

# Plataforma WEB para la Automatización y Control Integral de Parqueaderos Arparking

*WEB Platform For The Automation And Comprehensive Control Of Parking Spaces Arparking*

## Resumen

### Haddy Yised Moreno Agualimpia

Servicio Nacional de Aprendizaje (SENA),  
Grupo de Investigación en Innovación y Agroindustria  
(GIIA), Tecnólogo Análisis y Desarrollo de Software  
(ADSO), hymoreno@sena.edu.co

### Ronald Sneyder Hernández Gómez

Servicio Nacional de Aprendizaje (SENA),  
Semillero Teleinformática (SIT),  
Tecnólogo Análisis y Desarrollo de Software (ADSO),  
ronaldsneyder07@gmail.com

### Como citarse

Moreno-Agualimpia, H.Y. y Hernández-Gómez, R.S. (2024).  
Plataforma WEB Para La Automatización Y Control  
Integral De Parqueaderos Arparking. *Revista Encuentro  
SENNOVA del Oriente Antioqueño*, 10(1), 5-18.

Arparking, es una plataforma que pretende dar solución de automatización y control de parqueaderos de vehículos en el municipio de Guarne - Antioquia. Se desarrolla bajo la metodología Scrum utilizando herramientas de Entorno de Desarrollo Integrado (IDE), como ArgoUML, MySQL WORKBENCH. La codificación de módulos se desarrolla en HTML5, CSS5 y lenguaje de programación PHP, Javascript con la integración de Inteligencia Artificial (IA). Se captura la información de las placas de los vehículos que se encuentren en buen estado, bien sean blancas o amarillas y se asignan celdas. La plataforma permite acceder a cada uno de sus módulos por medio de un usuario y contraseña previamente registrados; adicionalmente, se almacena información de ingreso y salida de vehículos, a su vez, está conformado por los módulos de celda, cliente, empleado, vehículo, pago y tarifa, así como la asignación de políticas de seguridad utilizadas en el desarrollo de la aplicación.



## Abstract

Arparking, is a platform that aims to provide a solution for automation and control of vehicle parking in the municipality of Guarne- Antioquia. It is developed under the Scrum methodology using Integrated Development Environment (IDE), tools such as ArgoUML, MySQL WORKBENCH. The coding of modules is developed in HTML5, CSS5 and PHP programming language, Javascript with the integration of Artificial Intelligence (AI). The information of the license plates of vehicles that are in good condition, whether white or yellow, is captured and cells are assigned. The platform allows access to each of its modules through a previously registered username and password, additionally, information on the entry and exit of vehicles is stored; in turn it is made up of the cell, client, employee, vehicle, payment, rate modules, as well as the assignment of security policies used in the development of the application.

## Palabras clave

Inteligencia Artificial, Lenguaje de programación, Estrategia, Procesamiento de datos, software.

## Keywords

Artificial Intelligence, Programming language, Strategy, Data processing, software.

## Introducción

En la actualidad, el uso de las Tecnologías de la Información y Comunicación (TIC), se han consolidado como una herramienta fundamental para mejorar los procesos administrativos en diferentes áreas, entre ellas la gestión de parqueaderos. Las TIC admiten la automatización de tareas repetitivas, la mejora en la recolección de información y optimización del uso de los recursos disponibles. En este contexto, el municipio de Guarne afronta una problemática significativa en sus parqueaderos, relacionada con la falta de control en el ingreso y salida de vehículos, así como con la emisión oportuna de facturas a los clientes. Esta falta de automatización afecta negativamente la eficiencia operativa y la satisfacción del cliente.

Para abordar esta problemática, se han utilizado estudios y teorías fundamentales sobre la administración de parqueaderos y la integración con la IA. Cabe mencionar, la teoría de la automatización administrativa postula que la integración de sistemas digitales en procesos rutinarios no solo mejora la eficiencia, sino que también reduce los errores humanos y permite una toma de decisiones más informada (Davenport & Short, 1990). Así mismo, la Ley de Moore, predice el crecimiento exponencial en la capacidad de procesamiento de los chips informáticos, subrayando la viabilidad de implementar soluciones de IA cada vez más sofisticadas en la gestión del estacionamiento (Moore, 1965).

El uso de la IA en la gestión de parqueaderos facilita no solo el control de vehículos, sino que también permite la optimización del uso del espacio. La IA puede predecir patrones de ocupación y asignar celdas de manera eficiente, lo que mejora el flujo de tráfico dentro del parqueadero y maximiza la capacidad disponible (Russell & Norvig, 2016). Además, el uso de sistemas de reconocimiento de matrículas y detección automática de vehículos mal estacionados contribuye significativamente a la mejora del servicio y a la reducción de la congestión (Almawgani et al., 2021; Bin Mohd Nazri et al., 2020).



El objetivo principal de esta investigación es la implementación de un software de parqueaderos en el municipio de Guarne – Antioquia, que permita registrar el ingreso y salida de vehículos, asignar celdas de manera eficiente mediante el uso de la IA, y generar facturas automáticamente.

Para sustentar la viabilidad de esta solución, se ha recurrido a estudios que han explorado soluciones innovadoras en la gestión de parqueaderos. Estos estudios incluyen la detección automática de automóviles mal estacionados (Almawgani et al., 2021), aplicaciones de estacionamiento inteligente con reconocimiento de matrículas (Bin Mohd Nazri et al., 2020), y sistemas de estacionamiento basados en las preferencias de los usuarios (Mohandes et al., 2019).

