



MARIO ANDRÉ VELASCO GONZÁLEZ  
CARNET: 1546124

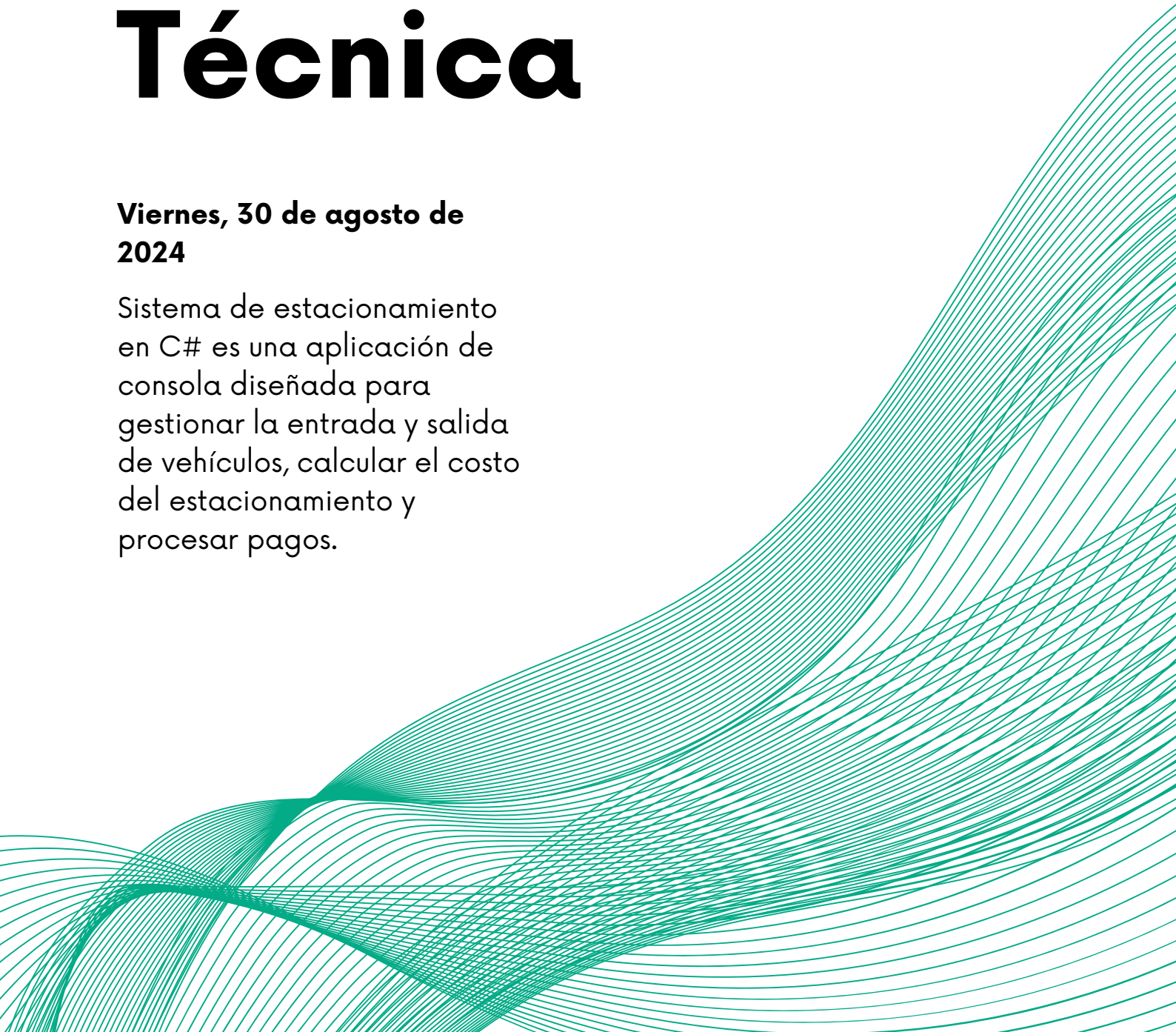
# Informe

---

# Documentación Técnica

**Viernes, 30 de agosto de  
2024**

Sistema de estacionamiento en C# es una aplicación de consola diseñada para gestionar la entrada y salida de vehículos, calcular el costo del estacionamiento y procesar pagos.





