



REPÚBLICA DEL ECUADOR

Escuela Politécnica Nacional

" E SCIENTIA HOMINIS SALUS "

La versión digital de esta tesis está protegida por la Ley de Derechos de Autor del Ecuador.

Los derechos de autor han sido entregados a la "ESCUELA POLITÉCNICA NACIONAL" bajo el libre consentimiento del (los) autor(es).

Al consultar esta tesis deberá acatar con las disposiciones de la Ley y las siguientes condiciones de uso:

- Cualquier uso que haga de estos documentos o imágenes deben ser sólo para efectos de investigación o estudio académico, y usted no puede ponerlos a disposición de otra persona.
- Usted deberá reconocer el derecho del autor a ser identificado y citado como el autor de esta tesis.
- No se podrá obtener ningún beneficio comercial y las obras derivadas tienen que estar bajo los mismos términos de licencia que el trabajo original.

El Libre Acceso a la información, promueve el reconocimiento de la originalidad de las ideas de los demás, respetando las normas de presentación y de citación de autores con el fin de no incurrir en actos ilegítimos de copiar y hacer pasar como propias las creaciones de terceras personas.

Respeto hacia sí mismo y hacia los demás.

ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

DESARROLLO DE UN PROTOTIPO DE APLICACIÓN WEB PARA LA RESERVACIÓN DE COMPUTADORAS Y AULAS DEL LABORATORIO DE INFORMÁTICA DE LA FIEE-EPN

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y REDES DE INFORMACIÓN

RODDY STEEVEN ZAMORA RODRÍGUEZ
roddy.zamora@epn.edu.ec

DIRECTORA: Ph. D. PRISCILLA FERNANDA JIMÉNEZ PAZMIÑO
CODIRECTOR: M. Sc. XAVIER ALEXANDER CALDERÓN HINOJOSA

Quito, agosto 2021

AVAL

Certificamos que el presente trabajo fue desarrollado por Roddy Steeven Zamora Rodríguez, bajo nuestra supervisión.

Ph. D. PRISCILLA FERNANDA JIMÉNEZ PAZMIÑO
DIRECTORA DEL TRABAJO DE TITULACIÓN

M. Sc. XAVIER ALEXANDER CALDERÓN HINOJOSA
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Roddy Steeven Zamora Rodríguez, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

RODDY STEEVEN ZAMORA RODRÍGUEZ

DEDICATORIA

A mis queridos padres Nely y Julio que con su amor, comprensión y apoyo incondicional me motivan a ser una mejor persona. Este trabajo es gracias a los valores, virtudes y buenas enseñanzas que me han inculcado.

A mi hermano Slayther por ser mi amigo y estar conmigo en los buenos y malos momentos.

AGRADECIMIENTO

Quiero agradecer primeramente a Dios por todas las bendiciones y oportunidades que me ha regalado a lo largo de mi vida.

A mis amados padres por ser mi principal ejemplo e inspiración, por su amor incondicional, y por todos los buenos valores que me han inculcado. A mi hermano por estar ahí en los buenos y malos momentos. Los quiero mucho.

A los profesores de la Escuela Politécnica Nacional por compartirme sus conocimientos a lo largo de mi carrera universitaria, una mención especial a mi tutor académico el ingeniero Víctor Hugo Reyes Cifuentes por todo su apoyo y consejos a lo largo de la carrera universitaria.

A los asistentes del Laboratorio de Informática de la Facultad de Ingeniería Eléctrica y Electrónica por brindarme la información necesaria para el desarrollo de este trabajo de titulación.

A todos mis familiares por brindarme su ayuda y buenos deseos, especialmente a mi tío Hicter por abrirme las puertas de su hogar y apoyarme cuando más lo necesitaba.

A mis amigos y compañeros de la universidad con los que compartí buenos y malos momentos, gracias por todos los momentos vividos que llevare siempre en mi corazón.

Finalmente agradecer a mi directora la ingeniera Priscilla Jiménez y a mi codirector el ingeniero Xavier Calderón por su paciencia, sus consejos y su colaboración en la realización de este trabajo de titulación.

"Tanto si crees que puedes, como si crees que no puedes, estás en lo cierto"

- Henry Ford

ÍNDICE DE CONTENIDO

Contenido

AVAL	I
DECLARACIÓN DE AUTORÍA.....	II
DEDICATORIA.....	III
AGRADECIMIENTO.....	IV
ÍNDICE DE CONTENIDO.....	V
ÍNDICE DE FIGURAS	VII
ÍNDICE DE TABLAS	X
ÍNDICE DE CÓDIGOS	XI
RESUMEN	XII
ABSTRACT	XIII
1. INTRODUCCIÓN.....	14
1.1 OBJETIVOS	14
1.2 ALCANCE	15
1.3 MARCO TEÓRICO.....	18
1.3.1 APLICACIONES WEB.....	18
1.3.2 ARQUITECTURA EN CAPAS	19
1.3.3 FRAMEWORK BLAZOR.....	21
1.3.4 MICROSOFT AZURE	27
2. METODOLOGÍA.....	28
2.1. DISEÑO	28
2.1.1 TABLERO KANBAN	28
2.1.2 DEFINICIÓN DE REQUERIMIENTOS.....	29
2.1.3 DISEÑO CAPA DE DATOS.....	47
2.1.4 DISEÑO CAPA LÓGICA DE NEGOCIO	48
2.1.5 DISEÑO CAPA DE PRESENTACIÓN	49
2.2. IMPLEMENTACIÓN.....	60
2.2.1. ACTUALIZACIÓN DEL TABLERO KANBAN	60
2.2.2. INSTALACIÓN DE LAS HERRAMIENTAS DE SOFTWARE	61
2.2.3. IMPLEMENTACIÓN CAPA DE DATOS.....	63
2.2.4. IMPLEMENTACIÓN DE CAPA DE NEGOCIO	65
2.2.5. IMPLEMENTACIÓN CAPA DE PRESENTACIÓN	67
2.2.6. ALOJAMIENTO DE LA APLICACIÓN EN MICROSOFT AZURE.....	68

3.	RESULTADOS Y DISCUSIÓN	71
3.1	ACTUALIZACIÓN DEL TABLERO KANBAN	71
3.2	PRUEBAS INICIALES DE FUNCIONAMIENTO	72
3.2.1	PRUEBA DE FUNCIONAMIENTO DE RESERVAR COMPUTADOR.....	73
3.2.2	PRUEBA DE FUNCIONAMIENTO DE RESERVAR AULA	75
3.2.3	PRUEBA DE FUNCIONAMIENTO DE MIS RESERVACIONES	77
3.2.4	PRUEBA DE FUNCIONAMIENTO DE CALIFICAR SERVICIO	77
3.3	PRUEBAS DE USUARIO.....	78
3.3.1	PRUEBAS DEL ROL USUARIO	78
3.3.2	PRUEBAS DEL ROL ASISTENTE DEL LABORATORIO	80
3.3.3	PRUEBAS DEL ROL ADMINISTRADOR TÉCNICO.....	82
3.4	ACTUALIZACIÓN FINAL DEL TABLERO KANBAN	83
4.	CONCLUSIONES Y RECOMENDACIONES.....	85
4.1	CONCLUSIONES.....	85
4.2	RECOMENDACIONES	86
5.	REFERENCIAS BIBLIOGRÁFICAS	87
	ANEXOS	91

ÍNDICE DE FIGURAS

Figura 1.1 Esquema general del prototipo	17
Figura 1.2 Arquitectura de 3 capas	19
Figura 1.3 Arquitectura de Blazor WebAssembly	23
Figura 1.4 Arquitectura de Blazor Server	24
Figura 1.5 Ejemplo del componente Blazor Scheduler.....	26
Figura 1.6 Ejemplo del componente Blazor Charts	26
Figura 2.1 Diagrama de casos de uso del módulo de registro	33
Figura 2.2 Diagrama de casos de uso del módulo de autenticación	33
Figura 2.3 Diagrama de casos de uso del módulo de reservaciones	34
Figura 2.4 Diagrama de casos de uso del módulo administración del laboratorio	35
Figura 2.5 Diagrama de casos de uso del módulo de reportes	35
Figura 2.6 Diagrama de casos de uso de administración de usuarios	36
Figura 2.7 Diagrama de contexto del sistema de reserva	36
Figura 2.8 Diagrama de actividades del módulo de registro	37
Figura 2.9 Diagrama de actividades del módulo de autenticación	38
Figura 2.10 Diagrama de actividades para crear reserva.....	39
Figura 2.11 Diagrama de actividades para aprobar o negar reservación	40
Figura 2.12 Diagrama de actividades para calificar el servicio de préstamo	40
Figura 2.13 Diagrama de actividades del registro de entrada y salida	41
Figura 2.14 Diagrama de actividades módulo de administración del laboratorio..	42
Figura 2.15 Diagrama de actividades del módulo de reportes	43
Figura 2.16 Diagrama de actividades de administración de usuarios	44
Figura 2.17 Diagrama de estados para la reserva de computadoras.....	45
Figura 2.18 Diagrama de estados para la reserva de aulas	46
Figura 2.19 Diagrama relacional de la base de datos	47
Figura 2.20 Diagrama de clases.....	48
Figura 2.21 Sketch de inicio de sesión, requerimiento RF002	49
Figura 2.22 Sketch de registro de usuarios, requerimiento RF001	49
Figura 2.23 Sketch del menú de usuario, requerimiento RF003	50
Figura 2.24 Sketch de reserva de computador del menú de usuario, requerimientoRF004.....	50
Figura 2.25 Sketch de reserva de aula del menú de usuario, requerimiento RF005	51

Figura 2.26 Sketch que muestra las reservaciones del usuario, requerimiento RF006.....	51
Figura 2.27 Sketch de calificar servicio del menú de usuario, requerimiento RF007	52
Figura 2.28 Sketch del menú del asistente del laboratorio	52
Figura 2.29 Sketch de las peticiones para reservar aulas, requerimiento RF005	53
Figura 2.30 Sketch para gestionar reservaciones, requerimientos: RF008 y RF009	53
Figura 2.31 Sketch para gestionar laboratorio, requerimientos RF010 y RF011..	54
Figura 2.32 Sketch para gestionar usuarios, requerimiento RF015	54
Figura 2.33 Sketch de registros de entrada y salida, requerimientos: RF0012 y RF013	55
Figura 2.34 Sketch de reportes estadísticos, requerimiento RF014.....	55
Figura 2.35 Sketch de comentarios y sugerencias, requerimiento RF014	56
Figura 2.36 Mapa de navegación web del usuario	57
Figura 2.37 Mapa de navegación web asistentes del laboratorio.....	58
Figura 2.38 Mapa de navegación web del administrador técnico.....	59
Figura 2.39 Creación de un proyecto Aplicación Blazor Server	61
Figura 2.40 Página web de inicio del proyecto Blazor Server	61
Figura 2.41 Administrador de paquetes NuGets	62
Figura 2.42 Interfaz gráfica realizada en Blazor Server	68
Figura 2.43 Establecer firewall del servidor Azure SQL Server.....	68
Figura 2.44 Configuración del firewall Azure SQL para la IP del cliente	69
Figura 2.45 Cadena de conexión de la base de datos Laboratorio	69
Figura 2.46 Acceso a la aplicación web alojada en Microsoft Azure	70
Figura 3.1 Credenciales de usuario incorrectas	72
Figura 3.2 Página principal para el rol usuario	73
Figura 3.3 Mensaje de horario no permitido al reservar un computador	73
Figura 3.4 Reserva de computador exitoso.....	74
Figura 3.5 Correo de inicio de sesión inusual de la cuenta Microsoft.....	75
Figura 3.6 Modal para crear una reserva de aula.....	76
Figura 3.7 Eventos del usuario actualizados en el calendario.....	76
Figura 3.8 Mensaje de horario no permitido al reservar un aula	77
Figura 3.9 Lista de reservaciones del usuario	77
Figura 3.10 Lista de reservaciones para evaluar.....	77

