

Universidad Nacional Pedro Henríquez Ureña
Facultad de Ciencias y Tecnología
Escuela de Informática

Modelado y simulación de un sistema ciberfísico para parqueo inteligente



Trabajo de Grado presentado por

Kisi Rafaela Paez Almonte

Franklin Antonio Frías Vallejo

para la obtención del grado Ingeniero en Sistemas Computacionales

Santo Domingo, D.N.

2021

AGRADECIMIENTOS

En primer lugar, le agradezco a Dios y a la Virgen de Guadalupe por iluminar mi camino a lo largo de todo este trayecto, por darme sabiduría, pero sobre todo por demostrarme una vez más que soy la niña de sus ojos por tantas bendiciones hacia mí. Gracias Dios.

A Doña Susana y Don Luis, gracias a ustedes por hacer realidad uno de mis sueños y por ayudarme a culminar mi carrera. De la misma manera, Dios le bendiga eternamente por toda la ayuda reciba de ustedes.

A mi familia, especialmente: mis padres Kusilandy Almonte y Rafael Paez, gracias por todo su amor y apoyo incondicional que me han brindado, sin ustedes hoy yo no fuera quien soy; mi abuela Ramona Muñoz por siempre estar para mí en todos los momentos y por motivarme cada día más a dar lo mejor de mí; Mi hermana Luz Paez por ser mi amiga, corregirme, pero también por apoyarme en todo este proceso. Gracias a mi familia porque todo lo que soy es gracias a ustedes, por estar conmigo en las buenas y en las malas. Este trabajo de grado es especialmente de ustedes.

A mis amistades, especialmente: Luz Pinales, Jatnna Polanco y Miguel Hernandez por estar para mí en todo momento, por su cariño y su amor hacía mí. Gracias por motivarme cada día para llegar a cumplir mi sueño de ser ingeniera.

Por último, agradezco a Franklin Frías por aceptar ser mi compañero de tesis, al director Ing. Ulises Agüero y nuestro asesor José Ramón Romero por guiarnos en todo este proceso, brindarnos su apoyo y conocimiento.

Kisi Rafaela Paez

Agradezco en primer lugar a Dios por haberme concedido el privilegio de llegar hasta este punto mi vida y carrera, proveyéndome de salud, inteligencia, sabiduría, fuerzas y ánimo para poder sobrellevar este compromiso de estudios el cual quise abandonar en muchas ocasiones.

A Kedren Skales, por convertirse en una segunda madre para mí y dedicarse a costear mis estudios en su totalidad, depositando toda su confianza de que podría convertirme en un buen profesional, razón por la que estaré eternamente agradecido, sin ella no habría podido cumplir con esta meta de mi vida; así como también, agradezco al ministerio STCHM y sus colaboradores porque ellos hicieron posible que Kedren me conociera y sea mi patrocinadora.

A mi familia, especialmente: mis padres Claudino Frías y Ursula Vallejo, gracias por todo su amor y apoyo incondicional, por su dedicación y porque a pesar de no tener los recursos necesarios para costear mis estudios, hicieron todo lo que estaba a su alcance para que yo estudiara y no pierda mi motivación; mi esposa Janibel Aquino, gracias por siempre estar para mí en todos los momentos y motivarme cada día más a dar lo mejor de mí, por su fiel compañía durante las largas noches de estudio durante estos años; a mi hijo Japheth Frias, por convertirse en una fuente de motivación para seguir adelante y por su gran alegría con la que me recibía todos los días al llegar a casa luego de un arduo día; mis hermanos Frangy y Randy por ser mis amigos entrañables y por su incondicional respeto.

A mis Maestros, especialmente a Eliseo Cabrera, Gabriel Villalona, Cesar Moya, José Romero y Juan Del Rosario por haber sido ejemplo e inspiración durante mi desarrollo como profesional y porque cada uno me enseñó algo en particular que marcó un antes y un después en mi vida personal, educativa y profesional; gracias mil.

Por último, pero no menos importante, agradezco a Kisi Paez por ser mi compañera de tesis, por su dedicación, proactividad y entusiasmo, al director Ing. Ulises Agüero y a nuestro asesor José Ramón Romero por guiarnos en todo este proceso, brindarnos su apoyo y conocimiento, sin el cual esto no sería posible.

Franklin Antonio Frías

RESUMEN

En el diario vivir, el uso de un vehículo es indispensable para poder tener comodidad, lo cual ayuda ahorrar tiempo, pero a la vez se puede complicar debido a que en algunas ocasiones se torna difícil encontrar un parqueo en un campus universitario. Por lo tanto, se debe poder efectuar la búsqueda de parqueo de manera precisa y rápida. Hoy en día la modelización y la simulación es un ejercicio necesario para hacer frente al análisis, pero también al diseño de sistemas con cierto grado de complejidad. Razón por la cual nos lleva a presentar este trabajo de grado.

El trabajo de grado que nos compete tiene como objetivo general demostrar el impacto que tiene el diseño de un sistema ciberfísico para un parqueo inteligente de un campus universitario por medio del modelado y simulación de este. Además, agregar la tecnología implementada en el concepto de Smart Cities.

Por último, este trabajo nos brinda la posibilidad de ofrecer más y mejores servicios a los usuarios, haciendo que a estos les resulte más efectivo el uso del tiempo, así como también la optimización de recursos. Una de las áreas donde podemos aplicar el concepto planteado anteriormente es lo que podemos denominar Parqueo Inteligente, permitiéndole a los usuarios la posibilidad de obtener de forma anticipada, información relacionada a la disponibilidad de parqueos, donde a la vez se pueda tomar decisiones más puntuales, oportunas y objetivas a la hora de buscar un parqueo disponible.

ABSTRACT

In everyday life, the use of a vehicle is indispensable to be comfortable, which helps to save time, but at the same time can be complicated because sometimes it is difficult to find a parking lot on a university campus. Therefore, it is necessary to be able to carry out the search of parking in a precise and fast way. Nowadays, modeling and simulation is a necessary exercise to face the analysis and design of systems with a certain degree of complexity. This is the reason why we present this paper.

The general objective of this thesis is to demonstrate the impact of designing a cyberphysical system for intelligent parking on a university campus through modeling and simulation. In addition, to add the technology implemented in the concept of Smart Cities.

Finally, this work gives us the possibility of offering more and better services to users, making them more effective in the use of time, as well as optimizing resources. One of the areas where we can apply the concept mentioned above is what we can call Intelligent Parking, allowing users the possibility of obtaining in advance, information related to the availability of parking and thus can make more timely and objective decisions when looking for an available parking.

INDICE DE TABLAS

Tabla 1 Cantidad de parqueos en la UNPHU.....	24
Tabla 2 Distancias de áreas de parqueo según la entrada al campus.....	25
Tabla 3 Cantidad de vehículos de entrada/salida al campus universitario según la hora	26
Tabla 4 Recursos de materiales esenciales para el sistema ciberfísico.....	62
Tabla 5 Recursos humanos para el sistema ciberfísico	63

INDICE DE ILUSTRACIÓN

Ilustración 1 Diseño del modelo del sistema ciberfísico	28
Ilustración 2 Diseño de componentes del modelo del sistema ciberfísico desde Teoría de Cola	30
Ilustración 3 Creación del modelo en la simulación.....	31
Ilustración 4 Definición de unidades de medida a utilizar en la simulación	31
Ilustración 5 Lectura de los datos del archivo .CSV en la simulación.....	32
Ilustración 6 Parámetros de configuración del campus en la simulación	33
Ilustración 7 Distribución de Poisson en la simulación	33
Ilustración 8 Método CreateModel en la simulación.....	34
Ilustración 9 Método Fill_data en la simulación	35
Ilustración 10 Método Step en la simulación	35
Ilustración 11 Método drive_from_Kennedy en la simulación.....	36
Ilustración 12 Método drive_from_Próceres en la simulación.....	37
Ilustración 13 Método road_to_Destination_Model en la simulación.....	38
Ilustración 14 Método road_to_Destination_Current en la simulación	39
Ilustración 15 Método car_leaving en la simulación.....	39
Ilustración 16 Método run_simulation_V2 en la simulación	40
Ilustración 17 Resultado de la ejecución del Método run_simulation_V2 en la simulación.....	41
Ilustración 18 Función cantidad de vehículos según el horario	42
Ilustración 19 Cantidad de vehículos (Con y sin parqueos) según el horario dentro de la UNPHU.....	43
Ilustración 20 Función cantidad de vehículos según el horario para cada área de parqueo	44

Ilustración 21 Cantidad de vehículos según el horario por cada área de parqueos dentro de la UNPHU	45
Ilustración 22 Función entradas de vehículos al campus por las diferentes puertas	46
Ilustración 23 Entrada de vehículos según las diferentes dos entradas y horario a la UNPHU	46
Ilustración 24 Función general de la comparación del sistema actual y propuesto	48
Ilustración 25 Función de la comparación del tiempo promedio para llegar a un parqueo disponible.....	48
Ilustración 26 Comparación del parqueo actual vs el propuesto según el tiempo en obtener un parqueo.	49
Ilustración 27 Diseño específico de integración del sistema ciberfísico e IoT en la realidad	53
Ilustración 28 Diseño general de integración del sistema ciberfísico e IoT en la realidad	54

ÍNDICE

CAPÍTULO 1	Introducción.....	2
1.1	Antecedentes.....	4
1.2	Definición del problema	6
1.3	Objetivos del Trabajo de Grado.....	6
1.3.1	Objetivos.....	6
1.3.2	Objetivos específicos	6
1.4	Justificación	7
1.4.1	Originalidad	7
1.4.2	Profundidad	8
1.4.3	Impacto.....	8
1.5	Alcance.....	9
1.6	Limitaciones.....	9
CAPÍTULO 2	Marco Teórico.....	10
2.1	Conceptos Generales.....	10
2.1.1	Sistema	10
2.1.2	Sistemas Ciberfísicos	10
2.1.3	Internet de las cosas (Iot)	11
2.1.4	Modelado	11
2.1.5	Validación de un modelo.....	12
2.1.6	Verificación de un modelo.....	12
2.1.7	Modelos de Simulación	12
2.1.8	Simulación	13
2.1.9	Smart City	15

2.1.10	Parqueo	15
2.1.11	Parqueo Inteligente	15
2.2	Conceptos de tecnologías utilizadas para el modelado y simulación	16
2.2.1	Python.....	16
2.2.2	Jupyter Notebook	16
2.2.3	Sympy	16
2.2.4	Función	17
2.2.5	Método	17
2.3	Conceptos de tecnologías utilizadas para posible implementación del sistema ciberfísico.....	17
2.3.1	Arduino	17
2.3.2	Red de sensores.....	18
2.3.3	Actuadores y/o sensores	18
2.3.4	Arduino IoT Cloud.....	19
2.3.5	Pantalla Led Smart Parking.....	19
2.3.6	API.....	19
CAPÍTULO 3	Marco metodológico.....	20
3.1	Diseño de la investigación	20
3.2	Técnica de recolección de datos	20
3.3	Técnica de análisis de datos	21
3.4	Herramientas de análisis de datos	22
3.5	Fórmulas de análisis de datos	23
CAPÍTULO 4	Resultados.....	24
4.1	Cumplimiento de objetivos	24

4.2	Concepción del sistema ciberfísico definiendo sus requerimientos en el contexto de campus de la UNPHU.	24
4.3	Modelo del sistema ciberfísico.	27
4.4	Diseño del modelo del sistema ciberfísico.	27
4.5	Diseño de componentes del modelo del sistema ciberfísico desde la perspectiva de Teoría de cola.	29
4.6	Simulación del sistema ciberfísico.	30
4.7	Resultado de la simulación del sistema ciberfísico.	42
4.8	Evaluación del modelo identificando ventajas, desventajas y oportunidades de mejora funcional.	49
4.9	Diseño del sistema ciberfísico para parqueo inteligente en la realidad.	50
4.9.1	Elementos del diseño del sistema ciberfísico en la realidad.	50
4.9.2	Mecanismo del diseño del sistema ciberfísico en la realidad.	51
4.9.3	Diseño específico y general de integración del sistema ciberfísico e IoT en la realidad.	53
4.9.4	Diseño pantalla led Smart Parking para el sistema ciberfísico en la realidad.	54
4.9.5	Diseño de la App móvil para el sistema ciberfísico en la realidad.	55
4.9.6	Pantalla del diseño de la App móvil para el sistema ciberfísico en la realidad.	56
4.9.6.1	Pantalla de inicio	56
4.9.6.2	Pantalla de parqueos según su edificio.	56
4.9.7	Diseño de la Página web para el sistema ciberfísico en la realidad.	57
4.9.8	Pantalla del diseño de la Página web para el sistema ciberfísico en la realidad.	57
4.9.8.1	Pantalla de inicio y reportes.	57
4.9.8.2	Pantalla de analítica y notificaciones	59

4.9.8.3	Pantalla de usuarios	60
4.10	Estudio de factibilidad	61
4.10.1	Factibilidad técnica	61
4.10.2	Factibilidad económica	61
4.10.3	Factibilidad operativa	62
CAPÍTULO 5	Conclusiones	64
CAPÍTULO 6	Recomendaciones	66
CAPÍTULO 7	Referencias.....	67
CAPÍTULO 8	Anexos.....	72
8.1	Preguntas de las entrevistas.....	72
8.2	Diagrama de actividad del sistema ciberfísico	73
8.3	Casos de uso	74
8.3.1	Caso de uso general.....	74
8.3.2	Caso de uso de consultar histórico - Administrador	75
8.3.3	Caso de uso de usuarios- Administrador.....	76

CAPÍTULO 1 INTRODUCCIÓN

Actualmente, en tiempos de constante avance y evolución tecnológica, la población ha crecido rápidamente, dando lugar a que el uso de vehículos aumente de la misma manera. Sin embargo, las calles, rutas e infraestructuras de movilidad vial no se han desarrollado a la misma velocidad. En grandes y desarrolladas ciudades, instituciones y estructuras, la congestión vehicular resulta un problema que obstaculiza el crecimiento, pero también el desarrollo.

Aunque dicho crecimiento y desarrollo poblacional ocasiona diversos problemas, existen iniciativas tecnológicas que pueden proporcionar posibles soluciones que contribuyan a solventar y abordar este peculiar problema. Esto es debido a que el desarrollo y avance tecnológico van a una gran velocidad, a tal punto de que la tecnología forma parte cada vez más de nuestra vida cotidiana.

Un sistema ciberfísico (Cyber-physical system, por sus siglas CPS) es la unión de un componente de software y parte de mecánicas o electrónicas. El control, la supervisión, la transferencia de datos y el intercambio de datos se realizan a menudo a través de Internet en tiempo real con ayuda de sistemas ciberfísicos (Mundo, 2015).

Por tanto, CPS aporta una visión que va más allá de objetos individuales ofreciendo servicios a través de Internet (Internet de las Cosas (IoT)), ya que aborda sistemas completos que se componen a partir de otros sistemas y que son capaces de sentir y controlar el mundo físico, entendiéndolo y aprendiendo de las interacciones que se producen, de forma que evolucionan hacia la creación de entornos inteligentes (Mundo, 2015).

Por otro lado, el modelado y la simulación son herramientas ampliamente usada en el diseño, evaluación y optimización de sistemas, sistemas logísticos de almacenamiento y transporte, sistemas de computadores y redes de comunicaciones, sistemas del sector servicios, sistemas de atención sanitaria y de respuesta ante emergencias, sistemas biológicos y ecológicos, sistemas económicos, entre otros.

La modelación y simulación hoy en día es importante ya que permite experimentar, pero a la vez ayuda a visualizar cómo se comporta un sistema y en qué área se pueden realizar mejoras en el funcionamiento del sistema basándonos en resultados obtenidos en la simulación.

Antes de empezar cualquier tipo de proyecto se debe de entender bien cómo es que funciona el sistema actual y cuáles son las variables involucradas para poder modelar el sistema. La manera más sencilla de que podamos determinar que el modelo sea válido es definiendo que las salidas del modelo están muy cerca de la realidad del sistema (Caliper, 2005).

Apoyados en los métodos actuales de sistemas ciberfísicos, modelado y la importancia de la simulación, se propone como meta presentar un trabajo de grado de modelado y simulación de un sistema ciberfísico de parqueo inteligente para un campus universitario, el cual consistirá en monitorear e informar en tiempo real la disponibilidad de parqueos. Además, nos brinda la posibilidad de ofrecer más y mejores servicios a los usuarios, haciendo que a estos les resulte más efectivo el uso del tiempo, así como también la optimización de recursos.

En conclusión, este presente trabajo de grado tiene el objetivo de presentar una simulación del comportamiento del parqueo actual en un campus universitario en diferentes escenarios y posterior a esto un análisis para poder determinar qué tan necesario e importante es la implementación de un sistema ciberfísico para parqueo inteligente utilizando tecnología de punta.

1.1 Antecedentes

Parqueadero inteligente mediante red de sensores inalámbricos (WSN) es un trabajo similar al que se va a realizar, lo cual consiste en el desarrollo e implementación de tecnología en lo que es la gestión vehicular donde se pueda tener una mejor calidad con respecto a los parqueos. En los últimos años, uno de los problemas que afecta a los usuarios es la falta de disponibilidad y búsqueda de parqueos, ya que puede reducir el tiempo perdido en encontrar un parqueo y reducir la contaminación. El parqueadero inteligente se desarrolla en tres plazas de parqueo, donde la información es obtenida mediante una red de sensores y enviada a un programa que muestre el uso de las plazas de parqueo (ESPE, 2014).

Otro trabajo semejante, es el diseño de un sistema de parqueo inteligente con aplicación móvil para mostrar espacios disponibles tiempo y valor a cancelar, donde se enfoca en el desarrollo de proveer herramientas ofimáticas para que los usuarios puedan a través de un smartphone realizar la búsqueda de un lugar donde aparcar su vehículo. Cabe destacar que debido al aumento exponencial de vehículos se ha llegado hasta el punto de que esta se ha visto ocupada en su totalidad, donde se tiene como consecuencia la gestión vehicular (Castro, 2017).

Otro similar, es el diseño de un prototipo de sistema de parqueo inteligente para el edificio de la FIE utilizando tecnologías basado en el Internet de las Cosas (Iot), un trabajo que consiste en el diseño de un sistema de parqueo inteligente para el edificio de la Facultad de Informática y Electrónica (FIE) de la Escuela Superior Politécnica de Chimborazo (ESPOCH) empleando tecnologías teniendo como base el Internet de las cosas. Este prototipo consiste en una red inalámbrica donde usa las tarjetas de desarrollo Arduino y Nodemcu, donde está conformado por dos nodos: el sensor y recolector de datos, que monitorean en tiempo real el estado de los parqueos de la FIE. En este prototipo se utilizaron sensores ultrasónicos para detectar la presencia de un vehículo, los cuales enviaran señales de ondas ultrasónicas al Arduino y para el envío de los datos hacia el nodo recolector es mediante bluetooth utilizando topología maestro-esclavo (Taípe, 2018).

