



**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

1

**Reserva tu Parqueadero**  
**Opción de Trabajo de grado Seminario**

Diego Fabian Arias Pérez  
Juan Sebastián Bermúdez Uribe  
Yeison Johanny Escudero Muñoz  
Corporación Universitaria Remington  
Facultad de Ingenierías  
Ingeniería de Sistemas  
Tutor: Luis Camargo Ortega  
2025.



**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

2

**Dedicatoria**

Dedicamos este proyecto a nuestros padres, por su amor incondicional, su ejemplo de lucha y su apoyo constante. A nuestra familia, por creer en nosotros incluso cuando nosotros mismo dudábamos, A nuestros amigos, por acompañarnos en cada paso. Y a nosotros mismos, por no rendirnos, por seguir adelante a pesar de los desafíos y por convertir este sueño en realidad.

**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

3

**Agradecimientos**

Queremos expresar nuestro agradecimiento a todas las personas que hicieron posible el desarrollo de este trabajo. En primer lugar, agradecemos al profesor del seminario por su orientación y disposición para resolver nuestras dudas en cada etapa del proceso. También extendemos nuestro reconocimiento a nuestras familias, quienes nos brindaron apoyo y paciencia durante las semanas de preparación. Finalmente, agradecemos a nuestros compañeros de clase por el ambiente de colaboración que se generó a lo largo del seminario.

**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

4

**Tabla de Contenidos**

**Contenido**

Resumen.....	<b>¡Error! Marcador no definido.</b>
Marco contextual.....	6
Desarrollo e implementación del aprendizaje .....	8
Nodos y su funcionalidad .....	29
Conclusiones.....	45
Referencias.....	47

**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

5

**Resumen**

Este proyecto tiene como objetivo principal el diseño e implementación de un sistema automatizado para la gestión de reservas de espacios en parqueaderos, orientado a mejorar significativamente la experiencia del usuario mediante la reducción de tiempos de espera y la eliminación de filas físicas. La solución propuesta permite al usuario realizar la reserva de un espacio desde su vehículo, evitando desplazamientos innecesarios y agilizando el proceso de ingreso al parqueadero.

La iniciativa busca reemplazar los procedimientos manuales tradicionales, los cuales suelen ser propensos a errores operativos, duplicidad de registros y falta de trazabilidad. Estos inconvenientes no solo afectan la eficiencia del servicio, sino que también generan molestias recurrentes para los usuarios. Mediante la automatización, se pretende establecer un flujo de trabajo más preciso, confiable y transparente, que facilite la administración del parqueadero y optimice la asignación de espacios disponibles.

El sistema está concebido bajo criterios de funcionalidad, eficiencia operativa y bajo costo de implementación, lo que lo convierte en una alternativa viable para entornos urbanos, institucionales o comerciales que requieran soluciones tecnológicas accesibles. Además, se contempla la integración con plataformas digitales y dispositivos

## **TRABAJO DE GRADO**

### **Opción Seminario-Diplomado.**

6

móviles, lo que permite una interacción intuitiva y en tiempo real entre el usuario y el sistema de reservas.

En términos generales, este proyecto demuestra que es posible desarrollar una solución tecnológica robusta y escalable para la gestión inteligente de parqueaderos, contribuyendo a la modernización de los servicios de movilidad urbana y a la mejora continua de la experiencia del usuario.

#### **Marco contextual**

##### **¿Por qué automatizar con bots y n8n?**

Hoy en día, muchos procesos que todavía se hacen “a mano” pueden automatizarse con muy poco código. Un ejemplo perfecto es la reserva de espacios en parqueaderos:

Normalmente alguien lleva una agenda en papel, un Excel, o recibe mensajes por WhatsApp para reservar cupos.

Eso genera errores, doble reserva, pérdida de información y mucho trabajo manual.

Además, no hay control claro sobre pagos, horarios, cancelaciones, etc.

En este proyecto se construyó un Bot de Telegram conectado con n8n y Google Sheets que permite:

Registrar a los usuarios de forma automática.

Mostrar solo los espacios realmente disponibles.

Permitir que el usuario elija horario y duración.

## TRABAJO DE GRADO Opción Seminario-Diplomado.

7

Crear una reserva con ID único.

Simular el pago por medio de un enlace.

Confirmar el pago y actualizar el estado en la base de datos.

Consultar la última reserva.

Cancelar la reserva bajo ciertas condiciones.

Todo esto ocurre **sin intervención humana**, después de que el flujo está configurado. Es decir:

una vez desplegado, el sistema funciona “solo”, respondiendo a cada usuario de forma automática y consistente.

BOT\_PARQUEADERO

### 2. ¿Qué es n8n y por qué lo usamos?

n8n es una herramienta de automatización de tipo low-code / no-code. Permite construir flujos (workflows) conectando nodos, como si fueran piezas de LEGO:

Cada nodo realiza una tarea: leer datos, llamar a una API, tomar una decisión, transformar información, etc.

Los flujos se diseñan visualmente: se conectan nodos con flechas, indicando el orden de ejecución. Se puede mezclar lógica visual con código JavaScript en nodos especiales como Function y Code.

En este proyecto, n8n se usa como el cerebro del sistema:

Recibe los mensajes del Bot de Telegram.

## **TRABAJO DE GRADO**

### **Opción Seminario-Diplomado.**

8

Llama a Google Sheets para leer y escribir datos. Aplica reglas de negocio: verificar espacios libres, calcular horas, generar IDs de reserva, validar si el usuario puede cancelar, etc.

Devuelve respuestas dinámicas al usuario en Telegram.

### **3. ¿Qué otros servicios intervienen?**

#### **3.1. Telegram (Bot de mensajería)**

Se creó un Bot de Telegram que:

Recibe mensajes de texto (por ejemplo /start).

Muestra menús con botones interactivos (inline keyboards).

Envía mensajes automáticos al usuario según la lógica de n8n.

El Bot no “piensa” por sí mismo Solo es la interfaz de conversación. Quien decide qué hacer realmente es el workflow de n8n que está por detrás.

#### **3.2. Google Sheets (Base de datos sencilla)**

En lugar de usar una base de datos compleja (MySQL, PostgreSQL, etc.), se usó Google Sheets como base de datos simple, por ser:

Gratis.