Figura 3.11 Lista de preguntas para calificar el servicio de préstamo..... 78

ÍNDICE DE TABLAS

Tabla 1.1 Descripción de los elementos principales de una aplicación web	18
Tabla 1.2 Descripción de los lenguajes de la capa de presentación	19
Tabla 1.3 Descripción de las tecnologías de la capa de lógica de negocio.....	20
Tabla 1.4 Descripción de los elementos principales de una aplicación BlazorServer	24
Tabla 2.1 Tablero Kanban con las actividades al inicio de la fase de diseño	28
Tabla 2.2 Requerimientos funcionales del prototipo.....	30
Tabla 2.3 Requerimientos no funcionales del prototipo.....	31
Tabla 2.4 Módulos del prototipo	32
Tabla 2.5 Tablero Kanban con las actividades al inicio de la fase deimplementación	60
Tabla 2.6 Paquetes NuGets instalados	62
Tabla 3.1 Tablero Kanban con las actividades al inicio de la fase de resultados .	71
Tabla 3.2 Resultado de las encuestas de satisfacción del rol usuario	79
Tabla 3.3 Errores encontrados en las pruebas del rol usuario	79
Tabla 3.4 Resultado de las encuestas de satisfacción del rol asistente	80
Tabla 3.5 Errores encontrados en las pruebas del rol usuario	81
Tabla 3.6 Sugerencias obtenidas en las pruebas del rol asistente	81
Tabla 3.7 Resultado de las encuestas de satisfacción rol administrador técnico .	82
Tabla 3.8 Actualización final del tablero Kanban.....	83

ÍNDICE DE CÓDIGOS

Código 1.1 Ejemplo de plantilla Razor.	22
Código 2.1 Referencias JavaScripts de las librerías y paquetes Nugets.....	63
Código 2.2 Crear y manipular una base de datos	63
Código 2.3 Creación de la tabla usuario	64
Código 2.4 Ingreso en la base de datos del administrador técnico	64
Código 2.5 Creación del procedimiento almacenado sp_ObtenerMisReservasPC	64
Código 2.6 Método para abrir una conexión a la base de datos	65
Código 2.7 Método para obtener en número de reservas de aulas.....	66
Código 2.8 Llamado de los métodos de la capa de negocio desde un archivoRazor	67
Código 2.9 Creación de una interfaz gráfica con un componente Razor	67

RESUMEN

El presente trabajo describe el diseño e implementación de un prototipo de aplicación web que permite a los estudiantes y profesores reservar aulas o computadoras del Laboratorio de Informática FIEE-EPN de manera online y al personal del laboratorio gestionar a través de un sistema informático el préstamo de recursos del Laboratorio de Informática.

El prototipo fue desarrollado con Blazor Server utilizando como gestor de base de datos SQL Server. Una vez terminado el prototipo este fue alojado en el servicio de computación en la nube Microsoft Azure.

El presente documento está estructurado por cuatro capítulos:

El primer capítulo presenta los conceptos acerca de las aplicaciones webs, la arquitectura de capas, Microsoft Azure y el *framework* Blazor. Además, se definen las herramientas tecnológicas utilizadas en el desarrollo de este prototipo.

El segundo capítulo presenta la metodología empleada para la realización del prototipo, los requerimientos funcionales y no funcionales de la aplicación, el diseño y la implementación de las capas que conforman el sistema de reservas, y se detallan los pasos a seguir para alojar la aplicación en Microsoft Azure.

El tercer capítulo contiene las pruebas de las diferentes funcionalidades del prototipo, los resultados de las encuestas de satisfacción de las pruebas de usuario, los errores encontrados y sus respectivas correcciones.

Finalmente, en el cuarto capítulo se exponen las conclusiones y recomendaciones obtenidas en el desarrollo de este trabajo de titulación.

PALABRAS CLAVE: Reservación online, Aplicación web, Microsoft Azure, Blazor, ASP.NET Core

ABSTRACT

The present work describes the design and implementation of a web application prototype that allows users to reserve classrooms or computers of the Informatic Laboratory FIEE-EPN online and for the laboratory staff to manage through a computer system the loan of the Informatic Laboratory's resources.

The prototype was developed with Blazor Server using SQL Server as database manager. Once the prototype is finished, it was hosted in the Microsoft Azure cloud computing service.

Four chapters structure this document:

The first chapter presents web applications, layer architecture, Microsoft Azure, and the Blazor framework's concepts. In addition, it includes the technological tools utilized for the development of this prototype.

The second chapter presents the methodology used to carry out the prototype, the application's functional and non-functional requirements, the design and implementation of the reservation management system, and the step-by-step instructions for hosting the application in Microsoft Azure.

The third chapter contains the tests of the different functionalities of the prototype, the results of the satisfaction surveys of the user tests, the errors found and their respective corrections.

Finally, the fourth chapter four presents the conclusions and recommendations obtained in the development of this work.

KEYWORDS: Online reservation, Web application, Microsoft Azure, Blazor, ASP.NET Core

1. INTRODUCCIÓN

La Facultad de Ingeniería Eléctrica y Electrónica de la Escuela Politécnica Nacional (FIEE-EPN) permite a los estudiantes usar los computadores del laboratorio de Informática de manera totalmente gratuita. Actualmente, para que los estudiantes puedan usar las computadoras deben acudir personalmente al laboratorio y preguntar al personal encargado acerca de la disponibilidad de estas, en algunos casos los laboratorios están ocupados debido a los horarios de clases o mantenimiento de los equipos, lo cual puede provocar que los estudiantes no puedan usar una computadora en la hora deseada.

En caso de que algún laboratorio esté disponible el estudiante debe presentar su cédula al personal que esté en el laboratorio y este le indicará el aula que podrá usar; una vez que el estudiante haya terminado de usar la computadora deberá llenar a mano las horas de entrada y salida, los programas que usó y que tan satisfecho está con el préstamo del laboratorio. Actualmente todo este proceso para usar una computadora o aula se lo debe realizar de manera presencial y la información de registro debe ser llenada a mano en hojas de papel.

Por estas razones se propone desarrollar un prototipo de aplicación web que permita reservar computadoras y aulas de manera online, permitiéndole a los estudiantes y profesores conocer los horarios disponibles de cada aula del laboratorio, además ayudará al personal del laboratorio a gestionar de manera organizada la reservación de aulas y computadoras, así como llevar un registro digital de entrada y salida del laboratorio.

El uso de un sistema de reservación en línea conlleva muchas ventajas ya que se puede realizar desde cualquier parte que se disponga conexión a internet, sin necesidad de realizar filas, algunos sistemas de reserva incluso dan la posibilidad al usuario de cambiar o cancelar sus reservaciones, entre otras ventajas [1]. Todo esto de manera sencilla y rápida ya que se evitarán los problemas de los antiguos sistemas de reservas como los que son por teléfono o de manera presencial, los cuales además de consumir mucho tiempo a los clientes, algunos solo están disponibles dentro de un intervalo de tiempo y requieren más miembros del personal para manejar las reservas [2].

1.1 OBJETIVOS

El objetivo general de este Proyecto Técnico es: desarrollar un prototipo de aplicación web para la reservación de computadoras y aulas del laboratorio de informática de la FIEE-EPN.

Los objetivos específicos del Proyecto Técnico son:

- Analizar los fundamentos teóricos de las herramientas necesarias para el desarrollo de este trabajo de titulación.
- Diseñar los elementos que conforman el prototipo de reserva de computadores y aulas del laboratorio de Informática de la FIEE-EPN en base a los requerimientos mencionados para cada uno de los roles.
- Implementar los elementos de la aplicación previamente diseñados.
- Validar el correcto funcionamiento del prototipo para cada uno de los roles especificados.

1.2 ALCANCE

En este trabajo se plantea desarrollar un prototipo de aplicación web que permita a los estudiantes y profesores reservar computadoras y aulas del laboratorio de informática de la FIEE-EPN en el horario que le corresponde a la FIEE. El prototipo será desarrollado con ASP.NET Core Blazor y SQLServer. El flujo de trabajo relacionado al presente trabajo de titulación será visualizado utilizando la metodología Kanban.

El esquema general del prototipo se muestra en la figura 1.1.

Inicialmente la aplicación mostrará una vista de inicio de sesión permitiendo que los usuarios se autenticuen para poder ingresar al sistema.

En caso de que el usuario no esté inscrito debe solicitar la creación de una cuenta, el sistema verificará que el usuario sea estudiante o profesor de la EPN y se procederá a crearle la cuenta. Después de tener la cuenta el usuario tendrá la opción de usar la contraseña que se le creó por defecto o la puede cambiar en el sistema.

Una vez que el usuario ingrese al sistema se le presentará una vista en la que se tendrán diferentes opciones, dichas opciones dependerán del tipo de usuario que ingresó al sistema. Los roles son los siguientes.

Usuario:

- Reservar un computador o aula del laboratorio.
- Cancelar o modificar una reservación.
- Realizar una calificación de experiencia indicando que tan satisfecho está con el servicio de reserva.

Asistente del laboratorio:

- Crear nuevos usuarios que quieran usar el sistema de reserva.
- Podrá modificar los horarios de préstamo de las aulas y computadoras.
- En casos de emergencia podrá cancelar o modificar las reservaciones ya hechas por los usuarios.
- Actualizar los programas de las computadoras de las respectivas aulas para que se muestre en el sistema de reserva.
- Aumentar o disminuir el número de computadoras disponibles por cada aula del laboratorio.
- Ver el registro de entrada y salida de los usuarios que ocuparon las aulas y computadoras.
- Generar reportes estadísticos respecto a la información de préstamo de aulas y computadoras.

Administrador técnico:

- Podrá realizar cualquier de las funcionalidades del asistente del laboratorio y del usuario.
- Podrá crear, editar, eliminar o visualizar todos los usuarios y asistentes del sistema de préstamos.