También, el proyecto Integración de tecnologías RFID y WSN en un sistema de estacionamiento inteligente que es basado en la integración de tecnologías por radiofrecuencia (RFID) y red de sensores (WSN), donde el sistema pueda recopilar información sobre el estado de cada aparcamiento y a la vez pueda guiar a los usuarios al parqueo más cercano que se encuentre disponible mediante una aplicación. Dicha aplicación cuenta con un sistema de billetera electrónica basado en NFC para permitir a los usuarios pagar la tarifa del parqueo sin tener que perder tiempo (Palano, Mainetti, Patrono, Stefanizzi, & Vergallo, 2014).

Por otro lado, el proyecto Diseño e implementación de un parqueo inteligente utilizando Arduino y un basado en internet de las cosas (IoT) tiene como objetivo de mejorar el proceso a la hora de buscar lugares para que el usuario pueda parquearse sin tener que perder tanto tiempo. Este consiste en la utilización de sensores ultrasónicos que sirven para detectar vehículos y a su vez transmite datos de manera digital al Arduino con ayuda de la plataforma Temboo y así los usuarios puedan ver la disponibilidad de parqueos disponibles a través de Twitter (Lindao, 2016).

Por último, sistema de estacionamiento inteligente con sensores IoT que consiste en la implementación de una red inalámbrica con Arduino y Nodemcu, con un nodo sensor y otro recolector de datos que determinan el estado de espacios de estacionamiento en tiempo real, la información se transmite por WiFi a un servidor K2S01 (FL-US) y a través de una interfaz Smartphone, el usuario puede reservar una plaza disponible. Las pruebas realizadas determinaron un error absoluto de ± 1 cm en mediciones de distancia, el 1% en repetitividad de datos y un 48,5% de mejora con respecto a pérdidas de tiempo al momento de buscar espacios, una muestra de que es un sistema eficiente y confiable (Ramos Valencia, Vera, Cabrera, & Salazar, 2019).

1.2 Definición del problema

En base a lo visto en los antecedentes, uno de los problemas que se enfrentan hoy en día es la disponibilidad y búsqueda de parqueos, donde conlleva a la pérdida de tiempo, contaminación al medio ambiente, alto consumo de combustible y sobre todo retraso en la hora de llegada a su destino final dentro del campus universitario.

Otro punto para tratar es que en varios campus no se cuenta con un mecanismo automático para el control de entrada y salida de los vehículos, sino que es llevado a cabo por una persona física, quien se encuentra en la entrada independiente del clima, la hora y la presencia de vehículos. Esto a su vez dificulta conocer la población vehicular dentro del campus, la cual permita tomar mejores decisiones en el futuro en cuanto a este punto.

De manera puntual podemos identificar tres posibles problemas:

- 1- Los usuarios no saben con anticipación dónde encontrar un parqueo disponible dentro del campus.
- 2- Desconocimiento de la cantidad de vehículos dentro del campus.
- 3- No hay suficientes parqueos para todos.

Partiendo de esto, para cubrir las necesidades ya expuestas se propone el modelado y simulación de un sistema ciberfísico de parqueo inteligente para un campus universitario.

1.3 Objetivos del Trabajo de Grado

1.3.1 Objetivos

Modelar y simular un sistema ciberfísico de parqueo para el campus de la UNPHU.

1.3.2 Objetivos específicos

- Concebir el sistema ciberfísico definiendo sus requerimientos en el contexto de campus de la UNPHU.
- Modelar el sistema ciberfísico.

- Simular el sistema ciberfísico.
- Evaluar el modelo identificando ventajas, desventajas y oportunidades de mejora funcional.

1.4 Justificación

Hoy en día un parqueo inteligente permite agilizar el tiempo y esfuerzo para poder parquear un vehículo. Además, ayuda a que los usuarios puedan con anticipación encontrar un parqueo disponible.

En la sección a continuación se definirán los elementos que justifican el desarrollo del proyecto, teniendo en cuenta la originalidad, la profundidad y el impacto del proyecto.

1.4.1 Originalidad

Los elevadores de parqueos son un sistema hidráulico con doble pistón que elevan los vehículos con el objetivo de optimizar el espacio dentro de un estacionamiento, duplica la capacidad con sistemas confiables, estables, duraderos y que requieren poco mantenimiento. Hoy en día en nuestro país es utilizado el concepto de parqueo inteligente o, mejor dicho, los mencionados Smart Parking que se encuentran ubicados en lugares estratégicos de Santo Domingo, donde a través de esta herramienta de parqueos les permite a los usuarios poder estacionarse mediante llave de acceso con un costo desde RD\$ 100 pesos la hora.

En base a lo visto en los antecedentes, pero también algunas investigaciones de trabajo de grado relacionados a parqueo inteligente y simulación, tales como:

- Diseño de un sistema de reserva de parqueo inteligente en la Universidad del Norte (De Lima Freidel, 2017).
- Diseño del despliegue de redes de sensores para parqueo inteligente (Valdivieso Armendariz, 2017).
- Diseño e implementación de un parqueo inteligente utilizando Arduino Yun (Rosales Lindao, 2016).

Se obtuvo como conclusión, de que trata solamente sobre la implementación de estacionamiento inteligente, sin embargo, ninguno ha contado con incluir simulaciones para verificar que tan factible es, ante de poner en producción su sistema. Por consiguiente, la originalidad de este trabajo de grado consiste en el modelado y simulación de un sistema ciberfísico de parqueo inteligente para un campus universitario, donde los usuarios puedan reducir su tiempo y visualizar la cantidad disponible de parqueos de manera anticipada, pero a la vez que tan factible sería la ejecución de este.

Un sistema ciberfísico para un parqueo inteligente es el enfoque en el ámbito de Smart City más popular y adoptada con mayor rapidez en todo el mundo. El objetivo principal es reducir el tiempo de búsqueda y optimizar los lugares de estacionamiento disponibles.

1.4.2 Profundidad

El diseño incluye el modelado y simulación de un sistema ciberfísico que podrá ser utilizado en un parqueo inteligente de un campus universitario.

El proyecto será tratado en varias etapas, las cuales son, el modelado y simulación de un sistema ciberfísico para parqueo inteligente. La simulación y el modelado incluyen tanto el funcionamiento actual de los parqueos de la UNPHU, así como la simulación del sistema ciberfísico con el fin de comparar la eficiencia e interoperabilidad de estos, pero a la vez poder ver que tan factible sería poder contar con un parqueo inteligente. Además, está limitado al entorno que comprende a los terrenos del campus de la Universidad Nacional Pedro Henríquez Ureña (UNPHU).

1.4.3 Impacto

Este trabajo puede beneficiar a instituciones, empresas y organizaciones donde requieran un parqueo inteligente para poder reducir la congestión vehicular. Además, a los usuarios por el hecho de que podrán saber con anticipación la disponibilidad de parqueos en tiempo real, permitiendo que puedan ahorrar tiempo en busca de un parqueo que posiblemente no encuentren, y así poder tomar decisiones oportunas.

Cabe destacar, que las entidades administrativas son beneficiadas en base a la data capturada por el sistema y así puedan tomar mejores decisiones para optimizar constantemente procesos que involucren flujo y distribución vehicular. Así como un aporte tecnológico significativo para el perfil y prestigio de la universidad u otra institución que implemente dicha solución.

1.5 Alcance

La propuesta a presentar dará a conocer una modelización y simulación de parqueo inteligente. Por otro lado, para el modelado y simulación se utilizará el campus de la UNPHU.

Este diseño de sistemas constará con lo siguiente elementos:

- Modelado y simulación del funcionamiento actual del parqueo de la UNPHU.
- Modelado y simulación del parqueo inteligente.
- Comparación en gráficas entre el parqueo actual y el parqueo propuesto.

Cabe destacar, que no se hará una implementación funcional real del sistema ciberfísico, solo una simulación utilizando el lenguaje Python.

1.6 Limitaciones

Este trabajo de grado presenta las siguientes limitantes:

- El sistema ciberfísico no asegura disponibilidad de parqueo para todos los usuarios.
- El sistema ciberfísico no controla el aparcamiento en zonas prohibidas o no destinadas a ese fin.
- El sistema ciberfísico no controla el desbordamiento de la cantidad de vehículo en el campus, ni el flujo vehicular dentro del campus.
- El diseño de la página web del sistema ciberfísico solo tendrá las pantallas más esenciales, que son: Inicio y reportes, Analítica y notificaciones, usuarios.

CAPÍTULO 2 MARCO TEÓRICO

En esta sección se definirán los conceptos y componentes relacionados al modelo y simulación de un sistema ciberfísico para un parqueo inteligente.

2.1 Conceptos Generales

2.1.1 Sistema

Es un conjunto de elementos relacionados entre sí y que funcionan como un todo. En base a esto podemos definir que un sistema informático (SI) es un sistema que permite almacenar y procesar información. En poca palabra, un sistema es el conjunto de partes interrelacionadas: hardware, software y personal informático (Marcelo & Osorio, 1998).

2.1.2 Sistemas Ciberfísicos

Los sistemas ciberfísicos consisten en incorporar componentes de un mundo físico con elementos de cómputo, almacenamiento remoto de información y redes de comunicación (parte cibernética). Estos sistemas interactúan a diferentes escalas de tiempo y espacio, con múltiples y diversos modos de operación y capacidades de configuración/reconfiguración. Además, los sistemas ciberfísicos deben tener la capacidad de comunicarse con otros sistemas, así como tener en cuenta información del contexto que facilite el aprendizaje colectivo (Nunes, 2018).

Hoy en día los sistemas ciberfísicos brindan interconexión, adaptabilidad y seguridad para poder interactuar con otros sistemas. Por otro lado, el mayor reto que plantean estos sistemas es demostrar su fiabilidad, seguridad y robustez en todas sus posibles situaciones, sobre todo en entornos críticos y en los que la respuesta debe ser en tiempo real (Gandhi, 2020).

Los ejemplos de sistemas ciberfísicos incluyen al sistema de red eléctrica inteligente, sistemas de automóvil autónomo, sistemas de monitoreo médico, sistemas de control del proceso, sistemas de robótica, domótica y pilotos automáticos aeronáuticos.

Además, el sistema ciberfísico implica un enfoque multidisciplinario, fusionando la teoría de cibernética, mecatrónica y la ciencia de diseño y de proceso (School, 2017).

2.1.3 Internet de las cosas (IoT)

El Internet de las cosas (IoT) consiste en tener la mayoría de los objetos estén permanentemente conectados a Internet para mejorar su funcionalidad, entre otras posibles utilidades, incluyendo aquellas que van más allá de un uso particular (Azahara, 2017).

Es una arquitectura emergente basada en la Internet global donde facilita el intercambio de bienes y servicios entre redes de la cadena de suministro, pero a la vez tiene un impacto importante en la seguridad y privacidad de los actores involucrados (Jordi Salazar, 2016).

La IoT trae beneficios de mejora de la gestión, el seguimiento de los activos y de los productos, aumenta la cantidad de datos de información y permite la optimización de equipos y uso de los recursos que puede traducirse en ahorro de costes. Además, ofrece la oportunidad de crear nuevos dispositivos interconectados inteligentes y explorar nuevos modelos de negocio (Jordi Salazar, 2016).

2.1.4 Modelado

El modelado ayuda y permite analizar problemas complejos del mundo real con el fin de predecir lo que podría suceder con algún curso de acción. En pocas palabras el modelado es una representación de un sistema real (Villalba & Moraleda, 2013).

El modelado del sistema permite establecer el límite de información entre el sistema que se está implementando y el entorno en que va a operar, es decir, es la descripción del sistema con todos los elementos esenciales para poder en base a eso realizar la simulación (Downey, 2017).

Un modelo proporciona los planos de un sistema. Pueden existir modelos muy detallados, así como modelos que den un vistazo de alto nivel del sistema que se construirá. Además, cada sistema puede ser descrito desde diferentes perspectivas

utilizando diferentes modelos, pero de la misma forma, puede ser estructural enfatizando la organización del sistema, o puede ser de conducta, enfatizando los aspectos dinámicos. En resumen, el modelo es una simplificación de la realidad (Booch, Rumbaugh, & Jacobson, 1999).

2.1.5 Validación de un modelo

La validación comprueba la precisión de la representación del modelo del sistema real. Por otro lado, la validación del modelo se define como "prueba de que un modelo computarizado dentro de su dominio de aplicabilidad posee un rango satisfactorio de precisión consistente con la aplicación prevista del modelo". (Sargent, 2011).

Un modelo debe construirse para un propósito específico o un conjunto de objetivos y su validez debe determinarse para ese propósito, donde a su vez implica garantizar que el modelo de simulación desarrollado represente correctamente el sistema real como objeto de estudio (Pérez C. , 2019).

2.1.6 Verificación de un modelo

La verificación es el proceso de confirmar que está correctamente implementado con respecto al modelo conceptual (coincide con las especificaciones y supuestos considerados aceptables para el propósito de aplicación dado). Además, durante la verificación, el modelo se prueba para encontrar y corregir errores en la implementación del modelo (Pérez C. , 2019).

El objetivo de la verificación del modelo es garantizar que la implementación del modelo sea correcta. De la misma forma, se utilizan varios procesos en conjunto de técnicas para asegurar que el modelo coincida con las especificaciones y supuestos con respecto al concepto del modelo (Sargent, 2011).

2.1.7 Modelos de Simulación

Existen varios tipos de modelo de simulación, entre los cuales son:

- **Modelo de simulación estático:** Es la representación de un sistema en un tiempo en particular, es decir, reproducir un sistema en el que tiempo no es importante (Law & Kelton, 1991).

- Modelo de simulación dinámico: Es la representación de un sistema que va evolucionando durante un tiempo. En pocas palabras, es reproducir un sistema en el que tiempo es una variable de interés (Law & Kelton, 1991).
- Modelo determinístico: Este modelo es aquel que no contiene variables aleatorias y obtiene una respuesta conocida mediante la entrada de una variable determinada (Law & Kelton, 1991).
- Modelo estocástico: Este modelo es aquel que contiene una o más variables aleatorias que varían con el tiempo, pero que a la vez evolucionan en función de otra variable (Law & Kelton, 1991).
- Modelo de simulación continua: Son aquellos que se representan prolongadamente en el tiempo. Este tipo de modelo se caracteriza por tener un rango de tiempo predeterminado (Law & Kelton, 1991).
- Modelo de simulación discreta: Este modelo representa fenómenos en donde las cantidades varían moderadamente sobre el tiempo (Law & Kelton, 1991).

2.1.8 Simulación

La simulación es una metodología poderosa para el diseño y análisis de sistemas complejos. El enfoque general de la simulación por computadora es representar la dinámica característica de un sistema del mundo real en un modelo. Las simulaciones son adecuadas para problemas en que no hay soluciones analíticas de forma cerrada. Ya que la mayoría de los problemas dinámicos en la práctica no puede ser representada y resuelta completamente usando ecuaciones matemáticas, la computadora. (Coss Bu, 2003).

La simulación es el proceso de diseñar y desarrollar un modelo computarizado de un sistema o proceso y conducir experimentos con el modelo con el propósito de entender el comportamiento del sistema o evaluar varias estrategias con las cuales se puede operar el sistema (Shannon, 1988).

Para calcular probabilidades de eventos complejos se puede recurrir a simular los experimentos, es decir, a realizar un experimento más sencillo o accesible que sea equivalente al original. Además, para estimar las probabilidades buscadas se debe

repetir muchas veces el experimento equivalente y calcular la frecuencia relativa de los eventos que se desean analizar. La simulación es experimentar qué sucedería en el futuro, siguiendo un determinado modelo (Maass & Anderson, 1985).

Los elementos de un modelo de la simulación pueden ser:

- Entidades: Representan los elementos que conforman el sistema tales como: Personas, maquinas, transporte, entre otro (Guasch & Figueras, 2005).
- Actividades: Son las distintas acciones que tienen lugar en el sistema e involucran a las entidades tanto temporales como permanentes (Guasch & Figueras, 2005).
- Eventos: Se define como una acción instantánea que se realiza en el sistema y que conlleva a un cambio en las variables de estado (Guasch & Figueras, 2005).

La simulación es muy importante en la Ingeniería de Sistemas porque presenta las siguientes ventajas en el diseño de estos:

- A través de un estudio de simulación, se puede estudiar el efecto de cambios internos y externos del sistema, al hacer alteraciones en el modelo del sistema y observando los efectos de esas alteraciones en el comportamiento del sistema (García, García, & Cárdenas, 2006).
- La técnica de simulación puede ser utilizada para experimentar con nuevas situaciones, sobre las cuales tiene poca o ninguna información. A través de esta experimentación se puede anticipar mejor a posibles resultados no previstos (García, García, & Cárdenas, 2006).
- Cuando nuevos elementos son introducidos en un sistema, la simulación puede ser usada para anticipar cuellos de botella o algún otro problema que puede surgir en el comportamiento del sistema (García, García, & Cárdenas, 2006).

2.1.9 Smart City

El concepto surgió originalmente vinculado al uso de las TIC como herramientas de gestión eficiente. Puede convenirse que la aplicación de las TIC para mejorar la calidad de vida de sus habitantes y asegurar un desarrollo económico, social y ambiental sostenible son elementos comunes a todas las definiciones (Ontiveros, Vizcaíno, & Lopez, 2017).

Por otra parte, el concepto exige una nueva relación con ciudadanos, turistas, proveedores y trabajadores públicos basada en la transparencia, la rendición de cuentas, la interacción y el diálogo, el adecuado uso y consumo de los recursos y la identificación temprana de necesidades. Hoy el concepto es más amplio y ambicioso, asociando el atributo Smart no solo a su dimensión digital, sino a los atributos deseables de sostenibilidad, eficiencia, participación, innovación, gobernanza e inclusión social. Ello, contemplando ámbitos tan fundamentales y diversos en el área urbana como la movilidad y el transporte, la eficiencia energética, la planificación urbanística, la gestión de residuos y la contaminación ambiental (aire, agua, suelo y/o acústica), la generación de valor económico y el bienestar social de los ciudadanos, entre otros (Ontiveros, Vizcaíno, & Lopez, 2017).

Smart City es aquella que utiliza el potencial de la tecnología y la innovación, junto al resto de recursos, para promover de manera más eficiente un desarrollo sostenible y, en definitiva, mejorar la calidad de vida de sus ciudadanos (Fernández, 2016).

2.1.10 Parqueo

Se refiere especialmente hablando de los recintos o lugares reservados para estacionar vehículos. Por otro lado, se conoce como parqueo al espacio físico donde se deja el vehículo por un tiempo indeterminado cualquiera o al acto de dejar inmovilizado un vehículo (Escobar, Moncada, & Urazán, 2016).

2.1.11 Parqueo Inteligente

El parqueo inteligente tiene como objetivo ayudar a los usuarios encontrar un parqueo a través de tecnología de manera rápida y precisa (Cardenas & Lemos, 2017).

La idea inicial de parqueo inteligente fue propuesta hace diez años debido al rápido aumento de teléfonos inteligentes, ya tienen la capacidad de poder conectarse a la internet y así ayuda a buscar información fuera de casa. Además, de que este concepto es la primera solución para disminuir lo que nos afecta hoy en día, que es la congestión vehicular (Cardenas & Lemos, 2017).

