Especificación de requisitos de software

Proyecto: Transformando el acceso vehicular

- By ing Ivan Sosa
- Jaimerivan@gmail.com
- 3008345754

2025

Contenido

1. Introducción

1.1 Propósito

Nuestro objetivo con este documento es dejar bien claro los requisitos que necesitamos para desarrollar el sistema de detección automática de placas vehiculares. Este sistema va a ayudar a los guardias de seguridad a validar si un vehículo tiene autorización de ingreso de manera automática y rápida.

1.2 Alcance

El proyecto es una aplicación web donde se utilizarán tecnologías del frontend como **HTML**, **CSS** y **Bootstrap** para la interfaz, mientras que la lógica estará a cargo de **JavaScript**, **jQuery** y **Axios**. En el backend, optamos por **Python** con el framework **Flask** porque es ligero y nos permite desarrollar rápido. Además, después de revisar varias opciones, vimos que **Tesseract OCR** o **OpenCV** son las mejores herramientas para procesar imágenes y reconocer las placas de los vehículos.

1.3 Personal involucrado

- **Jaimer Ivan Sosa** y **Michael Junior Vanegas Ramírez** (Desarrolladores principales).
- Rubén Darío (Asesor del proyecto).
- Equipo de seguridad de la universidad Cundinamarca (Usuarios principales del sistema)

1.4 Definiciones, acrónimos y abreviaturas

- **OCR**: Reconocimiento óptico de caracteres.
- **DOM**: Modelo de objetos del documento.
- **AJAX**: Asynchronous JavaScript and XML.
- **HTTP**: Protocolo de transferencia de hipertexto.

1.5 Referencias

- Documentación de **Flask**: https://flask.palletsprojects.com/
- Documentación de **OpenCV**: https://opencv.org/
- Documentación de **Tesseract OCR**: https://github.com/tesseract-ocr/tesseract

1.6 Resumen

Este documento detalla lo que necesitamos para llevar a cabo el proyecto, tanto a nivel técnico como de funcionalidad. Aquí especificamos qué herramientas y tecnologías vamos a usar, y cómo vamos a implementar el sistema.

---

2. Descripción general

2.1 Perspectiva del producto

El sistema se utilizará en áreas donde el acceso de vehículos está controlado, como estacionamientos o zonas de carga. Básicamente, una cámara capturará la imagen de la placa del vehículo, y gracias a las herramientas de procesamiento de imágenes como **Tesseract OCR** o **OpenCV**, podremos identificar el número de la placa y verificar en nuestra base de datos si ese vehículo tiene permiso para entrar. Decidimos usar **Heroku** para el despliegue porque nos facilita mucho el trabajo de poner todo en producción sin complicaciones.

2.2 Funcionalidad del producto

- **Captura de imagen**: Cuando un vehículo se acerque al punto de acceso, el sistema tomará una foto de la placa.
- **Reconocimiento de texto**: Aquí es donde entra en juego **Tesseract OCR** o **OpenCV** para extraer los caracteres de la placa.
- **Validación de acceso**: La placa se comparará con la base de datos en **PHP** para verificar si el vehículo está autorizado.
- **Notificación al guardia**: Si la placa está autorizada, el guardia recibirá una confirmación. Si no, se notificará que el vehículo no puede pasar.

2.3 Características de los usuarios

- **Guardia de seguridad**: Es quien va a usar el sistema para validar los accesos de los vehículos.
- **Administrador del sistema**: Es la persona encargada de actualizar la base de datos con las placas autorizadas.

2.4 Restricciones

- El sistema debe estar en **Heroku** para el despliegue porque nos facilita el manejo del servidor.
- La comunicación entre el frontend y el backend será mediante **Axios** para las solicitudes HTTP.

2.5 Suposiciones y dependencias

- Asumimos que las cámaras tendrán buena calidad para capturar imágenes claras de las placas, porque eso influye en la precisión del reconocimiento.
- El sistema necesita una conexión a Internet estable para que las consultas a la base de datos sean rápidas y eficientes.

---

3. Requisitos específicos

3.1 Requisitos comunes de las interfaces

3.1.1 Interfaces de usuario

- El guardia verá una interfaz simple y directa, desarrollada con **Bootstrap** para hacerlo más amigable. Aquí podrá ver la lista de placas autorizadas y recibir notificaciones cuando una placa sea identificada.

