Sistema de Parqueadero

Yanderson Jesús Ortiz Cova 192333

Kevin Steiman Sánchez Torres 192313

Tatiana Arengas Martínez 192421

Mariana Barbosa Torrado 192338

Jesús Guerrero

5 de diciembre de 2024.

Universidad Francisco de Paula Santander seccional Ocaña.

Norte de Santander.

Programación 1

**Resumen**

Desarrollo de un sistema automatizado para la gestión eficiente de parqueaderos de vehículos, diseñado para reemplazar los procesos manuales tradicionales y mejorar la operatividad del servicio. Este sistema permite registrar de manera ágil y precisa el ingreso de vehículos, almacenando datos clave como la placa, el modelo, el propietario y la marca. Además, automatiza la generación de recibos detallados al momento de la salida, indicando el monto a cancelar de acuerdo con el tiempo de uso del parqueadero.

Para el desarrollo del proyecto, se utilizó NetBeans como entorno de desarrollo, lo que permitió estructurar la lógica del programa y diseñar una interfaz gráfica amigable y funcional. La base de datos fue implementada con SQL, utilizando XAMPP para gestionar la conexión entre el sistema y la base de datos. Este enfoque combina herramientas tecnológicas robustas para ofrecer una solución moderna, práctica y confiable, orientada a optimizar la administración de parqueaderos y mejorar la experiencia del usuario final.

Este proyecto refleja la integración de conocimientos en programación, diseño de interfaces y gestión de bases de datos para resolver problemas reales.

**Tabla de Contenidos**

[Capítulo 1 Introducción e información general](#_Toc410628920)

Objetivos……………………………………………………………………………2

Alcance………………………………………………………………………………3

Software Development Life Cycle…………………………………………………...4

Codificacion………………………………………………………………………….9

Capítulo 2 Diagramas

SDLC………………………………………………………………………………12

Diagrama de flujo…………………………………………………………………..13

# Capítulo 1 Introducción e información general

El desarrollo de este proyecto fue posible gracias al esfuerzo conjunto de un equipo comprometido, en el que cada integrante desempeñó un rol esencial para alcanzar los objetivos establecidos. Yanderson Jesús Ortiz Cova y Mariana Barbosa Torrado se encargaron de la codificación, desarrollando la lógica del sistema y garantizando su correcto funcionamiento. Tatiana Arengas Martínez y Kevin Steiman Sánchez Torres asumieron la responsabilidad de la documentación, asegurando que cada aspecto técnico y funcional del proyecto estuviera claramente detallado y estructurado.

## Objetivos

Objetivos

1. Diseñar un programa funcional:

Desarrollar un sistema automatizado para la gestión de parqueaderos que sea eficiente, confiable y fácil de usar, abordando las necesidades de registro de vehículos, cálculo de tarifas y generación de recibos.

2. Crear el código:

Implementar la lógica del sistema mediante la codificación adecuada, garantizando que cada funcionalidad cumpla con los requisitos establecidos y esté optimizada para su uso.

3. Realizar los diagramas de flujo y gráficas correspondientes:

Elaborar diagramas de flujo y representaciones gráficas que faciliten la comprensión de la estructura del sistema, su funcionamiento y la interacción entre los diferentes componentes.

4. Implementar interfaces gráficas:

Diseñar y desarrollar interfaces gráficas intuitivas que permitan a los usuarios interactuar de manera sencilla con el sistema, mejorando la experiencia y la eficiencia operativa.

## Alcance

Este proyecto tiene como alcance el desarrollo de un sistema automatizado para la gestión de parqueaderos, el cual reemplazará los procesos manuales de registro y cobro de vehículos. El sistema estará diseñado para gestionar el ingreso y salida de vehículos, registrar información relevante como placa, modelo y propietario, calcular las tarifas de acuerdo con el tiempo de permanencia y generar recibos automáticos.

