



SCAV

CAPSTONE_006V

Informe avances

Proyecto

Integrantes

Nibaldo Quezada
Cristian Arroyo
Nicolas Peña

Docente

Fabián Alcántara

Contenido

Contenido

Introducción	3
Descripción del proyecto APT	4
Requerimientos de Alto Nivel:	4
Diagrama de arquitectura	6
Planificación	7
Avances del proyecto	8
Ajustes del proyecto.....	9
Metodología: Corrección hacia Scrum	9
Cronograma del Proyecto: Corrección de Sprints	10
Desarrollo de Aplicaciones: División entre Aplicación Administrativa y de Residentes	11

Introducción

El proyecto TechApps - SCAV tiene como objetivo desarrollar un sistema integral de control de acceso vehicular para el condominio Vista Parque 1 y 2. Esta iniciativa surge a partir de las dificultades que enfrenta el condominio en la gestión manual del acceso vehicular, lo que ocasiona congestión en los puntos de entrada y salida, además de vulnerabilidades de seguridad ante posibles robos. El sistema propuesto permitirá autorizar, monitorear y restringir el acceso de vehículos de manera automática, empleando tecnologías como reconocimiento de patentes, bases de datos, y aplicaciones móviles. Este proyecto permitirá a los estudiantes aplicar competencias clave de su perfil profesional, como la gestión de proyectos, el desarrollo de software y la implementación de soluciones informáticas para problemas reales, asegurando una experiencia práctica y acorde a los estándares de la industria.

Descripción del proyecto APT

El objetivo del proyecto es desarrollar un sistema que permita mejorar la gestión y seguridad del acceso vehicular en el condominio. El sistema deberá registrar, autorizar y monitorear el ingreso y salida de vehículos, asegurando un control eficiente y seguro.

Requerimientos de Alto Nivel:

Uso Administradores y Personal de Seguridad:

- Manejo de Registro de Residentes:
 - Registro y actualización de la información vehicular y personal de los residentes.
 - Administración de permisos de acceso para los vehículos registrados de cada residente.
- Manejo de Registro de Visitas:
 - Registro anticipado de visitas por parte de los residentes o por personal administrativo.
 - Autorización o denegación de acceso a las visitas, basada en políticas definidas por el recinto.
 - Notificación automática a los residentes sobre la llegada de una visita.
- Autorización de Vehículos Comerciales:
 - Registro de solicitudes de acceso para vehículos comerciales (ej. entregas de encomiendas, servicios de comida, servicios de transporte).
 - Revisión, aprobación o rechazo de dichas solicitudes por parte del personal autorizado.
- Ingreso de Vehículos Emergencia:
 - El conserje autoriza la entrada de forma manual en caso de vehículos que correspondan a Bomberos, ambulancia, policiales.
- Monitoreo y Reportes de Acceso:
 - Monitoreo en tiempo real de entradas y salidas de vehículos, mediante bitácora y reportería.
 - Generación de reportes de acceso diarios, semanales o mensuales, que incluyan datos sobre todos los ingresos, con filtros por tipo de vehículo, residente, visita, etc.
- Integración con API Externa para Validación Vehicular:
 - Validación automática de los datos vehiculares (marca, modelo, propietario) mediante una API externa, asegurando que solo vehículos registrados puedan acceder.

- Aplicación automática de multas:
 - El sistema deberá ser capaz de identificar cuando una visita permanece por más de las horas autorizadas (Definido por el condominio), y aplicar la multa correspondiente.

Uso Residentes:

- Solicitud de Acceso para Visitas:
 - App móvil donde los residentes pueden registrar visitas anticipadamente, proporcionando la información del vehículo.
 - Recepción de notificaciones y actualizaciones sobre el estado de la visita (aprobada, rechazada).
- Solicitud de Acceso para Servicios Comerciales:
 - Validación por parte del residente receptor del servicio, si se requiere.
- Recepción de Notificaciones y Alertas:
 - Envío de notificaciones automáticas a los residentes cuando una visita registrada llega al recinto.
 - Actualizaciones periódicas sobre políticas de acceso y seguridad del recinto.
- Consulta de Historial de Accesos:
 - Acceso a un historial personal de entradas y salidas, tanto de los vehículos del residente como de las visitas autorizadas, y servicios externos, disponible a través de la app.