Fácil de editar y revisar.

### **Desarrollo e Implementación del aprendizaje**

Se usan al menos estas hojas:

#### **1. USUARIOSFORM**

## **TRABAJO DE GRADO**

### **Opción Seminario-Diplomado.**

9

Guarda los datos del formulario de registro (nombre, cédula, teléfono, tipo de vehículo, placa, etc.).

Tiene un campo clave:

**CODIGO USUARIO (NO MODIFICAR!!!)**

que se asocia al chat\_id de Telegram.

## **2. ESPACIOS**

**Lista de espacios del parqueadero:**

**2.1.** Espacio\_id (ej.: E001, E002...)

**2.2.** Descripción

**2.3,** tipo (carro, moto, etc.)

**2.4 .** activo (TRUE / FALSE)

## **3. RESERVAS**

Guarda cada reserva creada:

**3.1** Reserva\_id

**3.2** usuario\_id

**3.3** espacio\_id

**3.4** fecha

**3.5** hora\_inicio

## **TRABAJO DE GRADO**

### **Opción Seminario-Diplomado.**

10

**3.6** hora\_fin

**3.7** Estado (reservado, pagado, cancelado, finalizado...)

**3.8** pago\_estado (pendiente, confirmado...)

**3.9** ttl\_min, created\_at, expires\_at, etc.

#### **4. Arquitectura general del Proyecto**

A un nivel alto, el funcionamiento se puede describir así:

**4.1.** El usuario escribe al Bot en Telegram (/start o toca un botón).

**4.2** Telegram envía ese evento a n8n mediante un webhook.

**4.3** n8n :

**4.3.1** Interpreta el mensaje (texto o callback).

**4.3.2** Consulta o escribe en Google Sheets.

**4.3.3** Decide qué hacer según el estado del usuario y sus reservas.

**4.3.4** Construye un mensaje de respuesta.

**4.4** n8n envía la respuesta de vuelta a Telegram.

**4.5** El usuario ve la respuesta en el chat (texto, botones, aviso de pago, etc.).

Este ciclo puede repetirse muchas veces hasta completar el flujo de:

Registro → Menú → Reservar → Pagar → Consultar → Cancelar / Salir.

#### **5-Herramientas y nodos utilizados en el workflow**

## **TRABAJO DE GRADO**

### **Opción Seminario-Diplomado.**

11

A continuación, se describen los tipos de nodos usados en el proyecto y su función.

BOT\_PARQUEADERO

#### **5.1. Nodos de Telegram**

##### **5.1.1 Telegram Trigger**

**Tipo:** n8n-nodes-base.telegramTrigger

**Función:** Es la puerta de entrada del workflow.

**Escucha:**

Mensajes nuevos (message)

Callbacks de botones (callback\_query)

Cada vez que el usuario escribe o pulsa un botón, este nodo dispara el flujo.

##### **5.1.2 Nodos de envío de mensajes de Telegram**

**Ejemplos en el workflow:**

Mensaje registrado

Mensaje no registrado

Mensaje en espera

Mensaje reservativa

Mensaje cancelar boton (en algunos casos vía HTTP Request directamente)

## TRABAJO DE GRADO Opción Seminario-Diplomado.

12

Mensaje p ago OK

Mensaje salir

Mensaje no espacios error

**Pueden ser de dos tipos:**

-Nodo **Telegram** nativo de n8n (n8n-nodes-base.telegram).

-Nodo **HTTP Request** que llama directamente a la API de Telegram:

**Su función general es:**

Enviar texto al usuario.

Adjuntar **inline keyboards** (botones).

Mostrar menús y opciones.

### 5.2 Nodos de Google Sheets

Se usan para conectar n8n con el archivo de Google Sheets del parqueadero.

#### 5.2.1 Buscar Usuario / Get Usuario

**Tipo:** n8n-nodes-base.googleSheets

**Operación:** leer filas filtrando por CODIGO USUARIO (NO MODIFICAR!!!) = chat\_id.

**Uso:** saber si el usuario ya está registrado o no.

#### 5.2.2 Get Espacios

## **TRABAJO DE GRADO**

### **Opción Seminario-Diplomado.**

13

Lee la hoja **ESPACIOS** filtrando por tipo de vehículo.

Permite saber qué espacios existen y cuáles están activos.

#### **5.2.3 Get RESERVAS / Get row(s) in sheet**

Lee la hoja **RESERVAS** para traer:

Reservas del usuario.

Estado actual de una reserva específica.

#### **Sirve para:**

Ver si el usuario ya tiene una reserva activa.

Consultar la última reserva.

Saber si hay reservas que se puedan cancelar.

#### **5.2.4 PagoOK (appendOrUpdate)**

Actualiza la fila de una reserva específica usando reserva\_id como llave.

Cambia los campos:

estado → pagado

pago\_estado → confirmado

Así, Google Sheets se comporta como una **base de datos relacional sencilla**, pero manejada desde una interfaz familiar.

## **TRABAJO DE GRADO**

### **Opción Seminario-Diplomado.**

14

#### **5.3 . Nodos de Lógica y Flujo (If, Switch, Merge)**

##### **5.3.1 If (MensajeOBoton)**

Nodo If

**Decide si la entrada viene de:**

Un mensaje normal o un callback de botón.

Permite tener ramas distintas según el tipo de interacción.

##### **5.3.2 If (Pago OMensaje)**

Nodo If

Revisa si el texto contiene "PAY\_".

Si lo contiene, se trata de un mensaje de confirmación de pago del simulador.

##### **5.3.3 If (Esta Registrado)**

Nodo If

Revisa el campo registrado que viene de Valida Registro.

Si true: envía menú principal.

Si false: envía mensaje con botón para registrarse.

## TRABAJO DE GRADO Opción Seminario-Diplomado.

15

### 5.3.4 Menu Opciones

Nodo Switch

Lee el callback\_data del botón pulsado en el menú principal.

**Deriva el flujo a:**

-Reservar cupo

-Consultar reserva

-Cancelar reserva

-Salir

### 5.3.5 Switch Duracion

Nodo Switch que interpreta distintos tipos de callback:

RESERVA\_ESPACIO\_...

RESERVA\_HORA\_...