2.2 Conceptos de tecnologías utilizadas para el modelado y simulación

2.2.1 Python

Es un lenguaje de programación creado por Guido van Rossum, se caracteriza por ser dinámico y es usado en diversas plataformas, por lo cual permite crear no solo sitios sino aplicaciones en una amplia variedad de sistemas. Python tiene la capacidad de trabajar con grandes volúmenes de datos. Es un lenguaje de scripting independiente de plataforma y orientado a objetos, en el cual se puede realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado, lo que significa que no se necesita compilar el código fuente para poder ejecutarlo, lo que ofrece ventajas como la rapidez de desarrollo e inconvenientes como una menor velocidad (NETEC, 2000).

2.2.2 Jupyter Notebook

Es una aplicación web de código abierto que le permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Los usos incluyen: limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos, aprendizaje automático y entre otros (Jupyter, 2015).

2.2.3 Sympy

Es un módulo que permite fácilmente que se creen modelos simulados de eventos discretos donde su lenguaje de programación se basa en Python. El módulo provee un modelador con componentes de una simulación modelo en la cual se incluyen procesos tanto para componentes activos como clientes, mensajes, y vehículos y recursos como

para componentes pasivos. Además, provee variables en el monitor que permiten recolectar estadísticas (Iglesias Torre, 2016).

2.2.4 Función

Una función en simulación es un grupo de instrucciones con un objetivo en particular y que se ejecuta al ser llamada desde otra función o procedimiento. Una función puede llamarse múltiples veces e incluso llamarse a sí misma. La función describe una secuencia de órdenes, donde estas órdenes cumplen con una tarea específica de una aplicación más grande. Por otro lado, las funciones pueden recibir datos desde afuera al ser llamadas a través de los parámetros y deben entregar un resultado (Saafigueroa, 2017).

2.2.5 Método

Un método en simulación se define como un conjunto de instrucciones englobadas dentro de una sola instrucción, la cual se manda llamar cuando es necesario realizar una tarea. Cabe destacar, que un método puede ser un procedimiento o una función, la diferencia es que le pertenece a una clase (Saafigueroa, 2017).

2.3 Conceptos de tecnologías utilizadas para posible implementación del sistema ciberfísico.

2.3.1 Arduino

Es una placa hardware (Placa de circuito) libre que incorpora un microcontrolador reprogramable y una serie de pines-hembra (Los cuales están unidos internamente a las patillas E/S del microcontrolador) que permiten conectar allí de forma muy sencilla diferentes sensores y actuadores. Los microcontroladores son circuitos integrados en los que se pueden grabar instrucciones, las cuales las escribes con el lenguaje de programación deseado en el entorno Arduino IDE (Torrente, 2013).

En la actualidad, Arduino se ha utilizado en miles de proyectos y aplicaciones diferentes. Además, existe un software Arduino que funciona en Mac, Windows y Linux donde los profesores, pero también los estudiantes lo utilizan para construir instrumentos científicos de bajo costo, para probar los principios de la química, la física, o para comenzar con la programación y la robótica (Arduino, 2018).

2.3.2 Red de sensores

Es una red con varios dispositivos distribuidos espacialmente, donde utilizan sensores para poder controlar diversas condiciones, entre ellas la temperatura, el sonido, la vibración, la presión y movimiento o los contaminantes. Los dispositivos son unidades autónomas que constan de un microcontrolador, una fuente de energía, un radiotransceptor y un elemento sensor (Aakvaag, 2006).

Los sensores son dispositivos electrónicos con la capacidad de detectar la variación de una magnitud física tales como temperatura, iluminación, movimiento y presión; y de convertir el valor de ésta, en una señal eléctrica ya sea analógica o digital (Caluva, 2005).

2.3.3 Actuadores y/o sensores

Es un dispositivo con la capacidad de generar una fuerza que ejerce un cambio de posición, velocidad o estado de algún tipo sobre un elemento mecánico, a partir de la transformación de energía. En pocas palabras, un actuador es un dispositivo que convierte la energía en movimiento o que se utiliza para aplicar fuerza. El dispositivo toma energía de una determinada fuente (que puede ser energía creada por aire, líquido o electricidad) y la convierte en el movimiento deseado (Becerra, 2016).

Los sensores son dispositivos electrónicos con la capacidad de detectar la variación de una magnitud física tales como temperatura, iluminación, movimiento y presión; y de convertir el valor de ésta, en una señal eléctrica ya sea analógica o digital (Caluva, 2005).

2.3.4 Arduino IoT Cloud

Arduino IoT Cloud permite conectar dispositivos a Internet y a otros instrumentos. Por consiguiente, es una plataforma de aplicación de Internet de las cosas (IoT) fácil de usar. Además, hace que sea muy sencillo para cualquiera desarrollar, pero también ayuda a gestionar aplicaciones de IoT, lo que permite centrarse en la resolución de problemas reales en el negocio o en la vida cotidiana. Esta herramienta hace que la creación de objetos conectados sea rápida, sencilla y segura (Becerra, 2016).

2.3.5 Pantalla Led Smart Parking

Son pantallas que consiste en ayudar a los usuarios a ver la disponibilidad de los parqueos en un determinado lugar de manera rápida y legible (Redrovan Castillo & Pizarro Ochoa, 2015).

2.3.6 API

Es un conjunto de herramientas, definiciones y protocolos que se utiliza para integrar los servicios y el software de aplicaciones. Además, permite que los productos y servicios se comuniquen con otros, sin tener que diseñar permanentemente una infraestructura de conectividad nueva (Hat, 2000).

Las API pueden ser privadas (para uso interno únicamente), compartidas (con partners específicos para brindar flujos de ingresos adicionales) o públicas (entidades externas pueden desarrollar aplicaciones que interactúen con sus API para fomentar la innovación) (Hat, 2000).

El propósito de las API es la integración, es decir, se encargan de conectar los datos, las aplicaciones y los dispositivos de la organización para que todas las tecnologías puedan comunicarse y trabajar mejor en conjunto. Cuando las tecnologías no pueden comunicarse entre sí o con otras tecnologías, se puede perder tiempo y dinero. Las API, junto con la integración distribuida y los contenedores, son una característica fundamental de la integración ágil (Hat, 2000).

CAPÍTULO 3 MARCO METODOLÓGICO

El siguiente estudio tiene el objetivo de generar evidencia para que se pueda identificar las ventajas, desventajas y oportunidades de mejora funcional con respecto al modelado y simulación de sistema ciberfísico de un parqueo inteligente.

3.1 Diseño de la investigación

Para el desarrollo del presente trabajo y dar solución al problema planteado, es necesario realizar una investigación en un marco de estudios de campo, debido a la necesidad de recopilar información directamente desde el objeto de estudio, en este caso, el campus universitario UNPHU, este estudio de investigación es de carácter cuantitativo.

La investigación cuantitativa implica el uso de herramientas informáticas, estadísticas, y matemáticas para obtener resultados. Es concluyente en su propósito ya que trata de cuantificar el problema y entender qué tan generalizado está mediante la búsqueda de resultados proyectables a una población mayor.

3.2 Técnica de recolección de datos

Para desarrollar esta investigación con respecto a la recopilación de la información fueron la observación y la entrevista, las cuales se especifican a continuación:

❖ **Observación:** Por medio de la observación se recolectan los datos que permiten la aplicación y el desarrollo del modelo a utilizar en la simulación. Durante el proceso de observación fue necesario el uso del cronómetro como instrumento para la medición de los siguientes tiempos:

- Tiempo que toma un vehículo desde que pasa la entrada hasta su parqueo destino.
- Tiempo de retraso al poder entrar o salir del campus una vez ubicados en punto de acceso.

- Tiempo de concurrencia con que entra y sale un vehículo al campus.
- ❖ **Entrevista:** Este instrumento de recopilación de información a utilizar nos sirvió para el desarrollo y descripción de las operaciones y funciones llevadas a cabo por el personal en el modelo de paqueos del campus, así como información de origen cuantitativo a partir de las experiencias y conocimientos de los entrevistados, dichas entrevistas son realizadas directamente a empleados del área de parqueos y de puntos de acceso vehicular al campus, a usuarios y a personal de seguridad. La información aportada por las entrevistas sirve de guía para el desarrollo de la descripción de estas operaciones del sistema actual y el desarrollo y descripción de un sistema propuesto mejorado.

3.3 Técnica de análisis de datos

Para realizar el estudio de los datos es necesario mencionar los modelos con el que es posible visualizar diferentes escenarios de acuerdo con la estructura del sistema de parqueos actual y aportar conclusiones y comparaciones objetivas con el sistema propuesto mediante gráficas como respuesta de la simulación. Por eso, los modelos para analizar los datos del presente trabajo de grado serán:

- ❖ **Modelo Estocástico Discreto:** Se caracteriza por los cambios que ocurren en el modelo mediante la presencia de un evento ocurrido aleatoriamente en base a una probabilidad (Rincón, 2011).
- ❖ **Modelo Cibernético:** Este modelo consiste en enfocar e identificar como está compuesta la funcionalidad y estructura de un recurso dando así una solución tecnológica (Pérez I. , 2016).
- ❖ **Teoría de cola:** Es el estudio de las colas y los procesos aleatorios que las caracterizan. Se trata de dar sentido matemático a los escenarios de la vida real. Por ejemplo, una multitud de vehículos haciendo cola para entrar a un campus universitario (Abad, 2002).

En resumen, esto será posible mediante la aplicación de un modelo Estocástico Discreto de índole cibernético en conjunto de la teoría de cola debido a que los cambios en el modelo sólo se darán cuando ocurran los eventos de entrada/salida de vehículos y liberación /ocupación de un parqueo, donde dichos eventos se producirán al azar mediante alguna probabilidad de ocurrencia, la cual será obtenida de los datos colectados.

3.4 Herramientas de análisis de datos

Para realizar la simulación en conjunto con las técnicas expuestas anteriormente. La herramienta para utilizar es:

- **Sympy:** Es un módulo que permite fácilmente que se creen modelos simulados de eventos discretos donde su lenguaje de programación se basa en Python (Iglesias Torre, 2016).
- **Jupyter Notebook:** Es una aplicación web de código abierto que le permite crear y compartir documentos que contienen código en vivo, ecuaciones, visualizaciones y texto narrativo. Los usos incluyen: limpieza y transformación de datos, simulación numérica, modelado estadístico, visualización de datos, aprendizaje automático y entre otros (Jupyter, 2015).
- **Visual Studio:** Es un conjunto de herramientas y otras tecnologías de desarrollo de software basado en componentes para crear aplicaciones eficaces y de alto rendimiento (Noticias, 2000).

Por último, con estas herramientas mencionadas anteriormente se observará la simulación del parqueo actual versus el parqueo inteligente propuesto mediante gráficas utilizando lenguaje Python. Además, lo factible que sería la implementación del presente trabajo de grado en un campus universitario.

3.5 Fórmulas de análisis de datos

Para realizar el análisis de los datos y la simulación en conjunto con las técnicas y herramientas expuestas anteriormente. Las fórmulas para utilizar son:

- **Tiempo:** Es una magnitud que sirve para medir la duración de uno o más eventos. Esto permite ordenarlos en una secuencia (pasado, presente, futuro) y determinar si ocurren o no en simultáneo (Santaolalla, 2017).

$$T = \frac{V}{D}$$

- **Distribución Poisson:** Se caracteriza por ser una distribución de probabilidad discreta que se puede aplicar a los sucesos de algún acontecimiento durante un intervalo de tiempo determinado (Arroyo, Bravo, & Muñoz, 2014).

$$P(x) = \frac{\mu^x \cdot e^{-\mu}}{x!}$$

CAPÍTULO 4 RESULTADOS

4.1 Cumplimiento de objetivos

En este capítulo se plasman los resultados de los objetivos generales y específicos, como se llevó a cabo la simulación del parqueo actual y el propuesto, ventajas, desventajas y oportunidades de mejora del presente trabajo de grado. A continuación, se muestra como fueron cumplidos los objetivos de este trabajo de grado:

4.2 Concepción del sistema ciberfísico definiendo sus requerimientos en el contexto de campus de la UNPHU.

Se concibió el sistema ciberfísico definiendo lo siguiente como requerimiento para realizar el modelo y la simulación de este:

- **La cantidad de parqueos por cada edificio.**

Se realizó un conteo de parqueos según al edificio al que corresponda para poder tener un control de la cantidad de parqueos que pueden estar disponibles o no al momento de realizar la simulación del sistema ciberfísico. En la siguiente tabla se puede observar el lugar del campus (Edificio) con su respectiva cantidad de parqueos de lateral derecho (Entrada al parqueo), lateral izquierdo (Salida del parqueo) y reservados (Parqueos para personas embarazadas o discapacitado).

Lugar del campus	Cantidad de Parqueos Lateral Izquierdo (LI)	Cantidad de Parqueos Lateral Derecho (LD)	Parqueos Reservados	Total de parqueos del lugar del campus.
Edificio 1 y Edificio 2	63	60	3	126
Campo Football Viejo	47	25	0	72
Edificio 3	52	66	3	121
Edificio 4	17	53	6	76
Edificio 5 (Auditorio)	84	12	0	96
Edificio 8 y Edificio 6	44	188	6	238
Edificio 7	28	0	0	28
Veterinaria/Odontología	107	0	0	107
			Total de Parqueos:	864

Tabla 1 Cantidad de parqueos en la UNPHU.

- **La distancia del área de parqueo referente a las entradas.**

Se realizó la medición desde las dos diferentes entradas al campus (Av. John F. Kennedy y Av. Los próceres) hacia cada área de parqueo correspondiente a su edificio para poder determinar la probabilidad de la entrada/salida del usuario según su elección en la simulación. En la siguiente tabla se puede observar el lugar del campus (Edificio) con su respectiva distancia de la entrada al parqueo que consta solamente de lateral derecho (Entrada al parqueo) debido que a partir de la medición se sumará 1 metro hasta llegar al último parqueo de dicho edificio, ya que es la medida de un parqueo para un vehículo. Cabe destacar que algunos edificios tienen 2 entradas al área de parqueo.

Lugar del campus	Entrada 1 (John F. Kennedy)		Entrada 2 (Av. Los Proceres)	
	Distancia de Parques Lateral Derecho (LD) 1	Distancia de Parques Lateral Derecho (LD) 2	Distancia de Parques Lateral Derecho (LD) 1	Distancia de Parques Lateral Derecho (LD) 2
Edificio 1 y Edificio 2	209.62 m		598.71 m	604.10 m
Campo Football Viejo	520.60 m		680.84 m	
Edificio 3 (Biblioteca)	409.74 m		416.50 m	
Edificio 4	375.00 m	409.60 m	351.01m	
Edificio 5 (Auditorio)	580.85 m		338.81 m	
Edificio 8 y Edificio 6	624.13 m	646.51m	272.89 m	272.64 m
Edificio 7	625.56 m		543.57 m	
Veterinaria/Odontología	621.51 m		834.36 m	

Tabla 2 Distancias de áreas de parqueo según la entrada al campus.

- **Cantidad de vehículos de entrada/salida al campus universitario según la hora.**

Se realizó una mensuración de la cantidad de vehículos que puede entrar o salir por minutos del campus universitario según la hora del día como se muestra en la siguiente tabla y así poder saber a través de una gráfica de comparación mediante la simulación que tan viable es el sistema ciberfísico propuesto en el presente trabajo de grado.

Horas	Cantidad de vehículos de entrada	Unidad de tiempo (Minutos) de entrada de vehículos	Cantidad de vehículos de salida	Unidad de tiempo (Minutos) de salida de vehículos
7:00 AM	1	1	1	5
8:00 AM	2	1	1	5
9:00 AM	2	1	1	10
10:00 AM	1	2	2	5
11:00 AM	1	3	3	1
12:00 PM	2	1	2	5
13:00 PM	1	3	2	1
14:00 PM	1	1	1	1
15:00 PM	3	5	1	5
16:00 PM	2	1	1	1
17:00 PM	7	1	1	5
18:00 PM	7	1	2	1
19:00 PM	3	1	2	1
20:00 PM	2	5	2	1
21:00 PM	1	5	5	1
22:00 PM	1	4	3	2
23:00 PM	0	0	9	2

Tabla 3 Cantidad de vehículos de entrada/salida al campus universitario según la hora

- **Indicadores que afectan la entrada/salida del campus.**

Los indicadores que se tomaron en cuenta para el modelado y la simulación del sistema ciberfísico fueron:

- El clima.
- El horario (Horas pico).
- Graduaciones.
- Eventos especiales.

Estos indicadores afectan el parqueo actual, pero en el sistema ciberfísico propuesto el clima y el horario no afecta el tiempo de entrada/ salida del campus debido a que al entrar o salir del campus será de manera automática por lo que se evita colas de vehículo.

Por otro lado, teniendo como base los puntos anteriores para la concepción del sistema ciberfísico, se tuvo en cuenta:

- La entrada/salida del campus que sea de manera automática para poder reducir el tiempo del usuario.
- Los usuarios puedan visualizar los parqueos disponibles o no acorde a cada edificio de manera anticipada al entrar al campus.
- Visualización de la cantidad de vehículos dentro del campus.
- Indicadores (Tiempo) que afecten el parqueo propuesto.

4.3 Modelo del sistema ciberfísico.

Para la realización del modelo del sistema ciberfísico se tomaron en cuenta dos enfoques, estos con diferentes escenarios para la simulación. Los enfoques son:

- **Enfoque para Parqueo actual:** Consiste en representar y analizar el sistema actual del parqueo de la UNPHU con el fin de tomar decisiones al comparar con el parqueo propuesto.
- **Enfoque para parqueo inteligente propuesto:** Consiste en representar un sistema de parqueo en el cual intervengan dispositivos ciberfísicos inteligentes con el fin de analizar la factibilidad de este en el parqueo de la UNPHU mediante el presente trabajo de grado.

4.4 Diseño del modelo del sistema ciberfísico.

El diseño del modelo del sistema ciberfísico para el parqueo se realiza tomando en base la teoría de cola, modelo estocástico discreto y modelo cibernético para poder realizar la simulación en sus respectivos escenarios y obtener un resultado de lo que actualmente existe versus lo propuesto en este trabajo de grado. Dentro del diseño del modelo se encuentra los siguientes elementos:

- **Población:** Corresponde a los usuarios que hacen uso del parqueo de la UNPHU.
- **Proceso de llegada:** Corresponde a la probabilidad de elegir un destino para ingresar al campus de la UNPHU.

- **E1 Vehículos:** Corresponde a la probabilidad de ingresar a la UNPHU por la entrada de la Av. John F. Kennedy.
- **E2 Vehículos:** Corresponde a la probabilidad de ingresar a la UNPHU por la entrada de la Av. Los Próceres.
- **Tiempo de espera:** Corresponde al tiempo en que dura el usuario para otorgarle el acceso al campus teniendo en cuenta los indicadores que pueden afectar.
- **Mecanismo de servidores (Parqueos):** Corresponde al tiempo de búsqueda de parqueo y cambio de decisiones ante disponibilidad de parqueo.
- **Proceso de salida:** Corresponde a los usuarios que hicieron uso del parqueo de la UNPHU.