Los módulos que integrarán el sistema son: **registro, autenticación, reservaciones, administración del laboratorio, reportes y administración de usuarios.**

- **Módulo de registro:** Permite a los usuarios que deseen usar el sistema de préstamo registrarse, deberán ingresar su número de cédula y el sistema verificará que dicho usuario pertenezca a la EPN, una vez hecha la verificación se le enviará al correo del usuario una contraseña para que pueda ingresar al sistema, dicha contraseña podrá ser cambiada por el usuario.
- **Módulo de autenticación:** Permite a los usuarios, asistentes del laboratorio y administrador técnico ingresar a la aplicación web mediante su cédula y contraseña, y actualizar datos como la contraseña y número de celular.
- **Módulo de reservaciones:** Permite a los usuarios realizar reservaciones de computadoras o de aulas del laboratorio. El usuario podrá ver, modificar o eliminar

sus reservaciones. Hay que destacar que las reservaciones de aulas hechas por los usuarios no serán modificables solo podrán ser eliminadas. Además, da la opción a los asistentes del laboratorio y administrador técnico hacer reservaciones de las aulas para los horarios de clases de cada semestre o algún evento en particular, y eliminar o actualizar las reservaciones de computadoras hechas por los usuarios.

- **Módulo de administración del laboratorio:** Permite gestionar las aulas, software y computadoras del laboratorio.
- **Módulo de reportes:** Facilita a los asistentes del laboratorio y administrador técnico ver los reportes estadísticos sobre el servicio de préstamo, y visualizar los registros de entrada y salida de usuarios. Si lo requiere el asistente o administrador podrá modificar los registros de entrada y salida.
- **Módulo de administración de usuarios:** El administrador técnico podrá realizar el CRUD (Create, read, update, delete) de los usuarios y asistentes del laboratorio. El asistente del laboratorio solo podrá crear nuevos usuarios.

En este trabajo de titulación se generará un producto final demostrable.

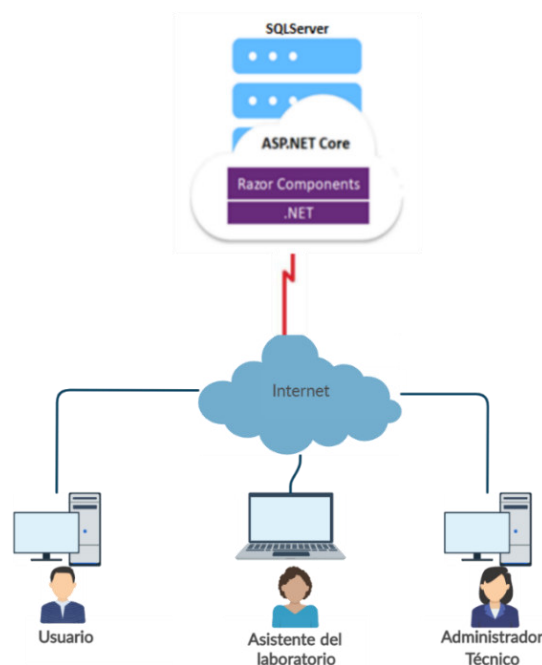


Figura 1.1 Esquema general del prototipo

1.3 MARCO TEÓRICO

A continuación, se presentan los fundamentos teóricos y las diferentes tecnologías necesarias para el desarrollo de este trabajo de titulación.

1.3.1 APLICACIONES WEB

Una aplicación web es un tipo especial de aplicación de tipo cliente servidor en la cual un usuario mediante un navegador realiza peticiones a una aplicación accesible a través de Internet o de una Intranet y que recibe una respuesta que se muestra en el navegador del usuario. En una aplicación web el cliente, el servidor web y el protocolo HTTP¹ mediante el que se comunican están estandarizados y no han de ser creados por el programador de la aplicación [3].

Las aplicaciones webs pueden ser diseñadas para una gran variedad de propósitos. Los usos más comunes de las aplicaciones webs incluye correo electrónico, tiendas online y servicios de almacenamiento [4]. Los elementos principales de una aplicación web se detallan en la tabla 1.1.

Tabla 1.1 Descripción de los elementos principales de una aplicación web

Cliente Web	<p>Un cliente web es un programa con el cual el usuario interactúa con la aplicación para solicitar al servidor web él envió de recursos que desea obtener. La parte del cliente web la mayoría de las veces está formada por etiquetas HTML y código ejecutable realizados en JavaScript [5].</p> <p>Las tecnologías que se suelen usar para programar el cliente web son: HTML, CSS, DHTML, lenguajes de scripts: JavaScript, VBScript, etc, ActiveX y pequeños programas en Java [6].</p>
Servidor Web	<p>Un servidor web es un programa que se ejecuta continuamente en un ordenador, manteniéndose a la espera de peticiones por parte de un cliente y que contestas a dichas peticiones de forma adecuada [7].</p> <p>Algunos de los lenguajes de programación más usados del lado del servidor son: Java, Python, PHP, Ruby, C#, C++ y Perl [8].</p>

¹ HTTP, de sus siglas en ingles "Hypertext transfer protocol" es un protocolo de la capa de aplicación que permite la transmisión de documentos hipermedia, como HTML.

1.3.2 ARQUITECTURA EN CAPAS

La arquitectura de software de múltiples capas es uno de los patrones arquitectónicos más populares en el desarrollo de software. Cada capa cubre una función y una responsabilidad en específico, lo cual permite administrar y mantener cada capa sin necesidad de alterar todo el programa. Esto simplifica enormemente la forma en que se administra la infraestructura del software. Este enfoque de varias capas resulta muy ventajoso para desarrollar aplicaciones alojadas en la nube, ya que los cambios que deban realizarse serán más simples y menos extensos [9].

La arquitectura de 3 capas se conforma por la capa de Presentación, la capa de Lógica de Negocio y la capa de Datos; tal como se muestra en la figura 1.2.



Figura 1.2 Arquitectura de 3 capas

1.3.2.1 Capa de presentación

La capa de presentación es la interfaz en la cual el usuario final interactúa con la aplicación. Esta capa suele ejecutarse en un navegador web, una aplicación de escritorio o una interfaz gráfica de usuario. La capa de presentación para navegadores webs generalmente se desarrolla utilizando HTML, CSS, Bootstrap y JavaScript [10]. Los lenguajes más comunes para el desarrollo de la capa de presentación se detallan en la tabla 1.2.

Tabla 1.2 Descripción de los lenguajes de la capa de presentación

HTML	De sus siglas en ingles “ <i>Hypertext Markup Language</i> ” es un lenguaje de etiqueta usado para crear páginas webs. “ <i>Hypertext</i> ” hace referencia a los hipervínculos que puede contener una página HTML. “ <i>Markup Language</i> ” hace referencia a la forma en que se utilizan las etiquetas para definir el diseño de página y los elementos dentro de la misma [11]. <p style="text-align: center;"><p> Este es un ejemplo de un párrafo en HTML. </p></p>
CSS	De sus siglas en ingles “ <i>Cascading Style Sheet</i> ” es utilizado para dar el aspecto y diseño de las páginas web. CSS ayuda a los

	desarrolladores web a crear una apariencia uniforme en varias páginas de un sitio web [12].
Bootstrap	Bootstrap es uno de los marcos de desarrollo web más populares que existen y se utiliza para desarrollar proyectos altamente receptivos en HTML, CSS y JavaScript. Con Bootstrap se reduce el tiempo para poner en marcha un sitio de web moderno. Bootstrap es mantenido por un equipo de desarrolladores en Github ² [13].
JavaScript	JavaScript es un lenguaje dinámico de programación liviano y se usa más comúnmente como parte de las páginas web, cuyas implementaciones permiten que los scripts del lado del cliente interactúen con el usuario y creen páginas dinámicas. El script debe estar incluido o referenciado en un documento HTML para que el código sea interpretado por el navegador [14].

1.3.2.2 Capa de lógica de negocio

Es la capa intermedia entre la capa de presentación y la capa de datos. Consiste en la lógica que realiza las funciones de la aplicación: procesamiento de datos, implementación de funciones, coordinación de varios usuarios y administración de recursos externos como datos o sistemas heredados [15]. Las tecnologías usadas en la capa de negocio para el desarrollo de este trabajo de titulación son C# y ADO.NET, las cuales se presentan en la tabla 1.3.

Tabla 1.3 Descripción de las tecnologías de la capa de lógica de negocio

C#	C# (pronunciado “ <i>See Sharp</i> ”) es un lenguaje de programación moderno, orientado a objetos y con seguridad de tipos, desarrollado y estandarizado por Microsoft. C# permite a los desarrolladores crear una gran variedad de aplicaciones seguras y sólidas que se ejecutan en el ecosistema .NET [16].
ADO.NET	ADO.NET es un conjunto de clases que brindan servicios de acceso a datos para programadores de .Net <i>Framework</i> ³ , ADO.NET soporta una gran variedad de necesidades de desarrollo, incluida la creación de clientes de base de datos [17].

² Github es un sistema de gestión de proyectos y control de versiones de código, así como una red social diseñada para desarrolladores.

³ Framework es una herramienta para desarrollo de software que permiten el desarrollo ágil de las mismas.

1.3.2.3 Capa de datos

Esta capa contiene las tablas, índices y datos administrados por la aplicación. Aquí se reciben las peticiones CRUD⁴ desde la capa de lógica de negocio [18].

Generalmente esta capa representa un sistema de gestión de base de datos relacionales como MySQL, MariaDB, Oracle, SQL Server, PostgreSQL, etc. Para este trabajo de titulación se usará Microsoft SQL Server Management Studio.

Microsoft SQL Server Management Studio (SSMS) es un entorno integrado de gestión de base de datos relacional desarrollado por la empresa Microsoft. SSMS ofrece un amplio grupo de herramientas gráficas con una serie de editores que permiten a los desarrolladores y administradores de base de datos acceder a información SQL⁵ [19].

1.3.3 FRAMEWORK BLAZOR

Blazor es un nuevo *framework* basado en C# en lugar de JavaScript que cambia la manera de realizar aplicaciones web con .NET, la palabra Blazor vienen de la combinación de dos términos que son Browser y Razor. Con Blazor se puede desarrollar componentes de interfaz de usuarios interactivas del lado del cliente utilizando C#, HTML y CSS; estos componentes son reutilizables y son la base de las aplicaciones Blazor [20].

Los proyectos Blazor se pueden ejecutar del lado del cliente directamente mediante Web Assembly o se lo puede ejecutar en el servidor con Blazor Server.

1.3.3.1 Componentes Razor

La mayoría de los archivos en los proyectos Blazor son archivos de extensión *.razor*. Razor es un lenguaje de plantillas basado en HTML y C# el cual es usado para generar una interfaz de usuario (UI) web de forma dinámica. Cada componente Razor es compilado dentro de una clase C# dentro de .NET cuando se crea el proyecto. Los componentes Razor tienen las siguientes características [21]:

- Definen la lógica de representación de la interfaz de usuario flexible.
- Controlan las acciones del usuario.
- Se pueden reutilizar y anidar.