El sistema estará compuesto por una interfaz gráfica amigable para el usuario, que permitirá interactuar fácilmente con el programa. Además, se implementará una base de datos que almacenará toda la información relacionada con los vehículos y las transacciones. Este proyecto se limitará a la creación de un prototipo funcional para su implementación en pequeños parqueaderos, con la posibilidad de expansión y adaptación a otros entornos.

El alcance no incluye la integración con sistemas externos ni la gestión de parqueaderos a gran escala, aunque el diseño permitirá su escalabilidad en el futuro.

### Software Development Life Cycle

El gráfico que se muestra corresponde al SDLC (Software Development Life Cycle) o Ciclo de Vida del Desarrollo de Software, un marco utilizado para estructurar y organizar las etapas de creación de un sistema de software. En este caso, se detalla el desarrollo de un sistema para la gestión de un parqueadero. A continuación, como se ve en la figura 1 se explica cada etapa:

### Análisis

En esta fase se registran los vehículos con datos como placa, marca, modelo, propietario, entre otros. Además, se muestra la lista de vehículos activos y se registra la salida de un vehículo con la fecha y la hora exacta. Esta etapa se centra en comprender las necesidades del sistema y cómo deben almacenarse y gestionarse los datos de los vehículos.

### Diseño

Aquí se diseña la estructura del sistema. Se especifica la base de datos, que gestionará los datos de los vehículos y sus relaciones, como las tablas de registros. También se define la interfaz gráfica, que en este caso se utiliza el componente Panel Swing, para ofrecer una interacción visual con el usuario. Además, se determina la lógica de negocio, como los métodos para insertar, actualizar, dar de alta y mostrar los datos.

### Implementación

Durante la implementación, se desarrolla el sistema real basado en los diseños previos. El usuario interactúa con el sistema a través de la interfaz, ingresando datos y consultando información. El sistema procesa estas solicitudes, realiza las operaciones correspondientes en la base de datos mediante consultas SQL, y devuelve los resultados al usuario. Los datos ingresados se almacenan en la base de datos para su gestión.

### Prueba

En esta fase se valida el sistema para asegurarse de que funciona correctamente. Se prueba el registro de vehículos, verificando que los datos se almacenan correctamente. También se realizan pruebas de salida de vehículos, comprobando que el sistema gestione adecuadamente las placas de vehículos y valide entradas o salidas de vehículos con placas inexistentes. Además, se verifica que el sistema esté actualizando correctamente la lista de vehículos activos y realizando las operaciones sin errores.

### Mantenimiento

El mantenimiento se centra en el soporte y actualización del sistema una vez que ha sido implementado. Aquí se agregan logs para registrar errores importantes y se realizan pruebas automáticas para garantizar que el sistema siga funcionando correctamente. También se incluyen mejoras en el sistema, como el cálculo de tarifas de estacionamiento según el tiempo. Esta fase asegura que el sistema siga siendo funcional y eficiente a largo plazo.

### Diagrama de flujo

El diagrama muestra el diseño de una base de datos para un sistema de gestión de parqueaderos como se ve en la Figura 2. Aquí está la descripción detallada de cada tabla y las relaciones entre ellas:

**1. tabla: tarifa**

Descripción: Define las tarifas para el uso del parqueadero.

Columnas:

1. id: Identificador único de la tarifa.
2. monto: Valor a cobrar.
3. tiempo\_hora: Tiempo asociado a esa tarifa en horas.

**2. tabla: tipo\_vehiculo**

Descripción: Describe los diferentes tipos de vehículos (por ejemplo, carro, moto, bicicleta).

Columnas:

1. id: Identificador único del tipo de vehículo.
2. nombre: Nombre descriptivo del tipo de vehículo.

**3. tabla: tarifa\_tipo\_vehiculo**

Descripción: Representa la relación entre las tarifas y los tipos de vehículos, ya que un tipo de vehículo puede tener tarifas específicas.