Uno de los hallazgos más relevantes de estos estudios es la eficiencia con la que los sistemas basados en IA pueden mejorar la experiencia del usuario y optimizar la operación del parqueadero. Por ejemplo, la implementación de un sistema de reconocimiento de matrículas no solo agiliza el proceso de entrada y salida de vehículos, reduciendo la posibilidad de fraudes y errores en la facturación (Bin Mohd Nazri et al., 2020). A la vez, la detección automática de vehículos mal estacionados permite una gestión más efectiva del espacio, lo que se traduce en una mayor capacidad de servicio y menor tiempo de espera para los usuarios. (Almawgani et al., 2021).

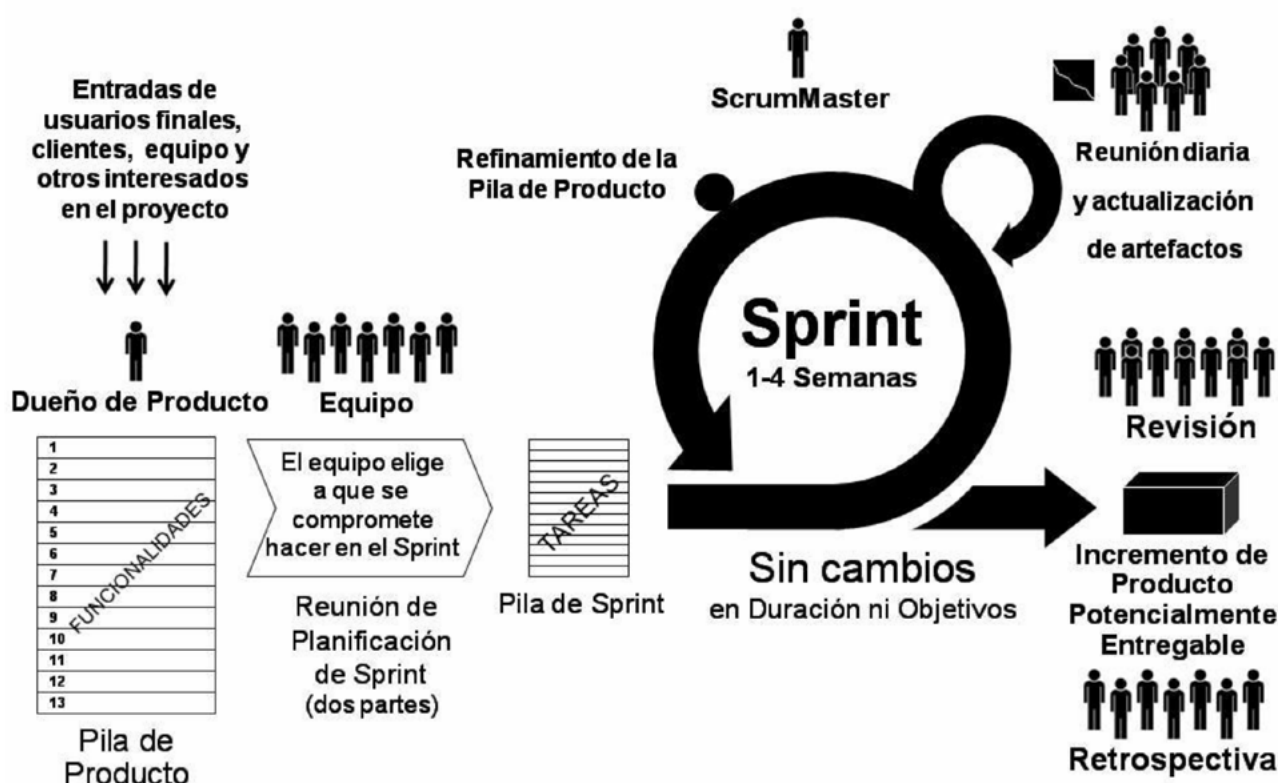
La construcción de esta plataforma en Guarne también busca aprovechar los datos recolectados para mejorar la gestión del espacio físico mediante el uso de modelos predictivos. Estos modelos, basados en técnicas de aprendizaje automático, pueden anticipar la demanda de estacionamiento en función de variables como la hora del día, el clima y los eventos especiales, permitiendo a los administradores del parqueadero tomar decisiones proactivas y optimizar la asignación de celdas (Goodfellow, Bengio & Courville, 2016). La implementación de un software que controle el ingreso y salida de vehículos calcule automáticamente las tarifas y utilice IA para la asignación de celdas en los parqueaderos de Guarne. Lo que representa una solución integral a la problemática actual. Los hallazgos de esta investigación sugieren que la automatización y el uso de tecnologías avanzadas no solo mejorarán la eficiencia

operativa y la satisfacción del cliente, sino que también contribuirán a un uso más sostenible y racional del espacio disponible. Además, el análisis de los datos recolectados permitirá optimizar continuamente la gestión del parqueadero, asegurando que este recurso se utilice de manera efectiva y que las operaciones se adapten a las necesidades cambiantes de la comunidad.

