

Requerimiento	Descripción y Explicación	Frontend (Cliente)	Tecnologías para Frontend	Backend (Servidor)	Tecnologías para Backend
1. Horarios y disponibilidad del parqueadero	El sistema debe manejar los horarios de apertura del parqueadero y generar multas si los vehículos se quedan fuera de horario.	El frontend debe mostrar los horarios de apertura y advertir a los usuarios sobre el horario permitido.	HTML para la estructura básica del contenido, Tailwind CSS o Bootstrap para el diseño y estilo, JavaScript para interactividad.	El backend debe verificar las horas de entrada y salida, y generar multas si se excede el horario permitido.	PHP para la lógica de negocio que maneje la validación de horarios, generando multas si es necesario. Base de datos para almacenar y gestionar los horarios y multas.
Descripción del horario: Lunes a viernes 6:00 a.m. a 10:00 p.m., Sábados 9:00 a.m. a 7:00 p.m., Domingos 9:00 a.m. a 12:00 p.m.				Se debe implementar lógica de validación de horarios y penalización automática si el vehículo excede el tiempo permitido.	
2. Capacidad y ubicación	El parqueadero tiene 120 puestos inicialmente, pero debe ser escalable, y cada puesto debe tener una identificación única.	El frontend debe mostrar un mapa o listado con los puestos disponibles, ocupados y su identificación.	HTML y CSS para la visualización de los puestos, JavaScript para crear mapas interactivos (usando bibliotecas como Leaflet o Google Maps API).	El backend debe gestionar la capacidad del parqueadero, asignando los puestos de manera dinámica.	PHP para gestionar la asignación de puestos y la escalabilidad del sistema. Base de datos para almacenar los datos de los puestos.
Escalabilidad y asignación de puestos:					

3. Tarifas diferenciadas según tipo de vehículo	El sistema debe aplicar tarifas diferentes para motocicletas, automóviles y camionetas, y facturar por hora para vehículos sin mensualidad, con un tiempo de gracia de 15 minutos.	El frontend debe permitir seleccionar el tipo de vehículo y calcular la tarifa visualmente para el usuario.	HTML, Tailwind CSS o Bootstrap para la interfaz, JavaScript para interactuar con el backend y mostrar las tarifas calculadas.	El backend debe realizar los cálculos de tarifas según el tipo de vehículo y el tiempo de uso, aplicando el tiempo de gracia.	PHP para implementar la lógica de cálculo de tarifas, validación de tiempo y tipo de vehículo. Base de datos para almacenar tarifas y tiempos.
Tarifas para vehículos no mensualizados: Facturación por hora o fracción, con tiempo de gracia.					
4. Mensualidades	El sistema debe registrar usuarios con mensualidades, validando su horario de entrada y salida. Si no se respeta el horario, se genera una multa.	El frontend debe mostrar y registrar a los usuarios con mensualidades, permitiendo la validación de horarios.	HTML, Tailwind CSS o Bootstrap para el diseño, JavaScript para validar y gestionar la información de usuarios.	El backend debe gestionar las mensualidades, validar las entradas y salidas, y generar multas si no se respeta el horario.	PHP para manejar las mensualidades, validación de tiempos de entrada y salida, y generación de multas. Base de datos para almacenar la información de los usuarios.
5. Validaciones automáticas	El sistema debe calcular automáticamente: horas totales de uso, multas por permanencia fuera del horario y considerar la fecha y moneda local para todas las operaciones.	El frontend debe mostrar las horas de uso, tarifas y posibles multas generadas de manera automática.	HTML para la estructura, CSS para diseño, JavaScript para la validación en tiempo real y cálculo de tarifas/multas.	El backend debe realizar todos los cálculos automáticos de horas, multas y validaciones de fechas.	PHP para el cálculo de horas y multas. Base de datos para registrar la fecha, hora y generar las operaciones necesarias para las validaciones automáticas.

Cálculo de horas y multas:				Se debe implementar un proceso automático que valide el tiempo de permanencia y calcule las multas.	
6. Informes detallados	El sistema debe generar reportes sobre uso de puestos, ingresos, multas generadas y pagos, y un resumen de mensualidades activas y vencidas.	El frontend debe permitir al usuario visualizar y descargar reportes de manera clara y amigable, con gráficos y tablas.	HTML, CSS, y JavaScript (con bibliotecas como Chart.js o D3.js) para representar gráficos interactivos y reportes visuales.	El backend debe generar reportes detallados, calculando los ingresos, las multas generadas y el estado de las mensualidades.	PHP para generar reportes dinámicos, procesar los datos y enviar la información a la interfaz para su visualización. Base de datos para almacenar la información utilizada en los reportes.