Columnas:

1. id: Identificador único de la relación.
2. tarifa\_id: Referencia a la tabla tarifa.
3. tipo\_vehiculo\_id: Referencia a la tabla tipo\_vehiculo.

**4. tabla: vehiculo**

Descripción: Información de los vehículos que ingresan al parqueadero.

Columnas:

1. id: Identificador único del vehículo.
2. tipo\_vehiculo\_id: Referencia al tipo de vehículo en tipo\_vehiculo.
3. placa: Placa del vehículo.
4. marca: Marca del vehículo.
5. modelo: Modelo del vehículo.
6. nombre\_propietario: Nombre del propietario del vehículo.

**5. tabla: aparcamiento**

Descripción: Registra la entrada y salida de vehículos en el parqueadero.

Columnas:

1. id: Identificador único del registro.
2. vehiculo\_id: Referencia al vehículo en la tabla vehiculo .
3. tiempo\_total: Tiempo total que el vehículo estuvo en el parqueadero (en horas).
4. monto\_total: Monto total cobrado.
5. fecha\_entrada: Fecha y hora de entrada del vehículo.
6. fecha\_salida: Fecha y hora de salida del vehículo.
7. activo: Indica si el registro está activo (1) o si el vehículo ya ha salido (0).

### Relaciones

Relación entre tarifa y tarifa\_tipo\_vehiculo:

La tabla tarifa\_tipo\_vehiculo conecta las tarifas con los tipos de vehículos. Por ejemplo, una tarifa puede ser específica para motocicletas.

Relación entre tipo\_vehiculo y tarifa\_tipo\_vehiculo:

Se establece que un tipo de vehículo puede estar asociado a múltiples tarifas mediante tarifa\_tipo\_vehiculo.

Relación entre tipo\_vehiculo y vehiculo:

Cada vehículo tiene un tipo asignado (por ejemplo, automóvil, motocicleta).

Relación entre vehiculo y aparcamiento:

Cada registro de aparcamiento está vinculado a un vehículo que ingresó al parqueadero.

### Codificación

Para empezar como se muestra en la figura 3, este fragmento de código representa una un modelo que organiza información sobre el uso de un aparcamiento (estacionamiento). En el cual lo usamos para guardar y trabajar con datos como las fechas, tarifas, tiempo de uso, y costos relacionados con un vehículo en un estacionamiento y guarda los siguientes datos:

1. **id**: Es un identificador único del registro del aparcamiento.
2. **tarifaId**: El identificador de la tarifa aplicada al estacionamiento.
3. **vehiculoId**: Identifica al vehículo que usó el estacionamiento.
4. **tiempoTotal**: Mide el tiempo total en horas, que el vehículo estuvo en el aparcamiento.
5. **montoTotal**: El costo total que se debe pagar por ese tiempo.
6. **fechaEntrada**: La fecha y hora en la que el vehículo entró.
7. **fechaSalida**: La fecha y hora en la que el vehículo salió.
8. **activo**: Indica si el registro sigue activo (por ejemplo, si el vehículo todavía está en el aparcamiento).

Ahora el siguiente código visto en la Figura 4, define como sirve para manejar información más específica sobre el aparcamiento, enfocándose en el propietario del vehículo y datos básicos de su ingreso:

1. nombrePropietario: El nombre del dueño del vehículo.
2. placa: número de placa del vehículo.
3. fechaEntrada: El momento exacto (fecha y hora) en que el vehículo ingresó al aparcamiento.
4. activo: Indica si el registro sigue activo (por ejemplo, si el vehículo aún está en el aparcamiento).