## Metodología

La metodología ágil Scrum (ver **Figura 1**), fue seleccionada para abordar los aspectos socioeconómicos relacionados con el estado actual de los parqueaderos en el municipio de Guarne, destacando su enfoque adaptable y orientado a resultados. Jeff Sutherland, co-creador de Scrum, resalta en “Scrum: The Art of Doing Twice the Work in Half the Time”, que esta metodología permite optimizar el trabajo en equipo y la entrega de valor en proyectos (Sutherland, 2014). Además, Cohn (2004, 2005) y Martin (2002, 2018), han aportado perspectivas prácticas sobre la planificación ágil, el uso de historias de usuario y la arquitectura limpia en el desarrollo de software. Scrum se organiza en roles, eventos y artefactos clave. Los roles incluyen el Product Owner, responsable de maximizar el valor del producto, el Scrum Máster, encargado de guiar al equipo en la metodología, y el Development Team, que construye el producto (Schwaber & Sutherland, 2020).

Los eventos principales comprenden el Sprint, la Sprint Planning, la Daily Scrum, la Sprint Review y la Sprint Retrospective, los cuales facilitan la adaptación continua y la entrega de incrementos funcionales. Los artefactos, como el Product Backlog, Sprint Backlog y el Incremento, estructuran el flujo de trabajo y priorizan tareas. (Schwaber & Sutherland, 2017).

**Figura 1***Metodología Scrum.*

*Nota.* Tomado de THE SCRUM PRIMER. (Deemer et al., 2009).

Para identificar los módulos de la plataforma de gestión de parqueaderos, se empleó una investigación descriptiva con enfoque cualitativo. Según Pardinas (1991), las encuestas permiten recopilar datos clave para analizar problemas específicos. En este caso, se aplicó una encuesta a una muestra de 40 personas, incluyendo comerciantes, funcionarios de la alcaldía del municipio de Guarne, docentes y miembros de la comunidad. Este instrumento permitió identificar retrasos y congestiones en los procesos manuales actuales de registro de vehículos, evidenciando la necesidad de una solución tecnológica.

Dentro de la metodología ágil Scrum aplicada al proyecto de construcción de la plataforma de gestión de parqueaderos en el municipio de Guarne, la implementación de IA ha jugado un papel crucial en mejorar la eficiencia y la experiencia, tanto para los usuarios como para los propietarios. A continuación, se describe el tipo de IA utilizada, las fases de su implementación y los resultados relevantes obtenidos. En el contexto del proyecto, se ha optado por integrar algoritmos de Aprendizaje Automático (Machine Learning), utilizando árboles de decisión y agentes inteligentes.

El propósito de estos modelos es optimizar la toma de decisiones en la gestión de los parqueaderos, tales como la predicción de la disponibilidad de espacios, la asignación de espacios de estacionamiento en tiempo real y la mejora de la seguridad en los procesos de entrada y salida de vehículos. Russell & Norvig (2021), definen:

- Árboles de decisión

Estos algoritmos se utilizan para clasificar patrones de comportamiento de los usuarios, como los hábitos de estacionamiento y las preferencias de los usuarios. Los árboles de decisión permiten que la plataforma realice predicciones y recomendaciones basadas en datos históricos sobre el uso de los parqueaderos.

- Agentes inteligentes

Se utilizan para monitorear las condiciones del parqueadero en tiempo real y gestionar de manera autónoma el flujo de vehículos. Estos agentes son capaces de realizar tareas como el ajuste dinámico de precios en función de la demanda, la asignación de espacios y la comunicación con los usuarios para mejorar la experiencia de usuario.

El modelo de IA utilizado en este proyecto es una versión personalizada, construida sobre un marco base de aprendizaje supervisado para los árboles de decisión y sistemas multiagentes. Este modelo fue entrenado utilizando datos históricos sobre el comportamiento de estacionamiento en los parqueaderos del municipio, los cuales fueron obtenidos a través de encuestas realizadas en la fase de análisis de requisitos:

### Entrenamiento y Base de Datos

La base de datos empleada para el entrenamiento contiene registros históricos de los parqueaderos, incluyendo detalles como horarios de alta demanda, patrones de ingreso y salida de vehículos, y perfiles de los usuarios. La base de datos fue enriquecida mediante el uso de sensores IoT y cámaras de seguridad para obtener datos en tiempo real, los cuales alimentan el sistema de IA. El entrenamiento del modelo se llevó a cabo durante la fase de desarrollo, utilizando herramientas de programación en Python y bibliotecas como Scikit-learn para el entrenamiento de los árboles de decisión y TensorFlow, para el desarrollo de agentes inteligentes. Los modelos fueron validados con datos de prueba obtenidos de una muestra de parqueaderos, lo que permitió ajustar los parámetros y asegurar la precisión del sistema.

### Fase de Análisis de Requisitos (Scrum)

Se identificaron las necesidades relacionadas con el uso de IA en el proyecto, como la predicción de la disponibilidad de espacios y la asignación dinámica de plazas de estacionamiento. Las encuestas y entrevistas con propietarios y usuarios ayudaron a definir los datos relevantes para alimentar los modelos de IA.

### Fase de Diseño del Sistema (Scrum)

En esta etapa, se diseñó la arquitectura del sistema de IA, determinando cómo se integrarían los modelos de aprendizaje automático en la plataforma. Se especificaron las variables a monitorear y se estructuró la base de datos que alimentaría al sistema.

### Fase de Desarrollo del Sistema (Scrum)

Se construyeron los algoritmos de IA y se entrenaron los modelos. Durante esta fase, se implementaron los árboles de decisión para la clasificación de patrones y los agentes inteligentes para gestionar el flujo de vehículos en tiempo real.

