Universidad de las Ciencias Informáticas

Facultad # 3

Título: Módulo de control de acceso vehicular mediante reconocimiento de matrícula para el Sistema Integral de Control de Acceso Daxcess.

Trabajo de diploma para optar por el título de Ingeniero en Ciencias Informáticas.

Autor: RafaelRubén Jorge González.

Tutores: Ing. Jorge Arce Martínez.
MSc. Enier Alarcón Barbán

La Habana, agosto de 2020

DECLARACIÓN DE AUTORÍA

Declaro ser el autor de la presente tesis y reconocer a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo.

Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

_____

Rafael Rubén Jorge González

Autor

_____ _____

Ing. Jorge Arce Martínez MSc. Enier Alarcón Barbán

Tutor **Tutor**

DATOS DE CONTACTO

Tutor: Ing. Jorge Arce Martínez.
Edad: 32.
Ciudadanía: Cubano.
Institución: XETID.
Título: Ingeniero en Ciencias Informáticas.
Cargo: Especialista Superior en TIC.

Tutor: MSc. Enier Alarcón Barbán.
Edad: 29.
Ciudadanía: Cubano.
Institución: Universidad de las Ciencias Informáticas (UCI).
Título: Ingeniero en Ciencias Informáticas.
Categoría Docente: Profesor Asistente.

AGRADECIMIENTOS

Expreso mi mayor muestra de gratitud al Ing. Jorge Arce Martínez y al MSc. Enier Alarcón Barbán, quiénes han seguido muy de cerca el desarrollo de este trabajo, además de prestarme su colaboración en todo momento.

No puedo pasar por alto a mis compañeros y mi familia por apoyarme en el transcurso de esta carrera universitaria.

Por último, a todas aquellas personas que de una manera u otra han intervenido en este trabajo, para hacer posible la culminación de mis estudios.

DEDICATORIA

A mis padres y demás familiares por el apoyo que me han brindado en esta etapa de mi vida.

A todos esos profesores y compañeros que me ayudaron a cumplir este reto de la universidad.

A la Revolución Cubana que me dado la posibilidad de graduarme como profesional.

RESUMEN

La empresa XETID se encuentra desarrollando el Sistema Integral de Control de Acceso, Dacxess para la gestión de control de acceso que se realizan en instalaciones o áreas de interés. En el presente trabajo se realiza el análisis, diseño e implementación del proceso de control de acceso vehicular mediante el reconocimiento de matrículas a través del flujo de video de cámaras con el objetivo de ayudar a evitar la ocurrencia de hechos extraordinarios como el hurto de bienes en empresas, el acceso indebido a las áreas restringidas o información sensible. Actualmente este proceso en las empresas cubanas es gestionado de forma manual, pues no existe un control de acceso vehicular centralizado que gestione este acceso automáticamente.

Con la implementación del módulo se contribuye a disminuir el tiempo de gestión de estos lugares y a facilitar su gestión. Este software posibilita el ahorro de recursos al país y ayuda a mejorar este proceso en las instalaciones cubanas.

Palabras clave: reconocimiento automático de matrículas, reconocimiento de matrículas, control de acceso vehicular,matrícula.

ABSTRACT

The XETID company is developing the Comprehensive Access Control System, Dacxess for the management of access control carried out in facilities or areas of interest. In this work, the analysis, design and implementation of the vehicle access control process through the recognition of license plates through the video flow of cameras is carried out in order to help prevent the occurrence of extraordinary events such as theft of property in companies, improper access to restricted areas or sensitive information. Currently this process in Cuban companies is managed manually, since there is no centralized vehicle access control that manages this access automatically.

With the implementation of the module, it helps to reduce the management time of these places and to facilitate their management. This software enables the country to save resources and helps to improve this process in Cuban facilities.

Keywords: Automatic Number Plate Recognition, License Plate Recognition,vehicular access control, license plate.

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

INTRODUCCIÓN

En la sociedad actual el desarrollo de las Tecnologías de la Información y las Comunicaciones (TIC) ha sido un factor importante en el desarrollo de la economía mundial y de los cambios rápidos en la sociedad, convirtiendo la información que antes estaba en los libros, en servicios especializados y bases de datos. Este desarrollo ha dado paso a la invención de nuevos bienes y servicios, mejorando así el procesamiento, transmisión y almacenamiento de la información, siendo de beneficio para la sociedad y los negocios.

Para las personas y las empresas siempre ha sido necesario proteger bienes materiales y áreas de intereses. Como solución a esta necesidad, inicialmente se realizaban guardias con vigilantes que utilizaban recursos como puertas con candados y cerraduras o cerrojos mecánicos. Actualmente con el uso de sistema de control de acceso se hace más sencillo y eficiente realizar este proceso.[1]

Los sistemas de control de acceso se basan en tres procesos fundamentales como son la identificación, la autenticación y la autorización. Estos son necesarios utilizarlos cuando se desea controlar el peaje dispuesto en los ramales de conexión con otras carreteras, los bienes y recursos de una entidad, el acceso de personas y vehículos a instalaciones o áreas de interés. Además, estos sistemas ofrecen independencia, automatización y seguridad en el proceso a controlar y a su vez funcionan como un mecanismo dinámico y eficiente. También permiten conceder o denegar el paso a un área restringidas en función de los parámetros de seguridad establecidos por una organización y llevar el tiempo real del registro cronológico de las entradas y salidas.

El uso de estos sistemas ayuda a evitar la ocurrencia de hechos extraordinarios como el hurto de bienes en empresas, el acceso indebido a las áreas restringidas o información sensible. Algunos de estos sistemas pueden controlar recursos como puerta, torniquete o talanquera. En estos se utilizan cuatro tipos de técnicas de identificación y autenticación los cuales se representan como algo que solamente el individuo conoce,ejemplo de ello, una clave, una tarjeta, una huella digital, una voz o simplemente patrones de escritura diseñados por un usuario[2].

Estos se clasifican en Sistemas de control de Acceso Autónomos, que controlan recursos sin estar conectados a un ordenador o sistema central debido a que no guardan registro de eventos, y en Sistemas de control de Acceso en Red, donde a través de un ordenador local o remoto permite llevar un registro de todas las operaciones realizadas. El proceso de control de acceso consiste en verificar si una entidad, por ejemplo, una persona, vehículo u ordenador que esté solicitando acceso a un recurso tiene los derechos necesarios para hacerlo. Existen varios sistemas que ejercen este proceso y entre ellos se puede encontrar el control de acceso vehicular en la cual estará centrada la investigación.

Los sistemas de control de acceso vehicular se implementan con el objetivo de llevar el control de los vehículos que circulan por un espacio público o privado, donde se permita una mayor seguridad con registros de entradas y salidas, horarios y grupos de acceso enzonas permitidas para aquellos vehículos que estén autorizados. También utilizan el control de acceso físico como puerta o torniquete o barrera física para controlar el acceso a un área determinada. Estos sistemas se encargan de realizar la lectura de etiquetas que se adhieren al parabrisas y cuentan con un circuito integrado que almacena información[3]yde lastarjetas de identificación por radiofrecuenciao RFID por sus siglas en inglésde largo alcance, las cuales también se ubican en el parabrisas de los vehículos para la autenticación e identificación. Además, usan sensores que verifican la identificación de quien solicita el acceso en función de la credencial que se muestre,así como pueden estar provistos de un teclado para la introducción de una contraseña o de un lector biométrico.

En Cuba algunas entidades realizan el control de los vehículos a través del sistema FE-AVL, sistema automático de localización vehicular. Este sistema permite garantizar una gestión óptima de los vehículos en una organización, minimizar costo de combustible y tener controlada la mercancía que se mueve o disponer en tiempo real de la ubicación de un auto o equipo que soporte un módulo GPS. También se utiliza el control vehicular en el cobro de peaje en Varadero donde se logró la implementación de barreras vehiculares que se desplazan de forma automática, mediante un mecanismo detector de presencia y el cobro de peaje se realiza de forma manual. Aunque en Cuba se realice este tipo de control, existe una carencia de sistemas que realicen este proceso automáticamente.

Actualmente la empresa XETID se encuentra desarrollando el Sistema Integral de Control de Acceso, Dacxess, el cual posee una arquitectura basada en componentes, que le permite la incorporación de diferentes mecanismos de control a una entidad o local, lo que puede ser utilizado para el control de asistencia y restricción de acceso. Los métodos incorporados al sistema están orientados solo al control de persona, por lo que no son registrados los diferentes medios de transportes que acceden. El no llevar el control de estos vehículos da paso a robos de carga y actos criminales que ocurren frecuentemente en las carreteras o en las instalaciones, siendo los más denunciados a las autoridades. Estos actos delictivos no solo presentan problemas de seguridad y provocan pérdidas económicas en las entidades, sino que también traen consecuencias para la competitividad de la empresa, dañando la imagen de esta y desalentando la inversión.

En el análisis de la problemática planteada anteriormente se define como

Problema de la investigación:

¿Cómo controlar el acceso de los medios de transporte mediante el Sistema Integral de Control de Acceso Daxcess?

Siendo el **objeto de estudio:** control de acceso vehicular, enmarcado en el **campo de acción:**el control de acceso vehicular por reconocimiento de matrícula.

Como **objetivo general** se propone: desarrollar un módulo para el Sistema Integral de Control de Acceso Daxcess que permita el control de acceso vehicular mediante reconocimiento de matrícula.

Para guiar el desarrollo de la presente investigación surgen las siguientes preguntas científicas:

1. ¿Cuáles son las características de los sistemas de control de acceso vehiculares?
2. ¿Cuáles son los requisitos funcionales para el módulo de control de acceso vehicular?
3. ¿Cómo implementar un módulo para el control de acceso vehicular mediante el reconocimiento de matrículas en el Sistema Integral de Control de Acceso, Daxcess?
4. ¿Cómo validar el software de control de acceso vehicular para un correcto funcionamiento en el Sistema Integral de Control de Acceso, Daxcess?

Con el propósito de dar cumplimiento a lo anteriormente planteado, se elaboraron las siguientes **tareas de investigación**:

1. Estudio de los sistemas homólogos para conocer características regulares en la implementación de los sistemas de control de acceso vehicular.
2. Identificación de los requisitos funcionales del módulo de control de acceso vehicular mediante el reconocimiento de matrículas para el Sistema Integral de Control de Acceso, Daxcess.
3. Implementación de las funcionalidades de los sistemas de control de acceso vehicular.
4. Aplicación de los diferentes tipos de pruebas funcionales y no funcionales.

Para la realización de la investigación se emplearon varios métodos científicos de investigación como:

Métodos Teóricos:

Modelación: Este posibilitó crear una abstracción del módulo para el sistema Dacxess con el objetivo de explicar su funcionamiento.

Análisis-Síntesis: Con el empleo de este método se pudo analizar su base teórica, conocer y comprender las etapas del proceso de control de acceso vehicular y descomprimirlo en pequeños procesos para poder alcanzar los objetivos trazados en esta investigación.

Métodos Empíricos:

Entrevista: Este posibilitó conocer el criterio respecto al problema presentado e identificar posibles funcionalidades a incorporar en la solución.

El presente documento se encuentra conformado por tres capítulos como se expresa a continuación:

Capítulo 1: Fundamentación teórica del módulo para el Sistema Integral de Control de Acceso Daxcess: en él, se exponen los principales conceptos vinculados a la investigación y se presentan las bases teóricas fundamentales, características de las tecnologías, lenguajes de programación utilizados y se describen las herramientas y metodologías utilizadas.

Capítulo 2: Análisis y diseño del módulo de control de acceso vehicular para el Sistema Integral de Control de Acceso Daxcess: donde se abarca el modelo del dominio que describe los controles de acceso y se describe la propuesta de solución para resolver el problema planteado. Se realiza la captura de los requisitos funcionales y no funcionales,así como los elementos fundamentales del diseño y de la arquitectura.

Capítulo 3: Construcción y validación del sistema. En el cual se exponen las métricas y pruebas utilizadas para la validación del diseño propuesto realizado, así como los resultados y la implementación de la aplicación.

Capítulo 1. Fundamentación teórica del módulo para el Sistema Integral de Control de Acceso Daxcess.

1.1 Introducción

En el presente capítulo se exponen las definiciones fundamentales y los elementos principales que justifican y soportan técnicamente la investigación. Se realiza un análisis sobre las funcionalidades de la aplicación. Se mencionan características de las tecnologías, lenguajes de programación utilizados y la metodología a utilizar en el desarrollo del módulo para el Sistema de Control de Acceso Daxcess.

1.2 Principales conceptos.

Control de acceso es un sistema automatizado que permite de forma eficaz, aprobar o negar el paso de personas o grupo de personas a zonas restringidas en función de ciertos parámetros de seguridad establecidos por una empresa, comercio, institución o cualquier otro.Los controles de acceso también hacen posible llevar un registro automatizado de los movimientos de un individuo o grupo dentro de un espacio determinado.[4]

Control de acceso: es el proceso de interacción entre un sujeto y un objeto que resulta un flujo de información de uno al otro donde el sujeto es la entidad que recibe información y el objeto es la entidad que lo provee. Está basado en tres conceptos fundamentales la identificación, autenticación y autorización. [5]

La página oficial de SERACIS define **control de Acceso** es un componente de seguridad que monitorea electrónicamente y controla el tráfico a través de cosas como puertas, entradas y ascensores; surge de la necesidad antigua de proteger bien esos recursos.[6]

El sitio web de TANGOID define que el **control de acceso** consiste en la verificación de si una entidad (una persona, ordenador, etc…) solicitando acceso a un recurso tiene los derechos necesarios para hacerlo. Un control de acceso ofrece la posibilidad de acceder

a recursos físicos (por ejemplo, a un edificio, a un local, a un país) o lógicos (por ejemplo, a un sistema operativo o a una aplicación informática específica). [7]

Para la investigación se optó emplear el siguiente concepto de control de acceso.