Ahora este fragmento de código visto en la figura 5 define los datos que son parte del sistema de facturación:

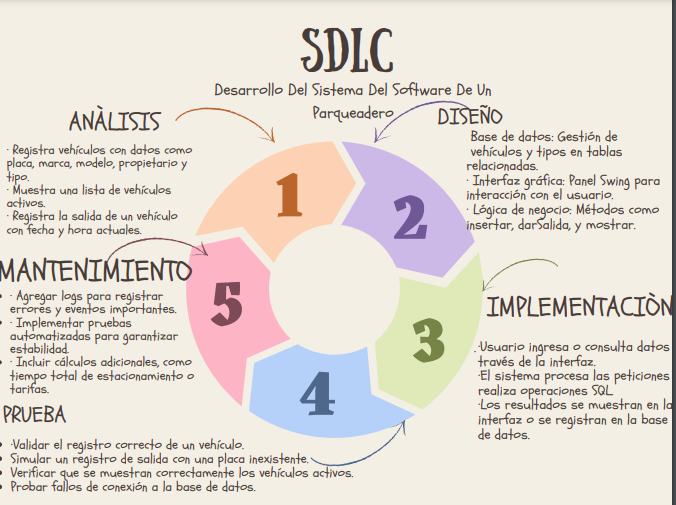
1. aparcamientoDto : Es el objeto donde se guardaran los datos
2. monto: Cantidad a pagar por el usuario
3. tiempoHora: El tiempo que permaneció el vehículo dentro del estacionamiento

Por último observamos en la figura 6 Este código sirve para organizar la información sobre un vehículo. Es útil para registrar y manejar detalles básicos relacionados con automóviles, motocicletas, u otros tipos de transporte que este en el estacionamiento:

1. **id**: Un número único que identifica al vehículo en el sistema.
2. **tipo**: El tipo de vehículo, como "automóvil", "moto", "camioneta", etc.
3. **placa**: El número de matrícula o placa del vehículo, que lo identifica legalmente.
4. **marca**: La marca del vehículo, como "Toyota", "Ford", o "Honda".
5. **modelo**: El modelo del vehículo, como "Corolla", "Mustang", o "Civic".
6. **propietario**: El nombre del dueño del vehículo.