### Fase de Prueba de Sistemas (Scrum)

Se realizaron pruebas exhaustivas de los modelos de IA para verificar su desempeño. Esto incluyó pruebas de validación y verificación para asegurar que los algoritmos de IA generaran las predicciones correctas y gestionaran adecuadamente los espacios de estacionamiento.

### Fase de Implementación del Sistema (Scrum)

Finalmente, los modelos de IA se integraron en la plataforma, y el sistema de gestión de parqueaderos comenzó a operar con la capacidad de predecir la disponibilidad de espacios y automatizar la asignación de espacios a los vehículos. Se realizaron ajustes en tiempo real para garantizar que los agentes inteligentes respondieran de manera eficiente a las demandas del sistema.

## Resultados y discusión

Guarne es un municipio ubicado en la subregión Oriente del departamento de Antioquia, Colombia. Limita al norte con Copacabana, Girardota y San Vicente; al este con San Vicente; al sur con Rionegro; y al oeste con Medellín (Alcaldía de Guarne, 2023). Tradicionalmente agrícola, la región ha experimentado transformaciones debido a la industrialización impulsada por el tránsito significativo en la autopista Medellín-Bogotá (Gobernación de Antioquia, 2024). En términos de ocupación laboral, esta dinámica ha tenido un impacto positivo en la economía local. Según la Cámara de Comercio del Oriente Antioqueño (2020), el número de empresas formalizadas en Guarne ha crecido considerablemente, contribuyendo al desarrollo económico y a la generación de empleo en la región. Tras el levantamiento de requerimientos funcionales y su posterior análisis, se definieron los módulos clave para la plataforma. (ver **Tabla 1**).

**Tabla 1**  
*Módulos de la plataforma.*

|                  |                |
|------------------|----------------|
| Inicio de sesión | Usuario        |
| Parqueadero      | Vehículo       |
| Celda            | Factura        |
| Tarifa           | Medios de pago |
| Reserva          |                |

*Nota.* Fuente: Elaboración propia. (2024).

La integración de IA en la plataforma ha generado importantes mejoras en la gestión de los parqueaderos de Guarne. Los resultados más relevantes son:

- Predicción precisa de disponibilidad

Algoritmos de IA predicen la disponibilidad de espacios en parqueaderos con una precisión superior al 85%, basada en datos históricos de demanda.

- Mejora de la experiencia del usuario

Los usuarios reciben recomendaciones en tiempo real sobre espacios disponibles y tarifas ajustadas a la demanda, incrementando la satisfacción general.

- Automatización en la asignación de espacios

Agentes inteligentes asignan automáticamente los espacios de estacionamiento, reduciendo tiempos de espera y mejorando la eficiencia operativa.

Para el desarrollo de la aplicación, se adoptó la metodología Scrum, que permitió priorizar las historias de usuario y concentrar esfuerzos en las funcionalidades críticas. El product backlog sirvió como una lista ordenada y dinámica de necesidades del usuario, que se ajusta continuamente a medida que el proyecto avanza. Esto garantizó: Atención a las funcionalidades más relevantes desde las primeras etapas del desarrollo y mayor satisfacción del cliente y eficiencia en el proceso de desarrollo. La identificación de necesidades del usuario se realizó mediante

encuestas aplicadas en la fase de análisis de requisitos. Este enfoque permitió recopilar información clave directamente de los usuarios, asegurando que las funcionalidades desarrolladas respondieran a sus expectativas. A continuación, se presentan las historias de usuario definidas para la plataforma:

### Historia de Usuario 1 Ingreso y Salida de Vehículos

**Como** usuario del parqueadero, **Quiero** que la aplicación registre automáticamente la hora de entrada y salida de mi vehículo, **Para** que se calcule de manera precisa el tiempo de uso del parqueadero y la tarifa correspondiente.

#### **Criterios de Aceptación**

La aplicación debe registrar la hora exacta de entrada del vehículo en el momento en que pasa por la entrada del parqueadero.

La aplicación debe registrar la hora exacta de salida del vehículo cuando pasa por la salida del parqueadero.

El sistema debe calcular automáticamente el tiempo total de estacionamiento basado en las horas de entrada y salida.

La tarifa por pagar debe calcularse de acuerdo con el tiempo de estacionamiento registrado.

### Historia de Usuario 2 Asignación Inteligente de Celdas

**Como** conductor que utiliza el parqueadero, **Quiero** que la aplicación asigne una celda de estacionamiento automáticamente basada en la disponibilidad y proximidad a la entrada, **Para** que pueda estacionar mi vehículo de manera rápida y conveniente.

#### **Criterios de Aceptación**

La aplicación debe mostrar las celdas disponibles en tiempo real.

Si la celda asignada previamente está ocupada, la aplicación debe asignar la siguiente celda disponible más cercana.

El usuario debe recibir una notificación o indicación clara de la celda asignada al ingresar al parqueadero.

### Historia de Usuario 3 Generación de Facturas

**Como** usuario del parqueadero, **Quiero** que la aplicación me genere una factura digital al momento de salir del parqueadero, **Para** que pueda pagar fácilmente y tener un registro de mis gastos.

#### **Criterios de Aceptación**

La factura debe generarse automáticamente al registrar la salida del vehículo.

La factura debe incluir detalles como la hora de entrada, hora de salida, tiempo total de estacionamiento, y la tarifa aplicada.

La factura debe ser legible y cumplir con los requisitos legales de facturación.