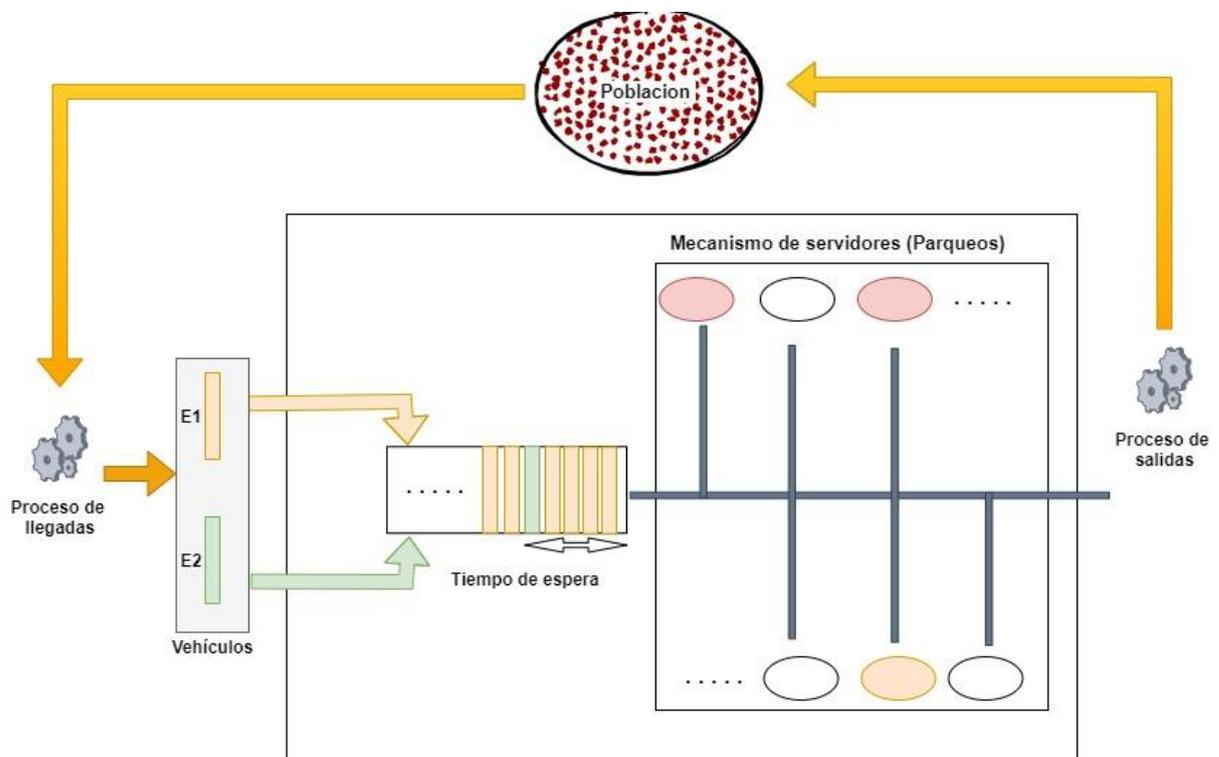


Ilustración 1 Diseño del modelo del sistema ciberfísico

4.5 Diseño de componentes del modelo del sistema ciberfísico desde la perspectiva de Teoría de cola.

El diseño de componentes del modelo del sistema ciberfísico fue realizado en base a teoría de cola para realizar la simulación y saber que tan satisfactorio es el presente trabajo de grado. Los componentes del modelo ciberfísico son los siguientes:

- **Generador de entidades:** Se encarga de generar las entidades que formarán parte del sistema para nuestro modelo de teoría de colas, apoyándonos en el proceso de llegada de Poisson a partir del número de ocurrencias de la llegada de un vehículo en un determinado momento.
- **Cola RSS (Random Selection of Service):** Bloque que se encarga de implementar la disciplina de la cola, la cual es de selección aleatoria tanto para la ocupación y liberación del servidor, ya que no depende del orden de llegada para que las entidades sean servidas; debido a ello, el enfoque de la disciplina es en el tiempo que toma un vehículo en llegar a un parqueo disponible, independientemente de su orden de llegada, porque la elección del servidor es a elección del conductor y puede que esto afecte en su tiempo de servicio.
- **Multiple Servidores:** Modela los servidores encargados de atender las necesidades de las entidades, en este caso, vehículos, la cual comprende como entrada los vehículos provenientes de la cola.
- **Salida de entidades:** Simula el proceso de liberación de servidores basados en la tasa de ocurrencia de dicho evento para un momento dado.
- **Inicializador de Ciclo:** Maneja el proceso de iteraciones para cada entidad generada en un tiempo dado.

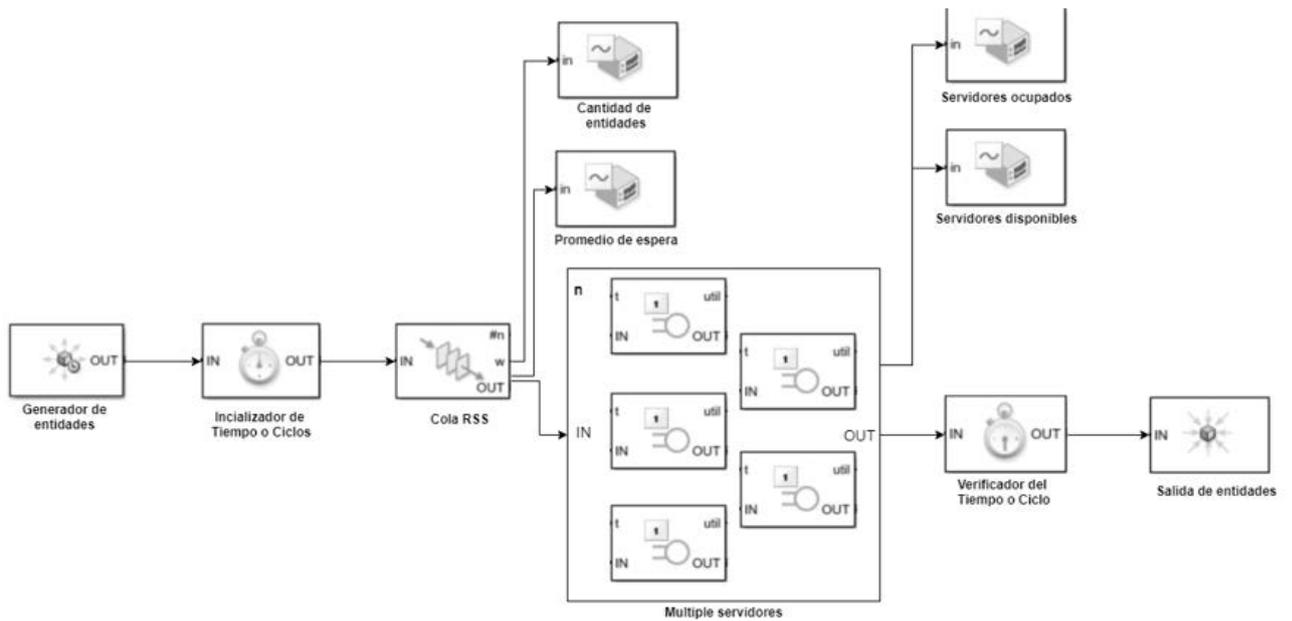


Ilustración 2 Diseño de componentes del modelo del sistema ciberfísico desde Teoría de Cola

4.6 Simulación del sistema ciberfísico.

Para realizar la simulación del sistema ciberfísico, se tomaron en cuenta los siguientes parámetros, métodos y funciones:

- **Creación del modelo**

Se crearon los siguientes objetos representativos del modelo:

1. UNPHU: Representa el contexto de la simulación
2. ParkingLot: Modelo del sistema de parqueos de cada edificio
3. Parking: Representa el espacio de estacionamiento de un vehículo
4. Entry: Representa La entrada al campus universitario.

```

In [3]: ParkingState = State(
        parkingName = "",
        distance_Kennedy = 0,
        distance_Proceres = 0,
        location_Name = "",
        isBusy = False
    )

    ParkingLotState = State(
        parkingLotname = "",
        parkingQty = 0,
        Parkings = [],
        busyQty = 0,
        availableQty = 0,
        resultInTime = TimeSeries()
    )

    EntryState = State(
        entryName = "",
    )

    )

    UNPHU = State(
        parkingLots = [],
        kennedy = {},
        proceres = {},
        current_count = 0,
        empty_tries_count = 0
    )

```

Out[3]:

	values
parkingLots	[]
kennedy	{}
proceres	{}
current_count	0
empty_tries_count	0

Ilustración 3 Creación del modelo en la simulación

- **Definición de las diferentes unidades de medida a utilizar en la simulación.**

Se definieron las unidades necesarias para realizar la simulación del sistema ciberfísico para poder visualizar los resultados de estos en las gráficas.

```

In [4]: km = UNITS.kilometer
        hour = UNITS.hour
        meter = UNITS.meter
        minute = UNITS.minute
        second = UNITS.second
        count = UNITS.count

```

Out[4]: count

Ilustración 4 Definición de unidades de medida a utilizar en la simulación

- **Lectura de los datos del archivo .CSV en la simulación.**

Se realizó la lectura de los datos del archivo .CSV que fueron recolectados para realizar la concepción del sistema ciberfísico teniendo en cuenta los requerimientos en el contexto del campus de la UNPHU.

```
In [5]: E2_E1 = pd.read_csv('data/parkings_build1_and_2.csv')
E3 = pd.read_csv('data/parkings_build3.csv')
E4 = pd.read_csv('data/parkings_build4.csv')
Auditory = pd.read_csv('data/parkings_build_auditory.csv')
E7 = pd.read_csv('data/parkings_build7.csv')
E8_E6 = pd.read_csv('data/parkings_build6_and_8.csv')
FootballField = pd.read_csv('data/parkings_football_field.csv')
Veterinary = pd.read_csv('data/parkings_veterinary.csv')

# For E2
E2_E1.index = E2_E1.No

# For E3
E3.index = E3.No

# For E4
E4.index = E4.No

# For Auditory
Auditory.index = Auditory.No

# For E7
E7.index = E7.No

# For E8_E6
E8_E6.index = E8_E6.No

# For football field
FootballField.index = FootballField.No

# For Veterinary
Veterinary.index = Veterinary.No
```

Ilustración 5 Lectura de los datos del archivo .CSV en la simulación

- **Parámetros de configuración del campus en la simulación.**

En este se estableció los parámetros de entrada predeterminados, que definen las reglas del sistema de parqueo, las cuales son:

1. Velocidad máxima dentro del campus.
2. Promedio de entrada de vehículos.
3. Probabilidad de acceso por las puertas.
4. Tasa de entrada/salida de vehículos según horario.

```

In [6]: car_flow_rate = pd.read_csv('data/car_flow_rate.csv')
car_flow_rate.index = car_flow_rate.time

t_0 = get_first_label(car_flow_rate.entry_total)
t_end = get_last_label(car_flow_rate.entry_total)

systemParams = System(
    t_0=t_0,
    t_end=t_end,
    max_speed = 15,
    avg_entry_rate = 5 * count / minute,
    avg_entry_proceees_rate = 1 * count / (5 * minute),
    avg_entry_kennedy_rate = 4 * count / minute,
    max_speed_reduction_rate = 0.4,
    count_parking = 864
)

```

Out[6]:

	values
t_0	7
t_end	23
max_speed	15
avg_entry_rate	5.0 count / minute
avg_entry_proceees_rate	0.2 count / minute
avg_entry_kennedy_rate	4.0 count / minute
max_speed_reduction_rate	0.4
count_parking	864

Ilustración 6 Parámetros de configuración del campus en la simulación

- **Distribución de Poisson en la simulación.**

La distribución de Poisson recoge sucesos independientes que ocurren en un soporte continuo. El número medio de sucesos por unidad de soporte se le conoce como λ y caracteriza la distribución. Poisson nos permite crear distribuciones de este tipo.

```

In [7]: #Probando el uso de la distribución de Poisson integrada en Python
X = ss.poisson(4)
X.cdf(2)

```

Out[7]: 0.23810330555354436

Ilustración 7 Distribución de Poisson en la simulación

- **Método CreateModel en la simulación.**

Es usado para la creación del modelo utilizado por la aplicación que representa las áreas de parqueos, recibiendo como parámetro un objeto con los datos extraídos del archivo .CSV que contiene la información recolectada del campus, el nombre del modelo y si contiene múltiples rutas para llegar a este a través de las dos entradas (E1 y E2).

```
In [8]: def createModel(lotName, file_data, multipleK = False, multipleP = False):  
        """ Creates the different parking for each parking area  
  
        LotName: Parking area name  
        file_data: Each parking info based on de csv file  
        """  
  
        lot = State(parkingLotName = lotName, parkingQty = 0, parkings = [], busyQty = 0, availableQty = 0, hasKennedyMultiple= multipleK, hasProceresMultiple = multipleP, resultInTime = TimeSeries())  
        parkings = []  
        #lot.parkingLotName = lotName  
        lot.parkingQty = len(file_data)  
        lot.availableQty = lot.parkingQty  
        for i in range(len(file_data)):  
            parking = State(  
                parkingName= file_data.Name.values[i],  
                distance_Kennedy= file_data.From_Kennedy_E1.values[i]/1000,  
                distance_Proceres=file_data.From_Proceces_E1.values[i]/1000,  
                location_Name= file_data.side.values[i],  
                isBusy = False)  
            parkings.append(parking)  
        lot.parkings = parkings  
        return lot
```

Ilustración 8 Método CreateModel en la simulación

- **Método Fill_data en la simulación.**

Este método es usado para inicializar el estado del modelo en general, es decir, darle vida al modelo, para empezar la simulación; esto ocurre cargando información que ha sido previamente tabulada en un archivo .CSV y utilizando nuestro método CreateModel que construimos previamente.

```

In [10]: def fill_data():
        """ Fill all model data to start each system componet state
        """
        UNPHU.parkingLots.append(createModel("E2_E1", E2_E1, False, True))
        UNPHU.parkingLots.append(createModel("E3", E3))
        UNPHU.parkingLots.append(createModel("E4", E4, True, False))
        UNPHU.parkingLots.append(createModel("Auditory", Auditory))
        UNPHU.parkingLots.append(createModel("E8_E6", E8_E6, True, True))
        UNPHU.parkingLots.append(createModel("E7", E7))
        UNPHU.parkingLots.append(createModel("Veterinary", Veterinary))
        UNPHU.parkingLots.append(createModel("FootballField", FootballField))
        UNPHU.kennedy = State( entryName = "Kennedy", countQty = 0)
        UNPHU.proceres = State( entryName = "Proceres", countQty = 0)

In [11]: #Call the fill_data method
        fill_data()

In [12]: global_SimulationInfo = State(cars_in = 0, cars_out =0, kennedy = 0, proceres
        = 0)

Out[12]:

```

	values
cars_in	0
cars_out	0
kennedy	0
proceres	0

Ilustración 9 Método Fill_data en la simulación

- **Método Step en la simulación.**

Este método es usado para simular una unidad de tiempo o un instante en la simulación, para cada vez que ocurra un evento de entrada o salida de vehículos al sistema; recibiendo como parámetro las probabilidades de que, en un instante cualquiera, un vehículo entre por la Av. Próceres o Av. John F. Kennedy.

```

In [13]: def step(state, p1, p2, useProposed = True):
        """Simulate one minute of time.

        state: parkings Lot State object
        p1: probability of a car arrival from Kennedy entrance
        p2: probability of a car arrival from Proceres entrance
        """

        if flip(p1):
            return drive_from_kennedy(state, useProposed)

        #if flip(p2):
            return drive_from_proceres(state, useProposed)

```

Ilustración 10 Método Step en la simulación

- **Método `drive_from_Kennedy` en la simulación.**

Este método es usado para simular el proceder de un vehículo cuya entrada es por la Av. John F. Kennedy; este necesita como parámetro, el modelo y los parámetros de configuración que definen el sistema actual o el propuesto.

```
In [14]: def drive_from_kennedy(state, usePropose):
    """Move one car from kennedy gate to the destination parking.

    state: unphu state with the List of parkings Lot as a state object
    """
    time_in_road = 0
    state.kennedy.countQty += 1
    #get parking Lots with available parkings
    notFullParkingLots = list(filter(lambda pl: pl.availableQty > 0, state.parkingLots))
    if(len(notFullParkingLots) == 0):
        state.empty_tries_count += 1
        #print('vacio')
        return time_in_road
    #Find a random number to select a Location
    locationIndex = randint(0, len(notFullParkingLots)-1)
    #Now We select a parking area from the Lot
    selectedParkingLot = [notFullParkingLots[index] for index in [locationIndex]][0]
    #call road method
    if(usePropose):
        time_in_road = road_to_Destination_Model(selectedParkingLot, True, False)
    else:
        time_in_road = road_to_Destination_Current(selectedParkingLot, True, False)

    state.kennedy.countQty += 1
    state.current_count += 1
    return time_in_road
```

Ilustración 11 Método `drive_from_Kennedy` en la simulación

- **Método `drive_from_Próceres` en la simulación.**

Es utilizado para simular el proceder de un vehículo cuya entrada es por la Av. Los Próceres. En este método se llevan a cabo las decisiones que tomará el conductor en base a la disponibilidad de parqueos y a donde se dirige.

```

In [15]: def drive_from_proceres(state, useProposed):
          """Move one car from proceres gate to the destination parking.

          state: unphu State object
          """
          time_in_road = 0
          state.proceres.countQty += 1

          #get parking lots with available parkings
          notFullParkingLots = list(filter(lambda pl: pl.availableQty > 0, state.parkingLots))
          if(len(notFullParkingLots) == 0):
              state.empty_tries_count += 1
              #print('vacio')
              return time_in_road
          #Find a random number to select a location
          locationIndex = randint(0, len(notFullParkingLots)-1)
          #Now we select a parking area from the lot
          selectedParkingLot = [notFullParkingLots[index] for index in [locationIndex]][0]
          #call road method
          if(useProposed):
              time_in_road = road_to_Destination_Model(selectedParkingLot, False, False)
          else:
              time_in_road = road_to_Destination_Current(selectedParkingLot, False, False)

          state.proceres.countQty += 1
          state.current_count += 1
          return time_in_road

```

Ilustración 12 Método drive_from_Proceres en la simulación

- **Método road_to_Destination_Model en la simulación.**

Este método es usado para el sistema propuesto con el objetivo de simular la trayectoria de un vehículo una vez haya entrado al sistema, afectando así el estado del modelo en base a las decisiones tomada previamente por el usuario, donde se puede calcular el tiempo de manejo desde el punto de entrada hasta destino final dentro del campus, apoyándonos en los parámetros de configuración establecidos dentro el campus de la UNPHU e información obtenida en entrevistas a usuario del parqueo.

```

In [16]: def road_to_Destination_Model(parkingLotState, isKennedy = True, hasMultipleEn
trance = False):
    """
    parkingLotState: The selected parking area destination
    """
    time_in_road = 0
    isFull = True
    #Selecting the first available parking
    parking = next ((p for p in parkingLotState.parkings if p.isBusy == False
), [])
    if(len(parking) > 0):
        if(isKennedy):
            time_in_road = (parking.distance_Kennedy /15) * 60
        else:
            time_in_road = (parking.distance_Proceres /15) * 60

        parking.isBusy = True
        parkingLotState.availableQty -= 1
        parkingLotState.busyQty += 1
    # else:
    #     print("Vacio")
    return time_in_road