3.1.2 Interfaces de hardware

- Usaremos una cámara conectada al sistema para tomar las fotos de las placas.

3.1.3 Interfaces de software

- El backend hecho con **Flask** se conectará con la base de datos en **PHP** para la validación de placas.

3.1.4 Interfaces de comunicación

- Decidimos usar **jQuery con AJAX** para que las actualizaciones en la interfaz del cliente sean inmediatas, y **Axios** para la comunicación entre el frontend y el backend.

3.2 Requerimientos funcionales

1. El sistema tomará la foto de la placa automáticamente cuando el vehículo se acerque.
2. Usaremos **Tesseract OCR** o **OpenCV** para leer los caracteres de la placa.
3. La placa se validará comparándola con nuestra base de datos.
4. El guardia será notificado en tiempo real si el vehículo tiene o no autorización.
5. El administrador podrá agregar o quitar placas de la base de datos fácilmente.

3.3 Requerimientos no funcionales

3.3.1 Requisitos de rendimiento

- El sistema debe procesar cada placa en menos de 2 segundos, porque queremos que el flujo de entrada de vehículos sea rápido.

3.3.2 Seguridad

- Toda la información sensible debe estar cifrada, especialmente los datos de las placas autorizadas.

3.3.3 Fiabilidad

- Necesitamos que el sistema tenga una tasa de éxito de al menos el 90% en la identificación de las placas.

3.3.4 Disponibilidad

- El sistema debe estar operativo el 99.9% del tiempo para no causar interrupciones en el control de acceso.

3.3.5 Mantenibilidad

- Queremos que la base de datos sea fácil de actualizar sin necesidad de interrumpir el servicio.

3.3.6 Portabilidad

- El sistema debe ser fácil de trasladar a otras plataformas si en algún momento queremos cambiar de proveedor de servicios.

Transformando el acceso vehicular

RECONOCIMIENTO DE PLACAS EN ACCION

Contenido

PROPUESTA DE SOFTWARE

Proponemos el desarrollo de un sistema innovador que utilizará cámaras instaladas en las entradas y salidas del campus universitario para capturar imágenes de las placas de los vehículos. Este sistema tiene como objetivo mejorar la seguridad y el control de acceso, asegurando que solo los vehículos autorizados puedan ingresar y salir.

Beneficios:

- Aumento de la Seguridad: Control más riguroso sobre quién ingresa al campus.
- Eficiencia en el Proceso de Autorización: Reducción de tiempos de espera al ingresar o salir.
- Facilidad de Uso: Interfaz intuitiva que mejora la experiencia del usuario.

02

IDENTIFICACION DEL PROBLEMA

Acceso no autorizado:
Personas no autorizadas pueden ingresar al campus sin ser detectadas.
ingresar vehiculos no registrados en la plataforma de la institución

- **Falta de monitoreo y control:**
Dificultad para llevar un registro exacto de los vehículos que ingresan y salen.

Congestión en la entrada/salida:
Procesos manuales de verificación de identidad que causan demoras.

- **Problemas de seguridad:**
Dificultad para identificar y actuar ante vehículos sospechosos o robados dentro del campus.

MISIÓN

Proponer un sistema de reconocimiento de placas de vehículos que mejore la seguridad y el control de acceso en instalaciones o recintos cerrados, facilitando un ambiente seguro y eficiente mediante el uso de tecnologías avanzadas de procesamiento de imágenes y gestión de datos.

VISIÓN

Ser la solución en gestión de acceso vehicular en instalaciones o recintos, innovando constantemente en el uso de tecnologías emergentes para crear un entorno más seguro y accesible, donde la tecnología y la seguridad se integran de manera efectiva.

OBJETIVO GENERAL

Ser la solución líder en gestión de acceso vehicular en instalaciones o recintos, innovando continuamente en el uso de tecnologías emergentes para crear un entorno más seguro y accesible, donde la tecnología y la seguridad se integran de manera efectiva.

objetivos especificos

1

Implementar un módulo de captura y procesamiento de imágenes para detectar y reconocer las placas vehiculares en tiempo real usando OpenCV y Python.

2

Desarrollar una base de datos centralizada que almacene la información de los vehículos registrados, incluyendo la matrícula, el historial de entradas y salidas, y los permisos de acceso.