### Historia de Usuario 4 Reserva Anticipada de Estacionamiento

**Como** conductor frecuente del parqueadero, **Quiero** poder reservar una celda de estacionamiento con anticipación a través de la aplicación, **Para** asegurarme de tener un lugar disponible cuando llegue.

#### **Criterios de Aceptación**

El usuario debe poder seleccionar una fecha y hora específica para la reserva.

La aplicación debe mostrar la disponibilidad de celdas para la fecha y hora seleccionadas.

Una vez realizada la reserva, la celda debe ser bloqueada para el uso exclusivo del usuario durante el tiempo reservado.

El usuario debe recibir una confirmación de la reserva con los detalles, incluyendo la celda asignada.

## Historia de Usuario 5

### Notificaciones de Tiempo Restante

**Como** usuario del parqueadero, **Quiero** recibir notificaciones en mi teléfono sobre el tiempo restante de mi estacionamiento, **Para** evitar exceder el tiempo y posibles recargos.

#### **Criterios de Aceptación**

El usuario debe recibir una notificación 15 minutos antes de que el tiempo de estacionamiento expire.

Si el tiempo se extiende, el usuario debe recibir una notificación adicional informando sobre el tiempo extra y la tarifa adicional.

Las notificaciones deben ser claras y aparecer tanto en la aplicación como en el dispositivo móvil del usuario.

El usuario debe tener la opción de extender el tiempo de estacionamiento directamente desde la notificación.

## Historia de Usuario 6

### Histórico de Uso del Parqueadero

**Como** usuario recurrente del parqueadero, **Quiero** poder acceder a un historial de mis usos del parqueadero a través de la aplicación, **Para** revisar mis patrones de uso y gastos.

#### **Criterios de Aceptación**

El historial debe mostrar una lista de todas las entradas y salidas del usuario, incluyendo fechas y horas.

Cada registro en el historial debe incluir la tarifa pagada y la celda utilizada.

El usuario debe poder filtrar el historial por rango de fechas.

El historial debe ser accesible en cualquier momento desde el perfil del usuario en la aplicación.

## Historia de Usuario 7

### Integración con Métodos de Pago Electrónico

**Como** usuario del parqueadero, **Quiero** que la aplicación esté integrada con métodos de pago electrónico como tarjetas de crédito, débito y billeteras digitales, **Para** pagar de manera segura y rápida sin necesidad de efectivo.

#### **Criterios de Aceptación**

La aplicación debe permitir la vinculación de diferentes métodos de pago en el perfil del usuario.

El pago debe ser procesado de manera segura y rápida al momento de generar la factura.

El usuario debe recibir una confirmación de pago inmediatamente después de realizar la transacción.

La aplicación debe manejar pagos fallidos y notificar al usuario si es necesario volver a intentar el pago.

#### **Captura de Placa por medio de IA**

La IA es un campo multidisciplinario que ha sido definido y analizado por diversos autores y expertos, ganado importancia en la historia de la humanidad. Un enfoque pionero en este campo fue propuesto por Alan Turing en 1950, quien introdujo lo que hoy se conoce como el “Test de Turing”. En su artículo “Computing Machinery and Intelligence”, Turing planteó la pregunta, “¿Pueden las máquinas pensar?” y sugirió que, en lugar de responder directamente, se debería considerar si una máquina puede imitar a un humano lo suficientemente bien como para que no se pueda distinguir de una persona en una conversación, (Turing, 1950). De igual manera, Haugeland (1985), expresa que la IA tiene “la interesante tarea de lograr que las computadoras piensen, máquinas con mente, en su amplio sentido literal”, (Merchán, 2015, p.20). En la misma línea, McCarthy (2007), uno de los padres fundadores de la IA, la definió como “la ciencia e ingeniería de hacer máquinas inteligentes, especialmente programas de cómputo inteligentes”(p.8), subrayando la importancia de la capacidad de las máquinas para realizar tareas que requerirían inteligencia si fueran realizadas por humanos. Por otro lado, Russell & Norvig (2010), definen la IA en su libro “Artificial Intelligence: A Modern Approach”, como “el estudio de los agentes inteligentes, siendo esta cualquier cosa que

puede percibir de su entorno y tomar acciones que maximicen sus posibilidades de éxito en algún objetivo o tarea” (p.36). Adicionalmente, la IA ha permitido avances en la industria automovilística, desempeñando un papel importante en la conducción y seguridad vial, lo que contribuye una movilidad segura, confortable y eficiente. Este proceso se logra por medio de la utilización de algoritmos avanzados de IA, los cuales escanean las placas de los vehículos, capturando los datos en tiempo real y asignando automáticamente celdas disponibles en el parqueadero. La **Figura 2**, ilustra el proceso de captura de placas mediante IA.

**Figura 2**

*Captura de placa por medio de Inteligencia Artificial.*



*Nota. Elaboración propia. (2024).*

## Prototipado

El diseño del prototipo del software se realiza teniendo en cuenta que es un modelo de preproducción, en el cual se busca medir el interés del cliente para posteriormente desarrollar la interfaz gráfica con características de claridad, coherencia y sencillez; permitiendo la interacción con el usuario final. Se utilizaron herramientas como Balsamiq Mockup, HTML5, CSS5, como se muestra en la **Figura 3**.