```

Ilustración 13 Método `road_to_Destination_Model` en la simulación

- **Método `road_to_Destination_Current` en la simulación.**

Este método es usado para el sistema actual con el objetivo de simular la trayectoria de un vehículo una vez haya entrado al sistema, afectando así el estado del modelo en base a las decisiones tomada previamente por el usuario, donde podremos calcular el tiempo de manejo desde el punto de entrada hasta destino, apoyándonos en las reglas de conducción establecidas para el campus de la UNPHU e información obtenida en entrevistas a usuario del parqueo.

```

In [17]: def road_to_Destination_Current(parkingLotState, isKennedy = True, hasMultiple
Entrance = False):
    """
    parkingLotState: The selected parking area destination
    """
    time_in_road = 0
    isFull = True
    speed = systemParams.max_speed - (systemParams.max_speed * systemParams.ma
x_speed_reduction_rate)
    #Selecting the first available parking
    parking = next ((p for p in parkingLotState.parkings if p.isBusy == False
), [])
    if(len(parking) > 0):
        if(isKennedy):
            time_in_road = (parking.distance_Kennedy /speed) * 60
        else:
            time_in_road = (parking.distance_Proceres /speed) * 60

        parking.isBusy = True
        parkingLotState.availableQty -= 1
        parkingLotState.busyQty += 1
    return time_in_road

```

Ilustración 14 Método road_to_Destination_Current en la simulación

- **Método car_leaving en la simulación.**

Este método es usado para simular el proceso de salida de vehículos, en base al patrón de salidas el cual fue definido mediante la técnica de investigación y muestreo, obteniendo así el número de veces que se espera que ocurra el fenómeno durante un intervalo dado al igual que se utiliza para las entradas.

```

In [18]: def car_leaving(state, qty):
    for i in range(qty):
        #select parkingLots with busy parking
        parkingLotsWithCars = list(filter(lambda pl: pl.busyQty > 0, state.parkingLots))
        if(len(parkingLotsWithCars) == 0):
            return
        locationIndex = randint(0, len(parkingLotsWithCars)-1)
        #Now We select a parking area from the Lot to let leave a car
        selectedParkingLot = [parkingLotsWithCars[index] for index in [locationIndex]][0]
        #now We select only busy parkings
        busyParkings = list(filter(lambda p: p.isBusy, selectedParkingLot.parkings))
        locationIndex = randint(0, len(busyParkings)-1)
        selectedParking = [busyParkings[index] for index in [locationIndex]][0]

        selectedParking.isBusy = False
        selectedParkingLot.busyQty -= 1
        selectedParkingLot.availableQty += 1
        state.current_count -= 1

```

Ilustración 15 Método car_leaving en la simulación

- **Método run_simulation_V2 en la simulación.**

Es usado para simular el proceso de entradas de vehículos, en base al patrón de entrada el cual fue definido mediante la técnica de investigación y muestreo, obteniendo así el número de veces que se espera que ocurra el fenómeno durante un intervalo dado, apoyándonos del método Step para representar ese momento dado; siendo este el punto inicial que desencadena la intercomunicación y ejecución de los procedimientos creados previamente que le dan vida a todo el desarrollo. Este nos devuelve la información total que es relevante para hacer las comparaciones de lugar.

```
In [19]: #Function use to simulate de nuev Model
def run_simulation_V2(state, p1, p2, system, useProposed = True):
    """Simulate the given number of time steps.

    state: parkingLot State object
    p1: probability of a car arrival from Kennedy entrance
    p2: probability of a car arrival from Procedes entrance
    num_steps: number of time steps
    """
    emptyCountSum = 0
    avgTimeGlobal = 0
    carsCount = 0
    avgTime = 0
    resultAvailableCount = TimeSeries()
    resultEmptyCount = TimeSeries()
    resultEntries = TimeSeries()
    resultCurrentCount = TimeSeries()
    resultAvgTimeInRoad = TimeSeries()
    resultsKennedy = TimeSeries()
    resultsProceres = TimeSeries()
    for t in linnrange(system.t_0, system.t_end + 1):
        for i in range(car_flow_rate.entry_total[t]):
            time = step(state, p1, p2, useProposed)
            avgTime += time
            avgTimeGlobal += time
            car_leaving(state, car_flow_rate.leaving_total[t])
            resultCurrentCount[t] = state.current_count
            resultEntries[t] = state.kennedy.countQty + state.proceres.countQty
            resultAvgTimeInRoad[t] = avgTime/car_flow_rate.entry_total[t]
            avgTime = 0
            carsCount += car_flow_rate.entry_total[t]
            resultsKennedy[t] = state.kennedy.countQty
            resultsProceres[t] = state.proceres.countQty
            resultEmptyCount[t] = state.empty_tries_count
            emptyCountSum += state.empty_tries_count
            state.empty_tries_count = 0
            resultAvailableCount[t] = sum(p.availableQty for p in state.parkingLot
s)
            for index in range(len(state.parkingLots)):
                state.parkingLots[index].resultInTime[t] = state.parkingLots[index
].busyQty

    return State(
        emtyTries = emptyCountSum,
        resultAvailableCount = resultAvailableCount,
        resultEmptyTries = resultEmptyCount,
        avgTimeGlobal = avgTimeGlobal/carsCount,
        entries = resultEntries,
        current = resultCurrentCount,
        AvgTimeInRoad = resultAvgTimeInRoad,
        resultsKennedy= resultsKennedy,
        resultsProceres = resultsProceres)
```

Ilustración 16 Método run_simulation_V2 en la simulación

- **Resultado de la ejecución del Método run_simulation_V2 en la simulación.**

En este se procedió a ejecutar el método run_simulation_V2, pasándole los parámetros necesarios para llevar a cabo su tarea, donde nos retorna los resultados en un objeto, el mismo es ejecutado dos veces con parámetros diferentes para simular el sistema actual y el propuesto para poder realizar la comparación entre los dos sistemas.

```
In [20]: result = run_simulation_V2(UNPHU, 0.7, 0.3, systemParams)
result
```

```
Out[20]:
```

	values
emptyTries	221
resultAvailableCount	7 815 8 707 9 593 10 587 11 ...
resultEmptyTries	7 0 8 0 9 0 10 0 11 ...
avgTimeGlobal	1.71895
entries	7 120 8 360 9 600 10 660 11...
current	7 48 8 156 9 270 10 276 11 ...
AvgTimeInRoad	7 1.759733 8 1.963900 9 1.932533 1...
resultsKennedy	7 90 8 252 9 420 10 458 11...
resultsProceres	7 30 8 108 9 180 10 202 11 ...

```
In [21]: resultCurrent = run_simulation_V2(UNPHU, 0.7, 0.3, systemParams, False)
resultCurrent
```

```
Out[21]:
```

	values
emptyTries	235
resultAvailableCount	7 801 8 693 9 579 10 573 11 ...
resultEmptyTries	7 0 8 0 9 0 10 0 11 ...
avgTimeGlobal	2.81471
entries	7 3453 8 3693 9 3933 10 3993 11...
current	7 62 8 170 9 284 10 290 11 ...
AvgTimeInRoad	7 2.955667 8 3.140556 9 3.204889 1...
resultsKennedy	7 2443 8 2617 9 2783 10 2825 11...
resultsProceres	7 1010 8 1076 9 1150 10 1168 11...

Ilustración 17 Resultado de la ejecución del Método run_simulation_V2 en la simulación

4.7 Resultado de la simulación del sistema ciberfísico.

Para visualizar el resultado de la simulación del sistema ciberfísico y mostrar que tan factible es el presente trabajo de grado, se tomaron en cuenta los siguientes escenarios:

- **Escenario 1: Probabilidad cantidad de vehículos según el horario (Vehículos con y sin parqueos).**

Este escenario en la simulación muestra la cantidad de vehículos en el campus de la UNPHU para cada hora hábil del día, en respuesta al patrón de llegada de vehículos determinado por la tasa de entrada/salida según horario. Además, de los vehículos que pueden o no quedarse sin parqueo según el horario. A continuación, se mostrará la función que se realizó para este escenario en la simulación y el resultado mediante gráficas de la simulación:

Función del escenario en la simulación

La siguiente función presenta la población existente dentro del campus para cada hora hábil del día.

```
In [23]: def show_carsQty_by_time(carCountByTime, emptyTries = 0):
plt.figure(figsize=(10, 5))
plot(carCountByTime, ':', label='Vehículos')
plot(emptyTries, '--', label='Vehículos sin parqueo')

decorate(xlabel='Hora',
ylabel='Cantidad de vehículos',
title='Vehículos por horario dentro de la UNPHU')

In [24]: #Car population grouped by time method call
show_carsQty_by_time(result.current, result.resultEmptyTries)
```

Ilustración 18 Función cantidad de vehículos según el horario

Resultado del escenario en la simulación mediante gráfica

En el resultado del presente escenario se puede observar, que tanto en el parqueo actual, como en el parqueo propuesto se contemplan vehículos que no encontraron parqueos disponibles, por lo que la distribución de los alrededores del campus puede estar organizada de diversas formas, las cuales no son controladas en el alcance del presente trabajo de grado.

En la siguiente gráfica, se puede observar que el flujo mayor de vehículos en el campus de la UNPHU se encuentra entre las 16:00 PM y 21:00 PM con un máximo estimado de 550 vehículos que encontraron un parqueo, mientras que, entre las 17:00 PM y 20:00 PM con un máximo estimado de 150 vehículos no hallaron parqueos.

Por otro lado, hay un factor clave que repercute de manera diferente que es el tiempo perdido de los usuarios que no encuentran parqueos, debido a que en el contexto del sistema propuesto, los usuarios podrán saber de antemano si hay o no parqueos disponibles, reubicándose de manera anticipada perdiendo la menor cantidad de tiempo posible, por el hecho de no merodear por el campus innecesariamente en busca de parqueos y a la vez se evita una congestión vehicular; por otro lado, en el sistema actual, aquellos usuarios que no encuentren parqueos, lo sabrán una vez realicen un recorrido en el campus completo para posteriormente reubicarse en alguna área dentro o fuera del mismo, invirtiendo una gran cantidad de tiempo en el proceso.

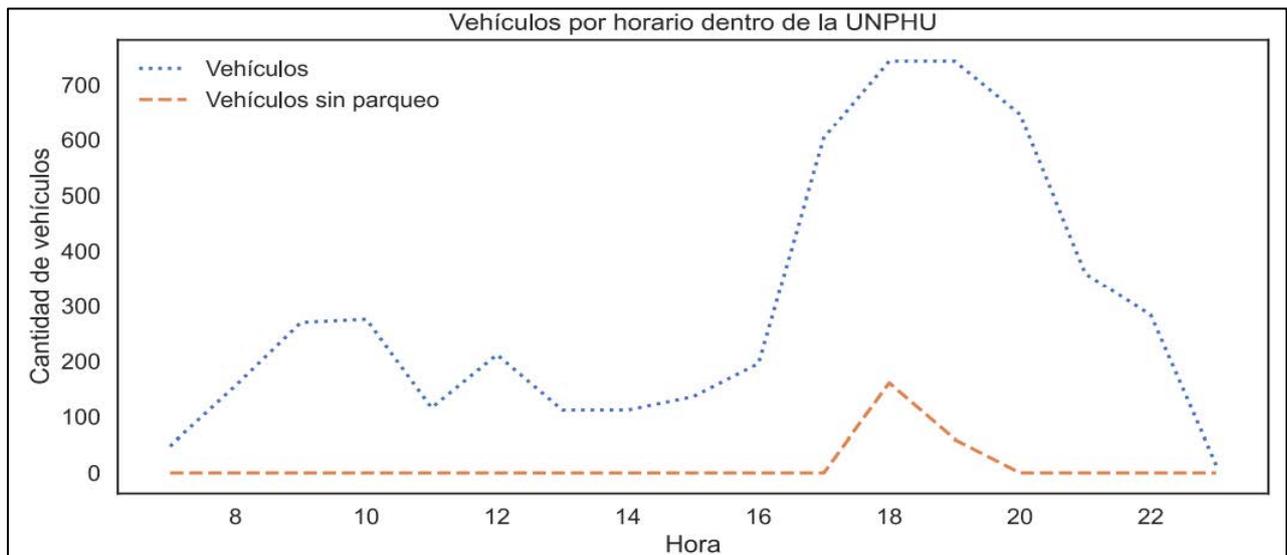


Ilustración 19 Cantidad de vehículos (Con y sin parqueos) según el horario dentro de la UNPHU

- **Escenario 2: Cantidad de vehículos según el horario para cada área de parqueo.**

En este escenario respetando el patrón de la gráfica anterior en la simulación, se muestra la distribución de la cantidad total de vehículos dentro del campus. Además, se puede ver un patrón similar al presentar una gráfica con la cantidad total de vehículos distribuida entre las diferentes áreas de parqueos dentro del campus de la UNPHU. A continuación, se mostrará la función que se realizó para este escenario en la simulación y el resultado mediante gráficas de la simulación:

Función del escenario en la simulación

La siguiente función presenta la población existente dentro del campus distribuida entre las áreas de parqueos por cada hora hábil de día.

```
In [25]: def show_cars_by_building(state):

    E2_E1_Result = next ((pl for pl in state.parkingLots if pl.parkingLotName
== "E2_E1"), [])
    E3_Result = next ((pl for pl in state.parkingLots if pl.parkingLotName ==
"E3"), [])
    E4_Result = next ((pl for pl in state.parkingLots if pl.parkingLotName ==
"E4"), [])
    Auditory_Result = next ((pl for pl in state.parkingLots if pl.parkingLotName
== "Auditory"), [])
    E8_E6_Result = next ((pl for pl in state.parkingLots if pl.parkingLotName
== "E8_E6"), [])
    E7_Result = next ((pl for pl in state.parkingLots if pl.parkingLotName ==
"E7"), [])
    Veterinary_Result = next ((pl for pl in state.parkingLots if pl.parkingLot
Name == "Veterinary"), [])
    FootballField_Result = next ((pl for pl in state.parkingLots if pl.parking
LotName == "FootballField"), [])
    plt.figure(figsize=(10, 5))
    plot(E2_E1_Result.resultInTime, '-', label = "E2_E1")
    plot(E3_Result.resultInTime, '-', label = "E3")
    plot(E4_Result.resultInTime, '-', label = "E4")
    plot(Auditory_Result.resultInTime, '-', label = "Auditory")
    plot(E8_E6_Result.resultInTime, '-', label = "E8_E6")
    plot(E7_Result.resultInTime, '-', label = "E7")
    plot(Veterinary_Result.resultInTime, '-', label = "Veterinary")
    plot(FootballField_Result.resultInTime, '-', label = "FootballField")

    decorate(xlabel='Hora',
            ylabel='Cantidad de vehiculos',
            title='Vehículos por horario para cada edificio dentro de la
UNPHU')
```

Ilustración 20 Función cantidad de vehículos según el horario para cada área de parqueo

Resultado del escenario en la simulación mediante gráfica

En la siguiente gráfica se puede ver que la máxima cantidad de vehículos dentro de la UNPHU se encuentra en los edificios E2_E1 y E8_E6 entre las 18:00 PM y 22:00 PM, mientras que la mínima cantidad de vehículos se localiza en el E7.

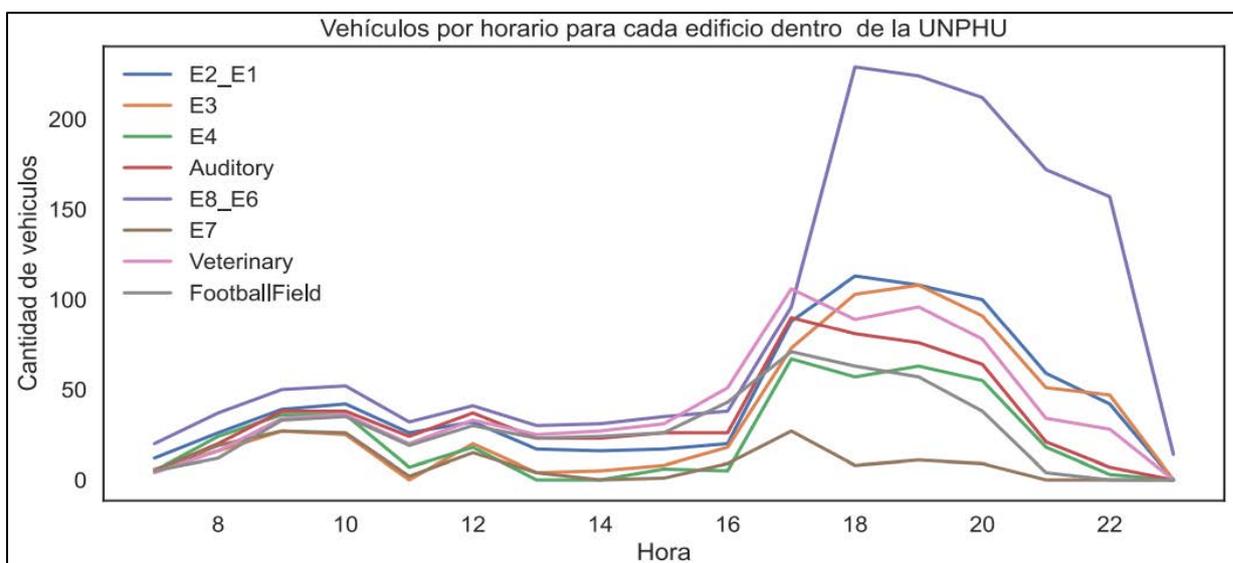


Ilustración 21 Cantidad de vehículos según el horario por cada área de parqueos dentro de la UNPHU

- **Escenario 3: Cantidad total de entradas de vehículos al campus por las diferentes puertas.**

En este escenario de la simulación muestra la cantidad de vehículos que entran al campus de la UNPHU por las diferentes puertas según la hora (Av. Los próceres y Av. John F. Kennedy). A continuación, se mostrará la función que se realizó para este escenario en la simulación y el resultado mediante gráficas de la simulación:

Función del escenario en la simulación

La siguiente función presenta la población acumulada de vehículos que fueron entrando por cada entrada acorde avanza el día.

```
In [27]: def show_gate_entries(kennedyCount, proceresCount):
plt.figure(figsize=(10, 5))
plot(kennedyCount, '-', label='Kennedy')
plot(proceresCount, '-', label='Proceres')

decorate(xlabel='Hora',
ylabel='Cantidad de vehículos',
title='Entrada de vehículos por las diferentes puertas según la hora')
```

Ilustración 22 Función entradas de vehículos al campus por las diferentes puertas

Resultado del escenario en la simulación mediante gráfica

En la simulación de este escenario, teniendo como base la información obtenida durante la recolección de datos con respecto a cada entrada al campus, se tuvo como resultado que aproximadamente 7 de cada 10 vehículos entran por la entrada de la Avenida John F. Kennedy y el restante entra por la Avenida Los Próceres como se puede observar en la siguiente gráfica. Además, se puede ver que el mayor flujo de vehículos que ingresan por las diferentes puertas se encuentra entre las 16:00 PM y 22:00 PM.