RESERVA\_DURACION\_...

CANCELAR\_RESERVA\_...

De esta forma, un solo switch organiza las siguientes etapas del flujo de reserva.

### 5.3.6 Merge / CombinaDatos

Junta información que viene de distintos nodos:

## TRABAJO DE GRADO Opción Seminario-Diplomado.

16

Datos del usuario

Datos de espacios

Datos de reservas

Todo para armar el contexto que se necesita en el nodo Reservar Cupo.

### **5.4. Nodos de código: Function y Code**

Son nodos clave, porque permiten meter lógica propia en JavaScript.

#### **5.4.1 Parsear Mensaje (Function)**

Lee la estructura de Telegram (update).

Devuelve un objeto simplificado:

tipo (mensaje/callback)

chat\_id

user\_id

nombre

username

texto

callback\_data

Esto hace que todo el workflow trabaje con un formato uniforme.

#### **5.4.2 ValidaRegistro (Code)**

## **TRABAJO DE GRADO**

### **Opción Seminario-Diplomado.**

17

Toma lo que devolvió Google Sheets (USUARIOSFORM).

Construye:

registrado (true/false)

mensaje\_menu (texto principal del menú)

mensaje\_registro

url\_registro con el chat\_id prellenado en el form.

Devuelve todo listo para que los nodos de Telegram solo envíen el mensaje.

#### **5.4.3 Reservar Cupo (Function)**

**Toma:**

Usuario

Espacios

Reservas

Verifica si ya tiene una reserva activa.

Si tiene → arma un mensaje con el detalle de la reserva activa.

Si no tiene → filtra espacios libres y activos, arma la lista de botones y el mensaje con los espacios disponibles.

#### **5.4.4 Construir Reserva (Code)**

**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

18

**Interpreta el callback de duración:**

RESERVA\_DURACION\_{horas}\_{fecha}\_{hora\_inicio}\_{espacio\_id}

**Calcula:**

-hora\_fin

-ttl\_min

-Reserva\_id único.

**Construye:**

-Objeto reserva para escribir en Sheets.

-URL de pago simulada.

-Mensaje de confirmación de reserva con botón "Pagar".

**5.4.5 Consultar Reserva (Function)**

Recibe todas las reservas del usuario.

Elige la más reciente (por row\_number o created\_at).

**Genera un mensaje con:**

-ID de reserva

-Espacio

-Fecha

## TRABAJO DE GRADO Opción Seminario-Diplomado.

19

-Horario

-Estado

-Pago

-TTL (si aplica)

### 5.4.6 Cancelar Reserva (Function)

Recibe las reservas del usuario.

#### **Filtra aquellas:**

-No canceladas / no finalizadas.

-Con al menos 120 minutos antes de la hora de inicio.

#### **Toma la más reciente y arma:**

-Mensaje de confirmación de cancelación.

-Botón con CANCELAR\_RESERVA\_{reserva\_id}.

### 5.4.7 ValidaPago (Code)

A partir del texto (/start\nPAY\_...) extrae el reserva\_id.

Devuelve reserva\_id y chat\_id para que el nodo PagoOK actualice la fila correcta en Sheets.

### 5.4.8 Salir (Function)

## **TRABAJO DE GRADO**

### **Opción Seminario-Diplomado.**

20

Genera mensaje de salida:

“Has salido. Para iniciar el proceso de reserva usa /start”

Resetea la interacción del usuario de forma amigable.

### **6- Ventajas de esta arquitectura puntual para este Proyecto**

**6.1 Practica de automatización real** sobre un caso de uso concreto (parqueadero), no un ejemplo abstracto.

#### **6.2 Manejo de la integración:**

Un Bot (Telegram),

Un orquestador (n8n),

Y una base de datos sencilla (Google Sheets).

#### **6.3 Lógica condicional:**

If

Switch

Manejo de estados

**6.4 Construcción de IDs únicos**, Calcular fechas y horas, y manejar una “ventana” de cancelación.

**6.5 Viabilidad del proyecto**, donde no siempre se necesita una gran infraestructura:

## TRABAJO DE GRADO Opción Seminario-Diplomado.

21

Con herramientas gratuitas o de bajo costo se pueden resolver problemas reales de la vida diaria.

### 7 Paso a paso del funcionamiento completo del Bot de reservas

A continuación, se explica de forma detallada, clara y pedagógica, cada una de las etapas del flujo del Bot.

#### 7.1. Paso 1 — Inicio del Bot y detección de mensajes

Todo comienza cuando un usuario escribe el commando “/start”

O cuando oprime cualquier botón del Bot en Telegram, o escribe cualquier palabra.

El nodo **Telegram Trigger** recibe el evento y lo envía al nodo **Parsear Mensaje**, que transforma el contenido en un formato uniforme con estos campos:

7.1.1 tipo → mensaje o callback

7.1.2 chat\_id

7.1.3 user\_id

7.1.4 texto

7.1.5 callback\_data

7.1.6 message\_id

Esto permite que todo el workflow hable en el mismo “idioma”, sin importar si retorno de un botón o de un texto.

#### 7.2. Paso 2 — Validación de registro

El nodo **ValidaRegistro**:

## TRABAJO DE GRADO Opción Seminario-Diplomado.

22

- 7.2.1 Toma el chat\_id del usuario.
- 7.2.2 Consulta la hoja **USUARIOSFORM**.
- 7.2.3 Determina si el usuario ya está registrado (registrado = true/false).
- 7.2.4 Construye:

Mensaje de menú si está registrado.

Mensaje con enlace al formulario si no lo está.

### **Resultado:**

Si **NO** está registrado → Envía mensaje con botón “REGISTRO” que abre Google Forms prellenado con su chat\_id.

Si **SÍ** está registrado → Envía el **menú principal**.

### **7.3. Paso 3 — Menú Principal**

El menú tiene cuatro opciones:

- 7.3.1 Reservar cupo
- 7.3.2 Consultar reserva
- 7.3.3 Cancelar reserva
- 7.3.4 Salir

**Cada opción envía un callback\_data identificada:**

MENU\_1\_RESERVAR

MENU\_2\_CONSULTAR

## TRABAJO DE GRADO Opción Seminario-Diplomado.

23

MENU\_3\_CANCELAR

MENU\_4\_SALIR

El nodo **Menu Opciones (Switch)** dirige el flujo a la funcionalidad adecuada.

