figura 21.** Pantalla consultar CAR usuario

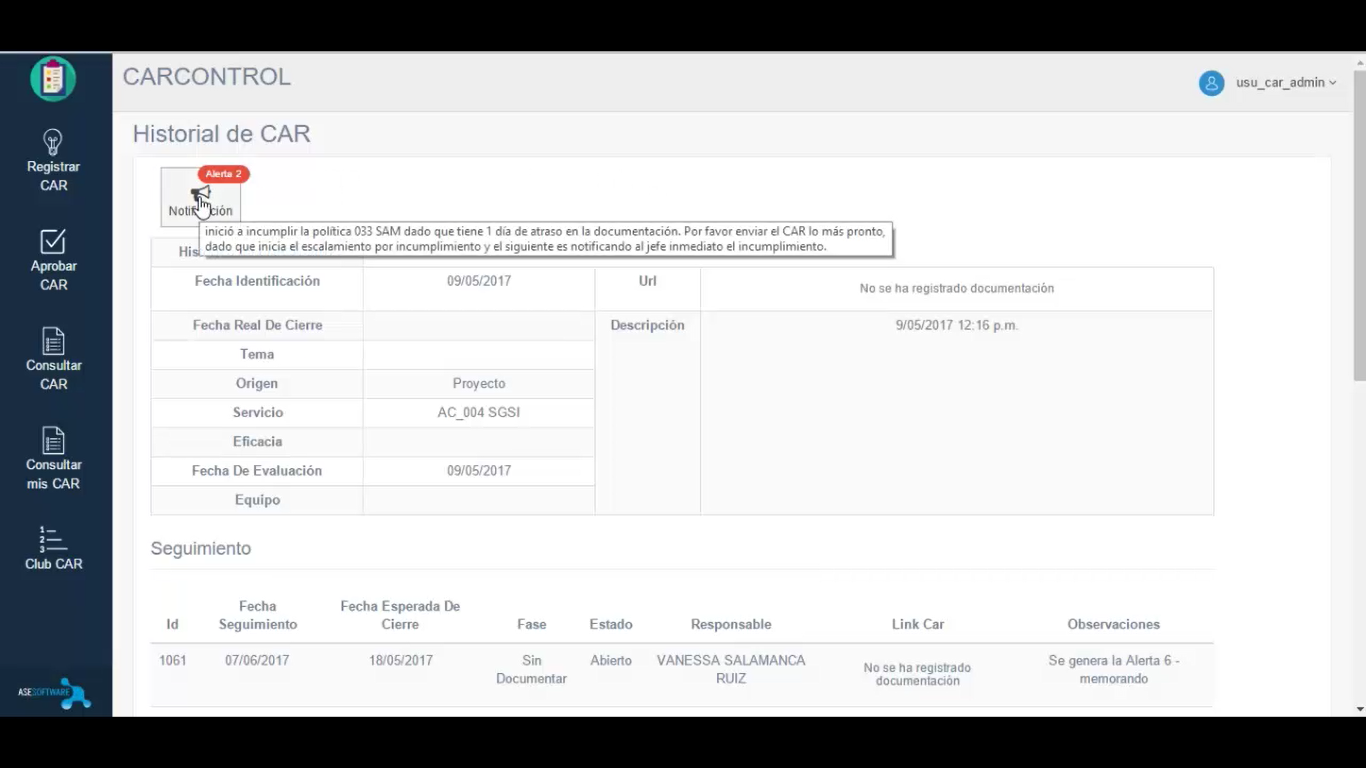


La funcionalidad consultar CAR por parte del usuario fue la base para el desarrollo de las funcionalidades de ver histórico y enviar el link de documentación. La pantalla de ver histórico se accede desde la consulta de CAR ya sea por parte del administrador o por parte del usuario.

En la parte superior de la pantalla aparece la notificación de alerta en la cual está el CAR, estas notificaciones aparecen de acuerdo al tiempo de demora que el responsable de CAR lleva con la documentación del mismo, alerta 1 hace referencia a que se tienen 8 días de plazo para realizar la documentación, la alerta 2 significa que lleva 8 días de retraso con la documentación, el detalle de estás alertas se puede observar a detalle en la política SAM-033 (Ver anexo digital 4).

En la parte superior de la pantalla se plasman todos los datos actuales con los que cuenta el CAR como lo son: el número consecutivo, la fecha de identificación, la fecha real de cierre la cual no aparece obligatoriamente en todos los registros debido a que esta solo aparece en los CAR que tienen estado cerrado y fase completado, los temas asociados, el equipo de trabajo que conforma el CAR, el link de la documentación más reciente y la descripción del CAR como se aprecia en la Figura 22.

**Figura 22.** Pantalla historial básica de CAR



En la parte inferior de la pantalla de consultar el histórico de CAR aparecen todos los cambios que se han efectuado al CAR en cuestión, en estos cambios se incluyen cambios de fases, cambios de estados, envío de documentación y cualquier cambio que se realice en la pantalla de gestionar.

En la tabla que evidencia los cambios que se han realizado sobre el CAR a lo largo del ciclo de vida se muestran los datos que son relevantes para el administrador del sistema y que sirven de ayuda para sacar algunos índices por parte del área de planeación, estos campos son el identificador único del registro del seguimiento para tener una idea de la cantidad de veces que se gestionó la solicitud, la fecha en la cual se realizó el cambio en el CAR, la fecha esperada de cierre ayuda a saber el avance que se ha tenido contra la cantidad de veces que se ha gestionado, de igual manera con los campos de fase y estado.

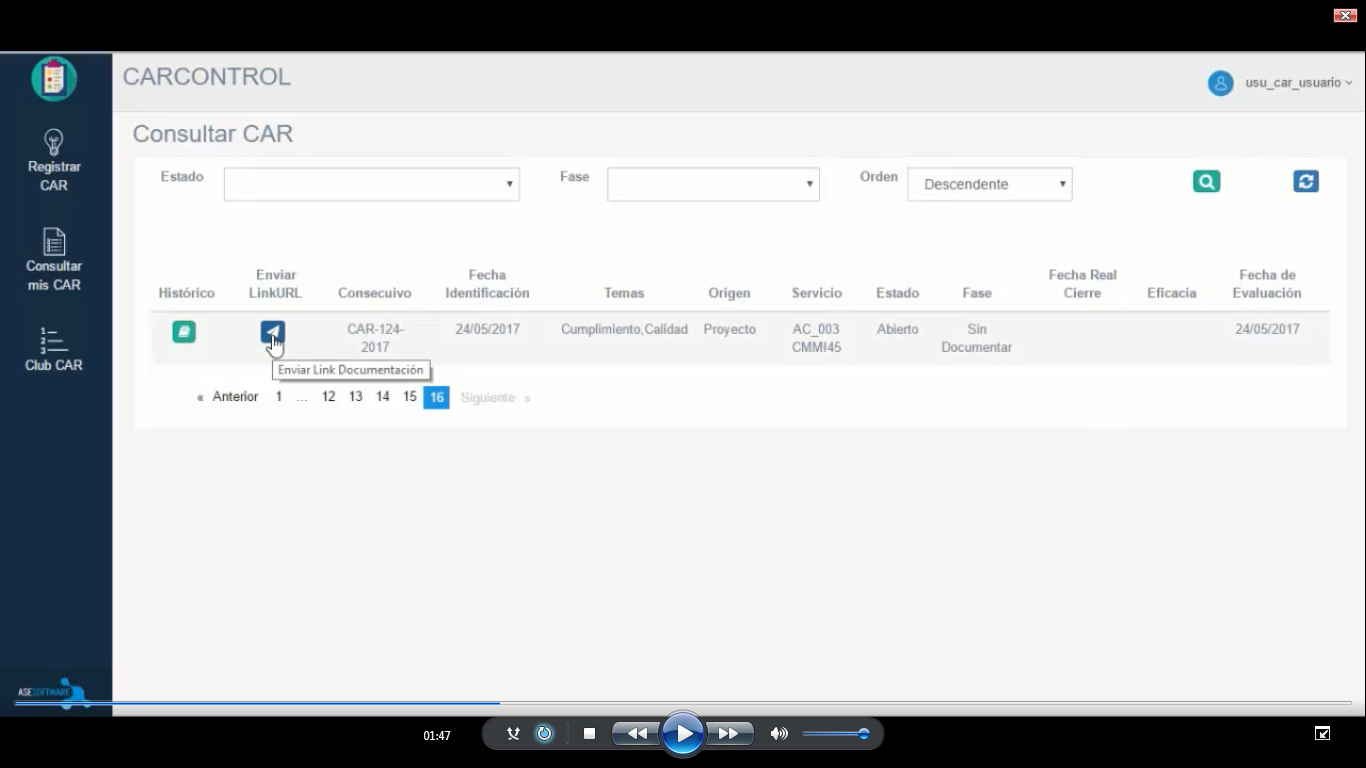
En la tabla de los registros históricos del CAR también aparece el nombre de los responsables que ha tenido a través del tiempo, debido a que un CAR no tiene que ser desarrollado en su totalidad por la misma persona, también aparece el campo del link de la documentación el cual es de vital importancia para el área de planeación ya que de acuerdo a la documentación es que se decide si el CAR avanza de estado o no. Para cada cambio que se realice sobre el CAR siempre aparece una descripción en el último registro indicando específicamente el cambio realizado, algunos de estos cambios son descritos por el administrador del sistema como es el caso de cambios de fases y estados, mientras que la generación de alertas por retrasos en el tiempo de la documentación es realizada de forma automática por el sistema, así como se observa en la Figura 23.