⁴ CRUD, de sus siglas en ingles "Create Read Update Delete" es el acrónimo que se usa para referenciar a las funciones básicas en base de datos.

⁵ SQL, de sus siglas en ingles "Structured Query Language" es el lenguaje de programación estándar más utilizado para base de datos relacionales.

- Se pueden compartir como bibliotecas de clases Razor o paquetes NuGet⁶

En el código 1.1 se muestra un ejemplo sencillo de un componente Razor, en donde desde la línea 1 hasta la línea 7 se encuentra código HTML y las líneas que se encuentran dentro de `@code {...}` pertenecen al código C# del lado del servidor; cada vez que se dé clic al botón (línea 7) se incrementará el contador en uno (línea 14) y se mostrará en un párrafo HTML (línea 5).

```
1  @page "/counter"
2
3  <h1>Contador</h1>
4
5  <p>Cuenta actual: @numero</p>
6
7  <button class="btn btn-primary" @onclick="IncrementCount">Haz clic</button>
8
9  @code {
10     private int numero = 0;
11
12     private void IncrementCount()
13     {
14         numero++;
15     }
16 }
```

Código 1.1 Ejemplo de plantilla Razor.

1.3.3.2 Blazor WebAssembly

Blazor WebAssembly es un *framework* de aplicaciones de página única (SPA) para compilar aplicaciones web del lado del cliente con .Net. WebAssembly (wasm) es un formato de código de bytes optimizados para descargas rápidas y una velocidad de ejecución rápida, wasm se admite en exploradores web sin complementos. Cuando se ejecuta una aplicación Blazor WebAssembly en un explorador web [22]:

- Las respuestas HTTP se comprimen.
- Los archivos Razor y de código C# se compilan ensamblados de .NET.
- El entorno de ejecución .NET y los ensamblados se almacenan en cache en el explorador.
- Emplea el entorno .NET y lo configura para cargar los ensamblados de la aplicación.

⁶ Nuget es un administrador de paquetes diseñado para permitir a los desarrolladores compartir código reutilizable.

- El entorno de ejecución emplea la interoperabilidad de JavaScript para gestionar la manipulación de DOM⁷ y las llamadas API⁸ del explorador.

La arquitectura de Blazor WebAssembly se muestra en la figura 1.3.

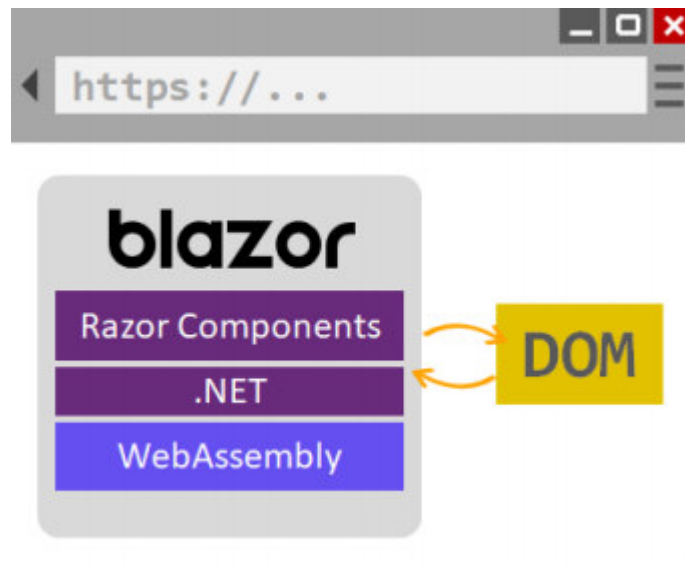


Figura 1.3 Arquitectura de Blazor WebAssembly [23]

1.3.3.3 Blazor Server

Blazor Server es un *framework* de SPA en donde las aplicaciones se ejecutan en el servidor desde una aplicación ASP.NET Core. Una aplicación Blazor Server se procesa previamente en respuesta a la primera solicitud del cliente, lo cual crea el estado de la interfaz de usuario en el servidor. Las actualizaciones de interfaz de usuario, el manejo de eventos y las llamadas de JavaScript se manejan a través de una conexión SignalR⁹. Algunos de los beneficios que ofrece Blazor Server son [24]:

- El tamaño de la descarga es mucho más pequeño que el de una aplicación Blazor WebAssembly, por lo que la aplicación Blazor Server carga más rápido.
- Se admiten clientes ligeros, por lo que las aplicaciones de Blazor Server funcionan en navegadores que no son compatibles con WebAssembly y en dispositivos con recursos limitados.

⁷ DOM, por sus siglas en inglés “Data Object Model” es una interfaz de programación para HTML y XML que da una representación completa orientada al objeto de la página web.

⁸ API es un conjunto de definiciones y protocolos que se utiliza para desarrollar e integrar el software de las aplicaciones.

⁹ SignalR es una biblioteca para ASP.NET que permite al servidor enviar notificaciones asincrónicas al cliente de la aplicación.

- El código C# de la aplicación, incluida los componentes de la aplicación no son ofrecidas al cliente.
- La aplicación aprovecha al máximo las capacidades del servidor, incluido el uso de cualquier API compatible con .NET Core.

La arquitectura de Blazor Server se muestra en la figura 1.4.

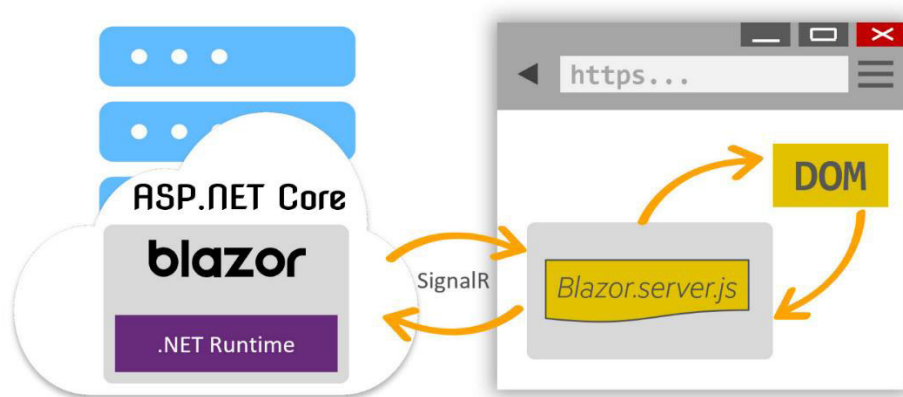


Figura 1.4 Arquitectura de Blazor Server [25]

La estructura de la plantilla de una aplicación Blazor Server se detalla en la tabla 1.4 [26].

Tabla 1.4 Descripción de los elementos principales de una aplicación Blazor Server

Data folder	En esta carpeta se tiene generalmente las clases que usará la aplicación Blazor Server.
Pages folder	Esta carpeta contiene los componentes (.razor) enrutables que componen la aplicación y la página raíz de Razor. La ruta de cada página se especifica mediante la directiva <i>@page</i> .
_Host.cshtml	Es la página raíz de la aplicación, cuando se solicita cualquier página de la aplicación, _Host.cshtml se representa y se devuelve en la respuesta.
Properties/launchSettings.json	Contiene la configuración del entorno de desarrollo.
Shared Folder	Aquí se tienen los componentes compartidos y las hojas de estilos. Los archivos que se encuentran

	en esta carpeta son accesibles a todos los elementos de la aplicación.
wwwroot	Contiene los activos estáticos públicos de la aplicación.
_Imports.razor	Aquí se incluye las directivas comunes de Razor para referenciarlos en los componentes de la aplicación, se usa <i>@using</i> para agregar los espacios de nombres (namespaces).
App.razor	El componente raíz de la aplicación que configura el enrutamiento del lado del cliente usando el <i>Routing component</i> .
Appsettings.json	Provee los ajustes de configuración para la aplicación.
Program.cs	Es el punto de entrada de la aplicación que configura el host ASP.NET Core.
Startup.cs	Contiene la lógica de inicio de la aplicación.

1.3.3.4 Syncfusion Blazor

Syncfusion Blazor es una biblioteca de componentes de interfaz de usuario empresarial moderna para crear aplicaciones Blazor WebAssembly o Blazor Server [27]. Syncfusion ofrece una licencia comunitaria para desarrolladores individuales y pequeñas empresas. En esta licencia están incluidos todos los productos disponibles en la versión Enterprise con soporte y actualizaciones incluidas [28].

La biblioteca de componentes Syncfusion Blazor contiene más de 65 controles de interfaz de usuarios de alto rendimiento y ligeros. Entre los componentes de Syncfusion Blazor están: cuadrículas, formularios, editores de textos, calendarios, horarios, botones, menú de navegaciones, gráficos, diagramas, entre otros [29].

El horario de Syncfusion Blazor (Blazor Scheduler) es un componente de calendario de eventos con funciones que ayudan a los usuarios a administrar sus eventos, citas y tiempo de manera eficiente. Este componente dispone de diferentes modos de visualización como la vista semanal, vista mensual y vista anual. Los usuarios pueden establecer el rango de horas de trabajos, así como los días laborables [30]. Un ejemplo del componente Blazor Scheduler se muestra en la figura 1.5.

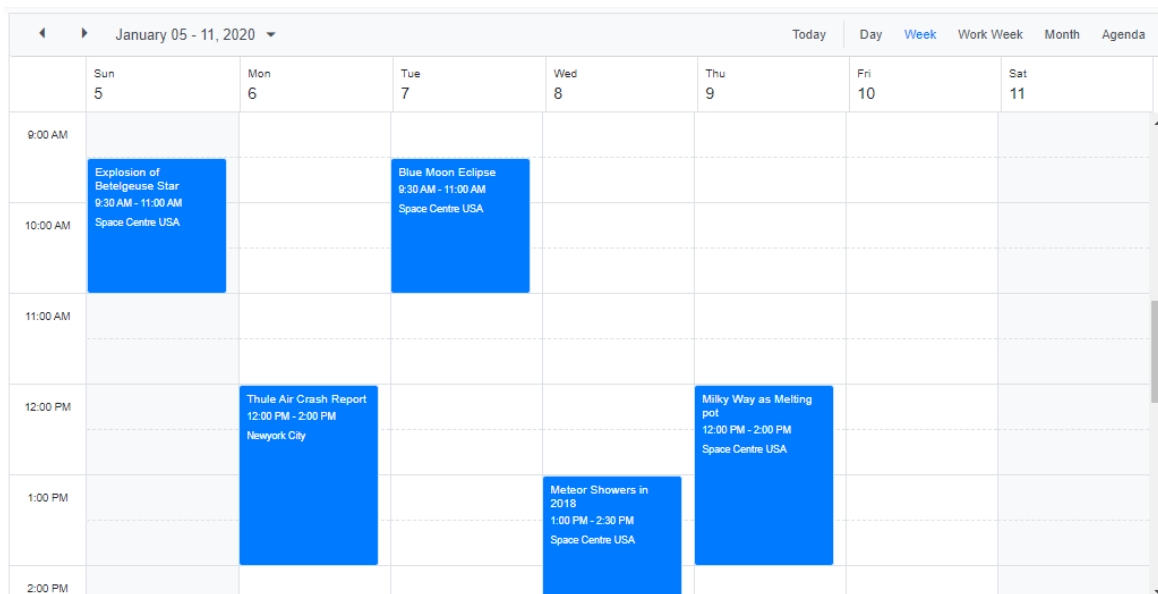


Figura 1.5 Ejemplo del componente Blazor Scheduler [31]

Otro componente importante es Blazor Charts el cual está diseñado para visualizar datos. Contiene una gran variedad de interfaces de usuario con más de 30 cuadros y gráficos, cada tipo de grafico se configura fácilmente y permite crear efectos visuales muy llamativos. Además, su alto rendimiento ayuda a procesar grandes cantidades de datos rápidamente y da la opción de exportar los gráficos a documentos PDF o a imágenes con formato SVG, PNG y JPEG [32]. Un ejemplo de un gráfico de columnas con Blazor Charts se muestra en la figura 1.6.

