**Pasos para Desplegar el Sistema en un Servidor**

**1. Preparación del Servidor**

Primero, necesitas un servidor (generalmente con un sistema operativo como Ubuntu) y conectarte a él (normalmente vía SSH). Una vez dentro, debes instalar el software necesario:

* **Node.js y npm:** Para ejecutar el backend.
* **PostgreSQL:** La base de datos del sistema.
* **Git:** Para descargar el código de tu repositorio.
* **PM2:** Un gestor de procesos que mantendrá tu aplicación funcionando 24/7 y la reiniciará si falla.
* **Nginx (Recomendado):** Un servidor web que actuará como "intermediario" (reverse proxy) para gestionar el tráfico, mejorar la seguridad y servir los archivos del frontend de manera eficiente.

**2. Configuración de la Base de Datos**

1. **Instalar PostgreSQL:** Sigue las guías de instalación para tu sistema operativo.
2. **Crear la Base de Datos y el Usuario:**
   * Conéctate a PostgreSQL y crea un usuario y una base de datos para la aplicación.
3. CREATE DATABASE estacionamiento\_db;
4. CREATE USER estacionamiento\_user WITH ENCRYPTED PASSWORD 'tu\_contraseña\_segura';
5. GRANT ALL PRIVILEGES ON DATABASE estacionamiento\_db TO estacionamiento\_user;
6. **Inicializar las Tablas:**
   * Necesitarás ejecutar el script [init\_db.js](code-assist-path:c:\Users\fagaraye\OneDrive%20-%20Iansagro%20S.A\Escritorio\MIS_%20PROYECTOS\Estacionamiento\backend\init_db.js) para crear las tablas (users, spots, reservations). Para ello, primero tendrás que configurar las variables de entorno del backend (ver paso 3) y luego ejecutar node init\_db.js desde la carpeta backend.

**3. Despliegue del Backend (La API)**

1. **Descargar el Código:** Clona tu proyecto desde el repositorio de Git en una carpeta de tu servidor (por ejemplo, /var/www/estacionamiento).
2. **Instalar Dependencias:** Navega a la carpeta backend y ejecuta:
3. npm install
4. **Crear el Archivo de Entorno:**
   * En la carpeta backend, crea un archivo llamado .env. Este archivo es **crucial** y contendrá las claves y configuraciones secretas que no deben estar en el código.
   * Añade las siguientes variables (reemplazando con tus datos):
5. *# Configuración de la Base de Datos*
6. DB\_USER=estacionamiento\_user
7. DB\_HOST=localhost
8. DB\_DATABASE=estacionamiento\_db
9. DB\_PASSWORD=tu\_contraseña\_segura
10. DB\_PORT=5432
11. *# Clave secreta para la autenticación*
12. JWT\_SECRET=genera\_una\_clave\_muy\_larga\_y\_segura\_aqui
13. *# Configuración del servicio de correo (ej. Gmail)*
14. EMAIL\_HOST=smtp.gmail.com
15. EMAIL\_PORT=587
16. EMAIL\_USER=tu\_correo@gmail.com
17. EMAIL\_PASS=tu\_contraseña\_de\_aplicacion\_de\_gmail
18. *# Correo para notificaciones de fin de semana*
19. COORDINATION\_EMAIL=correo\_coordinador@ejemplo.com

**4. Despliegue del Frontend (La Interfaz de Usuario)**

El frontend son solo archivos estáticos (HTML, CSS, JavaScript). La mejor manera de servirlos es con Nginx.

1. **Configurar Nginx:**
   * Nginx se encargará de mostrar tu [index.html](code-assist-path:c:\Users\fagaraye\OneDrive%20-%20Iansagro%20S.A\Escritorio\MIS_%20PROYECTOS\Estacionamiento\frontend\index.html) cuando alguien visite tu dominio y de redirigir las peticiones a la API (el backend) cuando sea necesario.
   * Crea un archivo de configuración para tu sitio en /etc/nginx/sites-available/estacionamiento.
   * Un ejemplo básico de configuración sería:
2. server {
3. listen 80;
4. server\_name tudominio.com; *# O la IP de tu servidor*
5. *# Ruta a los archivos del frontend*
6. root /var/www/estacionamiento/frontend;
7. index index.html;
8. location / {
9. try\_files $uri $uri/ =404;
10. }
11. *# Redirige las peticiones a /api hacia el backend de Node.js*
12. location /api {
13. proxy\_pass http://localhost:3001;
14. proxy\_http\_version 1.1;
15. proxy\_set\_header Upgrade $http\_upgrade;
16. proxy\_set\_header Connection 'upgrade';
17. proxy\_set\_header Host $host;
18. proxy\_cache\_bypass $http\_upgrade;
19. }
20. }
21. **Activar la Configuración:** Crea un enlace simbólico y reinicia Nginx.
22. sudo ln -s /etc/nginx/sites-available/estacionamiento /etc/nginx/sites-enabled/
23. sudo systemctl restart nginx

**5. Iniciar y Mantener la Aplicación**

Finalmente, para que tu backend se ejecute de forma continua, usa PM2.

1. **Iniciar la Aplicación:** Desde la carpeta backend, ejecuta:
2. pm2 start index.js --name "estacionamiento-api"
3. **Guardar la Configuración:** Para que PM2 reinicie la aplicación automáticamente si el servidor se reinicia:
4. pm2 save
5. pm2 startup

Con estos pasos, tu sistema estará funcionando en un servidor, accesible a través de tu dominio o IP, y gestionado de una manera robusta y segura.