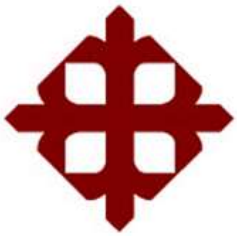


UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO
CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

TEMA:

Estudio de un Sistema de control Inteligente para el Estacionamiento del Canal y Radio UCSG.

AUTOR:

Angulo Angulo, Ariel Eduardo

Trabajo de Integración Curricular previo a la obtención del título de
INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN

TUTOR:

Ing. Bohórquez Escobar, Celso Bayardo. PHD.

Guayaquil, Ecuador

15 de febrero del 2024

UNIVERSIDAD CATÓLICA

DE SANTIAGO DE GUAYAQUIL

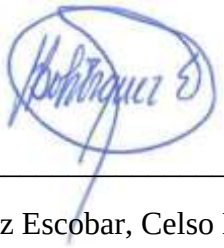
FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

CERTIFICACIÓN

Certificamos que el presente Trabajo de Integración Curricular fue realizado en su totalidad por el Sr. Angulo Angulo, Ariel Eduardo, como requerimiento para la obtención del título de INGENIERO EN ELECTRÓNICA Y AUTOMATIZACIÓN.

TUTOR



Ing. Bohórquez Escobar, Celso Bayardo. PHD.

DIRECTOR DE CARRERA

Ing. Bohórquez Escobar, Celso Bayardo. PHD.

Guayaquil, a los 15 días del mes de febrero del año 2024

UNIVERSIDAD CATÓLICA

DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

DECLARACIÓN DE RESPONSABILIDAD

Yo, Angulo Angulo, Ariel Eduardo

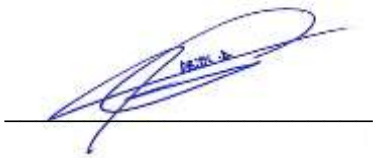
DECLARO QUE:

El trabajo de Integración Curricular: **Estudio de un Sistema de control Inteligente para el Estacionamiento del Canal y Radio UCSG,** previo a la obtención del Título de **Ingeniero en Electrónica y automatización**, ha sido desarrollado respetando derechos intelectuales de terceros conforme las citas que constan en el documento, cuyas fuentes se incorporan en las referencias o bibliografías. Consecuentemente este trabajo es de mi total autoría.

En virtud de esta declaración, me responsabilizo del contenido, veracidad y alcance del Trabajo de Integración Curricular referido.

Guayaquil, 15 días del mes de febrero del año 2024

EL AUTOR



Angulo Angulo, Ariel Eduardo

UNIVERSIDAD CATÓLICA

DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

AUTORIZACIÓN

Yo, Angulo Angulo, Ariel Eduardo

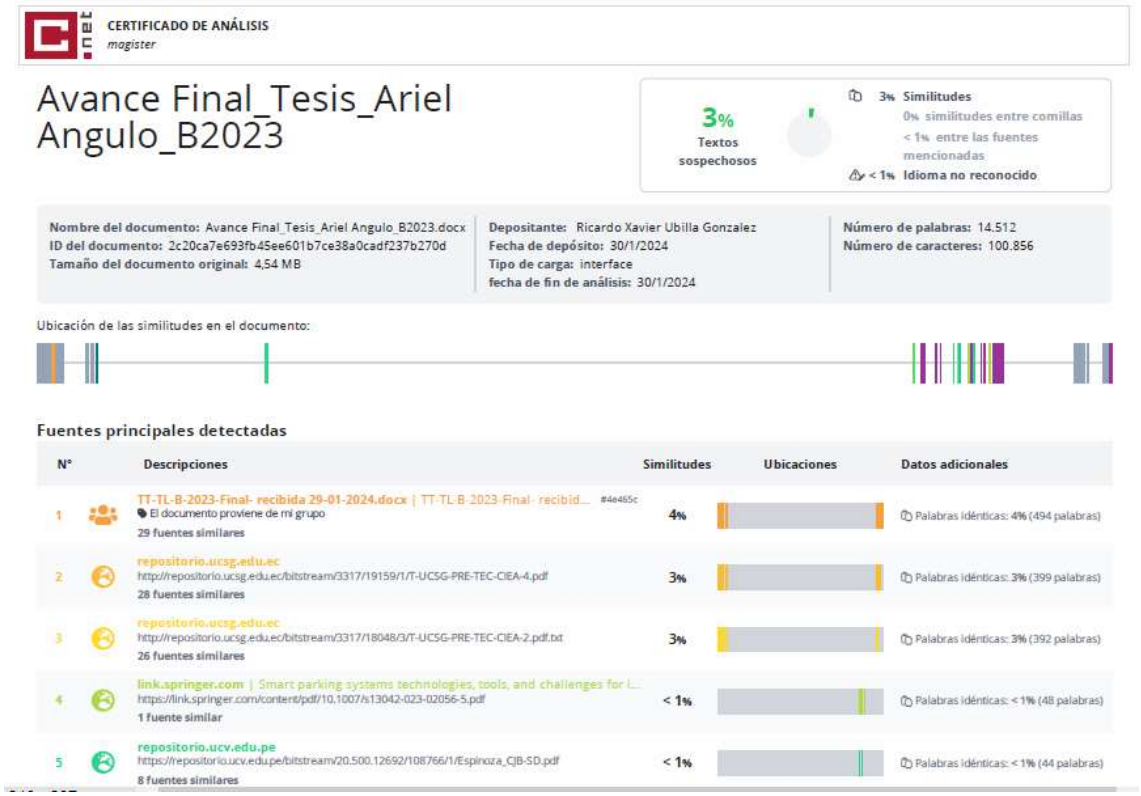
Autorizo a la Universidad Católica de Santiago de Guayaquil, la publicación, en la biblioteca de la institución del Trabajo de Integración Curricular: **Estudio de un Sistema de control Inteligente para el Estacionamiento del Canal y Radio UCSG**, cuyo contenido, ideas y criterios son de mi exclusiva responsabilidad y total autoría.

Guayaquil, 15 días del mes de febrero del año 2024

EL AUTOR

Angulo Angulo, Ariel Eduardo

REPORTE DE COMPILATIO



CERTIFICADO DE ANÁLISIS
magister

Avance Final_Tesis_Ariel Angulo_B2023

3% Textos sospechosos

3% Similitudes
0% similitudes entre comillas
< 1% entre las fuentes mencionadas
< 1% Idioma no reconocido

Nombre del documento: Avance Final_Tesis_Ariel Angulo_B2023.docx
ID del documento: 2c20ca7e693fb45ee601b7ce38a0cadf237b270d
Tamaño del documento original: 4,54 MB

Depositante: Ricardo Xavier Ubilla Gonzalez
Fecha de depósito: 30/1/2024
Tipo de carga: interface
fecha de fin de análisis: 30/1/2024

Número de palabras: 14.512
Número de caracteres: 100.856

Ubicación de las similitudes en el documento:

Fuentes principales detectadas

N°	Descripciones	Similitudes	Ubicaciones	Datos adicionales
1	TT-TL-B-2023-Final- recibida 29-01-2024.docx | TT-TL-B-2023-Final- recibid... #4e465c El documento proviene de mi grupo 29 fuentes similares	4%		Palabras idénticas: 4% (494 palabras)
2	repositorio.ucsg.edu.ec http://repositorio.ucsg.edu.ec/bitstream/3317/19159/1/T-UCSG-PRE-TEC-CIEA-4.pdf 28 fuentes similares	3%		Palabras idénticas: 3% (399 palabras)
3	repositorio.ucsg.edu.ec http://repositorio.ucsg.edu.ec/bitstream/3317/18048/3/T-UCSG-PRE-TEC-CIEA-2.pdf.txt 26 fuentes similares	3%		Palabras idénticas: 3% (392 palabras)
4	link.springer.com | Smart parking systems technologies, tools, and challenges for i... https://link.springer.com/content/pdf/10.1007/s13042-023-02056-5.pdf 1 fuente similar	< 1%		Palabras idénticas: < 1% (48 palabras)
5	repositorio.ucv.edu.pe https://repositorio.ucv.edu.pe/bitstream/20.500.12692/108766/1/Espinoza_CJB-SD.pdf 8 fuentes similares	< 1%		Palabras idénticas: < 1% (44 palabras)

Reporte COMPILATIO del estudiante Angulo Angulo Ariel Eduardo, de la Carrera de Ingeniería en Electrónica y Automatización R con el tema "**Estudio de un Sistema de control Inteligente para el Estacionamiento del Canal y Radio UCSG"**, mismos que se encuentra al 3% de coincidencias.

Ing. Bayardo Bohórquez Escobar, MSc
DOCENTE-TUTOR

AGRADECIMIENTO

Hoy, con el corazón lleno de gratitud y emoción, quiero expresar mi más sincero agradecimiento a cada uno de ustedes. En este viaje hacia la culminación de mi tesis, su amor, apoyo y aliento han sido mi mayor fortaleza. Agradezco a Dios por ser mi guía constante y por iluminar mi camino con su sabiduría divina. A mi amado papá Eduardo Angulo y a mi cariñosa mamá Alexa Angulo, su sacrificio y amor incondicional han sido mi inspiración y motivación constante.

A ti, Daniela Resabala por ser mi compañera y mi fuente de amor y apoyo. A mis hermanos por su compañía y cariño han sido mi refugio en los momentos difíciles. Agradezco de todo corazón a mi dedicado tutor, Ingeniero Bayardo Bohórquez., por su orientación experta y paciencia inquebrantable. También quiero reconocer y honrar a todos los ingenieros de mi facultad, cuyo conocimiento y dedicación han sido una fuente interminable de aprendizaje.

Este logro no es solo mío, sino de toda nuestra familia. Cada palabra de aliento, cada gesto de apoyo ha sido un pilar fundamental en este viaje. Con profunda gratitud, dedico este logro a cada uno de ustedes. Gracias por creer en mí, por estar a mi lado y por ser mi mayor motivación.

Con amor y agradecimiento eterno,

Ariel Eduardo Angulo Angulo

DEDICATORIA

En el camino de esta travesía académica, reconozco la guía divina de Dios, el pilar inquebrantable de mi papá Eduardo Angulo, la amorosa presencia de mi mamá Alexa Angulo, el apoyo incondicional de mi novia Daniela Resabala y la fortaleza de mis hermanos. Agradezco profundamente la sabiduría impartida por mi tutor Ingeniero Bayardo Bohórquez Escobar, así como el esfuerzo y dedicación de todos los ingenieros de mi facultad. Esta victoria también es de toda mi familia, cuyo amor y aliento han sido mi fuerza. Con gratitud eterna, dedico este logro a cada uno de ustedes, sabiendo que su apoyo ha sido el faro que me ha guiado hacia el éxito. ¡Gracias por estar siempre a mi lado!

Ariel Eduardo Angulo Angulo

UNIVERSIDAD CATÓLICA
DE SANTIAGO DE GUAYAQUIL

FACULTAD DE EDUCACIÓN TÉCNICA PARA EL DESARROLLO

CARRERA DE INGENIERÍA ELECTRÓNICA Y AUTOMATIZACIÓN

TRIBUNAL DE SUSTENTACIÓN

f. _____

ING. BOHORQUEZ ESCOBAR, CELSO BAYARDO PHD.

DIRECTOR DE CARRERA

f. _____

ING. UBILLA GONZÁLEZ, RICARDO XAVIER, MSC.

COORDINADOR DE ÁREA

f. _____

ING. ZAMORA CEDEÑO, NÉSTOR ARMANDO, MSC.

OPONENTE

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

Capítulo 2:

Capítulo 3:

INDICE DE TABLAS

Capítulo 3:

Resumen

El presente trabajo de integración curricular se basa en la formulación de un Sistema de Control Inteligente para el Estacionamiento del Canal y Radio UCSG a través de la utilización de Arduino Mega, este sistema conllevará a mejoras significativas en la gestión del estacionamiento. Se prevé que, al obtener información en tiempo real sobre la disponibilidad de espacios, el sistema logrará reducir la congestión, optimizar la asignación de lugares y enriquecer la experiencia global de estacionamiento para los usuarios. Se llevará a cabo una exhaustiva revisión de la literatura existente para recopilar información sobre sistemas de control de estacionamiento, tecnología Arduino, aplicaciones de sistemas inteligentes en entornos de estacionamiento. Además, se aspira que este proyecto de integración curricular contribuya en la necesidad de una distribución más efectiva de los espacios de estacionamiento. Tal sistema no solo mejoraría la experiencia de los usuarios, sino que además optimizaría el uso de los recursos disponibles.

Palabras claves: Control Inteligente, Arduino Mega, Sensores, Optimización de Espacios, Estacionamiento.

ABSTRACT

The present work of curricular integration is based on the formulation of an Intelligent Control System for the UCSG Canal and Radio Parking through the use of Arduino Mega, this system will lead to significant improvements in parking management. It is anticipated that by obtaining real-time information on space availability, the system will succeed in reducing congestion, optimizing space allocation and enriching the overall parking experience for users. A comprehensive review of existing literature will be conducted to gather information on parking control systems, Arduino technology, applications of smart systems in parking environments. In addition, it is aspired that this curriculum integration project will contribute to the need for a more effective distribution of parking spaces. Such a system would not only improve the user experience, but also optimize the use of available resources.

Keywords: Smart Control, Arduino Mega, Sensors, Space Optimization, Parking.

Capítulo 1: Descripción general del trabajo de titulación

1.1 Introducción

En los últimos decenios, el incremento de población y la urbanización han producido complicaciones importantes en el ámbito de la administración de provisiones de la ciudad. La administración del lugar de estacionamiento se ha convertido en una cuestión importante, debido a que la deficiencia de lugares de estacionamiento señalados ha generado dudas en torno a la comodidad que tienen los habitantes de las ciudades. Frente a este inconveniente, se han generado diversas soluciones de tecnología, siendo el sistema de control de automóviles uno de los más fundamentales.

El eje de la investigación de este trabajo se basa en el crecimiento y la verificación de un programa de control inteligente con el fin de que el estacionamiento del canal y la radio UCSG, se realiza con la ayuda de Arduino Mega. El objetivo principal de este programa es aumentar la efectividad y experiencia del estacionamiento para los usuarios y al mismo tiempo mejorar la eficiencia del sistema de estacionamiento.

1.2 Antecedentes

En la actualidad, los sistemas de control de estacionamiento han demostrado una importante transformación para hacer frente a la grande demanda de lugares para estacionarse que hay en las urbanizaciones. En tiempos antiguos, la administración de lugares para estacionamientos se apoyaba en la observación de la humanidad y en la

administración manual, esto a menudo generaba ineficacias y filas de espera excesivas para los conductores.

Con el avance de la tecnología, se han generado soluciones más prácticas, como los sistemas de estacionamiento automático. Estos sistemas utilizan sensores, cámaras y herramientas de información en tiempo real para controlar y supervisar la existencia de lugares para estacionar vehículos. A pesar de ello, varios de estos sistemas comerciales son costosos y no siempre se adaptan a las particularidades del entorno, como es el caso del Canal y Radio UCSG.