## ÍNDICE

---

# 01

Introducción

# 02

Arquitectura del Sistema

# 03

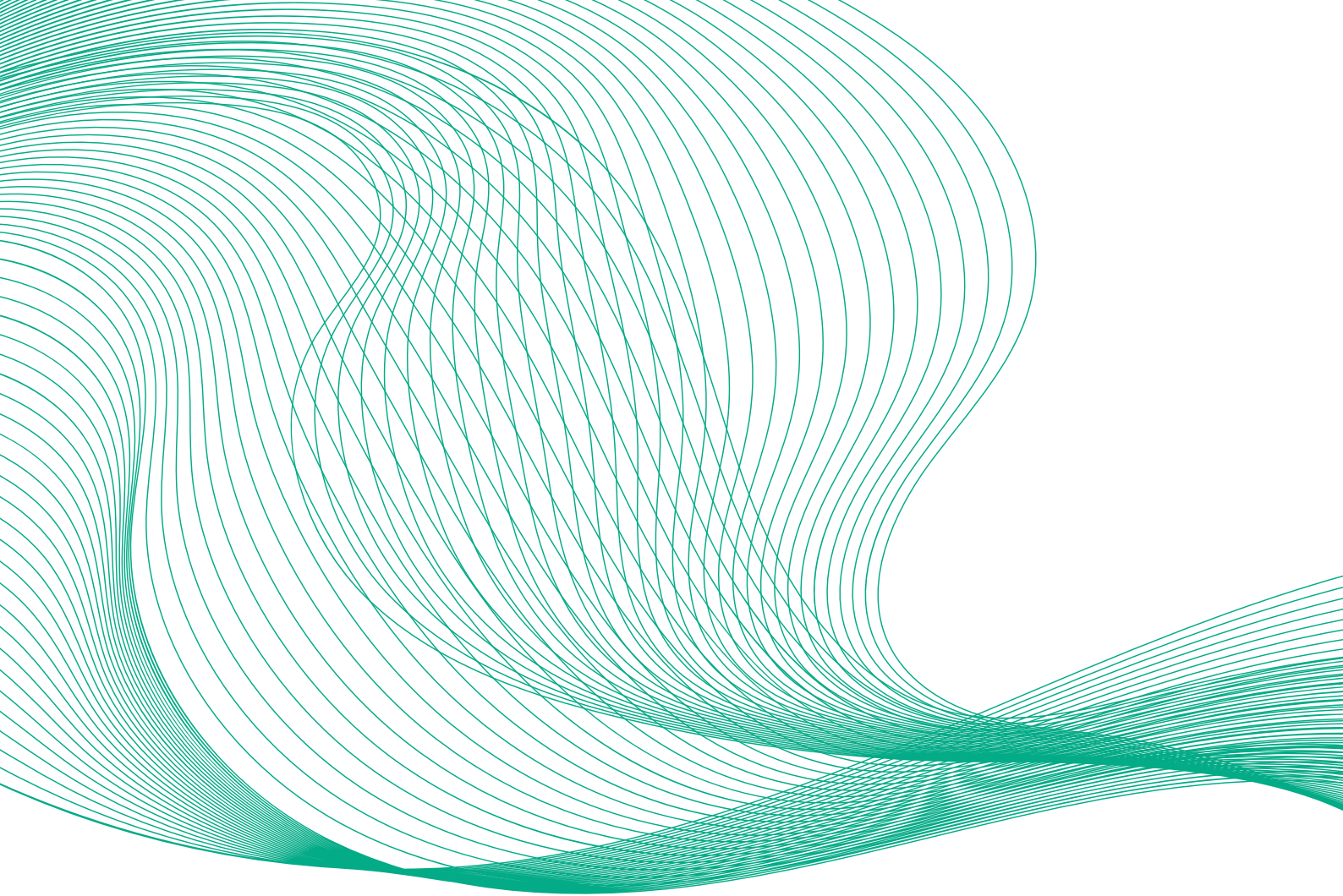
Aplicación de los Principios  
de POO

# 04

Diagrama de Clase

# 05

Descripción de los  
Métodos y Propiedades



## INTRODUCCIÓN

---

- Plante posibles soluciones a los requisitos que se solicitaban al querer registrar los vehículos
- Ingresando los datos necesarios, se pondrá una hora de registro para tener un cambio exacto al estar el vehículo estacionado.
- Al tener un cambio exacto, para asegurar un cobro justo, cualquier fracción de hora se cobra como una hora completa.