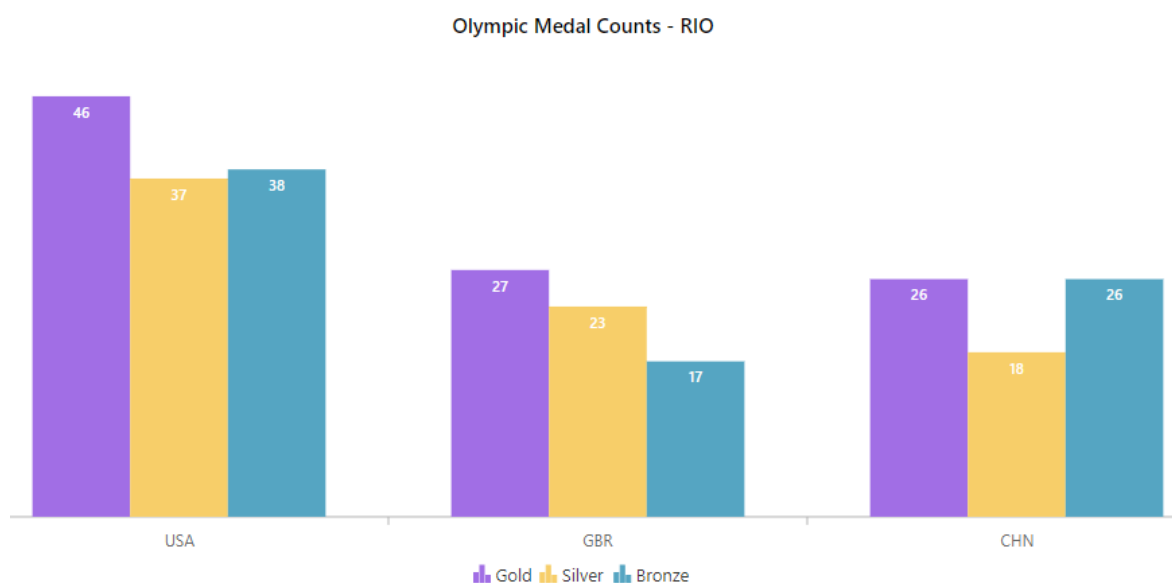


Figura 1.6 Ejemplo del componente Blazor Charts [33]

1.3.4 MICROSOFT AZURE

Microsoft Azure es el nombre comercial para los servicios de computación en la nube de Microsoft [34]. La plataforma de Azure está compuesta por más de 200 productos y servicios que permiten crear, ejecutar y administrar aplicaciones en la nube, con las herramientas y *framework* que el usuario desee [35].

Microsoft Azure App Service es una completa plataforma para crear, desarrollar y escalar aplicaciones web. Se puede trabajar con .NET, .NET Core, Node.js, Java, Python o PHP, también en contenedores o en ejecución en Windows o Linux. Con Microsoft Azure App Service se satisfacen exigentes requisitos de rendimiento empresarial, seguridad y cumplimiento normativo usando una plataforma de confianza que puede controlar más de 40000 millones de solicitudes al día [36].

Otro servicio importante de Microsoft Azure es Azure SQL Database, el cual es el servicio de base de datos relacional, inteligente y escalable creado para la nube. Con Azure SQL Database se elimina la complejidad que conllevan la configuración y administración de la alta disponibilidad, las copias de seguridad, los ajustes y otras tareas de la base de datos; lo cual permite a los desarrolladores centrarse en el desarrollo de las aplicaciones. Además, las capas de protección, los controles integrados y el detector de amenazas mantienen los datos seguros [37].

2. METODOLOGÍA

En este capítulo se detallan las diferentes etapas para el diseño e implementación de la capa de presentación, capa de negocio y la capa de datos que conforman el prototipo de aplicación web para la reserva de aulas y computadores del Laboratorio de Informática FIEE-EPN. Para el desarrollo del prototipo de aplicación web se usará la metodología de desarrollo de software Kanban.

2.1. DISEÑO

Aquí se definirán los requerimientos funcionales y no funcionales y se mostrará el diseño de las diferentes capas que conforman el prototipo de aplicación web. Se presentarán los diferentes diagramas necesarios para definir y abstraer los elementos de la aplicación web.

2.1.1 TABLERO KANBAN

Se presentan la lista de tareas por hacer, en progreso y realizadas al inicio de la fase adquisición de requerimientos y de diseño, dicha lista de tareas se muestra en la tabla 2.1.

Tabla 2.1 Tablero Kanban con las actividades al inicio de la fase de diseño

Por hacer	En progreso	Realizadas
Elaboración del diagrama de contexto.	Entrevista al personal y jefe del laboratorio	Estudio de las herramientas tecnológicas necesarias para el desarrollo del prototipo.
Elaboración del diagrama de casos de uso.	Entrevista a un grupo de estudiantes de la FIEE-EPN.	
Elaboración del diagrama de actividad.	Recopilación de datos necesarios para realizar el préstamo de las aulas y computadoras del laboratorio.	
Diseño de los sketches.		
Elaboración del diagrama de clases.		
Diseño de la base de datos.		
Instalar las herramientas tecnológicas necesarias para el desarrollo del prototipo.		
Codificar de la base de datos SQL Server.		
Codificar las vistas necesarias para la interfaz de usuario.		
Codificar las clases para las diferentes capas de nuestra aplicación.		

Por hacer	En progreso	Realizadas
Codificar la lógica de funcionamiento para los diferentes módulos del programa.		
Alojar la aplicación en el servicio de computación en la nube Microsoft Azure.		
Realizar las pruebas de cada funcionalidad del prototipo mediante pruebas de usuario.		
Realizar pruebas de validación de las funcionalidades.		
Corregir errores detectados.		

2.1.2 DEFINICIÓN DE REQUERIMIENTOS

Para definir los requerimientos del prototipo se realizaron entrevistas al jefe actual del laboratorio, a un grupo de estudiantes y a tres asistentes del Laboratorio de Informática. Hay que destacar que debido al confinamiento por el Covid-19 se optó por recopilar los requerimientos mediante preguntas cerradas y abiertas enviadas a los respectivos correos de los entrevistados. El detalle de las entrevistas se encuentra en el Anexo A y en el Anexo B.

Las entrevistas tienen como objetivo obtener información referente a:

1. El proceso para que a un usuario se le pueda prestar un computador o aula.
2. La documentación que deberá presentar un usuario para los diferentes tipos de reservaciones.
3. Los usuarios prioritarios al momento de hacer una reservación.
4. La información necesaria para realizar reportes estadísticos.
5. Las preguntas necesarias para que un usuario pueda realizar una calificación del servicio prestado.
6. Las funcionalidades específicas que los diferentes usuarios deseen en el sistema de reserva.

En base a la información entregada en las entrevistas se definieron los requerimientos de la aplicación. En la tabla 2.2 se muestran los requerimientos funcionales del prototipo.

Tabla 2.2 Requerimientos funcionales del prototipo

ID	Título	Descripción
RF001	Registro	El usuario que quiera usar el sistema deberá ingresar su número de cédula, con esto el sistema validará que el usuario pertenezca a la EPN y le permitirá ingresar otros datos como el número telefónico, luego se le enviará a su correo la contraseña con la que podrá ingresar al sistema.
RF002	Autenticación	Para ingresar al sistema el usuario deberá ingresar su número de cédula y contraseña.
RF003	Actualización de datos personales	El usuario podrá modificar su contraseña o número telefónico. Los otros datos del usuario no serán modificables.
RF004	Reservar computador	El usuario escogerá que computador reservar en base al horario disponible de préstamo y los programas que requiera. El usuario deberá indicar la fecha con la hora de inicio y de fin de su reserva. Al confirmar su reservación el sistema le enviará a su correo los detalles de su reserva.
RF005	Reservar aula	El usuario deberá escoger el aula que desea reservar indicando, la fecha, el motivo de la reserva, y las horas que usará el aula. Deberá adjuntar una documentación en la que se indique la autorización del uso del aula ya sea por las autoridades de la FIEE o por el jefe del laboratorio. El asistente del laboratorio en base a la documentación presentada por el usuario le aprobará o le negará la reservación del aula, en ambos casos se le notificará al usuario mediante correo.
RF006	Mis reservaciones	El usuario podrá ver, modificar o eliminar sus reservaciones. Hay que destacar que las reservaciones de aulas hechas por los usuarios no serán modificables solo podrán ser eliminadas.
RF007	Calificar servicio	El usuario podrá realizar una calificación de experiencia indicando que tan satisfecho quedó con el servicio de préstamo. Esta evaluación al servicio prestado se basará en varias preguntas cerradas y un campo para que el usuario deje recomendaciones o comentarios.
RF008	Reservación para clases o eventos	El asistente del laboratorio o el administrador técnico podrá hacer reservaciones de las aulas para los horarios de clases de cada semestre o algún evento en particular. En el caso de reservaciones para clases podrán repetirse diaria o semanalmente de acuerdo con las fechas que indique el asistente o administrador.