3

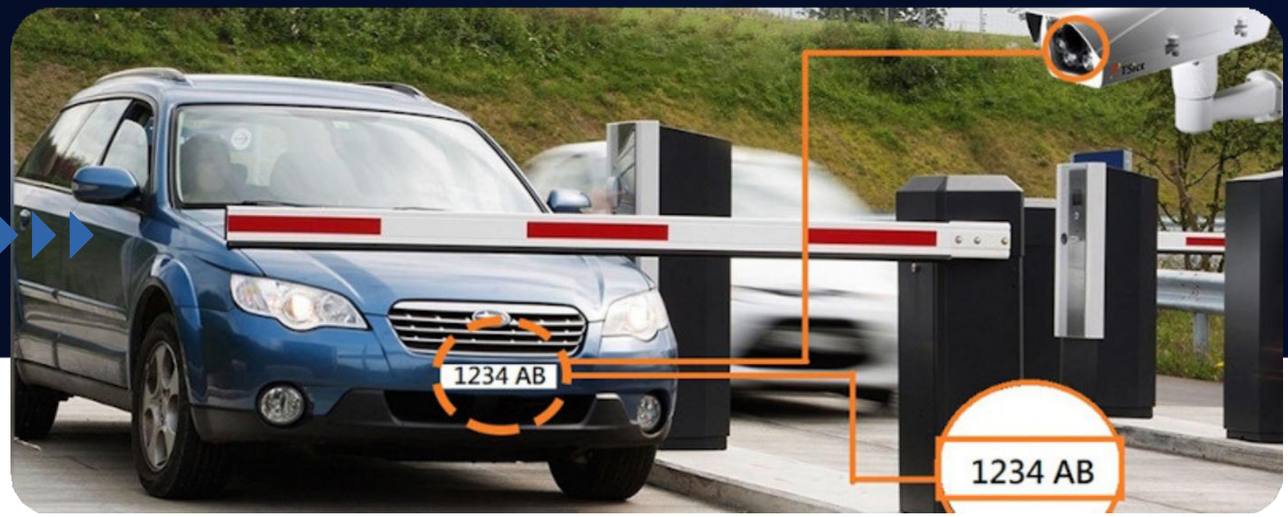
Optimizar el flujo de tráfico en las entradas y salidas del campus mediante un sistema de acceso automatizado que minimice la necesidad de intervención humana.

4

Optimizar el flujo de tráfico en las entradas y salidas del campus mediante un sistema de acceso automatizado que minimice la necesidad de intervención humana.

06

FUNCIONALIDAD



- Captura de Imágenes:
- Instalación de cámaras en puntos estratégicos para capturar imágenes de las placas
- de vehículos en tiempo real.
- Reconocimiento de Placas:
- Procesamiento de imágenes utilizando OpenCV para identificar y extraer caracteres de las placas de los vehículos.
- Registro Automático:
- Almacenamiento automático de los datos de las placas reconocidas en una base de datos centralizada.
- Verificación de Autorización:
- Comparación de las placas registradas con una lista de vehículos autorizados para determinar el acceso.
- Interfaz Web Intuitiva:
- Desarrollo de una aplicación web que permita a los usuarios (personal de seguridad) consultar información en tiempo real sobre el acceso vehicular.
- Notificaciones en Tiempo Real:
- Alertas automáticas en caso de que un vehículo no autorizado intente ingresar o salir del campus.
- Generación de Reportes:
- Creación de reportes periódicos sobre el acceso vehicular, incluyendo estadísticas sobre vehículos autorizados y no autorizados.
- Actualización y gestión de la base de datos para agregar, eliminar o modificar información sobre vehículos autorizados.

Estadísticas

ENCUESTA DE REQUERIMIENTOS

según nuestra encuesta realizada los resultados los identificamos de la siguiente manera:

Plan de acción

Metodologia cascada

Utilizaremos la metodología cascada porque su enfoque estructurado y secuencial permite una planificación clara y un desarrollo organizado, cada fase se completa antes de avanzar a la siguiente, lo que facilita la gestión de requisitos y la documentación.

SOFTWARE J.M

FASE DE REQUERIMIENTOS

Se llevará a cabo una recopilación exhaustiva de las necesidades del personal de seguridad y administradores del campus. Esto incluirá identificar requerimientos funcionales como el de la automatización de la entrada al campus y como principal la organización y seguridad del mismo

FASE DE ANALISIS

Se comprenderá su viabilidad y alcance. Esto implicará descomponer los requerimientos en funciones específicas, identificar interacciones entre componentes del sistema y analizar posibles restricciones técnicas y operativas.