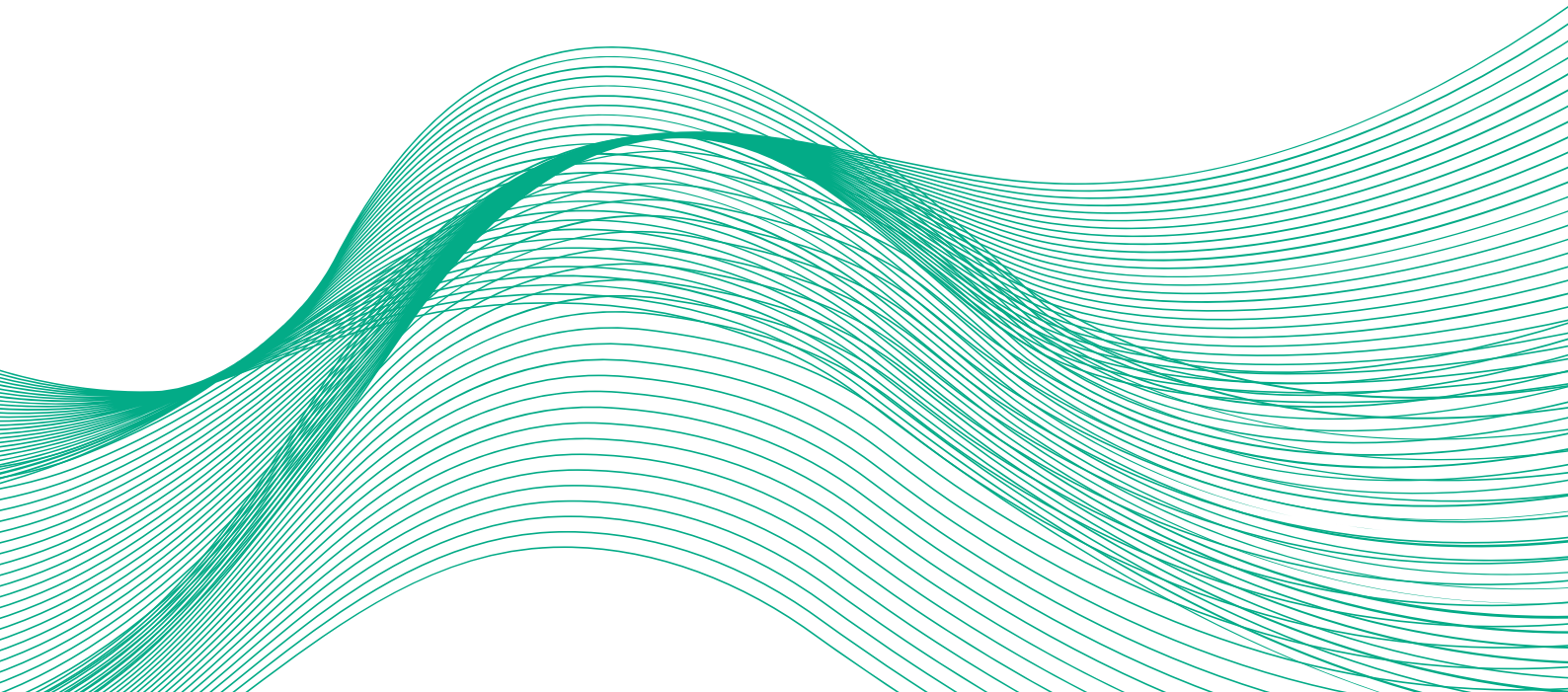
# DESCRIPCIÓN DE LA ARQUITECTURA DEL SISTEMA

---

El sistema de estacionamiento está construido utilizando una arquitectura orientada a objetos. Se organiza en varias clases que representan los componentes fundamentales del sistema:

- **Clase Estacionamiento:** Esta clase administra el estacionamiento en su totalidad. Gestiona la capacidad total, los espacios disponibles y la lista de vehículos estacionados.
- **Clase Vehículo:** Representa los vehículos que se estacionan en el sistema. Almacena detalles como la placa, la marca y la hora de entrada.
- **Clase Program:** La clase principal contiene el punto de entrada principal del sistema y gestiona la interacción con el usuario a través de un menú en consola.

El sistema está diseñado para ser modular y extensible, permitiendo la adición de nuevas funcionalidades o tipos de vehículos sin afectar significativamente el código existente.



# APLICACIÓN DE LOS PRINCIPIOS DE POO

---

## 01 Abstracción

El sistema simplifica la complejidad de las operaciones de estacionamiento al encapsular la lógica en métodos específicos dentro de las clases **Estacionamiento** y **Vehículo**.

