



Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería en Ciencias y Sistemas

**ANÁLISIS Y DISEÑO DE UNA APLICACIÓN PARA LA GESTIÓN DE
SALONES PARA LA ESCUELA DE CIENCIAS Y SISTEMAS, FACULTAD DE
INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

Luis Gilberto Barrios Vásquez
Asesorado por el Ing. William Escobar

Guatemala, mayo de 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA



FACULTAD DE INGENIERÍA

**ANÁLISIS Y DISEÑO DE UNA APLICACIÓN PARA LA GESTIÓN DE
SALONES PARA LA ESCUELA DE CIENCIAS Y SISTEMAS, FACULTAD DE
INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA**

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

LUIS GILBERTO BARRIOS VÁSQUEZ
ASESORADO POR EL ING. WILLIAM ESCOBAR

AL CONFERÍRSELE EL TÍTULO DE
INGENIERO EN CIENCIAS Y SISTEMAS

GUATEMALA, MAYO DE 2020

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
FACULTAD DE INGENIERÍA



NÓMINA DE JUNTA DIRECTIVA

DECANA	Inga. Aurelia Anabela Cordova Estrada
VOCAL I	Ing. José Francisco Gómez Rivera
VOCAL II	Ing. Mario Renato Escobedo Martínez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Christian Moisés De La Cruz Leal
VOCAL V	Br. Kevin Vladimir Armando Cruz
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANA	Inga. Aurelia Anabela Cordova Estrada
EXAMINADOR	Ing. Pedro Pablo Hernández Ramírez
EXAMINADOR	Ing. Luis Fernando Espino Barrios
EXAMINADOR	Ing. César Augusto Fernández Cáceres
SECRETARIO	Ing. Hugo Humberto Rivera Pérez

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

ANÁLISIS Y DISEÑO DE UNA APLICACIÓN PARA LA GESTIÓN DE SALONES PARA LA ESCUELA DE CIENCIAS Y SISTEMAS, FACULTAD DE INGENIERÍA, UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería en Ciencias y Sistemas con fecha 11 de febrero de 2019.

Luis Gilberto Barrios Vásquez

ACTO QUE DEDICO A:

- Dios** Por proveerme de vida, salud y sabiduría para alcanzar esta meta.
- Mis padres** Leslie de Barrios y Luis Ernesto Barrios, por su incondicional amor, apoyo y ejemplo de perseverancia en la vida.
- Mi hermana** María José Barrios Vásquez, por el apoyo y las palabras de ánimo que me brindó en todo momento.
- Mis amigos** Brayán Flores, César Morales, Luis Azurdía y Fabio de Paz, por las experiencias compartidas y el apoyo en todo momento.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por recibirme con las puertas abiertas en mi búsqueda de una educación superior.
Facultad de Ingeniería	Por formarme como un profesional enfocado en la innovación, invención y resolución de problemas.
Escuela de Ingeniería en Ciencias y Sistemas	Por proveerme de conocimientos especializados ayudándome así a mi formación profesional.
Ing. William Escobar	Por su valioso apoyo y asesoramiento.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES.....	III
GLOSARIO	VII
RESUMEN.....	IX
OBJETIVOS.....	XI
INTRODUCCIÓN	XIII
1. IDENTIFICACIÓN DEL PROBLEMA.....	1
1.1. Antecedentes.....	1
1.2. Justificación	2
1.3. Mercado objetivo	3
1.4. Benchmark de la aplicación.....	3
2. MARCO TEÓRICO.....	5
2.1. Teoría TTF.....	5
2.2. Arquitectura cliente-servidor	6
2.3. HTML.....	7
2.4. JavaScript.....	8
2.5. Bases de datos NoSQL	10
2.6. AngularJS	12
3. DISEÑO DE LA APLICACIÓN.....	15
3.1. Prototipos	15
3.2. Caso de uso	30
3.3. Diagramas de actividades	31

4.	DOCUMENTACIÓN DE LA APLICACIÓN	39
4.1.	Requisitos	39
4.2.	Herramientas de desarrollo	40
4.2.1.	Hardware.....	40
4.2.2.	Software	40
4.3.	Consideraciones de la implementación.....	41
5.	ANÁLISIS E INTERPRETACIÓN DE RESULTADOS.....	43
	CONCLUSIONES.....	49
	RECOMENDACIONES	51
	BIBLIOGRAFÍA.....	53

ÍNDICE DE ILUSTRACIONES

FIGURAS

1.	Prototipo de la página inicial	15
2.	Prototipo de panel de actividades del usuario	16
3.	Prototipo de ingreso a la aplicación.....	17
4.	Prototipo para solicitud de salón 1	18
5.	Prototipo para solicitud de salón 2	19
6.	Prototipo para el panel administrativo 1	20
7.	Prototipo para panel administrativo 2	20
8.	Prototipo para administrador de usuarios.....	21
9.	Prototipo para agregar usuario 1	21
10.	Prototipo para agregar usuario 2.....	22
11.	Prototipo para modificar usuario	22
12.	Prototipo para administrador de campus.....	23
13.	Prototipo para administrador de edificios	23
14.	Prototipo para administrador de salones.....	24
15.	Prototipo para agregar edificio	24
16.	Prototipo para agregar campus.....	25
17.	Prototipo para agregar salón.....	25
18.	Prototipo para validar solicitud	26
19.	Prototipo para validar solicitud con salón bloqueado	26
20.	Prototipo para validar solicitud con horarios agregados.....	27
21.	Prototipo para historial de solicitudes.....	27
22.	Prototipo para la consulta de horarios.....	28
23.	Prototipo de historial de horarios con consulta realizada	28

24.	Prototipo de panel de configuración 1....	29
25.	Prototipo de panel de configuración 2.....	29
26.	Caso de uso general de la aplicación	30
27.	Procedimiento para agregar ubicación operador	31
28.	Procedimiento para apartar horario operador	32
29.	Procedimiento para la autenticación de sesiones.....	33
30.	Procedimiento para consultar horarios	34
31.	Procedimiento para eliminar horario operador.....	35
32.	Procedimiento para enviar solicitud de salón.....	36
33.	Procedimiento para establecer fin de semestre	37
34.	Opinión de los auxiliares sobre el proceso de solicitud actual	43
35.	Facilidad de uso del nuevo sistema	44
36.	Opinión de los auxiliares sobre la accesibilidad de la información de los salones	45
37.	Información del sistema sobre las actividades en los salones.....	46
38.	Uso de la aplicación web para realizar la solicitud de un salón	47

TABLAS

I.	Mejora del proceso de solicitud de salones	44
II.	Facilidad de uso del nuevo sistema	45
III.	Información del sistema sobre las actividades en los salones.....	46
IV.	Información del sistema sobre las actividades en los salones.....	47
V.	Uso de la aplicación para apartar salones.....	48

GLOSARIO

<i>Megabyte</i>	Unidad de información que equivale a un millón de bytes.
<i>Gigabyte</i>	Unidad de información que equivale a mil millones de bytes.
<i>Benchmark</i>	Prueba de rendimiento o comparativa.

RESUMEN

En la formación de estudiantes de la Escuela de Ingeniería en Ciencias y Sistemas, se imparten diversos laboratorios de cada curso. Estos laboratorios son realizados en sus propios salones y horarios, independientes del curso. Los laboratorios son impartidos por auxiliares, quienes previamente realizan la gestión para obtener un salón un horario específico. Esta gestión de horarios tiene la desventaja de ser un procedimiento manual. Cada auxiliar debe presentarse presencialmente para solicitar un salón; este será agregado a un listado en una hoja de cálculo; luego, se verificará que no haya traslape con otros horarios antes de proveerle uno.

Por esta razón, se realiza el análisis y diseño de una aplicación que facilite la gestión y organización del uso de salones. De acuerdo a lo investigado, esta aplicación se planteó para ser de uso en un ordenador de escritorio y en un Smartphone, lo que llevó a la conclusión de diseñarla como una aplicación web.

Luego del desarrollo de la aplicación y la documentación de los diferentes procesos que ocurren en esta, se realizó una encuesta a cincuenta auxiliares de la Escuela de Ingeniería en Ciencias y Sistemas sobre si según su opinión, el procedimiento actual de solicitud de salones necesitaba correcciones y si la aplicación en cuestión llenaba ese vacío funcional. Los resultados de la encuesta confirmaron esto; se estableció que el procedimiento actual no era el más conveniente por la necesidad de presentarse físicamente para solicitar salones; además de no ser capaz de consultar las actividades en cualquier momento, cuestiones que la aplicación provee de primera mano.

OBJETIVOS

General

Implementar una aplicación que permita la gestión de los salones para la Escuela de Ingeniería en Ciencias y Sistemas de la Universidad de San Carlos de Guatemala.

Específicos

1. Implementar una herramienta mediante la cual se pueda gestionar la solicitud de salones de la Escuela de Ingeniería en Ciencias y Sistemas.
2. Facilitar la asignación de salones en periodos de tiempo establecidos.
3. Llevar un control histórico de las asignaciones realizadas a los diferentes salones de la Escuela de Ingeniería en Ciencias y Sistemas.

INTRODUCCIÓN

La Universidad de San Carlos de Guatemala es la única institución estatal en el país a cargo de impartir educación superior; con más de 300 años en esta labor ha cultivado y promulgado múltiples carreras universitarias y cursos educativos que apoyan este objetivo. Una de las facultades que pertenecen a la universidad es la de Ingeniería, en la cual funciona la Escuela de Ingeniería en Ciencias y Sistemas.

La Escuela de Ingeniería en Ciencias y Sistemas tiene asignada una cantidad de salones en específico, dentro de los cuales se llevan actividades exclusivas de la escuela. Estas actividades dependen de los laboratorios, por lo que usualmente se imparten clases acerca de conocimientos prácticos y conferencias que refuerzan lo aprendido.

Actualmente, para realizar la gestión y distribución de salones para tutores se debe realizar un proceso de selección para la asignación del salón, puesto que existen cursos que tienden a exceder la capacidad máxima de alumnos de los salones exclusivos de la carrera; es, entonces, cuando se debe enviar la solicitud del salón a Asuntos Estudiantiles, factor a tomar en cuenta a la hora de realizar la asignación del salón.

También, se verifica el equipo disponible en el salón, dado que algunos cursos requieren de equipo especial, como proyectores. Todo esto ha adquirido un mayor grado de complejidad recientemente puesto que ahora la cantidad de salones destinados a la Escuela de Ingeniería Ciencias y Sistemas es mayor, y la distribución de salones de forma manual es un proceso deficiente y tedioso. Es

por esta razón que la gestión de estos espacios es de vital importancia para el correcto funcionamiento de la Escuela de Ingeniería en Ciencias y Sistemas.