Otra definición según la página oficial de VISERCO un **control de acceso** es un sistema de seguridad que permite mantener una diversa cantidad de opciones al tratarse de estar protegidos, pues lo que no solo se limita a un área o espacio como tal, sino a un control total, lo que protege nuestra seguridad de forma: remota, móvil, integrada o física. Estos permiten brindarle al usuario una verificación completa de la identidad, basándose en un rasgo determinante o por opciones que puede ofrecer, esto con el fin de autorizar entradas o accesos a lugares e información que de manera libre no se pueden obtener, abarca tanto a entes físico como lógicos, lo cual es ideal para guardar datos confidenciales. [8]

Control de accesode personas es un sistema electrónico que restringe o permite el acceso de un usuario o persona a un área específica validando la identificación por medio de diferentes tipos de lectura y a su vez controlando el recurso por medio de un dispositivo eléctrico como un electroimán, pestillo, entre otros. [9]

Los **sistemas de control de accesos vehicular** se implementan para tener el control de los vehículos que circulan por un espacio público o privado, asegurando el paso a los vehículos permitidos y restringiendo a aquellos que no estén autorizados. Al integrar un sistema de control de accesos vehicular, se puede tener el control total, tanto de los residentes como de los visitantes. [10]

El **reconocimiento automático de matrículas**(**ANPR**por sus siglas en inglés) es un método de vigilancia en masa que utiliza reconocimiento óptico de caracteres en imágenes para leer las matrículas de los vehículos. Es utilizado en los carriles de acceso a los aparcamientos, tanto restringidos como públicos. En los restringidos suele utilizarse para permitir o denegar el acceso en función de que el vehículo esté o no autorizado. En los públicos se utiliza para identificar cada vehículo que accede al aparcamiento y asociar el ticket y la matrícula. [11]

Según la página oficial de TTCS define**ANPR:** hace referencia a que cuando una matrícula sea captada por una cámara, ésta a partir de la imagen tomada, pueda reconocer los caracteres y enviar en texto alfanumérico que leyó de la matrícula a una base de datos. Generalmente el ANPR se utiliza para automatizar los accesos en los residenciales o fraccionamientos generando clientes morosos y de pronto pago. [11]

El **reconocimiento de matrículas** (**License Plate Recognition** o**LPR**por sus siglas en inglés**):** esto hace referencia a que la necesidad es solamente visualizar una matrícula en diversas condiciones de luminaria o ambientales. [11]

Sistemas autónomos: Son sistemas sencillos que no se encuentran conectados entre sí, y se encargan de controlar los accesos a uno o varios lugares, pero como se trata de sistemas sin complicaciones, no se guardan registros del mismo.[8]

Sistemas de acceso en red: Son más usados y por ende más completos, pues los mismos están interconectados a un computador que registra tanto los accesos que se permiten, como las horas y a quienes se les da dicha autorización, por lo que tienen mayor monitoreo. [8]

Sistema de autenticación: Son aquellos métodos que se completan con la verificación de una clave de acceso, huella, código, por voz, facial, por matrícula y demás métodos que, al proveerlos, puede dar acceso al lugar o información correspondiente. [8]

Según el sitio web de REDESZONE la **autenticación:** verifica las identidades, por diferentes métodos (algo que se sabe, algo que se tiene, algo que se es). [12]

Para este trabajo se empleóel siguiente concepto de autenticación.

Otra definición según el sitio de OAS la **autenticación:** es el proceso que debe seguir un usuario para tener acceso a los recursos de un sistema o de una red de computadores. Este proceso implica identificación (decirle al sistema quién es) y autenticación (demostrar que el usuario es quien dice ser). La autenticación por sí sola no verifica derechos de acceso del usuario; estos se confirman en el proceso de autorización. [13]

Sistema de autorización: Es aquel método que se completa después de la autorización de un sistema o un ente ajeno al usuario, lo cual da permiso para obtener los datos o acceso a un determinado lugar privado. [8]

Otra definición según el sitio web de REDESZONE la **autorización** es lo que define a qué recursos de sistema el usuario autenticado podrá acceder. Que haya logrado pasar la instancia de la autenticación, no significa que podrá utilizar el sistema por completo como administrador. De acuerdo a una serie de reglas, normas y regulaciones propias de cada red interna, se determina que el usuario A tendrá acceso a los recursos X e Y. Sin embargo, el usuario B sólo podrá acceder al recurso Z. [12]

Según la página oficial de DEFINICIÓN ABC la **autorización** no es otra cosa que un permiso que faculta a alguien para que pueda desplegar una determinada acción, es decir, ella le permite a un individuo hacer algo que, aunque esté prohibido o vedado para el común de la gente o por x situación, podrá hacerlo porque dispone de ese permiso que es siempre expedido por una autoridad competente en la materia que se trate. [14]

Para este trabajo se empleó el siguiente concepto de autorización.

El sitio web de DEFINICIÓN define que la**Autorización:** es la parte de un sistema operativo que protege los recursos del sistema, de modo tal que sólo puedan ser utilizados por los usuarios que cuentan con permiso para eso. Por lo tanto, es una especie de permiso. Consiste en dar consentimiento para que otros hagan o dejen de hacer algo. También puede consistir en que una persona en concreto, que por determinados motivos no pueda realizar una acción necesaria para ella, establezca mediante el correspondiente documento acreditativo que autoriza a un familiar o amigo para que lleve a cabo aquella por él. [15]

Para este trabajo se empleó el siguiente concepto de identificación.

Identificación: es la acción y efecto de identificar o identificarse. Se refiere a reconocer siuna persona o una cosa es la misma que se busca, hacer que dos o más cosas distintas seconsideren como una misma, llegar a tener las mismas creencias o propósitos que otrapersona, dar los datos necesarios para ser reconocido. La identificación está

vinculada ala identidad, que es el conjunto de los rasgos propios de un sujeto o de una comunidad. Dichos rasgos caracterizan al individuo o al grupo frente a los demás.[15]

1.3 Sistemas de control de acceso.

Los sistemas de control de acceso (SCA) son un mecanismo que en función de la identificación ya autentificada permite acceder a datos o recursos. Este sistema restringe o permite el acceso de un usuario a un área específica validando la identificación por medio de diferentes tipos de lectura (clave por teclado, tags de proximidad o biometría) y a su vez controlando el recurso (puerta, torniquete o talanquera) por medio de un dispositivo eléctrico como un electroimán, cantonera, pestillo o motor. Los SCA se clasifican en dos tipos: Sistemas de control de Acceso Autónomos y Sistemas de control de Acceso en Red [16]. A continuación, se describen ambos.

1.3.1 Sistemas de control de Acceso Autónomos.

Los **sistemas de control de acceso autónomos** son sistemas que permiten controlar una o más puertas, sin estar conectados a un ordenador o a un sistema central, por lo tanto, no guardan registro de eventos. Aunque esta es la principal limitante, algunos sistemas tampoco pueden limitar el acceso por horarios o por grupos de puertas, esto depende de la robustez de la marca. Dentro de esta categoría se encuentran los lectores inteligentes, los controladores de una puerta y los controladores multipuertas. [17]

Ilustración 1 Sistemas de control de acceso autónomos [Fuente propia].

En el ejemplo (ver**¡Error! No se encuentra el origen de la referencia.**) muestra una puerta con un sistema de control de cerradura electrónica el cual permite el paso del personal y está controlado por un sistema de control de acceso biométrico de huellas dactilares del modelo 5000A que puede activar una alarma en caso de una violación de acceso.

1.3.1 Sistemas de control de Acceso en Red.

Los **sistemas de control de acceso en red** son sistemas que además de abrir puertas, barreras y tornos, se integran a través de un ordenador local o remoto, donde se hace uso de un software de control que permite llevar un registro de todas las operaciones realizadas sobre el sistema con fecha, horario, autorización, entre otros. Van desde aplicaciones sencillas hasta sistemas muy complejos y sofisticados según se requiera.[17]

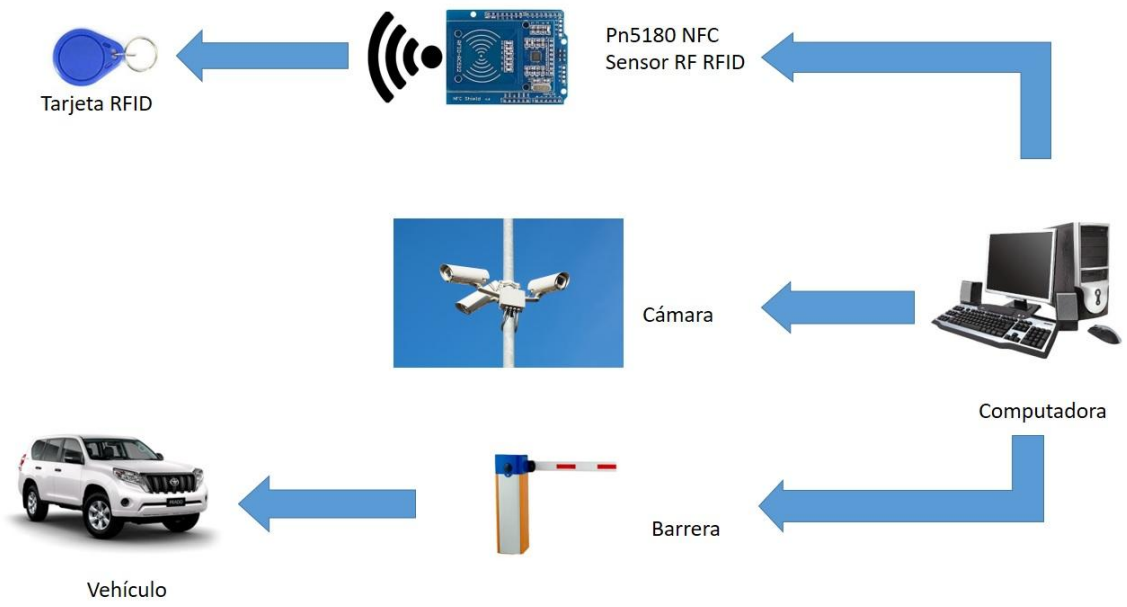


Ilustración 2 Sistema de control de acceso en red [Fuente propia].

En el ejemplo (ver**¡Error! No se encuentra el origen de la referencia.**)se encuentra un vehículo donde el chofer debe contener una tarjeta RFID la cual es captada por un sensor de proximidad en este caso de modelo Pn5180. Las cámaras de vigilancia reconocen la placa del vehículo identificado estas dos herramientas de reconocimiento y envían los datos recolectados a la computadora o servidor donde se encuentra el software de control de acceso vehicular.

En la investigación se optará por utilizar un Sistemas de control de Acceso en Red para la implementación del módulo de control de acceso vehicular para el Sistema Integral de Control de Acceso Daxcess, ya que se utilizará un software de control de acceso a través de un ordenador para llevar la gestión de todas las operaciones.

1.4 Soluciones Existentes.

Existen innumerables sistemas especializados y diseñados para realizar el control de acceso vehicular. A partir de la creciente demanda de estos sistemas se ha mejorado las formas de controlar los vehículos con el objetivo de mejorar este proceso de forma

eficiente y segura. En la búsqueda de soluciones durante la investigación, se encuentran en el mundo algunos sistemas de control de acceso vehicular que se necesitan analizar para tener una idea de cuál será la línea a seguir. Para ello se describirán algunos de estos sistemas y se resumirán sus principales características en función de la investigación.(verTabla 1)

El software **Genius Accesos** desarrollado por la empresa SYON es una aplicación avanzada con una interfaz intuitiva que ofrece todas las funcionalidades necesarias para el control de accesos en edificios, centros de negocios y fábricas, controlando la apertura tanto de tornos para acceso peatonal, como de puertas de acceso a despachos e incluso barreras para acceso de vehículos. El control de acceso vehicular lo gestiona a través de tarjetas controladoras y terminales con la posibilidad de realizar la identificación mediante la lectura de tarjeta de proximidad RFID. El sistema funciona incluso en ausencia de comunicación con el Servidor donde está instalado el software Genius Accesos. Este posee características tales como el control de accesos offline con monitorización de eventos en tiempo real (online), los perfiles de acceso y niveles de seguridad por usuario y por grupos, la función Anti-passback para evitar accesos fraudulentos, que posee una interfaz intuitiva para un fácil manejo del sistema. Está desarrollado en tecnología .Net., yutiliza Bases de Datos SQL Server y Access y un Sistemas operativos: Windows 2003/XP/VISTA/7. Posee una licencia de software privativo.[18]

El software **3LPR** desarrollado por la empresa i+D3 es un sistema de reconocimiento de matrículas LPR utilizado para identificar y registrar los vehículos que acceden o salen de un aparcamiento, consiguiendo un gran control de matrículas de vehículos. Este no sólo está enfocado a los parkings, sino que puede ser utilizado en todas aquellas instalaciones que necesiten controlar, vigilar y tener un registro de todos los vehículos que traspasan un determinado acceso. Ejemplo son los garajes privados de empresas, centros comerciales, peajes, hospitales, etc. Este se integra con otros sistemas, como 4Park o 3Access, enfocado a realizar un control de vehículos en instalaciones como parkings públicos o de empresas. Permite, además de identificar el número de matrícula, realizar un control de accesos automático a aquellos conductores que tengan permisos de aparcamiento, mediante la matrícula y teléfono móvil. Posee características tales como una máxima fiabilidad de reconocimiento de matrículas, un sistema LPR apto para coches, motos y camiones, incrementa la seguridad y control de vehículos, permite Accesos automáticos a

través de matrícula, un Sistema ANPR llave en mano y realiza accesos por móvil. Posee una licencia de software privativo.[19]

El software **gCARS** desarrollado por la empresa DATYS es un sistema de gestión de vehículos. Está diseñado, para el control de vehículos y equipos tecnológicos, en cualquier tipo de empresa u organización. Es capaz de controlar los recursos de una empresa, logrando una eficiente administración, el monitoreo constante de los procesos, reducir costos y/o gastos y lograr ahorro. Cuenta con un conjunto de funcionalidades y reportes, orientados a mejorar la gestión, tanto del parque automotriz, como del combustible y los recursos complementarios. Tiene como características una interfaz sencilla y amigable, compatibilidad con los diferentes navegadores, es un sistema multientidad, permite asociar los usuarios a los diferentes centros de costos a los que tienen permisos de realizar operaciones contables y cuenta con una estructura modular. Posee una licencia de software privativo.[20]

Tabla 1. Soluciones existentes.

Características	Genius Accesos	3LPR	gCARS	Daxcess
Gestiona el registro de acceso vehicular.	Sí	Sí	Sí	No
Utilizan recursos físicos en el control de acceso	Sí	Sí	No	Sí
Utilizan el reconocimiento de matrícula.	No	Sí	No	Sí
Software propietario.	Sí	Sí	Sí	Sí
Integración con sistemas nacionales de Gestion de recursos.	No	No	No	Si

Dado que no se puede integrar las soluciones existentes al Sistema Daxcess debido a que ninguna posee integración con sistemas nacionales de gestión de recursos,se decidió implementar un módulo para el Sistema Integral de Control de Acceso, Daxcess que realice el control de acceso vehicular utilizando el mecanismo ya implementado de

reconocimiento de matrículas de vehículos OpenALPR, ya que esto le permite un ahorro económico al país y soluciona la carencia del control vehicular.