### 7.4. Flujo completo: Reservar un cupo

Este es el flujo más complejo y el núcleo del proyecto. Se ejecuta en varias etapas:

#### 7.4.1. Buscar espacios disponibles

El nodo **Get Espacios** obtiene todos los espacios aptos para el tipo de vehículo del usuario.

Luego, el nodo **Reservar Cupo**:

Toma todas las reservas activas para determinar cuáles espacios están ocupados.

Filtra espacios activos y libres.

Limita la lista a máximo 12 para evitar saturación.

Crea un teclado dinámico:

Ejemplo:

E001 (Carro)

E002 (Carro)

E003 (Carro)

...

## TRABAJO DE GRADO Opción Seminario-Diplomado.

24

Si no hay espacios → Envía mensaje:

“ Todos los espacios disponibles están ocupados en este momento.”

### 7.4.2. Selección del espacio

Cuando el usuario elige un espacio, por ejemplo:

E002

**El Bot recibe el callback:**

RESERVA\_ESPACIO\_E002

El nodo **Seleccionar Hora** calcula automáticamente las horas disponibles.

### 7.4.3. Cálculo de horas de inicio (bloques de 30 minutos)

El Bot:

**7.4.3.1** Toma la hora actual.

**7.4.3.2** La ajusta al siguiente “bloque”:

Minutos 0–29 → muestra HH:30

Minutos 30–59 → muestra la siguiente hora exacta HH+1:00

**7.4.3.3** Usa siempre la **zona horaria de Colombia** (America/Bogota).

**7.4.3.4** Genera 6 options:

**Ejemplo:**

2025-11-14 16:30

2025-11-14 17:00

2025-11-14 17:30

## TRABAJO DE GRADO Opción Seminario-Diplomado.

25

2025-11-14 18:00

2025-11-14 18:30

2025-11-14 19:00

Cada una con su propio callback:

RESERVA\_HORA\_INICIO\_2025-11-14\_16:30\_E002

### 7.4.4. Selección de duración

El Bot muestra opciones de duración (horas), usualmente:

1 hora

2 horas

3 horas

4 horas

5 horas

Callback ejemplo:

RESERVA\_DURACION\_3\_2025-11-14\_16:30\_E002

### 7.4.5. Construcción de la reserva

El nodo **Construir Reserva**:

Calcula la hora final.

Genera un reserva\_id único (`{chat_id}_{timestamp}`).

Genera un link de pago simulado.

Crea un objeto con todos los datos de la reserva:

Ejemplo:

## TRABAJO DE GRADO Opción Seminario-Diplomado.

26

reserva\_id

usuario\_id

espacio\_id

fecha

hora\_inicio

hora\_fin

estado = "reservado"

pago\_estado = "pendiente"

ttl\_min

payment\_url

Luego el Bot envía:

Tu reserva ha sido creada.

Para completarla, realiza el pago aquí:

[ Botón de pago ]

### 7.5. Flujo: Confirmación del pago

Cuando el usuario presiona "Pagar", el sitio de pago simulado devuelve un mensaje a Telegram:

/start

PAY\_{reserva\_id}

El nodo **ValidaPago** extrae el reserva\_id.

Luego **PagoOK** actualiza en Google Sheets:

estado = pagado

## TRABAJO DE GRADO Opción Seminario-Diplomado.

27

pago\_estado = confirmado

Finalmente, el Bot muestra:

Pago confirmado.

“Tu reserva ya está activa y ha sido registrada correctamente”.

### 7.6. Flujo: Consultar reserva

El Bot:

7.6.1 Obtiene todas las reservas del usuario.

. 7.6.2 Elige la más reciente usando row\_number o created\_at.

### 7.6.3 Muestra:

Detalle de tu reserva

Espacio:

Fecha

Horario

Estado

Pago

TTL restante

Esto permite al usuario saber cuál es su reserva activa o más reciente.

### 7.7. Flujo: Cancelar reserva

El Bot analiza:

7.7.1 Las reservas activas.

7.7.2 Si faltan **al menos 120 minutos** para el inicio.

## TRABAJO DE GRADO Opción Seminario-Diplomado.

28

### 7.7.3 Si no está ya:

Cancelada

Finalizada

Si la reserva es elegible, envía:

¿Confirmas que deseas cancelar esta reserva?

[ Sí, cancelar ]

Callback:

CANCELAR\_RESERVA\_{reserva\_id}

El sistema actualiza Google Sheets y confirma la cancelación.

### 7.8. Flujo: Salir

7.8.1 El usuario puede salir desde el menú.

7.8.2 El Bot simplemente envía:

Has salido.

Para iniciar el proceso de reserva usa /start

Y cierra la interacción de forma ordenada.

### 8. Probable evolución y mejoras futuras del sistema

Este proyecto contiene las funciones básicas y relevantes del manejo de un Bot de reservas automatizadas, pero tiene un potencial enorme para seguir creciendo.

#### 8.1. Integrar un pago real

En vez del simulador, se puede conectar:

Mercado Pago

## **TRABAJO DE GRADO** **Opción Seminario-Diplomado.**

29

Nequi Link de pago

Daviplata

Wompi

PayU

Stripe o Paypal (con USD)

Esto volvería el sistema **comercializable** para parqueaderos reales.

### **8.2. Notificaciones automáticas**

Notificaciones por Telegram cuando:

Faltan 30 minutos para comenzar la reserva.

La reserva va a expirar.

La reserva ha finalizado.

La reserva fue cancelada.

Esto mejora la experiencia del usuario y reduce pérdidas de cupos.

### **8.3. Implementar un panel administrativo web**

Para vigilantes o administradores:

Ver reservas activas.

Ver espacios ocupados.

Cancelar manualmente.

## **TRABAJO DE GRADO** **Opción Seminario-Diplomado.**

30

Cambiar estados.

Agregar o quitar espacios.

Reportes de uso.

Puede hacerse con:

Google App Script

Retool

Nocodb

Bubble

o un simple Dashboard en Google Data Studio.