1.3 Definición del Problema

La carencia de un sistema de gestión de estacionamiento eficaz en el entorno de Canal y Radio UCSG ha generado una asignación ineficiente de espacios, resultando en una utilización inadecuada de los recursos disponibles y, en última instancia, provocando una sensación general de insatisfacción entre la comunidad universitaria y los visitantes. Esta problemática se intensifica considerablemente durante eventos especiales, como conferencias o actividades culturales, cuando la afluencia de vehículos se incrementa significativamente.

El problema reside en la necesidad imperante de implantar un sistema de control inteligente que permita una distribución más efectiva de los espacios de estacionamiento. Tal sistema no solo mejoraría la experiencia de los usuarios, sino que además optimizaría el uso de los recursos disponibles. La ausencia de una solución idónea no solo repercute negativamente en la calidad de vida en el entorno

universitario, sino que también constituye un desafío logístico que demanda una pronta atención.

1.4 Justificación del Problema

La justificación de esta problemática se apoya en varias cuestiones importantes que se distribuyen en varias áreas. El primer efecto, y quizás el más significativo, sea el que se note directamente dentro de la comunidad de las universidades: la falta de lugares de estacionamiento para automóviles es perceptible en los campus, esto afecta a los estudiantes, los profesores, el personal y los invitados, y genera alta ansiedad y malestar debido a la congestión de automóviles y la dificultad para hallar lugares de estacionamiento. Además, la concentración de automóviles en áreas planificadas aumenta el peligro de accidente y demora, esto pone en riesgo la seguridad de las personas y afecta la capacidad eficaz de la institución. La inefectividad en la distribución de lugares, a causa de la carencia de un sistema de control óptimo, genera zonas sobrantes y otras colmadas, ocasionando una distribución inefectiva. Este problemático no sólo tiene una influencia sobre la comodidad, sino que además posee una influencia ambiental debido a la emisión permanente de gases de origen industrial.

1.5 Objetivos del Problema de Investigación

1.5.1 Objetivo general

- Diseñar un Sistema de Control Inteligente, utilizando la plataforma de Arduino Mega, para el Estacionamiento del Canal y Radio UCSG.

1.5.2 Objetivos específicos

- Analizar las necesidades y demandas de estacionamiento en el entorno de Canal y Radio UCSG.

- Establecer el sistema de control basado en la plataforma Arduino Mega para el estacionamiento del Canal y Radio UCSG.
- Desarrollar una descripción del funcionamiento del sistema propuesto, destacando sus características.
- Realizar el diseño del diagrama de flujo y conexión del Sistema de Control Inteligente con Arduino Mega.

1.6 Hipótesis

La investigación contempla la formulación de un Sistema de Control Inteligente para el Estacionamiento del Canal y Radio UCSG a través de la utilización de Arduino Mega. La hipótesis plantea que la introducción de este sistema conllevará a mejoras significativas en la gestión del estacionamiento. Se prevé que, al proveer información en tiempo real sobre la disponibilidad de espacios, el sistema logrará reducir la congestión, optimizar la asignación de lugares y enriquecer la experiencia global de estacionamiento para los usuarios. Asimismo, se anticipa que la utilización de tecnología inteligente, como Arduino Mega, incrementará la eficiencia en la toma de decisiones, aportando a una gestión más eficaz de los recursos disponibles en el ámbito del Canal y Radio UCSG.

1.7 Metodología de Investigación

- Revisión Bibliográfica: Se llevará a cabo una exhaustiva revisión de la literatura existente para recopilar información sobre sistemas de control de estacionamiento, tecnología Arduino, aplicaciones de sistemas inteligentes en entornos de estacionamiento y estudios previamente realizados de naturaleza similar.

- Análisis del Entorno de Estacionamiento: Se realizará un análisis minucioso del entorno de estacionamiento en el Canal y Radio UCSG con el fin de comprender las necesidades, patrones de uso, problemáticas actuales y las demandas de los usuarios en esta área.
- Diseño del Sistema de Control Inteligente: A partir de los resultados que se obtuvieron de la investigación anterior, se dará el diseño de un sistema de control inteligente que se apoye en la plataforma Arduino Mega.

Capítulo 2: Fundamentación Teórica

2.1 Arduino Mega 2560

El Arduino Mega 2560 como se observa en la figura 2.1, es un integrante importante de la gran familia Arduino, se diferencia por su complejo conjunto de características que lo ubican como una potente herramienta, sobre todo pensado para trabajos de gran magnitud. En el centro de esta superficie está el microcontrolador ATmega2560, que funciona como el cerebro principal que determina las habilidades del Mega 2560. La importancia de este microcontrolador se hace notar al tomar en consideración que la totalidad de los pines del diseño Arduino Mega están vinculados a él, esto da la posibilidad de tener una interacción amplia y dinámica con distintos sensores, actuadores y otros complementos (Setyo, 2021).

Figura 2.1: Arduino Mega 2560 variante R3

Fuente: (Tecmikro, 2024)

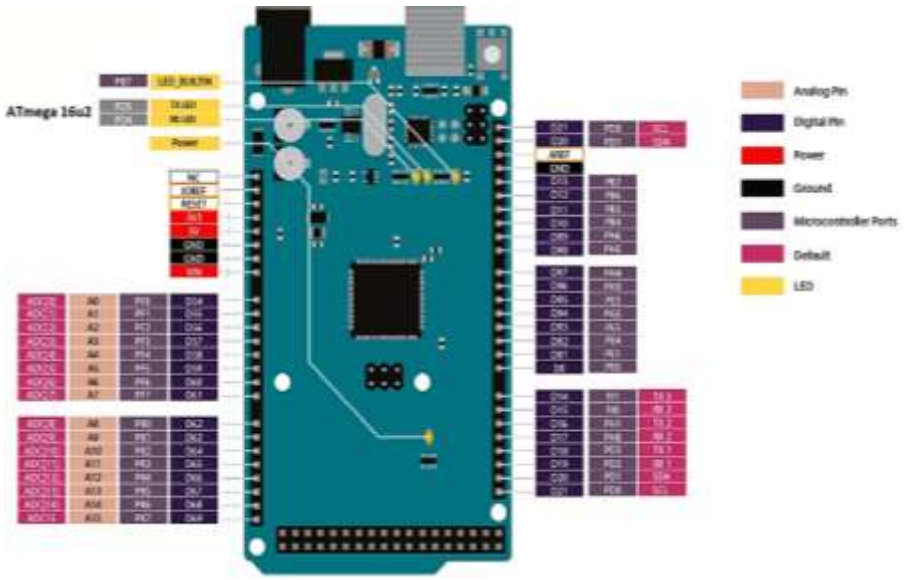
Sobre todo, el modelo Mega 2560 tiene una repartición equitativa de pines, más grande que la competencia más económica, el Arduino Uno. Esta sobreabundancia de pinos apoya el desarrollo de proyectos que requieren de múltiples

vínculos, otorgando a los artistas y constructores una mayor amplitud y versatilidad en sus creaciones. La conexión expandida, además de la consistencia de códigos entre el Arduino Uno y el Mega 2560, señala la amplitud de la versatilidad y la facilidad de uso de la placa, sobre todo para aquellos que realizan transiciones o utilizan simultáneamente las dos plataformas (Barrett, 2020).

Las particularidades identificables del Arduino Mega 2560, como su robusto microcontrolador ATmega2560 y su extensa disposición de pines, lo vuelven a elegir como la preferida de los entusiastas además de los profesionales que desean una placa de creación que fuera capaz de atender y sobrepasar las necesidades de los aplicativos más complicados con comodidad y fluidez (Sharath et al., 2021).

2.1.1 Arduino Mega 2560 la creación de prototipos

Figura 2.2: Pines analógicos, digitales de entrada/salida y de control del Arduino Mega 2560 variante R3



Fuente: (Watson, 2021)

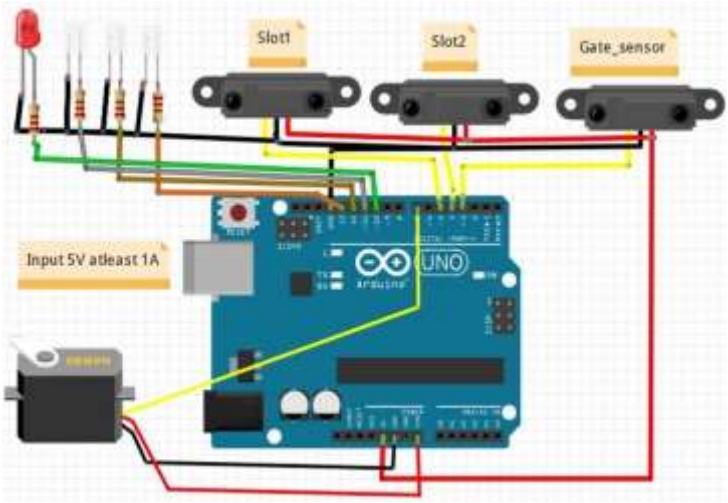
Dentro de la intrincada estructura de un sistema de estacionamiento automatizado, el Arduino Mega 2560 se constituye como un elemento fundamental que reduce significativamente la complejidad del proceso de creación, debido a su sobresaliente capacidad y armonía con varios componentes. Los 54 pines de entrada/salida en digital de la placa como se observa en la figura 2.2, que poseen 15 que posibilitan la modulación en la amplitud de pulsos (PWM), proveen una interconexión extensa, esto hace posible la combinación con otros sensores de desplazamiento y otros periféricos, que genera una manera más simple y eficaz de concebir diseños versátiles y altamente reactivos (Septian, 2022).

A partir de ser solo una cantidad masiva de pinos, el extenso espacio de memoria del Mega 2560 tiene un rol importante ya que posibilita una planificación más avanzada y el alojamiento de información, aspecto que es particularmente provechoso para gestionar los complicados flujos de datos que tienen. los sistemas de estacionamiento automáticos. También, la compatibilidad del Arduino Mega 2560 con el Arduino IDE, un espacio de programación sencillo de utilizar acelera la labor de creación. En la ocasión de proveer una superficie que se acopla sin dificultades con el Arduino IDE, los programadores tienen la posibilidad de escribir, probar y ejecutar sus códigos sin igual de manera, disminuyendo significativamente el lapso de elaboración y haciendo que el concepto se muestre como un prototipo funcional La mezcla de posibilidades amplias de entrada y salida, memoria mejorada y el soporte del software hace que el Arduino Mega 2560 sea una utilidad fundamental para la creación de sistemas complejos, como es el caso de los sistemas de estacionamiento inteligente (Alhassan, 2022).

2.2 Arduino Mega en los sistemas de aparcamiento inteligentes

En el ámbito de los sistemas de estacionamiento inteligentes, la placa Arduino Mega sirve como un componente crucial que sustenta la funcionalidad de toda la configuración. Utilizando sus considerables capacidades de procesamiento, Arduino Mega tiene la tarea de leer datos de cada sensor, que son vitales para el monitoreo en tiempo real y la gestión del espacio del sistema como se observa en la figura 2.3. Estos sensores, a menudo de naturaleza ultrasónica, están conectados al tablero de control que facilita su interacción con el Arduino Mega (Allbadi et al., 2021).

Figura 2.3: Sistema de parqueo inteligente con sensores infrarrojos con Arduino



Fuente: (Ecuarobot, 2020)

Además, la integración de un Escudo Ethernet es esencial, ya que permite una comunicación perfecta entre Arduino Mega y otros dispositivos en red, asegurando que la información sobre la disponibilidad de estacionamiento se pueda transmitir de manera efectiva. Esta sinergia entre Arduino Mega, los sensores ultrasónicos y Ethernet Shield ejemplifica el papel fundamental de la placa en la orquestación de los mecanismos de control del sistema de estacionamiento inteligente (Patro et al., 2020).

2.2.1 Diferencias entre los sistemas de aparcamiento inteligentes de los aparcamientos tradicionales

En una transformación importante con respecto a las maneras tradicionales de estacionar, los sistemas de estacionar inteligentes poseen información y tecnología para mejorar la vivencia de estar parqueado como se observa en la figura 2.4. A diferencia de los métodos tradicionales, que generalmente se basan en manuales de procedimientos específicos, el punto de vista inteligente utiliza una amplia gama de datos en tiempo real. Este punto de vista revolucionario la manera en la que los planificadores de la ciudad y los conductores toman las decisiones, ya que, gracias a la recolección de datos exhaustiva, es analizar posible las costumbres de los usuarios, las horas punta y las preferencias del lugar (Dogaroglu et al., 2021).

Figura 2.4: Implementación de parqueo inteligente en ciudades de China

Fuente: (Cnne, 2020)

La capacidad es el eje principal del sistema de estacionamiento inteligente, ya que posibilita la administración y la optimización de los recursos de manera dinámica. La habilidad de cambiar las tarifas y gestionar la movilidad en tiempo real evidencia una singularidad en la capacidad de respuesta a la demanda y la oferta de lugares para

estacionar, esto reduce la congestión y aumenta la utilidad del sistema de transporte. La comodidad del usuario final es igual de importante. La comodidad de encontrar lugares de estacionamiento libres, además de la posibilidad de hacer una reserva de un espacio con anterioridad a llegar, reduce significativamente la cantidad de tiempo que se tiene que buscar un espacio de estacionamiento, disminuyendo una afección común en las zonas urbanizadas (Hanzl, 2020).

La sencillez y transparencia de las soluciones de estacionamiento inteligentes resaltan la esencia de la usabilidad del usuario de estas, haciendo una diferencia clara con los procedimientos que a menudo son oscuros y complicados relacionados con el estacionamiento tradicional. Este punto de vista adelantada no solo aumenta la efectividad, sino que también redefine la sensación de estacionamiento, brindando una percepción más clara y beneficiosa para los que están involucrados en la actividad (Khalid et al., 2021).

2.3 Problemas comunes que se enfrentan en la gestión de estacionamientos tradicionales

La administración tradicional de lugares de estacionamiento es a menudo perjudicada por ineficacias que tienen una importancia grande para la movilidad y la sustentabilidad en las zonas metropolitanas como se observa en la figura 2.5. Uno de los problemas frecuentes es la administración deficiente del estacionamiento, que es una parte fundamental de los sistemas de transporte de las ciudades. Debido a métodos de operación antiguos, diversas poblaciones tienen dificultades para encontrar un espacio en las zonas de espera de los automóviles, esto genera congestión y deterioro del ecosistema. Investigaciones que se realizan en distintos centros de estudio de la

ciudad de manera indica que el sistema de estacionamiento actual no cumple con las necesidades modernas y no incorpora los últimos avances en tecnología con el fin de mejorar la administración y operación del estacionamiento (Parmar et al., 2020).