1.5 Metodología de desarrollo.

Las Metodologías de desarrollo de Software consisten en lograr la elaboración de un sistema informático eficiente, que cumpla con los requerimientos planteados. También imponen un proceso disciplinado sobre el desarrollo de software con el objetivo de hacerlo más predecible y eficiente. Una metodología de desarrollo de software tiene como objetivo aumentar la calidad del software que se produce en todas y cada una de sus fases de desarrollo. Estas deben ser adaptadas a las características de cada proyecto y estas se pueden dividir en ágiles y robustas.[21]

1.5.1. Metodología de desarrollo de software.

En el proceso de desarrollo de aplicaciones informáticas en la XETID se emplea el Proceso de Desarrollo de Software Prodesoft, como metodología de desarrollo de software, la misma será empleada en la solución que se propone para la realización del módulo para el Sistema de Control de Acceso Daxcess. Este proceso tiene como objetivo la producción eficiente de un producto de software con una planificación y estimación de recursos predecibles. Se basa en una combinación entre varios modelos de procesos como el de Componentes y el Iterativo e incremental. Al desarrollar el software por componentes aumenta el nivel de reutilización, escalabilidad, facilidadde mantenimiento y permite que al modificar o actualizar no afecte a otras partes del sistema. Al ser también Iterativo e Incremental se realiza a través de iteraciones donde en cada una se obtiene una versión funcional mejorada del producto y así se va desarrollando. Según [22] este proceso plantea que el ciclo de vida de un proyecto está compuesto por cinco fases inicio, modelación, construcción, explotación experimental y despliegue (ver Ilustración 3).

Ilustración 3. Metodología de desarrollo de software, Prodesoft.

En la fase de **Inicio** se logra una visión preliminar de la problemática a resolver y se definen los recursos relevantes para la ejecución del proyecto. Se definen los objetivos y el alcance del proyecto, se identifican los involucrados y ejecutores, se estima de manera general las actividades a realizar durante todo el ciclo de desarrollo del proyecto, se establece la estrategia a seguir para realizar la modelación del negocio y la captura de requisitos. De ser necesario se estiman los recursos materiales que deberán ser adquiridos.

En la fase de **Modelación** se capturan las partes esenciales del sistema, donde se identifican los procesos de negocio fundamentales y se aceptan los requisitos funcionales, obteniéndose la línea base de la arquitectura y una estrategia de construcción de la aplicación aprobada por los implicados en el proyecto. El hito fundamental de esta fase es la liberación de la arquitectura de sistema, datos y despliegue.

En la fase de **Construcción** se aclaran los requisitos restantes y se completa el desarrollo del sistema sobre una base estable de la arquitectura. En esta fase todas las características, componentes y requisitos deben ser integrados, implementados y probados en su totalidad, obteniendo una versión liberada del producto.

En la fase de **Explotación Experimental** se convierte la versión liberada del producto en una solución estable, donde se eliminan los errores que surgen durante las pruebas y se obtiene una certificación funcional y de seguridad del producto.

En la fase de **Despliegue** se instala y configura el sistema para un ambiente de producción real, se capacita al personal que usará la aplicación y se continúa dando soporte durante la explotación del sistema, culminando de ser preciso con transferencias tecnológicas.

Estas fases se descomponen en iteraciones, donde una iteración es un recorrido a través de todas las disciplinas de desarrollo (Modelación de proceso de negocio, Definición de Requisito, Diseño, Implementación y Prueba).Cada iteración va mejorando o añadiendo componentes a los anteriores.

1.6 Lenguajes y tecnologías utilizadas.

Se mostrarán las principales tecnologías y los lenguajes de programación que fueron utilizadas para la realización del módulo. Abordando las principales características de cada una de las herramientas.

1.6.1 Marco de trabajo.

En el marco de trabajo la investigación se utilizará el framework Qt y NodeJS.Qt 5.9.7 es un framework multiplataforma orientado a objetos con un armazón escrito en C++. Posee un preprocesador, el Recopilador del Meta-objeto (MOC), quese usa para extender el idioma de C++ con los rasgos como los signos y hendeduras. La aplicación o librería que lo usan pueden ser compilados por cualquier recopilador de C++. La herramienta de Entorno Integrado de Desarrollo (IDE) que se utilizará con este framework es el Qt Creator 4.7.1. [23]

Node.js v12.17.0.0posee una arquitectura orientada a eventos y está basado en el motor V8 de Google. Además, utiliza un modelo de entrada y salida sin bloqueo controlado por eventos, de esta manera lo hace un entorno ligero y eficiente. También permite que se ejecute cada petición de forma asíncrona en el servidor. Es multiplataforma y de código

abierto. Tiene como ventajas que la compilación de este se realiza en tiempo de ejecución, Just In Time (JIT).Esto trae consigo una mayor optimización a las funciones que más veces sean llamadas.A través de clúster permite tener una escalabilidad alta, se puede expandir el código añadiendo módulos de forma fácil gracias al Node Package Manager (NPM) y permite un alto rendimiento en proyectos donde se necesita la ejecución en tiempo real. Como lenguaje de programación este utiliza JavaScript y se utilizará con la herramienta Node-RED v1.0.2.[24]

1.6.2 Lenguaje Unificado de Modelado.

Lenguaje de Modelaje Unificado (UML) es un lenguaje de modelación estandarizado para especificar, visualizar, construir y documentar todos los elementos de un sistema de software. Además, proporciona un vocabulario y reglas para combinar y construir representaciones, modelos conceptuales y físicos del sistema. También permite representar varios modelos, combinando notaciones específicas de cada uno. Tiene como ventaja que es fácilmente entendible, se puede usar para diferentes tipos de sistemas y se consolida en muchas de las notaciones y conceptos más usadas orientados a objetos.[25]

1.6.3 Lenguajes de programación.

C++ es un lenguaje de programación orientado a objetos que toma la base del lenguaje C. Este posee tres paradigmas de la programación como lo es la programación estructurada, la programación genérica y la programación Orientada a Objetos En la actualidad, C++ es un lenguaje versátil, potente y general[26]. Este posee ventajas como el alto rendimiento, ya que puede hacer llamadas directas al sistema operativo.Es un lenguaje compilado para cada plataforma, posee gran variedad de parámetros de optimización y se integra de forma directa con el lenguaje ensamblador.Es un lenguaje actualizado ya que tiene muchos años y permite crear, relacionar y operar con datos complejos.Es multiplataforma y es extendido por lo que casi cualquier programa o sistema están escritos o tienen alguna parte escrita en estos lenguajes.[27]

JavaScript es un lenguaje de programación interpretado, dialecto del estándar ECMAScript. Se define como orientado a objetos, basado en prototipos, imperativo, débilmente tipado y dinámico. Este es muy versátil, se utiliza tanto en el lado del servidor como con el cliente y además en el desarrollo de aplicaciones móviles.[28]

1.6.4 Herramienta de Modelado.

Visual Paradigm 5.0 es una herramienta UML profesional que soporta el ciclo de vida completo del desarrollo de software. En el proceso de análisis y diseño orientados a objetos, construcción, pruebas y despliegue, permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Tiene disponibilidad en múltiples plataformas como: Windows y Linux. Es fácil de utilizar, instalar y actualizar. Presenta un ambiente gráfico agradable e intuitivo para el usuario. Posee un diseño centrado en casos de uso y enfocado al negocio que genera un software con mayor calidad. Permite la generación de código, compatibilidad para diferentes lenguajes, el desarrollo de aplicaciones con rapidez, facilidad y calidad con un menor costo de producción yla interoperabilidad entre diagramas ya que a partir de un diagrama puede obtener otro que guarde relación con el mismo.[29]

1.6.5 Herramienta de desarrollo.

Qt Creator 4.7.1 es un entorno de desarrollo integrado o IDE multiplataforma que está programado en C++, JavaScript y QML. Fue creado por Trolltech para la creación de aplicaciones con el frameworks Qt. Qt está diseñado para desarrollar aplicaciones e interfaces de usuario y desplegarlas a múltiples sistemas operativos tanto móviles como de escritorio. Esta tiene como ventajas que permite a un equipo de desarrolladores compartir un proyecto a través de diferentes plataformas de desarrollo, con una herramienta común para desarrollo y depurado. Posee un completado automático, tiene un depurador visual, herramientas para la rápida navegación del código y el editor de código ofrece resaltado de sintaxis, así como de código. También tiene dos editores visuales integrados: Qt Designer para la creación de interfaces de usuario de widgets Qt y Qt Quick Designer para el desarrollo de interfaces de usuarios animadas con el lenguaje QML. [30]

Node-RED v1.0.2 es un motor de flujos con enfoque al **internet de las cosas**(**Internet of things** o **IoT** por sus siglas en inglés), el cual permite definir gráficamente flujos de servicios, a través de protocolos estándares y ofrecer integración con apis de terceros. Es una herramienta visual y está programada en NodeJS. Los servicios que ofrece se realizan conectando entre si los nodos que se definieron para una solución. Tiene como característica que se pueden crear fácilmente nuevos nodos e instalarlos,por lo que cualquier persona u organización puede crearse sus propios nodos, adaptando el motor de flujo a las necesidades de su negocio.Puede ejecutarse desde dispositivos limitados como una Raspberry y en plataformas complejas como IBM Bluemix, Azure IoT o Sofia2 Platform. El editor de flujos de Node-RED consiste en una sencilla interfaz en HTML, accesible desde cualquier navegador.[24]

Los flujos programados en Node-RED se almacenan internamente en formato JSON y son portables entre distintas instalaciones de Node-RED. En cuanto a su instalación, existen dos alternativas el Modo Standalone que es donde se ejecuta como un proceso NodeJS independiente del resto de procesos y el Modo Embebido que es donde forma parte de una aplicación mayor, de forma que es responsabilidad de esta controlar el ciclo de vida del propio Node-RED. Ambas poseen seguridad tanto a nivel de control de acceso con usuario y contraseña, como con certificado SSL para acceder al editor por protocolo seguro HTTPS.

1.6.6 Base de Datos

PostgreSQL 9.3.2.1 es un sistema Gestor de Bases de Datos de Código libre muy potente y robusto. Además, es un sistema multiplataforma y se caracteriza por ser orientado a objetos y liberado bajo la licencia Berkeley Software Distribution (BSD). Tiene como ventajas su gran escalabilidad, ya que es capaz de ajustarse al número de CPU y a la cantidad de memoria disponible de forma óptima, su estabilidad y confiabilidad ya que no se han presentado nunca caídas de la base de datos, debido a su capacidad de establecer un entorno de alta disponibilidad. Otra ventaja es el uso del pgAdmin que es una herramienta gráfica capaz de administrar las bases de datos de forma fácil e intuitiva, y su Extensibilidad ya que se dispone de una gran variedad de extensiones en diferentes lenguajes de programación.[31]

1.7 Conclusiones

El estudio de los principales conceptos teóricos relacionados con el control de acceso permitió ganar en comprensión y adoptar criterios para continuar la investigación. Luego de realizar un estudio de las soluciones existentes, se llega a la conclusión de que ninguno se puede integrar al sistema Daxcess ya que son de software privativo o no cubren todas las necesidades que requiere para resolver el problema planteado de esta investigación. Se proponen las herramientas informáticas a utilizar como QT Creator y Node-RED con los lenguajes de programación C++ y JavaScript, como sistemas gestores de Base de Datos PostgreSQLy la metodología a utilizar ProDeSoft para el desarrollo del software que se utilizará en la elaboración del módulo informático para el Sistema Integral de Control de Acceso Daxcess.

Capítulo 2. Análisis y diseño delmódulo de control de acceso vehicular para el Sistema Integral de Control de Acceso Daxcess.

2.1 Introducción.

En el presente capítulo se especifican las características principales del módulo y se exponen los patrones de diseño utilizados. Además, se representa su proceso mediante el modelado del dominio y se identifican los requisitos funcionales y no funcionales. Se describe la arquitectura empleada para el desarrollo del módulo de control de acceso vehicular mediante el reconocimiento de matrículas en el Sistema Integral de Control de Acceso Daxcess.

2.2 Descripción de la solución.

El plug-in OpenALPR se encuentra en Servidor de Analítica del Sistema Integral de Control de Acceso Daxcess,el cual está diseñado para el reconocimiento de matrículas de vehículos a través del flujo video de las cámaras. El mismo una vez que reconozca una o varias matriculas permitirá comprobar su registro en el sistema y dará acceso al vehículo involucrado. Luego enviará estos accesos a la capa de Integración por WebSocket y esta enviará a la aplicación **Vista** los accesos a través de un TCPServer recibiendo los accesos el TCPCliente. La aplicación vista añadirá a la base de datos las matrículas para su registro y entrenamiento para un posterior reconocimiento de matrículas.

2.3 Modelo del dominio.

El Modelo de Dominio es la representación de conceptos u objetivos que son importantes para comprender y describir las clases más importantes dentro del contexto del sistema. Es el principal mecanismo para comprender el dominio del problema en cuestión.La Ilustración 4 muestra proceso se muestra como se realiza el control de acceso vehicular en la mayoría de las empresas cubanas, en el cual el guardia atiende al chofer de cada vehículo comprobando su acceso a las instalaciones, registrando por escrito su acceso en

el Libro de Registro y mediante una Puerta o barrera que generalmente se encuentra en los puntos de acceso puede controlar el acceso a los vehículos.[32]

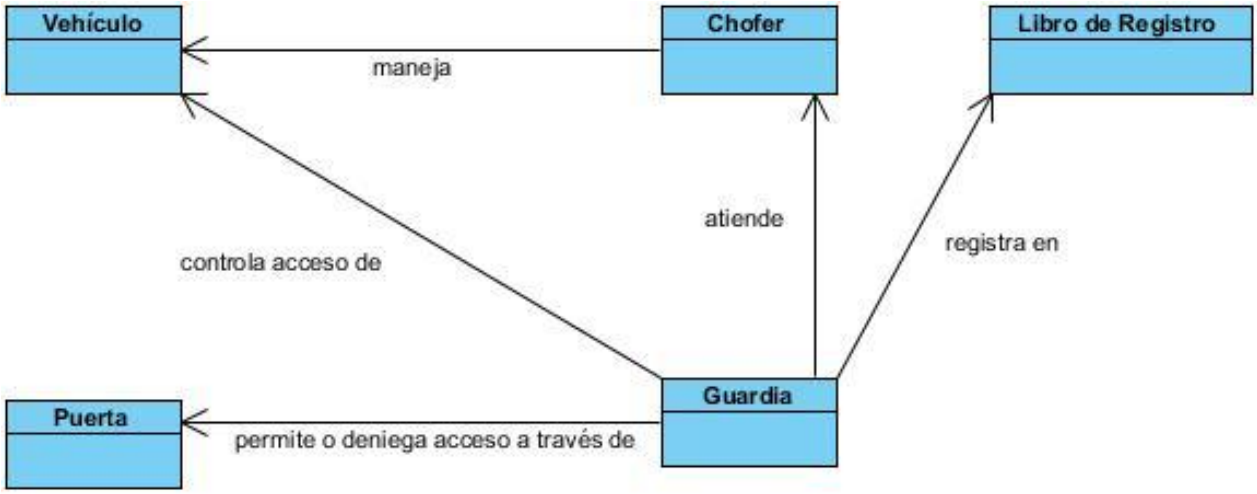


Ilustración 4. Modelo del dominio.

A continuación, se describen cada una de los conceptos representados en el modelo de dominio. (verTabla 2)

Tabla 2. Definición de clases del dominio.