### **8.4. Control de acceso físico**

Integrar:

Barreras automáticas

Lectores QR

Lectores de placas (LPR)

Cámaras

Sensores ultrasónicos para detectar ocupación

Los QR podrían generarse en n8n al crear la reserva.

### **8.5. Reserva avanzada por calendario**

## **TRABAJO DE GRADO**

### **Opción Seminario-Diplomado.**

31

Permitir al usuario reservar:

Mañana

Otra fecha en la semana

Esto implica un manejo más avanzado de solapamiento de horarios.

#### **8.6. Sistema de reportes inteligente**

Reportes automáticos:

Reservas por día.

Tasa de cancelación.

Tasa de ocupación real.

Horas pico.

Uso por tipo de vehículo.

Estos reportes pueden enviarse diario o semanalmente al administrador.

#### **8.9. Bot multilinguaje (ES/EN)**

Con un simple diccionario se puede internacionalizar el Bot.

#### **8.10. Lógica de tarifas variables**

Permitir:

Tarifas por hora

Tarifas nocturnas

Tarifas plan mes

## TRABAJO DE GRADO Opción Seminario-Diplomado.

32

Tarifas por tipo de vehículo

Tarifas especiales / visitantes

Y calcular el valor automático de la reserva.

### NODOS Y SU FUNCIONALIDAD

#### Nodo Parsear Mensaje (Function)

##### ¿Dónde está en el flujo?

Justo después del Telegram Trigger.

##### Qué recibe

El mensaje crudo que viene de Telegram:

Puede ser un **mensaje de texto** (/start, por ejemplo).

O un **callback** de un botón (MENU\_1\_RESERVAR, RESERVA\_ESPACIO\_E001, etc).

##### Qué hace

Detecta si lo que llegó es:

un **mensaje normal**, o un **click en un botón (callback)**.

A partir de eso arma un **JSON ordenado**, con campos estándar:

tipo: "mensaje" o "callback"

chat\_id

user\_id

nombre

## TRABAJO DE GRADO Opción Seminario-Diplomado.

33

username

texto (si es mensaje)

callback\_data (si es botón)

message\_id

### **Qué entrega**

Un solo objeto JSON limpito, por ejemplo:

Si es texto:

```
{  
  
  "tipo": "mensaje",  
  
  "chat_id": 12345,  
  
  "user_id": 12345,  
  
  "nombre": "Juan",  
  
  "username": "juan",  
  
  "texto": "/start",  
  
  "callback_data": null  
  
}
```

Si es callback:

```
{
```

## TRABAJO DE GRADO Opción Seminario-Diplomado.

34

```
"tipo": "callback",  
  
"chat_id": 12345,  
  
"user_id": 12345,  
  
"nombre": "Juan",  
  
"username": "juan",  
  
"texto": null,  
  
"callback_data": "MENU_1_RESERVAR"  
  
}
```

### **Nodo ValidaRegistro (Code)¿Dónde está?**

Después de leer en Sheets si el usuario está registrado (Buscar Usuario).

### **Qué recibe**

Del nodo de Google Sheets (USUARIOSFORM):

Los datos del formulario (si existe un registro con ese chat\_id).

Del Parsear Mensaje:

chat\_id, nombre, etc.

### **Qué hace**

1. Mira si **Sheets encontró o no una fila:**

Si hay row\_number → el usuario **está registrado.**

## TRABAJO DE GRADO Opción Seminario-Diplomado.

35

Si no hay fila → **no está registrado.**

2. Construye:

El **mensaje de menú** para usuarios registrados.

El **mensaje invitando a registrarse** si no está en la hoja.

3. Arma también la **URL del formulario prellenado** con el chat\_id.

### Qué entrega

Un JSON con:

chat\_id

registrado: true / false

Datos del usuario (si existen): nombre, cédula, teléfono, vehículo, placa...

mensaje\_menu: texto para el menú principal

mensaje\_registro: texto para invitar a registrarse

url\_registro: link al Forms con el chat\_id embebido

Ese JSON lo usan luego los nodos Mensaje Registrado y Mensaje No Registrado.

### ¿Nodo Reservar Cupo (Function)

#### ¿Dónde está?

En la rama de “1 Reservar cupo”, luego de:

Get Usuario

## TRABAJO DE GRADO Opción Seminario-Diplomado.

36

Get Espacios

Get RESERVASCombinaDatos

### Qué recibe

Datos del usuario (de Get Usuario):

tipo de vehículo, placa, código de usuario, etc.

Lista de espacios (ESPACIOS).

Lista de reservas (RESERVAS).

### Qué hace Identifica al usuario:

Saca usuario\_id, vehiculo, placa.

### Mira si ya tiene una reserva activa:

Revisa en la hoja RESERVAS si ese usuario\_id tiene una reserva **NO cancelada ni finalizada**.

Si ya tiene una reserva activa: Construye un mensaje con los detalles.

Y dice: "tienes una reserva activa, no te dejo crear otra".

### Si no tiene reserva activa, calcula los espacios libres:

Marca como "ocupados" todos los espacio\_id que estén en RESERVAS con estado activo. Toma de la hoja ESPACIOS solo los que:

tengan activo = TRUE,

## TRABAJO DE GRADO Opción Seminario-Diplomado.

37

no estén en la lista de ocupados.

**Arma el teclado de botones** con los espacios disponibles:

En filas de 3 botones (E001 (Carro), E002 (Carro), etc.).

Agrega botón final:

Cancelar → MENU\_4\_SALIR.

### Qué entrega

Según el caso:

Si ya tiene reserva activa:

```
{  
  
"chat_id": 12345,  
  
"crear_reserva": false,  
  
"mensaje": " Tienes una reserva activa...",  
  
"reserva_activa": { ... }  
}
```

Si puede crear reserva:

```
{  
  
"chat_id": 12345,  
  
"step": "ELEGIR_ESPACIO",  
}
```

## TRABAJO DE GRADO Opción Seminario-Diplomado.

38

```
"mensaje": " P Estos son los espacios disponibles...",  
  
"inline_keyboard": [ ... ],  
  
"usuario_id": "...",  
  
"vehiculo": "...",  
  
"placa": "...",  
  
"crear_reserva": true  
  
}
```

Ese inline\_keyboard lo usa luego MensajeEspacioBoton para mostrar los botones en Telegram. **Nodo Solicitar Hora (Code – después de elegir espacio)**

*(En el JSON sale como el Code que empieza con // === 1. Datos base === y usa RESERVA\_ESPACIO\_ en el callback)*

### **Qué recibe :**

chat\_idcallback\_data con el espacio elegido, por ejemplo:

RESERVA\_ESPACIO\_E005

A veces también:

usuario\_id

vehiculo

placa

## TRABAJO DE GRADO Opción Seminario-Diplomado.

39

### Qué hace

Extrae el espacio\_id del callback (E005).