**Figura 23.** Pantalla histórica avanzada de CAR



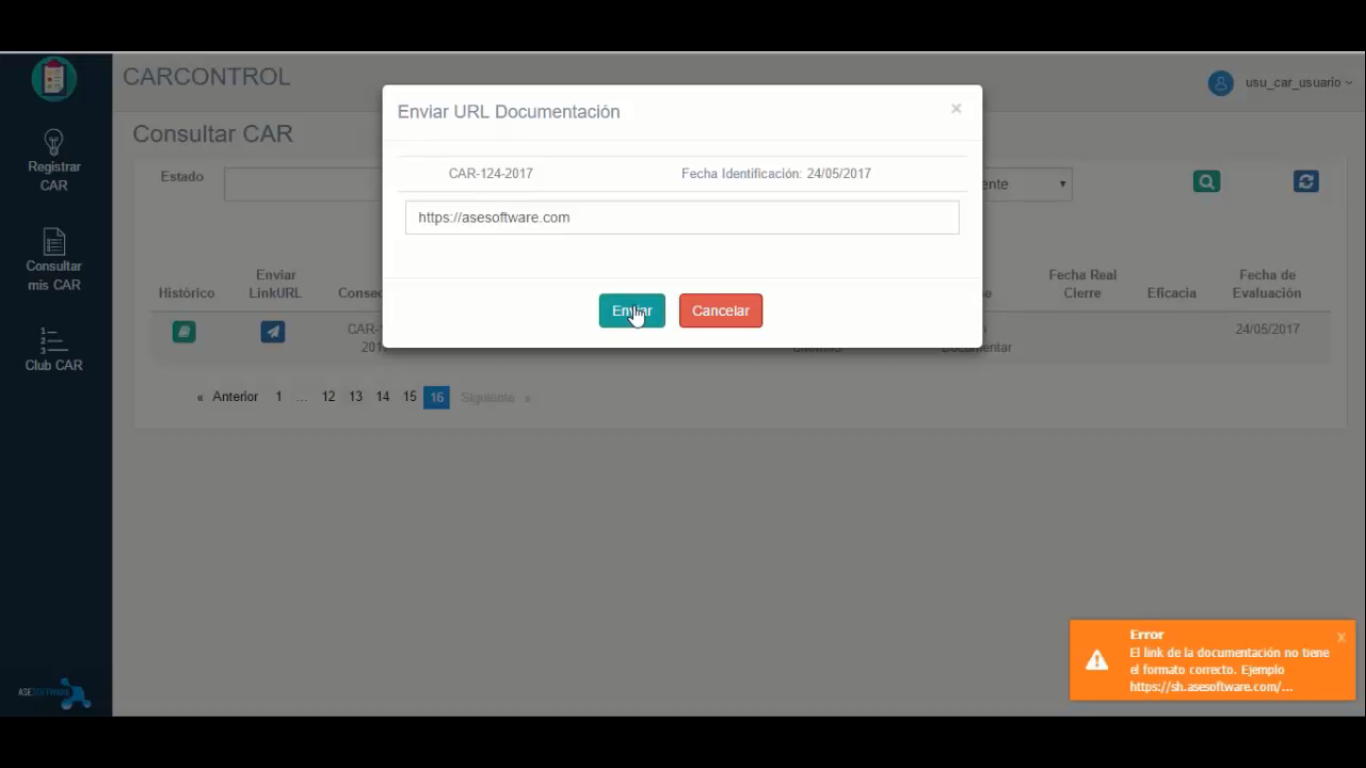
La siguiente funcionalidad en ser implementada es él envió del enlacede la documentación de un CAR (F9), la forma de acceder a esta funcionalidad es ingresando a la pantalla de consultar el CAR por usuario (F8), en la lista de los CAR en la parte izquierda aparece el icono de envío de documentación como se muestra en la Figura 24.

**Figura 24.** Acceso a la funcionalidad de enviar link de documentación



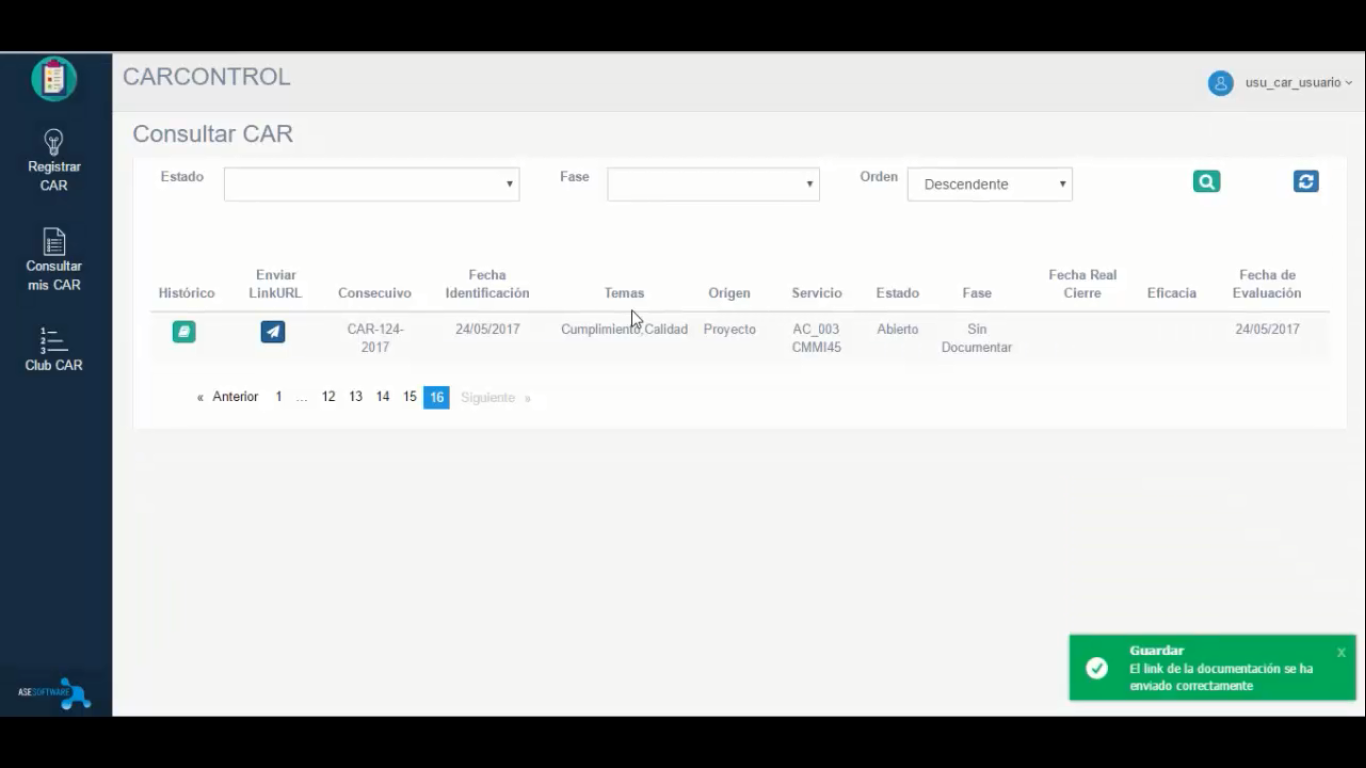
Una vez se ingresa a esta funcionalidad, aparece una ventana emergente con el número consecutivo del CAR y la fecha de identificación y en la parte inferior aparece un campo de texto donde el usuario debe ingresar el link de la documentación asociada, el sistema permite únicamente que este link pertenezca al repositorio de documentación de la compañía, de no ser así este le indica al usuario a través de un mensaje de advertencia que aparece en la parte inferior derecha de la pantalla la forma correcta de ingresar el link de la documentación así como se ve en la Figura 25