## 02 Encapsulamiento

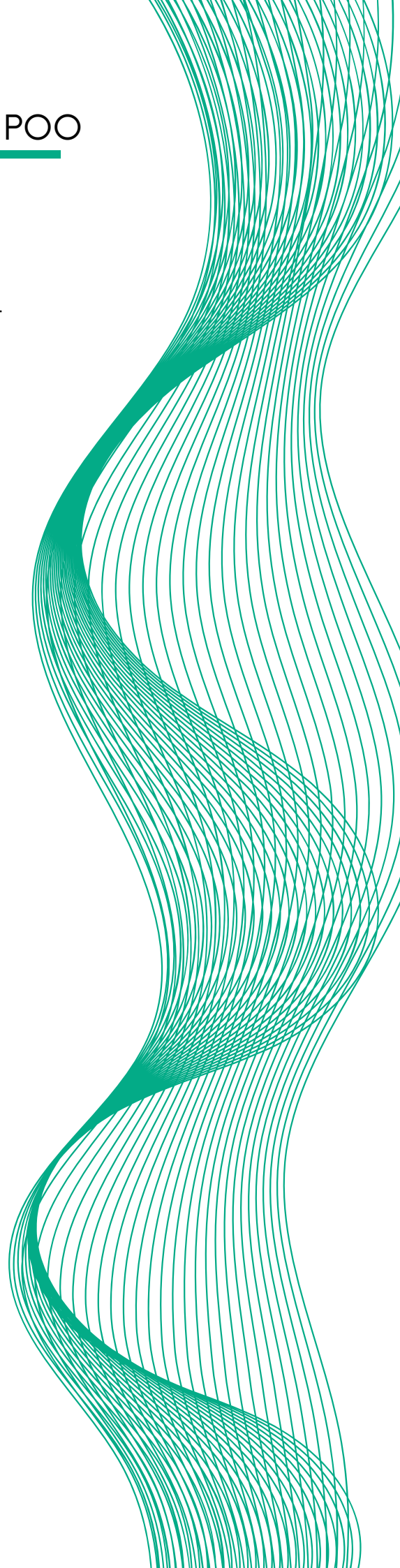
Las propiedades y métodos de las clases están bien definidos y encapsulados, lo que significa que los datos internos de las clases no son accesibles directamente desde fuera de la clase, sino a través de métodos públicos.

## 03 Herencia

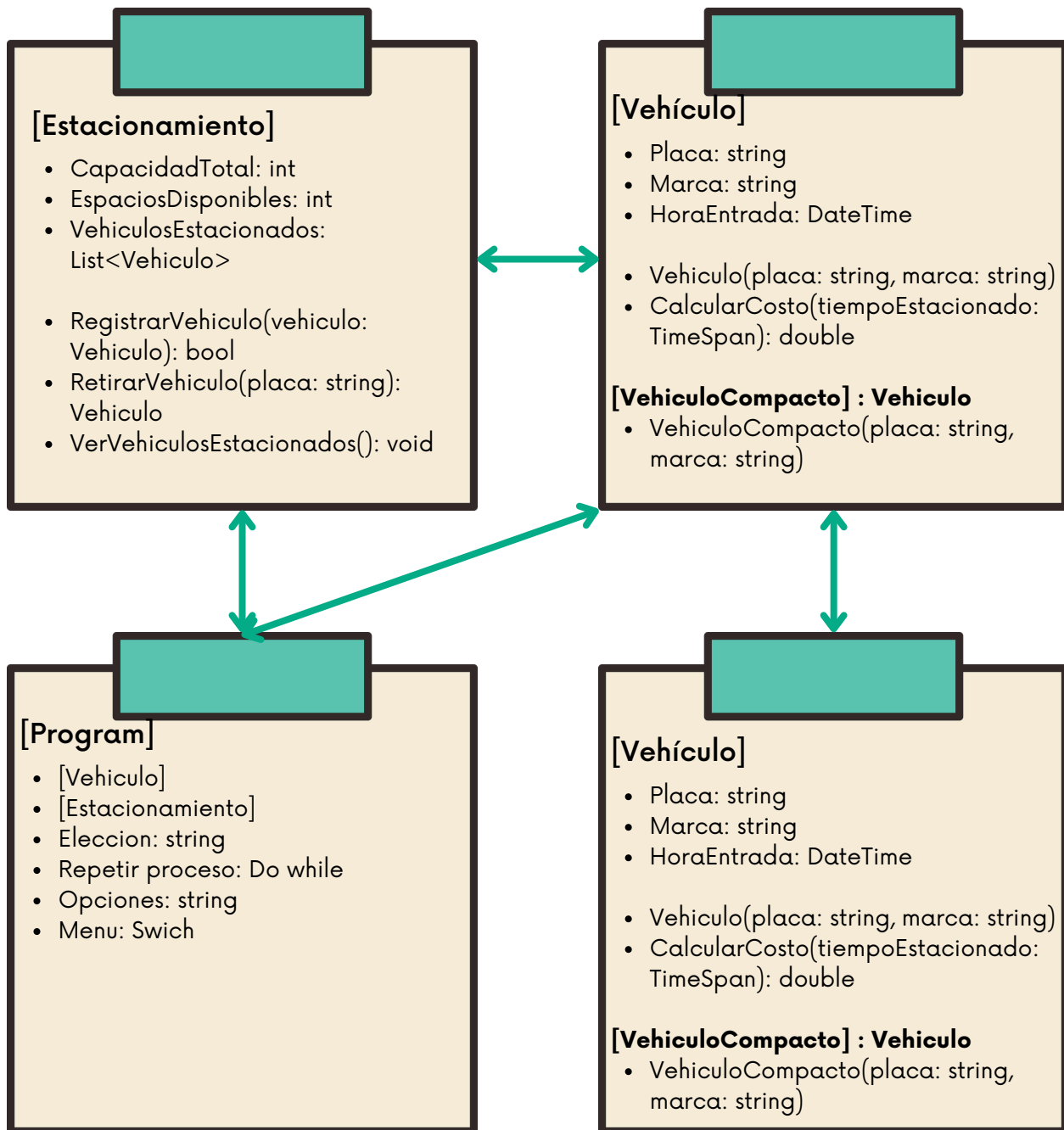
La clase **VehículoCompacto** hereda de **Vehículo**, demostrando cómo una clase base puede ser extendida para crear clases derivadas con características específicas.