Toma la **hora actual del servidor** y la redondea al próximo bloque:

Si son las 11:10 → empieza en 11:30.

Si son las 11:40 → empieza en 12:00.

Genera una lista de **6 horarios** en bloques de 30 minutos: Siempre en **hora de Colombia (America/Bogota)**.

Cada opción tiene:

texto: YYYY-MM-DD HH:MM

callback: RESERVA\_HORA\_INICIO\_FECHA\_HORA\_ESPACIO.

Crea un teclado inline con:

cada hora en una fila.un botón final:

Cancelar → MENU\_4\_SALIR.

### Qué entreg

Un JSON como:

```
{  
  "chat_id": 12345,  
  "mensaje": " Espacio seleccionado: E005..."
```

## TRABAJO DE GRADO Opción Seminario-Diplomado.

40

```
"inline_keyboard": [ ...lista de horas... ],  
"usuario_id": "...",  
"vehiculo": "...",  
"placa": "..."  
}
```

Que luego MensajeHorasBoton manda a Telegram.

### **Nodo Construir Reserva (Code)**

#### **¿Dónde está?**

En la rama de Switch Duracion, cuando el usuario ya eligió:

Espacio

Hora de inicio

Duración

#### **Recibe :**

callback\_data con todo empacado, por ejemplo:

RESERVA\_DURACION\_3\_2025-11-14\_18:00\_E009

chat\_id del usuario.

#### **Qué hace**

#### **Desarma el callback\_data:**

duracion\_horas = 3

fecha = 2025-11-14

**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

41

hora\_inicio = 18:00

espacio\_id = E009 **Calcula hora\_fin:**

Súmame duracion\_horas a la hora de inicio (formato 24h).

**Genera un reserva\_id único:**

chat\_id + timestamp.

**Construye la URL de pago:**

[https://strong-beijinho.../?reserva\\_id=...&usuario\\_id=...](https://strong-beijinho.../?reserva_id=...&usuario_id=...)

**Formatea las fechas de creación y expiración en hora Colombia:**

created\_at

expires\_at

**Arma el texto completo que se enviará por Telegram:** Fecha

Horario

Espacio

Duración

Link de pago

**Prepara el reply\_markup con el botón:**

Pagar reserva (URL).

## TRABAJO DE GRADO Opción Seminario-Diplomado.

42

### Qué entrega

Un objeto reserva listo para:

Guardar en Google Sheets (hoja RESERVAS).

Enviar por Telegram.

Ejemplo:

```
{  
  "reserva_id": "...",  
  "usuario_id": 12345,  
  "espacio_id": "E009",  
  "fecha": "2025-11-14",  
  "hora_inicio": "18:00",  
  "hora_fin": "21:00",  
  "estado": "reservado",  
  "pago_estado": "pendiente",  
  "ttl_min": 180,  
  "created_at": "2025-11-14 16:30:00",  
  "expires_at": "2025-11-14 19:30:00",  
  "payment_url": "https://...",  
  "mensaje_telegram": "Tu reserva ha sido creada...",  
  "reply_markup": { ... } } Nodo Consultar Reserva (Function)
```

### Qué recibe

## TRABAJO DE GRADO Opción Seminario-Diplomado.

43

Una lista de reservas de ese usuario (Get row(s) in sheet sobre RESERVAS).

### Qué hace

#### Si no hay ninguna reserva:

Organiza un mensaje tipo:

“No tienes reservas registradas”.

Si la respuesta es positive :

Busca la **más reciente** (por row\_number).Extrae:

Espacio

Fecha

hora inicio/fin

estado

pago

ttl (tiempo restante) si aplica.

Construye un mensaje resumen.

#### Qué entrega

JSON con:

chat\_id

tiene\_reserva: true/false

mensaje: texto con el detalle de la última reserva

reserva: objeto de la reserva encontrada

#### Nodo Cancelar Reserva (Function)

## TRABAJO DE GRADO Opción Seminario-Diplomado.

44

### Qué recibe

Lista de reservas del usuario (de la hoja RESERVAS).

### Qué hace

#### 1. **Filtra solo reservas elegibles para cancelación:**

- No deben estar en estado CANCELADO/FINALIZADO.
- Deben faltar **al menos 120 minutos** (2 horas) para el inicio.

#### 2. **Si no hay reservas elegibles:**

- Toma la más reciente y dice:

“Tu reserva ya no se puede cancelar porque faltan menos de 2 horas...”.

#### 3. **Si encuentra elegibles:**

- Toma la más reciente
- Calcula minutos que faltan para el inicio.
- Construye mensaje de confirmación.
- Arma teclado con:

botón: Sí, cancelar reserva con callback CANCELAR\_RESERVA\_<reserva\_id>.

**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

45

**Qué entrega**

**Si no se puede cancelar:**

puede\_cancelar: false

mensaje: texto explicando el por que no se puede cancelar

**Si se puede cancelar:**

puede\_cancelar: true

mensaje: texto con resumen de la reserva a cancelar

inline\_keyboard: botón de confirmación **Nodo ValidaPago (Code)**

**Qué recibe**

El texto que llega cuando el usuario regresa del pago:

Ej: /start PAY\_391459174\_1763154749455

**Qué hace**

Busca la parte que viene después de PAY\_.

Extrae el reserva\_id:391459174\_1763154749455.

**Qué entrega**

```
{  
  "reserva_id": "391459174_1763154749455",  
  "chat_id": 12345  
}
```

## TRABAJO DE GRADO Opción Seminario-Diplomado.

46

El flujo es usado para:

Buscar esa reserva en RESERVAS.

Marcarla como pagado / confirmado.

### **Nodo Salir (Function)**

#### **Qué recibe**

chat\_id (ya sea del ítem actual o desde “Menú Opciones”).

