

**DISEÑO E IMPLEMENTACIÓN DE APLICACIÓN TIPO  
WEBHOOK COMO BACKEND QUE RESPONDA A  
ROBOT DE TELEGRAM PARA SOPORTAR LA  
GESTIÓN DE RECURSOS DE LABORATORIOS DE LA  
UNIVERSIDAD TECNOLÓGICA DE PEREIRA.  
CASO DE ESTUDIO GRUPO DE INVESTIGACIÓN SIRIUS.**

Edilson Londoño Castañeda  
Sebastián Vélez Montoya

Trabajo de grado presentado como requisito  
para obtener el título de  
**Ingeniero de Sistemas y Computación**

Director  
Ramiro Andrés Barrios Valencia

UNIVERSIDAD TECNOLÓGICA DE PEREIRA  
Programa de Ingeniería de Sistemas y Computación.  
Pereira, Julio de 2020



DISEÑO E IMPLEMENTACIÓN DE APLICACIÓN TIPO WEBHOOK COMO BACKEND QUE  
RESPONDA A ROBOT DE TELEGRAM PARA SOPORTAR LA GESTIÓN DE RECURSOS DE  
LABORATORIOS DE LA UNIVERSIDAD TECNOLÓGICA DE PEREIRA.  
CASO DE ESTUDIO GRUPO DE INVESTIGACIÓN SIRIUS.

Esta obra está licenciada bajo la Licencia Creative Commons Reconocimiento - No Comercial (by-nc) la cual permite la generación de obras derivadas siempre que no se haga un uso comercial de las mismas. Tampoco se puede utilizar la obra original con finalidades comerciales.



# Agradecimientos

A nuestros padres Fernando Londoño Galeano, Martha Castañeda Castaño, Adriana Montoya Pelaez y Aldemar de Jesus Velez por su apoyo incondicional, siendo ellos nuestra motivación para seguir adelante sin ellos nada de esto hubiese sido posible.

A Maria del Pilar Montoya por su apoyo desde el principio de la carrera y por siempre creer en mis capacidades.

A la Universidad Tecnológica de Pereira por ser la institución que forjó nuestras bases de conocimiento, ética y disciplina.

A Ramiro Andrés Barrios Valencia nuestro director de proyecto, por la orientación, aportes en nuestro crecimiento tanto personal como profesional y en la confianza depositada en nosotros.

A la comunidad Open Source Pereira, por incentivar el aprendizaje colectivo y la pasión por compartir conocimiento.

A la facultad de Ingeniería de Sistemas y Computación, profesores y compañeros por su asesoría y acompañamiento durante nuestra formación.



# Tabla de Contenido

1.	Introducción	7
2.	Planteamiento del problema	9
3.	Justificación	11
4.	Objetivos generales y específicos	13
4.1.	Objetivo general. . . . .	13
4.2.	Objetivos específicos. . . . .	13
5.	Metodología	15
5.1.	Control del Flujo. . . . .	16
5.2.	Manejo de Versionamiento . . . . .	16
6.	Marco Referencial	17
6.1.	Antecedentes. . . . .	17
6.2.	Marco teórico . . . . .	18
6.3.	Marco conceptual. . . . .	19
7.	Diseño	20
7.1.	Análisis. . . . .	20
7.2.	Alcance. . . . .	20
7.3.	Historias de usuario . . . . .	21
7.4.	Casos de uso. . . . .	23
7.5.	Modelo de vistas de arquitectura 4+1	30
7.5.1.	Diagrama de casos de uso. . . . .	30
7.5.2.	Diagrama de secuencia. . . . .	31
7.5.3.	Diagrama de actividades . . . . .	32
7.5.4.	Diagrama de clases. . . . .	33
7.5.5.	Diagrama de despliegue. . . . .	33
7.5.6.	Diagrama de componentes. . . . .	34

<b>8.</b>	<b>Plan de Pruebas</b>	<b>35</b>
8.1.	Historial de Versiones. . . . .	35
8.2.	Resumen Ejecutivo. . . . .	35
8.3.	Criterios de Aceptación o Rechazo. . . . .	36
<b>9.</b>	<b>Resultados del Plan de Pruebas</b>	<b>37</b>
9.1.	Incidencias Encontradas. . . . .	37
<b>10.</b>	<b>Conclusiones</b>	<b>39</b>
<b>11.</b>	<b>Trabajos Futuros</b>	<b>40</b>
<b>12.</b>	<b>Bibliografía</b>	<b>41</b>
<b>13.</b>	<b>Anexos: Manual de Usuario</b>	<b>43</b>

# Capítulo 1

## Introducción

En el ámbito de la gestión de proyectos, se han desarrollado procesos, técnicas y filosofías sobre el mejor enfoque para la asignación de recursos. Estos incluyen debates sobre la asignación de recursos funcionales frente a la multifuncional, así como procesos propuestos por organizaciones como el Project Management Institute (PMI) a través de su metodología de gestión de proyectos (Project Management Body of Knowledge, PMBOK).<sup>1</sup>

Estas metodologías enfatizan mucho el uso de personal administrativo, el cual tiene como objetivo encargarse de la gestión, aunque esto no garantiza el flujo de sus procesos en su totalidad debido a que se presentan nuevos problemas durante el desarrollo como: La falta de capacitación, exceso o falta de personal, métodos de selección poco apropiados, problemas de comunicación, escalabilidad, entre otros.

Esto ha llevado a las empresas y organizaciones a buscar otros tipos de estrategia orientados hacia el uso de la tecnología y adaptándose a la llegada de la cuarta revolución industrial. Según un estudio, “Más del 90 % de los altos directivos que utilizan la automatización inteligente señala que su organización tiene un rendimiento superior a la media en términos de gestión del cambio organizativo como respuesta a las nuevas tendencias empresariales”.<sup>2</sup> Las bases tecnológicas en que se apoya esta orientación, entre otras son las siguientes: Big data y análisis de datos, Cloud Computing, Ciberseguridad, Robótica e Internet de las cosas.<sup>3</sup>

Especialmente en el área de la robótica se encuentran metodologías de aplicación que pueden automatizar la gestión de recursos. “Más del 90 % de los altos directivos afirman que ya existe en sus empresas cierto nivel de automatización inteligente”.<sup>4</sup> Los Bots son una de estas aplicaciones de software que ejecutan tareas automatizadas más conocidas como scripts a través de internet y son un futuro cada vez más presente. El Microsoft Inspire 2018 recalcó: “En 2025 los bots protagonizarán el 95% de las interacciones con los clientes. La intervención humana se reservará sólo para casos muy puntuales, con los que se potenciará la efectividad y se optimizarán los recursos”.<sup>5</sup>

La tecnología aumenta la comodidad humana, haciendo cada vez las tareas de una forma más rápida. Ahora las nuevas generaciones están interesadas en conseguir respuestas rápidas a preguntas concretas. Y esa función es la que los bots cumplen a la perfección.

## CAPÍTULO 1. INTRODUCCIÓN

# Capítulo 2

## Planteamiento del problema

El crecimiento de las organizaciones, no solo de personal sino de infraestructura, ha hecho que la mala gestión de los recursos se convierta en una problemática que afecta el buen flujo de trabajo.<sup>2</sup> La concurrencia puede ocasionar retrasos en los procesos, incluso en el peor de los casos, que estos no se lleven a cabo.

Esto ha llevado a las organizaciones a buscar metodologías para la administración de los recursos haciendo uso de personal. Pero esta alternativa genera gastos, hace que el proceso sea más lento, el personal esté expuesto a errores, los recursos dependan de la disposición de un tercero y además que no se lleve un control de daños. Las aplicaciones pueden ser una solución viable, pero se convierte en una responsabilidad más para los grupos ya que deben disponer de recursos para su alojamiento y se debe dar mantenimiento periódico.