1. IDENTIFICACIÓN DEL PROBLEMA

1.1. Antecedentes

Con la accesibilidad que el internet ha proporcionado en los últimos años, más el uso popular de sitios web para soluciones de software, ha hecho que la utilidad de un servicio dependa en gran manera de si este posee un sitio web. La forma como los consumidores de servicios se relacionan con esto ha cambiado mucho; de acuerdo con un estudio de GJK y Greven de 2015, para el 90 % de los usuarios es vital que la información sea posible encontrarse *online*, de igual forma los servicios.

La Facultad de Ingeniería posee un sistema para el control de alumnos, tutores y catedráticos de la facultad. En dicho sistema se puede ver si las notas finales de los cursos ya han sido aprobadas, cantidad de créditos, realizar la publicación de notas finales, generar boleta de pago para la inscripción y realizar la asignación de cursos.

Actualmente, en la Escuela de Ingeniería en Ciencias y Sistemas de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala existe un sistema informático que permite la gestión y el seguimiento de las actividades de los tutores de cátedra; provee de funciones como el registro de horarios de entrada y salida mediante el marcaje con huella digital, comunicación con los alumnos y control de sus notas, calendarización de actividades, entre otras.

1.2. Justificación

Actualmente, la gestión de solicitud de salones para la Escuela de Ingeniería en Ciencias y Sistemas no es óptima. Requiere de forma inicial la solicitud del salón, por lo que el tutor debe anticipar el uso del salón para acudir a reservar el mismo. Si solicitara un salón que excede la capacidad de personas de los salones disponibles se extiende la solicitud a Asuntos Estudiantiles, en cualquier caso, debe presentarse físicamente para iniciar el procedimiento. A continuación, se debe apartar el salón en el horario solicitado si cumple los requisitos para ser otorgado (cantidad de alumnos que lo van a utilizar, equipo, disponibilidad, entre otros). Esto implica la tarea de identificar un salón, con la capacidad adecuada y el equipo adecuado en un período de tiempo que no fuera ocupado previamente.

La tarea se vuelve deficiente ya que este control se realiza manualmente. Una a una las solicitudes se deben validar si cumplen con los requerimientos; además, dichas solicitudes se registran presencialmente. Tampoco se tiene un registro histórico de quienes utilizan los salones o su equipo. No existe una forma para consultar que salones están activos actualmente o suspendidos de uso por mantenimiento.

Por esta razón, es necesario construir una herramienta web que facilite la solicitud de salones sin necesidad de un acercamiento presencial; además, permitir la consulta de las actividades que se desarrollan en los salones y la organización de estos a los tutores que hayan hecho solicitudes; que permite así la automatización del proceso, y hace más fácil el proceso de solicitud de salones para la Escuela de Ingeniería en Ciencias y Sistemas.

1.3. Mercado objetivo

Para asegurar el correcto enfoque de la aplicación es necesario delimitar quienes utilizarán la aplicación, a quienes les beneficia y quiénes la necesitan. Para esto se utilizarán los siguientes criterios mediante los cuales se definirá que grupo de personas es de potencial interés para esta investigación.

- Edad: dado que se espera que lo utilicen estudiantes y conferencistas, se dirigirá la aplicación a personas de 18 años en adelante.
- Sexo: no existe restricción de acuerdo con si la utiliza un hombre o una mujer.
- Geografía: en cualquier parte de Guatemala donde haya unidades educativas de la Escuela de Ingeniería en Ciencias y Sistemas de la Universidad de San Carlos de Guatemala.
- Conocimientos: no se requieren conocimientos especializados previamente para el manejo de la aplicación.
- Rol: está dirigido a estudiantes de la Escuela de Ingeniería en Ciencias y Sistemas y a los administradores de los salones de la misma escuela.

1.4. *Benchmark* de la aplicación

Para determinar las características con las que debiera contar la aplicación, o los niveles de servicio que debiera prestar, es importante realizar una comparación con otros sistemas similares o dentro del mismo contexto; lo que

permitirá definir el punto de partida para las características de la aplicación. Para este análisis se tomarán en cuenta 2 sitios web:

- Portal de Ingeniería USAC (<https://portal.ingenieria.usac.edu.gt>).
- Sitio web del Desarrollo de Transferencia Tecnológica (<https://dttecys.org>).

Estos sitios proporcionan servicios al mismo grupo objetivo de esta investigación; por lo tanto, se considera adecuado su análisis.

Las características que poseen estas aplicaciones son:

- Aplicación web.
- Diseño adaptable a distintas resoluciones de renderizado (*responsive*).
- Función de autenticación de usuarios.
- Disponibilidad media a alta de la aplicación.
- Diseño intuitivo y poca capacitación para su uso.
- Provee almacenamiento de la información independiente del cliente que consume la aplicación.
- Recolección de información personal de los usuarios.

2. MARCO TEÓRICO

2.1. Teoría TTF

La teoría del ajuste de la tecnología a las tareas realizadas (*task-technology fit* en inglés) propone que una tecnología será utilizada y se desempeñará de mejor manera en la medida que esta sea capaz de resolver las tareas que realiza. Para que una tecnología sea utilizada debe estar diseñada de forma tal que facilite las tareas que automatiza a comparación de métodos tradicionales. Para medir que tan bien una tecnología realiza las tareas para la que fue diseñada se tienen varios criterios. Uno de estos es la evaluación de dicha tecnología en 8 criterios (propuestos por Goodhue and Thompson en 1995). Estos criterios son:

- Calidad: la percepción de valor en el sistema.
- Localización: si es sencillo ubicar al sistema y las funciones que provee.
- Autorización: las restricciones que existen sobre su uso.
- Compatibilidad: que tan fácilmente se integra con los procesos relacionados.
- Facilidad de uso: que tan sencillo es aprender a utilizar el software.
- Tiempo de producción: si produce resultados en el tiempo estipulado o adecuado.

- Confiabilidad: el sistema es capaz de realizar los procesos repetidamente con resultados consistentes.
- Relación con los usuarios: que tanto lo utiliza el usuario objetivo.

2.2. Arquitectura cliente-servidor

La arquitectura cliente servidor es un modelo de computación a través del cual las interacciones existen entre dos componentes principales del sistema: un servidor que distribuye el contenido y gestiona respuestas a determinadas solicitudes y un cliente que consume los contenidos del servidor y con base en esos contenidos puede realizar distintas solicitudes a este. La distribución del contenido necesita un medio que usualmente es algún tipo de red, como el internet.

El procedimiento llevado a cabo es sencillo; la comunicación inicia con una solicitud de un cliente hacia un servidor; el servidor valida la solicitud mediante la revisión si se tiene autorización de respuesta, o que respuesta se desea. Por último, se envía la respuesta al cliente quien utilizará dicha respuesta para algún otro procedimiento o se encargará de desplegar dicha información a un usuario.

Las interacciones entre el cliente y servidor pueden no tener restricciones, ya que un cliente puede conectarse a múltiples servidores en un momento dado y un servidor puede proporcionar múltiples y distintos contenidos a clientes que lo soliciten. El ejemplo más popular de esta arquitectura es apreciable en la entrega de contenido web. Cada vez que se realiza una conexión a un sitio web, el navegador realiza una solicitud al servidor web, quien dependiendo de la ruta a la que se haya accedido entregará el contenido que corresponde, una página web, para su renderizado en el navegador.

2.3. HTML

Por sus siglas en inglés, HTML significa *Hyper Text Markup Language*. Es un lenguaje de marcas de hipertexto que facilita la distribución del contenido web. Su funcionamiento depende de las marcas o *tags* colocados en un documento. Los *tags* se encuentran encerrados entre corchetes angulares: < y > con un texto que los identifica dentro de los dos corchetes. Algunos de los *tags* más importantes son los siguientes:

- Head: es la etiqueta que indica el inicio de un documento HTML.
- Title: especifica cuál es el título del documento.
- Body: contiene el contenido principal de un documento HTML. Esta etiqueta existe una sola vez.
- Header: contiene elementos del encabezado del documento.
- P: especifica un párrafo en el documento.
- Div: permite tener agrupaciones de más *tags* dentro de este.

Los *tags* indican el lugar y la forma en la que debe ser renderizado el contenido. Usualmente, HTML utiliza otros lenguajes de soporte para añadir funcionalidad al contenido que distribuye, algunos de estos lenguajes son: CSS, JavaScript y PHP. Cada una de las marcas en un documento HTML es interpretada por un navegador web, quien se encarga de renderizar el documento.

2.4. JavaScript

JavaScript es un lenguaje de programación mayormente asociado a la web, ya que provee de capacidades interactivas a los sitios web. Se caracteriza por facilitar la programación orientada a objetos, creación de prototipos y tener tipado débil; esto significa que la declaración de tipos de datos en las variables no es común. El lenguaje está estandarizado mediante ECMAScript, que es una especificación de cómo debe ser escrito el lenguaje. De los navegadores contemporáneos solo Internet Explorer no sigue el estándar de ECMAScript, este utiliza un intérprete basado en JScript, la versión de JavaScript de Microsoft.

Javascript es un lenguaje interpretado, por lo que el código sigue siendo legible en su estado original luego de haber sido ejecutado, ya que existe un motor intérprete encargado de leer cada declaración, validar si su sintaxis es correcta y ejecutar la acción que indica. Aunque JavaScript y Java parezcan ser el mismo lenguaje, no es así. Oracle es dueño de ambos lenguajes, pero tienen capacidades y sintaxis distintas uno del otro.

Al igual que muchos otros lenguajes, JavaScript posee la capacidad de cambiar el flujo de ejecución de sus sentencias mediante estructuras de control. Cada estructura de control posee un ámbito. Antes de entrar en detalle sobre las estructuras de control cabe aclarar que, a comparación de otros lenguajes, una variable al ser declarada tiene como ámbito toda la función en la que fue declarada a menos que se utilice la palabra reservada *let*, esto es diferente de la mayoría de lenguajes de programación, donde las variables solo son utilizables en el ámbito en el que fueron creadas. Entre las estructuras de control del flujo de ejecución del código se encuentran:

- Switch case: es un conjunto de casos, en cada cual se ejecutará el código de alguno de estos, si se cumple con la expresión condicional de este.
- If else: es la estructura de control que permite la ejecución de código basada en condiciones.
- While: ejecuta un conjunto de sentencias en un ciclo hasta que la condición que rige el ciclo sea falsa.
- Do while: similar al ciclo while, except que siempre se ejecutan una vez las sentencias en este, luego se ejecutarán si la condición que lo rige se cumple.
- For: es un ciclo que ejecuta una cierta cantidad de veces, establecidas en la estructura, el conjunto de sentencias del que se compone.
- For in: ciclo que itera en las propiedades de un objeto.
- Continue: omite la iteración actual y da inicio a la siguiente.
- Break: termina cualquier ciclo.
- Return: termina la ejecución de una función o procedimiento, además permite la devolución de valores en una función.