**Qué hace** Garantiza que tiene el chat\_id.

Construye un mensaje simple:

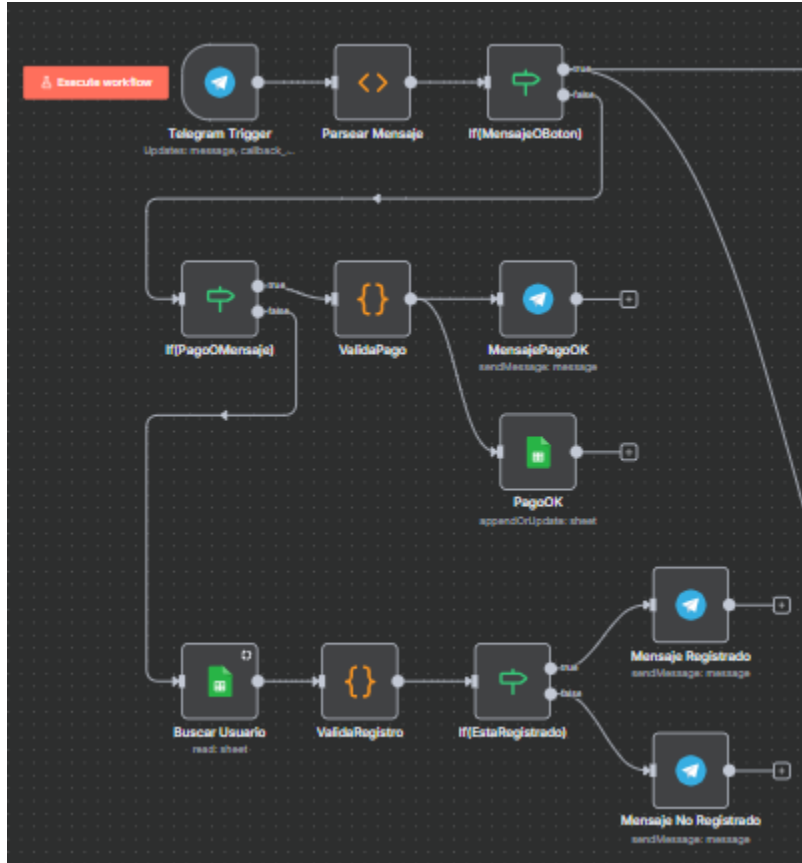
"Has salido. Para iniciar el proceso de reserva usa /start"

#### **Qué entrega:**

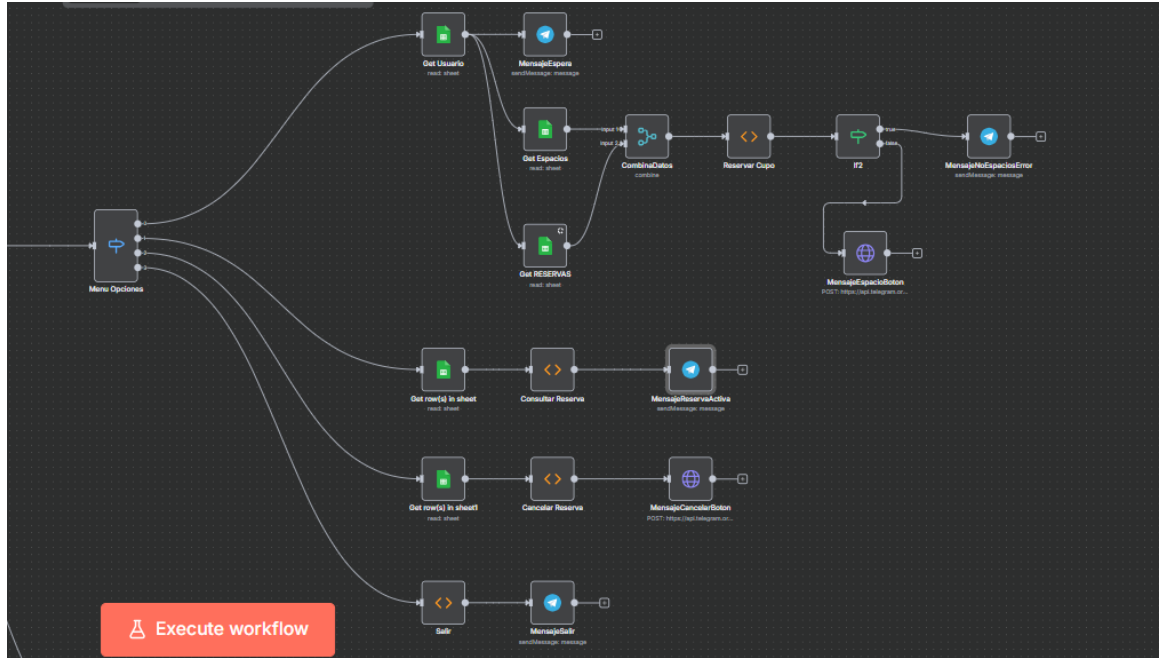
```
{  
  "chat_id": 12345,  
  "mensaje": "Has salido...\n\nPara iniciar el proceso de reserva usa /start"  
}
```

Que luego Mensaje Salir envía a Telegram.