Concepto	**Descripción**
Vehículo	Es un medio de transporte de personas o cosas.
Chofer	Es una persona que por oficio, conduce un automóvil.
Guardia	Es el personal encargado de las funciones de vigilancia o defensa.
Puerta	Armazón de madera, hierro u otra materia, que engoznada o puesta en el quicio y asegurada por el otro lado con llave, cerrojo u otro instrumento, sirve para impedir la entrada o salida.
Libro de Registro	Es un archivo o documento donde se plasma la información de las entradas realizadas durante un período de tiempo determinado.

2.4 Definición de requisitos.

La definición de requisitos especifica las condiciones o capacidades que el sistema debe cumplir y las restricciones bajo las cuales debe operar, para lograr un entendimiento entre el especialista y el equipo de desarrollo para especificar las necesidades reales de forma que satisfaga sus expectativas.[33]

Los requisitos permiten a los ingenieros de software, entender el problema en el que trabajarán. Incluye un conjunto de tareas que conduce a comprender cuál será el impacto del software sobre el negocio. Se especifica lo que el cliente quiere y cómo van a interactuar los usuarios finales con el software. *[34]*

Los requisitos de software se clasifican en: requisitos funcionales y requisitos no funcionales.

2.4.1Técnicas empleadas en la captura de requisitos.

La captura de los requisitos es una actividad del proceso de desarrollo de software, donde el equipo de desarrollo extrae de diversas fuentes de información las necesidades que debe cumplir el sistema para alcanzar los objetivos en el software que se desea implementar. Dada la complejidad que implica este proceso, se desarrollan técnicas que permitan realizarlo de forma eficiente. De las distintas técnicas existentes se utilizaron las siguientes:[35]

Entrevista: Es una de las técnicas de la ingeniería de requisitos y de uso ampliamente extendido. Las entrevistasle permiten al programador analizar problemas a través de preguntas que le permitan comprender los objetivos trazados.Para llegar a la solución se realizaron pequeños encuentros con el jefe del centro donde se fueron obteniendo las necesidades del software.

Análisis de documentación: Esta técnica analiza los distintos sistemas ya desarrollados que estén estrechamente relacionados con el sistema que se desea construir. Se

estudiaron las documentaciones en los sitios webs oficiales de los sistemas de control de acceso ya existentes como Genius Accesos, gCARS y 3LPR, prestándole atención principalmente a sus funcionalidades, características y las formas de realizar el proceso de control de acceso vehicular .

2.4.2 Requisitos funcionales.

Requisitos Funcionales: definen el comportamiento interno de un software, son condiciones que el sistema ha de cumplir. Estos muestran las funcionalidades que deben satisfacerse para cumplir con las especificaciones de software. [36]

RF 1. Agrupación de requisitos: Gestionar Vehículo
RF1.1 Adicionar vehículo.
RF1.2 Modificar vehículo.
RF1.3 Eliminar el vehículo.
RF1.4 Listar los vehículos.

RF2. Agrupación de requisitos: Gestionar Reporte
RF1.1 Adicionar reporte.
RF 1.2 Modificar reporte.
RF1.3 Eliminar el reporte.
RF1.4 Listar los reportes.

RF3. Entrenar matrícula
RF4. Entrenar todas las matrículas.
RF5. Definir conexión a base de datos postgresql.
RF 5.1 Iniciar conexión a base de datos postgresql.
RF 5.2 Cerrar conexión a base de datos postgresql.
RF6. Definir conexión TCP Cliente.
RF 6.1 Iniciar conexión TCP.
RF 6.2 Cerrar conexión TCP.
RF7. Mostrar datos del Chofer.
RF8. Mostrar en pantalla los datos del vehículo identificado.

RF9. Generar alerta visual que indique el resultado del acceso.

2.4.3 Especificación de los requisitos de software.

La metodología de desarrollo Prodesoft utilizada para la realización del módulo, plantea que en las fases deModelación y Construcciónse acepten y especifiquen todos los requisitos funcionales para luego implementar.Con la especificación de requisitos se describen detalladamente cada uno de ellos para lograr un mejor entendimiento por parte del equipo de desarrollo.

A continuación se muestra en la

Tabla *3* y Tabla 4la descripción de la funcionalidad Adicionar vehículoy Modificar vehículoperteneciente al requisito funcional Gestionar Vehículo. Permitiendo un mejor entendimiento delmódulo,haciéndose uso del artefacto propuesto por la metodología.

Tabla 3. Especificación de requisito “AdicionarVehículo” del grupo de requisito “Gestionar Vehículo”.

Conceptos tratados	**Conceptos**	**Atributos**
	AdicionarVehículo.	• No. Matrícula. • Entidad. • Marca. • Modelo. • Año de Fabricación. • No. Circulación. • No. Chasis. • No. Motor. Color vehículo.
Precondiciones	**Precondiciones**	**Pre-requisitos**
	No aplica.	No aplica.

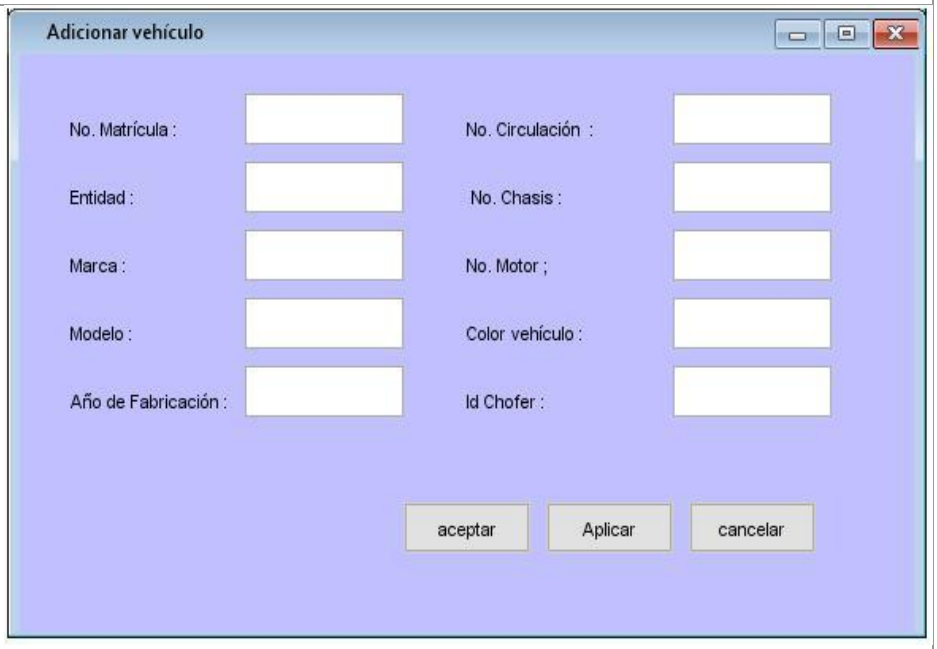
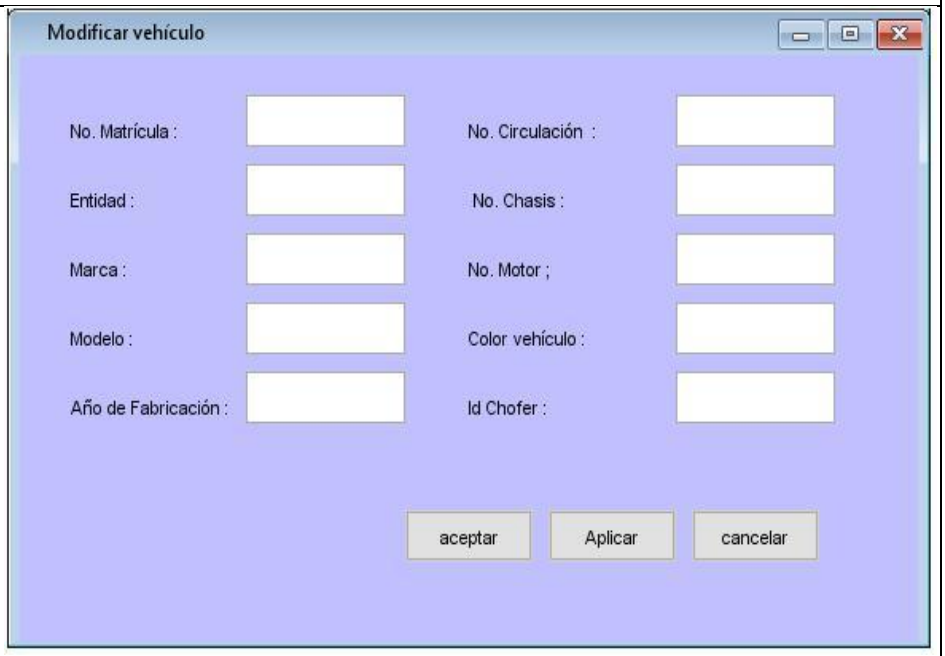
Descripción	1. Se activa la interfaz hacienda clic en el botón adicionar. 2. Se muestran los campos para insertar un vehículo. 3. Insertar los datos en los campos del vehículo. 4. Se guarda la información a la base de datos.
Validaciones	No aplica.
Post-condiciones	Se mostrará los datos del vehículo insertado.
Post-requisitos	Navegar entre los hechos.
Prototipo de interfaz	

Tabla 4. Especificación de requisito “Modificar Vehículo” del grupo de requisito “Gestionar Vehículo”.

Conceptos tratados	Conceptos	Atributos
	Modificar Vehículo.	• No. Matrícula. • Entidad. • Marca. • Modelo. • Año de Fabricación. • No. Circulación. • No. Chasis. • No. Motor.

		Color vehículo.
Precondiciones	**Precondiciones**	**Pre-requisitos**
	Debe haber un o varios vehículos en la base de datos postgresql.	No aplica.
Descripción	1. Se activa la interfaz hacienda clic en el vehículo previamente insertado. 2. Se muestran los campos del vehículo con los datos almacenados y se modifican. 3. Insertar los datos en los campos del vehículo a modificar. 4. Se guarda la información a la base de datos.	
Validaciones	No aplica.	
Post-condiciones	Se mostrará los datos del vehículo modificado.	
Post-requisitos	Navegar entre los hechos.	
Prototipo de interfaz		

2.4.3 Requisitos no funcionales.

Requisitos No Funcionales:son propiedades que hacen al producto atractivo, usable, rápido y confiable. Se refieren a todos los requisitos que no describen información a guardar, ni funciones a realizar. También se conocen como un conjunto de características

de calidad, que son necesarias para tener en cuenta al diseñar e implementar el software. [36]

RNF 1. Usabilidad.

RNF 1.1. El módulo no debe permitir el cierre de una interfaz hasta que todas las interfaces derivadas de esta hayan sido cerradas.

RNF 1.2. Se debe utilizar un tamaño de fuente apropiado para facilitar la lectura.

RNF 1.3. El sistema debe presentar mensajes de error que permitan identificar el tipo de error al usuario.

RNF 1.4. Interfaces uniformes y con los mismos colores y diseños.

RNF 1.5. El sistema debe mostrar el nombre de la funcionalidad que se esté utilizando en la parte superior izquierda de la interfaz.

RNF 1.6. El sistema para identificar las herramientas y opciones que posee, debe mostrar el ícono correspondiente a cada una de ellas y en la mayoría de sus casos el nombre a su lado derecho.

RNF 2. Confiabilidad.

RNF 2.1. El sistema debe validar automáticamente la información contenida en los formularios de ingreso.

RNF 3. Rendimiento.

RNF 3.1. El sistema debe estar en capacidad de permitir el desarrollo en el futuro de nuevas funcionalidades, modificar o eliminar funcionalidades después de su construcción y despliegue.

RNF 3.2. El sistema debe ser capaz de mantener un rendimiento y una estabilidad adecuada ante la gestión de amplios volúmenes de datos.

RNF 4. Restricciones del diseño.

RNF 4.1. Serán utilizados como lenguajes de programación JavaScript y C++.

RNF 4.2. Se empleará PostgreSQL como base de datos relacional.

RNF 5. Requerimiento de Ayuda y Documentación.

RNF 5.1. El sistema tendrá siempre visible la opción de Ayuda, lo que posibilitará un mejor aprovechamiento por parte de los usuarios de sus funcionalidades.

RNF 5.2. El sistema deberá estar completamente documentado, cada uno de los componentes de software que forman parte del mismo deberán estar debidamente documentado tanto en el código fuente como en el manual de usuarios.

RNF 6. Portabilidad.

RNF 6.1. El sistema debe funcionar en sistemas operativos de la familia GNU/Linux y Windows.

RNF 7. Software.

RNF 7.1. El sistema operativo soportado por el software son Linux y Windows en las computadoras del cliente y el servidorrespectivamente.

RNF 7.2. Las computadoras clientes y servidor deben contener las librerías de Qt 4.7.4.

RNF 7.3. El servidor de base de datos es PostgreSQL –v9.3.

RNF 8. Hardware (HDW).

RNF 8.1. La computadora cliente debe poseer:

1. Cliente: HDD 50GB y4 GB RAM o superior.
2. Servidor: HDD 1000GBy 16 GB RAM o superior.

RNF 8.2. La memoria RAM: 1 GB para la computadora cliente y servidor.

RNF 8.3. La Interfaz de red local a utilizar (Ethernet).

2.5 Arquitectura del sistema.

La arquitectura de software es la estructura del sistema y es un diseño que se crea en etapas tempranas del desarrollo del software. Este presenta un conjunto de patrones que guían la construcción de un software. Esta estructura representa un diseño del sistema que tiene como propósito principal satisfacer los atributos de calidad del sistema, y servir como guía en el desarrollo. También identifica los elementos más importantes de un sistema, así como sus relaciones, es decir da una visión global del sistema.

2.5.1Diseño de arquitectura de software.

Arquitectura basada en Plug-in

El patrón basado en plug-in define las reglas generales de organización en términos de un patrón y las restricciones en la forma y la estructura de un grupo numeroso y variado de sistemas de software [37]. Su objetivo es establecer una estructura para todos los

componentes del sistema. Un plug-in es una aplicación que aporta una función nueva a un programa informático y se considera cualquier pieza de software que extienda o cambie el comportamiento o interfaz de una aplicación. Esta aplicación adicional es ejecutada por la aplicación principal y se conoce por sus siglas en inglés API[38].