FASE DE DISEÑO

Se crearán especificaciones detalladas para cada componente del sistema, incluyendo la arquitectura del software, la interfaz de usuario y la base de datos. Se desarrollarán prototipos de la interfaz web para asegurar que sea intuitiva y fácil de usar.

FASE DE PRUEBAS

Se llevarán a cabo diversas pruebas para asegurar que todas las funcionalidades operen según lo previsto y que los requisitos establecidos se cumplan adecuadamente y también pruebas de integración para evaluar cómo interactúan los distintos módulos del sistema

FASE DE DESARROLLO

Se llevará a cabo la codificación de todos los componentes definidos en la fase de diseño. Los desarrolladores implementarán las funcionalidades del sistema, incluyendo la captura de imágenes, el procesamiento y reconocimiento de placas, y la creación de la interfaz web.

FASE DE MANTENIMNIENTO

Se llevarán a cabo actividades continuas para garantizar el correcto funcionamiento y la actualización del sistema a lo largo del tiempo.

TÉCNICAS PARA EL LEVANTAMIENTO DE REQUERIMIENTOS

10

Entrevistas con los Usuarios Finales:
Realizar entrevistas estructuradas con el personal de seguridad, administración de la universidad, y potencialmente algunos estudiantes para entender sus necesidades y expectativas respecto al sistema.

Observación Directa:
Observar el flujo actual de vehículos en los puntos de acceso al campus para identificar problemas específicos, tales como demoras o procedimientos ineficientes.

Revisión de Documentación Existente:
Analizar documentos actuales relacionados con las políticas de acceso vehicular, reportes de seguridad, y registros de incidentes para entender los requerimientos legales y administrativos.

Cuestionarios y Encuestas:
Distribuir cuestionarios a un grupo más amplio de usuarios potenciales para recopilar datos sobre sus necesidades y prioridades.

10 TÉCNICAS DE ANÁLISIS DE REQUERIMIENTOS

Análisis de Tareas:
Descomponer las actividades del control vehicular actual en tareas más pequeñas para identificar cómo se podría mejorar o automatizar con el nuevo sistema.

Modelado de Casos de Uso:
Crear diagramas de casos de uso que describan cómo diferentes tipos de usuarios interactuarán con el sistema, identificando requisitos funcionales clave.

11 TÉCNICAS DE ESPECIFICACIÓN DE REQUERIMIENTOS

Historias de Usuario:
Crear historias de usuario claras y concisas para cada funcionalidad del sistema. Por ejemplo, "Como miembro del personal de seguridad, quiero poder recibir alertas automáticas para vehículos no autorizados para tomar medidas rápidamente".

Prototipos:
Desarrollar prototipos de la interfaz de usuario para validar la usabilidad y funcionalidad con los usuarios finales antes de la implementación completa.

TÉCNICAS DE VALIDACIÓN DE REQUERIMIENTOS

10

Revisiones con Stakeholders:
Organizar sesiones de revisión periódicas con los interesados clave para validar que los requerimientos documentados reflejan correctamente sus necesidades.

Pruebas de Usabilidad:
Realizar pruebas de usabilidad con prototipos para asegurarse de que los usuarios finales puedan interactuar con el sistema de manera efectiva y eficiente.

Revisión de Consistencia y Completitud:
Verificar que todos los requerimientos estén completos, no se contradigan, y sean consistentes con los objetivos del proyecto.

CONCLUSIONES

El desarrollo de un sistema de reconocimiento de placas abordará problemas críticos de seguridad y eficiencia en el control vehicular. Mediante el uso de tecnologías avanzadas como OpenCV y Python, y siguiendo un proceso estructurado de levantamiento y análisis de requerimientos, se garantizará que el sistema cumpla con las necesidades de todos los stakeholders y se integre de manera efectiva en las operaciones diarias del recinto o instalación.

Este enfoque proporcionará una solución robusta y escalable que no solo mejorará la seguridad, sino que también optimizará la experiencia de todos los usuarios involucrados.

¡Gracias por su atención!

INGENIERIA DE SOFTWARE