A pesar de que JavaScript es un lenguaje de tipado débil, maneja varios tipos tanto primitivos como no primitivos:

Tipos de dato primitivos:

- String: representa un conjunto de caracteres.
- Numbers: cualquier número decimal o entero, negativo o positivo.
- Booleans: valor booleano, que puede ser verdadero o falso.
- Undefined: este valor lo conservan las variables cuando no se les ha asignado valor alguno en su declaración o posterior asignación.
- Null: es un tipo de valor que representa el valor vacío.

Tipos de dato no primitivos:

- Array: es un conjunto de valores del mismo tipo.
- Objects: es un conjunto de pares de valor-propiedad, siendo la propiedad la que identifica al valor. Estos valores pueden ser primitivos y no primitivos.

2.5. Bases de datos NoSQL

Las bases de datos NoSQL funcionan de forma diferente a las bases de datos tradicionales. Esto debido a que las necesidades de los negocios han cambiado, ahora es necesario que las aplicaciones sean capaces de servir a grandes cantidades de usuarios, facilitar el crecimiento horizontal y en la nube y almacenar mucha información que debe ser accedida velozmente. Para resolver estas necesidades existen diferentes tipos de soluciones que encajan en la definición de base de datos NoSQL.

Las bases de datos se caracterizan por ser capaces de escalar considerablemente sin comprometer el rendimiento. También, son capaces de manejar grandes volúmenes de datos estructurados o no estructurados. Promueven una estructura de escalado horizontal, contrario a una estructura monolítica que es tradicional en una base de datos relacional.

Entre los tipos de base de datos NoSQL se tiene:

- Bases de datos documentales: estas bases de datos relacionan una clave con un documento. Cuando habla de documento se refiere a una estructura de datos que almacena conjuntos de información, pudiendo ser clave valor, clave matriz, JSON o XML entre otros.
- Almacenes de grafos: basan el almacenamiento en redes de datos.
- Almacenes clave-valor: este es el tipo de base de datos NoSQL más sencillo, estos almacenan un valor que puede ser obtenido a partir de una clave o atributo definido.
- Bases de datos columnares: a diferencia de las bases de datos tradicionales, donde se utilizan filas para almacenar cada registro; en las bases de datos columnares se almacenan grupos de campos como columnas; esto permite que las consultas puedan abarcar grandes conjuntos de datos con un mejor desempeño.

2.6. AngularJS

AngularJS es un *framework* del tipo Modelo Vista Controlador (MVC) para JavaScript. Es común en la parte del desarrollo del *front end* de las aplicaciones web y se utiliza especialmente para construir Single Page Apps (SPA). Una SPA es un sitio web donde se cargan todos los contenidos que este ofrece sin necesidad de refrescar el navegador.

La forma de agregar funcionalidad de AngularJS es a través de extender el lenguaje HTML. Esto lo hace agregando atributos a los tags, o tags completamente nuevos que permitan servir contenido dinámico que tengan un enlace bidireccional. A lo que se refiere el enlace bidireccional, es a que los valores mostrados en la interfaz (vista) puedan modificarse y esto se sincronice instantáneamente con la lógica del código (controlador) y viceversa.

La forma de extender el lenguaje HTML se realiza a través de directivas. Las directivas de AngularJS inician con el prefijo ng-. Algunas de las directivas utilizadas son las siguientes:

- Directiva ng-app: es la encargada de indicar dónde empieza una aplicación de AngularJS, por lo tanto, define el elemento raíz. Contendida en esta directiva se pueden inicializar otros módulos o directivas de AngularJS.
- Directiva ng-init: la directive ng-init inicializa la aplicación de AngularJS y la data asociada a esta. Al momento de ser leída inicializa valores a las variables.
- Directiva ng-model: esta directiva define una variable asociada a un elemento del HTML.

- Directiva ng-repeat: la directiva ng-repeat itera sobre los elementos de una lista, repitiendo el HTML al cual fue asociado.

3. DISEÑO DE LA APLICACIÓN

3.1. Prototipos

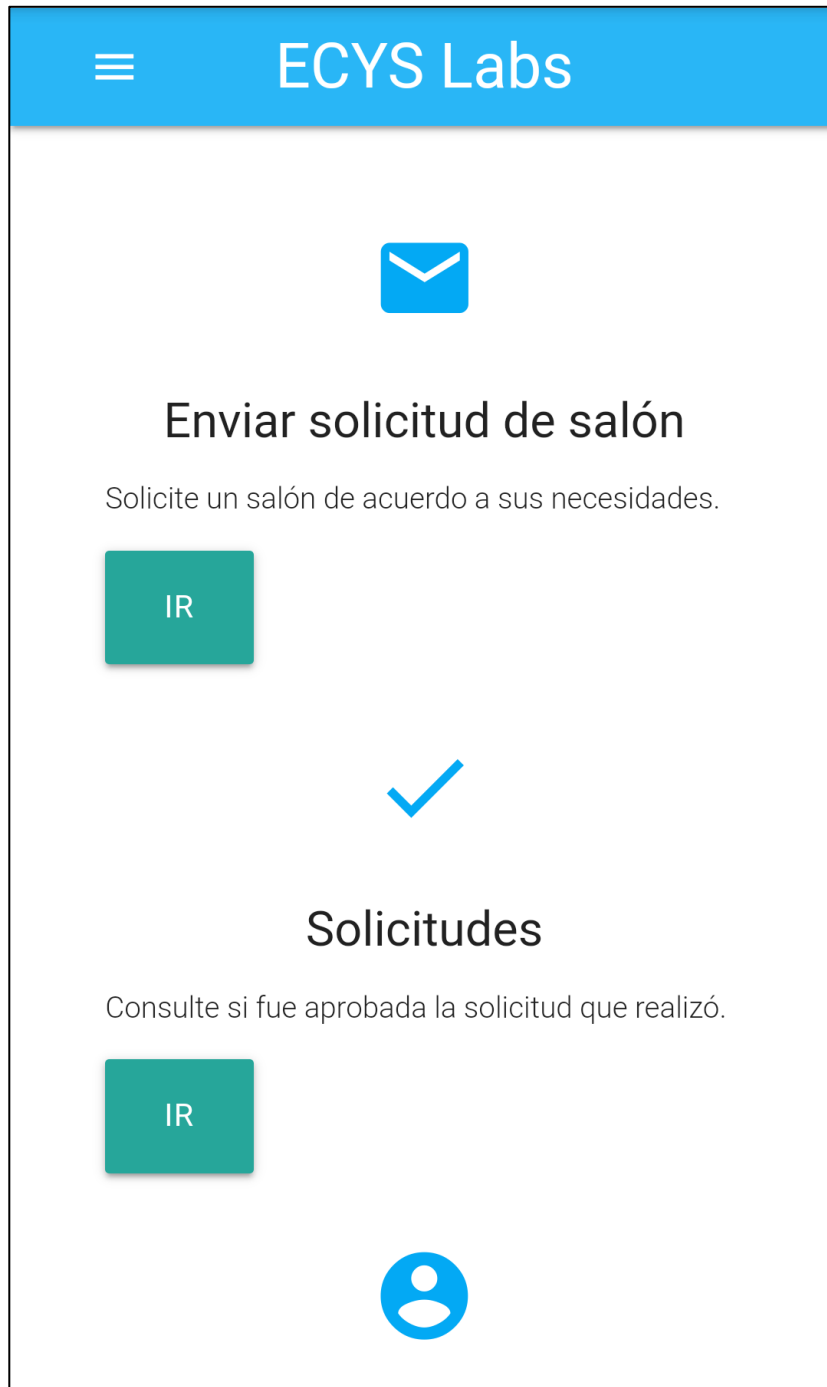
En esta sección se muestran los prototipos de la aplicación, los cuales son una representación visual de forma tentativa acerca de cómo se distribuyen en diferentes pantallas las funcionalidades planteadas.

Figura 1. Prototipo de la página inicial



Fuente: elaboración propia.

Figura 2. **Prototipo de panel de actividades del usuario**



Fuente: elaboración propia.

Figura 3. Prototipo de ingreso a la aplicación



The image shows a mobile application login screen. At the top, there is a blue header bar with a white hamburger menu icon on the left and the text "ECYS Labs" in white. Below the header, the main content area is white and contains a light gray rounded rectangle. Inside this rectangle, the text "Ingresa a su cuenta" is displayed in a large, bold, orange font. Below this text are two input fields: the first is labeled "Correo electrónico" and the second is labeled "Contraseña", both in a light gray font. At the bottom of the rounded rectangle, there are two orange buttons with white text: "INGRESAR" on the left and "CREAR CUENTA" on the right. The entire screen is framed by a thin black border, and there is a solid orange bar at the very bottom.

Fuente: elaboración propia.

Figura 4. **Prototipo para solicitud de salón 1**

☰ ECYS Labs

Solicitud de salón

Seleccione una ubicación:

CAMPUS CENTRAL ▼

Motivo por el cual desea utilizar el salón:

- Conferencia (duración 1 día)
- Laboratorio (duración durante el semestre)
- Taller (duración 1 día)
- Otro (duración 1 día)

Fecha del evento o inicio de actividades:

Wednesday:03/April/2019

Necesita proyector

Nombre del curso

Fuente: elaboración propia.

Figura 5. **Prototipo para solicitud de salón 2**

Wednesday:03/April/2019

Necesita proyector


Nombre del curso

Nombre del catedrático

Sección

Hora inicio: 10:30 ▼ Hora inicio: 10:30 ▼

Cantidad de alumnos

 ENVIAR SOLICITUD

Fuente: elaboración propia.

Figura 6. **Prototipo para el panel administrativo 1**



Fuente: elaboración propia.

Figura 7. **Prototipo para panel administrativo 2**



Fuente: elaboración propia.