Las arquitecturas orientadas a plug-ins proveen una manera de extender las capacidades de una aplicación adicionando nuevas funcionalidades[39].Pueden ser varias interfaces de plug-in, pero un plug-in implementa exactamente una de esas interfaces. No es muy común que un complemento implemente varias interfaces. Además, un complemento puede heredar de una clase. Dentro del código de plug-in, las clases de la biblioteca se utilizan para realizar tareas comunes a varios plug-ins. La biblioteca contiene rutinas generales, que de otra manera tendrían que ser implementadas por un número de complementos o la aplicación principal. Toda la funcionalidad adicional es proporcionada por los complementos individuales, que pueden pertenecer a diferentes categorías plug-in según la interfaz de plug-in implementada. (verIlustración 5)

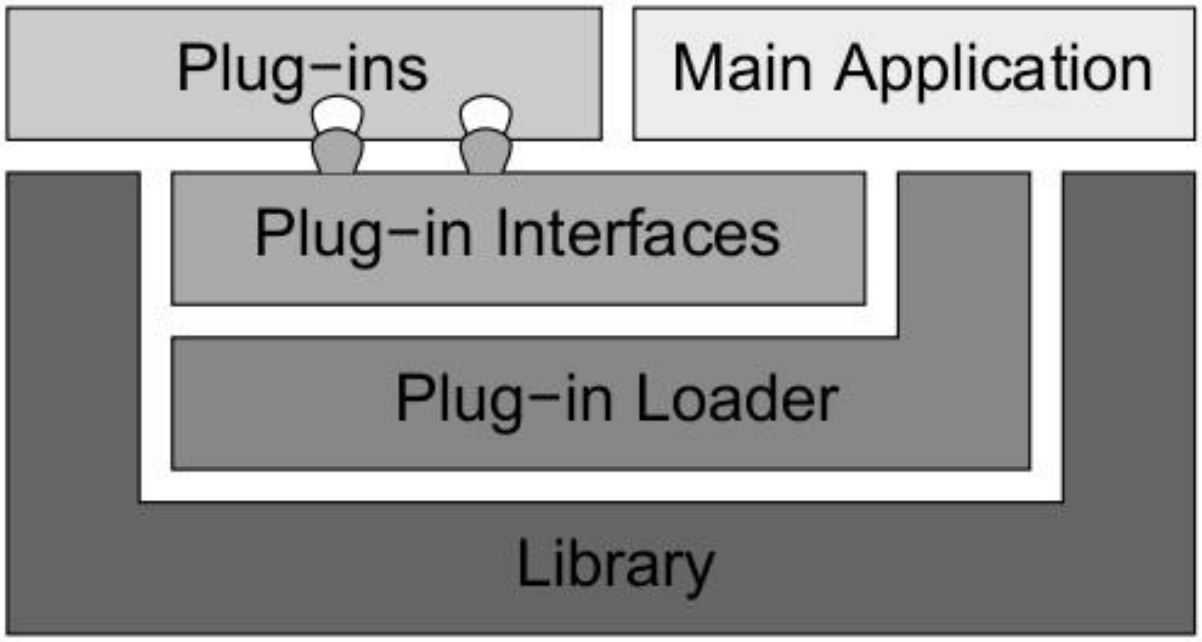


Ilustración 5. Estilo Arquitectónico de Plug-in basada en aplicación *[38]*.

El Plug-inLoader administra todos los complementos y por cada tipo de complemento puede haber un cargador de complemento adicional. Normalmente, los complementos se

inicializan primero y se recuperan sus nombres. Cuando la aplicación principal o la biblioteca desea acceder a un complemento, se llama al cargador de complementos. La aplicación principal se implementa en la parte superior de la biblioteca y tiene acceso a todosplug-ins a través del cargador de plug-in y de las interfaces de plug-in. Todas las partes del sistema que no se puedan modelar como plug-ins pertenece a la aplicación principal. Este estilo no contiene ninguna variante arquitectónica y se ve de forma independiente. Se enfoca en una aplicación que pueda soportar varios plug-in permitiendo que se extienda en tiempo de ejecución mediante la carga dinámica de módulos o clases que no conoce durante la compilación. Esta arquitectura de plug-in basado en el mecanismo que ofrece framework Qt permite un alto nivel de funcionamiento a la aplicación desarrollada. También las aplicaciones pueden ser extendidas mediante plugins haciendo uso de la clase(PluginLoader), la cual permite detectar y cargar los plugins en tiempo de ejecución. Siendo así un sistema extensible y reusable, cubriendo las necesidades existentes en el proyecto, Sistema Integral de Control de Acceso Daxcess.[38]

2.5.2 Diagrama de componentes.

Un diagrama de componentes muestra las organizaciones y dependencias lógicas entre componentes de software donde este es una parte física del sistema. Los diagramas de componentes describen los elementos físicos del sistema y sus relaciones. Se puede decir que un componente es la materialización de una o más clases, porque una abstracción con atributos y métodos pueden ser implementados en los componentes. Los elementos de modelado dentro de un diagrama de componentes serán componentesy paquetes. En la **¡Error! No se encuentra el origen de la referencia.** se muestra del

diagrama de componentespara la solución propuesta.[40]

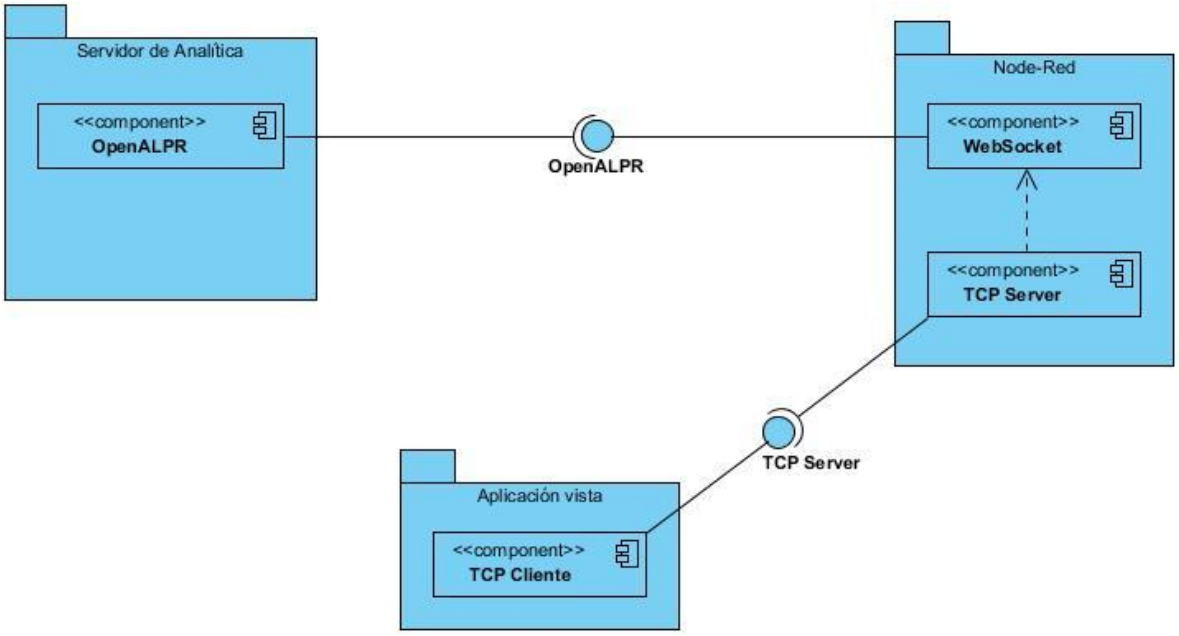


Ilustración 6 Diagrama de componentes.

Descripción de cada uno de los componentes que integran el diagrama:

Pluing OpenALPR: Es el encargado de reconocer las matrículas de un flujo de video.
WebSocket: Se encarga de recibir las matrículas reconocidas en la capa de Integración.
TCP Server: Se encarga de enviar las matrículas reconocidas desde la capa de Integración hacia el visual.
TCP Cliente: Es la clase que consume servicios de la capa de integración, recibiendo las matrículas con acceso concedido.

2.6 Diseño del Sistema.

En el diseño del sistema de softwarecrea una representación o modelo del software, que proporciona detalles acerca de las estructuras de datos, las arquitecturas, las interfaces y los componentes del software que son necesarios para implementar el sistema.[41]

2.6.1 Patrones de diseño.

Un patrón de diseño es una descripción de un problema y su solución que recibe un nombre y se puede emplear en otros contextos. En teoría indica la manera de utilizarlo en diversas circunstancias.Para realizar un diseño más eficiente es conveniente utilizar uno o varios patrones de diseño. Los patrones de diseño describen un problema que ocurre repetidas veces en algún contexto determinado de desarrollo de software, y entregan una buena solución ya probada. Esto ayuda a diseñar correctamente en menos tiempo, a construir problemas reutilizables y extensibles, facilita la documentación, y facilita la comunicación entre los miembros del equipo de desarrollo. Dentro de estos patrones de diseño se encuentran los **patrones Gang of Four** (**GOF** por sus siglas en inglés) y los **patrones de asignación de responsabilidades a objetos** (General Responsibility Assignment Software Patterns o **GRASP** por sus siglas en inglés)[42]

Se utilizó los patronesGOF que son una descripción de clases y objetos que se comunican entre sí, adaptada para resolver un problema general de diseño en un contexto particular. A continuación, se describenlos patrones GOF utilizados:[43]

Los **creacionales**que proporcionan mecanismos de creación de objetos que incrementan la flexibilidad y la reutilización de código existente como:

Singleton: también llamado Instancia únicaes un patrón de diseño creacional que permite asegurar que una clase tenga una única instancia, a la vez que proporciona un punto de acceso global a dicha instancia. Ejemplo: la clase (BaseDatos) ya que cada conexión de un cliente solo debe tener una instancia única.

Factory Method: también llamado método fábrica oconstructor virtual es un patrón de diseño creacional que proporciona una interfaz para crear objetos en una superclase, mientras permite a las subclases alterar el tipo de objetos que se crearán. Ejemplo: en la clase (Daxsess).

Los **patrones estructurales** explican cómo ensamblar objetos y clases en estructuras más grandes a la vez que se mantiene la flexibilidad y eficiencia de la estructura como puede ser:

Adapter: también llamadoadaptador, envoltorio o wrapper, este es un patrón de diseño estructural que permite la colaboración entre objetos con interfaces incompatibles.Ejemplo: en la clase (Vehiculo).

Los **patrones de comportamiento** se encargan de una comunicación efectiva y la asignación de responsabilidades entre objetos.

Mediator: también llamado mediador,intermediary o controller, este es un patrón de diseño de comportamiento que permite reducir las dependencias caóticas entre objetos. También restringe las comunicaciones directas entre los objetos, forzándolos a colaborar únicamente a través de un objeto mediador. Ejemplo: en la clase (Daxsess).

Durante el desarrollo de la solución también fueron utilizados los patronesGRASP. A continuación, se describen estos patrones utilizados:

Alta Cohesión: Concibe clases con responsabilidades moderadas en un área funcional que colaboran con otras clases para llevar a cabo las tareas. Constituye una premisa del Sistema Integral de Control de Acceso Daxcessdonde cada componente realiza una labor única dentro del sistema y auto-identificable creando un sistema relativamente fácil de mantener, entender y reutilizar.

Bajo acoplamiento: Soporta el diseño de clases que son más independientes, lo que reduce el impacto del cambio. Constituye otra de las premisas de Sistema Integral de Control de Acceso Daxcess garantizando que cada componente no dependa unos de otros a no ser estrictamente necesario.

Creador: la asignación de responsabilidades relacionadas con la creación de objetos. Se aplica al crear nuevos objetos que en conjunto logren formar la representación del módulo en el componente OpenALPR al crear instancias que permitan la conexión del módulocon el componente de Base de datos PostgresSQL como lo es PostgresDBStore.

Experto: Indica que la responsabilidad de la implementación de un método debe ser en la clase que contiene la información necesaria para cumplir con esta responsabilidad. Se aplica en OpenALPR que tiene la información necesaria para el manejo de

reconocimiento de matrículas de vehículos que le permite dar solución a los distintos requerimientos funcionales.

2.6.2 Diagramas de clases del diseño.

Los diagramas de clases de diseño representan una abstracción de una o varias clases y la forma en que se relacionan entre sí en la implementación del sistema, donde estas definen los objetos, con los cuales se implementan los casos de uso. Las particularidades del lenguaje de programación influyen en el diseño de las clases (ver **¡Error! No se encuentra el origen de la referencia.**).[44].