Figura 2.5: Parqueo tradicional no inteligente

Fuente: (Came, 2023)

Además, la deficiencia de una buena integración entre las zonas de estacionamiento y otras redes de transporte puede provocar una movilidad desorganizada, que afectará la fluidez del tráfico y generará demoras. Un inconveniente adicional reconocido en zonas es la oposición para persuadir a los involucrados a ejecutar nuevos métodos y estrategias de conducción que prefieren la confiabilidad y la eficiencia sobre el empleo cotidiano de automóviles particulares. En conjunto, estos problemas resaltan la importancia fundamental de transformaciones en la administración del estacionamiento que puedan atender las dificultades actuales de la movilidad en las ciudades. El contexto en cuestión señala la importancia de tomar enfoques novedosos que concuerdan con las necesidades actuales y promuevan una administración de automóviles más económica y sustentable (Diène et al., 2020).

2.3.1 Preocupaciones ambientales asociadas con los estacionamientos mal administrados

La mala administración de los estacionamientos no solo resta valor a las características innatas de los sistemas de estacionamiento electrónicos, sino que además genera una serie de inquietudes ambientales que tienen la posibilidad de afectar negativamente las zonas urbanas como se observa en la figura 2.6. Cuando la estructura de organización del estacionamiento no es buena, es posible que genere un uso ineficaz del espacio y contribuya a la congestión, ya que los conductores transitan por los lugares de estacionamiento en búsqueda de lugares libres. Esta corriente no sólo incrementa la cantidad de vehículos que están inactivos, sino que además acelera el desgaste de los propios lugares de estacionamiento, esto genera la necesidad de reparaciones y repavimentación más asiduamente. Este período, por su parte, implica un incremento de consumo de recursos y la generación de desechos de obras de construcción (Cotrone, 2022).

Figura 2.6: Contaminación de parqueaderos

Fuente: (Dickie, 2013)

Además, la falta de instalaciones higiénicas adecuadas en estas zonas es posible que genere acumulación de contaminación, ya que la corriente proveniente de

las zonas de espera en los aeropuertos suele contener sustancias como el petróleo, los metales pesados y otros elementos que son dañinos para el medio ambiente y las fuentes de agua cercanas. Este contexto reitera la importancia de manejar un enfoque reflexivo acerca de la administración de lugares de estacionamiento, incorporando consideraciones acerca de la sustentabilidad y de los objetivos ambientales como prioritarios. La ejecución y planificación de estrategias acerca del uso del espacio, además de mejorar la eficiencia de la utilización de este, también puede mitigar los efectos dañinos sobre el ecosistema, esto es, puede generar una ciudad más equitativa y sustentable (Bannon, 2023).

2.4 Beneficios operativos de adoptar sistemas de estacionamiento inteligentes

Los provechos operativos originados por la utilización de sistemas de estacionamiento automatizados son muchos y tienen una influencia directa en la eficiencia y en la perdurabilidad de los negocios. En particular, resaltan los provechos de la optimización del espacio, donde los sistemas de estacionamiento inteligentes utilizan información en tiempo real para que cada centímetros que se utilizan sean realmente efectivos. Este punto de vista maximiza la capacidad disponible y, por ende, tiene la capacidad de aumentar los ingresos de los proveedores. De las cuestiones económicas, la utilización eficaz del espacio tiene una importante influencia sobre la administración de la movilidad en las ciudades (Espinoza, 2022).

En la medida en la que los conductores obtienen información precisa acerca de la existencia de lugares de estacionamiento, los sistemas de inteligencia ayudan a acortar la cantidad de tiempo que los automóviles pasan en búsqueda de un espacio para estacionarse. Esta reducción tiene una influencia directa en la congestión del

tráfico, lo que aumenta la fluidez del tráfico en las zonas urbanas. También, es imposible pasar por alto la utilidad del punto de vista ambiental de esta perspectiva. La reducción del número de automóviles que transitan en búsqueda de un lugar para estacionarse implica una notable reducción de las emisiones, esto ayuda a que la ciudad sea más limpias y sustentables (Saharan et al., 2020).

Figura 2.7: Tipos de sistemas de registro para parqueaderos inteligentes

Fuente: (Cogniteq, 2023)

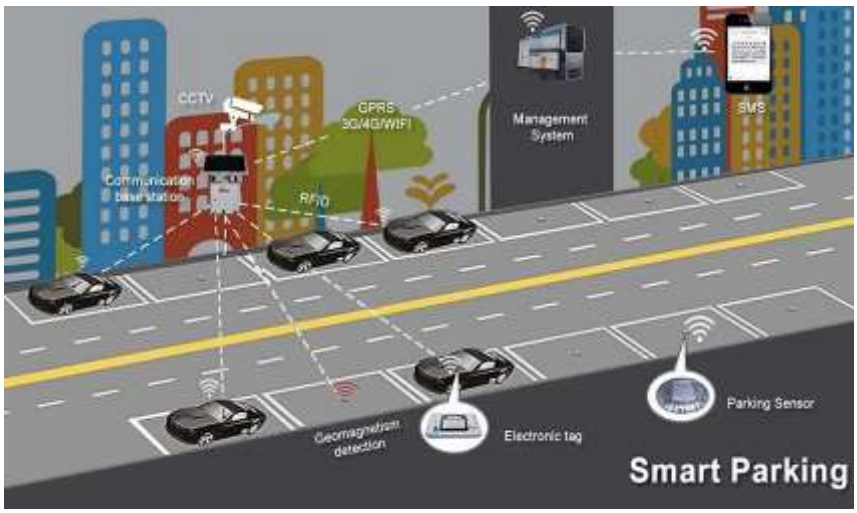
Estas bondades operativas, fundadas en la capacidad de la tecnología avanzada para economizar la administración del estacionamiento y mejorar la vivencia del usuario como se observa en la figura 2.2, se resalta el potencial revolucionario de los sistemas de administración del estacionamiento dentro del urbanismo. El ejecutar estrategias para estos arreglos no solo aumenta la eficiencia financiera y operante, sino que además tiene un efecto beneficioso sobre la calidad de vida de las personas y el medio ambiente a largo plazo (Lom & Pribyl, 2021).

2.4.1 Impactos económicos a largo plazo de los estacionamientos inteligentes

Los provechos operativos originados por la utilización de sistemas de estacionamiento automatizados son muchos y tienen una influencia directa en la eficiencia y en la perdurabilidad de los negocios como se observa en la figura 2.8. En

particular, resaltan los provechos de la optimización del espacio, donde los sistemas de estacionamiento inteligentes utilizan información en tiempo real para que cada centímetros que se utilizan sean realmente efectivos. Este punto de vista maximiza la capacidad disponible y, por ende, tiene la capacidad de aumentar los ingresos de los proveedores. De las cuestiones económicas, la utilización eficaz del espacio tiene una importante influencia sobre la administración de la movilidad en las ciudades (Liu et al., 2020).

Figura 2.8: Ejemplo de estacionamiento inteligentes en ciudades



Fuente:(Argüello, 2021)

En la medida en la que los conductores obtienen información precisa acerca de la existencia de lugares de estacionamiento, los sistemas de inteligencia ayudan a acortar la cantidad de tiempo que los automóviles pasan en búsqueda de un espacio para estacionarse. Esta reducción tiene una influencia directa en la congestión del tráfico, lo que aumenta la fluidez del tráfico en las zonas urbanas. También, es imposible pasar por alto la utilidad del punto de vista ambiental de esta perspectiva. La reducción del número de automóviles que transitan en búsqueda de un lugar para

estacionarse implica una notable reducción de las emisiones, esto ayuda a que la ciudad sea más limpias y sustentables (Intercomp, 2020).

Estas bondades operativas, fundadas en la capacidad de la tecnología avanzada para economizar la administración del estacionamiento y mejorar la vivencia del usuario, resaltan el potencial revolucionario de los sistemas de administración del estacionamiento dentro del urbanismo. El ejecutar estrategias para estos arreglos no solo aumenta la eficiencia financiera y operante, sino que además tiene un efecto beneficioso sobre la calidad de vida de las personas y el medio ambiente a largo plazo (Krishnamurthy & Ngo, 2020).

2.5 Demanda creciente de mejores soluciones de estacionamiento

A medida que las ciudades en desarrollo crecen significativamente y el parque de vehículos aumenta considerablemente, la necesidad de soluciones adecuadas en términos de estacionamiento se vuelve cada vez más clara. La administración y dirección responsable de las áreas de espera de vehículos públicos se encuentra entre las piezas fundamentales para la perpetuidad del trabajo de la ciudad y para la comodidad del usuario. No basta con ampliar el número de plazas de aparcamiento para vehículos; La verdadera esencia se encuentra en la instalación de métodos que no sólo generen alta eficiencia, sino que también brinden beneficios materiales a los usuarios. La utilidad y fiabilidad de la infraestructura de rotación de vehículos es fundamental en esta circunstancia. La falta de una adecuada administración y seguimiento genera una mayor probabilidad de daños y fallas económicas, además de penalizar a los usuarios por estar ubicados en lugares equivocados. Este contexto muestra la importancia de soluciones de aparcamiento que no sólo estén

cuidadosamente planificadas, sino también flexibles y puedan adaptarse a las diversas necesidades de los usuarios (Quercus, 2023).

La incorporación de tecnología en los sistemas de gestión del aparcamiento puede facilitar un control más específico y también ser útil para afrontar la falta de control que forma parte de diversas formas de gestión del aparcamiento o parking en zonas urbanas. Esta incorporación de tecnología no sólo apoya la gestión eficaz de los espacios libres, sino que también ayuda a mejorar la experiencia del usuario al proporcionar información en tiempo real sobre la existencia de los espacios, para que los usuarios puedan tomar decisiones informadas. En consecuencia, la confluencia de la tecnología con la gestión del automóvil no es sólo un avance, sino que también es una necesidad fundamental para asegurar la eficiencia y precisión en la gestión de estos importantes terrenos (Golan, 2023).

2.5.1 Tecnologías clave que darán forma al futuro del estacionamiento

Está en marcha una transformación importante en la forma en que paramos, liderada por el uso de nuevas herramientas inteligentes que tienen como objetivo aumentar la comodidad del conductor y también aumentar la eficacia de los proveedores de estacionamiento. servicios de espera. En este contexto, el desarrollo de herramientas complejas, como sensores más grandes y análisis de datos, juega un papel importante en la utilización del espacio de estacionamiento y en la prevención de atascos. La incorporación de estos sensores proporciona información en tiempo real sobre la existencia de plazas libres, esto apoya a los conductores en la búsqueda de plazas que se encuentran libres, además, el análisis de datos puede predecir las

temporadas de mayor asistencia, alterando tarifas o disponibilidad. Correspondientemente (Buitleir, 2023).

Asimismo, el uso de sistemas de pago automatizados y tecnología para la identificación de patentes no solo genera comodidad para el usuario, sino que también reduce significativamente las filas de espera en las terminales de control, esto ayuda a aumentar la satisfacción del cliente. Esta combinación de métodos representa un posible prototipo de transformación constante en la gestión del estacionamiento, demostrando una comunidad donde la efectividad, la conveniencia y la satisfacción se encuentran de manera sinérgica. En esta nueva era de la gestión de automóviles, el vínculo entre estas herramientas se extiende más allá de la simple modernidad: establece un límite que separa lo que pueden hacer los conductores de lo que pueden hacer los aparcadores. La capacidad de adivinación del análisis de información no sólo nos permite adivinar cifras de uso, sino también cambiar disponibilidad y costos de forma dinámica, esto genera una experiencia más cómoda y única para los usuarios (Sabra, 2023).

2.5.2 Tecnologías de sensores en la gestión del aparcamiento

El empleo de herramientas que brindan sentimiento ha sido propuesto como una solución fundamental para solucionar las dificultades que presentan los actuales sistemas de administración del estacionamiento en las zonas urbanizadas. Los sensores inteligentes tienen un rol significativo dentro de la administración de los propietarios de las plazas de aparcamiento, ya que dan información en tiempo real de la existencia de estas, esto genera una enorme reducción en la utilización de las escasas herramientas destinadas a la administración del estacionamiento. Estos artefactos

funcionan haciendo un registro de la entrada y salida de los automóviles, brindando una estimación estática de la asistencia en una determinada época del año (Shroud et al., 2023).

Figura 2.9: Sensores en estacionamientos

Fuente: (Mapfre, 2020)

La recolección de información a través de estos medidores no sólo ofrece datos en tiempo real sobre la condición de las entradas y las salidas, sino que además tiene una grande influencia en la correcta administración de las zonas de espera de las terminales de transporte. A medida en que disminuyen la cantidad de desplazamiento no necesario de los automóviles y disminuyen las colisiones dentro del lugar de estacionamiento, estos sensores se vuelven en componentes fundamentales para mejorar la fluidez del tráfico y la vivencia del conductor (Novable, 2023).

También, la incorporación de sensores singulares como son los ultrasonido, el radar y las imágenes dentro de los sistemas de administración de estacionamiento genera la habilidad de ejecutar acciones adelantadas como la administración de estacionamiento asistida y también la administración de estacionamiento inteligente

como se observa en la figura 2.9. Estas asistencias posibilitan a los conductores desplazarse y orientarse en lugares en donde no existen complicaciones, disminuyendo la cantidad de tiempo necesario para hallar un espacio de estacionamiento y mejorando la fiabilidad del procedimiento en su conjunto. La fuerza del valor conocido no sólo economiza el tiempo de los conductores, sino que además tiene un efecto positivo en la disminución del consumo de aceite y de las partículas. Gracias a la tecnología mencionada, es más sencillo hallar y elegir lugares de estacionamiento, esto tiene una importancia significantemente en el desarrollo de una ciudad más verde y sustentable (Patel, 2021).

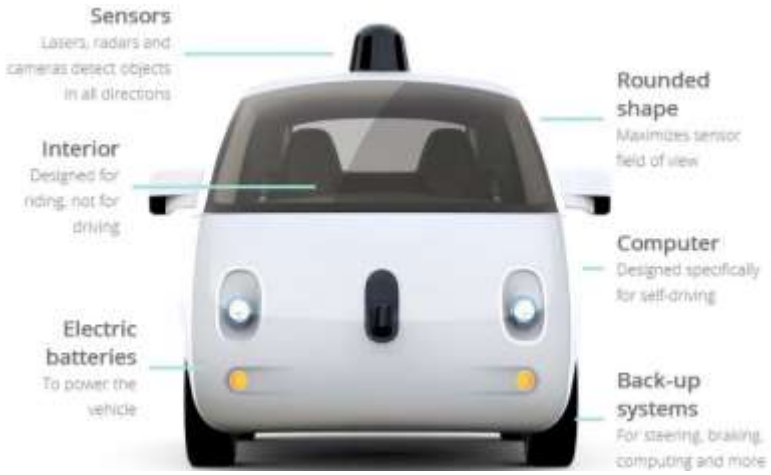
2.5.3 Vehículos autónomos a los futuros sistemas de aparcamiento

Ante los problemas inherentes a los sistemas de aparcamiento, los urbanistas y legisladores se encuentran ante una oportunidad inmejorable con la llegada de los automóviles automatizados (AV). La incorporación de estos coches a los dispositivos de transporte urbano podría generar una importante transformación y una nueva concepción de los métodos de aparcamiento, aportando posibilidades de optimización del espacio que podrían traducirse en una reducción considerable del número de plazas de aparcamiento de vehículos necesarias. Este novedoso punto de vista permitiría a las ciudades redirigir espacios urbanos de alto valor hacia usos más rentables, como áreas verdes o centros comerciales, siempre que vaya acompañado de objetivos de desarrollo económico y social sostenible (Parker, 2023).