Figura 8. **Prototipo para administrador de usuarios**

Nombre	Apellido	No. Registro Estudiantil	DPI	Correo electrónico	Tipo	Eliminar	Modificar
test1	test1	33333	5555555555	test1@gmail.com	estudiante		
Luis	Barrios	201403794	2872150080101	luisgil95crash@gmail.com	administrador		
test2	test2	123456	123456789	test2@gmail.com	estudiante		

Fuente: elaboración propia.

Figura 9. **Prototipo para agregar usuario 1**

Agregar usuario

*Nombres

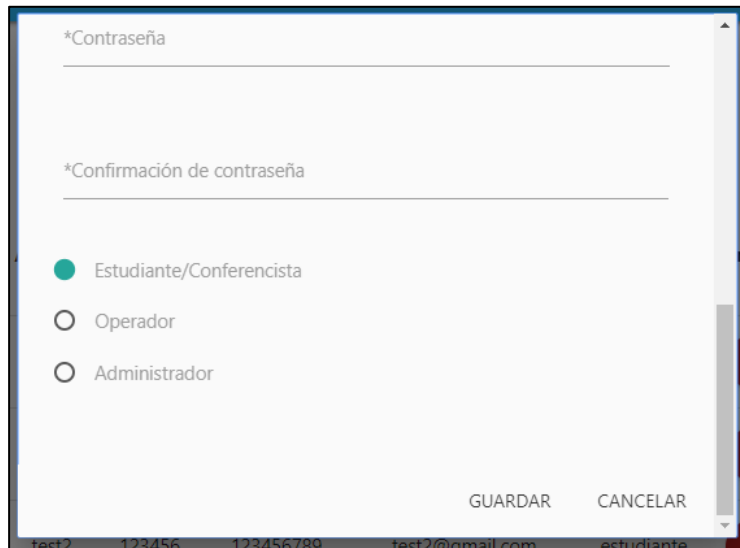
*Apellidos

Número de Registro Estudiantil

*Número de DPI

Fuente: elaboración propia.

Figura 10. **Prototipo para agregar usuario 2**



Prototipo de formulario para agregar usuario. El formulario contiene los siguientes campos y elementos:

- *Contraseña
- *Confirmación de contraseña
- Radio buttons para seleccionar el rol:
 - Estudiante/Conferencista
 - Operador
 - Administrador
- Botones: GUARDAR y CANCELAR

En la parte inferior del formulario se muestran los datos de prueba: test2, 123456, 123456789, test2@gmail.com, estudiante.

Fuente: elaboración propia.

Figura 11. **Prototipo para modificar usuario**



Prototipo de formulario para modificar usuario. El formulario tiene el título "Modificar usuario" en naranja y los siguientes campos:

- *Nombres: Luis
- *Apellidos: Barrios
- Número de Registro Estudiantil: 201403794
- *Número de DPI: 2872150080101

Fuente: elaboración propia.

Figura 12. Prototipo para administrador de campus



Fuente: elaboración propia.

Figura 13. Prototipo para administrador de edificios



Fuente: elaboración propia.

Figura 14. **Prototipo para administrador de salones**



Prototipo de administrador de salones. La interfaz muestra una barra de navegación con 'Campus', 'Edificio' y 'Salón'. El título principal es 'Administrador de ubicaciones'. Hay dos menús desplegables: 'Campus' (seleccionado 'Campus Central') y 'Edificio' (seleccionado 'Edificio T3').

Código	Nombre	Campus	Edificio	Capacidad	Proyector	Eliminar
014	Centro Tecnológico No.1	CC	T3	30	true	

Fuente: elaboración propia.

Figura 15. **Prototipo para agregar edificio**



Prototipo para agregar edificio. El título es 'Agregar edificio'. Se solicita seleccionar una ubicación en un menú desplegable con el texto 'Escoja un Campus'. Hay dos campos de texto: 'Nombre' y 'Código o alias'. En la parte inferior hay botones 'GUARDAR' y 'CANCELAR'.

Fuente: elaboración propia.

Figura 16. **Prototipo para agregar campus**



Agregar campus

Nombre Código o alias

Dirección

GUARDAR CANCELAR

Fuente: elaboración propia.

Figura 17. **Prototipo para agregar salón**



Agregar salón

Seleccione una ubicación:

Escoja un Campus

Seleccione un edificio:

Escoja un Edificio

Nombre Código o alias

Fuente: elaboración propia.

Figura 18. Prototipo para validar solicitud

8/2/2019
Seminario de Sistemas 2
Sección A
luisgil95crash@gmail.com

Ubicación:
Campus Central

Correo solicitante:
luisgil95crash@gmail.com

Curso:
Seminario de Sistemas 2

Nombre: Luis Gilberto Apellido: Barrios Vásquez

Inicio: 8:50 Fin: 10:30 Fecha: 8/2/2019

Edificio para la solicitud
Escoja un edificio

Salón para la solicitud
Escoja un salón

BLOQUEAR SALÓN

Comentario:

RECHAZAR SOLICITUD

Fuente: elaboración propia.

Figura 19. Prototipo para validar solicitud con salón bloqueado

8/2/2019
Seminario de Sistemas 2
Sección A
luisgil95crash@gmail.com

Ubicación:
Campus Central

Correo solicitante:
luisgil95crash@gmail.com

Curso:
Seminario de Sistemas 2

Nombre: Luis Gilberto Apellido: Barrios Vásquez

Inicio: 8:50 Fin: 10:30 Fecha: 8/2/2019

LIBERAR SALÓN

Información del salón
Código: 014
Nombre: Centro Tecnológico No.1
Capacidad: 30
Proyector: true
Duración de periodo: 50 minutos

Thursday:18/April/2019

CONSULTAR DISPONIBILIDAD

Horarios disponibles:
Horario inicial

AGREGAR HORARIO

Fuente: elaboración propia.

Figura 20. **Prototipo para validar solicitud con horarios agregados**

No. Personas 50	Proyector true	Fecha	Horario	
		18-4-2019	8:00 - 8:50	
Uso Laboratorio		18-4-2019	8:50 - 9:40	

APROBAR SOLICITUD Y APARTAR SALÓN

Comentario: _____

RECHAZAR SOLICITUD

Fuente: elaboración propia.

Figura 21. **Prototipo para historial de solicitudes**

Historial de solicitudes

Fecha inicio: Thursday:18/April/2019 Fecha final: Thursday:18/April/2019

FECHA CORREO TODAS

- 18/4/2019
luisgil95crash@gmail.com
Seminario de Sistemas 2
- 18/4/2019
luisgil95crash@gmail.com
Seminario de Sistemas 2
- 18/4/2019
luisgil95crash@gmail.com
Inteligencia Artificial

ELIMINAR SOLICITUD

Estado: Aprobada

Ubicación: Campus Central

Correo del solicitante: _____

Fuente: elaboración propia.

Figura 22. **Prototipo para la consulta de horarios**

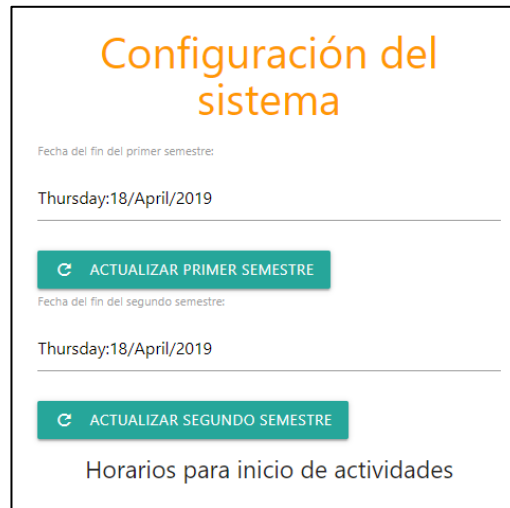
Fuente: elaboración propia.

Figura 23. **Prototipo de historial de horarios con consulta realizada**

Hora	Curso	Actividad	Ocupante	Eliminar
7:10 - 8:00	Libre	Libre	Nadie	
8:00 - 8:50	Libre	Libre	Nadie	
8:50 - 9:40	Seminario de Sistemas 2	Laboratorio	luisgil95crash@gmail.com	
9:40 - 10:30	Seminario de Sistemas 2	Laboratorio	luisgil95crash@gmail.com	
10:30 - 11:20	Libre	Libre	Nadie	
11:20 - 12:10	Libre	Libre	Nadie	
12:10 - 13:00	Libre	Libre	Nadie	
13:00 - 13:50	Libre	Libre	Nadie	

Fuente: elaboración propia.

Figura 24. **Prototipo de panel de configuración 1**



Fuente: elaboración propia.

Figura 25. **Prototipo de panel de configuración 2**

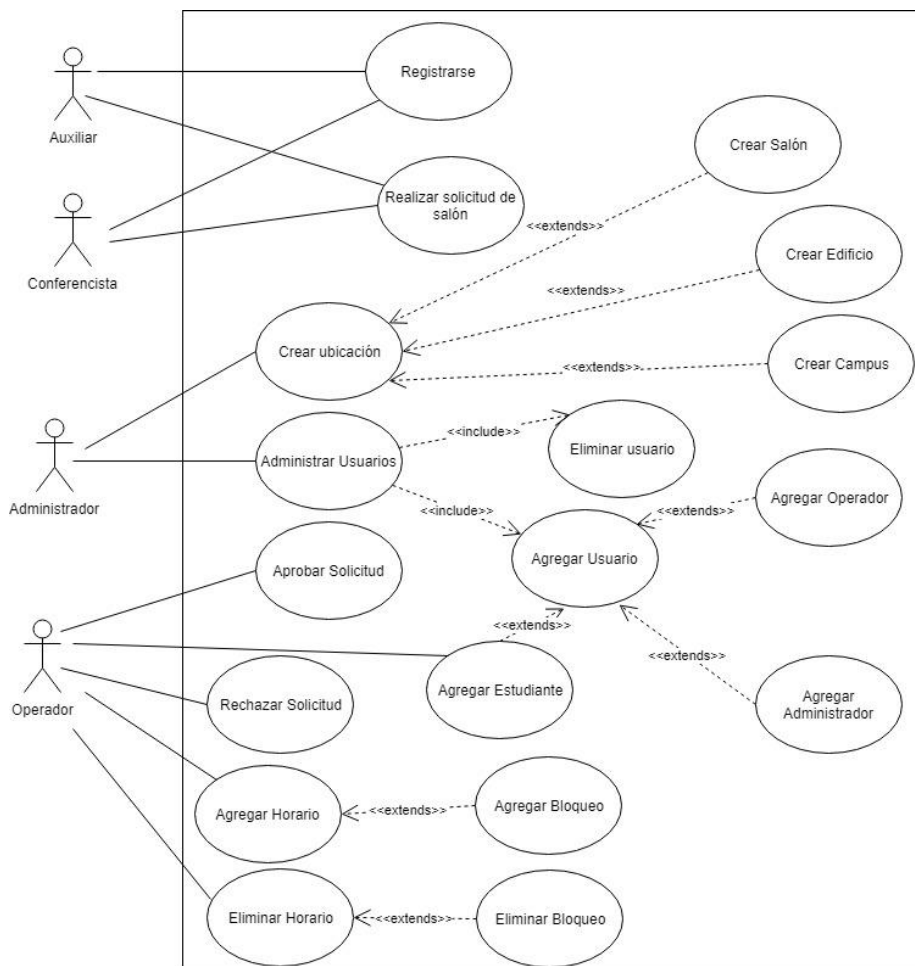


Fuente: elaboración propia.

