📄 Contrato de Desarrollo de Software

Este contrato establece las herramientas, responsabilidades y tareas que el equipo de desarrollo seguirá para completar el proyecto propuesto. El objetivo es desarrollar una aplicación con autenticación basada en JSON Web Tokens (JWT) utilizando una arquitectura de frontend, backend y base de datos.

# 1. Herramientas a Utilizar

* Backend:
* Lenguaje: Python 3.x
* Framework: FastAPI
* Base de datos: PostgreSQL/MySQL/SQLite
* Autenticación: JWT (PyJWT o equivalente)
* Frontend:
* Lenguaje: JavaScript/TypeScript
* Framework: React o Vue
* Librerías: Axios, React Router o Vue Router
* Almacenamiento de sesión: localStorage
* Otros:
* Control de versiones: Git (GitHub/GitLab/Bitbucket)
* Gestión de tareas: Jira, Trello o equivalente
* Entorno: Docker (opcional)

# 2. Desglose de Tareas por Sprints

## Sprint 1 – Configuración del Proyecto

* Crear carpeta base del proyecto FastAPI.
* Configurar entorno virtual de Python.
* Instalar dependencias iniciales (fastapi, uvicorn, sqlalchemy, etc.).
* Crear archivo main.py con la estructura básica de FastAPI.
* Configurar conexión a la base de datos.
* Definir modelo simple de usuario (username, password\_hash).
* Crear carpeta base del proyecto React o Vue.
* Instalar dependencias iniciales (react-router, axios, etc.).
* Configurar estructura de carpetas (components, pages, services).

## Sprint 2 – Autenticación

* Crear ruta /register para registrar nuevos usuarios.
* Validar datos de registro.
* Hashear contraseñas antes de guardarlas.
* Crear ruta /login para validar credenciales.
* Generar y devolver JWT al iniciar sesión.
* Implementar middleware para validar JWT.
* Crear formulario de login.
* Conectar formulario con API /login.
* Guardar JWT en localStorage.
* Redirigir al usuario después de iniciar sesión.

## Sprint 3 – Rutas Protegidas

* Crear ruta protegida /parking-spots.
* Configurar middleware para rechazar peticiones sin token.
* Probar con tokens válidos e inválidos.
* Agregar token a cada petición API en el frontend.
* Detectar token inválido o expirado.
* Redirigir automáticamente al login en caso de error.

## Sprint 4 – Operaciones CRUD

* Crear modelo de datos para el recurso.
* Implementar endpoint para crear registros.
* Implementar endpoint para leer registros.
* Implementar endpoint para actualizar registros.
* Implementar endpoint para eliminar registros.
* Asegurar que todas las operaciones requieran JWT válido.
* Crear formulario para nuevo registro.
* Crear tabla para listar registros.
* Agregar botones de editar y eliminar.
* Conectar acciones con la API usando el token.

## Sprint 5 – Finalización

* Implementar validaciones en formularios.
* Manejar errores y mostrar mensajes claros.
* Mejorar interfaz de usuario.
* Realizar pruebas básicas.
* Documentar el proyecto.

Nota: Este documento estará sometido a cambios dependiendo a solicitudes realizadas por el cliente, por lo cual, cuando ello ocurra se realizará una actualización a la versión del documento

# Firmas:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Representante del Cliente Representante del equipo