**Figura 3**

*Prototipo de Inicio de sesión Arparking.*



*Nota. Elaboración propia. (2024).*

## Modelado de datos

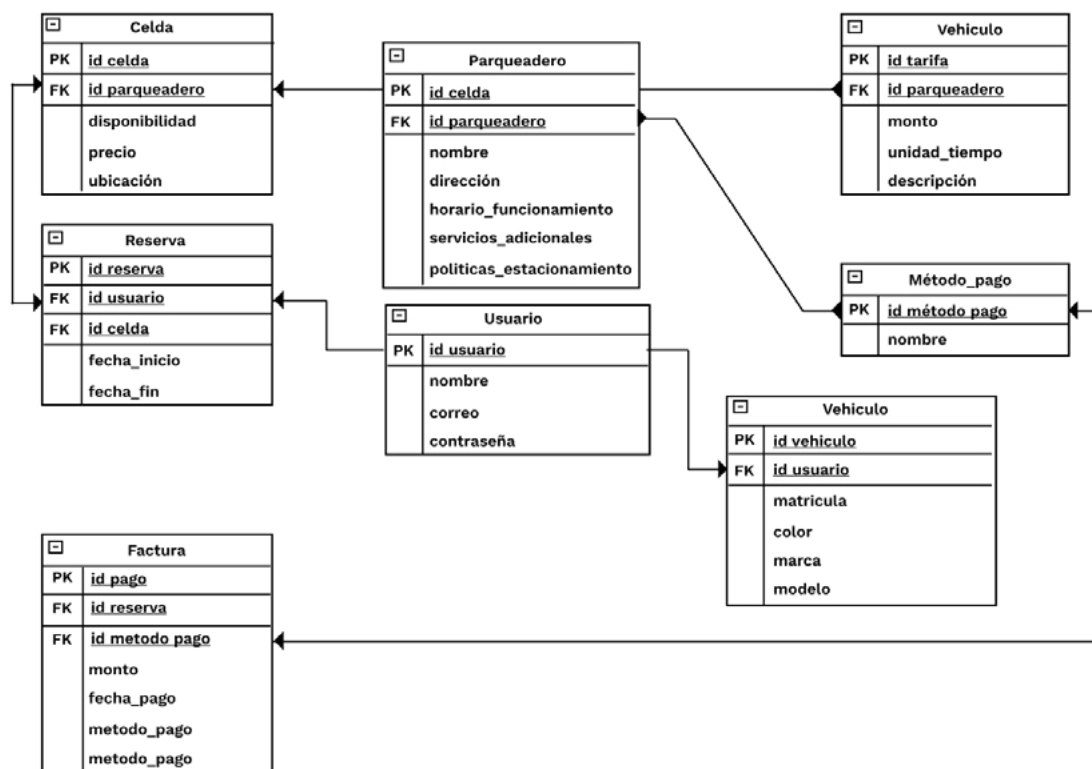
En la moderación de datos se realiza el procesamiento de la información. Para ello se diseña el modelo entidad relación y el diagrama relacional, lo que permite conocer la información que se almacenará en la base de datos. Con base en lo anterior, se procedió a construir la base de datos se utilizan herramientas como WORKBENCH para el modelamiento y MYSQL como gestor de bases de datos, como se evidencia en la **Figura 4**.

## Programación

Para el desarrollo de la aplicación se utilizaron diferentes herramientas, tales como, servidor XAMPP, lenguaje de programación PHP, Python, JavaScript y JQuery; a través de las cuales se desarrollaron la parte funcional de la plataforma. Adicionalmente, se integra con una IA donde se logró la captura de la información de la placa de los vehículos. Posteriormente se asigna una celda disponible y se envía a guardar a la base de datos. Como se muestra en la **Figura 5 y Figura 6**.

**Figura 4**

Modelo Entidad Relación de la Base de Datos.



Nota. Elaboración propia. (2024).

**Figura 5**

Codificación en Python Preprocesamiento de datos históricos.

```

# Importar bibliotecas necesarias
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score
import joblib # Para guardar el modelo entrenado

# 1. Preprocesamiento de datos históricos
def cargar_y_preprocesar_datos(ruta_archivo):
    """
    Carga y preprocesa los datos históricos del parqueadero.
    """
    # Cargar datos desde un archivo CSV
    datos = pd.read_csv(ruta_archivo)

    # Convertir columnas de fecha y hora a formato datetime
    datos['fecha'] = pd.to_datetime(datos['fecha'])
    datos['hora'] = pd.to_datetime(datos['hora'], format='%H:%M:%S').dt.hour
  
```

Nota. Elaboración propia. (2024).

**Figura 6**

Codificación en Python Entrenamiento del Modelo.

```

return X, y

# 2. Entrenamiento del modelo
def entrenar_modelo(X, y):
    """
    Entrena un modelo de clasificación para predecir disponibilidad de espacios.
    """
    # Dividir los datos en entrenamiento y prueba
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=

    # Crear y entrenar un modelo Random Forest
    modelo = RandomForestClassifier(n_estimators=100, random_state=42)
    modelo.fit(X_train, y_train)

    # Evaluar el modelo
    y_pred = modelo.predict(X_test)
    precision = accuracy_score(y_test, y_pred)
    print(f"Precisión del modelo: {precision * 100:.2f}%")

    # Guardar el modelo entrenado
    joblib.dump(modelo, 'modelo_arparking.pkl')
    print("Modelo guardado como 'modelo_arparking.pkl'.")

```

Nota. Elaboración propia. (2024).