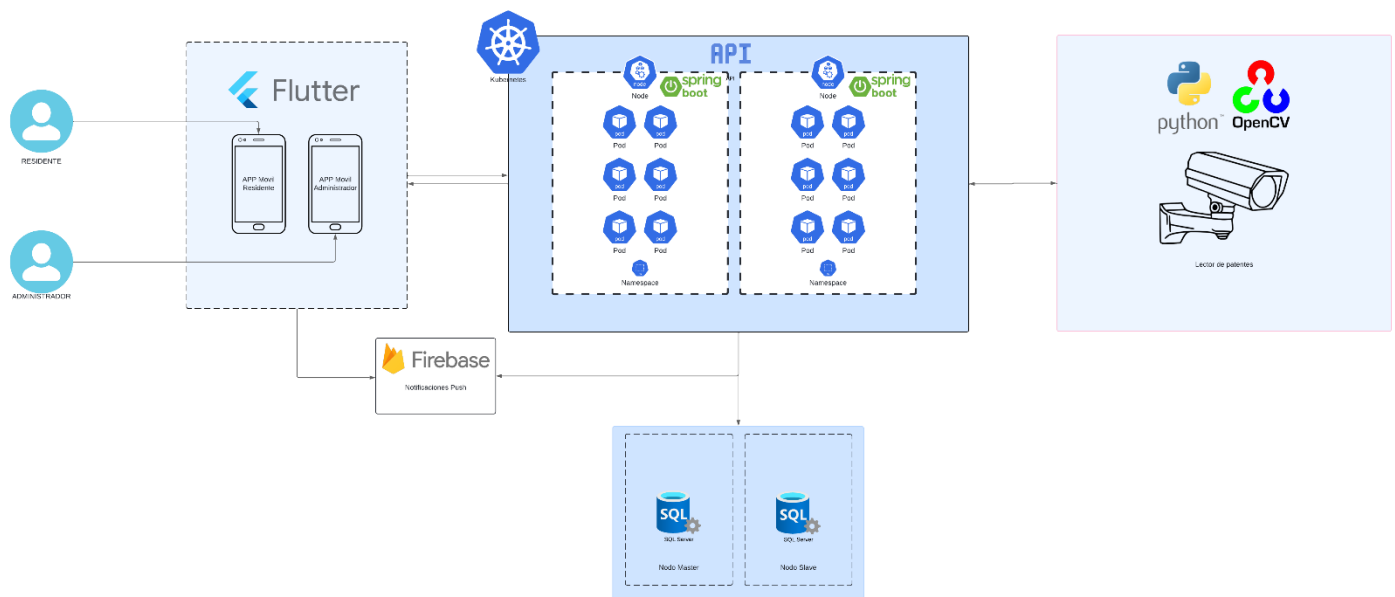
Arquitectura del Sistema:

- Lector de patente: Python OpenCV
- Base de Datos: SQL Server
- Api Rest: Spring Boot
- App Móvil: Flutter
- Api Externa: Auth ApiKey

Este sistema permitirá una gestión eficaz del acceso vehicular, mejorando la seguridad y el control en recintos residenciales, y garantizando una experiencia fluida tanto para los residentes como para los visitantes y proveedores.

Diagrama de arquitectura

El sistema se compone de tres componentes principales, el primero corresponde a una API desarrollada en Springboot con Java17, el segundo un sistema de lectores de patentes realizado en Python 3 con OpenCV para el procesamiento de imágenes, y el tercero, dos aplicaciones desarrolladas en Flutter, una para los usuarios del condominio y otro para los administrativos. Por otra parte, el sistema contará con una base de datos relacional implementada en SQL Server. Finalmente se realizará las notificaciones del sistema con Firebase mediante su servicio de notificaciones-push.