3.2. Caso de uso

A través de los casos de uso se describen las acciones o actividades implicadas con el sistema. Se identificaron para el caso de uso múltiples actores, comprendidos en dos grupos, un grupo de usuarios finales de la aplicación (auxiliar, conferencista) y un grupo de administradores (administradores y operadores).

Figura 26. Caso de uso general de la aplicación

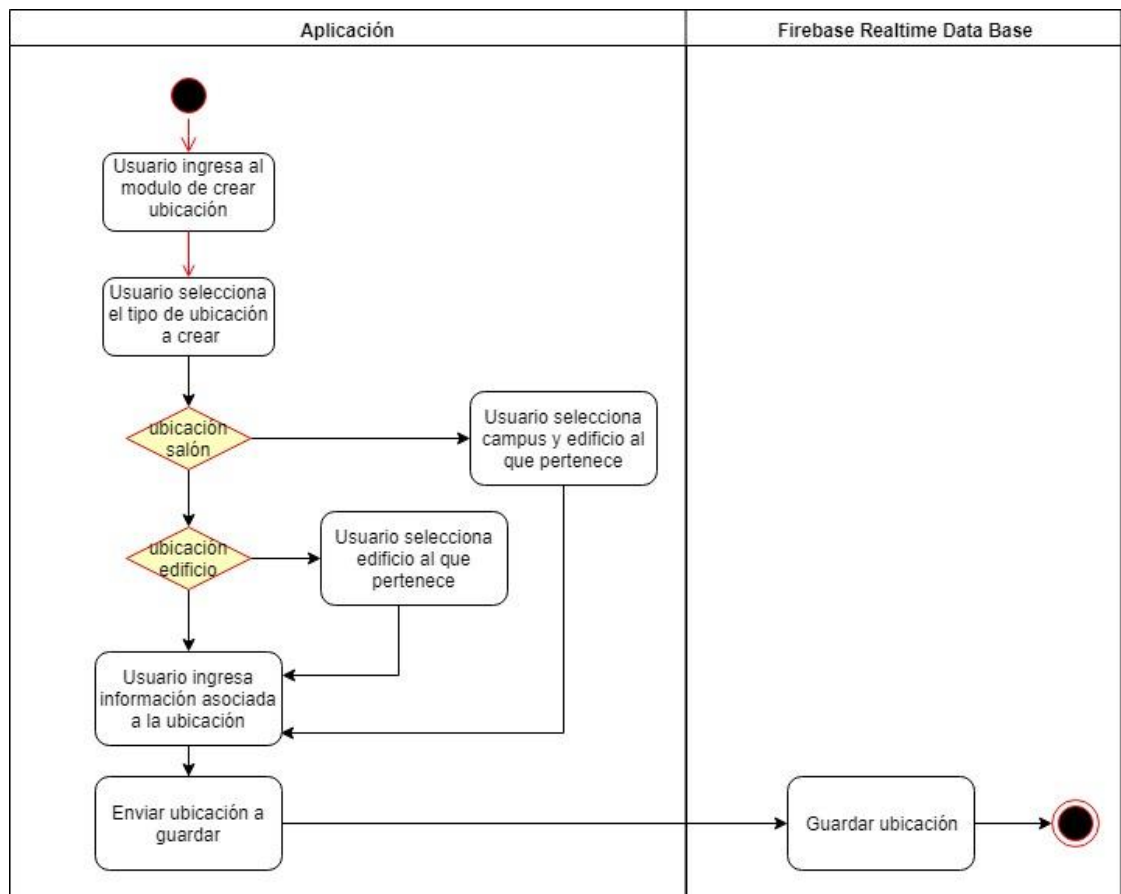


Fuente: elaboración propia.

3.3. Diagramas de actividades

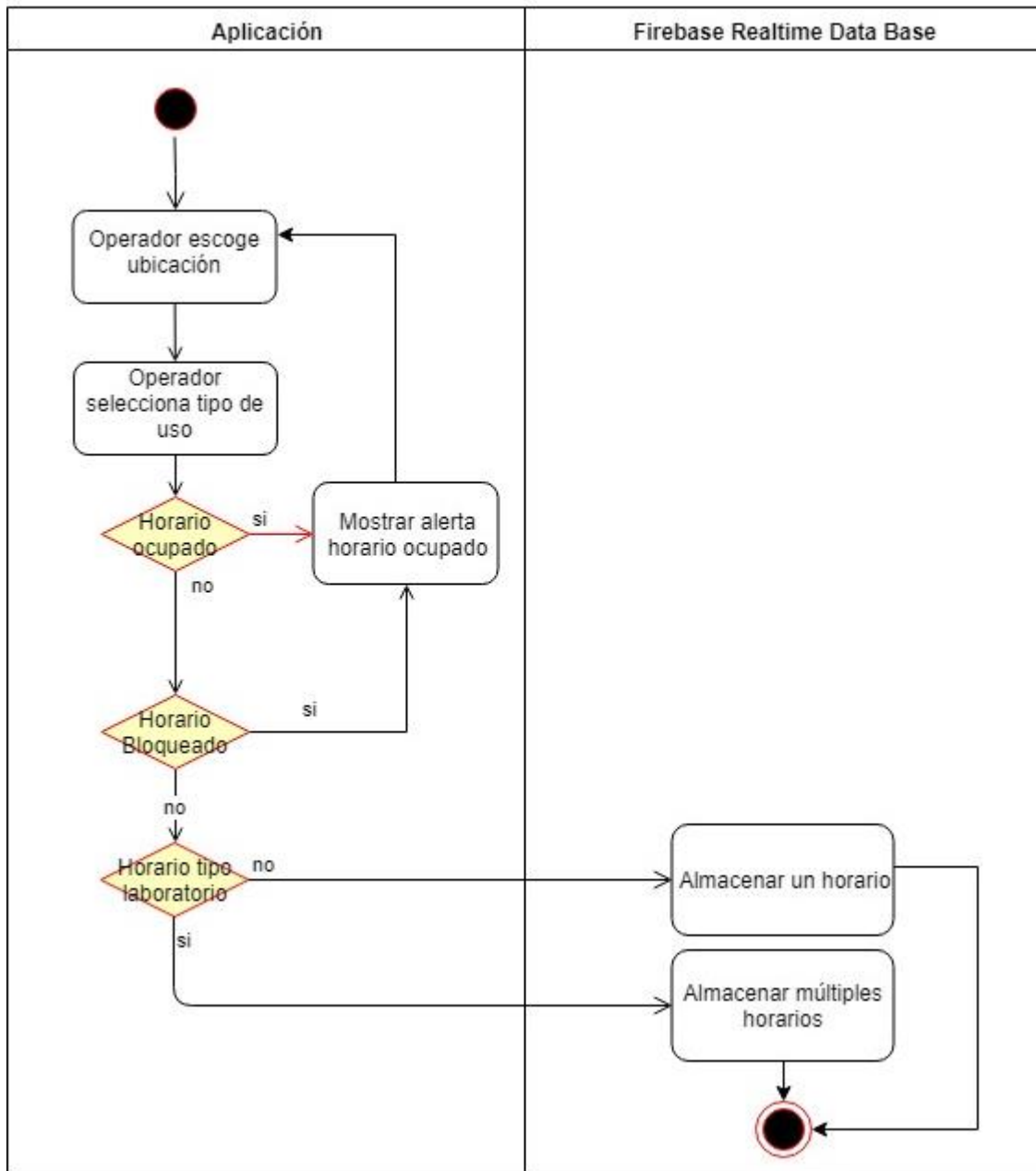
A través de un conjunto de diagramas de actividades se ha planteado los diferentes flujos y procedimientos lógicos que realiza el sistema propuesto.