En el caso del grupo de investigación Sirius, se presenta una problemática desde la administración de espacios físicos y el manejo de hardware ya que al momento de hacer uso de un espacio físico, no se tiene un proceso de reserva, por lo que simplemente una persona accede a la sala sin poder consultar previamente si esta se encuentra disponible. Esto causa problemas de concurrencia al momento en que una o mas personas intenten tener acceso a una sala y esta no se encuentre disponible en el momento.

En cuanto a hacer uso de un dispositivo de hardware, la persona toma el dispositivo sin notificar a nadie, generando dificultades al momento de que otra persona requiera de este dispositivo y no lo encuentre en su respectivo lugar.

Teniendo en cuenta la problemática que presenta la gestión de recursos y que las metodologías usadas para optimizar estos procesos no garantizan un mejoramiento considerable; ¿Se puede utilizar tecnologías de la información (y/o bots) como gestor de recursos físicos para los laboratorios de la UTP, bien sea como herramienta de soporte o alternativa al personal de laboratorio? y si pudiéramos medir el impacto sobre los recursos de tiempo, dinero, humano y ambiental, ¿estos impactos serían positivos?

## CAPÍTULO 2. PLANTEAMIENTO DEL PROBLEMA

# Capítulo 3

## Justificación

El uso de recursos de una organización es indispensable para llevar a cabo sus procesos. Este proyecto hace más fácil el control y la administración de una manera automatizada que evita pérdidas de tiempo y retrasos en el buen desarrollo del grupo, ya sea desde la ejecución de las actividades como la relación entre sus miembros. Esta herramienta beneficiará en muchos aspectos los grupos ya que se reducirán costos, se evitará la necesidad de terceros y se tendrá información de manera rápida y eficaz.

Además al momento de llevar un seguimiento de las actividades realizadas se pueden prevenir daños y podrá mejorar la distribución de los recursos y así optimizar y regular procesos de entrega y tiempos de uso ya sea de lugares y/o dispositivos.

## CAPÍTULO 3. JUSTIFICACIÓN

# Capítulo 4

## Objetivos generales y específicos

### 4.1. Objetivo General

Diseñar e implementar una aplicación tipo webhook como backend que responda a robot de telegram que soporte la gestión de recursos de laboratorios de la Universidad Tecnológica de Pereira. Caso de estudio Grupo de investigación Sirius.

### 4.2. Objetivos Específicos

1. Investigar el funcionamiento de la automatización a través de bots.
2. Consultar el manejo de los recursos de la infraestructura interna del laboratorio.
3. Elaborar un análisis del sistema.
4. Realizar el diseño de la aplicación.
5. Implementar los requerimientos de la aplicación.
  - 5.1. Implementar la aplicación tipo webhook integrado a bot de telegram.
  - 5.2. Implementar el registro de la información obtenida con el bot.
6. Llevar a cabo las pruebas del sistema.

## CAPÍTULO 4. OBJETIVOS GENERALES Y ESPECÍFICOS

# Capítulo 5

## Metodología

Para el desarrollo del proyecto se utilizó la metodología de desarrollo ágil Kanban debido a que usa un mecanismo de control visual para hacer seguimiento del trabajo conforme este viaja a través del flujo de valor.

En Kanban se requiere el uso de un tablero, donde se representa cada uno de los estados por los cuales puede pasar una tarea. El tablero está estructurado por columnas que van a ser estar divididas por: Pendiente, Seleccionadas, En desarrollo, Urgente, Testeo y Finalizado. Este tablero se conoce como Workflow y nos permite hacer retrospectivas del flujo de trabajo.

Por otro lado tenemos las Tarjetas, las cuales contienen el nombre de la tarea, la información como por ejemplo: la descripción, la fecha por la que esta ha pasado por una columna, persona encargada de desarrollar la tarea en cada uno de los estados y demás información que se considere relevante. Al tener una tarea atrasada por falta de información, dicha tarea será señalizada y puesta en otra columna del tablero para ser replanteada.<sup>6</sup>

Se utilizó Kanban porque es un sistema simple y sencillo de implementar y permite el desarrollo de una metodología ágil como scrum, pero con la ventaja de que nos permite tener grupos de trabajo reducidos, da una vista a todos los integrantes del grupo sobre las tareas que se están ejecutando, esto permite que no ocurran problemas al llevar la misma tarea por diferentes personas, garantizando una visualización del desarrollo del proyecto. Gracias a su estructura permite detectar retrasos y permite replantear una tarea identificando y realizando una retrospectiva de las dificultades presentadas.<sup>7</sup>

Limitar el trabajo en proceso es una de las principales prácticas que enmarca el método Kanban y lo hace tan eficiente. Los límites WIP de Kanban aseguran que su equipo mantendrá un ritmo de trabajo óptimo sin exceder su capacidad de trabajo.<sup>8</sup>

## 5.1. Control del Flujo

Para llevar un mejor control del flujo de trabajo etiquetaron por colores de la siguiente manera:

### **Flujo 1:**

Color: Amarillo

#### **Componentes:**

1. Investigar el funcionamiento de la automatización a través de bots.
2. Consultar el manejo de los recursos de la infraestructura interna del laboratorio.
3. Elaborar requerimientos del sistema

### **Flujo 2:**

Color: Azul

#### **Componentes:**

4. Realizar el diseño de la aplicación.

### **Flujo 3:**

Color: Naranja

#### **Componentes:**

- 5.1. Implementar la aplicación tipo webhook integrado a bot de telegram

### **Flujo 4:**

Color : Fucsia

#### **Componentes :**

- 5.2. Implementar el registro de la información obtenida con el bot.

### **Flujo 5:**

Color: Verde

#### **Componentes:**

6. Llevar a cabo las pruebas del sistema

Se realizaron reuniones presenciales cada semana, cada una de un tiempo máximo de 20 minutos donde los miembros del equipo mostraron su progreso y dificultades presentadas. El flujo de trabajo se realizó por medio de la herramienta Trello la cual proporciona un tablero en donde cada uno de los miembros involucrados puede ver el desarrollo de las tareas en simultáneo.

## 5.2. Manejo de Versionamiento

El desarrollo se realizó a través del sistema de control de versiones llamado git ya que nos permite realizar trabajo en paralelo. Además, el proyecto se ha almacenado en el servidor de Github el cual nos proporciona un repositorio privado y remoto, realizando un control detallado del flujo y de las incidencias.

# Capítulo 6

## Marco Referencial

### 6.1. Antecedentes

Los chatbots se desarrollaron por primera vez a mediados de los años 60. “Elizta” fue el primer chatbot creado en el MIT (Instituto de Tecnología de Massachusetts) en 1966, que más tarde se denominó “chatter bot”. El objetivo principal de este chatbot era habilitar una interfaz de comunicación para conectarse entre la máquina y el hombre. Al ser una innovación pionera de la época, Elizta ganó una gran popularidad al imitar las respuestas de un psicoterapeuta no direccional en una entrevista psiquiátrica preliminar.

La revolución de los chatbots se produjo con el lanzamiento del asistente personal exclusivo de Apple “Siri”. Siri presenta una amplia gama de comandos de usuario, que incluyen la programación de eventos, la verificación de información básica, el manejo de la configuración del dispositivo, la realización de acciones telefónicas, recordatorios, la navegación por Internet, la navegación por áreas y la posibilidad de interactuar con las aplicaciones de iOS. Watson de IBM se expandió aún más en la interrupción a través del juego de procesamiento de lenguaje natural.