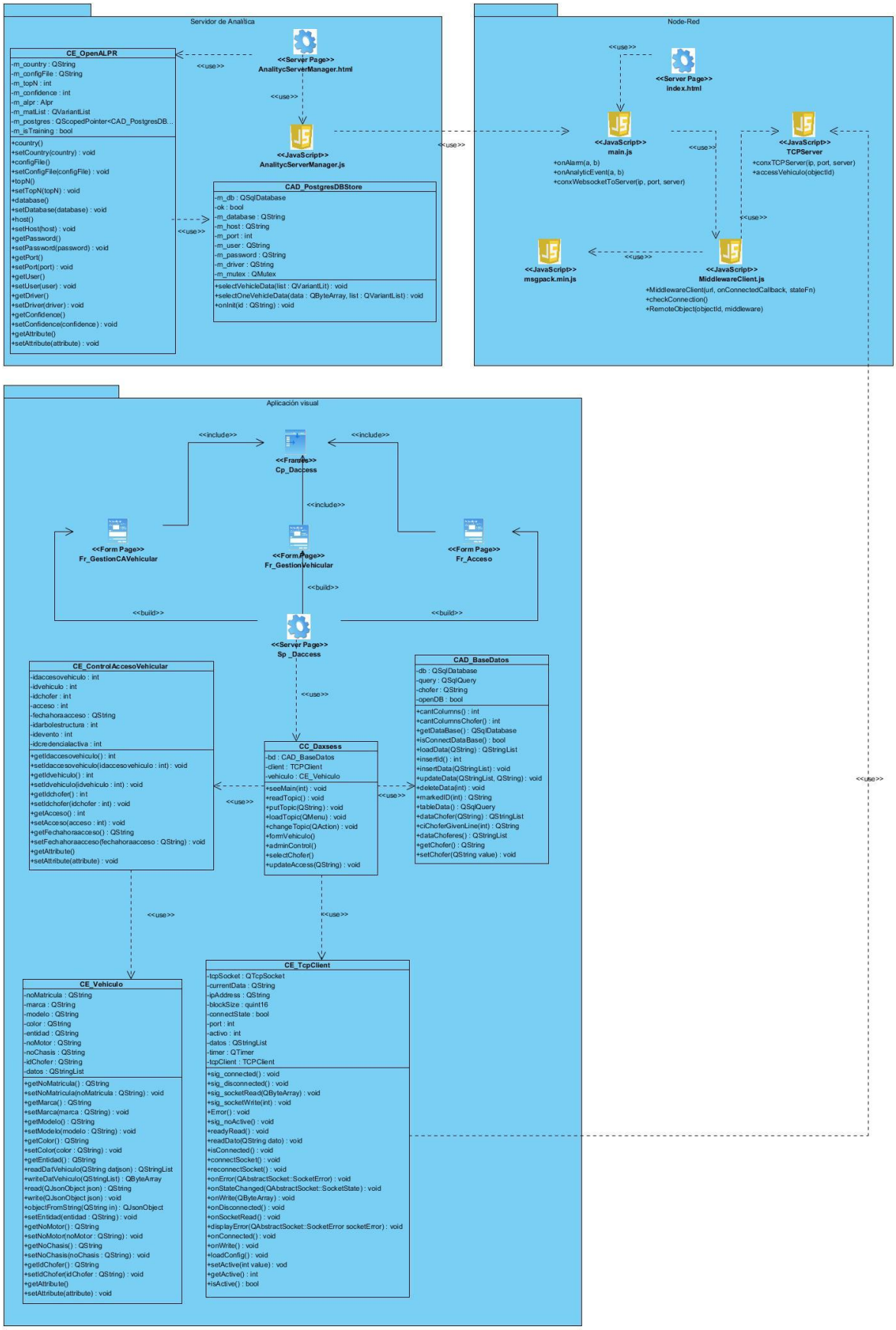

Servidor de Analítica
CE_OpenALPR
-m_country : QString
-m_configFile : QString
-m_topN : int
-m_confidence : int
-m_alpr : Alpr
-m_matList : QVariantList
-m_postgres : QScopedPointer<CAD_PostgresDB...
-m_isTraining : bool
+country()
+setCountry(country) : void
+configFile()
+setConfigFile(configFile) : void
+topN()
+setTopN(topN) : void
+database()
+setDatabase(database) : void
+host()
+setHost(host) : void
+getPassword()
+setPassword(password) : void
+getPort()
+setPort(port) : void
+getUser()
+setUser(user) : void
+getDriver()
+setDriver(driver) : void
+getConfidence()
+setConfidence(confidence) : void
+getAttribute()
+setAttribute(attribute) : void
<<use>>
<<Server Page>>
AnalitycServerManager.html
<<use>>
<<JavaScript>>
AnalitycServerManager.js
<<use>>
CAD_PostgresDBStore
-m_db : QSqlDatabase
-ok : bool
-m_database : QString
-m_host : QString
-m_port : int
-m_user : QString
-m_password : QString
-m_driver : QString
-m_mutex : QMutex
+selectVehicleData(list : QVariantLit) : void
+selectOneVehicleData(data : QByteArray, list : QVariantList) : void
+onInit(id : QString) : void
<<use>>
Node-Red
<<use>>
<<Server Page>>
index.html
<<JavaScript>>
main.js
+onAlarm(a, b)
+onAnalyticEvent(a, b)
+conxWebsocketToServer(ip, port, server)
<<use>>
<<JavaScript>>
TCPServer
+conxTCPServer(ip, port, server)
+accessVehiculo(objectId)
<<use>>
<<JavaScript>>
msgpack.min.js
<<use>>
<<JavaScript>>
MiddlewareClient.js
+MiddlewareClient(url, onConnectedCallback, stateFn)
+checkConnection()
+RemoteObject(objectId, middleware)
Aplicación visual
<<include>>
<<Frames>>
Cp_Daccess
<<include>>
<<include>>
<<Form Page>>
Fr_GestionCAVehicular
<<Form Page>>
Fr_GestionVehicular
<<Form Page>>
Fr_Acceso
<<build>>
<<build>>
<<build>>
<<Server Page>>
Sp _Daccess
<<use>>
CE_ControlAccesoVehicular
-idaccesovehiculo : int
-idvehiculo : int
-idchofer : int
-acceso : int
-fechahoraacceso : QString
-idarbolestructura : int
-idevento : int
-idcredencialactiva : int
+getIdaccesovehiculo() : int
+setIdaccesovehiculo(idaccesovehiculo : int) : void
+getIdvehiculo() : int
+setIdvehiculo(idvehiculo : int) : void
+getIdchofer() : int
+setIdchofer(idchofer : int) : void
+getAcceso() : int
+setAcceso(acceso : int) : void
+getFechahoraacceso() : QString
+setFechahoraacceso(fechahoraacceso : QString) : void
+getAttribute()
+setAttribute(attribute) : void
<<use>>
CC_Daxsess
-bd : CAD_BaseDatos
-client : TCPClient
-vehiculo : CE_Vehiculo
+seeMain(int) : void
+readTopic() : void
+putTopic(QString) : void
+loadTopic(QMenu) : void
+changeTopic(QAction) : void
+formVehiculo()
+adminControl()
+selectChofer()
+updateAccess(QString) : void
<<use>>
CAD_BaseDatos
-db : QSqlDatabase
-query : QSqlQuery
-chofer : QString
-openDB : bool
+cantColumns() : int
+cantColumnsChofer() : int
+getDataBase() : QSqlDatabase
+isConnectDataBase() : bool
+loadData(QString) : QStringList
+insertId() : int
+insertData(QStringList) : void
+updateData(QStringList, QString) : void
+deleteData(int) : void
+markedID(int) : QString
+tableData() : QSqlQuery
+dataChofer(QString) : QStringList
+ciChoferGivenLine(int) : QString
+dataChoferes() : QStringList
+getChofer() : QString
+setChofer(QString value) : void
<<use>>
<<use>>
<<use>>
CE_Vehiculo
-noMatricula : QString
-marca : QString
-modelo : QString
-color : QString
-entidad : QString
-noMotor : QString
-noChasis : QString
-idChofer : QString
-datos : QStringList
+getNoMatricula() : QString
+setNoMatricula(noMatricula : QString) : void
+getMarca() : QString
+setMarca(marca : QString) : void
+getModelo() : QString
+setModelo(modelo : QString) : void
+getColor() : QString
+setColor(color : QString) : void
+getEntidad() : QString
+readDatVehiculo(QString datjson) : QStringList
+writeDatVehiculo(QStringList) : QByteArray
+read(QJsonObject json) : QString
+write(QJsonObject json) : void
+objectFromString(QString in) : QJsonObject
+setEntidad(entidad : QString) : void
+getNoMotor() : QString
+setNoMotor(noMotor : QString) : void
+getNoChasis() : QString
+setNoChasis(noChasis : QString) : void
+getIdChofer() : QString
+setIdChofer(idChofer : QString) : void
+getAttribute()
+setAttribute(attribute) : void
CE_TcpClient
-tcpSocket : QTcpSocket
-currentData : QString
-ipAddress : QString
-blockSize : quint16
-connectState : bool
-port : int
-activo : int
-datos : QStringList
-timer : QTimer
-tcpClient : TCPClient
+sig_connected() : void
+sig_disconnected() : void
+sig_socketRead(QByteArray) : void
+sig_socketWrite(int) : void
+Error() : void
+sig_noActive() : void
+readyRead() : void
+readData(QString dato) : void
+isConnected() : void
+connectSocket() : void
+reconnectSocket() : void
+onError(QAbstractSocket::SocketError) : void
+onStateChanged(QAbstractSocket::SocketState) : void
+onWrite(QByteArray) : void
+onDisconnected() : void
+onSocketRead() : void
+displayError(QAbstractSocket::SocketError socketError) : void
+onConnected() : void
+onWrite() : void
+loadConfig() : void
+setActive(int value) : vod
+getActive() : int
+isActive() : bool

Ilustración 7 Diagrama de clases del diseño.

2.7Diagrama Entidad Relación.

En el diagrama Entidad Relación se precisa las clases de datos que apoyan las funciones del negocio definidas en el modelo de negocios. Tiene como objetivo definir el comportamiento que debe ser implementado en la base de datos y asegurarse de que los datos persistentes son almacenados consistente y eficientemente.[45]

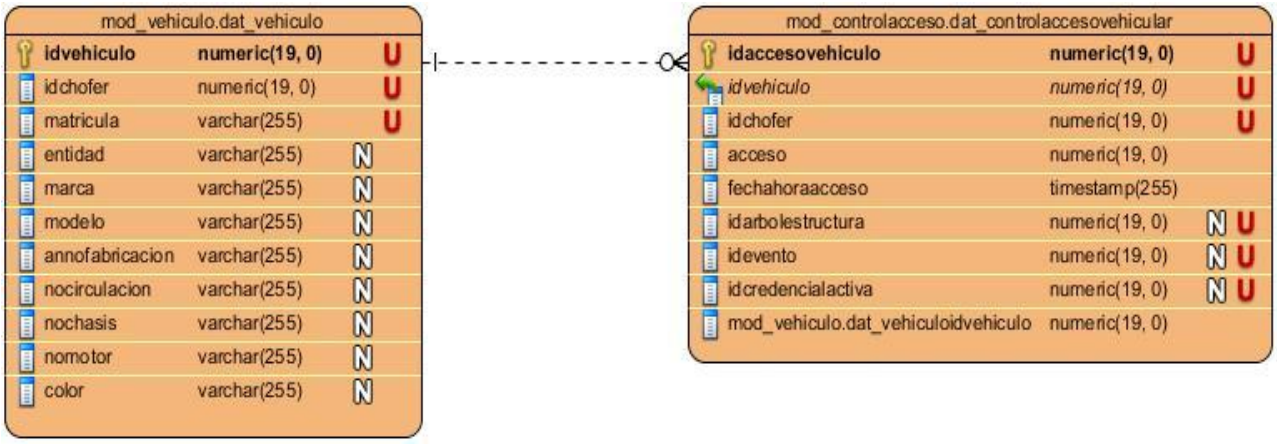


Ilustración 8. Diagrama Entidad Relación.

En la Ilustración 8 se muestran los datos persistentes que son almacenados. La tabla (mod_vehiculo.dat_vehiculo) contiene los datos de los vehículos almacenados y en la tabla (mod_controlacceso.dat_controlaccesovehicular) se muestran los registros realizados por estos vehículos autorizados. Existe entre ambos una relación de un vehículo a varios registros de controles de acceso. En diagrama Entidad Relacion solo se muestran las tablas de base de datos del Sistema Integral de Control de Acceso Daxcess necesarias para llevar el registro y control de acceso vehicular en el sistema.

2.8Conclusiones.

Al finalizar este capítulo se lograron especificar las características principales del módulo de control de acceso vehicular y una mejor explicación de su proceso para lograr una mejor implementación del software. El diseño del dominio ayudó a entender la estructura

del software y las técnicas de obtención de requisitos permitieron que el software tenga funcionalidades parecidas a la de los sistemas similares.Seleccionar la arquitectura basada en Plug-in facilitará la integración del módulo al sistema. También se vieron los patrones de diseño utilizados, aportando una mejor estructura al código en la implementación.El diagrama de entidad relación mejora la gestión de las clases en el acceso de datos utilizadapara el desarrollo del módulo informático para el Sistema Integral de Control de Acceso Daxcess.

Capítulo 3. Construcción y validación del sistema.

3.1 Introducción.

En el presente capítulo se especifican las principales características del algoritmo de inteligencia artificial OpenALPR. Se expondrá el diagrama de despliegue y las pruebas realizadas al software. En la descripción del algoritmo se mencionará el lenguaje de programación escrito, como funciona este y sus principales usos. Además, el diagrama de despliegue mostrará cómo estará conformado para el momento de su despliegue dentro del Sistema Integral de Control de Acceso, Daxcess.

3.2 Algoritmo de Inteligencia Artificial OpenALPR.

El plug-in de OpenALPR es una biblioteca de reconocimiento automático de matrículas de código abierto escrita en C ++ con enlaces en C #, Java, Node.js y Python. Esta biblioteca analiza imágenes y secuencias de video para identificar matrículas. La salida es la representación de texto de los caracteres de la matrícula. El software puede utilizarse de muchas formas diferentes ya que puede reconocer las matrículas de las transmisiones de la cámara. Los resultados son navegables y pueden activar alertas. El repositorio de datos puede estar en la nube o almacenarse por completo dentro de su red en el sitio. Otra de sus aplicaciones es que puede reconocer las matrículas de las transmisiones de la cámara y enviarle los resultados a su propia aplicación, procesando un archivo de video y almacenando los resultados de la matrícula en una base de datos CSV y SQLite. También puede analizar imágenes fijas desde la línea de comandos. Además, se integra el reconocimiento de matrículas en su aplicación directamente en el código (C / C ++, C #, VB.NET, Java, Python, Node.js).[46]

La Interfaz de Programa de Aplicación (API) de OpenALPR es una forma para que los programadores hagan uso del motor de OpenALPR en su propio software.La API OpenALPR reconoce las placas de matrícula dadas imágenes individuales.

La instancia de la biblioteca debe inicializarse una vez para cada subproceso. Hay un tiempo de inicialización significativo (unos segundos), por lo que no debe destruir la instancia hasta que haya terminado de usarla. Esta instancia no es segura para subprocesos por diseño. Una instancia ALPR debe usarse en un solo hilo y no compartirse entre hilos. OpenALPR opera como una tubería. La entrada es una imagen, varios procesamientos ocurren en etapas y la salida son los posibles números de placa en la imagen.

La fase de **canalización** ocurren en el siguiente orden: [47]

1. Detección (o Detection): la clase (regiondetector.cpp)encuentra posibles regiones de matrículas.
2. Binarización (o Binarization): la clase (binarizewolf.cpp) convierte la imagen de la región de la placa en blanco y negro.
3. Análisis de Char (o Char Analysis): la clase(characteranalysis.cpp) encuentra "manchas" del tamaño de un carácter en la región de la placa.
4. Bordes de placa (o Plate Edges): las clases (platelines.cpp) y (platecorners.cpp) encuentran los bordes y forma de la matrícula.
5. Enderezar (o Deskew):la clase (licenseplatecandidate.cpp) transforma la perspectiva en una vista directa basada en el tamaño de matrícula ideal.
6. Segmentación decarácter (o Character Segmentation:la clase (charactersegmenter.cpp)aísla y limpia los personajes para que puedan procesarse individualmente.
7. OCR: la clase (ocr.cpp)analiza la imagen de cada personaje, proporciona múltiples letras y confidencias posibles.
8. Postprocesamiento (Post Processing):la clase (postprocess.cpp)crea una lista top n de posibilidades de placa basada en confidencias del Reconocimiento Óptico de Caracteres (OCR). También realiza una coincidencia de expresiones regulares con las plantillas de región si se solicita.

La fase de **detección**: ocurre una vez para cada imagen de entrada.Utiliza el algoritmo LBP (generalmente utilizado para la detección de rostros) para encontrar posibles regiones de matrículas (x, y, ancho, alto).Cada una de estas regiones se envía a las últimas fases de la canalización para su posterior procesamiento.La fase de detección

suele ser la fase de procesamiento más intensivo.Puede acelerarse por GPU para mejorar el rendimiento.

La fase de **binarización**ocurrirá varias veces y todas las fases posteriores, una para cada posible región de matrícula. Esta creará múltiples imágenes binarias para cada región de placa. Se utilizarán varias imágenes binarias para proporcionar la mejor posibilidad de encontrar todos los personajes. Por ejemplo, una sola imagen binarizada puede perder caracteres si la imagen es demasiado oscura o demasiado clara. La binarización utiliza el método Wolf-Jolien, así como el método Sauovola con varios parámetros. Cada imagen binaria se procesará en fases posteriores.