Figura 27. **Procedimiento para agregar ubicación operador**



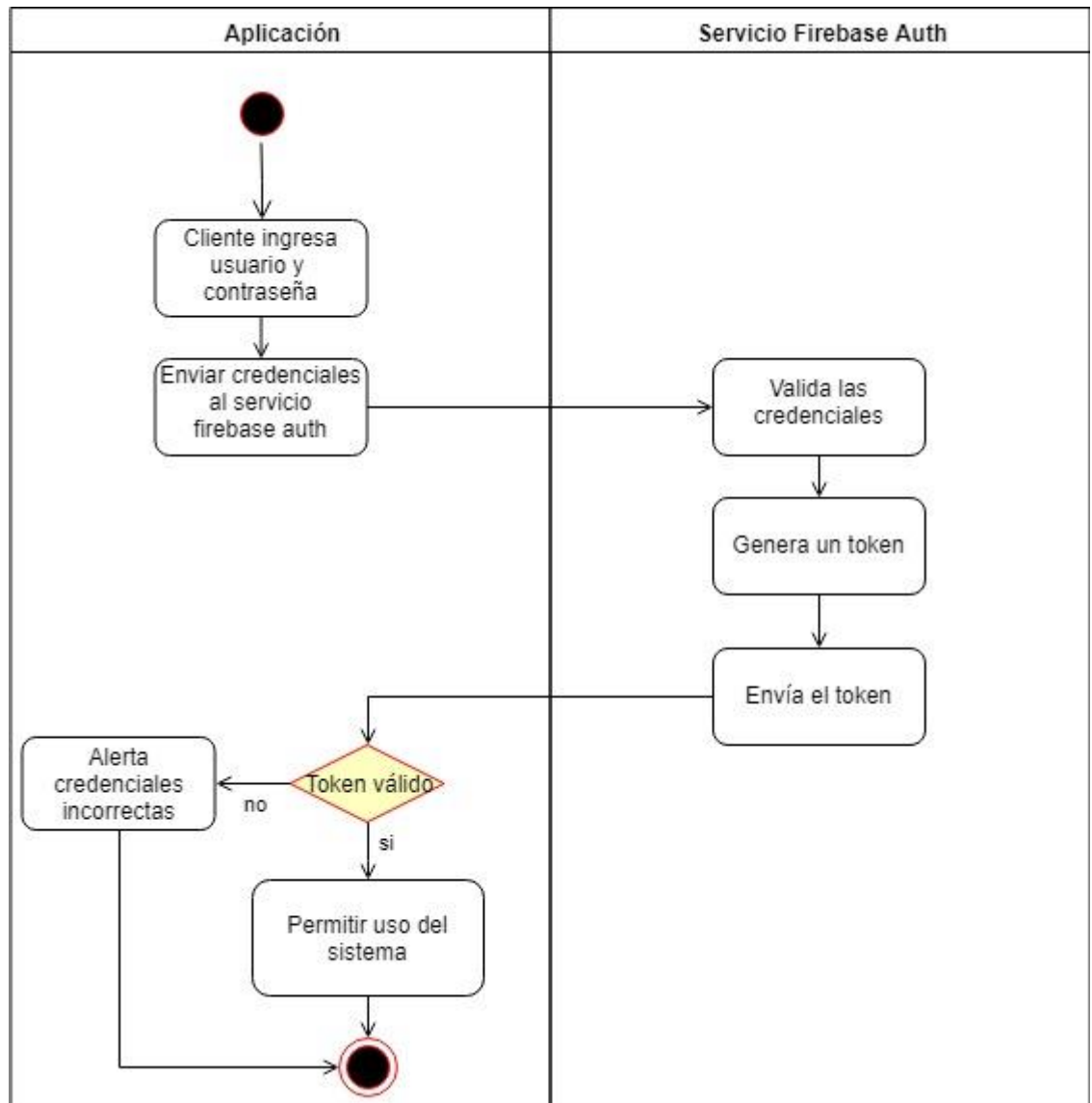
Fuente: elaboración propia.

Figura 28. Procedimiento para apartar horario operador



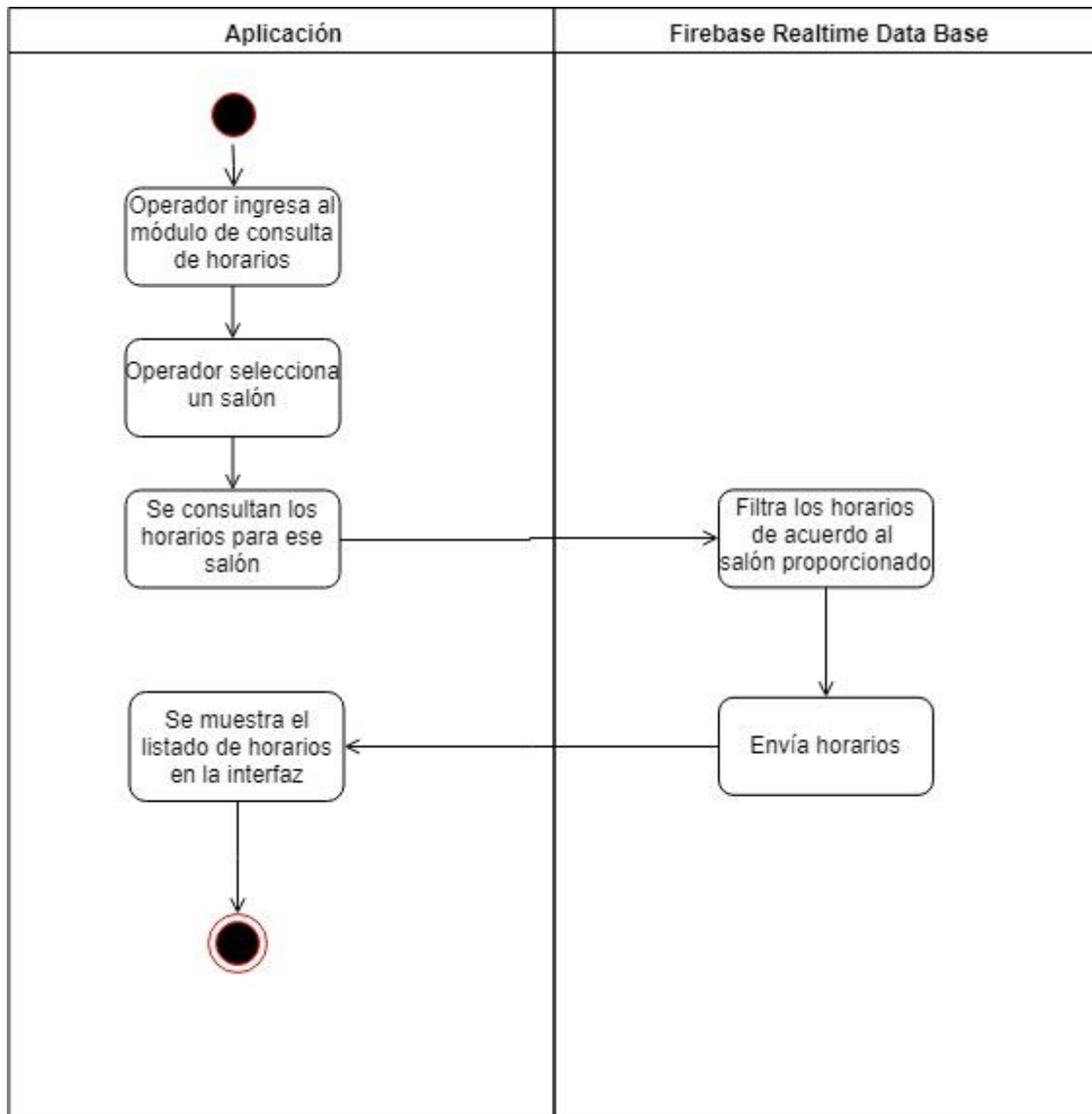
Fuente: elaboración propia.

Figura 29. Procedimiento para la autenticación de sesiones



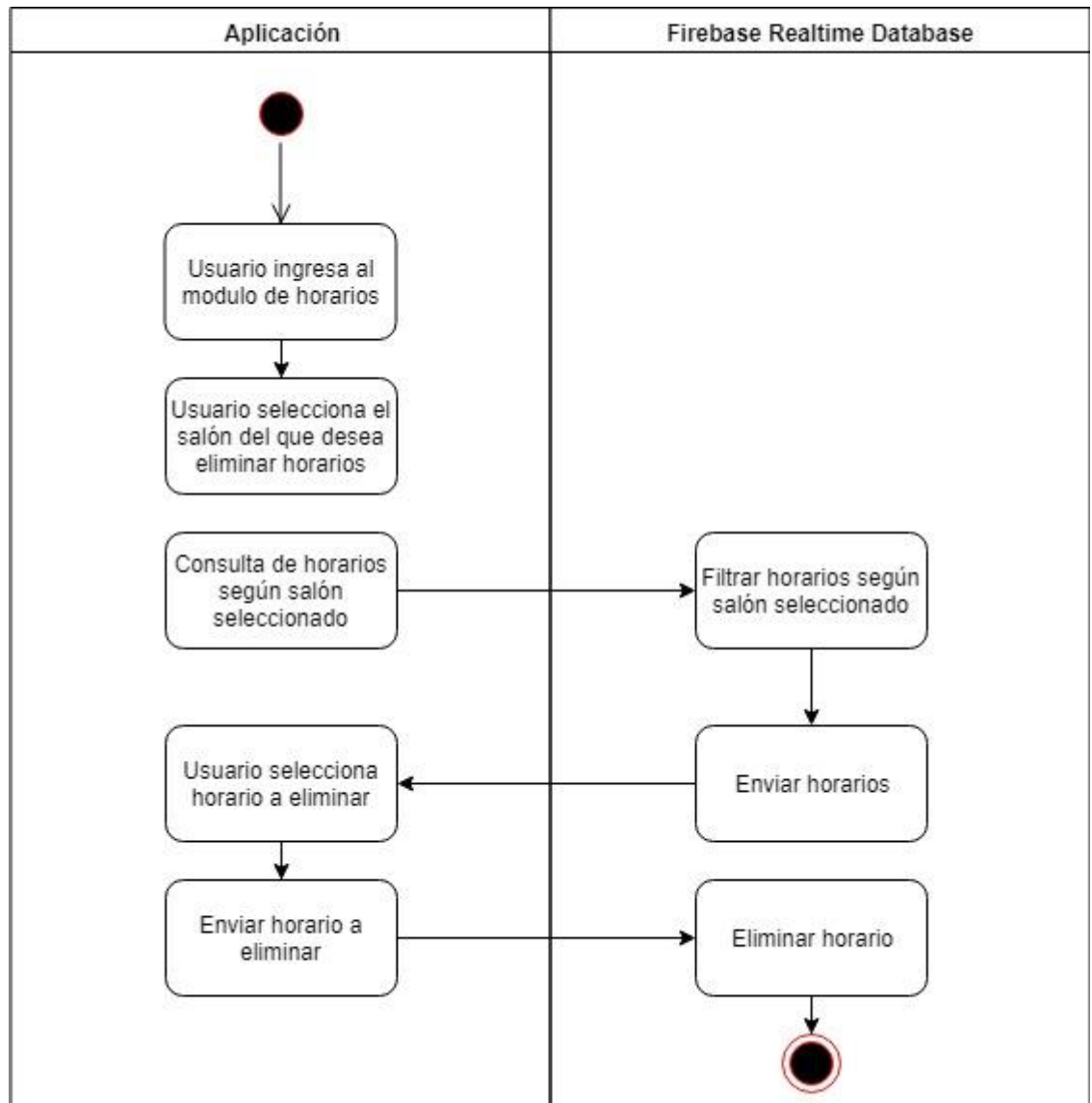
Fuente: elaboración propia.