**Figura 25.** Pantalla advertencia envío link de documentación



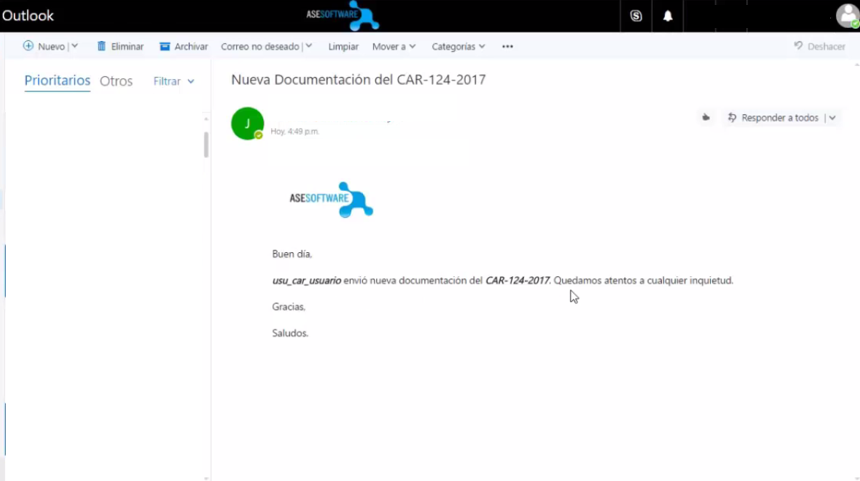
Cuando el link de la documentación se digita correctamente el sistema le indica al usuario que esta se ha enviado correctamente como se ve en la Figura 26.

**Figura 26.** Pantalla éxito envío link de documentación



Una vez se envía el link de documentación, el administrador del sistema recibe un correo de notificación especificando el usuario y el CAR del cual se ha enviado la documentación, así como se observa en la Figura 27.

**Figura 27.** Notificación envió de documentación

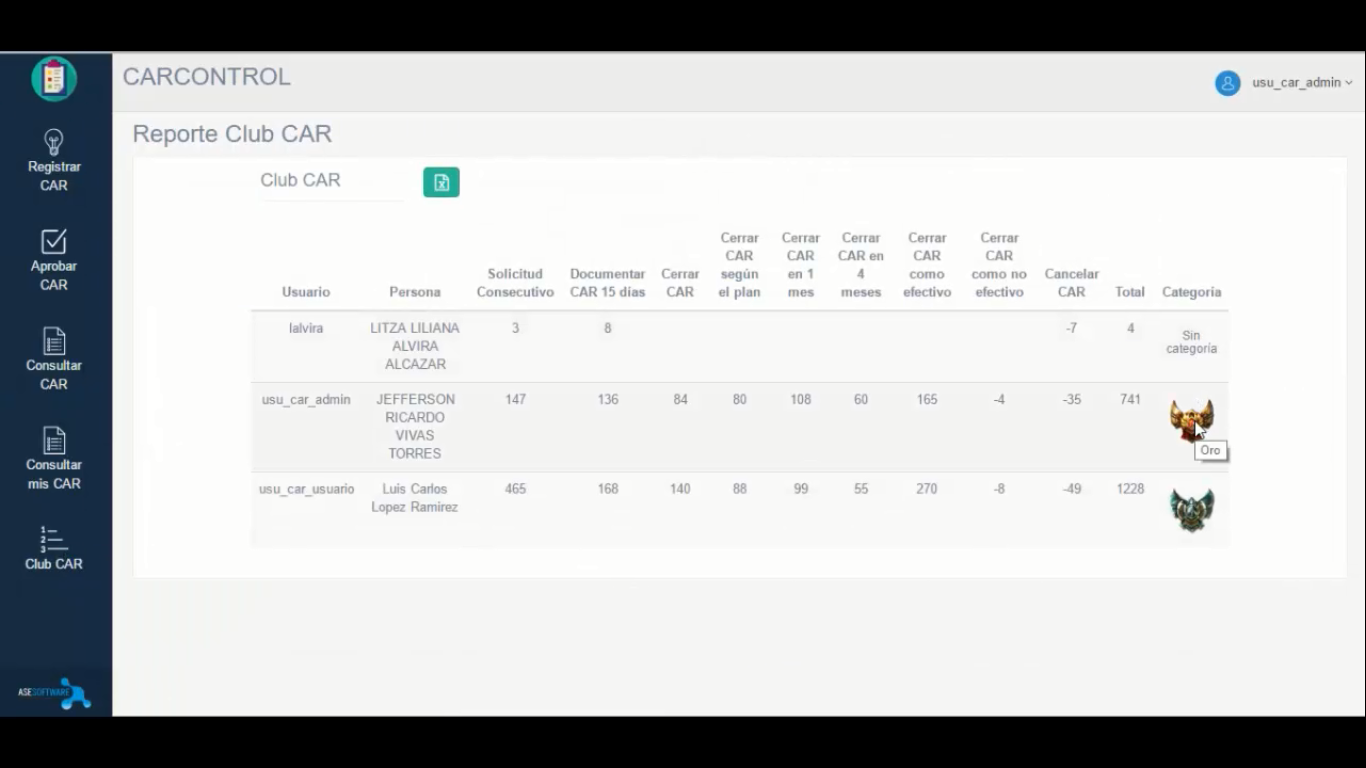


La funcionalidad del club CAR puede ser accedida por los dos tipos de usuarios del sistema, el administrador puede visualizar los puntajes parciales y totales de cada persona que sea responsable de un CAR en ese momento, en la pantalla del club CAR aparece el nombre de usuario utilizado por la compañía para identificar a las personas, el cual es generalmente la primera letra del nombre y el apellido completo, luego en la tabla aparecen los puntajes parciales por los cuales ha obtenido alguna puntuación el CAR en cuestión.

Los puntajes parciales corresponden a casos como los siguientes: cuando se acepta la solicitud del CAR, el sistema la asignatres puntos, cuando se realiza la documentación del mismo en un tiempo menor a 15 días desde la fecha de identificación el sistema la asigna 8 puntos, cuando el CAR se cierra según el plan, es decir cuando este pasa por cada una de las fases y etapas hasta ser completado el sistema también le asigna un puntaje, de la misma manera se asigna un puntaje menor cuando se cierra el CAR sin seguir el plan adecuado, de la misma manera el sistema va asignando puntaje cuando el CAR se cierra en tiempo definidos en la guía del club CAR.

Cuando un CAR se cancela el sistema le resta cierto puntaje al total acumulado hasta ese momento, al final de cada mes calendario de la compañía, se publican estos resultados al público dándole a cada persona responsable de un CAR una categoría según el puntaje total hasta el momento, estas categorías pueden ser: oro, plata, bronce, esmeralda y diamante, tal como se observa en la Figura 28.

**Figura 28**. Pantalla club CAR

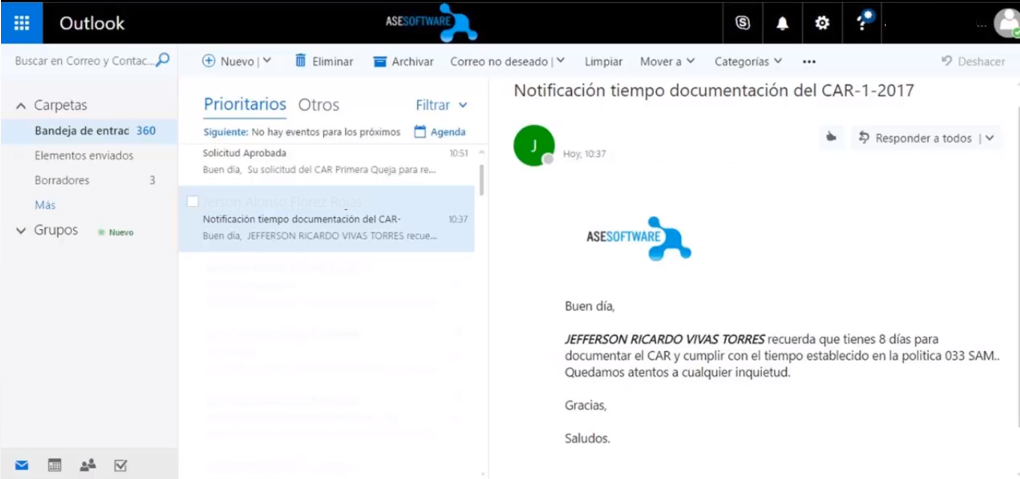


En la pantalla del club CAR existe la funcionalidad de descargar el log de puntos a formato Excel para la generación de indicadores por parte del área de planeación.