ID	Título	Descripción
RF009	Gestionar reservaciones	El asistente del laboratorio o el administrador técnico podrá modificar o eliminar sus reservaciones o las reservaciones de computadoras echas por los usuarios. Para la reserva de aulas hechas por los usuarios el administrador técnico podrá eliminarlas, el asistente del laboratorio solo podrá verlas. Se les notificará a los usuarios a los cuales se le hizo algún cambio en sus reservaciones a través de correo electrónico.
RF010	Gestionar computadoras	El asistente del laboratorio o el administrador técnico podrá crear, ver, actualizar o eliminar computadoras del laboratorio.
RF011	Gestionar aulas	El asistente del laboratorio o el administrador técnico podrá crear, ver, actualizar o eliminar las aulas del laboratorio. Además, de agregar o quitar software en las distintas aulas del laboratorio.
RF012	Ingresar registro de entrada y salida	El asistente del laboratorio podrá registrar si el usuario uso o no las computadoras y aulas, el nombre del usuario quien uso el laboratorio, el aula y la hora de entrada y de salida.
RF013	Visualización de registro de entrada y salida	El asistente del laboratorio o el administrador técnico podrá visualizar el registro de entrada y salida de usuarios en la fecha que desee. Además de eliminar o editar algún registro que considere este incorrecto.
RF014	Reportes estadísticos	El asistente del laboratorio o el administrador técnico podrá generar un reporte estadístico en base a información como: los programas que el usuario usó, número de mayor concurrencia de usuarios, etc.
RF015	Gestión de usuarios del sistema	El administrador técnico podrá crear, editar, eliminar o visualizar todos los usuarios y asistentes del sistema de préstamos. El asistente del laboratorio solo podrá crear y visualizar usuarios.

Los requerimientos no funcionales se muestran en la tabla 2.3.

Tabla 2.3 Requerimientos no funcionales del prototipo

ID	Título	Descripción
RNF001	Presentación	La interfaz debe ser amigable e intuitiva para los usuarios.
RNF002	Autenticación	El ingreso al sistema estará restringido por un usuario y contraseña
RNF003	Permiso	El acceso a los diferentes módulos del sistema estará definido por el rol de cada usuario

ID	Título	Descripción
RNF004	Datos	Se usará Microsoft SQL Server para almacenar la información del aplicativo.
RNF005	Acceso a través de la web	El prototipo debe ser accesible a través de internet para lo cual se la alojará en la plataforma de computación en la nube Microsoft Azure.

2.1.2.1 Módulos del prototipo

Los módulos del prototipo se definen mediante la agrupación de los requerimientos funcionales. La tabla 2.4 muestra el detalle de los módulos y su relación con los requerimientos funcionales.

Tabla 2.4 Módulos del prototipo

Módulo	Descripción	ID
Módulo de registro	Permite a los usuarios que deseen usar el sistema de préstamo registrarse, deberán ingresar su número de cédula y el sistema verificará que dicho usuario pertenezca a la EPN, una vez hecha la verificación se le enviará al correo del usuario una contraseña para que pueda ingresar al sistema, dicha contraseña podrá ser cambiada por el usuario.	RF001
Módulo de autenticación	Permite a los usuarios, asistentes del laboratorio y administrador técnico ingresar a la aplicación web mediante su cédula y contraseña, y actualizar datos como la contraseña y número de celular.	RF002 RF003
Módulo de reservaciones	Permite a los usuarios realizar reservaciones de computadoras o de aulas del laboratorio. El usuario podrá ver, modificar o eliminar sus reservaciones. Hay que destacar que las reservaciones de aulas hechas por los usuarios no serán modificables solo podrán ser eliminadas. Además, da la opción a los asistentes del laboratorio y administrador técnico hacer reservaciones de las aulas para los horarios de clases de cada semestre o algún evento en particular, y eliminar o actualizar las reservaciones de computadoras hechas por los usuarios.	RF004 RF005 RF006 RF007 RF008 RF012
Módulo de administración del laboratorio	Permite gestionar las aulas, software y computadoras del laboratorio.	RF009 RF010 RF011
Módulo de reportes	Facilita a los asistentes del laboratorio y administrador técnico ver los reportes estadísticos sobre el servicio de préstamo, y visualizar los registros de entrada y salida de usuarios. Además, el asistente o administrador creará los registros de entrada y salida.	RF013 RF014
Módulo de administración de usuarios	El administrador técnico podrá realizar el CRUD de los usuarios y asistentes del laboratorio. El asistente del laboratorio solo podrá crear nuevos usuarios.	RF015

2.1.2.2 Diagramas de casos de uso

Para cada módulo del prototipo se mostrará el diagrama de caso de uso correspondiente para mostrar las interacciones del usuario con el sistema. En el diagrama de caso de uso para el módulo registro que se muestra en la figura 2.1, en este diagrama se observan las acciones necesarias para que el usuario se registre en la aplicación web.

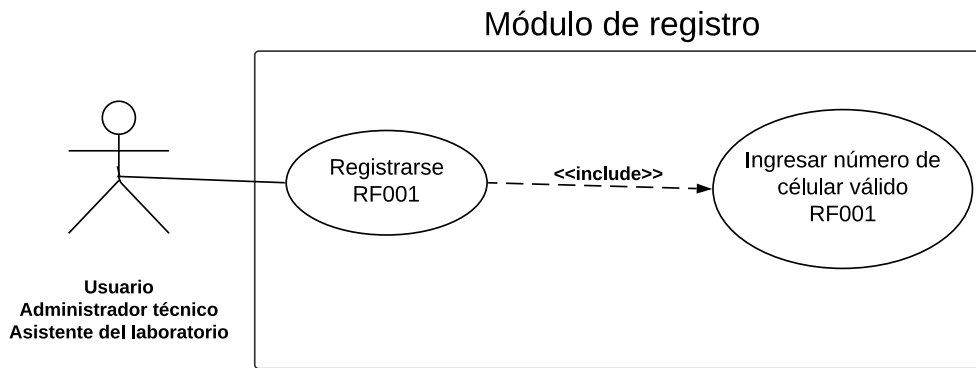


Figura 2.1 Diagrama de casos de uso del módulo de registro

La figura 2.2 muestra el diagrama de uso del módulo de autenticación, en donde el usuario, administrador y asistente pueden ingresar al sistema y pueden cambiar su contraseña o número telefónico.

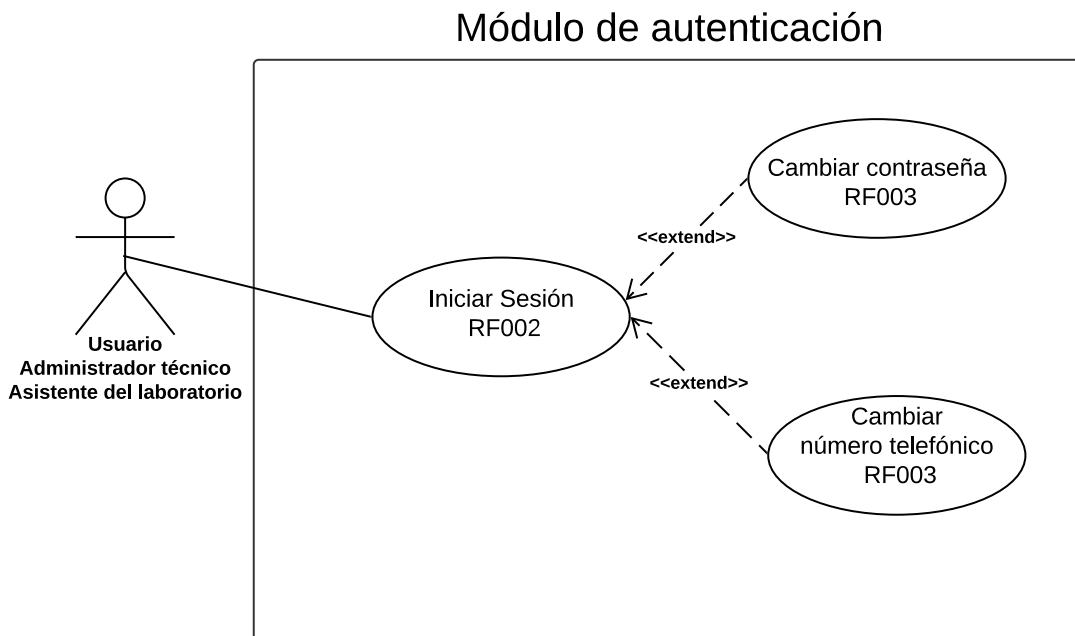


Figura 2.2 Diagrama de casos de uso del módulo de autenticación

El diagrama de casos de uso del módulo reservaciones se muestra en la figura 2.3.

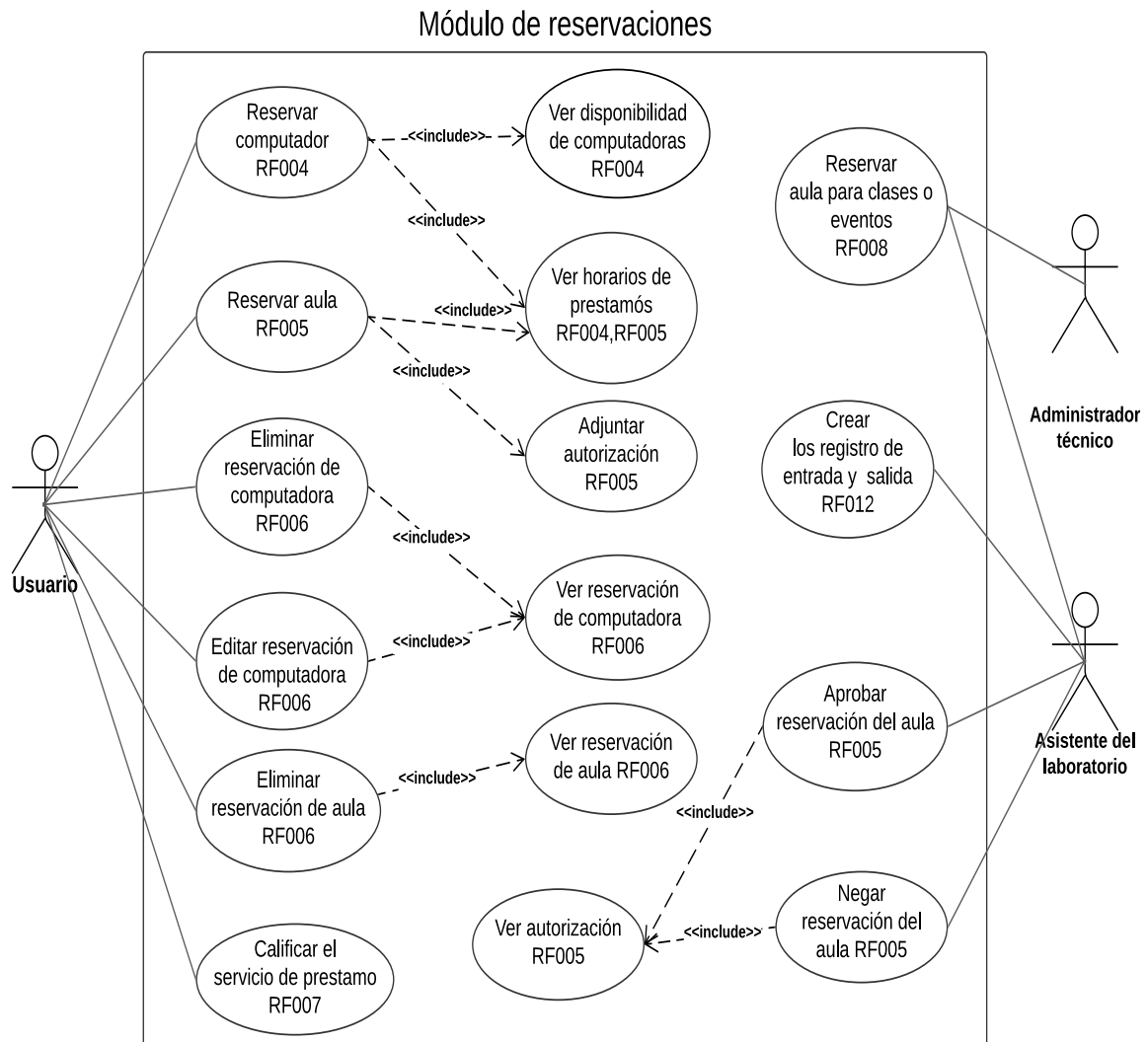


Figura 2.3 Diagrama de casos de uso del módulo de reservaciones

En la figura 2.4 se muestra el diagrama de casos de uso del módulo de administración del laboratorio.