Amazon lanzó Alexa y Google desarrolló su propio agente de diálogo “DialogFlow” que ofrece capacidades de procesamiento de lenguaje natural. El agente de Google también ofrece una plataforma de integración de una ventana para trabajar con marcos de Facebook, Twitter, Cortana, Skype, Slack, Alexa y muchos más.

Las empresas líderes están utilizando BOTS para aumentar su eficiencia. E incluso se están implementando este tipo de tecnologías en diversas industrias, como el comercio minorista, el entretenimiento y las redes sociales, entre otras, no solo para adquirir información sino también para actuar de acuerdo con la intención de los usuarios. Los chatbots han penetrado en todas las operaciones que van desde responder consultas hasta obtener patrones de usuario.

Una encuesta reciente revela que alrededor del 80% de las empresas en todo el mundo planean implementar un sistema de chatbot funcional para 2020. Como las empresas orientadas al consumidor tienen que ver con la experiencia, tener un sistema de chatbot integrado con su infraestructura actual sólo aumentará su ventaja.<sup>9</sup>

## 6.2. Marco Teórico

### 6.2.1. Webhooks

Los webhooks son llamadas Http post que envían solicitudes a una url especificada por un usuario cuando un evento sucede, Los webhooks logran conexiones de larga duración y son usados para enviar eventos en tiempo real.

Los webhooks son "devoluciones de llamada HTTP definidas por el usuario". Suelen activarse por algún evento, como enviar código a un repositorio o publicar un comentario en un blog. Cuando se produce ese evento, el sitio de origen realiza una solicitud HTTP a la URL configurada para el webhook. Los usuarios pueden configurarlos para que los eventos en un sitio invoquen el comportamiento en otro.<sup>10</sup>

### 6.2.2. API

Es una interfaz informática que define las interacciones entre múltiples intermediarios de software. Define los tipos de llamadas o solicitudes que se pueden hacer, cómo hacerlas, los formatos de datos que se deben usar, las convenciones a seguir, etc. También puede proporcionar mecanismos de extensión para que los usuarios puedan ampliar la funcionalidad existente de varias maneras y en diferentes grados.<sup>11</sup>

#### **Representational state transfer (REST)**

Es un estilo arquitectónico de software que define un conjunto de restricciones que se utilizarán para crear servicios web. Los servicios web REST permiten a los sistemas solicitantes acceder y manipular representaciones textuales de recursos web mediante el uso de un conjunto uniforme y predefinido de operaciones sin estado . Otros tipos de servicios web, como los servicios web SOAP , exponen sus propios conjuntos arbitrarios de operaciones.<sup>12</sup>

#### **Arquitectura Cliente Servidor**

Es una estructura de aplicación distribuida que divide tareas o cargas de trabajo entre los proveedores de un recurso o servicio, llamados servidores, y los solicitantes de servicios, llamados clientes. A menudo, los clientes y servidores se comunican a través de una red informática en un hardware separado, pero tanto el cliente como el servidor pueden residir en el mismo sistema.<sup>13</sup>

## **6.3. Marco Conceptual**

### **6.3.1. Bots**

Es un tipo de agente de software al servicio de la gestión de proyectos de software y la ingeniería de software. Un bot de software tiene una identidad y aspectos potencialmente personificados para servir a sus partes interesadas.<sup>14</sup>

### **6.3.2. Chatbots**

Es un programa que simula mantener una conversación con una persona al proveer respuestas automáticas a entradas hechas por el usuario. Habitualmente, la conversación se establece mediante texto, aunque también hay modelos que disponen de una interfaz de usuario multimedia.<sup>15</sup>

### **6.3.3. Backend**

Backend es la capa de acceso a datos de un software o cualquier dispositivo, que no es directamente accesible por los usuarios, además contiene la lógica de la aplicación que maneja dichos datos.<sup>16</sup>

### **6.3.4. Comando**

Es una instrucción que el usuario proporciona a un sistema informático, desde la línea de órdenes. Siendo la instrucción el conjunto de datos insertados en una secuencia estructurada.<sup>16</sup>

### **6.3.5. Gestión de Recursos**

Es la implementación eficiente y eficaz de los recursos de una organización cuando se necesitan. Estos recursos pueden incluir recursos financieros, inventario, habilidades humanas, los recursos de producción, o tecnología de la información.<sup>17</sup>

# Capítulo 7

## Diseño del sistema

### 7.1. Análisis

En la siguiente sección se busca especificar de forma clara la arquitectura y el diseño de la aplicación, además de explicar cómo interactúan los distintos usuarios con la aplicación.

De igual manera, se busca plasmar en el presente los términos claves que definen estructuralmente el sistema y cómo en estos se manifiesta el cumplimiento de los requisitos.

### 7.2. Alcance

Este proyecto es realizado en el marco de proyecto de grado para la carrera Ingeniería de Sistemas y Computación de la Universidad Tecnológica de Pereira bajo la dirección del director Ramiro Andrés Barrios. El proyecto parte de los requerimientos y apunta a la finalización de una aplicación funcional desplegada en un servidor de Heroku. La aplicación será accesible desde la red social Telegram. El primer prototipo funcional se va a implementar para el grupo de investigación sirius, y tendrá como características la administración y gestión de recursos en el laboratorio, ya sean dispositivos o espacios.

### 7.3. Historias de usuario

Durante la primera etapa de desarrollo se realizó el levantamiento de requerimientos por medio de historias de usuario, las cuales van a estar representadas por un conjunto de tablas.

Historia:	Agregar el Bot
ID:	1
Rol:	Investigador
Descripción:	Se debe permitir agregar el bot por medio de la aplicación telegram.

*Tabla 7.1: Historia 1. Agregar el Bot. Referencia: Investigador.*

Historia:	Reservar Recurso
ID:	2
Rol:	Investigador
Descripción:	Se debe permitir reservar un recurso y guardarlo automáticamente en un historial de google drive.

*Tabla 7.2: Historia 2. Reservar Recurso. Referencia: Investigador.*

Historia:	Listar Recursos
ID:	3
Rol:	Investigador
Descripción:	Se debe permitir listar los recursos almacenados en un documento de google drive.

*Tabla 7.3: Historia 3. Listar Recursos. Referencia: Investigador.*

Historia:	Entregar Recurso
ID:	4
Rol:	Investigador
Descripción:	Se debe permitir entregar una reserva realizada con anterioridad, y almacenar el historial en el documento de google automáticamente.

*Tabla 7.4: Historia 4. Entregar Recurso. Referencia: Investigador.*

Historia:	Cancelar Reserva
ID:	5
Rol:	Investigador
Descripción:	Se debe permitir cancelar una reserva realizada con anterioridad.

*Tabla 7.5: Historia 5. Cancelar Reserva. Referencia: Investigador.*

Historia:	Listar Reservas
ID:	6
Rol:	Investigador
Descripción:	Se debe permitir listar las reservas realizadas con anterioridad.