[illegible]

Avances del proyecto

Actualmente nos encontramos en la fase de desarrollo de las aplicaciones móviles en flutter, a continuación, se adjunta un cuadro resumen con el estado de cada tarea realizada durante el proyecto.

Actividades y Estado de Avance Materiales:

Actividad	Estado	Dificultades	Materiales/Recursos Físicos	Lenguajes/Entorno
Lector de Patentes	Completado	Dificultades para hacer pruebas en vivo con la cámara en la calle debido a condiciones climáticas y tráfico	Cámara de video, PC	Python, OpenCV
Diseño de Base de Datos	Completado	N/A	PC, servidor SQL	SQL Server
API REST	En curso	N/A	PC, servidor para la API	Java, Spring Boot
Aplicación Móvil en Flutter	En curso	N/A	PC, smartphone para pruebas	Flutter, Dart
Pruebas de Integración	No iniciado	Coordinación para pruebas con cámara en tiempo real en un entorno controlado	Cámara de video, PC, red local para la integración	Python, Java, SQL, Flutter

Ajustes del proyecto

Metodología: Corrección hacia Scrum

Desde el inicio del proyecto TechApps - SCAV, la metodología Scrum fue la propuesta y establecida como la guía para gestionar el desarrollo. Sin embargo, durante las primeras fases de implementación, por un error del equipo, las actividades comenzaron a gestionarse bajo un enfoque más tradicional. Este desvío se debió a la organización de tareas como bloques independientes, con fechas de entrega fijas y poca interacción entre los módulos en desarrollo (lector de patentes, API REST, base de datos y aplicación móvil).

Al identificar este error, se realizó un ajuste inmediato para volver al enfoque ágil Scrum, lo cual permitió:

1. Recuperar la planificación iterativa e incremental: Se organizaron los trabajos en los sprints, lo que garantiza mayor flexibilidad para incorporar cambios según las necesidades.
2. Mejor coordinación y colaboración: Las reuniones diarias (daily meetings) se reintrodujeron para alinear al equipo y resolver bloqueos rápidamente.
3. Entregas parciales frecuentes: Se retomaron las entregas incrementales en cada sprint, lo que permitió presentar avances funcionales, como el lector de patentes y la integración de la API con la base de datos.
4. Retroalimentación continua: Se fomenta la interacción constante con los involucrados para garantizar que cada módulo se desarrollará conforme a las expectativas y requerimientos del proyecto.

Este ajuste ha sido crucial para recuperar la dinámica ágil que se había perdido. El equipo se alineó nuevamente con la metodología Scrum, permitiendo una mejor sincronización entre los módulos, optimizando los tiempos de desarrollo y asegurando que el proyecto avance según lo planeado.

Cronograma del Proyecto: Corrección de Sprints

Inicialmente, el proyecto TechApps - SCAV se organizó en un número diferente de sprints, lo cual no permitía una planificación adecuada para la implementación de todos los módulos. Tras una revisión del cronograma, se realizó una corrección en la cantidad y duración de los sprints, con el fin de optimizar el desarrollo y cumplir con los objetivos planteados.

Nuevo Cronograma:

- Total de Sprints: 9 Sprints.
- Duración de cada Sprint: 2 semanas cada uno.
- Distribución por Módulos:
 1. Gestión de Proyecto y Backlog: Sprint 1 (S1) y (S2), Sprint 2 (S3) y (S4).
 2. Desarrollo del Sistema de Reconocimiento de Patentes: Sprint 3 (S5) y (S6).
 3. Desarrollo del Modelo de Base de Datos: Sprint 4 (S7) y (S7).
 4. Implementación de API REST: Sprint 5 (S9) y (S10).
 5. Aplicación Móvil en Flutter: Se ajustó a dos sprints completos: Sprint 6 (S11) y (S11), Sprint 7 (S13) y (S14).
 6. Pruebas de Calidad e Integración: Sprint 8 (S15) y (S16).
 7. Marcha Blanca y Capacitación: Sprint 9 (S17) y (S18).