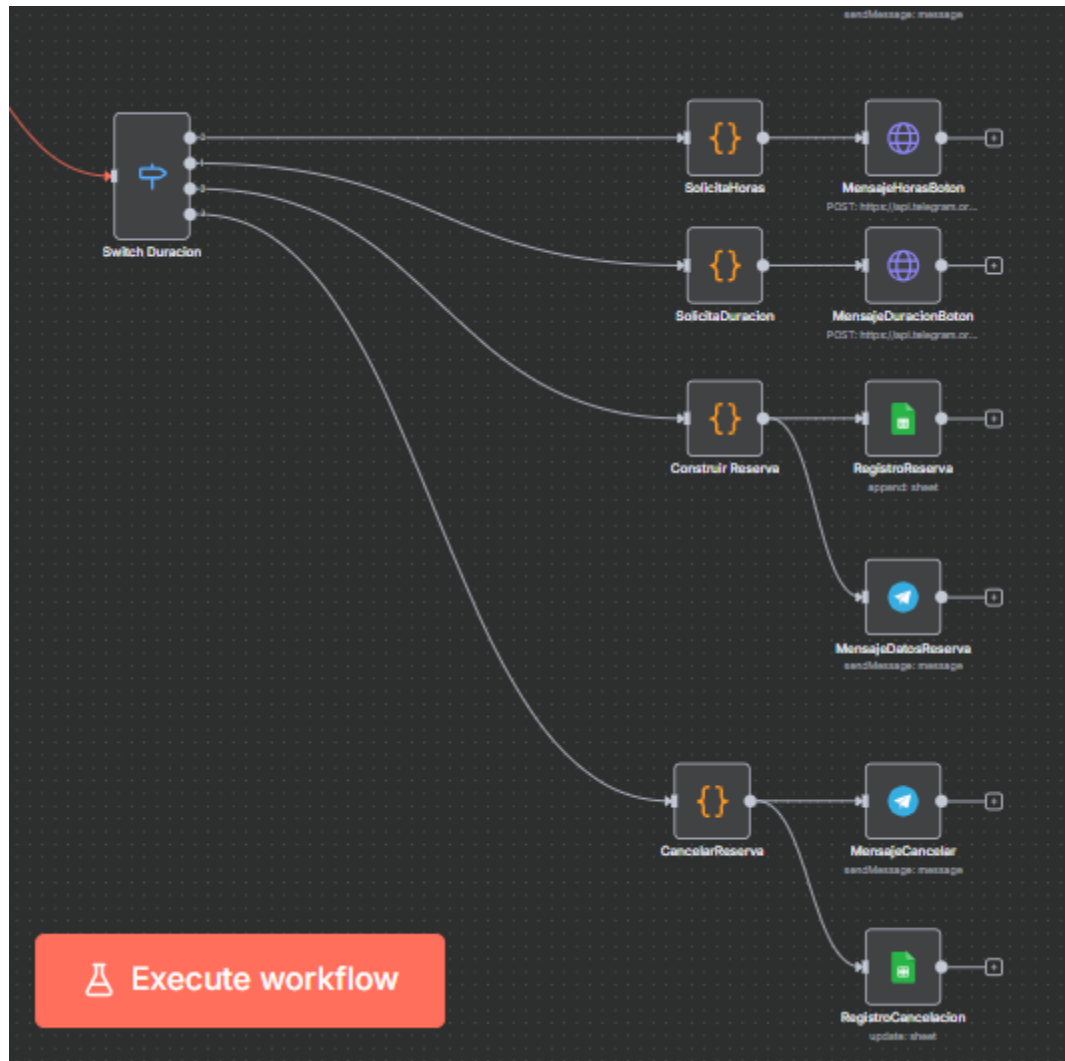
**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**



## TRABAJO DE GRADO Opción Seminario-Diplomado.



**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**



**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

50

**Conclusiones**

La implementación de un sistema automatizado para la gestión del parqueadero, utilizando n8n como motor de orquestación de flujos y Telegram como interfaz de comunicación, permitió validar la eficacia de las herramientas de automatización en entornos operativos cotidianos. Esta integración demostró que es posible reducir significativamente la carga manual de tareas repetitivas, como el registro de entradas y salidas, la notificación de disponibilidad y la interacción con los usuarios, mediante procesos automatizados que operan en tiempo real.

Desde una perspectiva técnica, el proyecto implicó el diseño de un flujo modular en n8n que conecta múltiples nodos para la recepción de eventos, el procesamiento de datos y la emisión de respuestas a través de la API de Telegram. Este enfoque no solo optimizó la eficiencia operativa, sino que también mejoró la trazabilidad de las acciones y la consistencia en la comunicación con los usuarios finales.

Más allá de la arquitectura tecnológica, el desarrollo del sistema evidenció la importancia de estructurar procesos claros, anticipar escenarios de uso y diseñar soluciones que respondan a necesidades concretas del entorno. La automatización no se limitó a la ejecución de tareas, sino que se convirtió en una herramienta estratégica

**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

51

para mejorar la experiencia del usuario y facilitar la toma de decisiones basada en datos.

En conclusión, el resultado del proyecto confirma que incluso procesos aparentemente simples pueden beneficiarse de una transformación significativa cuando se aplican tecnologías de automatización adecuadas. La combinación de n8n y Telegram se consolidó como una solución robusta, escalable y adaptable, capaz de aportar valor tangible en contextos operativos reales.

**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

52

**Referencias**

- Telegram. (s.f.). Telegram Bot API. <https://core.telegram.org/bots/api>
- Telegram. (s.f.). Creating Bots. <https://core.telegram.org/bots#creating-a-new-bot>
- N8n. (s.f.). n8n Documentation. <https://docs.n8n.io>
- N8n. (s.f.). Telegram Node (n8n-nodes-base.telegram).  
<https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.telegram/>
- N8n. (s.f.). Function Nodes (JavaScript). <https://docs.n8n.io/code/function-nodes/>
- N8n. (s.f.). HTTP Request Node. <https://docs.n8n.io/integrations/builtin/core-nodes/n8n-nodes-base.httprequest/>
- Google Developers. (s.f.). Google Sheets API — Concepts.  
<https://developers.google.com/sheets/api/guides/concepts>
- Mozilla Developer Network (MDN). (s.f.). JavaScript Guide.  
<https://developer.mozilla.org/es/docs/Web/JavaScript/Guide>
- Fazt. (2020). JavaScript en 1 hora. YouTube.  
<https://www.youtube.com/watch?v=z95mZVUcJ-E>
- Red Hat. (s.f.). ¿Qué es una API?. <https://www.redhat.com/es/topics/api/what-are-application-programming-interfaces>

**TRABAJO DE GRADO**  
**Opción Seminario-Diplomado.**

53

SendPulse. (s.f.). Telegram Bot: Concepto y funcionamiento.

<https://sendpulse.com/support/glossary/telegram-bot>

Fireship. (2022). Telegram Bots in 100 Seconds. YouTube.

<https://www.youtube.com/watch?v=jrHIYzCB6ho>

Código Facilito. (2019). Cómo crear un bot de Telegram. YouTube.

<https://www.youtube.com/watch?v=4tJPNQw1HIE>