*Tabla 7.6: Historia 6. Listar Reservas. Referencia: Investigador.*

## 7.4. Casos de Uso

### Actores

Actor	Usuario sin Bot	
Descripción	Es un usuario que no tiene el bot en su telegram.	
Características	Puede agregar el bot.	
Autor	Sebastián Vélez Montoya	Versión: 1.0

*Tabla 7.7: Actor 1. Usuario sin Bot. Referencia: Ninguna.*

Actor	Usuario con Bot	
Descripción	Es un usuario que tiene en su telegram el bot registrado.	
Características	Puede reservar recursos.	
Autor	Sebastián Vélez Montoya	Versión: 1.0

*Tabla 7.8: Actor 2. Usuario con Bot. Referencia: Investigador.*

## Guiones

ID Caso de uso:	001		
Nombre C.U.:	Registro del bot al usuario		
Creado por:	Sebastián Vélez	Última Actualización Por:	Sebastián Vélez

Actores:	Usuario sin Bot	
Descripción:	El usuario podrá agregar el bot de telegram, siempre y cuando esté registrado en el documento de miembros del grupo de investigación.	
Disparador:	El usuario agrega el bot a su telegram.	
Precondiciones:	El usuario debe tener una cuenta de telegram y estar previamente registrado en el documento de miembros del grupo de investigación.	
Postcondiciones:	Después de agregado, el Usuario sin Bot se convierte en un Usuario con Bot, y tiene acceso al flujo de reserva.	
Flujo Normal:	Paso:	Condición:
	1.	El usuario sin bot agrega el bot a su telegram.
	2.*	El sistema valida su registro en el documento.
	3.	El sistema permite agregar el bot al usuario.
Flujo Alternativo:	Paso:	Condición:
	2.*	El sistema valida que el usuario no está registrado.
	3.	El sistema no permite agregar el bot.
	4.	El sistema pide al Usuario sin bot que compruebe los datos.
Excepciones:	Código:	Descripción:
	001.E.1	Los datos ingresados no son válidos.
Prioridad:	Alta.	
Frecuencia de Uso:	Alta.	

Tabla 7.9: Caso de uso 1. Registro del bot al usuario. Referencia: Usuario sin bot.

ID Caso de uso:	002		
Nombre C.U.:	Listar Recursos		
Creado por:	Sebastián Vélez	Última Actualización Por:	Sebastián Vélez

Actores:	Usuario con Bot	
Descripción:	El usuario con bot podrá listar los recursos que están registrados en el drive del grupo de investigación.	
Disparador:	El usuario realiza la petición de Listar Recursos en el bot.	
Precondiciones:	El usuario debe tener el bot en su telegram.	
Postcondiciones:	Ninguna.	
Flujo Normal:	Paso:	Condición:
	1.	El usuario realiza la petición de listar recursos al bot.
	2.	El sistema retorna los recursos que están almacenados en el drive del grupo de investigación.
Prioridad:	Alta.	
Frecuencia de Uso:	Alta.	

Tabla 7.10: Caso de uso 2. Listar Recursos. Referencia: Usuario con bot.

ID Caso de uso:	003		
Nombre C.U.::	Reservar Recurso		
Creado por:	Sebastián Vélez	Última Actualización Por:	Sebastián Vélez

Actores:	Usuario con Bot	
Descripción:	El usuario con bot podrá realizar una reserva de recurso indicando el respectivo recurso, la fecha y la duración de la reserva.	
Disparador:	El usuario realiza la petición de Reservar Recurso en el bot.	
Precondiciones:	El usuario debe tener el bot en su telegram.	
Postcondiciones:	Se guarda el historial de reserva de dispositivos en Google Drive.	
Flujo Normal:	Paso:	Condición:
	1.	El usuario realiza la petición de reserva de recurso al bot.
	2.	El sistema pide al usuario los datos necesarios para realizar la reserva.
	3.*	El sistema valida si el rango reserva está disponible.
	4.	El sistema agrega la reserva en el historial de reservas.
Flujo Alternativo:	Paso:	Condición:
	3.*	El sistema valida el rango ingresado esta siendo ocupado.
	4.	El sistema pide otro rango de reserva al usuario.
Excepciones:	Código:	Descripción:
	003.E.1	Rango de reserva no válido.
Prioridad:	Alta.	
Frecuencia de Uso:	Alta.	

Tabla 7.11: Caso de uso 3. Reservar Recurso. Referencia: Usuario con bot.

ID Caso de uso:	004		
Nombre C.U.::	Entregar Recurso		
Creado por:	Sebastián Vélez	Última Actualización Por:	Sebastián Vélez

Actores:	Usuario con Bot	
Descripción:	El usuario con bot podrá realizar una entrega de recurso indicando el id del recurso.	
Disparador:	El usuario realiza la petición de Entregar Recurso en el bot.	
Precondiciones:	El usuario debe tener el bot en su telegram y una reserva de recurso.	
Postcondiciones:	Se cambia el estado de reserva en el Drive.	
Flujo Normal:	Paso:	Condición:
	1.	El usuario realiza la petición de entrega de recurso al bot.
	2.	El sistema pide al usuario el id del recurso.
	3.	El sistema valida que el id corresponda con una reserva.
	4.	El sistema cambia el estado de la reserva.
Flujo Alternativo:	Paso:	Condición:
	3.*	El id no corresponde a una reserva.
	4.	El sistema pide otro id de recurso.
Excepciones:	Código:	Descripción:
	004.E.1	Id de recurso no corresponde a una reserva.
Prioridad:	Alta.	
Frecuencia de Uso:	Alta.	

Tabla 7.12: Caso de uso 4. Entregar Recurso. Referencia: Usuario con bot.

ID Caso de uso:	005		
Nombre C.U.::	Cancelar Reserva		
Creado por:	Sebastián Vélez	Última Actualización Por:	Sebastián Vélez

Actores:	Usuario con Bot	
Descripción:	El usuario con bot podrá cancelar una reserva de recurso indicando el id del recurso.	
Disparador:	El usuario realiza la petición de Cancelar Reserva en el bot.	
Precondiciones:	El usuario debe tener el bot en su telegram y una reserva de recurso.	
Postcondiciones:	Se cambia el estado de reserva en el Drive.	
Flujo Normal:	Paso:	Condición:
	1.	El usuario realiza la petición de cancelar reserva de recurso al bot.
	2.	El sistema pide al usuario el id del recurso.
	3.*	El sistema valida que el id corresponda con una reserva.
	4.	El sistema cambia el estado de la reserva.
Flujo Alternativo:	Paso:	Condición:
	3.*	El id no corresponde a una reserva.
	4.	El sistema pide otro id de recurso.
Excepciones:	Código:	Descripción:
	005.E.1	Id de recurso no corresponde a una reserva.
Prioridad:	Alta.	
Frecuencia de Uso:	Alta.	

Tabla 7.13: Caso de uso 5. Cancelar Reserva. Referencia: Usuario con bot.

ID Caso de uso:	006		
Nombre C.U.:	Listar Reservas		
Creado por:	Sebastián Vélez	Última Actualización Por:	Sebastián Vélez

Actores:	Usuario con Bot	
Descripción:	El usuario con bot podrá listar las reservas realizadas con anterioridad.	
Disparador:	El usuario realiza la petición de Listar Reservas en el bot.	
Precondiciones:	El usuario debe tener el bot en su telegram y una reserva de recurso.	
Postcondiciones:	Ninguna	
Flujo Normal:	Paso:	Condición:
	1.	El usuario realiza la petición de listar reservas de recurso al bot.
	2.	El sistema retorna las reservas asociadas al usuario.
Prioridad:	Alta.	
Frecuencia de Uso:	Alta.	

Tabla 7.14: Caso de uso 6. Cancelar Reserva. Referencia: Usuario con bot.