Para identificar las necesidades y percepciones de los usuarios sobre la plataforma propuesta, se aplicó una encuesta como parte del levantamiento de requerimientos. Este instrumento fue diseñado siguiendo los lineamientos establecidos por Pardinás (1991), garantizando una recolección de datos sistemática y ordenada. La encuesta se dirigió a una muestra de 40 personas, compuesta por comerciantes, funcionarios de la alcaldía, docentes y miembros de la comunidad general del municipio de Guarne. Este enfoque permitió capturar las necesidades de los usuarios, identificar problemas en los procesos actuales y definir los requisitos para la plataforma de gestión de parqueaderos. Las preguntas de la encuesta se diseñaron para obtener datos específicos sobre las problemáticas del proceso actual, como el registro manual de vehículos, y las expectativas de los usuarios en relación con la aplicación. Los resultados fueron analizados cuantitativa y cualitativamente, priorizando las funcionalidades más relevantes para el desarrollo de la plataforma.

El análisis de los resultados de la encuesta permitió identificar los siguientes puntos críticos que justificaron el desarrollo de la aplicación:

#### Registro Manual de Ingreso y Salida de Vehículos

28 de las 40 personas encuestadas, afirmaron que el proceso manual es lento y propenso a errores ya que incluye actividades como el registro físico de la información del vehículo y del usuario en formatos de papel, la asignación de celdas basada en disponibilidad visual o consulta directa al personal, y el cálculo manual del tiempo de permanencia para la generación de facturas. Estas tareas son realizadas sin el apoyo de herramientas digitales, lo que las hace propensas a errores, como la duplicidad de datos, omisiones o inconsistencias en los registros. Además, la falta de un sistema estructurado genera retrasos en la atención al usuario, especialmente en horas de alta demanda, y dificulta la trazabilidad de los datos, impidiendo un seguimiento eficiente

de las entradas y salidas., generando retrasos en la atención y dificultades en la trazabilidad de los datos. Este resultado, que representa el 70% de los participantes, evidenció la necesidad de automatizar el registro de entradas y salidas mediante una plataforma digital.

#### Falta de Control sobre la Asignación de Celdas

20 de las 40 personas encuestadas, equivalente al 50%, mencionaron la ausencia de un sistema eficiente para asignar celdas de manera equitativa y adaptada a la demanda, lo que genera congestión en ciertas áreas del parqueadero. Se concluyó que la asignación inteligente de celdas, mediante algoritmos, sería una solución eficiente.

#### Generación Manual de Facturas

Se identificó que la facturación manual es propensa a errores y no optimiza el tiempo de servicio. Esto llevó a definir como módulo clave la generación automática de facturas digitales, agilizando el proceso tanto para los usuarios como para los propietarios. Este hallazgo fue señalado por 25 de las 40 personas encuestadas, equivalentes al 62.5%. En base a estos resultados, se definieron los módulos principales de la plataforma, tales como: inicio de sesión, parqueadero, asignación de celdas, tarifas, reservas, usuarios, vehículos, facturas y medios de pago. La priorización de estos módulos se realizó bajo la metodología Scrum, asegurando que las funcionalidades críticas para los usuarios fueran desarrolladas primero.

### Importancia de la Ejecución del Proyecto

La implementación de este proyecto reviste gran relevancia para la comunidad de Guarne y el sector de transporte y logística en la región. La automatización de los procesos de parqueo y facturación permitirá:

Mejorar la Eficiencia Operativa: Optimización de los tiempos de registro y atención en los parqueaderos.

Aumentar la Seguridad y Trazabilidad: registro confiable de vehículos que ingresan y salen, beneficiando tanto a usuarios como a propietarios de parqueaderos.

Reducción de errores humanos en la gestión diaria.

Incrementar la Satisfacción del Usuario: ofrecimiento de una experiencia más ágil y eficiente, con características como la asignación inteligente de celdas y la facturación automatizada.

Fortalecer la Economía Local: Mejora en la administración de los parqueaderos, incrementando la rotación de vehículos y optimizando los ingresos.

## Conclusiones

El desarrollo de la plataforma de automatización y control de vehículos en los parqueaderos del municipio de Guarne involucró la participación de aprendices del Semillero de Investigación en Teleinformática (SIT) y de instructores SENA del CIAA. Este trabajo colaborativo demostró ser una estrategia clave para el éxito del proyecto, permitiendo un análisis detallado de los requerimientos de propietarios y usuarios, así como el diseño y desarrollo de la solución tecnológica. La combinación de teoría y práctica fortaleció las competencias técnicas y profesionales de los participantes, proporcionando un conocimiento integral en áreas clave del saber.

La implementación de la metodología Scrum, fue un factor determinante en la administración de los recursos del proyecto. Esta metodología permitió al equipo descomponer el proyecto en módulos manejables, adaptándose de forma estructurada y ágil a los cambios y desafíos. Como resultado, se logró mejorar tanto la calidad del producto final como la satisfacción de los usuarios, alcanzando de manera eficiente los objetivos propuestos.

La aplicación ArParking representa una solución efectiva para la automatización y control de vehículos en los parqueaderos de Guarne. Al integrar IA para la asignación de espacios de estacionamiento, la plataforma optimiza la gestión operativa, resolviendo problemas como la congestión y la ocupación ineficiente del espacio. Este avance no solo mejora la experiencia de los usuarios, sino que también impulsa el desarrollo local al fortalecer las competencias técnicas de los aprendices involucrados en el proyecto.