El paradigma del cambio propuesto en el diseño del aparcamiento no es sólo una cuestión de abordar las dificultades actuales, sino que también pretende coincidir con los objetivos de progreso más importantes. Los coches sin conductor como se

observa en la figura 2.10, debido a su precisión y conectividad, tienen la capacidad de revolucionar la forma en que se utiliza el espacio de aparcamiento existente. Esto se lograría reduciendo aún más el espacio y eliminando la necesidad de que los usuarios se muevan físicamente hacia y desde los automóviles. Esta transformación en la forma de pensar sobre el estacionamiento en las ciudades tiene la capacidad de permitir que las ciudades busquen el mayor uso de sus recursos naturales y respondan de manera más efectiva a las necesidades cambiantes de sus habitantes (Kaur & Garg, 2023).

Figura 2.10: Carros inteligentes con parqueo inteligente automático.



Fuente: (Dans, 2020)

Es esencial mencionar que las siguientes investigaciones sobre la mejora de estas tecnologías de autoayuda pueden potencialmente disminuir la cantidad de espacio de estacionamiento en áreas urbanizadas. Este aumento liberaría la entrada a una zona más sostenible y fértil de la ciudad, donde la movilidad se integra correctamente con el entorno construido. En última instancia, la confluencia de tecnología independiente y planificación urbana podría generar una nueva era en la gestión del estacionamiento, superando los límites actuales y promoviendo un urbanismo más inteligente y sostenible (Jo et al., 2023).

Capítulo 3: Aportes de la investigación

El presente capitulo se basa en el estudio y diseño de un sistema de control inteligente para el estacionamiento del Canal y Radio UCSG, en la figura 3.1 se observa la ubicación en Google Earth del estacionamiento mencionado. Conforme a la necesidad de automatizar las diversas etapas del sistema actual de parqueo que posee el Canal y Radio de la Universidad.

Figura 3.1: Estacionamiento del Canal y Radio UCSG - Google Earth

Elaborada por: Autor

El área de estacionamiento ubicada en las instalaciones del Canal y Radio UCSG, tal y como se muestra en la figura 3.1, está compuesta por cinco filas designadas para la ubicación de vehículos. De estas cinco filas, dos han sido específicamente asignadas para facilitar la entrada y salida de los vehículos, mientras que las tres filas restantes cuentan con un espacio predeterminado para albergar hasta diez vehículos cada una. Esto proporciona un total de espacio para treinta vehículos dentro del área de estacionamiento. En vista de esta capacidad definida, es imperativo

implementar un sistema de control que permita monitorear la disponibilidad de espacio para los treinta vehículos en el estacionamiento, asegurando así una gestión eficiente y ordenada de este.

3.1 Descripción del funcionamiento del sistema propuesto

A continuación, se presenta una descripción exhaustiva de los casos de uso cuando el sistema detecta y no detecta un vehículo, también se presenta el diagrama de flujo de la figura 3.2:

Figura 3.2: Diagrama de flujo del funcionamiento del dispositivo

Elaborada por: Autor

Caso de Uso: Entrada de un Vehículo

Actores:

- Vehículo

Flujo Principal:

1) El vehículo ingresa a la zona de detección.

2) El sistema activa los sensores infrarrojos pertinentes.
3) El sistema realiza la lectura del estado de los sensores.
4) En caso de detectarse un vehículo (al menos un sensor activado):
    - El sistema actualiza su estado interno.
    - El sistema activa el actuador correspondiente para cerrar la barrera.
    - La información actualizada se refleja en la pantalla LCD.

Flujo Alternativo:

1) Si la barrera ya se encuentra cerrada, no se lleva a cabo ninguna acción adicional.
2) Si la zona de detección ya está ocupada, el sistema no ejecuta ninguna acción adicional.

Postcondiciones:

1) El vehículo queda registrado como estacionado.
2) La barrera se encuentra cerrada.
3) La pantalla LCD presenta la información actualizada.

Caso de Uso: Salida de un Vehículo

Actores:

- Vehículo

Flujo Principal:

1) El vehículo sale de la zona de detección.
2) El sistema activa los sensores infrarrojos correspondientes.
3) La lectura del estado de los sensores es realizada por el sistema.
4) En ausencia de la detección de algún vehículo (todos los sensores desactivados):

- El sistema actualiza su estado interno.
- El sistema activa el actuador correspondiente para abrir la barrera.
- La información actualizada se refleja en la pantalla LCD.

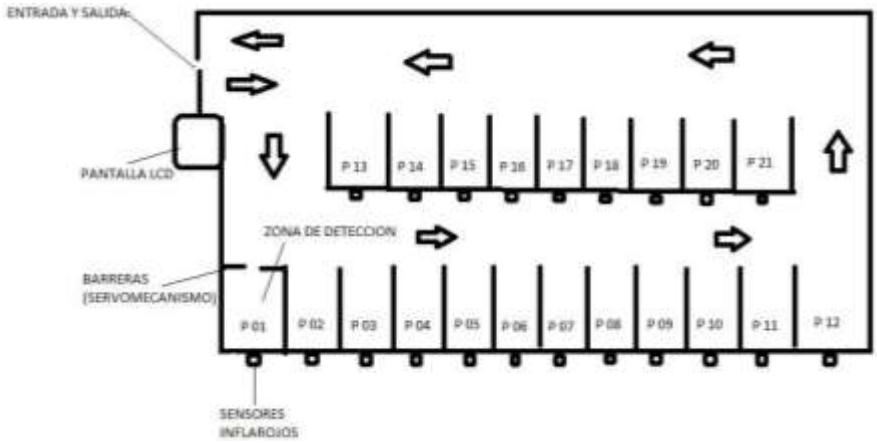
Flujo Alternativo:

1) Si la barrera ya está abierta, no se lleva a cabo ninguna acción adicional.
2) Si la zona de detección ya está desocupada, el sistema no ejecuta ninguna acción adicional.

Postcondiciones:

1) El vehículo queda registrado como fuera de la zona de detección.
2) La barrera se encuentra abierta.
3) La pantalla LCD muestra la información actualizada.

Figura 3.3: Diagrama pictórico en 2d del estacionamiento



Elaborada por: Autor

3.2 Descripción de los componentes del sistema propuesto

A continuación, se presentan los elementos que conforma el sistema de control inteligente para el estacionamiento del Canal y Radio UCSG.

Arduino Mega (Mega 2560):

- **Función:** Desempeña un papel crucial como el núcleo de control del sistema.
- **Descripción:** El Arduino Mega 2560, actuando como la unidad central de procesamiento, orquesta de manera integral todas las funciones del sistema. Es el receptor de datos provenientes de los sensores, tomador de decisiones basadas en dichos datos y conductor de las acciones de los actuadores correspondientes. Además, se encarga de la administración de la interfaz de la pantalla LCD, la cual presenta información pertinente acerca del estado actual del estacionamiento.

Sensores Infrarrojos (IR - TCRT5000):

- **Función:** Su propósito principal es detectar la presencia de vehículos en la zona de detección.
- **Descripción:** Los sensores infrarrojos TCRT5000 emiten luz infrarroja y cuantifican la cantidad reflejada. Cuando un vehículo penetra la zona de detección, intercepta parte de la luz reflejada, generando una señal analógica que es captada por los sensores. Esta información se transmite al Arduino para su posterior procesamiento.

Multiplexor para Sensores (CD74HC4067):

- **Función:** Simplifica el proceso de multiplexación de señales provenientes de los sensores IR.
- **Descripción:** El multiplexor CD74HC4067 permite al Arduino seleccionar y leer datos de varios sensores infrarrojos de forma eficiente. Controla los

canales de los sensores y dirige la señal al Arduino, reduciendo la complejidad del cableado y mejorando la capacidad de expansión del sistema.

Actuadores (Servomecanismos):

- **Función:** Tienen la responsabilidad de gestionar la apertura y cierre de las barreras o puertas automáticas.
- **Descripción:** Los servomecanismos, actuadores fundamentales en el sistema, reciben comandos específicos del Arduino para ejecutar las acciones de abrir y cerrar las barreras. Transforman las señales digitales en movimiento físico, permitiendo un control preciso sobre la posición de las barreras o puertas.

Multiplexor para Actuadores (CD74HC154E):

- **Función:** Encargado de llevar a cabo la multiplexación de señales provenientes de los actuadores.
- **Descripción:** Similar al multiplexor de sensores, el CD74HC154E gestiona de manera eficiente la conexión de múltiples actuadores al Arduino. Facilita la activación selectiva de cada actuador, simplificando la lógica de control y mejorando la modularidad del sistema.

Pantalla LCD (I2C 16x2 LCD Display):

- **Función:** Proporciona una interfaz visual para visualizar información acerca del estado del estacionamiento.
- **Descripción:** La pantalla LCD ofrece una representación gráfica de datos relevantes, como la cantidad de espacios de estacionamiento ocupados y

disponibles. El Arduino transmite datos a través del bus I2C para actualizar de manera dinámica la pantalla, reflejando así el estado actual del sistema.

Módulo I2C para LCD (PCF8574):

- **Función:** Facilita la comunicación I2C entre el Arduino y la pantalla LCD.
- **Descripción:** El módulo PCF8574 sirve como enlace crucial para la comunicación I2C entre el Arduino y la pantalla LCD. Simplifica la conexión entre estos elementos y reduce el número de pines requeridos, permitiendo una integración más eficiente entre el controlador central y la interfaz de visualización.

Tabla 3.1 *Elementos utilizados para la implementación del parqueadero inteligente*

Elemento	Modelo	Función	Tipo
Arduino Mega	Mega 2560	Control central del sistema	Digital
Sensores Infrarrojos (IR)	TCRT5000	Detección de presencia de vehículos	Analógico
Multiplexor para Sensores	CD74HC4067	Multiplexación de señales de sensores IR	Digital
Actuadores	Servomecanismos	Control de barreras o puertas automáticas	Digital
Multiplexor para Actuadores	CD74HC154E	Multiplexación de señales de actuadores	Digital

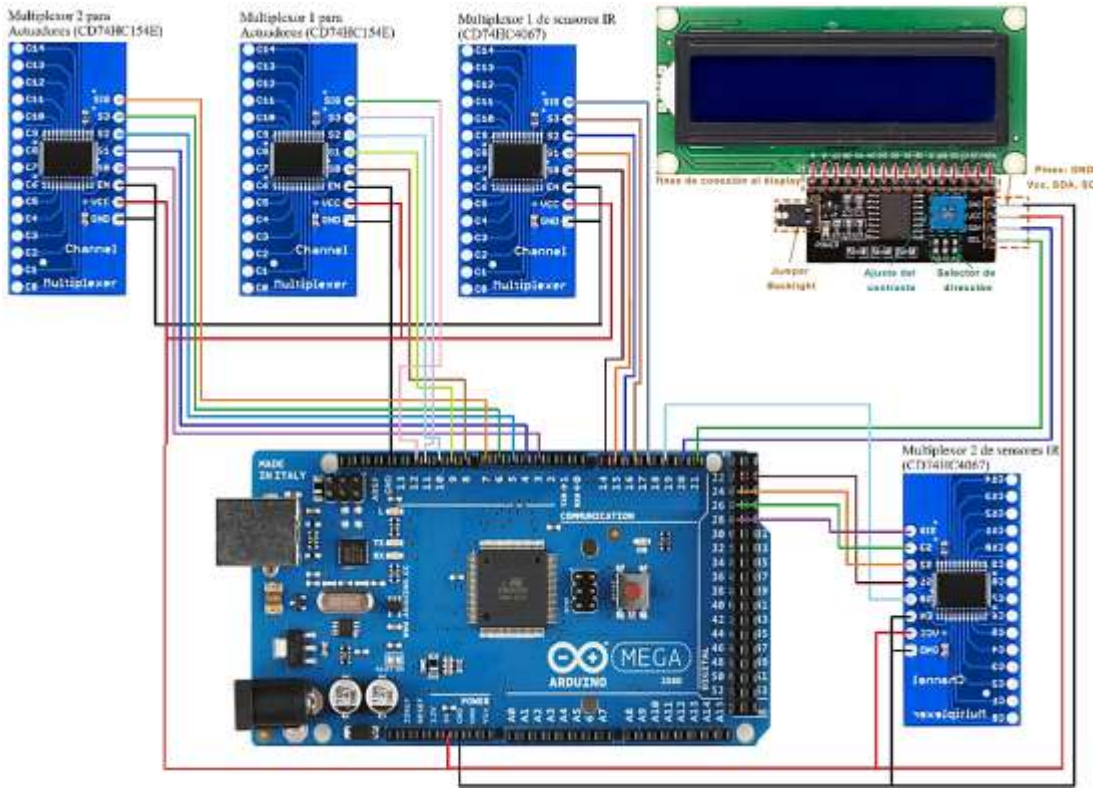
Pantalla LCD	I2C 16x2 LCD Display	Mostrar información sobre el estado del estacionamiento	Digital
Módulo I2C para LCD	PCF8574	Comunicación I2C para el LCD	Digital

Fuente: Autor

3.3 Conexiones de los elementos del sistema propuesto

Se detallan los voltajes de operación y las conexiones de los componentes y módulos empleados en el proyecto, tal como se muestra en la figura 3.4.

Figura 3.4: Diagrama de conexiones de los multiplexores y display con el Arduino Mega



Elaborada por: Autor

Arduino Mega (Mega 2560):

- Función: Cumple la función crucial como el núcleo de control del sistema.

- Descripción: El Arduino Mega 2560 desempeña un papel central en la gestión del sistema, operando a un voltaje nominal de 5V. Su conexión con la fuente de alimentación se realiza a través de un cable USB. La integración con otros componentes se logra mediante pines digitales y analógicos, asignados estratégicamente para controlar los multiplexores, actuadores y la comunicación con la pantalla LCD. Esta compleja coordinación permite una operación eficiente y sincronizada de todos los elementos del proyecto.

Sensores Infrarrojos (IR - TCRT5000):

- Función: Su misión principal es la detección de la presencia de vehículos en la zona de detección, como se observa en la figura 3.5.
- Descripción: Los sensores infrarrojos TCRT5000, con un requerimiento de voltaje de 5V, poseen tres pines fundamentales: VCC (5V), GND (Tierra), y OUT (salida analógica). La conexión a pines analógicos del Arduino permite la transmisión de información analógica precisa sobre la detección de vehículos, facilitando así el procesamiento de datos por parte del sistema.