## 7.5. Modelo de vistas de arquitectura 4+1

### 7.5.1. Diagrama de casos de uso

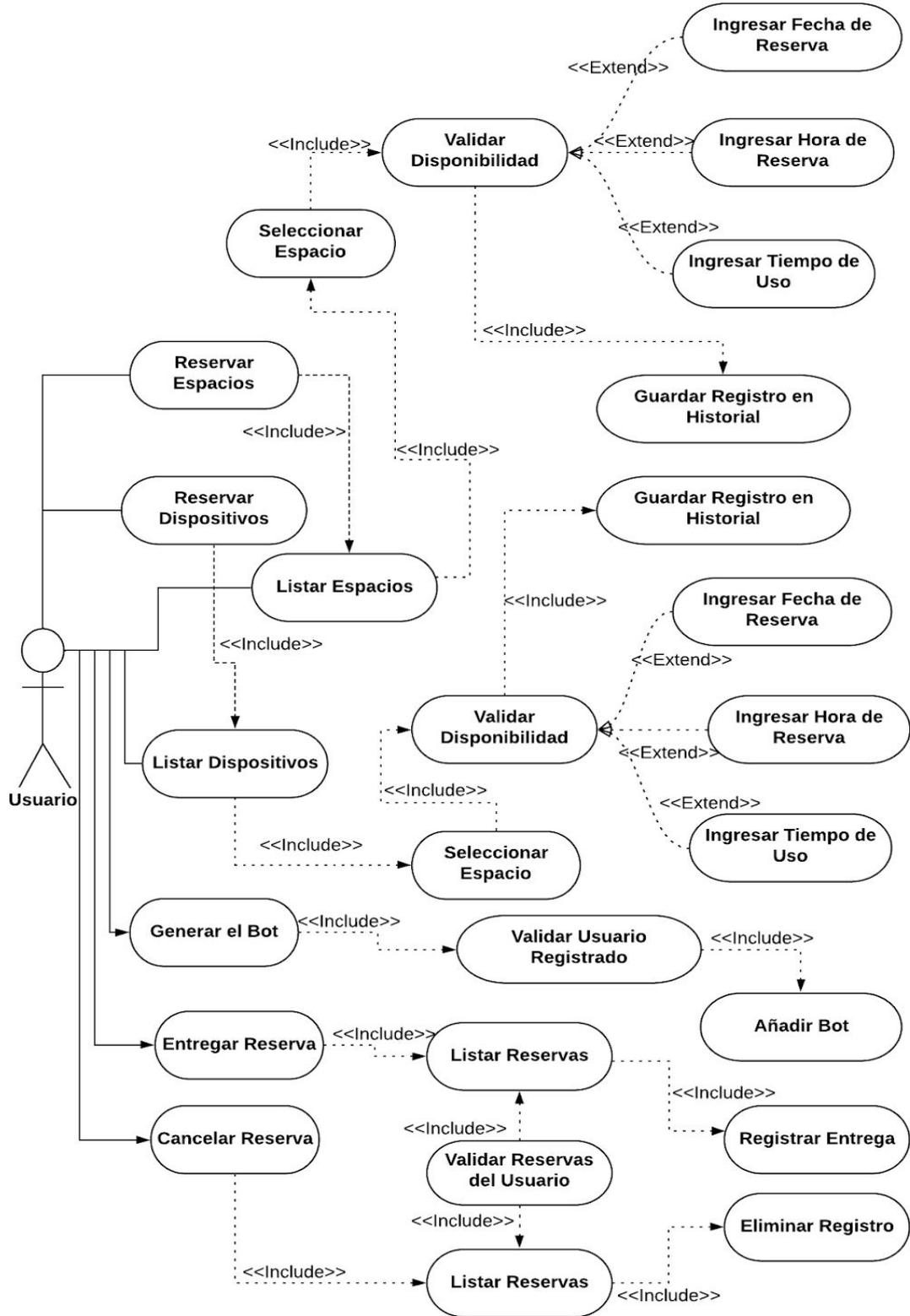


Figura 7.1: Diagrama de casos de uso. Referencia: Investigador.

## 7.5.2. Diagrama de secuencia

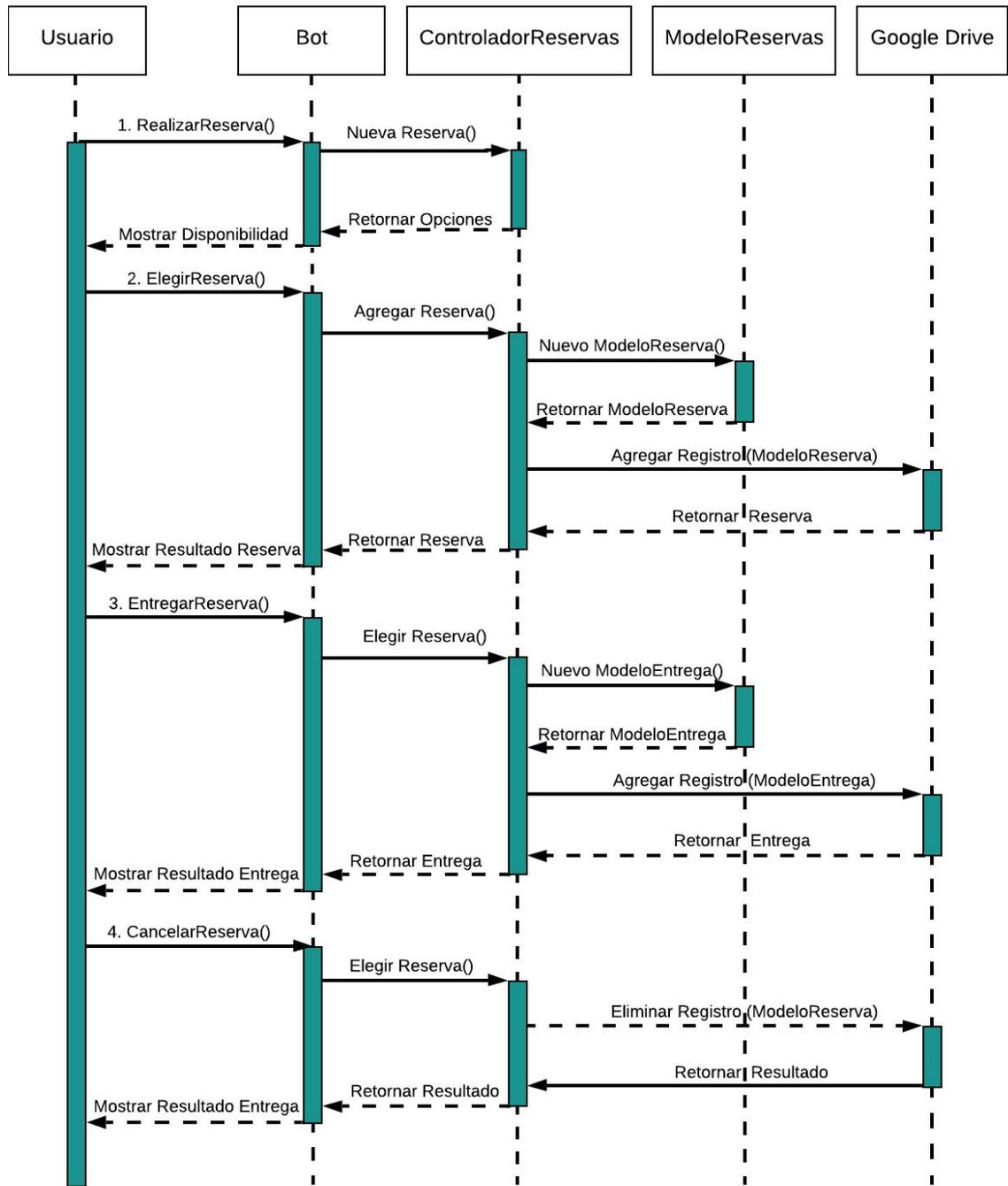


Figura 7.2: Diagrama de secuencia. Referencia: Investigador.