Luego de tener construidas las funcionalidades del sistema se procede a implementar el servicio Windows para el envío de notificaciones, estás notificaciones se envían al correo del administrador del sistema indicando cuando a un CAR se le genera una de las 6 alertas posibles establecidas en política SAM-033 de la compañía.

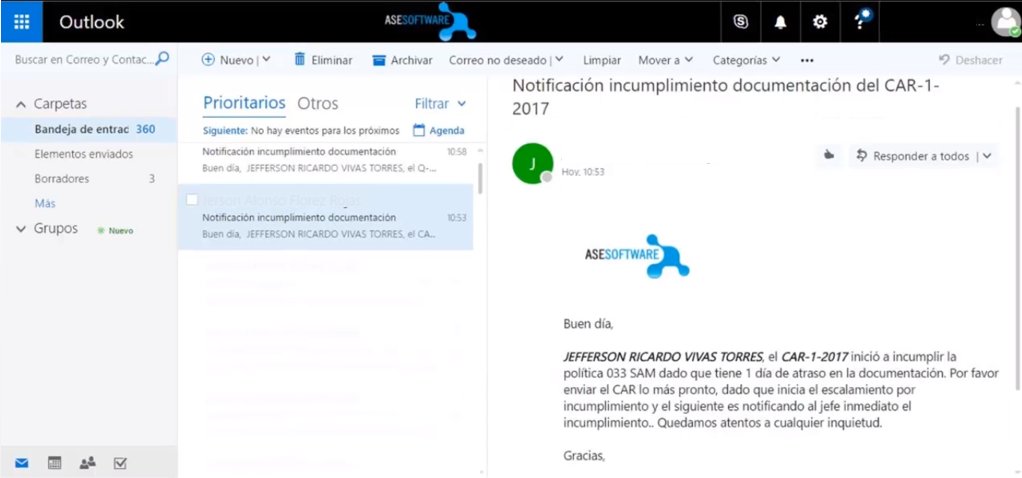
La primera notificación de alerta indica que el CAR tiene 8 días para ser documentado como se observa en la Figura 29.

**Figura 29.** Notificación alerta 1



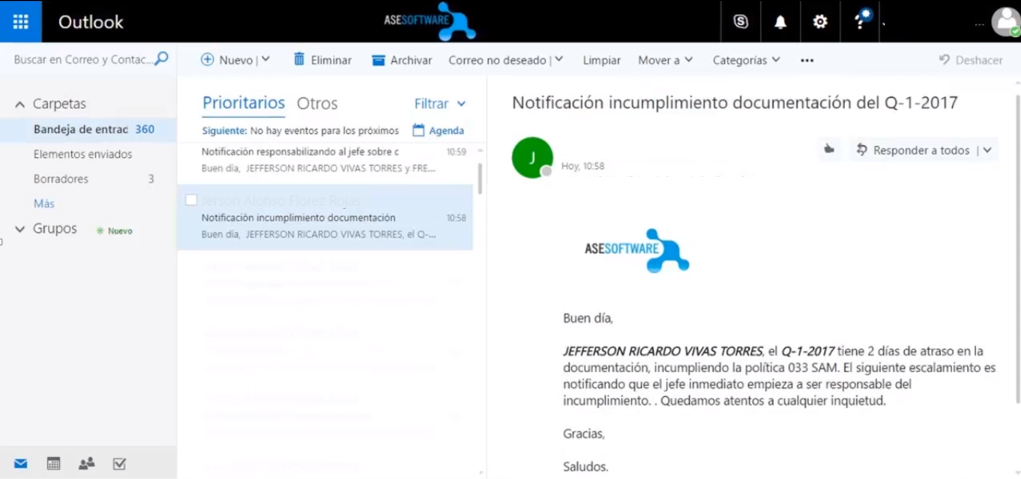
La segunda notificación de alerta indica que el CAR inició a incumplir la política SAM-033 debido a que tiene un día de atraso en el tiempo para realizar la documentación como se observa en la Figura 30.

**Figura 30.** Notificación alerta 2



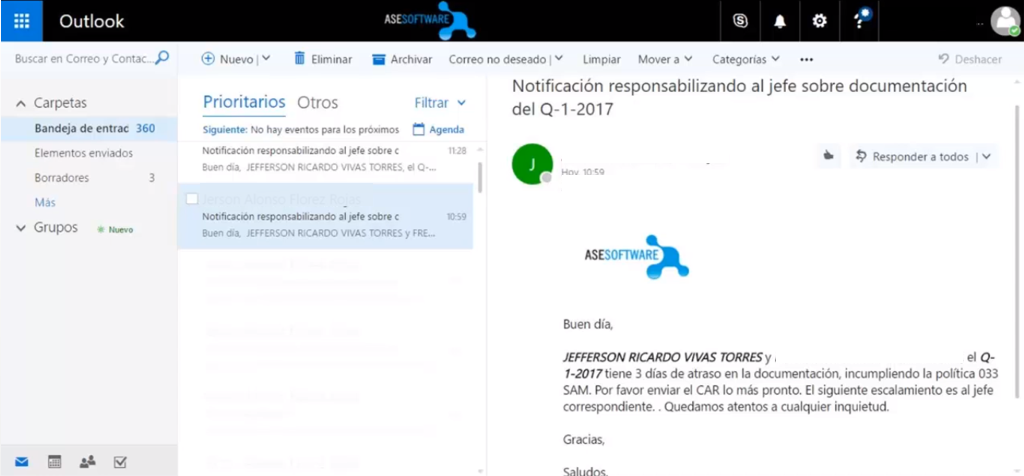
La tercera notificación de alerta indica que el CAR sigue incumpliendo la política SAM-033 debido a que tiene dos días de atraso en el tiempo para realizar la documentación como se observa en la Figura 31.

**Figura 31.** Notificación alerta 3



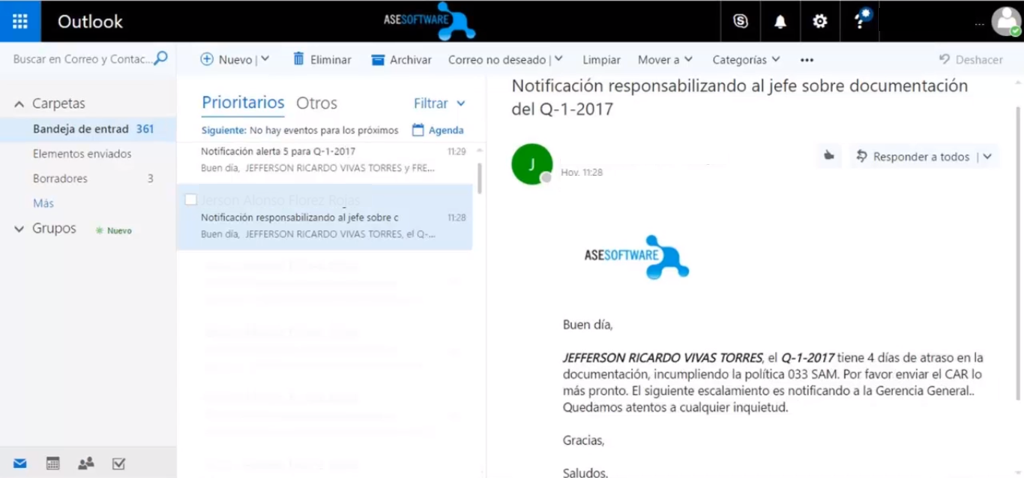
La cuarta notificación de alerta indica que el CAR tiene tres días de atraso en la entrega de la documentación correspondiente, aquí se notifica a la persona responsable del CAR y al jefe inmediato de la persona como se observa en la Figura 32.