Figura 3.5: Sensores Infrarrojos (IR - TCRT5000)

Fuente: (Piqoa, 2022)

Multiplexor para Sensores (CD74HC4067):

- Función: Simplifica el proceso de multiplexación de señales provenientes de los sensores IR.

- Descripción: El CD74HC4067, diseñado para la multiplexación de señales de los sensores IR, opera a un voltaje estándar de 5V. Su conexión a pines digitales del Arduino, especialmente diseñados para el control de selección de canales (S0-S3) y un pin Z para la salida de datos, permite una gestión eficiente y ordenada de las señales provenientes de los sensores distribuidos en la zona de detección.

Actuadores (Servomecanismos):

- Función: Encargados de gestionar la apertura y cierre de las barreras o puertas automáticas, como se observa en la figura 3.6.
- Descripción: Los servomecanismos, esenciales en el sistema, operan a un voltaje estándar de 5V y se conectan a pines digitales del Arduino. La manipulación precisa de la posición de las barreras se logra mediante el envío de pulsos a través de un solo cable de señal, controlado por el núcleo central del sistema.

Figura 3.6: Servomecanismo MG995

Fuente: (Diosdado, 2019)

Multiplexor para Actuadores (CD74HC154E):

- Función: Realiza la multiplexación de señales provenientes de los actuadores.

- Descripción: Similar al multiplexor de sensores, el CD74HC154E funciona a un voltaje de 5V y se conecta a pines digitales del Arduino. Sus funciones incluyen el control de selección de canales (S0-S3) y un pin Z para la salida de datos, contribuyendo así a una gestión eficiente y simplificada de las señales provenientes de los actuadores distribuidos en el sistema.

Pantalla LCD (I2C 16x2 LCD Display):

- Función: Presenta información relevante sobre el estado del estacionamiento.
- Descripción: La pantalla LCD opera a un voltaje estándar de 5V y establece comunicación con el Arduino mediante el protocolo I2C. Su conexión se realiza a través de un módulo I2C (PCF8574), que simplifica las conexiones y utiliza únicamente dos pines del Arduino (SDA y SCL). Este diseño eficiente facilita la actualización dinámica de la pantalla según el estado actualizado del sistema.

Módulo I2C para LCD (PCF8574):

- Función: Facilita la comunicación I2C entre el Arduino y la pantalla LCD.
- Descripción: El módulo PCF8574, operando a un voltaje estándar de 5V, se conecta tanto al LCD como al Arduino, facilitando la comunicación I2C entre ambos componentes. Con tan solo dos pines (SDA y SCL) para la transmisión de datos entre el Arduino y la pantalla LCD, este módulo actúa como un enlace crucial para garantizar una integración eficiente y ordenada entre el controlador central y la interfaz de visualización.

Tabla 3.2 *Voltajes y corrientes de funcionamiento para el dispositivo*

Elemento	Voltaje de Funcionamiento	Consumo de Corriente Estimado
Arduino Mega	5V	200 mA
Sensores Infrarrojos (IR)	5V	20 mA (por sensor)
Multiplexor para Sensores	5V	10 mA
Actuadores	5V	150 mA (por actuador)
Multiplexor para Actuadores	5V	15 mA
Pantalla LCD	5V	100 mA
Módulo I2C para LCD	5V	5 mA

Fuente: Autor

La tabla 3.3 se proporciona un listado completo de los componentes que forman parte del sistema, indicando claramente cómo se conectan a los pines específicos del Arduino Mega.

Tabla 3.3 *Tipo y gestión de pines para los elementos en el Arduino Mega*

Elemento	Tipo	Pin (Arduino Mega)
Módulo I2C y LCD	Digital	SDA, SCL
Multiplexor 1 para Actuadores (CD74HC154E)	Digital	S0-S3: 8, 9, 10, 11, Z: 12
Multiplexor 1 de sensores IR (CD74HC4067)	Digital	S0-S3: 14, 15, 16, 17, Z: 18
Multiplexor 2 para Actuadores (CD74HC154E)	Digital	S0-S3: 3, 4, 5, 6, Z: 7
Multiplexor 2 de sensores IR (CD74HC4067)	Digital	S0-S3: 19, 22, 24, 26, Z: 28

Fuente: Autor

A continuación, se proporciona una tabla de conexión para los sensores y actuadores conectados a los multiplexores. La tabla 3.4 ofrece un detallado registro

sobre la conexión de los sensores mediante el primer multiplexor. Cada sensor se encuentra asignado a un canal específico dentro del CD74HC4067, siendo proporcionado el respectivo pin de conexión con el Arduino Mega. Los sensores, cuyos números oscilan entre 1 y 15, están asociados a canales numerados del 0 al 14, y los pines de conexión se extienden desde A0 hasta A14.

En similitud con la tabla 3.4, la tabla 3.5 presenta de manera explícita la disposición de los sensores, si bien en este caso, se realiza a través del segundo multiplexor. Los sensores numerados entre 16 y 30 están asignados a canales que van del 0 al 14, y los pines de conexión con el Arduino Mega varían desde A0 hasta A14.

Tabla 3.4 *Pines utilizados para los sensores en el Multiplexor 1 (CD74HC4067)*

Multiplexor Sensores 1 (CD74HC4067)		
Sensor	**Canal**	**Pin de Conexión**
1	0	A0
2	1	A1
3	2	A2
4	3	A3
5	4	A4
6	5	A5
7	6	A6
8	7	A7
9	8	A8
10	9	A9
11	10	A10
12	11	A11
13	12	A12
14	13	A13
15	14	A14

Fuente: Autor

Tabla 3.5 *Pines utilizados para los sensores en el Multiplexor 2 (CD74HC4067)*

Multiplexor Sensores 2 (CD74HC4067)		
Sensor	**Canal**	**Pin de Conexión**
16	0	A0
17	1	A1
18	2	A2
19	3	A3
20	4	A4
21	5	A5
22	6	A6
23	7	A7
24	8	A8
25	9	A9
26	10	A10
27	11	A11
28	12	A12
29	13	A13
30	14	A14

Fuente: Autor

La información presentada en la tabla 3.6 proporciona una especificación exhaustiva acerca de la conexión de los actuadores a través del primer multiplexor de actuadores. En este contexto, cada actuador ha sido asignado de manera exclusiva a un canal específico del CD74HC154E, y se ha indicado el pin correspondiente para establecer la conexión con el Arduino Mega.

La información presentada en la tabla 3.6 proporciona una especificación exhaustiva acerca de la conexión de los actuadores a través del primer multiplexor de actuadores. En este contexto, cada actuador ha sido asignado de manera exclusiva a

un canal específico del CD74HC154E, y se ha indicado el pin correspondiente para establecer la conexión con el Arduino Mega. La asignación específica de los actuadores, numerados del 1 al 15, a los canales del 0 al 14, permite una organización clara y eficiente de los componentes. Además, los pines de conexión se encuentran designados de manera precisa como Y0 hasta Y14, lo que facilita aún más la comprensión y configuración del sistema.

Tabla 3.6 *Pines utilizados para los actuadores en el Multiplexor 1 (CD74HC154E)*

Multiplexor Actuadores 1 (CD74HC154E)		
Sensor	**Canal**	**Pin de Conexión**
1	0	Y0
2	1	Y1
3	2	Y2
4	3	Y3
5	4	Y4
6	5	Y5
7	6	Y6
8	7	Y7
9	8	Y8
10	9	Y9
11	10	Y10
12	11	Y11
13	12	Y12
14	13	Y13
15	14	Y14

Fuente: Autor

Continuando con la coherencia de la información proporcionada en la tabla 3.7, la tabla actual amplía y detalla la conexión de los actuadores, concentrándose especialmente en el segundo multiplexor de actuadores.

En este contexto, los actuadores identificados con números del 16 al 30 están cuidadosamente vinculados a canales que abarcan del 0 al 14 en el CD74HC154E. Además, se ha prestado atención meticulosa al detalle al detallar los pines de conexión con el Arduino Mega, los cuales están numerados desde Y0 hasta Y14.

Tabla 3.7 *Pines utilizados para los actuadores en el Multiplexor 2 (CD74HC154E)*

Multiplexor Actuadores 2 (CD74HC154E)		
Sensor	**Canal**	**Pin de Conexión**
16	0	Y0
17	1	Y1
18	2	Y2
19	3	Y3
20	4	Y4
21	5	Y5
22	6	Y6
23	7	Y7
24	8	Y8
25	9	Y9
26	10	Y10
27	11	Y11
28	12	Y12
29	13	Y13
30	14	Y14

Fuente: Autor

3.4 Programación en Arduino IDE del sistema propuesto

En la figura 3.7 se observa el código incluye bibliotecas esenciales para las comunicaciones I2C, el control de servomecanismos y la interfaz de visualización mediante un LCD. La sección de definiciones de pines establece los pines específicos

en el Arduino Mega que se utilizan para controlar los multiplexores y otros dispositivos asociados. El primer multiplexor (SENSOR_MUX1) se configura con pines digitales para controlar la selección del canal (S0-S3) y un pin Z para la salida de datos. El segundo multiplexor (SENSOR_MUX2) se configura de manera similar.

Es importante destacar que esta configuración proporciona una forma de seleccionar un canal particular en cada multiplexor, lo que permite la lectura secuencial de múltiples sensores IR conectados a través de esos canales. Cada sensor tiene un conjunto único de pines asociados con un canal específico en los multiplexores. El código presenta una estructura organizada que facilita la lectura y el mantenimiento. Este enfoque modular y bien documentado contribuye a la legibilidad y mantenimiento del código en proyectos más extensos.

Figura 3.7: Programación de librerías y multiplexores para los sensores del dispositivo

```
#include <Wire.h>
#include <Servo.h>
#include <LiquidCrystal_I2C.h>

// Definición de pines para el multiplexor 1 de sensores IR
#define SENSOR_MUX1_S0 14
#define SENSOR_MUX1_S1 15
#define SENSOR_MUX1_S2 16
#define SENSOR_MUX1_S3 17
#define SENSOR_MUX1_Z  18

// Definición de pines para el multiplexor 2 de sensores IR
#define SENSOR_MUX2_S0 19
#define SENSOR_MUX2_S1 22
#define SENSOR_MUX2_S2 24
#define SENSOR_MUX2_S3 26
#define SENSOR_MUX2_Z  28
```

Elaborada por: Autor

La utilización de estas definiciones de pines en el código permite la configuración y control eficiente de canales para la activación selectiva de actuadores. En el primer multiplexor (ACTUATOR_MUX1), los pines S0 a S3 determinan la

selección del canal, mientras que el pin Z actúa como salida de datos. Estos pines son esenciales para dirigir la señal a un canal específico y recibir la información correspondiente del actuador conectado. Similarmente, el segundo multiplexor (ACTUATOR_MUX2) se configura de manera análoga, pero con diferentes pines para garantizar un control independiente.

La función principal de estos multiplexores es permitir al Arduino seleccionar de manera precisa y secuencial diferentes actuadores conectados al sistema. La definición de pines establece la estructura necesaria para activar canales específicos, lo que a su vez controla la operación de los actuadores asociados. Este código se integra eficientemente en un sistema más amplio basado en Arduino, donde la capacidad de seleccionar y activar actuadores específicos es esencial.

Figura 3.8: Programación de los multiplexores para los actuadores del dispositivo

```
// Definición de pines para el multiplexor 1 de actuadores
#define ACTUATOR_MUX1_S0 8
#define ACTUATOR_MUX1_S1 9
#define ACTUATOR_MUX1_S2 10
#define ACTUATOR_MUX1_S3 11
#define ACTUATOR_MUX1_Z  12

// Definición de pines para el multiplexor 2 de actuadores
#define ACTUATOR_MUX2_S0 3
#define ACTUATOR_MUX2_S1 4
#define ACTUATOR_MUX2_S2 5
#define ACTUATOR_MUX2_S3 6
#define ACTUATOR_MUX2_Z  7
```

Elaborada por: Autor

En la figura 3.8 se establece las definiciones de pines y variables esenciales para la integración de un servomecanismo (actuador) y la gestión de estados de sensores y actuadores en un entorno Arduino. Estas definiciones forman parte de un

programa más extenso destinado a controlar un sistema automatizado, donde el servomecanismo, la pantalla LCD y el seguimiento de estados son componentes críticos para el proyecto.

La directiva #define SERVO_PIN 9 asigna el pin digital 9 al servomecanismo, proporcionando una referencia clara y fácilmente modificable para el control del actuador. La elección de este pin específico está vinculada a la configuración física del hardware y puede ajustarse según sea necesario.

Figura 3.9: Programación de los pines de los sensores y actuadores del dispositivo

```
// Definición de la cantidad total de sensores y actuadores
#define NUM_SENSORS 30
#define NUM_ACTUATORS 30

// Variables para almacenar el estado de sensores y actuadores
int sensorStates[NUM_SENSORS];
int actuatorStates[NUM_ACTUATORS];

Servo myServo;  // Crear una instancia del objeto Servo
LiquidCrystal_I2C lcd(0x27, 16, 2);  // Dirección I2C y dimensiones
```

Elaborada por: Autor

Además, se establece la cantidad total de sensores y actuadores mediante las directivas #define NUM_SENSORS 30 y #define NUM_ACTUATORS 30, como se observa en la figura 3.9. Estos valores determinan la dimensión de los arreglos sensorStates y actuatorStates, que almacenan los estados respectivos de los sensores y actuadores. La capacidad de gestionar hasta 30 sensores y actuadores proporciona flexibilidad para aplicaciones escalables.

La instancia Servo myServo; crea un objeto Servo llamado myServo, que actúa como una interfaz de control para el servomecanismo conectado al pin definido

anteriormente. El objeto en cuestión simplifica la labor de control del actuador, posibilitando la acción de abrir y cerrar puertas o muros en función de las selecciones del sistema.

La creación de un objeto de la clase LiquidCrystal_I2C con la dirección I2C 0x27 y un tamaño de pantalla de 16x2, comunica y establece la relación con la pantalla LCD. Este componenté visual tiene un rol significativo dentro de la transmisión de datos importantes acerca de la condición del sistema.

Se inicia la comunicación por Serial.begin(9600) , como se observa en la figura 3.9, lo que da la posibilidad de transferir información entre distintos dispositivos Arduino y otros a través del puerto serie con una velocidad de 9600 baudios. Esta transmisión en serie es fundamental para la depuración y el control del sistema.