### 7.5.4. Diagrama de clases

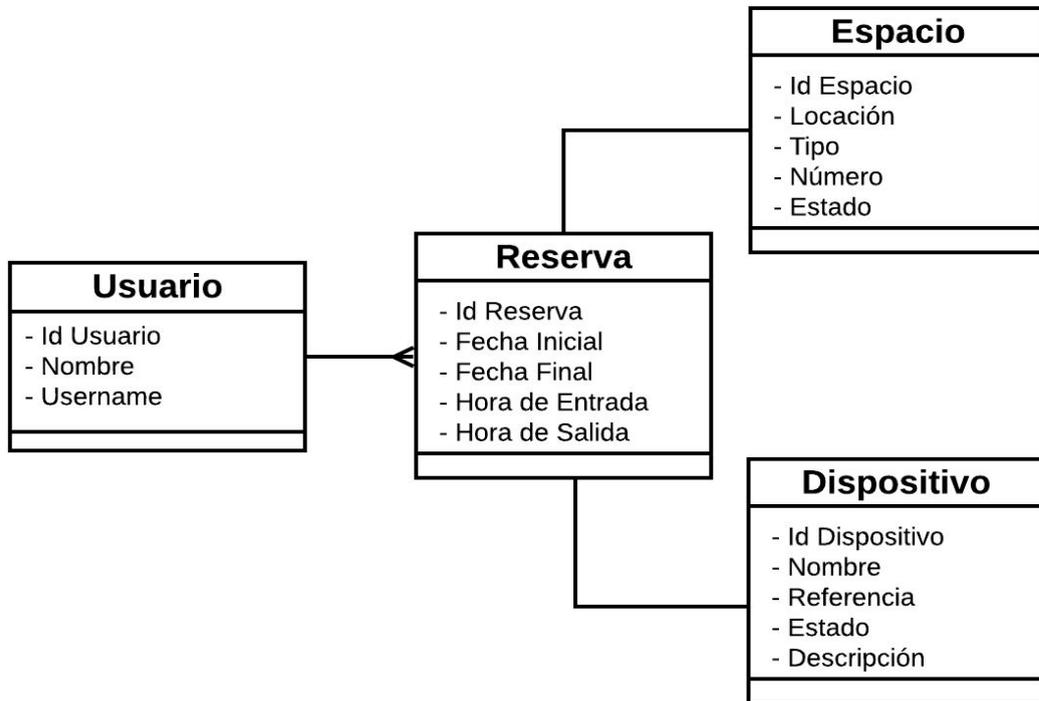


Figura 7.4: Diagrama de clases. Referencia: Investigador.

### 7.5.5. Diagrama de despliegue

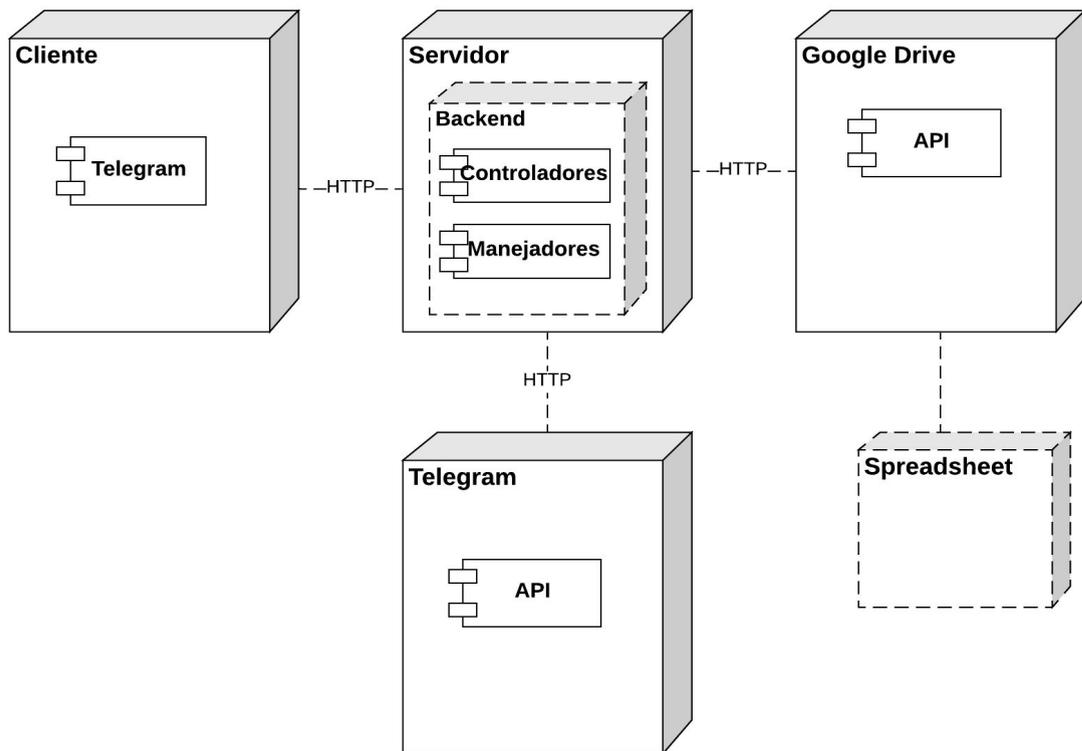


Figura 7.5: Diagrama de despliegue. Referencia: Investigador.