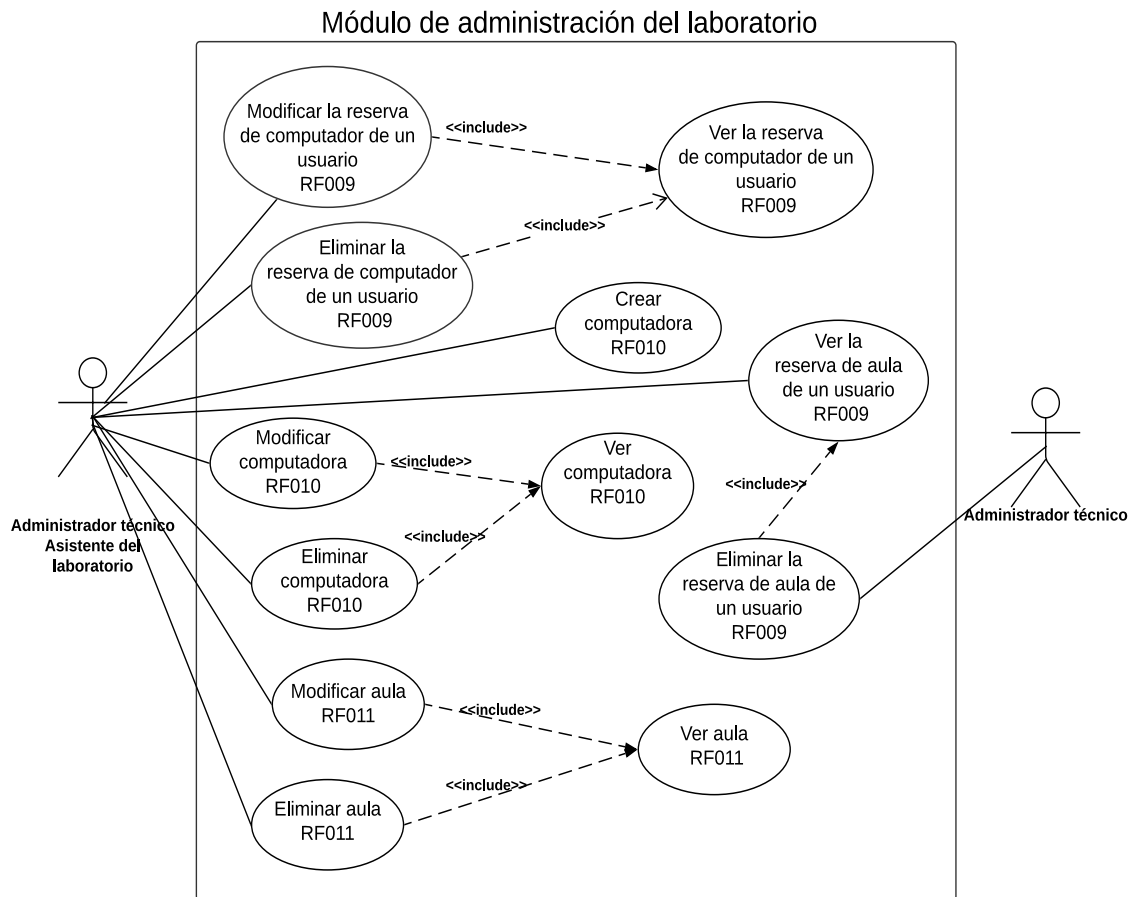


Figura 2.4 Diagrama de casos de uso del módulo administración del laboratorio

El diagrama de casos de uso del módulo de reportes se muestra en la figura 2.5.

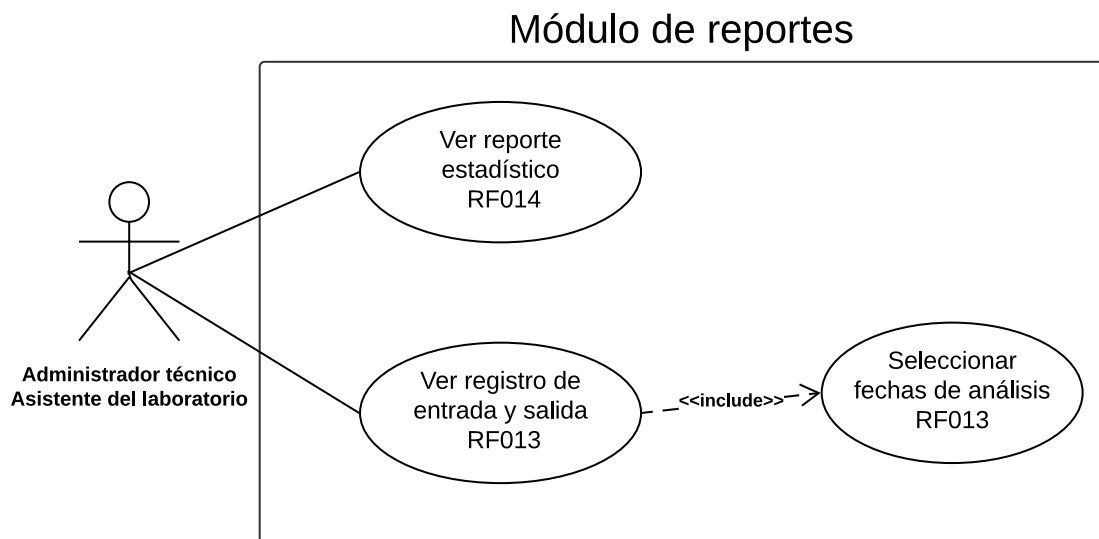


Figura 2.5 Diagrama de casos de uso del módulo de reportes

La figura 2.6 muestra el diagrama de casos de uso del módulo de administración de usuarios.

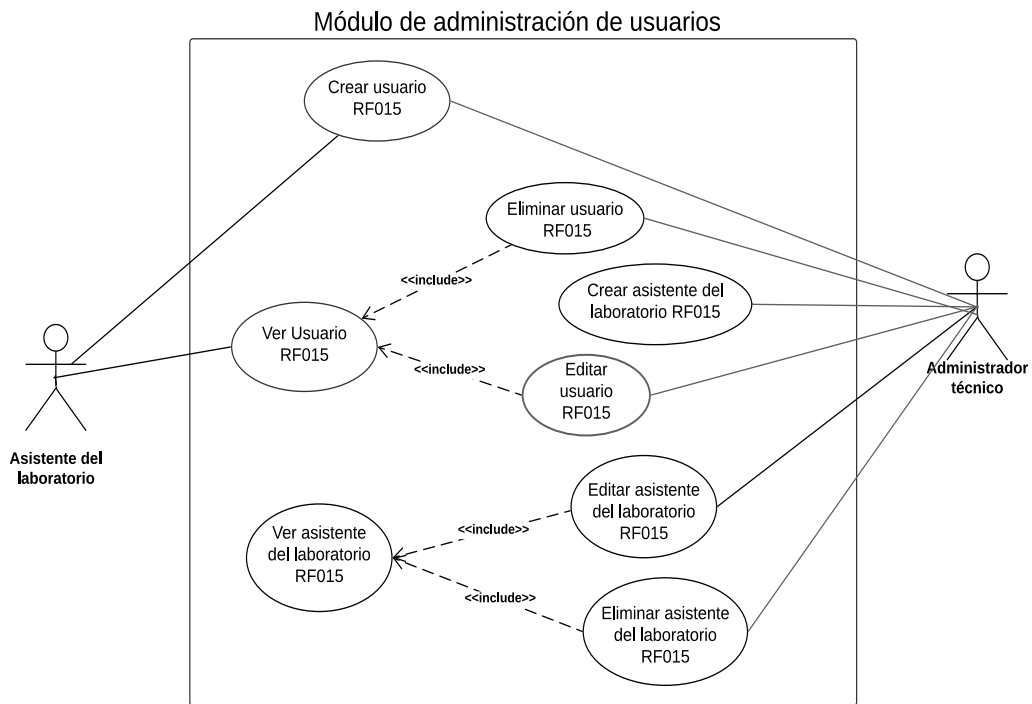


Figura 2.6 Diagrama de casos de uso de administración de usuarios

2.1.2.3 Diagramas de contexto

A continuación, en la figura 2.7 se muestra el diagrama de contexto del aplicativo de reserva en el cual se abstraen las interacciones existentes entre los agentes externos y el sistema.



Figura 2.7 Diagrama de contexto del sistema de reserva

2.1.2.4 Diagramas de actividades

El diagrama de actividad de cada módulo del prototipo de aplicación web mostrará el flujo de trabajo para realizar las diferentes actividades del sistema de reserva. La figura 2.8 muestra el diagrama de actividades correspondiente al módulo de registro en el que se encuentra el requerimiento RF001.