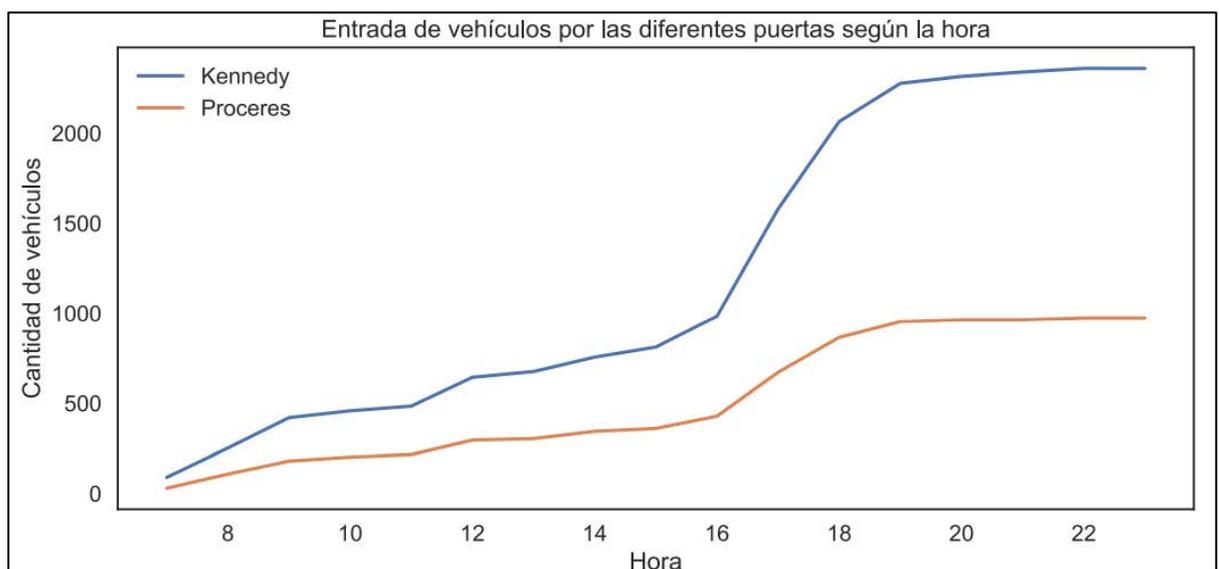


Ilustración 23 Entrada de vehículos según las diferentes dos entradas y horario a la UNPHU

- **Escenario 4: Comparación del tiempo promedio que tarda un vehículo al llegar a un parqueo (Sistema actual vs Sistema propuesto).**

Este escenario de la simulación muestra una de las ventajas al utilizar un sistema de parqueos inteligente, la cual es una reducción considerable en el tiempo de manejo para llegar a un parqueo disponible, partiendo de la siguiente premisa:

- No conocer la ubicación y/o disponibilidad de un destino
- La velocidad con la que nos dirigimos, pero al mismo tiempo se ve reducida debido al factor de incertidumbre y oportunidad.

Por eso uno de los problemas que se enfrentan hoy en día es la disponibilidad y búsqueda de parqueos, donde conlleva pérdida de tiempo, contaminación al medio ambiente, alto consumo de combustible y sobre todo retraso en la hora de llegada a su destino final dentro del campus universitario, además, puede observar en la gráfica lo factible que es este trabajo grado para la UNPHU y poder implementar su parqueo inteligente.

Dicha simulación fue construida y fundamentada en base experimentos dentro del campus mediante entrevistas hechas a usuarios que manejan vehículos, en las cuales nos expresaban su proceder en el modo conducción ante situaciones que involucraban el conocimiento y desconocimiento previo sobre la ubicación, pero también la disponibilidad del destino, enfocándonos mayormente en un contexto de parqueos.

Función del escenario en la simulación

Esta función nos muestra el tiempo promedio que toma un vehículo para llegar al parqueo para cada hora correspondiente una vez ha entrado al campus; nos muestra tanto para el sistema actual como para el sistema propuesto.

```
In [22]: print('El tiempo promedio de manejo para llegar a un parqueo:')
print('Sistema propuesto : ', round(result.avgTimeGlobal, 2), 'minutos')
print('Sistema Actual : ', round(resultCurrent.avgTimeGlobal, 2), 'minutos')

diff = resultCurrent.avgTimeGlobal - result.avgTimeGlobal
relDiff = diff/resultCurrent.avgTimeGlobal
print('Muestra una reducción en el tiempo de un', round(relDiff * 100,0), '%')

El tiempo promedio de manejo para llegar a un parqueo:
Sistema propuesto : 1.72 minutos
Sistema Actual : 2.81 minutos
Muestra una reducción en el tiempo de un 39.0 %
```

Ilustración 24 Función general de la comparación del sistema actual y propuesto

```
In [29]: def show_vs_time_avg(currentAvg, proposedAvg):
plt.figure(figsize=(10, 5))
plot(currentAvg, '-', label='Sistema Actual')
plot(proposedAvg, '-', label='Sistema Propuesto')
# plot(youngBoth, '-bo', Label='both')
# plot(aLLAges, '-', Label='aLLAges')

decorate(xlabel='Hora',
         ylabel='Tiempo promedio (Minutos)',
         title='Comparación del tiempo promedio que tarda un vehículo en llegar a un parqueo disponible')
```

Ilustración 25 Función de la comparación del tiempo promedio para llegar a un parqueo disponible

Resultado del escenario en la simulación mediante gráfica

En la siguiente gráfica se puede mostrar que en el sistema actual como tiempo promedio máximo es de 3.5 minutos para obtener un parqueo, mientras que el sistema propuesto se tarda como tiempo promedio máximo 2.2 minutos. En pocas palabras, se pudo demostrar que tan viable es el presente trabajo de grado debido a que reduce 1.3 minutos en tiempo promedio máximo. En el tiempo promedio mínimo en el actual es de 2.0 minutos y el propuesto es de 1.2 minutos, lo que indica que el método planteado disminuye en un 0.8 minutos el tiempo promedio para conseguir un parqueo.

Por otro lado, el tiempo promedio de manejo para llegar a un parqueo:

- Sistema propuesto: 1.72 minutos.
- Sistema actual: 2.81 minutos.

En conclusión, teniendo como base todo lo mencionado en este punto queda demostrado que con el sistema propuesto hay una reducción en el tiempo de un 39%, es decir, se evidencia que tan factible es para el campus de la UNPHU contar con un sistema ciberfísico para un parqueo inteligente.



Ilustración 26 Comparación del parqueo actual vs el propuesto según el tiempo en obtener un parqueo.

4.8 Evaluación del modelo identificando ventajas, desventajas y oportunidades de mejora funcional.

- **Ventajas:** Este modelo propuesto puede beneficiar a instituciones, empresas y organizaciones donde requieran un parqueo inteligente para poder reducir la congestión vehicular. Además, a los usuarios por el hecho de que podrán saber con anticipación la disponibilidad de parqueos en tiempo real, permitiendo que puedan ahorrar mucho tiempo en busca de un parqueo, y así poder tomar decisiones oportunas.
- **Desventajas:** Este modelo cuenta con una desventaja y es debido a que no se incluye la implementación o desarrollo a pequeña escala del sistema ciberfísico.
- **Oportunidad de mejora funcional:** En este modelo propuesto tiene oportunidad de mejora funcional, como son:

- Asegurar disponibilidad de parqueo para todos los usuarios.
- Controlar el aparcamiento en zonas prohibidas o no destinadas a ese fin.
- Controlar el desbordamiento de la cantidad de vehículo en el campus, ni el flujo vehicular dentro del campus.
- Inclusión de reserva de un parqueo a la App.

4.9 Diseño del sistema ciberfísico para parqueo inteligente en la realidad.

En base a lo visto en el resultado de la simulación, se demostró que tan factible es poder contar con un sistema ciberfísico orientado al Smart Parking. Por consiguiente, se procederá a realizar el diseño del proyecto propuesto para parqueo inteligente utilizando Arduino e Internet de las Cosas (IoT).

El diseño consiste en monitorear e informar en tiempo real la disponibilidad de parqueos, pero así mismo la población vehicular actual a través de una aplicación móvil. Esto será posible gracias a una red de dispositivos ciberfísicos, actuadores y sensores que detectaran la presencia de vehículos en un área de estacionamiento específico, pero también el control de entrada y salida de los vehículos de la UNPHU.

Por otro lado, la información será capturada por una tarjeta Arduino y enviada a un servicio API a través de internet, donde los usuarios podrán ver la información en su dispositivo en tiempo real. Además, esto logrará que los conductores pierdan la menor cantidad de tiempo posible en el transcurso de encontrar un parqueo disponible.

4.9.1 Elementos del diseño del sistema ciberfísico en la realidad.

Los elementos que componen el diseño del sistema ciberfísico son los siguientes:

- **Arduino MKR WIFI 1010:** Es usado para capturar la información de los sensores y actuadores para enviarla a la nube de Arduino mediante WIFI.
- **API:** Utilizado para la comunicación entre la App móvil, aplicación web, servicios externos y la información recolectada por el sistema ciberfísico.
- **App móvil:** Permite visualizar la disponibilidad de parqueos de manera previa en el campus de la UNPHU a través del dispositivo del usuario.

- **Pantalla led Smart Parking:** Usado para mostrar la cantidad de parqueos disponibles según el área donde se encuentre este, es decir, en cada edificio que tenga estacionamiento se colocará esta herramienta.
- **Red de sensores (Ultrasónico):** Ayuda a detectar si el parqueo está disponible o no mediante la distancia desde 2 cm a 400 cm. La distancia sería programada según desea la organización para localizar la presencia de un vehículo a través del sensor ultrasónico.
- **Red de actuadores:** Utilizado para permitir la entrada y salida de los vehículos mediante un lector identificativo en los vehículos.
- **Servicio de almacenamiento en la nube:** Es utilizado para alojar la infraestructura y servicios del sistema en la nube, ya sean de procesamiento o de almacenamiento de data.
- **Servidor de Base de Datos:** Permite almacenar toda la información sobre el estado del proyecto propuesto en cada momento, la cual es colectada por los diferentes sensores y actuadores, provocando cambios en el estado del sistema a través de los eventos que se van dando en el mismo.

4.9.2 Mecanismo del diseño del sistema ciberfísico en la realidad.

Las perspectivas del mecanismo del diseño del sistema ciberfísico para parqueo inteligente utilizando Arduino e Internet de las cosas (IoT) pueden ser de la siguiente manera:

a. Desde la perspectiva del Usuario:

- El usuario al ingresar o salir de campus de la UNPHU ya sea por la Av. Los Próceres o Av. John F. Kennedy, podrá realizarlo de manera automática. Esto es gracias a los sensores de proximidad que detectará el vehículo e inmediatamente el control de acceso vehicular procederá a levantarse para que pueda entrar o retirarse del lugar.
- Una vez el usuario haya ingresado a la UNPHU podrá visualizar de 2 manera la disponibilidad de parqueos según el área, las cuales son:

- 1- **Pantalla Led Smart Parking:** El usuario podrá ver la disponibilidad de parqueo en tiempo real según el edificio, es decir, en cada área de estacionamiento se colocará esta herramienta con el objetivo de que aquellos conductores que tengan desconocimiento de la App puedan de manera previa saber dónde hay aparcamiento desocupado.

- 2- **APP móvil:** El usuario mediante su dispositivo podrá visualizar los parqueos disponibles. Por consiguiente, el conductor ingresará a la APP y podrá visualizar de manera rápida los estacionamientos disponibles con su respectivo edificio.

b. Desde la perspectiva del administrador:

- El administrador de cara a la UNPHU podrá visualizar la disponibilidad de parqueos según el área de la siguiente manera:
 - **Página web:** El administrador mediante su computadora logrará contemplar en tiempo real los datos que son enviados a la nube, la cantidad de vehículos dentro del campus, los aparcamientos disponibles, los estacionamientos ocupados y reportes de este con el objetivo de recolectar información para tomar decisiones futuras o alguna mejora al sistema.

En conclusión, toda la información recolectada será enviada a la nube y presentada en tiempo real. Además, este mecanismo permite que los usuarios puedan ahorrar tiempo, pero a la vez ayuda a la UNPHU a ser catalogada como una universidad que utiliza tecnología de punta para el bienestar de sus usuarios.

4.9.3 Diseño específico y general de integración del sistema ciberfísico e IoT en la realidad.

En las siguientes ilustraciones se puede observar el diseño de integración del sistema ciberfísico e IoT para parqueo inteligente de manera específica por área de parqueo y general por toda la zona del campus, teniendo en cuenta el detalle:

- La información capturada de los parqueos por el Arduino MKR WIFI 1010 mediante por los sensores y actuadores será enviada a través de internet Wifi a la nube de Arduino.
- La APP móvil y la aplicación web podrá mostrar en tiempo real la disponibilidad de parqueos desde la nube mediante el servicio en la nube (API y base de datos) e Internet.
- Los servicios externos (aplicaciones) de la UNPHU pueden comunicarse a través de la API y la nube.
- La base de datos podrá almacenar toda la información relacionada al sistema.

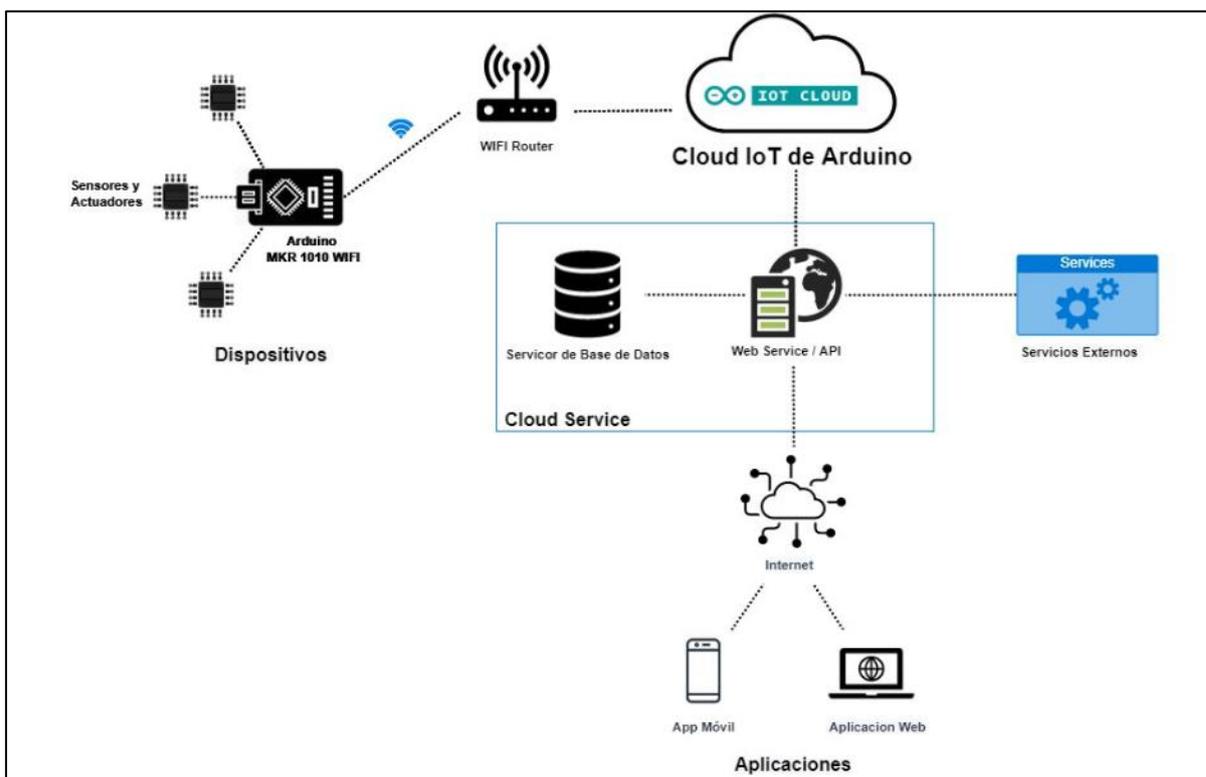


Ilustración 27 Diseño específico de integración del sistema ciberfísico e IoT en la realidad

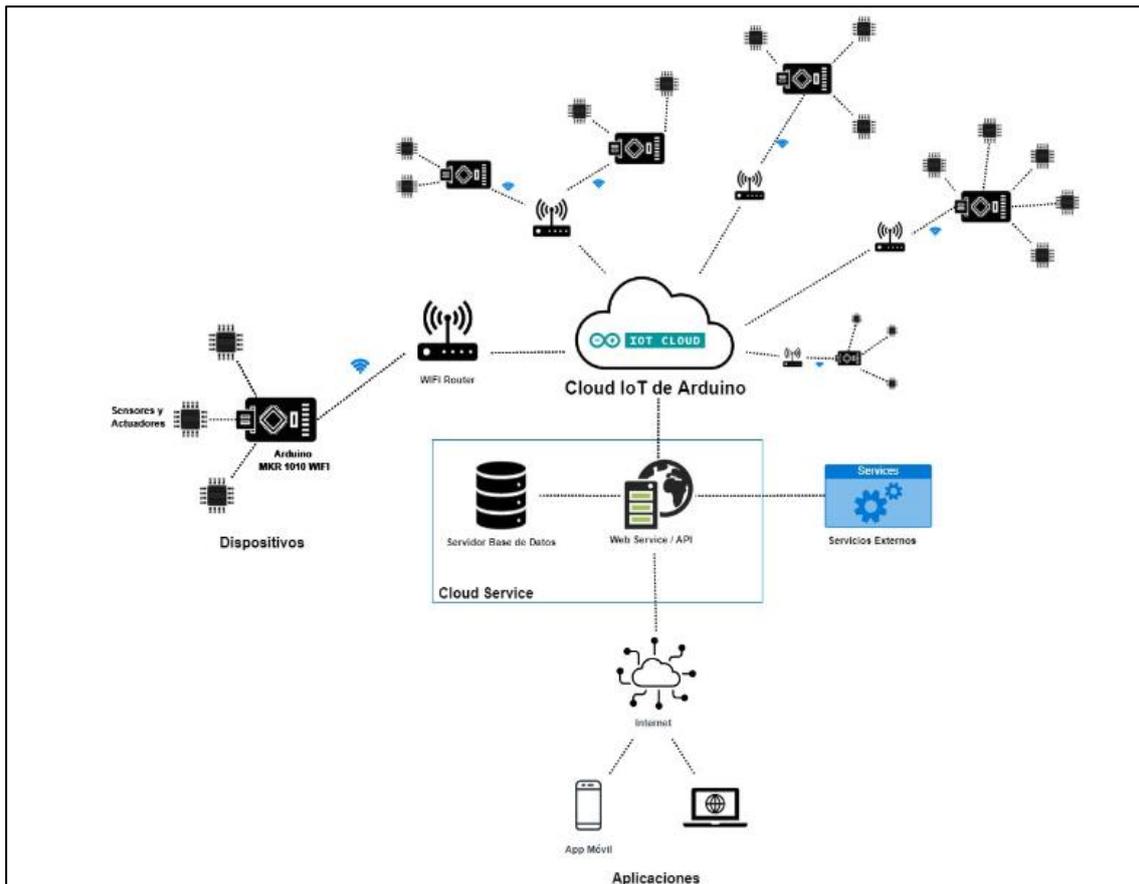


Ilustración 28 Diseño general de integración del sistema ciberfísico e IoT en la realidad

4.9.4 Diseño pantalla led Smart Parking para el sistema ciberfísico en la realidad.

Las pantallas Led Smart Parking mostrará la cantidad de parqueos disponibles según el área donde se encuentre este, es decir, en cada edificio que tenga estacionamiento se colocará esta herramienta. Cabe destacar, que este instrumento se podrá visualizar también los aparcaderos normales y los de usuarios discapacitados.