## 04 Polimorfismo

El polimorfismo no se explota completamente en esta versión, la estructura de la **clase Vehículo** permite la sobreescritura de métodos como **CalcularCosto** en futuras subclases para ofrecer cálculos de costos personalizados según el tipo de vehículo.



## DIAGRAMA DE CLASES



## DESCRIPCIÓN DE LOS MÉTODOS Y PROPIEDADES

---

- Clase Estacionamiento:
  - CapacidadTotal (Propiedad): Define la capacidad total del estacionamiento, es decir, cuántos vehículos pueden estacionarse simultáneamente.
  - EspaciosDisponibles (Propiedad): Muestra el número de espacios que quedan disponibles en el estacionamiento.
  - VehiculosEstacionados (Atributo privado): Una lista que contiene todos los vehículos actualmente estacionados.
  - RegistrarVehiculo (Método):
    - Descripción: Añade un vehículo a la lista de vehículos estacionados si hay espacio disponible.
    - Retorno: Devuelve true si el vehículo fue registrado exitosamente; de lo contrario, devuelve false.
  - RetirarVehiculo (Método):
    - Descripción: Busca un vehículo por su placa, calcula el costo de estacionamiento basado en el tiempo transcurrido, procesa el pago y, si todo es correcto, retira el vehículo del estacionamiento.
    - Retorno: Devuelve el vehículo retirado si el proceso fue exitoso; de lo contrario, devuelve null.
  - VerVehiculosEstacionados (Método):
    - Descripción: Muestra una lista con los vehículos que están actualmente estacionados.





- Clase Vehiculo:
  - Placa (Propiedad): Almacena la placa del vehículo, que actúa como identificador único.
  - Marca (Propiedad): Almacena la marca del vehículo.
  - HoraEntrada (Propiedad): Almacena la hora exacta en que el vehículo ingresó al estacionamiento.
  - Vehiculo (Constructor):
    - Descripción: Inicializa un objeto Vehiculo con una placa y una marca, estableciendo automáticamente la hora de entrada como la hora actual.
  - CalcularCosto (Método):
    - Descripción: Calcula el costo del estacionamiento en función del tiempo que el vehículo ha permanecido estacionado. La tarifa es de 10 unidades monetarias por hora.
    - Retorno: Devuelve el costo total como un double.
- Clase VehiculoCompacto : Vehiculo
  - VehiculoCompacto (Constructor):
    - Descripción: Hereda todas las propiedades y métodos de Vehiculo, y permite la creación de instancias de vehículos compactos.