**Figura 32**. Notificación de alerta 4



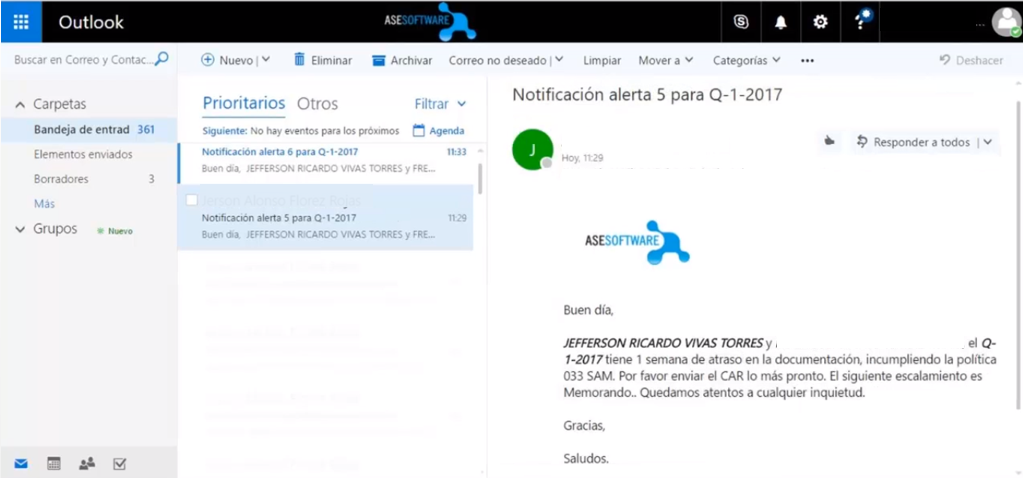
La cuarta notificación de alerta con un escalamiento más le indica al responsable del CAR que este tiene un atraso de cuatro días en la entrega de la documentación correspondiente, esta notificación se escala al siguiente jefe correspondiente en la jerarquía organizacional de la compañía como se observa en la Figura 33.

**Figura 33.** Notificación de alerta 4.1



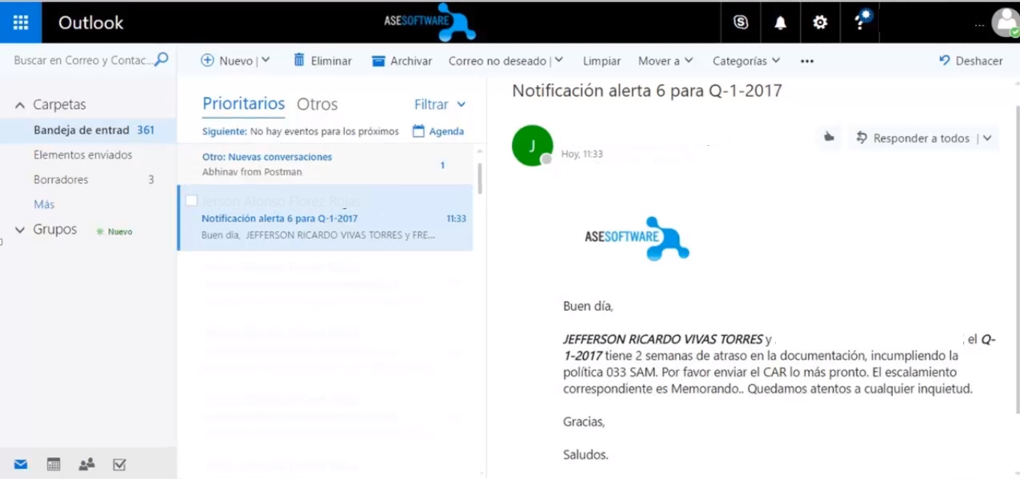
La quinta notificación de alerta indica que el CAR tiene una semana de atraso en la entrega de la documentación correspondiente, el correo que notifica se dirige al responsable del CAR con copia al jefe inmediato y al gerente general de la compañía como se observa en la Figura 34.

**Figura 34.** Notificación de alerta 5



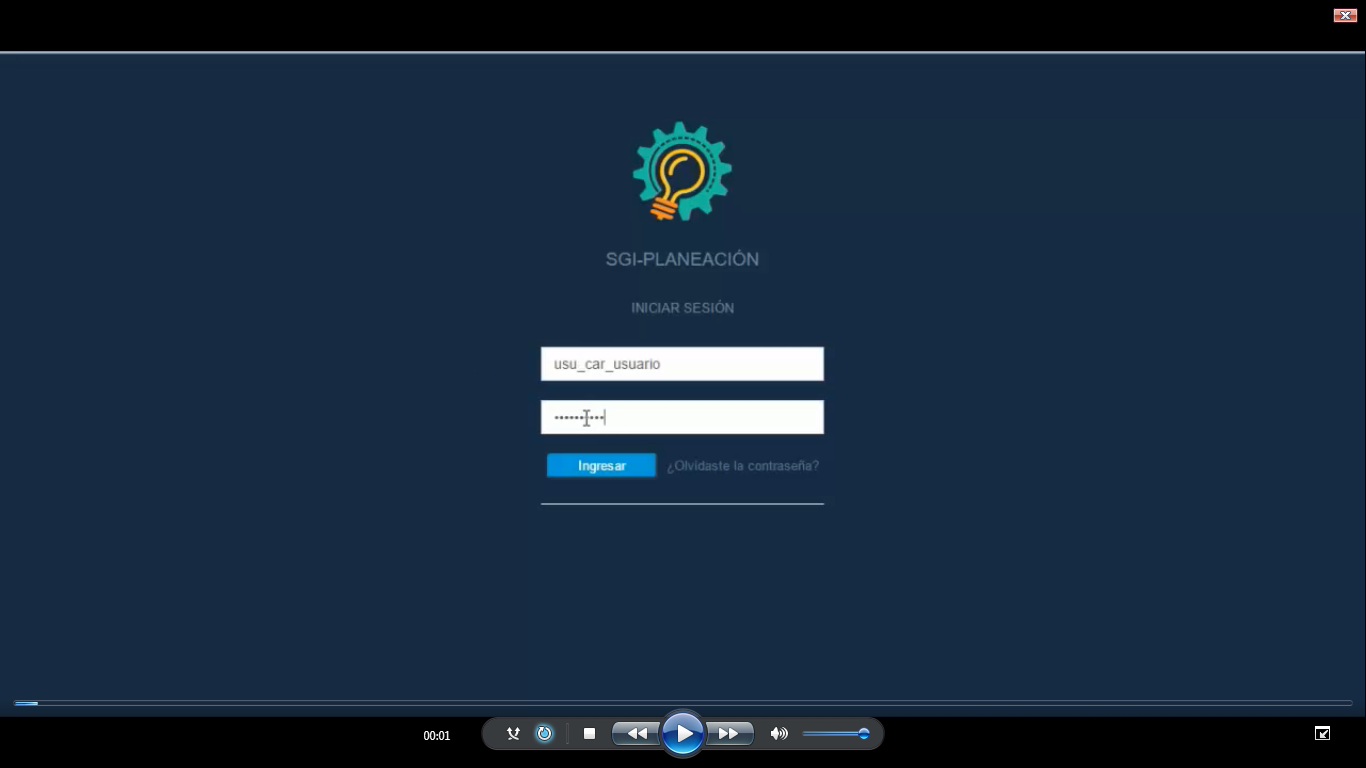
La última notificación de alerta corresponde a dos semanas de atraso en la documentación del CAR, este correo se dirige al responsable del CAR, al jefe directo y al gerente general de la compañía, además se realiza memorando sobre la hoja de vida de la persona, esta notificación se envía como se observa en la Figura 35