Figura 3.10: Configuración de los pines del multiplexor para sensores y actuadores

```
void setup() {
  Serial.begin(9600);

  // Inicialización de la pantalla LCD
  lcd.begin(16, 2);

  // Configuración de pines para los multiplexores de sensores IR y actuadores
  configureMultiplexer(SENSOR_MUX1_S0, SENSOR_MUX1_S1, SENSOR_MUX1_S2, SENSOR_MUX1_S3, SENSOR_MUX1_Z);
  configureMultiplexer(SENSOR_MUX2_S0, SENSOR_MUX2_S1, SENSOR_MUX2_S2, SENSOR_MUX2_S3, SENSOR_MUX2_Z);
  configureMultiplexer(ACTUATOR_MUX1_S0, ACTUATOR_MUX1_S1, ACTUATOR_MUX1_S2, ACTUATOR_MUX1_S3, ACTUATOR_MUX1_Z);
  configureMultiplexer(ACTUATOR_MUX2_S0, ACTUATOR_MUX2_S1, ACTUATOR_MUX2_S2, ACTUATOR_MUX2_S3, ACTUATOR_MUX2_Z);

  // Inicialización del servomecanismo
  myServo.attach(SERVO_PIN);

  // Inicialización de variables
  initializeArray(sensorStates, NUM_SENSORS);
  initializeArray(actuatorStates, NUM_ACTUATORS);
}
```

Elaborada por: Autor

Luego, se inicia la inicialización del monitor LCD a través del comando lcd.begin(16, 2). El paso previo establece las bases de la disposición de la pantalla,

declarando que la misma tiene 16 letras por 2 columnas. La pantalla LCD proporciona una interfaz visual crucial para presentar información relevante sobre el estado del sistema de manera legible y comprensible. Posteriormente, se realiza la configuración de los pines para los multiplexores de sensores infrarrojos y actuadores mediante llamadas a la función configureMultiplexer.

Esta función se encarga de establecer la dirección de los pines necesarios para el control y la lectura de los multiplexores. Los multiplexores son dispositivos esenciales que facilitan la conexión y gestión eficiente de múltiples sensores y actuadores, permitiendo un manejo ordenado y modular de la información. La inicialización del servomecanismo sigue con la línea myServo.attach(SERVO_PIN). Este paso vincula el objeto Servo, previamente creado en la sección de definiciones, al pin específico del Arduino destinado al control del servomecanismo. La función attach establece esta conexión esencial para permitir el movimiento preciso del actuador. Finalmente, se procede con la inicialización de las variables de estado de sensores y actuadores mediante las funciones initializeArray.

Figura 3.11: Configuración de lectura para los sensores y actuadores

```
void loop() {
  // Lectura y procesamiento de estados de sensores y actuadores
  readSensorStates();
  processSensorStates();

  readActuatorStates();
  processActuatorStates();

  // Visualización en la pantalla LCD
  displayOnLCD();

  delay(1000);
}
```

Elaborada por: Autor

La función loop como se observa en la figura 3.11, en el código Arduino representa el núcleo operativo del programa, y se ejecuta constantemente una vez finalizada la configuración inicial. Su objetivo principal es encargarse de la lectura, procesamiento y visualización continua de los estados de sensores y actuadores, así como mostrar esta información en la pantalla LCD. Para comenzar, llama a la función readSensorStates(), que recopila todos los detalles más recientes sobre el estado de los sensores infrarrojos. Los sensores tienen un impacto significativo a la hora de identificar la existencia de vehículos en la zona asignada. Los datos recopilados son vitales para llegar a conclusiones bien informadas sobre el estado del área de estacionamiento.

Posteriormente se inicia la función ProcessSensorStates(), asumiendo la responsabilidad de examinar y manejar los estados del sensor recientemente adquiridos. Este complejo procedimiento implica tomar decisiones cruciales basadas en los datos recopilados, como modificar variables internas o iniciar acciones particulares. Mientras tanto, la función readActuatorStates() recupera simultáneamente información sobre los estados del actuador. En este contexto dado, los actuadores son responsables de gobernar los mecanismos de apertura y cierre de barreras o puertas automatizadas. Comprender su estado se vuelve imperativo para coordinar adecuadamente las medidas esenciales para garantizar un control meticuloso del acceso al aparcamiento.

A continuación, la función ProcessActuatorStates() asume la responsabilidad de analizar y evaluar los estados recientemente adquiridos de los actuadores. Este procedimiento puede implicar la realización de acciones específicas, como abrir o cerrar barreras o puertas, en función de determinadas condiciones detectadas por los

sensores. Posteriormente, se emplea la función displayOnLCD() para exhibir la información recopilada en la pantalla LCD. Esta característica asegura que la pantalla se actualice constantemente con detalles pertinentes sobre la ocupación y disponibilidad de espacios de estacionamiento, brindando así a los usuarios una interfaz visual lúcida. Además, se incorpora una pausa de retraso (1000) dentro de cada iteración del bucle, creando un intervalo de un segundo entre operaciones.

Figura 3.12: Configuración de entrada y salida del multiplexor

```
void configureMultiplexer(int s0, int s1, int s2, int s3, int z) {
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
  pinMode(z, INPUT);
}

void readSensorStates() {
  // Lectura de estados de sensores IR
  for (int i = 0; i < NUM_SENSORS; i++) {
    selectSensorChannel(i);
    sensorStates[i] = digitalRead(SENSOR_MUX1_Z) || digitalRead(SENSOR_MUX2_Z);
  }
}
```

Elaborada por: Autor

La función configureMultiplexer como se observa en la figura 3.12 tiene como propósito establecer la configuración inicial de los pines asociados a un multiplexor. Recibe como parámetros los pines de selección de canal (s0, s1, s2, s3) y el pin de salida de datos (z). Primero, configura los pines de selección como salidas mediante la función pinMode con la constante OUTPUT. Luego, configura el pin de salida de datos como entrada utilizando pinMode con la constante INPUT. En el contexto del código, esta función se utiliza para configurar los pines de los multiplexores asociados tanto a los sensores infrarrojos (IR) como a los actuadores.

Por otro lado, la función readSensorStates se encarga de la lectura de los estados de los sensores infrarrojos (IR). Utiliza un bucle for para iterar sobre todos los

sensores definidos por la constante NUM_SENSORS. En cada iteración, se selecciona el canal del sensor mediante la llamada a la función selectSensorChannel(i). Posteriormente, se realiza la lectura del estado del sensor y se almacena en el arreglo sensorStates. La lectura de los estados se lleva a cabo mediante la operación lógica de los pines de salida de ambos multiplexores. La función que configureMultiplexer establece los parámetros iniciales de los pines que están asociadas a los multiplexores y el readSensorStates se encarga de tomar estado de los sensores de infrarrojos.

La función selectSensorChannel como se observa en la figura 3.13 juega un papel crucial en el proceso de selección del canal de los multiplexores asociados a los sensores infrarrojos (IR). Su propósito es determinar y establecer el canal específico que será leído en un momento dado, permitiendo así la lectura precisa del estado de un sensor en particular. En detalle, la función toma como argumento el número del canal que se desea seleccionar. Luego, mediante el uso de la función digitalWrite, se manipulan los pines de selección (S0, S1, S2, S3) en ambos multiplexores, SENSOR_MUX1 y SENSOR_MUX2, para configurar el canal deseado.

Figura 3.13: Configuración de los canales del multiplexor

```
void selectSensorChannel(int channel) {
  // Selección de canal en los multiplexores de sensores IR
  digitalWrite(SENSOR_MUX1_S0, channel & 0x01);
  digitalWrite(SENSOR_MUX1_S1, (channel >> 1) & 0x01);
  digitalWrite(SENSOR_MUX1_S2, (channel >> 2) & 0x01);
  digitalWrite(SENSOR_MUX1_S3, (channel >> 3) & 0x01);

  digitalWrite(SENSOR_MUX2_S0, channel & 0x01);
  digitalWrite(SENSOR_MUX2_S1, (channel >> 1) & 0x01);
  digitalWrite(SENSOR_MUX2_S2, (channel >> 2) & 0x01);
  digitalWrite(SENSOR_MUX2_S3, (channel >> 3) & 0x01);
}
```

Elaborada por: Autor

La expresión (channel & 0x01) obtiene el bit menos significativo del número del canal, determinando el estado del pin correspondiente a la unidad (S0). De manera

similar, las expresiones (channel >> 1) & 0x01, (channel >> 2) & 0x01, y (channel >> 3) & 0x01 obtienen los bits siguientes del número del canal, correspondientes a los pines S1, S2 y S3, respectivamente. Estos valores son escritos en los pines adecuados de ambos multiplexores. Esta función es esencial para la operación coordinada de los multiplexores y garantiza que el sistema pueda direccionar eficientemente la lectura hacia el sensor correcto en el momento preciso.

El objetivo principal de la función ProcessSensorStates como se observa en la figura 3.14, es manejar el procesamiento de los estados del sensor de infrarrojos (IR) que se utilizan en el sistema. Siempre que se solicite esta función, mostrará una secuencia de mensajes en la consola de salida que indicarán la categoría específica de información que se está procesando; en esta situación, son los estados del sensor de infrarrojos. Usando el comando Serial.println, se imprimirá una línea divisoria y luego se ejecutará la función printArray, lo que permitirá una presentación bien organizada de los estados actuales del sensor.

Figura 3.14: Configuración de los estados y lectura de los sensores y actuadores

```
void processSensorStates() {
  // Procesamiento de estados de sensores IR
  Serial.println("---- Estados de Sensores IR ----");
  printArray("Sensor", sensorStates, NUM_SENSORS);
}

void readActuatorStates() {
  // Lectura de estados de actuadores
  for (int i = 0; i < NUM_ACTUATORS; i++) {
    selectActuatorChannel(i);
    actuatorStates[i] = digitalRead(ACTUATOR_MUX1_Z) || digitalRead(ACTUATOR_MUX2_Z);
  }
}
```

Elaborada por: Autor

La función readActuatorStates desempeña un papel crucial en el contexto en el que se utiliza. Su objetivo fundamental es leer los estados de los actuadores. Para

lograr esto, se emplea un bucle for dentro de la función. Este bucle recorre en iteración todos los actuadores que se han definido en el sistema, ejecutando la función selectActuatorChannel para cada actuador individual. Dentro de esta función, el estado del actuador se obtiene leyendo de dos pines específicos, a saber, ACTUATOR_MUX1_Z y ACTUATOR_MUX2_Z. Luego, el resultado recuperado se almacena en la matriz actuadorStates.

Es de suma importancia leer los estados de los actuadores para comprender la disposición actual de los componentes mecánicos en el sistema. El funcionamiento de estos actuadores, encargados de gestionar las barreras o puertas automáticas, se manifiesta en los datos adquiridos a través de esta función particular. La utilización de multiplexores para los actuadores requiere seleccionar el canal correcto antes de leer el estado, garantizando así coherencia en la información obtenida.

La colaboración entre ProcessSensorStates y readActuatorStates juega un papel crucial en el monitoreo y control del sistema. Esta colaboración ofrece una perspectiva completa sobre los estados de los sensores y actuadores IR, lo que facilita el seguimiento de las condiciones del estacionamiento en tiempo real. Además, estos valiosos datos son esenciales para tomar decisiones automatizadas, como la gestión de barreras, en función de eventos específicos detectados por los sensores.

La función selectActuatorChannel como se observa en la figura 3.15, desempeña un papel crítico en el sistema, ya que se encarga de la selección precisa del canal en los multiplexores de actuadores. Este procedimiento es esencial para dirigir la operación hacia un actuador específico dentro del conjunto total. En otras palabras,

cuando se invoca esta función con un parámetro que representa el canal deseado, se activan los pines de control correspondientes en los multiplexores de actuadores.

Figura 3.15: Configuración de los estados y lectura de los sensores y actuadores

```
void selectActuatorChannel(int channel) {
  // Selección de canal en los multiplexores de actuadores
  digitalWrite(ACTUATOR_MUX1_S0, channel & 0x01);
  digitalWrite(ACTUATOR_MUX1_S1, (channel >> 1) & 0x01);
  digitalWrite(ACTUATOR_MUX1_S2, (channel >> 2) & 0x01);
  digitalWrite(ACTUATOR_MUX1_S3, (channel >> 3) & 0x01);

  digitalWrite(ACTUATOR_MUX2_S0, channel & 0x01);
  digitalWrite(ACTUATOR_MUX2_S1, (channel >> 1) & 0x01);
  digitalWrite(ACTUATOR_MUX2_S2, (channel >> 2) & 0x01);
  digitalWrite(ACTUATOR_MUX2_S3, (channel >> 3) & 0x01);
}
```

Elaborada por: Autor

El procedimiento comienza mediante el establecimiento de los pines de selección (S0, S1, S2 y S3) en ambos multiplexores de actuadores (ACTUATOR_MUX1 y ACTUATOR_MUX2). Estos pines se configuran utilizando operadores bit a bit para interpretar el valor del canal. Por ejemplo, digitalWrite(ACTUATOR_MUX1_S0, channel & 0x01) establece el pin S0 en el multiplexor de actuadores 1 según el bit menos significativo del canal. Este enfoque se repite para los pines S1, S2 y S3, utilizando desplazamientos de bits para abordar cada posición binaria del número de canal.

La función asegura la correcta identificación y direccionamiento del canal deseado entre los múltiples canales disponibles. Esto resulta fundamental, ya que cada canal está asociado a un actuador específico dentro del sistema. La correcta selección del canal permite acceder y operar directamente sobre el actuador correspondiente, garantizando así una gestión eficiente y precisa de los elementos mecánicos.

La función processActuatorStates como se observa en la figura 3.16, desempeña una función central en la gestión de los estados de los actuadores en el sistema, contribuyendo significativamente a la lógica de control y respuesta del conjunto. Cuando se ejecuta, realiza varias operaciones cruciales. Primero, muestra por el puerto serie una indicación clara y detallada de los estados de los actuadores, creando un registro informativo bajo el encabezado "---- Estados de Actuadores ----". Este registro proporciona una instantánea visual de la condición activada o desactivada de cada actuador en el sistema, facilitando la supervisión y la resolución de problemas.

Figura 3.16: Configuración de activación de los actuadores

```
void processActuatorStates() {
  // Procesamiento de estados de actuadores
  Serial.println("---- Estados de Actuadores ----");
  printArray("Actuador", actuatorStates, NUM_ACTUATORS);

  // Acciones en respuesta al estado activado del actuador
  for (int i = 0; i < NUM_ACTUATORS; i++) {
    if (actuatorStates[i] == 1) {
      moveBarrier(i + 1);
    }
  }
}
```

Elaborada por: Autor

Posteriormente, la función realiza acciones específicas en respuesta al estado activado del actuador. Un bucle for itera sobre cada actuador en el sistema, evaluando si su estado es igual a 1, lo que indica que está activado. En caso afirmativo, invoca la función moveBarrier con el número de actuador incrementado en 1. Este procedimiento busca activamente los actuadores en un estado activado y desencadena operaciones correspondientes, como el movimiento de barreras o puertas automáticas.

La función processActuatorStates se integra de manera integral en el ciclo de ejecución principal y se ejecuta de manera regular para supervisar y gestionar los

estados de los actuadores. Su capacidad para realizar operaciones específicas en respuesta a condiciones particulares contribuye a la adaptabilidad y eficiencia del sistema, ya que permite respuestas inmediatas a cambios en el entorno del estacionamiento.