Figura 30. **Procedimiento para consultar horarios**



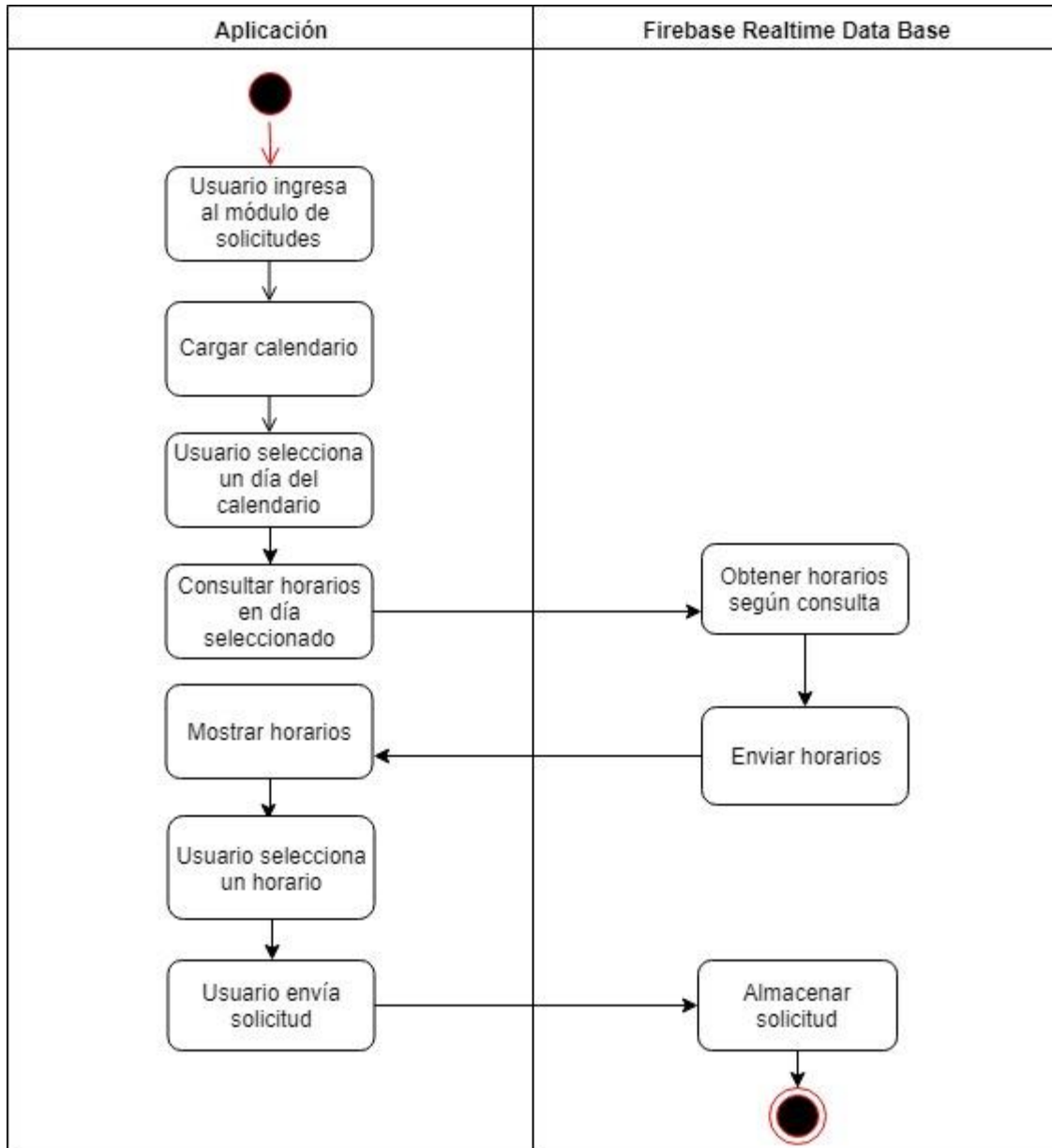
Fuente: elaboración propia.

Figura 31. Procedimiento para eliminar horario operador



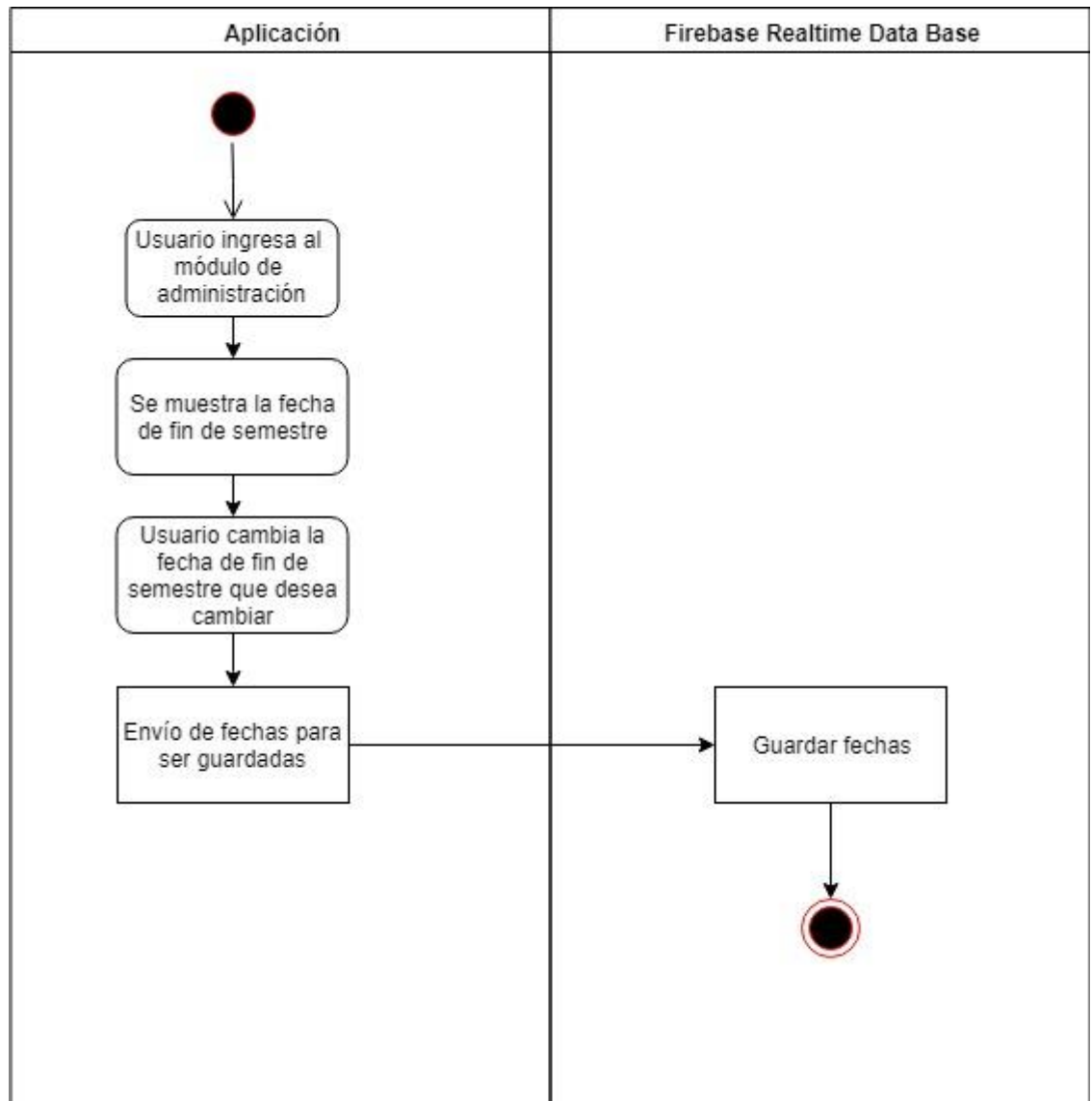
Fuente: elaboración propia.

Figura 32. Procedimiento para enviar solicitud de salón



Fuente: elaboración propia.

Figura 33. Procedimiento para establecer fin de semestre



Fuente: elaboración propia.

4. DOCUMENTACIÓN DE LA APLICACIÓN

4.1. Requisitos

A continuación, se muestra una serie de requerimientos necesarios para los operadores y los administradores puedan utilizarla.

- Operadores y administradores
 - PC u ordenador portátil
 - 2 gigabytes de RAM
 - Navegador Google Chrome versión 73.0.3683.103 o superior
 - 100 megabytes de espacio en disco
 - Conexión a internet

- Estudiantes
 - Smartphone
 - 512 megabytes de RAM
 - Navegador Google Chrome 73.0.3683.103 o superior
 - Conexión a internet

 - PC u ordenador portátil
 - 2 gigabytes de RAM

- Navegador Google Chrome versión 73.0.3683.103 o mayor
- Conexión a internet

4.2. Herramientas de desarrollo

A continuación, se detallan las herramientas utilizadas, tanto físicas como lógicas para la implementación de este sistema.

4.2.1. Hardware

Se detallan las especificaciones de la computadora que realizó el desarrollo de este sistema, condiciones mínimas ideales para su mantenimiento.

- Ordenador portátil
 - 6 gigabytes de RAM
 - Procesador Intel Core I5
 - Tarjeta de red
 - Acceso a internet

4.2.2. Software

Se detallan las herramientas de software utilizadas tanto para fabricar la aplicación y necesarias para dar mantenimiento a las mismas.

- NodeJS v10.15.1
- Firebase Hosting
- Firebase Realtime Database
- Firebase Functions

- Firebase Authentication
- AngularJS 1.7
- jQuery 2.1.1
- Materialize 1.0.0

4.3. Consideraciones de la implementación

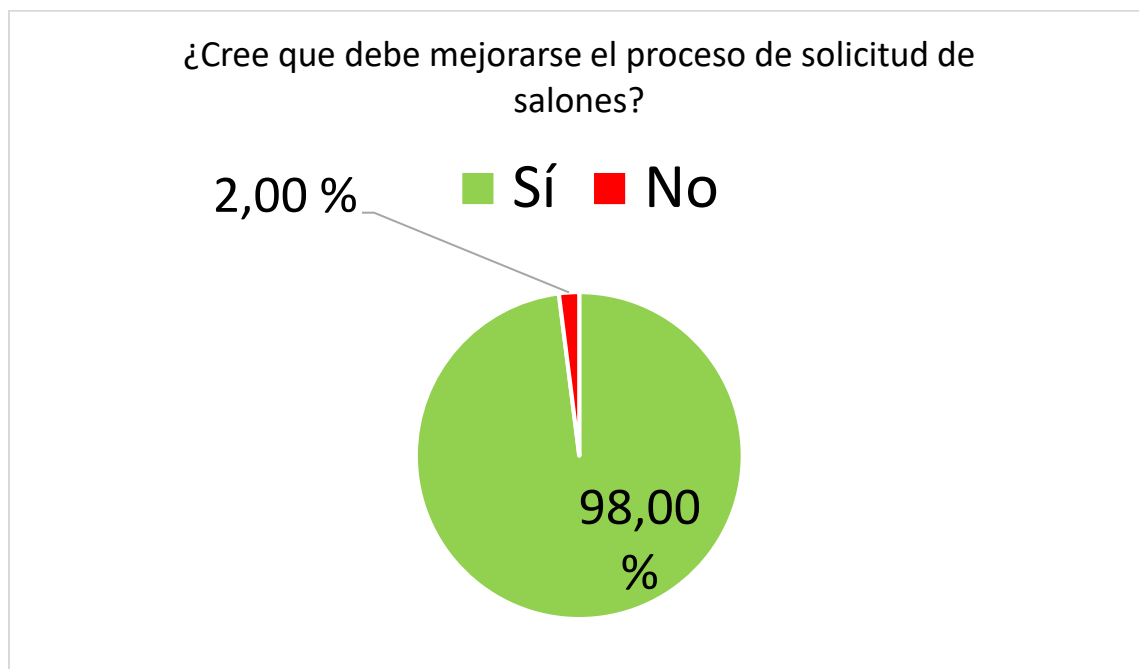
En esta sección se describen las consideraciones a tener presentes para implementar esta o una nueva versión de la aplicación:

- Acceso a internet
- Poseer una instalación de NodeJS en la versión recomendada
 - Poseer una instalación del complemento FirebaseCLI
- Poseer las credenciales del proyecto de Firebase

5. ANÁLISIS E INTERPRETACIÓN DE RESULTADOS

A continuación, se presentan los resultados que produjo una encuesta a 50 personas relacionadas con el proceso de reserva de salones, quienes utilizaron el sistema desarrollado; se busca, de esta forma, validar los objetivos de la investigación.

Figura 34. **Opinión de los auxiliares sobre el proceso de solicitud actual**



Fuente: elaboración propia.

Tabla I. **Mejora del proceso de solicitud de salones**

Respuestas	No.	%
Sí	49	98,00 %
No	1	2,00 %
Total	50	100,00 %

Fuente: elaboración propia.

La mayoría de los encuestados considera que debe mejorarse la forma en la que se solicitan salones; esto da lugar a que el sistema en desarrollo sea utilizado por la población objetivo y así pueda eliminarse una necesidad.

Figura 35. **Facilidad de uso del nuevo sistema**



Fuente: elaboración propia.

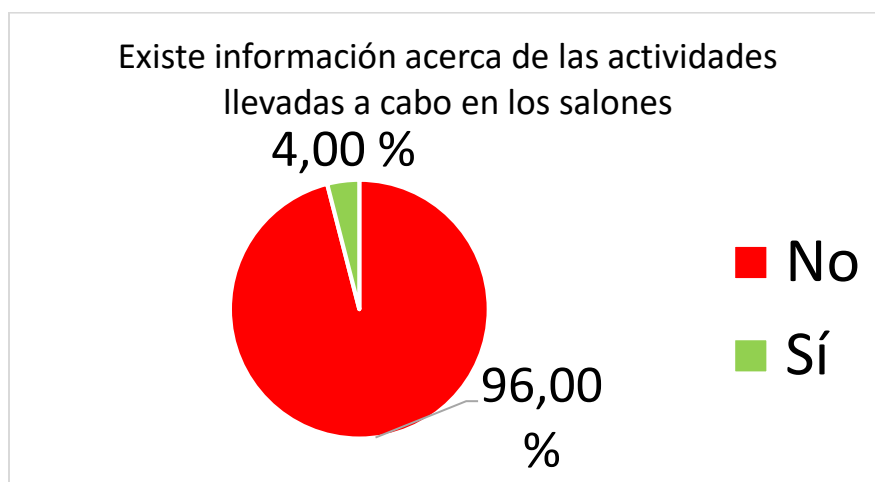
Tabla II. **Facilidad de uso del nuevo sistema**

Respuestas	No.	%
Sí	49	97,67 %
No	1	2,33 %
Total	50	100,00 %

Fuente: elaboración propia.

La mayoría de los encuestados opina que el sistema proporcionará una mayor facilidad para apartar salones. Esto viene a apoyar el primer objetivo del desarrollo de la aplicación: Implementar una herramienta mediante la cual se pueda gestionar la solicitud de salones de la escuela de Ingeniería en Ciencias y Sistemas.

Figura 36. **Opinión de los auxiliares sobre la accesibilidad de la información de los salones**



Fuente: elaboración propia.

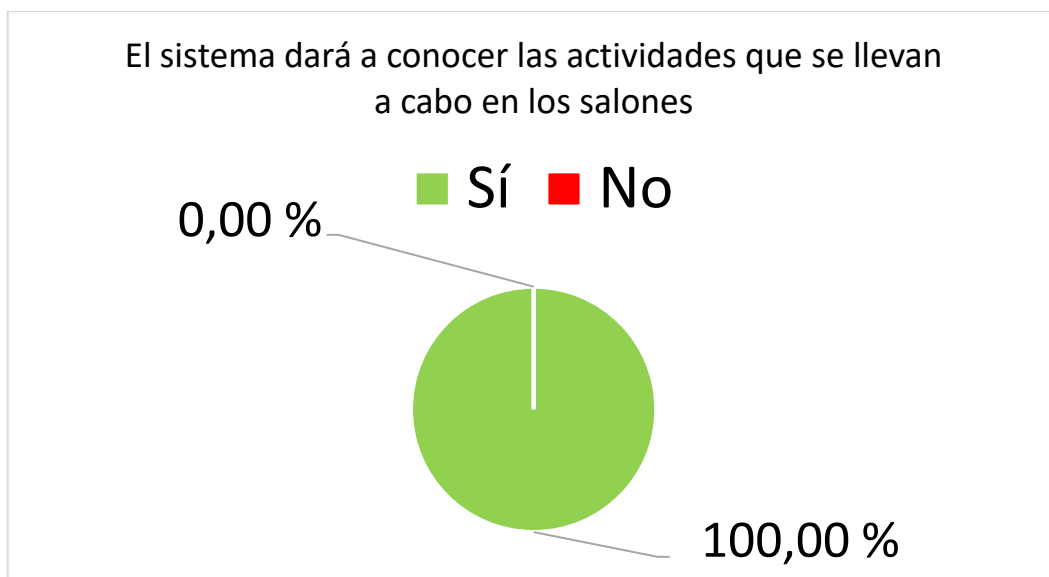
Tabla III. **Información del sistema sobre las actividades en los salones**

Respuestas	No.	%
Sí	48	96,00 %
No	2	4,00 %
Total	50	100,00 %

Fuente: elaboración propia.

La mayoría de los encuestados considera que no existe forma de consultar que actividades se llevan a cabo en los salones de la escuela; se demuestra que existe un problema con lugar para que sea resuelto por el sistema desarrollado.

Figura 37. **Información del sistema sobre las actividades en los salones**



Fuente: elaboración propia.

Tabla IV. **Información del sistema sobre las actividades en los salones**

Respuestas	No.	%
Sí	50	100,00 %
No	0	0,00 %
Total	50	100,00 %

Fuente: elaboración propia.

Todos los encuestados opinaron que el sistema desarrollado facilitará el acceso de información sobre las actividades que se llevan a cabo en los salones de la escuela. Esto valida el objetivo de desarrollo que busca: Llevar un control histórico de las asignaciones realizadas a los diferentes salones de la Escuela de Ingeniería de Ciencias y Sistemas.

Figura 38. **Uso de la aplicación web para realizar la solicitud de un salón**



Fuente: elaboración propia.

Tabla V. **Uso de la aplicación para apartar salones**

Respuestas	No.	%
Si	49	98,00 %
No	1	2,00 %
Total	50	100,00 %

Fuente: elaboración propia.

La mayoría de los encuestados acepta la idea de utilizar una aplicación para apartar y consultar horarios en los salones. Con esto se provee que el sistema no solo se percibe útil, sino que también sea utilizado.

CONCLUSIONES

1. Se realizó el desarrollo de un sistema para la gestión de solicitudes de salones, utilizando JavaScript, AngularJS y Firebase como componentes principales.
2. De acuerdo a la percepción de los usuarios, el sistema desarrollado es capaz de gestionar las solicitudes de salones de la escuela; facilita así el control y orden de las solicitudes.
3. De acuerdo a los usuarios el sistema desarrollado provee una forma de consultar las actividades que se llevan a cabo en los distintos salones de la escuela.
4. El sistema desarrollado facilita la asignación de salones en periodos de tiempo establecidos.

RECOMENDACIONES

1. Motivar a los estudiantes a utilizar la aplicación, estableciéndola como la única vía para gestión de solicitud de salones.
2. Impartir capacitación de uso del sistema a los operadores para que estos puedan operarlo correctamente y obtener su mayor beneficio.
3. Proveer de una conexión a internet estable para los operadores ya que esta conexión es vital para el correcto funcionamiento del sistema.

BIBLIOGRAFÍA

1. AngularFire. *A real time backend for AngularJS from Firebase*. [en línea]. <<https://www.firebase.com/docs/web/libraries/angular/api.html>>. [Consulta: 11 de febrero de 2019].
2. AngularJS. *API Reference*. [en línea]. <<https://docs.angularjs.org/api>>. [Consulta: 11 de febrero de 2019].
3. Firebase. *Firebase Guides*. [en línea]. <<https://firebase.google.com/docs/guides/>>. [Consulta: 11 de febrero de 2019].
4. Is Theory. *Information Systems Theories*. [en línea]. <https://is.theorizeit.org/wiki/Main_Page>. [Consulta: 11 de febrero de 2019].
5. JSON. *Introducing JSON*. [en línea]. <<https://www.json.org/>>. [Consulta: 11 de febrero de 2019].
6. Marketing Directo. *La importancia de contar con una página web en la era del nuevo consumidor*. [en línea]. <<https://www.marketingdirecto.com/digital-general/digital/la-importancia-de-contar-conuna-pagina-web-en-la-era-del-nuevo-consumidor>>. [Consulta: 11 de febrero de 2019].

7. MongoDB. *NoSQL Databases Explained*. [en línea]. <<https://www.mongodb.com/nosql-explained>>. [Consulta: 11 de febrero de 2019].
8. Mozilla Developer Network. *JavaScript*. [en línea]. <<https://developer.mozilla.org/es/docs/Web/JavaScript>>. [Consulta: 11 de febrero de 2019].
9. W3SCHOOLS. *Introduction to HTML*. [en línea]. <https://www.w3schools.com/html/html_intro.asp>. [Consulta: 11 de febrero de 2019].