# Capítulo 2 Diagramas.

## SDLC

figura 1. Software Development Life Cycle

## Diagrama de flujo

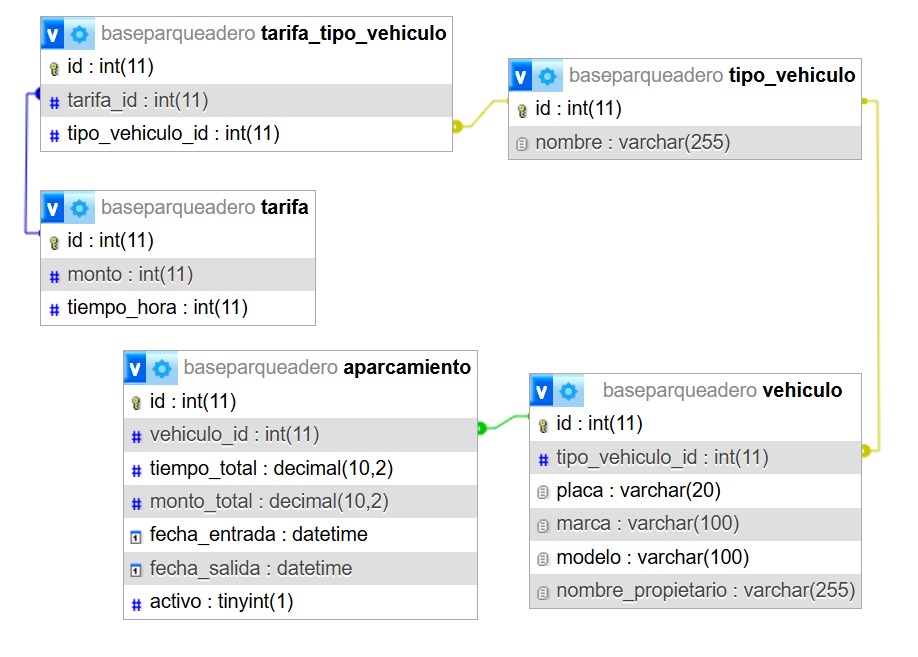


Figura 2. Diagrama de flujo

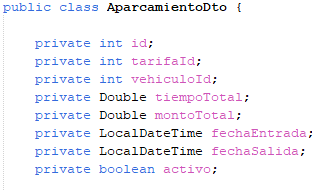


Figura 3. Aparcamiento

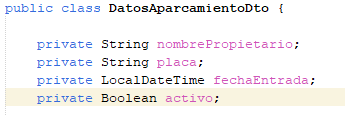


Figura 4. Datos de Aparcamiento

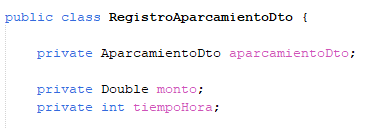


Figura 5. Registro de Aparcamiento



Figura 6.Datos del vehículo

# Capítulo 3 Resultados y discusión.

El desarrollo del sistema automatizado para la gestión de parqueaderos permitió cumplir con los objetivos establecidos, proporcionando un prototipo funcional con características clave que abordan las necesidades de pequeños parqueaderos. A continuación, se detallan los principales resultados obtenidos:

1. Programa Funcional

Logro: Se desarrolló un sistema operativo eficiente y confiable que automatiza el registro de vehículos, calcula tarifas con base en el tiempo de permanencia y genera recibos automáticamente.

Impacto: Se eliminaron los procesos manuales, mejorando la precisión en los cálculos y reduciendo errores humanos.

2. Codificación del Sistema

Logro: Yanderson Jesús Ortiz Cova y Mariana Barbosa Torrado desarrollaron el código del sistema, implementando la lógica para funcionalidades como:

Registro de vehículos con datos como tipo, placa, marca, modelo y propietario.

Cálculo automático de tarifas basado en el tiempo de ingreso y salida.

Generación de recibos con desglose de costos.

El código fue optimizado para garantizar un funcionamiento fluido.

Impacto: El sistema funciona correctamente y cumple con los requisitos planteados, proporcionando un entorno estable y confiable.

3. Diagramas de Flujo y Representaciones Gráficas

Logro:Se elaboraron diagramas de flujo que detallan la estructura y funcionamiento del sistema, como:

Flujo de ingreso y salida de vehículos.

Proceso de cálculo de tarifas.

Interacción entre los módulos de registro, base de datos y generación de recibos.

Gráficos adicionales ilustran la interacción entre la interfaz gráfica y la lógica del sistema.

Impacto: Estos recursos facilitan la comprensión del sistema tanto para desarrolladores como para usuarios interesados en futuras expansiones o adaptaciones.

4. Interfaces Gráficas

Logro: Se implementaron interfaces gráficas intuitivas que permiten al usuario:

Registrar vehículos de forma sencilla.

Consultar detalles de vehículos estacionados.

Generar recibos con un solo clic.

Impacto: La experiencia del usuario fue mejorada significativamente, haciendo que el sistema sea accesible incluso para personas con conocimientos básicos en tecnología.

5. Base de Datos

Logro: Se creó una base de datos que almacena información de manera eficiente, incluyendo:

Registros de vehículos.

Historial de transacciones.

Datos relacionados con las tarifas aplicadas.

Impacto: La base de datos asegura que la información se mantenga organizada y accesible para consultas futuras.

6. Alcance y Limitaciones

Logro:El sistema cumple con las funciones esenciales para la gestión de pequeños parqueaderos, incluyendo:

Registro de ingreso y salida.

Generación de recibos.

Cálculo de tarifas basadas en tiempo.

Se diseñó pensando en la posibilidad de futuras expansiones y escalabilidad.

Limitación: El prototipo no incluye integración con sistemas externos ni está adaptado para operaciones en parqueaderos a gran escala, aunque es posible ampliar su alcance con desarrollos posteriores.