La función moveBarrier como se observa en la figura 3.17, constituye un componente crítico del sistema al facilitar el control físico de barreras o puertas automáticas asociadas a actuadores específicos. Al ser invocada, realiza una serie de pasos diseñados para manipular el servomecanismo y, por ende, la posición de la barrera o puerta. Su estructura y operación se describen en detalle a continuación. Primero, se emite un mensaje informativo a través del puerto serie indicando que se está moviendo la barrera o puerta para el actuador en cuestión. Este mensaje proporciona una trazabilidad visual y registrada de las acciones ejecutadas, siendo de utilidad para la supervisión y el análisis retrospectivo de eventos en el sistema.

Figura 3.17: Configuración para apertura y cierre del actuador

```
void moveBarrier(int actuatorNumber) {
  // Mover la barrera o puerta utilizando el servomecanismo
  Serial.println("Moviendo la barrera o puerta para el actuador " + String(actuatorNumber));

  // Ajusta estos valores según la configuración física de tu sistema
  myServo.write(90);  // Angulo para cerrar la barrera
  delay(1000);        // Espera para simular el tiempo de movimiento
  myServo.write(0);   // Angulo para abrir la barrera
  delay(1000);        // Espera para simular el tiempo de movimiento
}

void initializeArray(int array[], int size) {
  // Inicializar un arreglo con ceros
  for (int i = 0; i < size; i++) {
    array[i] = 0;
  }
}
```

Elaborada por: Autor

Luego, se efectúan ajustes específicos según la configuración física del sistema. En este caso, se utilizan valores angulares para controlar el movimiento del

servomecanismo, lo que a su vez controla la posición de la barrera o puerta. El valor 90 representa el ángulo para cerrar la barrera, mientras que el valor 0 indica el ángulo para abrir la barrera. Estos valores pueden ser adaptados según las necesidades y las características específicas del entorno de estacionamiento. Adicionalmente, se incorporan pausas temporales mediante la función delay para simular el tiempo de movimiento de la barrera. Estas pausas contribuyen a la sincronización y a la representación realista de los procesos físicos asociados al desplazamiento de la barrera o puerta.

La función moveBarrier se convierte, así, en un eslabón crucial en la implementación del control físico de accesos en el sistema, destacándose por su capacidad para interactuar directamente con los actuadores y traducir las decisiones lógicas del sistema en acciones tangibles y observables. Por otro lado, la función initializeArray desempeña una función fundamental en la fase de inicialización del sistema. Su propósito es establecer un arreglo con ceros, lo que proporciona un estado base consistente y predecible para los elementos asociados, en este caso, los estados de los sensores y actuadores. Al asignar valores iniciales de cero a cada elemento del arreglo, se establece una condición conocida y estable en la que el sistema puede basar sus futuras operaciones y decisiones.

Figura 3.18: Configuración para apertura y cierre del actuador

```
void printArray(String label, int array[], int size) {
  // Imprimir el contenido de un arreglo
  for (int i = 0; i < size; i++) {
    Serial.print(label + " ");
    Serial.print(i + 1);
    Serial.print(": ");
    Serial.println(array[i]);
  }
  Serial.println();
}
```

Elaborada por: Autor

La función printArray como se observa en la figura 3.18, desempeña un papel fundamental en la visualización y el monitoreo de los estados de los sensores y actuadores en el sistema. Su propósito principal radica en imprimir de manera estructurada y legible el contenido de un arreglo específico, lo que facilita la interpretación y el análisis de los datos recopilados. Cuando se invoca esta función, se le proporcionan tres parámetros esenciales: una etiqueta que identifica el tipo de elemento en el arreglo (ya sea un sensor o un actuador), el propio arreglo de datos y su tamaño. La etiqueta se utiliza para contextualizar la información, brindando claridad sobre la categoría de elementos que se están presentando.

Dentro de la función, se emplea un bucle for que itera a través de cada elemento del arreglo. Durante cada iteración, se emiten mensajes secuenciales a través del puerto serie. Estos mensajes incluyen la etiqueta específica, el índice del elemento actual incrementado en uno para una numeración más intuitiva, y el valor correspondiente almacenado en el arreglo.

Figura 3.19: Configuración del display del estacionamiento inteligente

```
void displayOnLCD() {
  // Mostrar información en la pantalla LCD
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Estacionados: ");
  lcd.print(countOccurrences(sensorStates, NUM_SENSORS, 1));

  lcd.setCursor(0, 1);
  lcd.print("Disponibles: ");
  lcd.print(countOccurrences(sensorStates, NUM_SENSORS, 0));
}
```

Elaborada por: Autor

Se muestra la etiqueta "Estacionados:" seguida de la cantidad de sensores que actualmente detectan la presencia de vehículos como se observa en la figura 3.19. Esta información se obtiene mediante la función countOccurrences, que cuenta el número

de sensores cuyo estado es igual a 1, indicando que están actualmente ocupados. La siguiente línea de la pantalla LCD, posicionada en (0, 1), presenta la etiqueta "Disponibles:" seguida de la cantidad de sensores que indican la disponibilidad de espacios de estacionamiento. Nuevamente, la función countOccurrences es utilizada para determinar la cantidad de sensores con un estado igual a 0, lo que indica que los espacios asociados están libres.

Figura 3.20: Configuración del contador de vehículos parqueados

```
int countOccurrences(int array[], int size, int target) {
  // Contar ocurrencias de un valor en un arreglo
  int count = 0;
  for (int i = 0; i < size; i++) {
    if (array[i] == target) {
      count++;
    }
  }
  return count;
}
```

Elaborada por: Autor

La función countOccurrences como se observa en la figura 3.20, desempeña una función crucial en el sistema al proporcionar un método eficiente para determinar el número de ocurrencias de un valor específico en un arreglo. Este valor específico, denominado "target", representa la condición que se busca contar dentro del arreglo.

Cuando se activa esta función, se inicia una variable count en cero, la cual actúa como acumulador para registrar las ocurrencias del valor objetivo. Luego, a través de un bucle for, la función examina cada elemento del arreglo, comparándolo con el valor objetivo definido por target. Si se encuentra una coincidencia, el contador se incrementa en uno. Este proceso de inspección y conteo continúa hasta que todos los elementos del arreglo han sido evaluados. Finalmente, el resultado, es decir, el

número total de ocurrencias del valor objetivo en el arreglo se devuelve como un entero.

En la Tabla 3.8 se observa los precios de los Elementos para el Parqueadero Inteligente contiene una investigación completa de los componentes esenciales para la creación del sistema. Este detallado análisis de costos permite un estudio detallado de la totalidad de los componentes a partir de su precio individual, la cantidad demandada y el costo total que tiene, de esta manera se proporciona una visión integral de la ejecución del proyecto.

Tabla 3.8 *Costo de los elementos para el parqueadero inteligente*

Elemento	**Costo** Unidad (**USD)**	**Cantidad**	**Total (USD)**
Arduino Mega	25	1	25
Sensores Infrarrojos TCRT5000	2	30	60
Actuadores	25	30	750
Pantalla LCD 16x2	6	1	6
Módulo I2C para LCD	3,5	1	3,5
Multiplexor (CD74HC4067)	5	30	150
Multiplexor (CD74HC154E)	5	30	150
Fuente DC de 5v 30A	30	2	60
Fuente DC de 9v 30A	45	2	90

Fuente: Autor

Conclusiones y recomendaciones

Conclusiones

- El diseño del Sistema de Control Inteligente con Arduino Mega para el estacionamiento representa un avance significativo. Este sistema promete ofrecer mejoras sustanciales en la gestión del estacionamiento, brindando una solución eficiente y adaptable a las necesidades específicas del entorno de Canal y Radio UCSG.
- La programación y diseño del diagrama de conexiones para el Sistema de Control Inteligente con Arduino Mega demuestran un enfoque estructurado y detallado.
- El estudio teórico han fijado una base sólida para potenciales mejoras futuras en el Sistema de Control Inteligente para el Estacionamiento del Canal y Radio UCSG. Estas mejoras son esenciales para garantizar la adaptabilidad y eficiencia del sistema en una posible implementación real.

Recomendaciones

- Considerar la integración de un sistema de notificación o una interfaz de usuario para permitir a los conductores acceder a información en tiempo real sobre la disponibilidad de estacionamiento.
- Realizar sesiones de capacitación y concientización para la comunidad universitaria sobre el funcionamiento del sistema antes de su implementación.
- Evaluar posibles integraciones con otros sistemas de gestión de la universidad para garantizar una administración coordinada y efectiva de recursos.
- Establecer protocolos de seguridad cibernética para proteger la integridad y la privacidad de los datos recolectados por el sistema, considerando posibles vulnerabilidades y medidas de protección.

Bibliografías

Alhassan, S. (2022). A Robust Platform for Mobile Robotics Teaching and Developing Using Arduino's Integrated Development Environment (IDE) for Programming the Arduino MEGA 2560. *Department of Electrical and Computer Engineering: Dissertations, Theses, and Student Research*. https://digitalcommons.unl.edu/elecengtheses/138

Allbadi, Y., Shehab, J. N., & Jasim, M. M. (2021). The Smart Parking System Using Ultrasonic Control Sensors. *IOP Conference Series: Materials Science and Engineering*, *1076*(1), 012064. https://doi.org/10.1088/1757-899X/1076/1/012064

Argüello, F. (2021, septiembre 21). *Espacios de estacionamiento inteligentes—Infoteknico*. https://www.infoteknico.com/espacios-de-estacionamiento/

Bannon, E. (2023, abril 26). *How Parking Affects Our Urban Ecosystem*. https://www.parking.net/parking-industry-blog/parking-network/how-parking-affects-our-urban-ecosystem

Barrett, S. F. (2020). Arduino Platforms. En S. F. Barrett (Ed.), *Arduino I: Getting Started* (pp. 33-73). Springer International Publishing. https://doi.org/10.1007/978-3-031-79915-0_2

Buitleir, D. (2023, mayo 10). Workplace Parking: Past, Present, and Future Predictions. *Wayleadr Blog*. https://wayleadr.com/blog/workplace-parking-past-present-future-predictions/

Came. (2023, septiembre 18). *Problemas que resuelven los estacionamientos—Came Soluciones*. https://camesoluciones.com/blog/problemas-que-resuelven-los-estacionamientos/

Cnne, I. (2020, enero 11). Crean un nuevo estacionamiento inteligente en China para ganar más espacio | Video. *CNN*. https://cnnespanol.cnn.com/video/estacionamiento-inteligente-china-diagonal-robotica-vo-imagen-dia-cafe/

Cogniteq. (2023, noviembre 29). *Smart Parking Systems: Types & Benefits | Cogniteq*. https://www.cogniteq.com/blog/smart-parking-systems-what-they-are-and-why-theyre-beneficial

Cotrone, V. (2022, agosto 11). *Green Parking Lots: Mitigating Climate Change and the Urban Heat Island*. https://extension.psu.edu/green-parking-lots-mitigating-climate-change-and-the-urban-heat-island

Dans, E. (2020). *Vehículos autónomos, esquemas de propiedad e implicaciones*. https://www.enriquedans.com/2016/05/vehiculos-autonomos-esquemas-de-propiedad-e-implicaciones.html

Dickie, C. (2013, julio 31). *Parking Lot Pollutants And How Can They Be Minimized?* Universal Site Services. https://www.universalsiteservices.com/what-are-parking-lot-pollutants-and-how-can-they-be-minimized/

Diène, B., Rodrigues, J. J. P. C., Diallo, O., Ndoye, E. H. M., & Korotaev, V. V. (2020). Data management techniques for Internet of Things. *Mechanical Systems and Signal Processing*, *138*, 106564.https://doi.org/10.1016/j.ymssp.2019.106564

Diosdado, R. (2019, julio 5). *Control de servo-motor con Arduino*. Zona Maker. https://www.zonamaker.com/control-de-servomotores-con-arduino

Dogaroglu, B., Caliskanelli, S. P., & Tanyel, S. (2021). Comparison of intelligent parking guidance system and conventional system with regard to capacity utilisation. *Sustainable Cities and Society*, *74*, 103152. https://doi.org/10.1016/j.scs.2021.103152

Ecuarobot. (2020, marzo 10). Sistema de aparcamiento automatizado Arduino. *EcuaRobot*. https://ecuarobot.com/2020/03/10/sistema-de-aparcamiento-automatizado-arduino/

Espinoza, J. B. (2022). Sistema de estacionamiento inteligente aplicando internet de las cosas (IoT), para gestionar el parqueo vehicular del garaje Ebenezer, Bagua Grande 2023. *Repositorio Institucional - UCV*. https://repositorio.ucv.edu.pe/handle/20.500.12692/108766

Golan, D. (2023, agosto 20). *Automated Parking AI: The Future of Safe Urban Transportation*. Hailo. https://hailo.ai/blog/backing-into-the-future-unlocking-the-potential-of-automated-parking/

Hanzl, J. (2020). Parking Information Guidance Systems and Smart Technologies Application Used in Urban Areas and Multi-storey Car Parks. *Transportation Research Procedia*, *44*, 361-368. https://doi.org/10.1016/j.trpro.2020.02.030

Intercomp. (2020, febrero 10). The economic benefits of Smart Parking Systems® < Smart Parking Systems—Intercomp Innovation. *Smart Parking Systems - Intercomp Innovation*. https://smartparkingsystems.com/i-benefici-economici-di-smart-parking-systems/

Jo, Y., Ha, J., & Hwang, S. (2023). Survey of Technology in Autonomous Valet Parking System. *International Journal of Automotive Technology*, *24*(6), 1577-1587. https://doi.org/10.1007/s12239-023-0127-1

Kaur, G., & Garg, H. (2023). A novel algorithm for autonomous parking vehicles using adjustable probabilistic neutrosophic hesitant fuzzy set features. *Expert Systems with Applications, 226*, 120101. https://doi.org/10.1016/j.eswa.2023.120101

Khalid, M., Wang, K., Aslam, N., Cao, Y., Ahmad, N., & Khan, M. K. (2021). From smart parking towards autonomous valet parking: A survey, challenges and future Works. *Journal of Network and Computer Applications, 175*, 102935. https://doi.org/10.1016/j.jnca.2020.102935

Krishnamurthy, C. K. B., & Ngo, N. S. (2020). The effects of smart-parking on transit and traffic: Evidence from SFpark. *Journal of Environmental Economics and Management, 99*, 102273. https://doi.org/10.1016/j.jeem.2019.102273

Liu, J., Wu, J., & Sun, L. (2020). Control method of urban intelligent parking guidance system based on Internet of Things. *Computer Communications, 153*, 279-285. https://doi.org/10.1016/j.comcom.2020.01.063

Lom, M., & Pribyl, O. (2021). Smart city model based on systems theory. *International Journal of Information Management, 56*, 102092. https://doi.org/10.1016/j.ijinfomgt.2020.102092

Mapfre. (2020). *Sensores de aparcamiento*. Fundación MAPFRE. https://www.fundacionmapfre.org/educacion-divulgacion/seguridad-vial/sistemas-adas/tipos/sensores-de-aparcamiento/