### 7.5.6. Diagrama de componentes

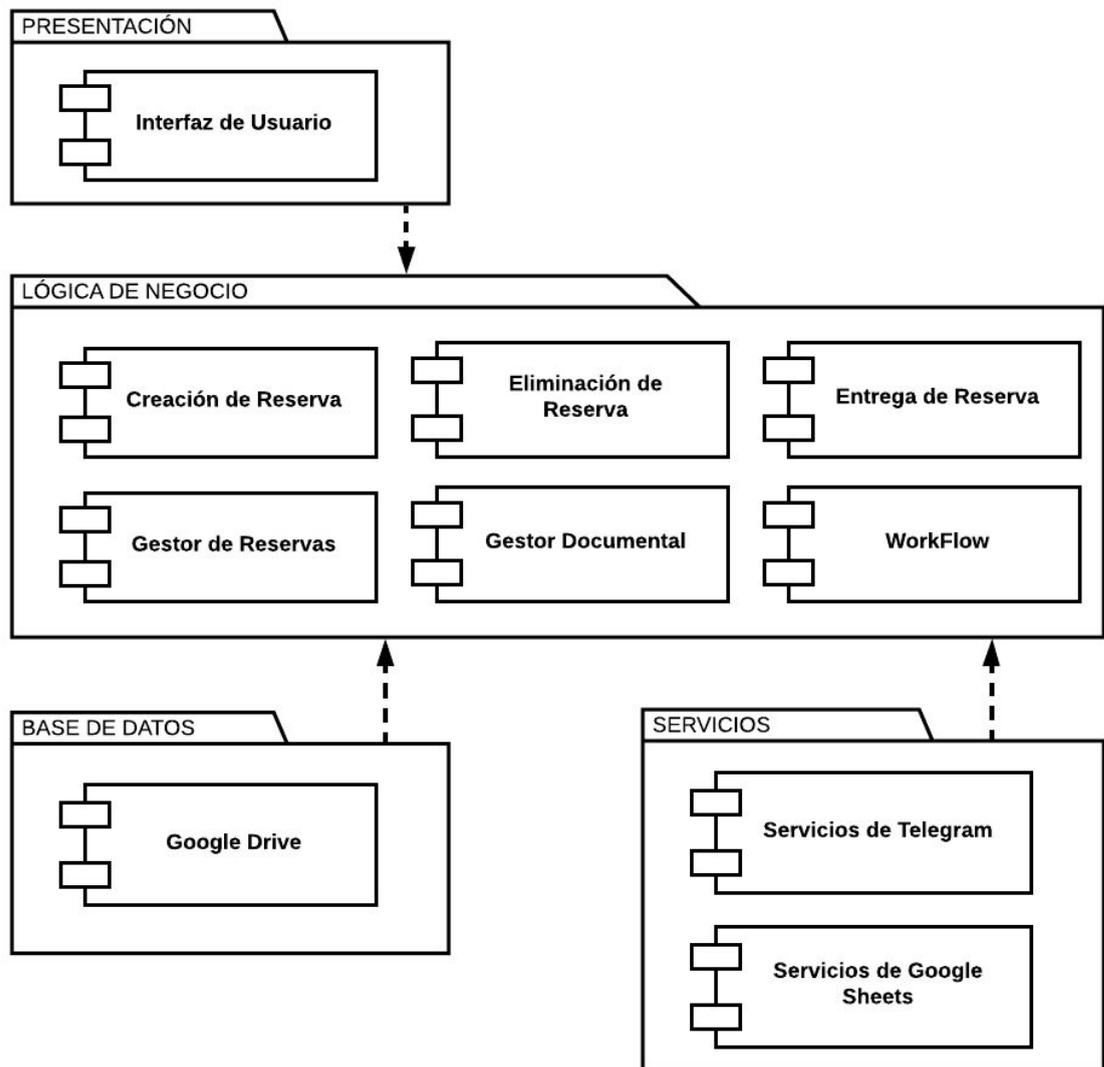


Figura 7.6: Diagrama de componentes. Referencia: Investigador.

# Capítulo 8

## Plan de Pruebas

### 8.1. Historial de Versiones

Versión	Fecha	Descripción
0.0.1	25/06/2020	Primera versión funcional sin pruebas
0.1.1	28/06/2020	Corrección incidencias del plan de pruebas
0.1.2	29/06/2020	Corrección de las variables de entorno en producción
1.0.0	30/06/2020	Primera versión funcional con pruebas

Tabla 8.1: Plan de Pruebas. Historial de versiones.

### 8.2. Resumen Ejecutivo

Se hicieron pruebas de caja negra con el fin de verificar la coherencia de las entradas y las salidas del sistema. Estas pruebas permitieron reconocer las fallas funcionales e identificar la concordancia del sistema con el diseño establecido.

Se realizaron las pruebas a cada una de las funcionalidades del sistema donde:

En la inserción de datos en google spreadsheet, se verificó el correcto almacenamiento de los datos correspondientes y los cambios de estado en el historial para: reserva, entrega y cancelar.

Para la lectura de los datos en google spreadsheet se verificaron las salidas para los comandos de listar espacios y listar dispositivos.

En los pasos de reservar se verificó que no se almacenaron registros en rangos de fechas iguales para un mismo recurso, se validó que el formato de fecha sea el adecuado y que la suma de horas con respecto a la primera fecha ingresada sea correcta.

En la operación de entregar se validó que los recursos de salida correspondan a reservas con una fecha inicial menor a la del día de la petición, además que su estado en historial se modifique correctamente y en el caso de cancelar recurso que la fecha inicial sea mayor a la fecha ingresada en la petición de la operación.

En el caso de los espacios, las pruebas se realizaron con datos proporcionados por el grupo de investigación y para los dispositivos se tomaron datos de prueba basados en el levantamiento de requerimientos. Además, se tomaron dos usuarios de prueba los cuales son los mismos que realizaron la ejecución.

Para el registro se validaron las peticiones a la api, tanto de telegram como la del servidor, y se verificó el almacenamiento correcto de los usuarios en google spreadsheet.

### **8.3. Criterios de Aceptación o Rechazo**

Completar 100% de pruebas unitarias, cobertura de todos los componentes y líneas de código, además de verificar la totalidad de defectos corregidos, entre otros.

# Capítulo 9

## Resultados del Plan de Pruebas

### 9.1. Incidencias Encontradas

Incidencia	#1
Descripción	Se permiten cruces de horarios al crear una reserva
Fecha Creación	Junio 25, 2020
Fecha Finalización	Junio 28 2020
Asignación	Edilson Londoño Castañeda
Creador	Edilson Londoño Castañeda
Tester	Sebastian Velez Montoya, Edilson Londoño Castañeda
Tipo	Funcional
Prioridad	Highest

Tabla 9.1: Incidencias Plan de Pruebas. Incidencia 1.

Incidencia	#2
Descripción	La fecha en paso 3 de reservar se actualiza con la hora del servidor
Fecha Creación	Junio 25, 2020
Fecha Finalización	Junio 28 2020
Asignación	Edilson Londoño Castañeda
Creador	Edilson Londoño Castañeda
Tester	Sebastian Velez Montoya, Edilson Londoño Castañeda
Tipo	Funcional
Prioridad	Highest

Tabla 9.2: Incidencias Plan de Pruebas. Incidencia 2.

Incidencia	#3
Descripción	Los textos no se ajustan a la experiencia de usuario y se generan salidas por consola
Fecha Creación	Junio 25, 2020
Fecha Finalización	Junio 28 2020
Asignación	Sebastian Velez Montoya
Creador	Edilson Londoño Castañeda
Tester	Sebastian Velez Montoya, Edilson Londoño Castañeda
Tipo	Visual
Prioridad	Medium

*Tabla 9.3: Incidencias Plan de Pruebas. Incidencia 3.*

Incidencia	#4
Descripción	Al no existir ningún recurso disponible, no retrocede al paso anterior
Fecha Creación	Junio 25, 2020
Fecha Finalización	Junio 28 2020
Asignación	Edilson Londoño Castañeda
Creador	Sebastian Velez Montoya
Tester	Sebastian Velez Montoya, Edilson Londoño Castañeda
Tipo	Sugerencia
Prioridad	Medium

*Tabla 9.4: Incidencias Plan de Pruebas. Incidencia 4.*

Incidencia	#5
Descripción	Cuando no existe un usuario en google spreadsheets la aplicación deja de funcionar.
Fecha Creación	Junio 28, 2020
Fecha Finalización	Junio 28 2020
Asignación	Edilson Londoño Castañeda
Creador	Sebastian Velez Montoya
Tester	Sebastian Velez Montoya, Edilson Londoño Castañeda
Tipo	Funcional
Prioridad	High

*Tabla 9.5: Incidencias Plan de Pruebas. Incidencia 5.*

# Capítulo 10

## Conclusiones

- El prototipo desarrollado puede ayudar a que las organizaciones optimicen los tiempos y tengan un control detallado sobre la administración de los recursos.
- Las herramientas que proporcionan las plataformas de mensajerías permiten implementar sistemas avanzados de administración de forma sencilla.
- Debido a la poca madurez de la librería de telegram bot api, se dificulta encontrar información en la documentación.
- Google spreadsheets proporciona herramientas de manipulación de datos que facilitaron los filtros de información, funcionalidad clave al momento de realizar el desarrollo de la api.
- Tener una estructura definida desde el inicio del proyecto, aseguró un trabajo fluido y en paralelo por parte de los desarrolladores, además de realizar un código con buenas prácticas y escalable.
- El conocimiento previo sobre los lenguajes de programación usados en el proyecto, ayudó a tener una buena experiencia como desarrollador durante el transcurso de la implementación.