**Figura 35.** Notificación de alerta 6

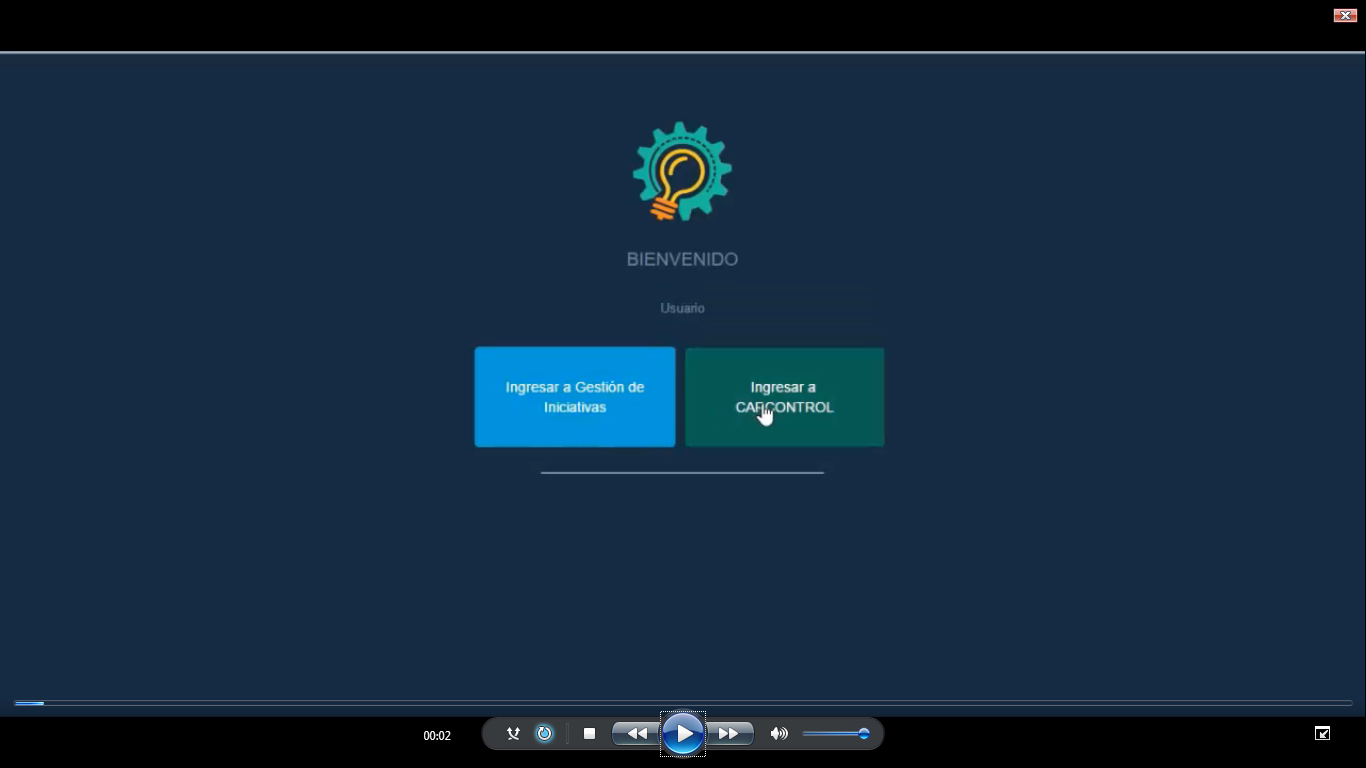


La última funcionalidad implementada es el ingreso a la aplicación, para lo cual se definieron dos tipos de usuarios: el responsable del CAR y el administrador del sistema, estos roles fueron asignados a cada uno de los usuarios de la compañía a través del directorio activo de la misma, la pantalla de ingreso está dividida en dos componentes, la primera es el formulario de inicio de sesión como se ve en la Figura 36 y la segunda en el ingreso al módulo de gestión de CAR, en esta pantalla aparecen otros sistemas relacionado al negocio del área de planeación de la compañía como se observa en la Figura 37.

**Figura 36.** Pantalla de ingreso al sistema



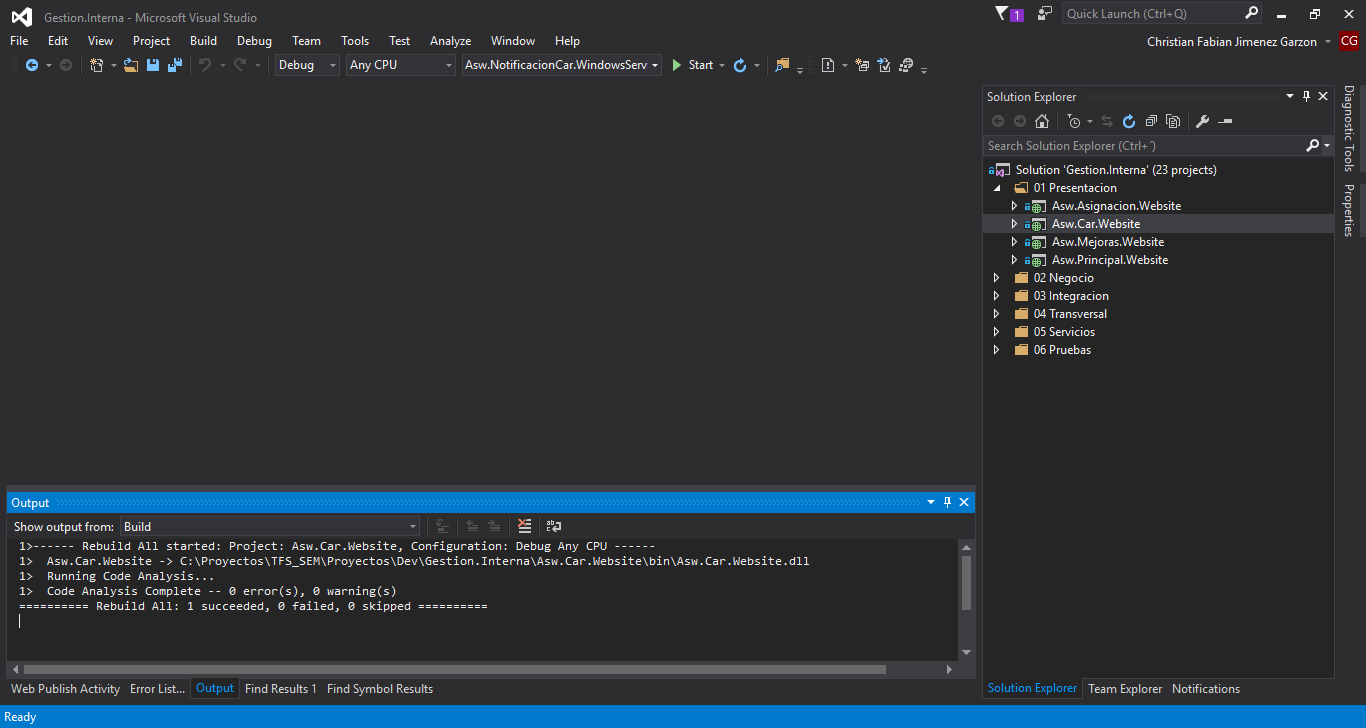
**Figura 37.** Pantalla de acceso al módulo de CAR



Para ver el funcionamiento completo del sistema, puede dirigirse a ver el video de la aplicación en funcionamiento (Ver anexo digital 12).

6.4.3 Ejecución del análisis estático de código. Para llevar a cabo esta tarea se utilizó la herramienta de análisis de código de Visual Studio, con la cual se examinó el código para buscar defectos y prácticas no recomendadas de programación. La revisión se ejecutó para cada una de las capas que componen la aplicación: Presentación, negocio, integración, transversal y servicios, en la Figura 38 se observa la ejecución del análisis de código estático para la capa de presentación, para ver la ejecución de las demás etapas se puede dirigir al documento de ejecución de análisis de código estático (Ver anexo digital 10).

**Figura 38.** Análisis estático de código capa de presentación



6.4.4 Diseño, construcción y ejecución de pruebas funcionales. El proceso de pruebas de Asesoftware se descompone en subprocesos, en los cuales se encuentran las pruebas de sistema, las cuales van a ser aplicadas al desarrollo del proyecto a través de pruebas funcionales, verificando el cumplimiento de todos los requerimientos funcionales considerando la aplicación de forma integral: software y hardware.

Dentro del ciclo de pruebas funcionales, se ejecutaron las pruebas de tipo Smoke Test, con el objetivo de revelar fallos simples que permitan determinar si las funcionalidades básicas del software se comportan correctamente.

Para realizar la ejecución de pruebas funcionales, se definió una estrategia dando prioridad a cada uno de los requerimientos del sistema como se observa en la Figura 39, el plan de pruebas completo se encuentra en el documento de plan detallado de pruebas (Ver Anexo Digital 14).

**Figura 39.** Priorización ejecución de pruebas funcionales



Luego de dar prioridad a cada uno de los requerimientos del sistema, se diseñaron y ejecutaron los casos de prueba por cada requerimiento.

A continuación, se presenta un ejemplo del diseño y ejecución de casos de prueba para el REQ01. El caso de prueba se denomina: Agregar más de cinco personas al Equipo CAR en el formulario de registro, en el cual se especifican las precondiciones, los pasos, salidas y evidencias del caso de prueba.

Precondiciones

* El usuario pertenece a la compañía Asesoftware
* Ingresar a la pantalla de registro CAR

Pasos

* Ingresar al sistema
* Ingresar al formulario de registro de un CAR.
* Escribir el nombre en el campo “Equipo CAR” de la persona que se desea seleccionar
* Dar click sobre el nombre de la persona que se desea seleccionar en la lista que aparece debajo del campo “Equipo CAR”
* Realizar los pasos seis veces más

Salidas

* El sistema le muestra al usuario una tabla debajo del campo “Equipo CAR” con cada una de las personas seleccionadas y un botón de “Eliminar”
* El sistema muestra en la parte inferior derecha de la pantalla un mensaje de advertencia refiriéndose a que no se pueden agregar más de cinco personas.