Estarán fabricados con resistencia al agua y al polvo. Además, tienen amplios ángulos de visión y ofrecen resoluciones de muy alta calidad para poder ser observadas a distancias muy cercanas. Por consiguiente, presentará la información en tiempo real teniendo como base los servicios en la nube para que pueda ser proyectada a través de esta herramienta. Por último, estos pueden ser obtenidos mediante empresas que realicen ventas de artículos orientados al Smart Parking, pero a la vez se ajuste a las necesidades de la UNPHU.

4.9.5 Diseño de la App móvil para el sistema ciberfísico en la realidad.

La App Móvil para el sistema ciberfísico permitirá al usuario mediante su dispositivo visualizar los parqueos disponibles. Por consiguiente, los pasos para utilizarla son:

- Ingresar a la App UNPHU Parqueos.
- Visualizar el edificio que desea aparcar con sus respectivos parqueos disponibles.
- El usuario procederá a dirigirse al edificio del parqueo para estacionarse.

Por otro lado, para poder captar la información que se refleja en la App, pero a la vez utilizarla se debe de tener en cuenta lo siguiente:

- La App UNPHU Parqueos podrá ser usado por cualquier usuario sin tener que iniciar sesión en dicha herramienta.
- Los parqueos disponibles se visualizarán de la siguiente manera según su edificio al que corresponda:
 - **PR:** Parqueo regular, que corresponde a los estacionamientos normales donde los usuarios pueden aparcar.
 - **PE:** Parqueos especiales, que corresponde a los estacionamientos especiales donde las embarazadas y usuarios discapacitados puedan aparcar.

4.9.6 Pantalla del diseño de la App móvil para el sistema ciberfísico en la realidad.

4.9.6.1 Pantalla de inicio



En esta pantalla de inicio de la App móvil, se muestra el logo de la UNPHU y el nombre para dar bienvenida al usuario.

4.9.6.2 Pantalla de parqueos según su edificio



En esta pantalla se muestra los parqueos según al edificio al que corresponda. Además, se puede visualizar los estacionamientos disponibles ya sea normal o especial.

4.9.7 Diseño de la Página web para el sistema ciberfísico en la realidad.

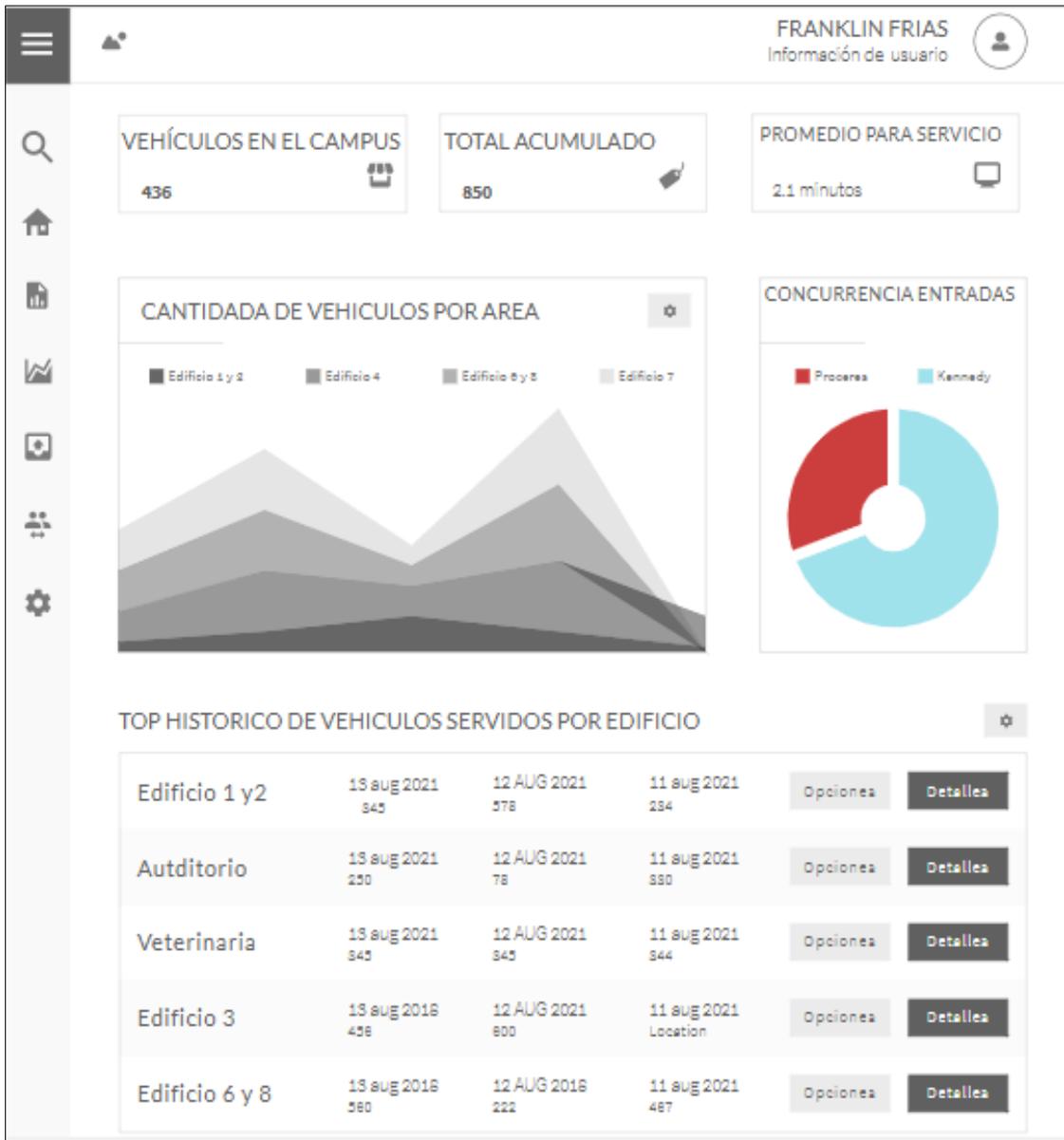
La Página web para el sistema ciberfísico permitirá al administrador mediante su computadora poder contemplar en tiempo real los datos que son enviados a la nube, la cantidad de vehículos dentro del campus, los aparcamientos disponibles, los estacionamientos ocupados y reportes de este con el objetivo de recolectar información para tomar decisiones futuras o alguna mejora al sistema. Por consiguiente, las opciones que tendrá son:

- Inicio y reportes
- Analítica y notificaciones
- Usuarios

4.9.8 Pantalla del diseño de la Página web para el sistema ciberfísico en la realidad.

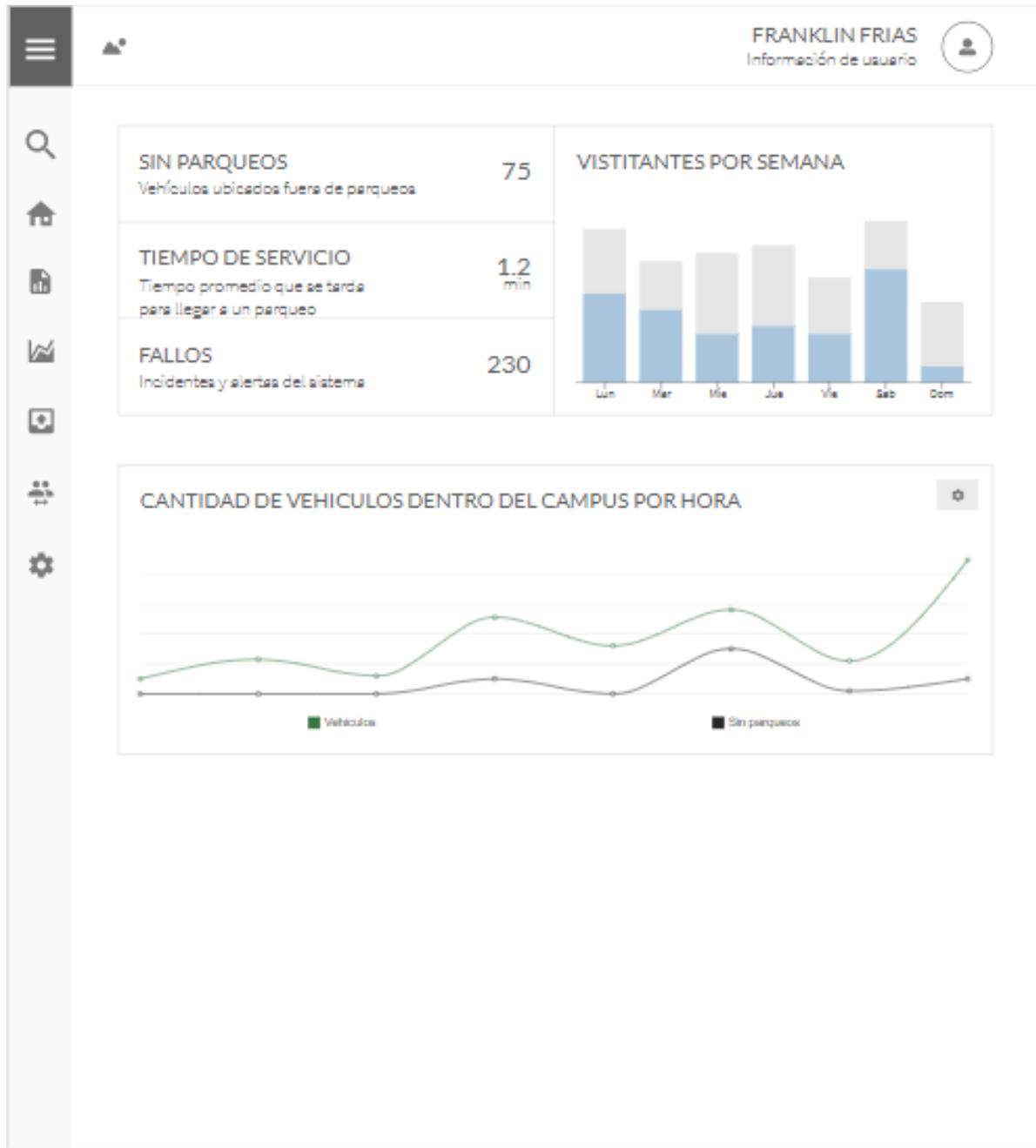
4.9.8.1 Pantalla de inicio y reportes

En esta pantalla se visualiza el histórico de vehículos por edificio, concurrencias de entrada, la cantidad de vehículo por área e información en tiempo real del campus. En pocas palabras, es un resumido de todos los datos recolectados por el sistema ciberfísico. Además, permite generar reporte de los antes mencionados.



4.9.8.2 Pantalla de analítica y notificaciones

En esta pantalla se visualiza las informaciones por semana, los vehículos con y sin parqueos, las fallas, pero a la vez mostrar las notificaciones de incidentes que pueden ocurrir en el sistema ciberfísico.



4.9.8.3 Pantalla de usuarios

En la siguiente pantalla se visualiza las informaciones necesarias para agregar un nuevo usuario para que pueda ingresar a la página web del sistema ciberfísico.

The screenshot shows a web application interface for adding a new user. The top right corner displays the user's name 'FRANKLIN FRIAS' and 'Información de usuario' next to a profile icon. A vertical sidebar on the left contains navigation icons for home, search, and settings. The main content area is titled 'Detalles de Nuevo Usuario' and features several input fields: 'Usuario', 'Correo electrónico', 'Nombre', and 'Apellido'. There is also a dropdown menu for 'Rol' and a checked checkbox for 'Enviar confirmación por correo'. To the right, under 'Foto de perfil', there is a placeholder image and a button labeled 'Seleccione Imagen'. At the bottom of the form are two buttons: 'Agregar Usuario' and 'Cancelar'.

4.10 Estudio de factibilidad.

En base al diseño del proyecto propuesto, se realizó el estudio de factibilidad con el objetivo de obtener información para poder tomar decisiones referentes a la implementación del sistema ciberfísico. A continuación, se muestra la composición de la investigación:

4.10.1 Factibilidad técnica

Esta factibilidad permite saber que el sistema ciberfísico para parqueo inteligente utilizando Arduino e Internet de las cosas (IoT) pueda ser desarrollado con los recursos técnicos (Hardware y software) que tiene en la actualidad la UNPHU.

En el diseño del sistema ciberfísico se cuenta con un API donde permite que cualquier producto o servicios pueda comunicarse con otros, sin tener que realizar una infraestructura de conectividad nueva; por eso se puede adecuar perfectamente a los recursos técnicos de la UNPHU. Además, la interfaz de programación de aplicaciones puede ser realizada en el lenguaje ASP.NET, pero a la vez usando tecnologías de Microsoft donde son de los utilizados actualmente en la UNPHU.

En conclusión, al contar con un API realizado en el lenguaje ya mencionado anteriormente y a la vez conectar con cualquier otra aplicación de la UNPHU queda bajo el mismo esquema de la institución; por lo cual el sistema ciberfísico es factible técnicamente.

4.10.2 Factibilidad económica

Esta factibilidad permite saber si es posible o no invertir en el sistema ciberfísico para parqueo inteligente utilizando Arduino e Internet de las cosas (IoT) en la UNPHU. Cabe destacar, que este es un proyecto innovador que colocaría a la UNPHU en la primera universidad del país que contara con un trabajo como este con tecnología de punta. A continuación, se describe los costos de los recursos materiales esenciales para el desarrollo:

Recursos materiales		
Elementos	Cantidad de elementos	Precios RD\$
Arduino MKR WIFI 1010	10	30,000
Sensores ultrasónicos	864	345,600
Pantalla Led Smart Parking	8	80,000
Actuadores	4	3,000
Total:	886	458,600

Tabla 4 Recursos de materiales esenciales para el sistema ciberfísico

En conclusión, teniendo como base la factibilidad técnica queda demostrado que los recursos necesarios para el desarrollo del sistema ciberfísico para parqueo inteligente se encuentran actualmente en la UNPHU; por lo que el costo de desarrollo es mínimo, pero a la vez permite que la Universidad tenga un gran impacto debido a que hoy en día, la tendencia es la tecnología y lo denominado Smart Parking.

4.10.3 Factibilidad operativa

Esta factibilidad permite determinar si el sistema ciberfísico para parqueo inteligente utilizando Arduino e Internet de las cosas (IoT) cumple con la función necesaria para que sea implementado en la UNPHU. Por otro lado, los usuarios y las personas que se encuentran en la entrada al campus han expuesto lo importante que sería contar con un proyecto innovador como el presente que facilite poder visualizar los parqueos disponibles, además, el acceso más rápido a la organización.

En la siguiente tabla se muestra el costo de los recursos humanos para poder cumplir con la operativa debida referente al proyecto propuesto:

Recursos Humanos				
Cargo	Cantidad	Hora	Costo individual x hora RD\$	Costo Total RD\$
Analista de redes	2	70	1,160	81,200
Analista de sistemas	1	65	1,450	94,250
Analista de software	2	150	1,740	261,000
Administrador de base de datos	1	70	1,740	121,800
Ingeniero de pruebas	2	60	1,450	87,000
Total:	8	415	7,540	645,250

Tabla 5 Recursos humanos para el sistema ciberfísico

Por último, al realizar las entrevistas a los empleados y usuarios se concluye que a nivel operativo el desarrollo del sistema ciberfísico es factible, ya que la UNPHU cuenta con los recursos humanos necesarios para comenzar con el proyecto propuesto.

CAPÍTULO 5 CONCLUSIONES

El objetivo principal de este trabajo de grado fue modelar y simular un sistema ciberfísico de parqueo inteligente para el campus de la UNPHU.

Debido a la innovación de este trabajo de grado, fue necesario utilizar varios tipos de modelos (Modelo estocásticos discreto, modelo cibernético y teoría de cola) para el modelado y simulación donde para monitorear e informar en tiempo real la disponibilidad de parqueos. Además, nos brinda la posibilidad de ofrecer más y mejores servicios a los usuarios, haciendo que a estos les resulte más efectivo el uso del tiempo, así como también la optimización de recursos. Además, nos facilita visualizar en gráfica que óptimo sería implementar este sistema ciberfísico en la UNPHU.

Este trabajo de grado se enfocó en poder satisfacer las siguientes necesidades que hoy en día es un problema, cuales son:

- La entrada/salida del campus que sea de manera automática para poder reducir el tiempo del usuario.
- Los usuarios puedan visualizar los parqueos disponibles o no acorde a cada edificio de manera anticipada al entrar al campus.
- Visualización de la cantidad de vehículos dentro del campus.

Por otro lado, teniendo en cuenta los puntos mencionados anteriormente se tuvo en cuenta algunos casos para poder realizar el modelo y la simulación del sistema ciberfísico de parqueo inteligente, esto fue con objetivo de ver como actualmente funciona el parqueo versus el propuesto y en base a las informaciones concluidas poder darse cuenta de que este trabajo de grado sería un proyecto innovador para una organización. Los casos estudiados fueron:

- Tiempo promedio de entrada/Salida.
- Probabilidad cantidad de vehículos según el horario.

- Tiempo promedio en conseguir parqueo.
- La probabilidad de elegir un destino.
- La probabilidad de cambio de decisiones ante disponibilidad de parqueo.
- Probabilidad de la entrada principal o secundaria, es decir, punto de entrada/salida.
- Impacto del clima al tiene de servicio de entrada y salida del campus como indicadores.
- Comparaciones del parqueo actual vs el propuesto según el tiempo de obtener un parqueo.

El presente trabajo de grado evidencia que al poder contar con un sistema ciberfísico para un parqueo inteligente ayuda en lo siguiente:

- Optimización de estacionamiento.
- Reducción de la contaminación.
- Reducción de tráfico.
- Proporcionar datos en tiempo real.

En este trabajo de grado se demostró teniendo como base las gráficas de la simulación de que tan factible es para la UNPHU contar con un sistema ciberfísico de parqueo inteligente utilizando tecnología de punta debido a que le permite visualizar en tiempo real la disponibilidad de parqueos. Además, de que este trabajo de grado reduce en un 39% el tiempo del usuario para llegar y obtener un parqueo dentro del campus.

Por último, poder realizar el desarrollo del diseño de sistema ciberfísico para parqueo inteligente colocaría a la UNPHU en la primera universidad del país que cuente con un mecanismo tan innovador, pero a la vez tendrá un gran impacto debido a que hoy en día, la tendencia es la tecnología y lo denominado Smart Parking. Además, dicha propuesta permite que lo primero sea el bienestar y comodidad de los usuarios.

CAPÍTULO 6 RECOMENDACIONES

Para este trabajo de grado, este capítulo se dedicará a mostrar las recomendaciones obtenidas a lo largo de este proyecto, teniendo en cuenta lo siguiente:

En primer lugar, como recomendación para la UNPHU u otra organización en conjunto de este presente trabajo de grado en caso de que presente un desbordamiento de vehículos o cogestión vehicular debido a que no haya suficientes parqueos, como sugerencia y pensando en el futuro, se propone colocar en áreas no utilizadas elevadores de vehículos para que todos los usuarios puedan contar con un parqueo, pero a la vez, no recurran a estacionarse en lugares prohibidos.

En segundo lugar, la reservación de parqueos a la App donde los usuarios puedan conservar un parqueadero. Por otro lado, la incorporación de un mecanismo automático para la entrada y salida del campus a través de un identificativo colocado en el vehículo para que le indique al conductor como llegar a su estacionamiento guardado mediante GPS.