Impacto del Ajuste:

- Redistribución del Tiempo: Las fechas se ajustaron para que los módulos más críticos tengan más tiempo asignado, como el desarrollo de la aplicación móvil en Flutter, que se desarrollará a lo largo de 2 sprints completos.
- Optimización del Flujo: El resto de los módulos se desarrollarán en sprints únicos para evitar la fragmentación del trabajo.

Desarrollo de Aplicaciones: División entre Aplicación Administrativa y de Residentes

Durante la planificación inicial del proyecto TechApps - SCAV, se estableció el desarrollo de una única aplicación que abarcaría tanto las funciones administrativas como las de los residentes. Sin embargo, durante la fase de implementación, identificamos que combinar ambas funcionalidades en una sola app generaba complejidad innecesaria y dificulta la usabilidad tanto para administradores como para residentes.

Por esta razón, se decidió dividir el proyecto en dos aplicaciones independientes:

1. Aplicación Administrativa:

- Funciones enfocadas en la gestión de residentes, visitas, y vehículos comerciales.
- Acceso a reportes y monitoreo en tiempo real de los accesos vehiculares.
- Administración de permisos y gestión del flujo vehicular.
- Automatización de multas

2. Aplicación de Residentes:

- Registro anticipado de visitas.
- Recepción de notificaciones en tiempo real sobre la llegada de visitas y entregas.
- Visualización del historial de accesos personales.
- Solicitud y autorización de acceso para servicios comerciales.

Enfoque Actual:

Actualmente, el desarrollo del proyecto está enfocado en la aplicación administrativa, dado que esta es prioritaria para la puesta en marcha del sistema. Se ha avanzado significativamente en la implementación de funciones críticas como la gestión de permisos y la generación de reportes, ya que estas funcionalidades son esenciales para asegurar una operación eficiente del control de acceso vehicular en el condominio Vista Parque.

Evidencias de avance

A continuación, se adjuntarán algunos prints de pantalla correspondientes a los avances de los distintos módulos desarrollados durante el proyecto.

Módulo Lector de patentes:



Ilustración 1 - Implementación lector de patentes

Módulo API Rest:

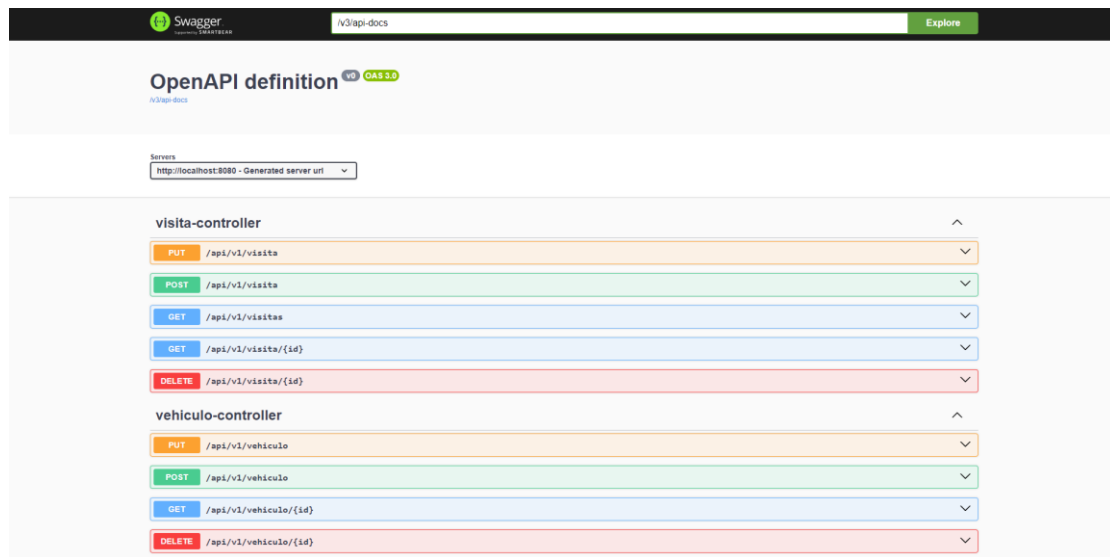


Ilustración 2 - Implementación API Rest

Módulo Flutter:

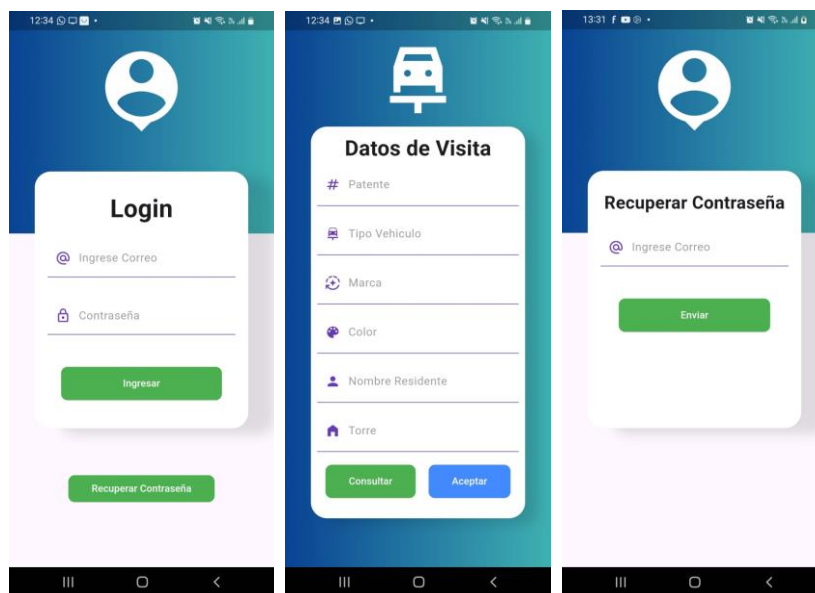


Ilustración 3 - Pantallas aplicación Flutter

Modelo Entidad Relación:

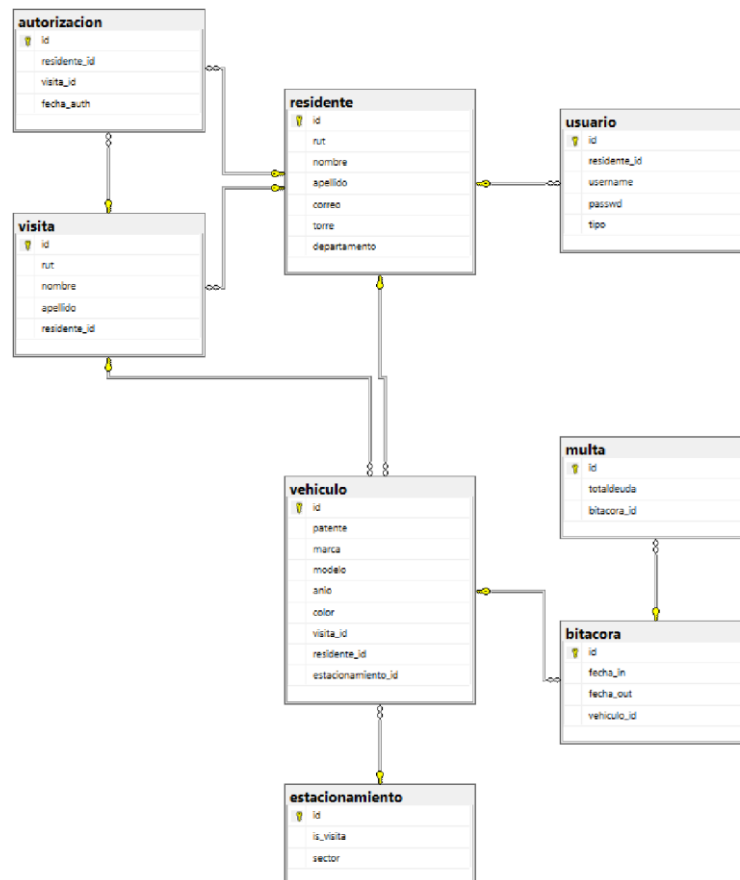


Ilustración 4 - MER