La fase de **análisis de char**analiza decaracteres intentando encontrar regiones del tamaño de un carácter en la región de la placa. Lo hará encontrando primero todas las manchas conectadas en la región de la placa de matrícula. Luego, buscará manchas aproximadamente del ancho y alto de un caracter de placa de matrícula y con partes superiores e inferiores que estén en línea recta con otras manchas de ancho y alto similar. Este se realizará varias veces en la región.Primero se detectarán los caracteres pequeños, luego los caracteresmás grandes serán cazados gradualmente.Si no se encuentra nada en la región, se desechará y no seguirá ningún procesamiento adicional.Si se encuentran caracteres potenciales, la región de caracteres se guardará y se llevará a cabo un procesamiento adicional.

En la fase**bordes de placa** se buscarán los bordes de la matrícula. Tenga en cuenta que la fase de detección solo se encargará de identificar una posible región en la que pueda existir una matrícula. A menudo, se detectará una región ligeramente más grande o más pequeña que la placa real, pero se buscarán los bordes precisos superior e inferior y derecho o izquierdo de la placa de la matrícula.El primer paso es encontrar todas las líneas necesarias para la región de la matrícula.En platelines.cpp, se procesará la imagen de la placa y se calculará una lista de líneas horizontales y verticales.Platecorners usará esta lista, así como la altura del carácter (calculada en Análisis de caracteres), para encontrar los bordes de la línea de placa más probables.Utilizará varios pesos configurables para determinar qué borde es más lógico. Intentará usar un borde predeterminado (basado en el ancho y alto ideal de la placa) para ver si eso hace una buena combinación.

Lafase de **enderezar** volverá a mapear la región de la placa a un tamaño y orientación estándar. Idealmente, esto producirá una imagen de placa correctamente orientada sin rotación ni inclinación.

La fase de **segmentación de carácter**intentará aislar a todos los personajes que componen la imagen del plato.Utilizará un histograma vertical para encontrar espacios en los caracteres de la placa. Esta fase también limpiará las casillas de los personajes eliminando pequeñas manchas desconectadas y descalificando las regiones de los personajes que son demasiado cortas. También intentará eliminar las regiones de "borde" para que el borde de la placa no se clasifique de forma inapropiada como "1" o "l".

La fase de **OCR** analiza cada carácter de forma independiente. Para cada imagencalculará todos los caracteres posibles y sus confidencias.

La fase de **Postprocesamiento**Dada una lista de todos los caracteres y confidencias de OCR posibles, el procesamiento posterior determinará las mejores combinaciones posibles de letras de placa. Se organizará como una lista de "n" principales. El procesamiento posterior descalificará a todos los personajes por debajo de un umbral en particular. También tendrá un umbral "suave"; los caracteres por debajo de este umbral aún se agregarán a la lista posible. Sin embargo, se agregará un posible carácter en blanco porque el carácter de baja confianza puede no ser realmente parte del plato.El procesamiento posterior también manejará la validación de la región, si se solicita.

El uso de la detección de movimiento aumenta enormemente la eficiencia del agente OpenALPR. En lugar de monitorear cada píxel de cada fotograma de un video, el software ignora las áreas del video que no han cambiado (y por lo tanto no pueden contener una matrícula). Cuando se detecta movimiento, solo se analizará la parte en la que se encuentra el vehículo. Para proporcionar la mayor cantidad de lecturas posibles, OpenALPR también utiliza un búfer de imagen configurable. Cuando se detecta una gran cantidad de movimiento, los fotogramas de vídeo se colocan en este búfer y se procesan. Por lo tanto, si el video contiene momentos de inactividad, los recursos de la CPU procesarán los datos de video más antiguos para proporcionar la mayor cantidad posible de lecturas de matrículas.

En una transmisión de video, una sola placa se ve muchas veces cuando pasa por la cámara. OpenALPR puede reconocer esa misma placa 60 veces. La función de agrupación de matrículas rastrea la matrícula a medida que se mueve, entregando un único resultado para el número de matrícula que se puntúa en función del número de reconocimientos. Por lo tanto, el procesamiento de alta velocidad produce un número de matrícula muy preciso.[47]

3.3 Diagrama de despliegue.

El diagrama de despliegue establece una correspondencia entre la arquitectura software y la arquitectura hardware del sistema a entregar. Este tiene como objetivo mostrar la disposición de las particiones físicas del sistema de información y la asignación de los componentes software a estas particiones. También muestra las conexiones físicas entre el hardware y las relaciones entre componentes y muestra el hardware usado y los componentes instalados en el hardware.[48]

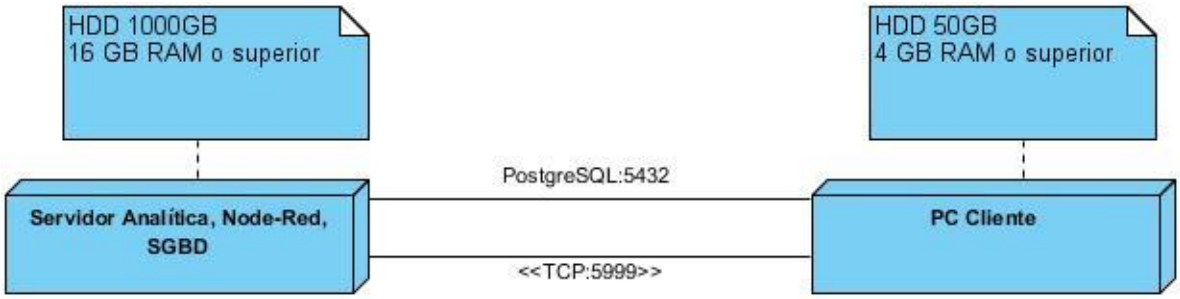


Ilustración 9 Diagrama de despliegue.

En la **¡Error! No se encuentra el origen de la referencia.** se muestra como quedara desplegado el sistema una vez entregado.El servidor de Analítica, el de Node-Red y el de SGBD o Sistema de Gestor de Bases de Datos con PostgreSQL se encuentran en un mismo servidor. El Servidor de Analítica está montado en el sistema operativo Ubuntu 14.4, este una vez ya reconocida la matrícula comprueba si esta se encuentra en el servidor SGBD. Si el acceso es concedido lo envía a la capa de integraciónNode-Red. Luego de recibir los datos Node-Redlos envía a la aplicación encargadade la vista que se encuentra en la PC Cliente el cual se puede encontrar montado en los sistemas operativos Linux y Windows. La aplicación del componente de lavistaque se comunica con la base de datos para gestionar los vehículos y el control de acceso vehicular.

3.4 Estrategia de prueba.

Las pruebas del software son un elemento crítico para la garantía de calidad del software y representa una visión final de las especificaciones, del diseño y de la codificación. Una vez generado el código fuente, el software debe ser probado para descubrir y corregir el máximo de errores posibles antes de su entrega al cliente. Se establecen varias normas como[49]:

1 La prueba es el proceso de ejecución de un programa con la intención de descubrir un error.
2 Un buen caso de prueba es aquel que tiene una alta probabilidad de mostrar un error nodescubierto hasta entonces.
3 Una prueba tiene éxito si descubre un error no detectado hasta entonces.

Los datos que se van recogiendo a medida que se realiza la prueba, proporcionando una buena indicación de la fiabilidad del software y de alguna manera, indica la calidad del software como un todo[41].

Para realizarlas pruebas al módulo OpenAPLR del Servidor de Analítica, con el objetivo de descubrir y corregir errores existentes en este, fueron empleadas las pruebas de caja blanca y caja negra.

3.4.1 Caja Blanca.

La prueba de caja blanca es un método de diseño de casos de prueba que utiliza la estructura de control del diseño procedimental para obtener los casos de prueba[41].
La técnica de prueba blancaque se aplica a la solución desarrollada será la Prueba del Camino Básico, que permite obtener una medida de la complejidad lógica del diseño procedimental y usar esa medida como guía para la definición de un conjunto básico de caminos de ejecución, garantizando con estos que durante la prueba se ejecute por lo menos una vez cada sentencia del programa[41].
En la Ilustración 10 se muestra el fragmento de código correspondiente a la funcionalidad “insertVehiculo(QStringList datos)”la cual fue seleccionada para aplicar la prueba.

```
void Daxcess::insertVehiculo(QStringList datos)
{
    if(datos.length()!=0){//1
    bd->insertData(datos);//2
    }//3
}//4
```

Ilustración 10. Código del método insertVehiculo.

Ilustración 11. Grafo de flujo asociado al método**insertVehiculo**.

Se identificaron 4 bloques de ejecución atendiendo a las dependencias procedimentales las cuales construyen el grafo de flujo del método representado en la Ilustración 11. Teniendo esta información se procede al cálculo de la complejidad ciclomática.

La complejidad ciclomática (V (G)) es una métrica de software extremadamente útil pues proporciona una medición cuantitativa de la complejidad lógica de un programa. Donde el valor calculado define el número de caminos independientes del conjunto básico de un programa y da un límite superior para el número de pruebas que se deben realizar para asegurar que se ejecute cada sentencia al menos una vez. Esta operación se puede calcular de tres formas:[32]

1. V (G) = A - N + 2, donde A es la cantidad total de aristas y N la cantidad total de nodos.
2. V (G) = P + 1, donde P es la cantidad total de nodos predicados (nodos de los cuales parten dos o más aristas).
3. V (G) = R, donde R es el número de regiones del grafo de flujo.

1) A=4

N=4

V(G)=0+2

V(G)=2

2) P=1

V(G)=1+1

V(G)=2

3) R=2

V(G)=2

Luego de aplicar las tres fórmulas para obtener un resultado seguro se obtiene una complejidad ciclomática V (G) = 2. Por lo tanto, se obtendrán 2 caminos de pruebas, que se muestran a continuación:

Camino básico #1: 1-2-3-4.

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro QStringList datos, no está vacío, contiene todos los datos de un vehículo

Entrada:QStringList datos, = [idvehiculo] => 1, [idchofer] =>2,[matricula] => B075293, [entidad] =>UCI,[marca] => BMW, [modelo] => Serie 5 G30 sedan, [annofabricacion] => 2006,[nocirculacion] => C398267,[nochasis] => LLCJPJ4C8HA101957, [nomotor] =>B10S2511020KC2, [color]=>blanco.

Resultados esperados: Se espera que sea adicionado un nuevovehículo en el sistema.

Camino básico # 2: 1-3-4.

Descripción: El dato de entrada cumplirá con el siguiente requisito:

El parámetro QStringList datos, está vacío, no contiene todos los datos de un vehículo

Entrada: QStringList datos, = [idvehiculo] =>null, [idchofer] => null, [matricula] => null, [entidad] => null, [marca] => null, [modelo] => 1, [annofabricacion] => null, [nocirculacion] => null, [nochasis] => null, [nomotor] => null, [color]=> null.

Resultados esperados: Se espera que se alerte al usuario que el vehículo no ha sido adicionado por tener los campos vacíos.

3.4.2 Caja Negra.

Las pruebas de caja negrase centran en los requisitos funcionales del software. Esta permite al ingeniero del software obtenerconjuntos de condiciones de entrada que ejerciten completamente todos los requisitos funcionales de un programa[41].
Caja Negra es un método, del nivel de prueba Pruebas Funcionales que se utilizapara asegurar la calidad de la aplicación desarrollada utilizando la técnica de:

Partición equivalente:
La partición equivalente es una técnica del método de prueba de caja negra que divide el campo de entrada de un programa en clases de datos de los que se pueden derivar casos de prueba. Un caso de prueba ideal descubre de forma inmediata una clase de error, quede otro modo, requerirían la ejecución de muchos casos antes de detectar el error genérico[41].

Estas se centran en lo que se espera de una funcionalidad, es decir, intentan encontrar casos en que la funcionalidad no se atiene a su especificación, esta prueba se limita a brindar solo datos como entrada y estudiar la salida, sin preocuparse de lo que pueda estar haciendo la funcionalidad internamente, es decir, solo trabaja sobre su interfaz externa.

Para cada uno de los requisitos funcionales fueron definidos casos de prueba, de ellosse muestra el siguiente: caso de prueba para el requisito Adicionar Vehículo.
Condiciones de ejecución:

1. Se debe identificar y autenticar ante el sistema, además debe tener los permisos para ejecutar esta acción.
2. Se debe seleccionar la opción del menú: Gestión Vehicular/Adicionar vehículo.

Tabla 5. Caso de prueba para el requisito Adicionar Vehículo.

Nombre del Requisito	Descripción General	Escenarios de prueba	Flujo del escenario
1: Adicionar vehículo	El sistema debe permitir adicionar vehículos	EP 1.1: Adicionar vehículo introduciendo datos	➢ Se introducen los datos del vehículo

		validos	correctamente. ➢ Se presiona el botón Aceptar. ➢ Se muestra un mensaje de información. ➢ Se presiona el botón Aceptar
		EP 1.2: Adicionar vehículo introduciendo datos válidos presionando el botón Aplicar.	➢ Se introducen los datos del vehículo correctamente. ➢ Se presiona el botón ➢ Aplicar. ➢ Se muestra un mensaje de información. ➢ Se presiona el botón ➢ Aceptar.
		EP 1.3: Adicionar vehículo introduciendo datos inválidos	➢ Se introducen los datos inválidos del recurso humano. ➢ Se presiona el botón ➢ Aceptar. ➢ Se muestra un mensaje de error.
		EP 1.4: Adicionar	➢ Se introducen

		vehículo dejando campos vacíos.	los datos dejando algún campo en blanco. ➢ Se presiona el botón ➢ Aceptar. ➢ Se muestra un mensaje informando del error. ➢ Se presiona el botón ➢ Aceptar.
		EP 1.5: Cancelar	➢ Se introducen o no los datos del recurso humano. ➢ Se presiona el botón Cancelar.

La aplicación desarrollada fue probada y tuvo un correcto funcionamiento a partir de las pruebas realizadas.Con la aplicación de las Pruebas de Caja Negra se demostró que cada una de las funciones de la aplicación es totalmente operativa, se obtuvieron los resultados esperados como respuesta del conjunto de condiciones de entrada, comprobándose el correcto funcionamiento de las funcionalidades del módulo de control de acceso vehicular según su especificación.

3.4.3 Pruebas de Aceptación.

En las pruebas de aceptación serán probadas las funcionalidades exigidas por el cliente, que se han implementado. Las pruebas de aceptación se realizan con el cliente o usuarios finales para validar que el software hace lo que fue pactado con el cliente en los requisitos.Las pruebas de aceptación se llevarán a cabo redactando los casos de prueba,

teniendo en cuenta el orden y la prioridad en que han sido asignadas las funcionalidades.Las pruebas de aceptación correspondiente a cada una de las funcionalidades serán representadas mediante tablas divididas por las secciones siguientes:

- Código de la prueba de aceptación.
- Número de la historia de usuario a la que se le realiza la prueba.
- Nombre de la funcionalidad.
- Descripción de la funcionalidad.
- Condiciones de ejecución de la funcionalidad.
- Entrada y pasos de ejecución que realiza el usuario con el objetivo de obtener el resultado
- esperado.
- Resultado esperado.
- Evaluación de la prueba.