Por último, la integración del proyecto propuesto con el sistema UNPHU SIST para que los estudiantes automáticamente realicen su selección de materias e ingresen a la UNPHU, la App le indique el parqueo más cerca según la materia que le corresponde. Además, la implementación real del sistema ciberfísico para un parqueo inteligente.

CAPÍTULO 7 REFERENCIAS

- Aakvaag, J.-E. (2006). Redes de Sensores Inalambricos. *Revista ABB*.
- Abad, C. R. (2002). *Introducción a la simulación y a la Teoría de cola*. España: Netbiblo.
- Arduino. (5 de Febrero de 2018). *Arduino*. Obtenido de Arduino.cc:
<https://www.arduino.cc/en/Guide/Introduction>
- Arroyo, I., Bravo, L., & Muñoz, F. (2014). *Distribuciones Poisson y Gamma: Una Discreta y Continua Relación*. México.
- Azahara. (2017). *Qué es el internet de las cosas*. Sevilla, España: Geographica Carto.
- Becerra, V. (19 de Diciembre de 2016). *Apreniendo Arduino*. Obtenido de Sensores y actuadores:
<https://aprendiendoarduino.wordpress.com/2016/12/18/sensores-y-actuadores/>
- Booch, G., Rumbaugh, J., & Jacobson, I. (1999). *The Unified Modeling Language Use Guide*. Estados Unidos: Pearson Educación.
- Caliper. (Febrero de 2005). *TransModeler Introducción*. Obtenido de
<https://www.caliper.com/pdfs/transmodeler-espanol.pdf>
- Caluva, M. (2005). *Sensores Industriales*. Obtenido de
<https://sites.google.com.site/654sensoresindustriales/home>
- Cardenas, S., & Lemos, H. (2017). *es aquella que utiliza el potencial de la tecnología y la innovación, junto al resto de recursos, para promover de manera más eficiente un desarrollo sostenible y, en definitiva, mejorar la calidad de vida de sus ciudadanos*. México.

- Castro, F. L. (2017). *Diseño de un sistema de parqueo inteligente con aplicación móvil para mostrar espacios disponibles tiempo y valor a cancelar.*
- Code, D. (2018). *Modelos de Markov*. Charleston: Createspace Independent Pub .
- Coss Bu, R. (2003). *Simulación un enfoque práctico*. México: Limusa.
- De Lima Freidel, J. P. (20 de Noviembre de 2017). *Universidad del norte*. Obtenido de Diseño de un sistema de reserva de parqueo en la Universidad del Norte:
<http://manglar.uninorte.edu.co/handle/10584/7731#page=1>
- Downey, A. B. (2017). *Modeling and Simulation in Python*. California: Green Tea Press.
- Escobar, D., Moncada, C., & Urazán, C. (2016). Definición de áreas de estacionamiento en una zona. *Espacios*, 5-10.
- ESPE. (2014). Prototipo de Parqueadero Inteligente Mediante Red., (pág. 181).
- Fernández, M. (2016). *Descifrar las Smart Cities*. España.
- Gandhi, M. (13 de Febrero de 2020). *AutyCom*. Obtenido de Sistemas Ciberfísicos:
<https://www.autycom.com/sistemas-ciberfisicos/#:~:text=En%20la%20actualidad%20los%20sistemas,eficiente%20en%20la%20era%204.0>
- García, E., García, H., & Cárdenas, E. (2006). *Simulación y análisis de sistemas con ProModel*. México: Pearson Educación de México.
- Guasch, A., & Figueras, J. (2005). *Modelado y simulación*. . España.
- Hat, R. (2000). *API*. Obtenido de <https://www.redhat.com/es/topics/api>
- Iglesias Torre, I. (2 de Junio de 2016). *SILLO.TIPS*. Obtenido de <https://silo.tips/download/simpy-para-que-se-utiliza>
- Jordi Salazar, S. S. (2016). *Internet de las cosas*. Chequia, Alemania: TechPedia.
- Jupyter. (2015). *Jupyter Notebook*. Obtenido de <https://jupyter.org/>

- Law, A., & Kelton, D. (1991). *Simulation, Modeling and Analysis*. Singapore: McGraw-Hill Book.
- Lindao, L. N. (2016). *Diseño e implementación de un parqueo inteligente utilizando arduino yun basado en internet de las cosas (IoT)*.
- Maass, A., & Anderson, R. (1985). *Un Modelo de Simulación para Sistemas de Regadío*. Madrid: Consejo Superior de Investigaciones Científicas.
- Marcelo, A., & Osorio, F. (1998). Introducción a los Conceptos Básicos de la. *Cinta moebio*, 46-48.
- Mundo, E. (17 de Junio de 2015). *El Mundo*. Obtenido de <https://www.elmundo.es/economia/2015/06/17/55814682e2704e4e328b4576.html>
- NETEC. (2000). *¿Que es Python?* Obtenido de <https://www.netec.com/que-es-python>
- Noticias, M. (Mayo de 2000). *MSN*. Obtenido de <https://www.msn.com/es-cl/noticias/microsoftstore/%C2%BFqu%C3%A9-es-y-para-qu%C3%A9-sirve-visual-studio-2017/ar-AAAnLZL9>
- Nunes, D. S. (2018). *A Practical Introduction To Human-in-theLoop Cyber-Physical Systems*. New Jersey, United States: Wiley.
- Ontiveros, E., Vizcaíno, D., & Lopez, V. (2017). *Las ciudades del futuro: inteligentes, digitales y sostenibles*. España: Ariel.
- Palano, L., Mainetti, L., Patrono, L., Stefanizzi, M., & Vergallo, R. (2014). Integración de tecnologías RFID y WSN en un sistema de estacionamiento inteligente. *22a Conferencia Internacional de Software, Telecomunicaciones y Redes de Computadoras (SoftCOM)*, (pág. 20). Croacia.
- Pérez, C. (29 de Noviembre de 2019). *ENEB*. Obtenido de La Validación Del Modelo De Simulación: <https://claudioeneb.es/2019/11/29/la-validacion-del-modelo-de-simulacion/>

- Pérez, I. (2016). *Modelo cibernético de la red conversacional: Modelo de sistema viable de conversaciones*. Venezuela: Publicia.
- Ramos Valencia, V., Vera, R., Cabrera, C., & Salazar, J. (2019). sistema de estacionamiento. *Revista Ibérica de Sistemas e Tecnologías de Información*.
- Redrovan Castillo, F., & Pizarro Ochoa, M. (2015). *Desarrollo de una aplicación móvil para la gestión de espacios en un parqueadero de un centro comercial*.
- Rincón, L. (2011). *Procesos Estóxicos*. México.
- Rosales Lindao, L. N. (Diciembre de 2016). *Universidad Politécnica Salesiana*. Obtenido de Diseño e implementación de un parqueo inteligente utilizando arduino yun: <https://dspace.ups.edu.ec/handle/123456789/13461>
- Saafigueroa, L. (2017). *Programación Básica: En lenguaje natural*. Buenos aires: Redusers.
- Santaolalla, J. (2017). *Inteligencia física: Aprende a ver el mundo con la mente de un físico*. Barcelona: Plataforma editorial.
- Sargent, R. G. (2011). *Verification and validation of Simulations Models*. Syracuse.
- School, O. B. (2017). La automatización industrial y los sistemas ciberfísicos. *UIC Barcelona*, 1.
- Shannon, R. E. (1988). *Simulación de sistemas*. México: Trillas.
- Taipe, E. J. (2018). *Diseño de un prototipo de sistema de parqueo inteligente para el edificio de la FIE utilizando tecnologías basado en el Internet de las Cosas*.
- Torrente, O. (2013). *Curso práctico de formación de arduino*. Madrid: RC Libros.
- Valdivieso Armendariz, C. (9 de Febrero de 2017). *Espol*. Obtenido de Diseño del despliegue de redes de sensores para parqueo inteligente: <https://www.dspace.espol.edu.ec/handle/123456789/37275>

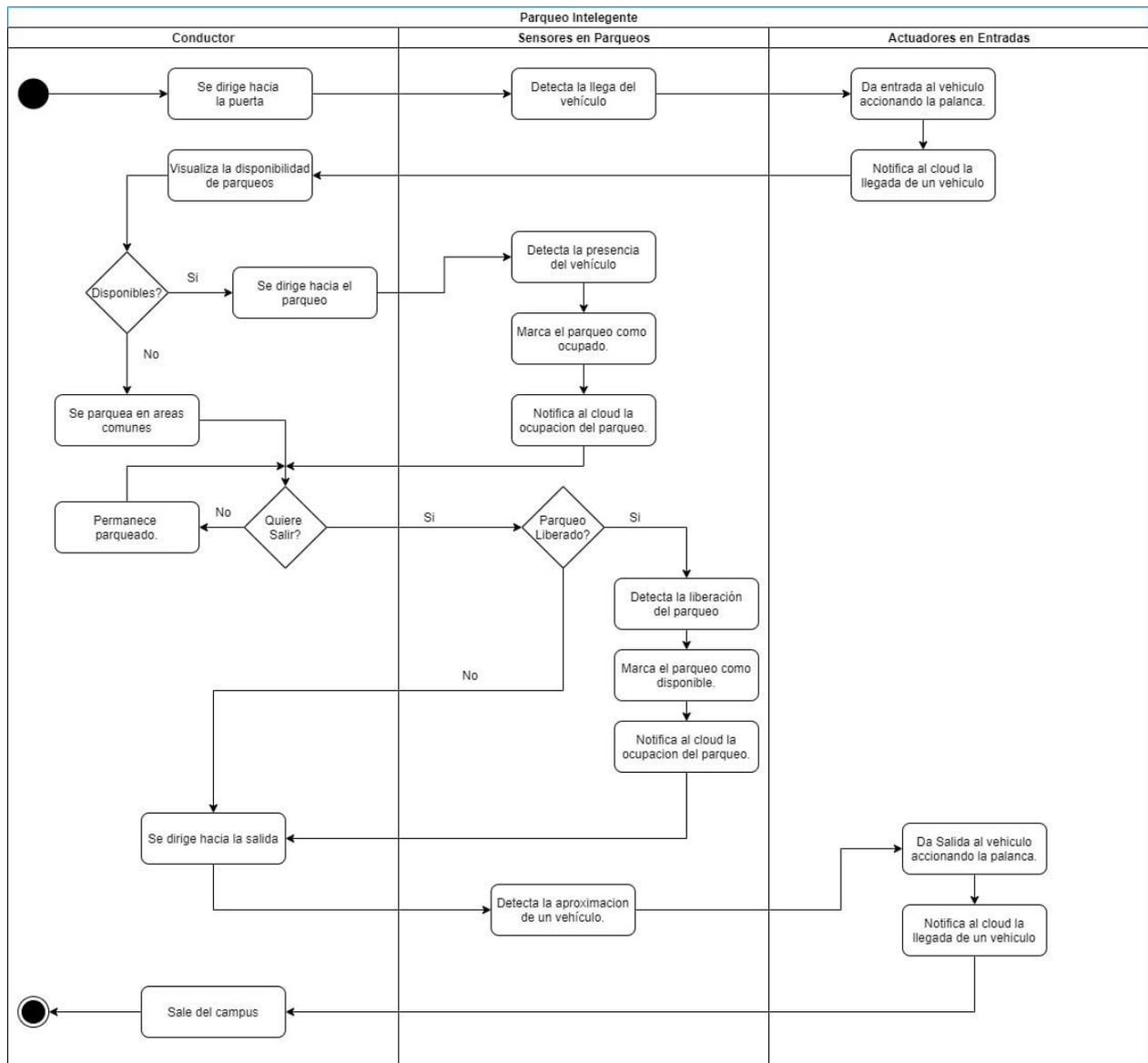
Villalba, C., & Moraleda, A. (2013). *Modelado y Simulación de eventos discretos*.
Madrid.

CAPÍTULO 8 ANEXOS

8.1 Preguntas de las entrevistas

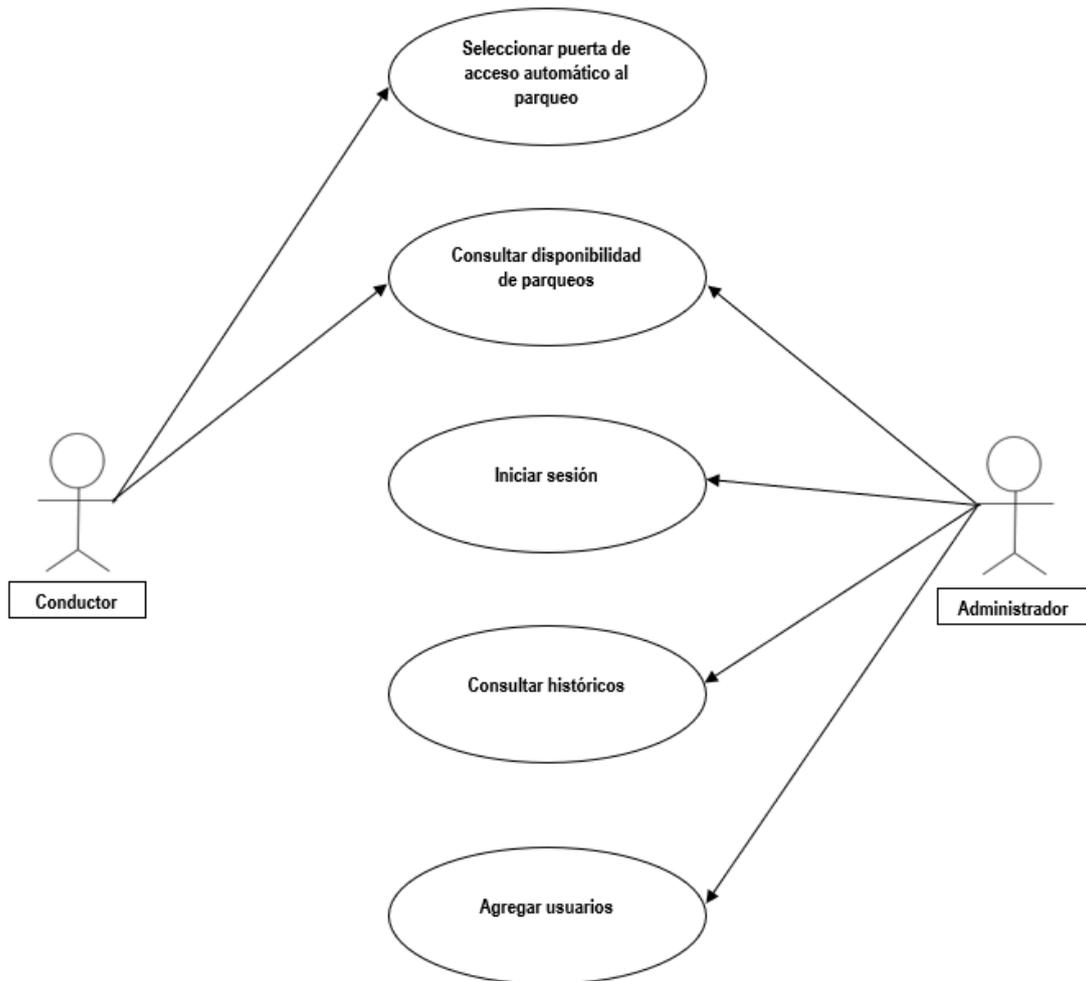
1. ¿Qué es un parqueo inteligente para usted?
2. ¿Qué es un sistema ciberfísico para usted?
3. ¿Qué es una simulación para usted?
4. ¿Qué es un modelado para usted?
5. ¿Cuántos parqueos existe en la UNPHU?
6. ¿Qué entrada es más concurrentes referentes a cada edificio?
7. ¿Cómo es el flujo vehicular en la mañana y en la tarde?
8. ¿Cómo es el comportamiento en cada entrada al campus en horas pico y en clima lluvioso?
9. ¿Cuál es la velocidad máxima para circular en la UNPHU?
10. ¿Qué tiempo tarda un usuario en entrar a la UNPHU?
11. ¿Cree usted que un sistema ciberfísico para un parqueo inteligente pueda ser importante a la hora de reducir el tiempo y la contaminación al medio ambiente?

8.2 Diagrama de actividad del sistema ciberfísico

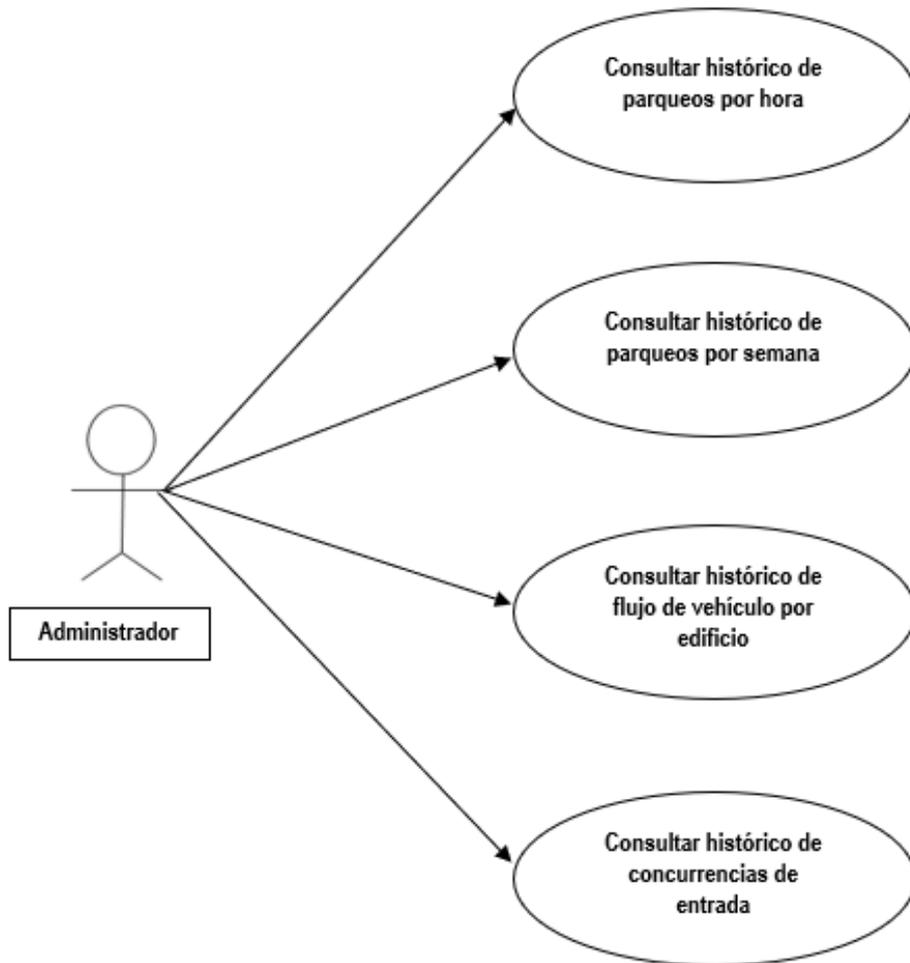


8.3 Casos de uso

8.3.1 Caso de uso general



8.3.2 Caso de uso de consultar histórico - Administrador



8.3.3 Caso de uso de usuarios- Administrador