Novable. (2023, febrero 28). *5 Innovative solutions in the smart parking industry: Staff Picked*. https://novable.com/5-innovative-ai-solutions-in-the-smart-parking-industry-staff-picked/

Parker, W. (2023, diciembre 14). The Future of Autonomous Vehicles: Adapting Pay and Display Services for Self-Driving Cars. *Medium*. https://medium.com/@williamparker24091987/the-future-of-autonomous-vehicles-adapting-pay-and-display-services-for-self-driving-cars-76c596abc620

Parmar, J., Das, P., & Dave, S. M. (2020). Study on demand and characteristics of parking system in urban areas: A review. *Journal of Traffic and Transportation Engineering (English Edition)*, *7*(1), 111-124. https://doi.org/10.1016/j.jtte.2019.09.003

Patel, B. (2021, septiembre 29). *IoT Based Smart Parking Management System: Benefits + Potential*. https://www.spaceotechnologies.com/blog/build-parking-management-system-reasons/

Patro, S. P., Patel, P., Senapaty, M. K., Padhy, N., & Sah, R. D. (2020). IoT based Smart Parking System: A Proposed Algorithm and Model. *2020 International Conference on Computer Science, Engineering and Applications (ICCSEA)*, 1-6. https://doi.org/10.1109/ICCSEA49143.2020.9132923

Piqoa. (2022, octubre 1). *SENSOR INFRARROJO TCRT5000 (para seguidor de línea)—Piqoa*. https://www.piqoa.com/producto/sensor-tcrt5000-para-seguidor-de-linea/

Quercus. (2023). *Current trends in the parking industry*. https://quercus-technologies.com/news/current-trends-in-the-parking-industry

Sabra, Z. (2023, julio 19). *6 Technologies and Trends That Will Impact the Future of Transportation*. Mead & Hunt. https://meadhunt.com/6-trends-transportation/

Saharan, S., Kumar, N., & Bawa, S. (2020). An efficient smart parking pricing system for smart city environment: A machine-learning based approach. *Future Generation Computer Systems*, *106*, 622-640. https://doi.org/10.1016/j.future.2020.01.031

Septian, M. S. (2022). Design and Build an Arduino Mega-Based Automatic Lawn Mower. *Instal : Jurnal Komputer*, *14*(01), Article 01. https://doi.org/10.54209/jurnalkomputer.v14i01.31

Setyo, H. (2021, abril). *Fire Extinguisher Wheel Robot Based on Arduino Mega 2560 R3 with Android Smartphone Control | Buletin Ilmiah Sarjana Teknik Elektro*. http://www.journal2.uad.ac.id/index.php/biste/article/view/1760

Sharath, G. S., Hiremath, N., & Manjunatha, G. (2021). Design and analysis of gantry robot for pick and place mechanism with Arduino Mega 2560 microcontroller and processed using pythons. *Materials Today: Proceedings*, *45*, 377-384. https://doi.org/10.1016/j.matpr.2020.11.965

Shroud, M., Eame, M., Elsaghayer, E., Almabrouk, A., & Nassar, Y. (2023). *Challenges and Opportunities in Smart Parking Sensor Technologies*. *1*, 44-59.

Tecmikro. (2024). *Mega 2560 R3 para Arduino®*. Tecmikro Ecuador. https://tecmikro.com/tarjetas-programables/332-mega-2560-r3-para-arduino.html

Watson, D. (2021, enero 20). *Introduction to Arduino Mega 2560 Rev3—The Engineering Projects*. https://www.theengineeringprojects.com/2021/01/introduction-to-arduino-mega-2560-rev3.html/?amp=1

Anexo 1

Código implementado para la programación del Arduino Mega

```
#include <Wire.h>
#include <Servo.h>
#include <LiquidCrystal_I2C.h>

#define SENSOR_MUX1_S0 8
#define SENSOR_MUX1_S1 9
#define SENSOR_MUX1_S2 10
#define SENSOR_MUX1_S3 11
#define SENSOR_MUX1_Z  7

#define SENSOR_MUX2_S0 12
#define SENSOR_MUX2_S1 13
#define SENSOR_MUX2_S2 14
#define SENSOR_MUX2_S3 15
#define SENSOR_MUX2_Z  6

#define ACTUATOR_MUX1_S0 4
#define ACTUATOR_MUX1_S1 5
#define ACTUATOR_MUX1_S2 6
#define ACTUATOR_MUX1_S3 7
#define ACTUATOR_MUX1_Z  12
```

```
#define ACTUATOR_MUX2_S0 16
#define ACTUATOR_MUX2_S1 17
#define ACTUATOR_MUX2_S2 18
#define ACTUATOR_MUX2_S3 19
#define ACTUATOR_MUX2_Z  20
#define SERVO_PIN 9

#define NUM_SENSORS 30
#define NUM_ACTUATORS 30

int sensorStates[NUM_SENSORS];
int actuatorStates[NUM_ACTUATORS];

LiquidCrystal_I2C lcd(0x27, 16, 2);
void setup() {
  Serial.begin(9600);
  lcd.begin(16, 2);

  configureMultiplexer(SENSOR_MUX1_S0, SENSOR_MUX1_S1,
SENSOR_MUX1_S2, SENSOR_MUX1_S3, SENSOR_MUX1_Z);
  configureMultiplexer(SENSOR_MUX2_S0, SENSOR_MUX2_S1,
SENSOR_MUX2_S2, SENSOR_MUX2_S3, SENSOR_MUX2_Z);
```

```
  configureMultiplexer(ACTUATOR_MUX1_S0, ACTUATOR_MUX1_S1,
ACTUATOR_MUX1_S2, ACTUATOR_MUX1_S3, ACTUATOR_MUX1_Z);
  configureMultiplexer(ACTUATOR_MUX2_S0, ACTUATOR_MUX2_S1,
ACTUATOR_MUX2_S2, ACTUATOR_MUX2_S3, ACTUATOR_MUX2_Z);
  myServo.attach(SERVO_PIN);

  initializeArray(sensorStates, NUM_SENSORS);
  initializeArray(actuatorStates, NUM_ACTUATORS);
}

void loop() {
  readSensorStates();
  processSensorStates();
  readActuatorStates();
  processActuatorStates();
  displayOnLCD();
  delay(1000);
}

void configureMultiplexer(int s0, int s1, int s2, int s3, int z) {
  pinMode(s0, OUTPUT);
  pinMode(s1, OUTPUT);
  pinMode(s2, OUTPUT);
  pinMode(s3, OUTPUT);
```

```
  pinMode(z, INPUT);
}

void readSensorStates() {
  for (int i = 0; i < NUM_SENSORS; i++) {
    selectSensorChannel(i);
    sensorStates[i] = digitalRead(SENSOR_MUX1_Z) ||
digitalRead(SENSOR_MUX2_Z);
  }
}

void selectSensorChannel(int channel) {
  digitalWrite(SENSOR_MUX1_S0, channel & 0x01);
  digitalWrite(SENSOR_MUX1_S1, (channel >> 1) & 0x01);
  digitalWrite(SENSOR_MUX1_S2, (channel >> 2) & 0x01);
  digitalWrite(SENSOR_MUX1_S3, (channel >> 3) & 0x01);

  digitalWrite(SENSOR_MUX2_S0, channel & 0x01);
  digitalWrite(SENSOR_MUX2_S1, (channel >> 1) & 0x01);
  digitalWrite(SENSOR_MUX2_S2, (channel >> 2) & 0x01);
  digitalWrite(SENSOR_MUX2_S3, (channel >> 3) & 0x01);
}

void processSensorStates() {
```

```
  Serial.println("---- Estados de Sensores IR ----");
  printArray("Sensor", sensorStates, NUM_SENSORS);
}
void readActuatorStates() {
  for (int i = 0; i < NUM_ACTUATORS; i++) {
    selectActuatorChannel(i);
    actuatorStates[i]          =          digitalRead(ACTUATOR_MUX1_Z)          ||
digitalRead(ACTUATOR_MUX2_Z);
  }
}

void selectActuatorChannel(int channel) {
  digitalWrite(ACTUATOR_MUX1_S0, channel & 0x01);
  digitalWrite(ACTUATOR_MUX1_S1, (channel >> 1) & 0x01);
  digitalWrite(ACTUATOR_MUX1_S2, (channel >> 2) & 0x01);
  digitalWrite(ACTUATOR_MUX1_S3, (channel >> 3) & 0x01);

  digitalWrite(ACTUATOR_MUX2_S0, channel & 0x01);
  digitalWrite(ACTUATOR_MUX2_S1, (channel >> 1) & 0x01);
  digitalWrite(ACTUATOR_MUX2_S2, (channel >> 2) & 0x01);
  digitalWrite(ACTUATOR_MUX2_S3, (channel >> 3) & 0x01);
}

void processActuatorStates() {
```

```
  Serial.println("---- Estados de Actuadores ----");
  printArray("Actuador", actuatorStates, NUM_ACTUATORS);

  for (int i = 0; i < NUM_ACTUATORS; i++) {
    if (actuatorStates[i] == 1) {
      moveBarrier(i + 1);
    }
  }
}

void moveBarrier(int actuatorNumber) {
  Serial.println("Moviendo la barrera o puerta para el actuador " +
String(actuatorNumber));

  myServo.write(90);
  delay(1000);
  myServo.write(0);
  delay(1000);
}

void initializeArray(int array[], int size) {
  for (int i = 0; i < size; i++) {
    array[i] = 0;
  }
```

```
}


void printArray(String label, int array[], int size) {
  for (int i = 0; i < size; i++) {
    Serial.print(label + " ");
    Serial.print(i + 1);
    Serial.print(": ");
    Serial.println(array[i]);
  }
  Serial.println();
}


void displayOnLCD() {
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("Estacionados: ");
  lcd.print(countOccurrences(sensorStates, NUM_SENSORS, 1));


  lcd.setCursor(0, 1);
  lcd.print("Disponibles: ");
  lcd.print(countOccurrences(sensorStates, NUM_SENSORS, 0));
}


int countOccurrences(int array[], int size, int target) {
```

```
  int count = 0;
  for (int i = 0; i < size; i++) {
    if (array[i] == target) {
      count++;
    }
  }
  return count;
}
```

Glosario

ATmega2560: Un microcontrolador que se encuentra entre la familia AVR y que se utiliza en las placas de Arduino, debido a que tiene la capacidad de gestionar datos y la variedad de proyectos que es posible hacer con él.

Automóviles automatizados (AV): Automóviles que poseen características que les proporciona la capacidad de operarse sin la colaboración de un ser humano, utilizando sensores y sistemas de control.

Ethernet Shield: Un complemento que añade placas de Arduino adicionales que posibilitan la comunicación por internet y la administración de estas.

IoT (Internet de las Cosas): Termino que se utiliza para nombrar la agrupación de objetos materiales por medio de la internet, posibilitando la recolección y transmisión de datos.

PWM (Modulación de Ancho de Pulso): Práctica que se utiliza para controlar la fuerza que se le da a electrónicos, controlando la cantidad de tiempo que pasa en una situación alta o baja.

Sensor: Dispositivo que detecta cambios en el contexto y transforma esa información en energía eléctrica, este instrumento está usado en proyectos de electrónicos.

Servomotor: Motor de electricidad que sirve para generar movimiento con una precisión alta, normalmente usado en proyectos de robótica y automatización.

DECLARACIÓN Y AUTORIZACIÓN

Yo, **Angulo Angulo, Ariel Eduardo** con C.C: **0803582972** autor del Trabajo de Integración Curricular: **Estudio de un Sistema de control Inteligente para el Estacionamiento del Canal y Radio UCSG,** previo a la obtención del título de **Ingeniero en Electrónica y Automatización,** en la Universidad Católica de Santiago de Guayaquil.

1.- Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar a la SENESCYT en formato digital una copia del referido trabajo de integración curricular para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.

2.- Autorizo a la SENESCYT a tener una copia del referido trabajo de integración curricular, con el propósito de generar un repositorio que democratice la información, respetando las políticas de propiedad intelectual vigentes.

Guayaquil, 15 de febrero del año 2024

Angulo Angulo, Ariel Eduardo

C.C: 0803582972

REPOSITORIO NACIONAL EN CIENCIA Y TECNOLOGÍA			
FICHA DE REGISTRO DE TESIS/TRABAJO DE INTEGRACIÓN CURRICULAR			
TÍTULO Y SUBTÍTULO:	Estudio de un Sistema de control Inteligente para el Estacionamiento del Canal y Radio UCSG.		
AUTOR(ES)	Angulo Angulo, Ariel Eduardo		
REVISOR(ES)/TUTOR(ES)	Ing. Bohórquez Escobar, Celso Bayardo. PHD.		
INSTITUCIÓN:	Universidad Católica de Santiago de Guayaquil.		
FACULTAD:	Facultad de Educación Técnica para el Desarrollo		
CARRERA:	Ingeniería Electrónica y Automatización		
TITULO OBTENIDO:	Ingeniero en Electrónica y Automatización		
FECHA DE PUBLICACIÓN:	15 de febrero del 2024	**No. DE PÁGINAS:**	73
ÁREAS TEMÁTICAS:	Parqueadero Inteligente		
PALABRAS CLAVES/ KEYWORDS:	Control Inteligente, Arduino Mega, Sensores, Optimización de Espacios, Estacionamiento.		

RESUMEN:

El presente trabajo de integración curricular se basa en la formulación de un Sistema de Control Inteligente para el Estacionamiento del Canal y Radio UCSG a través de la utilización de Arduino Mega, este sistema conllevará a mejoras significativas en la gestión del estacionamiento. Se prevé que, al obtener información en tiempo real sobre la disponibilidad de espacios, el sistema logrará reducir la congestión, optimizar la asignación de lugares y enriquecer la experiencia global de estacionamiento para los usuarios. Se llevará a cabo una exhaustiva revisión de la literatura existente para recopilar información sobre sistemas de control de estacionamiento, tecnología Arduino, aplicaciones de sistemas inteligentes en entornos de estacionamiento. Además, se aspira que este proyecto de integración curricular contribuya en la necesidad de una distribución más efectiva de los espacios de estacionamiento. Tal sistema no solo mejoraría la experiencia de los usuarios, sino que además optimizaría el uso de los recursos disponibles.

ADJUNTO PDF:	☒ SI	☐ NO
CONTACTO CON AUTOR/ES:	**Teléfono:** +593-985353531	E-mail: anguloariel137@gmail.com
CONTACTO CON LA INSTITUCIÓN: COORDINADOR DEL PROCESO DE UTE	**Nombre:** Ing. Bohórquez Escobar, Celso Bayardo PHD.	
	Teléfono: +593- 995147293	
	E-mail: celso.bohorquez@cu.ucsg.edu.ec	

SECCIÓN PARA USO DE BIBLIOTECA	
Nº. DE REGISTRO (en base a datos):	
Nº. DE CLASIFICACIÓN:	
DIRECCIÓN URL (tesis en la web):	