Evidencias

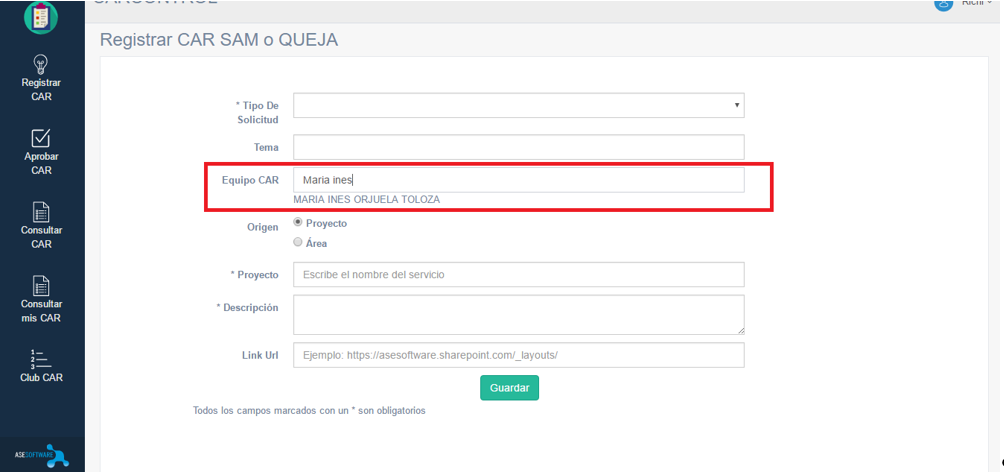
Escribir el nombre en el campo “Equipo CAR” de la persona que se desea seleccionar, ver Figura 40.

**Figura 40.** Evidencia 1 prueba funcional REQ01



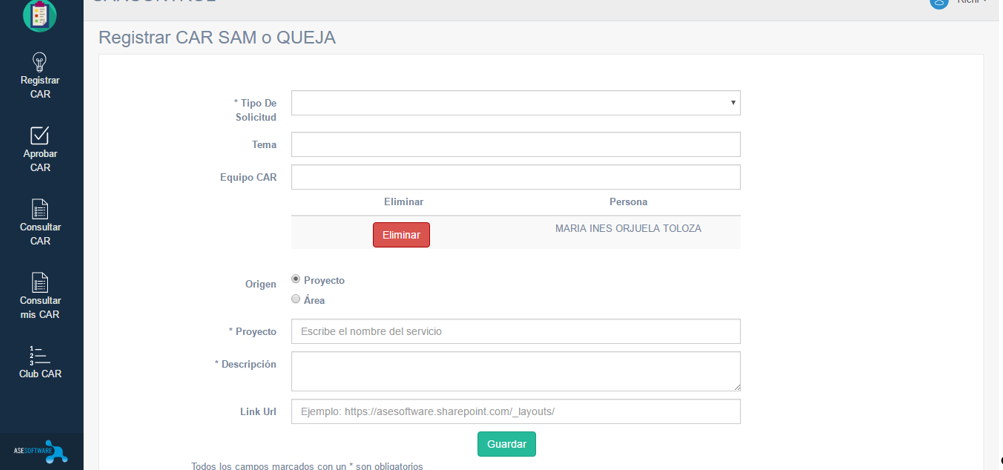
Dar click sobre el nombre de la persona que se desea seleccionar en la lista que aparece debajo del campo “Equipo CAR”, ver Figura 41.

**Figura 41.** Evidencia 2 prueba funcional REQ01



El sistema le muestra al usuario una tabla debajo del campo “Equipo CAR” con cada una de las personas seleccionadas y un botón de “Eliminar”, ver Figura 42

**Figura 42.** Evidencia 3 prueba funcional REQ01



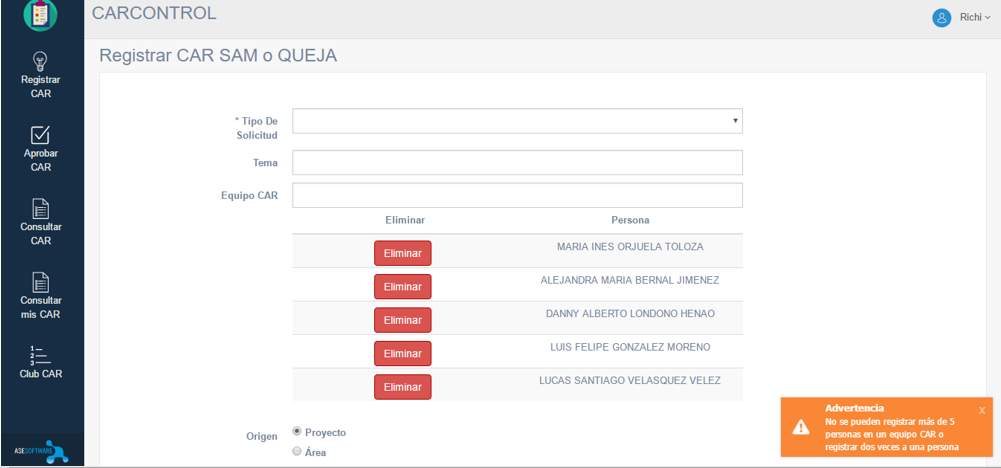
Realizar los pasos anteriores seis veces, ver Figura 43

**Figura 43**.Evidencia 4 prueba funcional REQ01



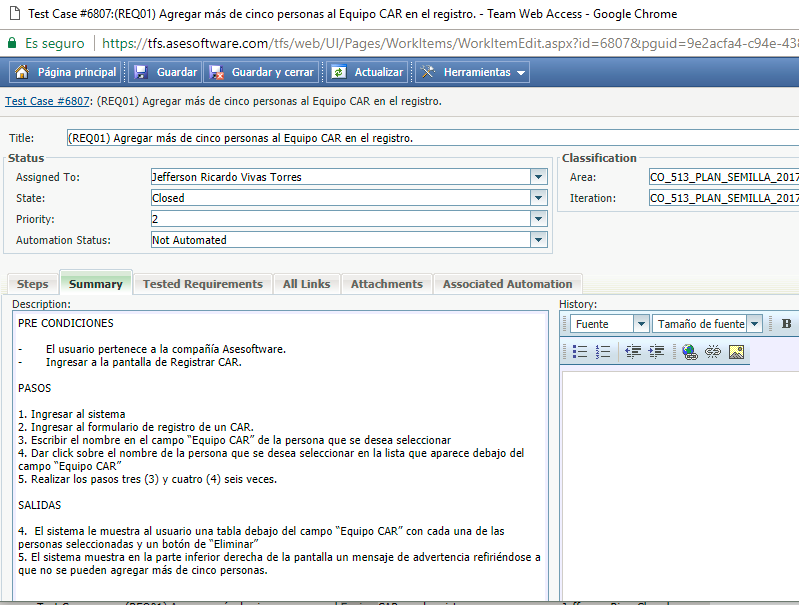
El sistema muestra en la parte inferior derecha de la pantalla un mensaje de advertencia refiriéndose a que no se pueden agregar más de cinco personas, ver Figura 44.

**Figura 44.** Evidencia 5 prueba funcional REQ01



El reporte de incidencias de las pruebas funcionales se registró en la herramienta Team Fundation Server(TFS) utilizada por la compañía, como se observa en la Figura 45.

**Figura 45.** Evidencia ejecución casos de prueba TFS



Cada uno de los casos de prueba diseñados, construidos y ejecutados para el proyecto, se pueden observar a detalle en (Ver Anexo Digital 15).

6.4.5 Ejecución de la integración de la aplicación. Como se puede observar en el documento de diseño detallado la integración de los componentes de software que componen la aplicación se llevó a cabo de manera continua, para llevar a cabo esta tarea se hizo uso de la herramienta de control de código fuente Team Foundation Server utilizada por la compañía en los proyectos de desarrollo de software.

6.4.6 Documentación del producto. Durante el desarrollo de esta actividad se obtuvo el manual de usuario de la aplicación (Ver anexo digital 11) en el cual se entregan las pautas necesarias para hacer uso del sistema y la descripción detallada del uso de cada una de las funcionalidades, además se obtuvo el documento del manual técnico (Ver anexo digital 9), en este se puede encontrar el paso a paso necesario para realizar la instalación y configuración de las herramientas necesarias para que la aplicación funcione correctamente.