Finalmente, la experiencia adquirida a través de este proyecto constituye un valioso precedente para futuros esfuerzos tecnológicos en la región. Este desarrollo destaca el potencial de la innovación como motor de crecimiento sostenible, contribuyendo significativamente a la modernización de servicios esenciales en Guarne y marcando un punto de referencia para iniciativas similares.

## Referencias

- Alcaldía de Guarne. (2023, Julio 6). Identificación del municipio. Información del municipio. <https://www.guarne-antioquia.gov.co/publicaciones/55/informacion-del-municipio/>
- Alcaldía de Guarne. (2023, Julio 6). Identificación del municipio. Información del municipio. <https://www.guarne-antioquia.gov.co/publicaciones/55/informacion-del-municipio/>
- Bin Mohd Nazri, M. I., Alsharif, M. H., Mohd Nasir, N. A., & Hassan, N. (2020). Smart parking system using IoT technology with vehicle recognition and automatic billing. *Journal of Computer Science*, 16(7), 902-913. <https://doi.org/10.3844/jcssp.2020.902.913>
- Cámara de Comercio Oriente Antioqueño. (2020, noviembre). Analítica de datos y aporte al proceso de IVC (inspección, vigilancia y control) para identificar el nivel de informalidad en el municipio de Guarne. *Cámara de Comercio del Oriente Antioqueño*. <https://ccoa.org.co/wp-content/uploads/2021/01/INFORME-Guarne.pdf>
- Cohn, M. (2004). *User stories applied: For agile software development*. Addison-Wesley Professional.
- Cohn, M. (2005). *Agile estimating and planning*. Prentice Hall.
- Davenport, T. H., & Short, J. E. (1990). The new industrial engineering: Information technology and business process redesign. *Sloan Management Review*, 31(4), 11-27. <https://sloanreview.mit.edu/article/the-new-industrial-engineering-information-technology-and-business-process-redesign/>
- Deemer, P., Benefield, G., & Larman, C. (2009). *Scrum primer*. GoodAgile. [https://www.goodagile.com/scrumprimer/scrumprimer\\_es.pdf](https://www.goodagile.com/scrumprimer/scrumprimer_es.pdf)
- Gobernación de Antioquia. (2024, agosto 6). *Guarne Corregimientos de Antioquia*. <https://corregimientos.antioquia.gov.co/guarne/>
- Goodfellow, I., Bengio, Y., & Courville, A. (Ed.). (2016). *Deep learning*. MIT Press.
- Haugeland, J. (Ed.). (1985). *Artificial intelligence: The very idea*. MIT Press.
- Martin, R. C. (Ed.). (2002). *Agile software development: Principles, patterns, and practices*. Prentice Hall.
- Martin, R. C. (Ed.). (2018). *Clean architecture: A craftsman's guide to software structure and design*. Pearson Education.
- McCarthy, J. (2007). What is artificial intelligence?. <http://www-formal.stanford.edu/jmc/whatisai/>
- Merchán, G. (2015). Inteligencia artificial informe. [https://www.academia.edu/7908741/inteligencia\\_artificial\\_informe](https://www.academia.edu/7908741/inteligencia_artificial_informe)
- Mohandes, M., Deriche, M., Abuelma'Atti, M. T., & Tasadduq, N. (2019). Preference-based smart parking system in a university campus. *IET Intelligent Transport Systems*, 13(2), 376-384. <https://doi.org/10.1049/iet-its.2018.5207>
- Mohandes, S. R., Zhang, Y., & Li, C. (2019). User-preference based intelligent parking system for smart cities. *IEEE Transactions on Intelligent Transportation Systems*, 20(11), 4245-4255. <https://doi.org/10.1109/TITS.2019.2900657>
- Moore, G. E. (1965). Cramming more components onto integrated circuits. *Electronics*, 38(8), 114-117. <https://www.cs.utexas.edu/~fussell/courses/cs352h/papers/moore.pdf>

- Pardinas, F. (Ed.). (1991). *Metodología y técnicas de investigación en ciencias sociales*. Siglo veintiuno editores sa. <https://cirma.org.gt/library/images/4/4b/57927.pdf>
- Russell, S., & Norvig, P. (Ed.).(2010). *Artificial intelligence: A modern approach*. Pearson Education.
- Russell, S., & Norvig, P. (Ed.).(2016). *Artificial intelligence: A modern approach*. Pearson Education.
- Russell, S., & Norvig, P. (Ed.).(2021). *Artificial intelligence: A modern approach*. Pearson Education.
- Schwaber, K., & Sutherland, J. (2017). *The Scrum guide: The definitive guide to Scrum: The rules of the game*. <https://scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf>
- Schwaber, K., & Sutherland, J. (2020). *The Scrum guide: The definitive guide to Scrum: The rules of the game*. <https://www.scrumguides.org/scrum-guide.html>
- Sutherland, J. (Ed.).(2014). *Scrum: The art of doing twice the work in half the time*. Crown Business New York. [https://www.agileleanhouse.com/lib/lib/News/More\\_Praise\\_for\\_Scrum\\_The\\_Art\\_of\\_Doing\\_T.pdf](https://www.agileleanhouse.com/lib/lib/News/More_Praise_for_Scrum_The_Art_of_Doing_T.pdf)
- Turing, A. (1950). Computing machinery and intelligence. *Mind, New Series*, 59(236), 433-460. <https://phil415.pbworks.com/f/TuringComputing.pdf>