A continuación, se muestra el caso de prueba de aceptación correspondiente a la RF1AdicionarVehículo se probará que los datos del vehículo que se introduzcaquede registrado (ver Tabla 6).

Caso de Prueba de Aceptación	
Código: CP_RF1	RF1:AdicionarVehículo
Responsable: Rafael Rubén Jorge González	
Descripción: Probar que los datos del vehículo que se introduzcaquede registrado.	
Condiciones de ejecución: Se debe identificar y autenticar ante el sistema, además debe tener los permisos para ejecutar esta acción.	
Entrada y Pasos de ejecución: 1. Pulsar “Adicionar vehículo” en Gestión Vehicular de la barra de navegación. 2. Introducir los datos del vehículo. 3. Ejecutar la acción “Aceptar”.	
Resultado esperado: Que el sistema envíe el resultado de que se añadiócorrectamente el vehículo al sistema.	

Evaluación de la prueba: Satisfactoria.

Tabla 6 Caso de Prueba de Aceptación "Adicionar Vehículo".

En el caso de prueba de aceptación correspondiente a la RF 2ModificarVehículo se probará que los datos del vehículo que se modifiquenquede registrado (ver Tabla 7).

Caso de Prueba de Aceptación	
Código: CP_RF2	RF2:AdicionarVehículo
Responsable: Rafael Rubén Jorge González	
Descripción: Probar que los datos del vehículo que se modifiquequede registrado.	
Condiciones de ejecución: Se debe identificar y autenticar ante el sistema, además debe tener los permisos para ejecutar esta acción.	
Entrada y Pasos de ejecución: 1. Pulsar “Modificarvehículo” en Gestión Vehicular de la barra de navegación. 2. Introducir los datos del vehículo. 3. Ejecutar la acción “Aceptar”.	
Resultado esperado: Que el sistema envíe el resultado de que se modificócorrectamente el vehículo del sistema.	
Evaluación de la prueba: Satisfactoria.	

Tabla 7 Caso de Prueba de Aceptación "Modificar Vehículo".

3.5Resultado de Pruebas.

Se aplicaron un total de 9 casos de pruebas para realizar un correcto funcionamiento del módulo en 3 iteraciones. En la primera iteración se detectaron 4 no conformidades que fueron corregidas totalmente.

En la segunda iteración se detectaron dos nuevas no conformidades igualmente corregidas En la tercera y última iteración no se detectaron errores. Estos resultados se muestran en la Ilustración 12a continuación en la gráfica siguiente:

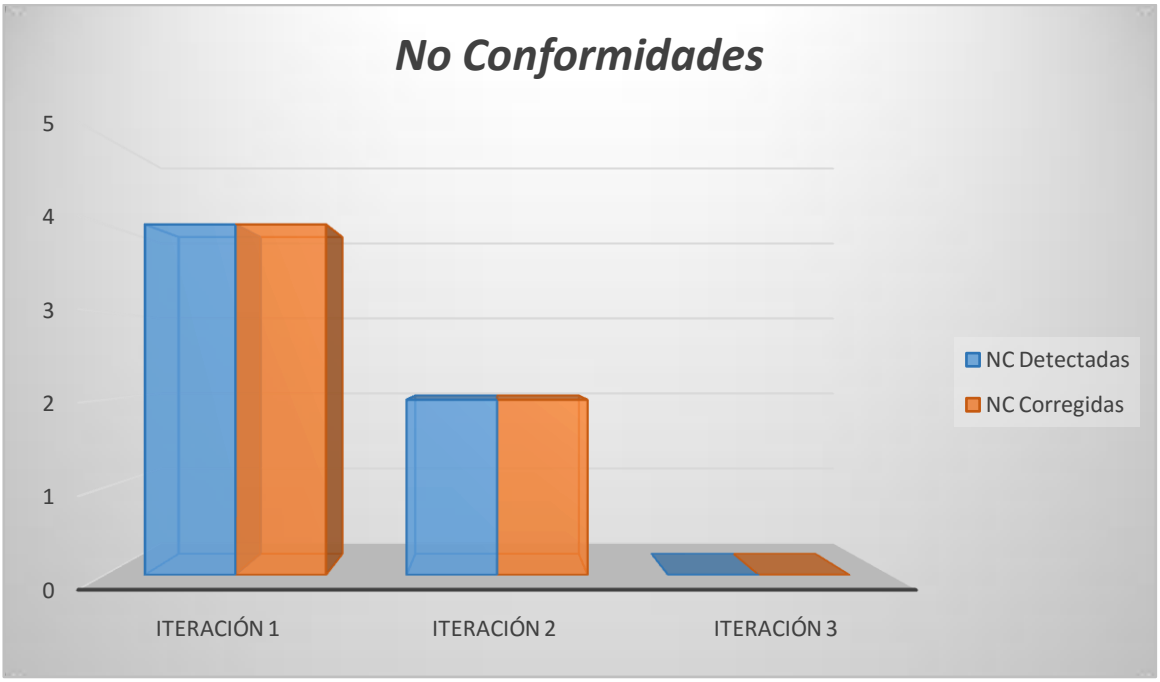


Ilustración 12. Relación de No Conformidades (NC) Detectadas y Corregidas por iteraciones.

3.6.Conclusiones.

En este capítulo se especificó las principales características del algoritmo de inteligencia artificial OpenALPR que permite su uso en la representación de texto de los caracteres de las matrículas vehiculares para una mejor gestión en el control de acceso vehicular. Se presentó un diagrama de despliegue que establece una correspondencia entre la arquitectura software y la arquitectura hardware para el sistema a entregar. En la Prueba de Caja Negra se demostró que cada una de las funciones de la aplicación es totalmente operativa y se obtuvieron los resultados esperados como respuesta según su especificación. En los resultados de las Pruebas de Aceptación se aplicaron 9 casos de pruebas y en 3 iteraciones se logró una correcta validación del código.

CONCLUSIONES

- En este trabajo se analizaron los principales conceptos teóricos y las soluciones existentes relacionadas con el control de acceso vehicular aportando una base teórica del software para un mayor conocimiento del tema a tratar e implementar un software de control de acceso vehicular que se integreal Sistema Integral de Control de Acceso Daxcess. En la realización de este proceso se tomó como guía la metodología de desarrollo de software Prodesoftque realiza la planificación del software a implementar y ayuda a que se realice de forma controlada y eficiente.
- La arquitectura basada en plug-in permite la integración armónica entre el servidor de analítica y el módulo software implementado OpenALPR, permitiendo que se reutilice el código y aportando un mejor diseño para la realización de losrequisitos funcionales.

- El algoritmo API OpenAlpr es un valioso aporte en la construcción del módulo ya que permite analizar imágenes y secuencias de video.

- La estrategia de prueba definida arrojó4 no conformidades del software que fueron corregidasen 3 iterando hasta que no se detectaron mas no conformidades.El método de caja blanca se realizó a través de la Prueba del Camino Básico que permitió obtener una medida de la complejidad lógica del diseño procedimental. Las pruebas de Caja Negra se aplicaron a través del método de Partición de Equivalencia aportando mejoras al software ya que a partir de ellas se eliminaron una serie de errores y de no conformidades antes de entregar el producto al cliente.

Bibliografía

[1] «SCIELO,» [En línea]. Available: http://scielo.sld.cu/scielo.php?script=sci_arttext&pid=S0258-59362011000100009.

[2] «SEGU-INFO,» [En línea]. Available: https://www.segu-info.com.ar/logica/identificacion.

[3] «DREAMIT,» 9 6 2020. [En línea]. Available: https://dreamit.cl/blog/stickers-tag-acceso-vehicular-sin-contacto/.

[4] «SISCA,» [En línea]. Available: http://sisca.co/que-es-un-control-de-acceso/.

[5] M. Martin R. Mondragón Sotelo, «Capítulo 4. Control de acceso,» de *Seguridad Informática*, "Seguridad Informática. Capítulo 4 Control de acceso", Martin R. Mondragón Sotelo, Mygnet, pp. 18-20.

[6] «SERACIS,» [En línea]. Available: https://www.seracis.com/apoyos-detalle/controles-de-acceso .

[7] «TANGOID,» [En línea]. Available: https://www.tangoid.com.ar/control-de-acceso.

[8] «VISERCO,» [En línea]. Available: https://www.viserco.com/control-de-acceso-que-es-y-su-importancia.

[9] «IDBUILD,» [En línea]. Available: https://www.idibuild.es/sistemas-de-seguridad-y-control-de-accesos-de-las-empresas/.

[10] DOINTECH. [En línea]. Available: http://www.dointech.com.co/control-acceso-vehicular.html.

[11] «TTCS,» [En línea]. Available: https://www.ttcs.es/20-ttcs-sl/reconocimiento-de-matriculas/10145-reconocimiento-de-matriculas.html.

[12] «REDESZONE,» [En línea]. Available: https://www.redeszone.net/tutoriales/seguridad/diferencias-autenticacion-autorizacion/.

[13] «OAS,» [En línea]. Available: http://www.oas.org/en/citel/infocitel/2006/junio/seguridad_e.asp.

[14] «DEFINICIÓN ABC,» [En línea]. Available: www.definicionabc.com/general/autorizacion.php.

[15] «DEFINICIÓN,» [En línea]. Available: https://definicion.de/autorizacion/.

[16] «MASSCOMM,» [En línea]. Available: https://www.masscomm.es/blog/control-de-accesos-y-control-de-presencia-todo-lo-que-debes-saberA.

[17] «TECNOSEGURO,» [En línea]. Available: https://tecnoseguro.com.

[18] «SYON,» [En línea]. Available: https://www.syon.es/productos/control-de-presencia-y-accesos/software/genius-accesos/.

[19] «I+D3,» [En línea]. Available: https://imasdetres.com/mx/software-reconocimiento-de-placas-lpr/.

[20] «DATYS,» [En línea]. Available: http://www.datys.cu.

[21] C. L. T. y. P. O. Penadés, Métodologías ágiles para el desarrollo de software., Penadés, Carmen, Letelier Torresy Patricio Orlanndo. 2006. Métodologías ágiles para el desarrollo de software. 2006. Vol. 5. ISSN-e 1666-1680., 2006.

[22] G. (. Méndez, «Proceso Software y Ciclo de vida.,» [En línea]. Available: www.fdi.ucm.es/profesor/gmendez/docs/is0809/02-ProcesoCicloDeVida.pdf.

[23] Q. -. P. |. Q. f. A. Development., «QTCreator,» [En línea]. Available: https://www.qt.io/product/development-tools.

[24] «NodeRed,» [En línea]. Available: https://nodered.org.

[25] R. Pressman, Modelos de Proceso., Modelos de Proceso. Parte 2. 2014, 2014.

[26] F. J. C. Sierra, Enciclopedia del lenguaje C++, Sierra, Francisco Javier Ceballos. 2012. Enciclopedia del lenguaje C++. s.l. : , 2012.

[27] D. Jordan, Implementation benefits of C++ language mechanisms., JORDAN, David. Implementation benefits of C++ language mechanisms. Communications of the ACM. 1990. Vol. 33, no. 9, p. 61–64., 1990.

[28] Brandendaugh, Aplicaciones JavaScript., Madrid,España : Brandendaugh. 2000. Aplicaciones JavaScript. Madrid,España : O'Reilly & , 2000..

[29] V. PARADIGM. [En línea]. Available: https://prezi.com/j84ywfydzvit/visual-paradigm/.

[30] Q. 5, Introducción al Framework Qt 5 para el desarrollo de aplicaciones multiplataforma., SOFTPEI. 2013. Introducción al Framework Qt 5 para el desarrollo de aplicaciones multiplataforma., 14 de Mayo de 2013..

[31] «PostgreSQL,» [En línea]. Available: https://www.postgresql.org.

[32] R. S. Pressman, Ingeniería de Software. Un enfoque práctico., Pressman, Roger S.

2002. Ingeniería de Software. Un enfoque práctico. s.l. : Mc Graw Hill, 2002. ISBN:8448132149., 2002.

[33] UCID, Proceso de Desarrollo y Gestión de Proyectos de Software. 2012., UCID, Unidad de Compatibilización Integración y Desarrollo De Software para la Defensa. 2012..

[34] Pressman, (2013), Pressman. 2013. Ingeniería de Requisitos Parte 1. 6ta Edición. 2013. ISBN:9789701054734..

[35] R. Tome, Captura de requisitos., Tome, Rtzaida. Captura de requisitos. 2011., 2011.

[36] Somerville, Software Engineering., http://eva.uci.cu/mod/resource/view.php?id=9269&subdir=/Sommerville_8va_edicion., 2007.

[37] (. Reynoso, Introducción a la Arquitectura de Software, Reynoso, C. B. (2005). Introducción a la Arquitectura de Software..

[38] (. Johannes, Ligthweight Plug-in Based Aplication Development., Johannes Mayer, I. M. (2006). Ligthweight Plug-in Based Aplication Development..

[39] I. C. V. Tamayo, «Arquitectura de Plug-in para Sistemas de Visualización Médica.,» *IDICT. CIGET-Matanzas.*

[40] Mancha, «Mancha, Universidad de Catilla La. 2011. UCLM.,» 2011. [En línea]. Available: http://www.dsi.uclm.es/asignaturas/42530/pdf/M2tema12.pdf..

[41] (. Pressman, Ingenieria de Software, Un enfoque práctico., Pressman, Roger S. 2005. Ingenieria de Software, Un enfoque práctico. 2005..

[42] C. Larman, UML y Patrones., Mexico : Larman, Craig. 1999. UML y Patrones. Mexico : s.n., 1999. 970-17-0261-1., 1999.

[43] «refactoring,» [En línea]. Available: https://refactoring.guru/es/design-patterns.

[44] (. Larman, Introducción al análisis y diseño orientado., Larman, Craig. 2003. UML y patrones. Introducción al análisis y diseño orientado..

[45] S. S. Rios, «Rios, Sergio Sánchez. 2011. slideshare. 28 de Abril de 2011.,» 28 4 2011.. [En línea]. Available: http://es.slideshare.net/SergioRios/unidad-2-modelo-de-datos..

[46] «OPENALPR,» [En línea]. Available: https://www.openalpr.com.

[47] «DOCOPENALPR,» [En línea]. Available: http://doc.openalpr.com.

[48] «Melchor, Alex. 2012. Prezi,» 22 11 2012. [En línea]. Available: https://prezi.com/e_gpb7xev_im/tutorial-diagramas-de-despliegue/.

[49] G. Meyers, The Art of Software Testing, [MYE79] Myers, G., The Art of Software Testing, Wiley, 1979, 1979.

[50] «ECURED,» [En línea]. Available: https://www.ecured.cu/Control_de_acceso.