## REALIZACIÓN DE LA TRANSICIÓN

6.5.1 Configuración del ambiente de producción. Para realizar esta actividad se siguieron las pautas descritas en el manual técnico de la aplicación donde se encuentran los pre requisitos de instalación en lo que tiene que ver con hardware, software, estaciones de trabajo y servidores, en este documento también se encuentran los pasos necesarios para la instalación e información relevante para dar soporte y futuros desarrollos sobre el aplicativo.

6.5.2 Instalación del software. En la sección del procedimiento de instalación descrita en el manual técnico de la aplicación (Ver anexo digital 11), se encuentran los pasos necesarios para la instalación de la base de datos del sistema, la instalación del servicio REST, la instalación del sitio web y la configuración del servicio Windows para el envío de correos.

6.5.3 Capacitación. De acuerdo al grado de usabilidad de la aplicación se decidió realizar una sesión de capacitación a las personas con rol de administrador las cuales pertenecen al área de planeación, luego se ejecutó una prueba sobre las personas participantes en la capacitación para verificar si era necesario o no dar una segunda charla la cual no fue necesaria.

# CONCLUSIONES

La implementación del servicio Windows y la funcionalidad de generación de alertas y notificaciones minimizaron los atrasos en el seguimiento a los CAR, con esta implementación se realiza diariamente una revisión de forma automática a los CAR que se encuentran irrespetando las reglas establecidas en la política SAM-033, notificando vía correo electrónico a los usuarios y liberando de realizar esta tarea al personal del área de planeación.

La construcción del módulo de consulta de CAR, incidió positivamente en el interés de los miembros de Asesoftware por participar activamente en la metodología propuesta por la compañía ya que, en la jornada de capacitación a los usuarios, estos mostraron gran interés en el uso de la aplicación al querer aparecer en la lista del club CAR que se publica periódicamente.

La funcionalidad de gestionar la información de los CAR impactó positivamente en relación al proceso que se manejaba anteriormente para esta actividad, luego de la implementación del sistema el administrador puede estar enterado de manera inmediata cuando un responsable de CAR registre o realice un avance en el desarrollo del mismo, identificando fácilmente a que fase y estado debe continuar el CAR, contando con una interfaz gráfica usable y eficaz para registrar los cambios pertinentes.

El diseño y la construcción del módulo de registros de solicitudes de CAR agilizó el proceso mediante el cual el administrador del sistema recibe las peticiones de levantamiento de los CAR, debido a que esta tarea es realizada ahora de forma autónoma por los funcionarios de la compañía a través de la funcionalidad del sistema dispuesta para esta actividad, en lugar de buscar personalmente a la persona encargada de este proceso en las instalaciones de la compañía.

Durante la fase de entrevistas con el cliente se pudo establecer que ser perspicaz durante la ejecución de las preguntas, documentar cada detalle, ser puntuales en las preguntas luego de la primera entrevista y tomar diferentes puntos de vista teniendo en cuenta las necesidades que tiene el cliente, ayuda favorablemente a entender el problema y a resolver satisfactoriamente el levantamiento de los requerimientos.

Se pudo concluir que, al realizar un análisis adecuado del problema, no se presentan dificultades en el momento realizar el diseño detallado y el modelo entidad relación, los cuales reflejan el comportamiento del sistema, además se puede establecer fácilmente el tipo de arquitectura que se va a usar y se hace una estimación de puntos funcionales más precisa para definir el costo en tiempo en la etapa de desarrollo del proyecto.

En relación a la experiencia de la pasantía, el crecimiento personal y profesional aumentaron considerablemente al comprender el proceso completo que se ejecuta para el desarrollo de un proyecto de software en la vida real, el rol que desempeña cada persona en el equipo de trabajo, analistas de sistemas, analistas de requerimientos, analistas de pruebas, arquitectos de software, directores de proyectos y el valor del trabajo en equipo de cada una de las personas relacionadas al proyecto.

# REFERENCIAS

[1] Entity Framework, ADO.NET [En línea]. Disponible en: https://msdn.microsoft.com/es-es/library/bb399567(v=vs.110).aspx [Accedido: 24-Abr-2017].

[2] AZAUSTRE, Carlos. ¿Qué es angularJS? [En línea]. Disponible en internet: https://carlosazaustre.es/empezando-con-angular-js/ [Accedido: 27-Abr-2017]

[3] W3schools, Inicio de Bootstrap. [En línea]. Disponible en internet: https://www.w3schools.com/bootstrap/bootstrap\_get\_started.asp [Accedido: 29-Abr-2017]

[4] Microsoft, C#. [En línea]. Disponible en internet: https://docs.microsoft.com/es-es/dotnet/csharp/csharp [Accedido: 05-May-2017].

[5] W3C, Html5. [En línea]. Disponible en internet: https://www.w3.org/TR/html5/ [Accedido: 05-May-2017].

[6] JSON, Introducción a Json. [En línea]. Disponible en internet: http://www.json.org/json-es.html [Accedido: 06-May-2017].

[7] RAMIREZ, Oliver. Asp .net asp net y framework. [En línea]. Disponible en internet: http://www.academia.edu/6324536/Asp\_.net\_asp\_net\_y\_framework [Accedido: 07-May-2017].

[8] Ingenio DS, Patrones arquitectónicos. [En línea]. Disponible en: https://ingeniods.wordpress.com/2013/09/16/patrones-arquitectonicos/ [Accedido: 09-May-2017].

[9] Asesoftware, «NOSOTROS», Asesoftware. [En línea]. Disponible en: http://asesoftware.com/site/nosotros/. [Accedido: 13-May-2017].

[10] Asesoftware, Políticas [EN línea]. Disponible en: http://asesoftware.com//web/index.php/es/sgc/politicas.html [Accedido: 17-May-2017].

[11] Asesoftware S.A.S., «Asesoftware - activos de proceso». [En línea]. Disponible en: http://asesoftware.com/web/index.php/es/sgc/activos-de-proceso.html#gh. [Accedido: 27-May-2017].

[12] Microsoft, Introduction a Windows Service Applications. [En línea]. Disponible en internet: https://msdn.microsoft.com/es-es/library/d56de412(v=vs.110).aspx [Accedido: 30-May-2017].

[13] Data Transfer Object, DTO. [En línea]. Disponible en: https://stackoverflow.com/questions/1612334/difference-between-dto-vo-pojo-javabeans/1612671 [Accedido: 14-Jun-2017].

[14] Microsoft, Typescript para Visual Studio 2015. [En línea] Disponible en internet: https://www.microsoft.com/en-us/download/details.aspx?id=48593 [Accedido: 19-Jun-2017].

[15] Microsoft, Ensamblados en Common Language Runtime. [En línea]. Disponible en internet: https://msdn.microsoft.com/es-es/library/k3677y81(v=vs.110).aspx [Accedido: 19-Jun-2017].

[16] Model View Controller, MVC [En línea]. Disponible en: http://www.w3ii.com/es/mvc\_framework/mvc\_framework\_introduction.html [Accedido: 19-Jun-2017]

[17] Inyección de dependencias, Patrón de diseño [En Línea]. Disponible en: https://msdn.microsoft.com/es-es/communitydocs/net-dev/csharp/inyeccion-de-dependencias [Accedido: 23-Jun-2017].

[18] Proxy, Patrón de diseño [En Línea]. Disponible en: https://informaticapc.com/patrones-de-diseno/proxy.php [Accedido: 28-Jun-2017].

[19] Repositorio, Patrón de diseño. [En Línea]. Disponible en: https://msdn.microsoft.com/en-us/library/ff649690.aspx [Accedido: 1-Jul-2017]

[20] Facade, Patrón de diseño. [En Línea]. Disponible en: http://programacion.net/articulo/patrones\_de\_diseno\_xi\_patrones\_estructurales\_facade\_1014 [Accedido: 4-Jul-2017].

[21] Microsoft, Programación Asíncrona en .NET 4.5 [En Línea]. Disponible en: https://msdn.microsoft.com/es-es/communitydocs/net-dev/dev/programacion-asincrona-en-net [Accedido: 14-Jul-2017]

[22] Patrones de diseño software [En Línea]. Disponible en: https://informaticapc.com/patrones-de-diseno/proxy.php [Accedido: 09-Sep-2017].

[23] Patrones de diseño, Patrones estructurales. [En Línea]. Disponible en: http://programacion.net/articulo/patrones\_de\_diseno\_xi\_patrones\_estructurales\_facade\_1014 [Accedido: 09-Sep-2017]