# Capítulo 11

## Trabajos Futuros

En el prototipo funcional desarrollado, se pretende extender sus características para permitir el control de roles en el laboratorio, limitando el uso de recursos según los parámetros determinados por el grupo de investigación. Además con las actualizaciones de google spreadsheets, se estima una mejora en el filtrado, optimizando los tiempos de búsqueda y almacenamiento.

Una propuesta de mejora, es generar diferentes reportes de historial con respecto a un tiempo determinado, para así realizar estudios sobre análisis de datos, y a partir de esto, mejorar las características de su distribución y operación interna de la organización.

Con respecto a los tiempos de reserva se pretende agregar un calendario que permita visualizar los las reservas generadas con el fin de mejorar la experiencia de usuario y permitir un control por parte de los roles principales.

Con el fin de proporcionar ayuda con respecto al funcionamiento operativo del bot se pretende crear una página informativa la cual tenga diferentes módulos como: manual de usuario, información del bot, actualizaciones y un panel de sugerencia.

Además, debido a que el prototipo es completamente flexible y escalable a cualquier contexto que requiera administrar recursos, se podría pensar en escalar la aplicación directamente en una organización o entidad completa como la universidad.

# Capítulo 12

## Bibliografía

- [1] A Guide to the Project Management Body of Knowledge, Fourth Edition. Newtown Square, Pennsylvania: Project Management Institute (PMI). (2008). ISBN 978-1-933890-51-7.
- [2] Carrol, R., Finch, G., Hance, M., Moye, C., Perkins, M., Williams, B., and Williams, D La evolución de la automatización de procesos, IBM España, (2010)
- [3] Sniderman, Brenna; Mahto, Monika; Cotteleer, Mark J. &quot;Industry 4.0 and manufacturing ecosystems Exploring the world of connected enterprises; (PDF). Deloitte. Retrieved 25 June (2019).
- [4] Carrol, R., Finch, G., Hance, M., Moye, C., Perkins, M., Williams, B., and Williams, D La evolución de la automatización de procesos, IBM España, (2010)
- [5] La revolución de los bots se centra en la interacción con los clientes, Awerty, Microsoft Experience, (2018). <https://www.awerty.net/category/microsoft-inspire-2018>
- [6] Kanban y Scrum, obteniendo lo mejor de ambos, Henrik Kniberg; Mattias Skarin, (2010) C4Media Inc, editores de InfoQ.com.
- [7] Metodología Kanban: pros y contras en la gestión de proyectos, Daniel Mas, 27 Marzo (2019). <https://www.fhios.es/metodologia-kanban-pros-y-contras>
- [8] Kanban: The Ultimate Guide to Kanban Methodology for Agile Software Development, James Edge, CreateSpace Independent Publishing Platform, (2018).
- [9] La línea de tiempo de la historia de los Chatbots: antes, ahora y mañana, Sharon Emmanuel, Planet chatbot, (2019).
- [10] What are WebHooks and How Do They Enable a Real-time Web, Phil Leggetter Developer Advocate, Real-Time Web Software & Technology Evangelist (2020)
- [11] Standards, APIs, Interfaces and Bindings, Emery, David, (2016)
- [12] Representational State Transfer (REST) Architectural Styles and the Design of Network-based Software Architectures, Fielding, Roy Thomas (2000).
- [13] Distributed Application Architecture, Sun Microsystem. Archived from the original, ORACLE Fielding (1999).
- [14] Defining and Classifying Software Bots: A Faceted Taxonomy, Lebeuf, Carlene; Zagalsky, Alexey; Foucault, Matthieu; Storey, Margaret-Anne, (2019).

[15] Comportamiento Adaptable de Chatbots Dependiente del Contexto, Rodríguez, Juan Manuel; Merlino, Hernán; Fernández, Enrique, (2014).

[16] Speech synthesis (backend) Texas A&M University, Gutierrez Osuna, Ricardo, (2016)

[17] Una Guía de los Fundamentos de la Dirección de Proyectos, Tercera Edición. Newtown Square, Pennsylvania: Project Management Institute PMI. (2004)

# Capítulo 13

## Anexos: Manual de Usuario

### 13.1. Realizar Reserva

**Reserva de Espacio: COMANDO** → /reservar\_espacio

Paso	Formato	Ejemplo
1- Elegir Locación	[NumeroEdificion*][Bloque Edificio(?)]-[Piso*][Salón*]	15D-305
2- Elegir Tipo de Espacio	[type*]	WORKSTATION
3- Escribir Hora y fecha	[dd/mm/aa*] [hh:mm[tt]	18/06/20 11:31 am
4- Elegir o escribir el Número de Horas	[Num*]	4
5- Elegir el espacio de la lista de disponibles.	[IdEspacio*]	15D-305-WS-3

Tabla 13.1: Manual de Usuario 1. Reservar Espacio.

**Reserva de Dispositivo: COMANDO** → /reservar\_dispositivo

Paso	Formato	Ejemplo
1- Elegir Dispositivo	[Nombre_Dispositivo*]	Arduino_Uno
2- Escribir Hora y fecha	[dd/mm/aa*] [hh:mm[tt]	18/06/20 11:31 am
3- Elegir o escribir el Número de Horas	[Num*]	4
5- Elegir la Referencia de la lista de disponibles.	[IdDispositivo_Referencia*]	Arduino_Uno_1

Tabla 13.2: Manual de Usuario 2. Reservar Dispositivo.

## 13.2. Entregar Reserva

**Entrega de Espacio: COMANDO** → /entregar\_espacio

Paso	Formato	Ejemplo
1- Elegir Locación de la lista de reservas.	[Locación*]-[type*]-[id*]	15D-305-WS-1
2- Elegir la fecha y hora de la lista de rangos.	[dd/mm/aa*] [hh:mm[tt]	18/06/20 11:31 am

Tabla 13.3: Manual de Usuario 3. Entregar Espacio.

**Entrega de Espacio: COMANDO** → /entregar\_dispositivo

Paso	Formato	Ejemplo
1- Elegir Dispositivo de la lista de reservas.	[Locación*]-[type*]-[id*]	15D-305-WS-1
2- Elegir la fecha y hora de la lista de rangos.	[dd/mm/aa*] [hh:mm[tt]	18/06/20 11:31 am

Tabla 13.4: Manual de Usuario 4. Entregar Dispositivo.

## 13.3. Cancelar Reserva

**Cancelar Reserva de Espacio: COMANDO** → /cancelar\_reserva\_espacio

Paso	Formato	Ejemplo
1- Elegir Espacio de la lista de reservas.	[Locación*]-[type*]-[id*]	15D-305-WS-1
2- Elegir la fecha y hora de la lista de rangos.	[dd/mm/aa*] [hh:mm[tt]	18/06/20 11:31 am

Tabla 13.5: Manual de Usuario 5. Cancelar Reserva de Espacio.

**Cancelar Reserva de Dispositivo: COMANDO** → /cancelar\_reserva\_dispositivo

Paso	Formato	Ejemplo
1- Elegir Dispositivo de la lista de reservas.	[Locación*]-[type*]-[id*]	15D-305-WS-1
2- Elegir la fecha y hora de la lista de rangos.	[dd/mm/aa*] [hh:mm[tt]	18/06/20 11:31 am

Tabla 13.6: Manual de Usuario 6. Cancelar Reserva de Dispositivo.

## 13.4. Otros Comandos

**Obtener lista de comandos:** COMANDO → /ayuda

**Listar Reservas de Espacios:** COMANDO → /listar\_reservas\_espacios

**Listar Reservas de Dispositivos:** COMANDO → /listar\_reservas\_dispositivo

**Listar Espacios Registrados:** COMANDO → /listar\_espacios

**Listar Dispositivos Registrados:** COMANDO → /listar\_dispositivos