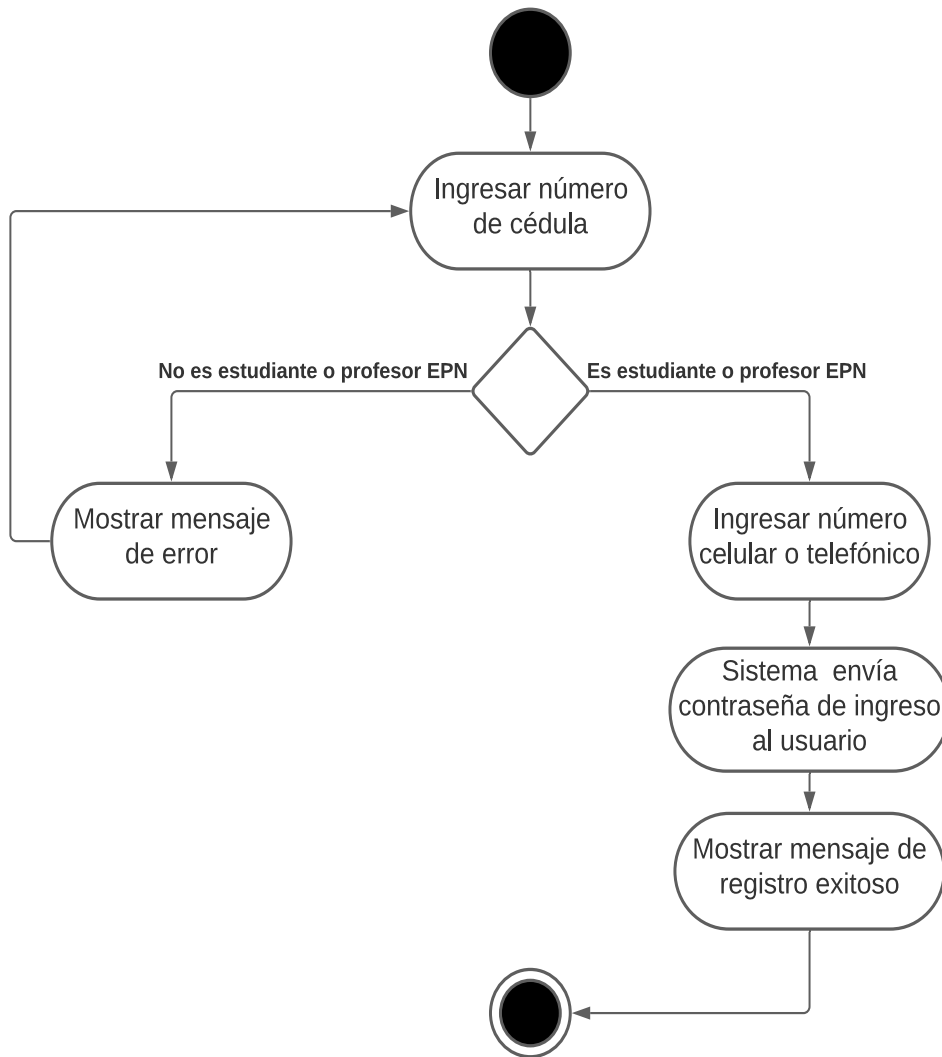


Figura 2.8 Diagrama de actividades del módulo de registro

La figura 2.9 muestra el flujo de actividades para que un usuario puede autenticarse en la aplicación web, estas actividades están relacionadas con los requerimientos RF002 y RF003.

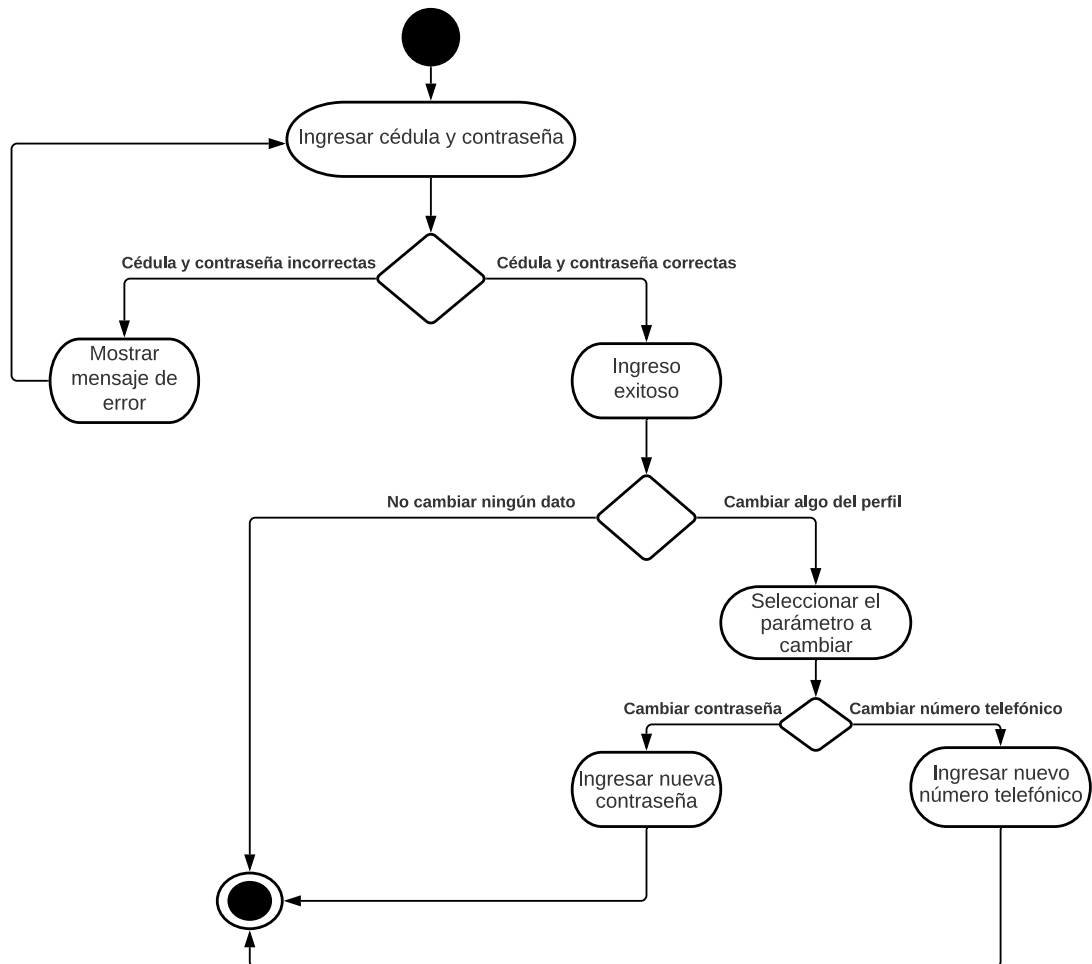


Figura 2.9 Diagrama de actividades del módulo de autenticación

El diagrama correspondiente a la creación de nuevas reservaciones (requerimientos RF004, RF005, RF006, RF008) del módulo de reservaciones se muestra en la figura 2.10.

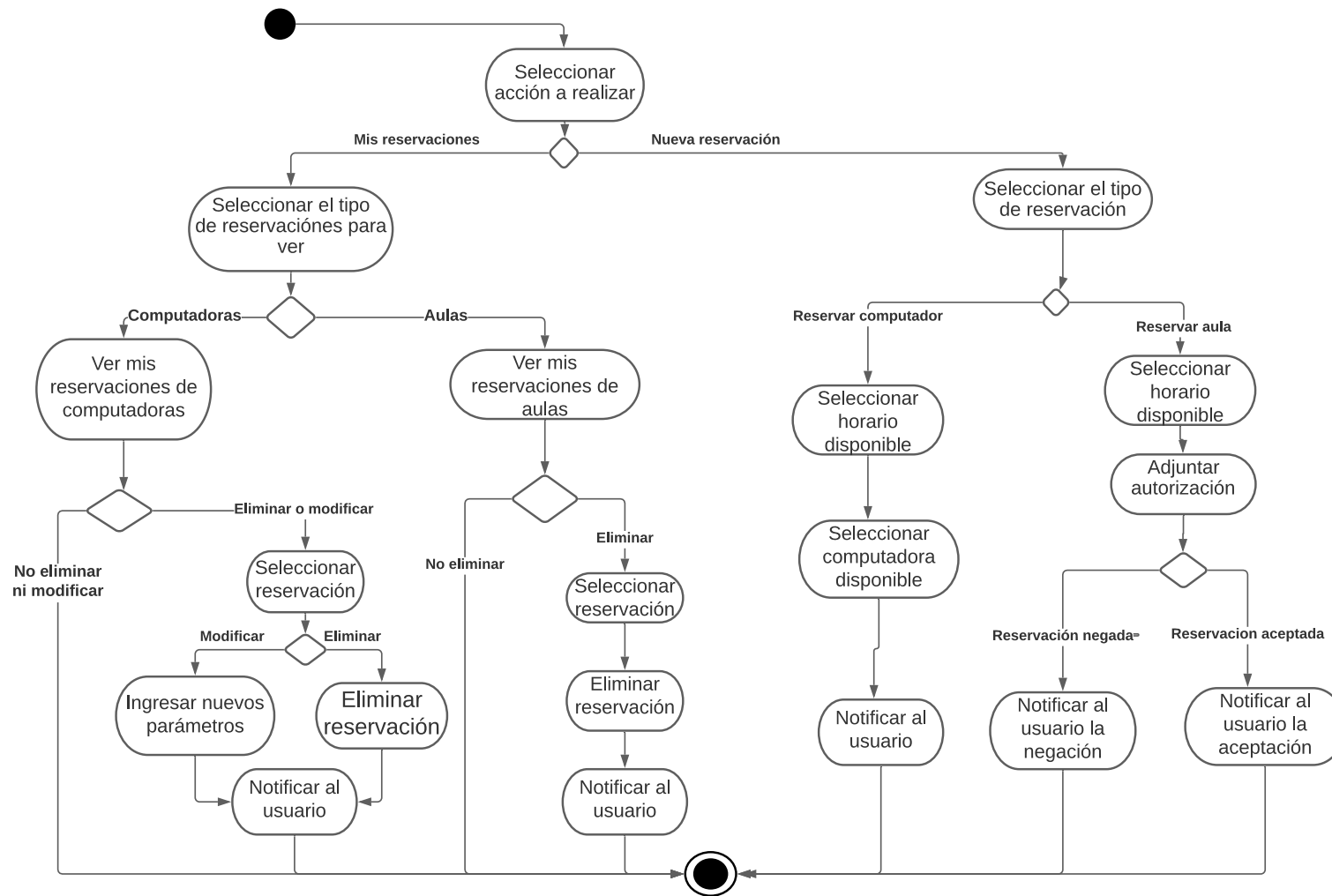


Figura 2.10 Diagrama de actividades para crear reserva

Las actividades para que el asistente del laboratorio apruebe o niegue la reserva de aula de algún usuario (requerimiento RF005) están reflejadas en el diagrama de la figura 2.11.

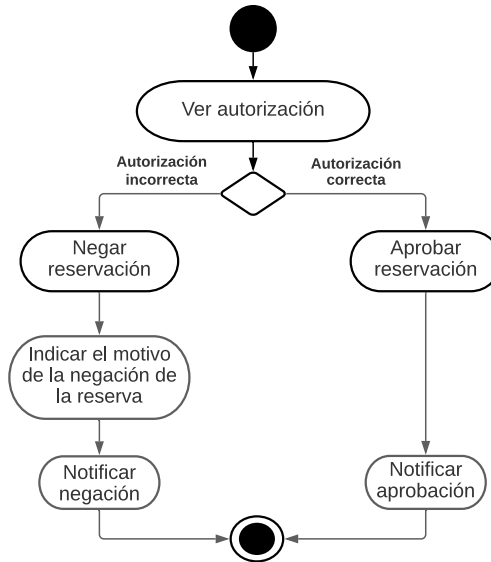


Figura 2.11 Diagrama de actividades para aprobar o negar reservación

El diagrama de actividades para calificar el servicio de préstamo, el cual corresponde al requerimiento RF007, se muestra en la figura 2.12.

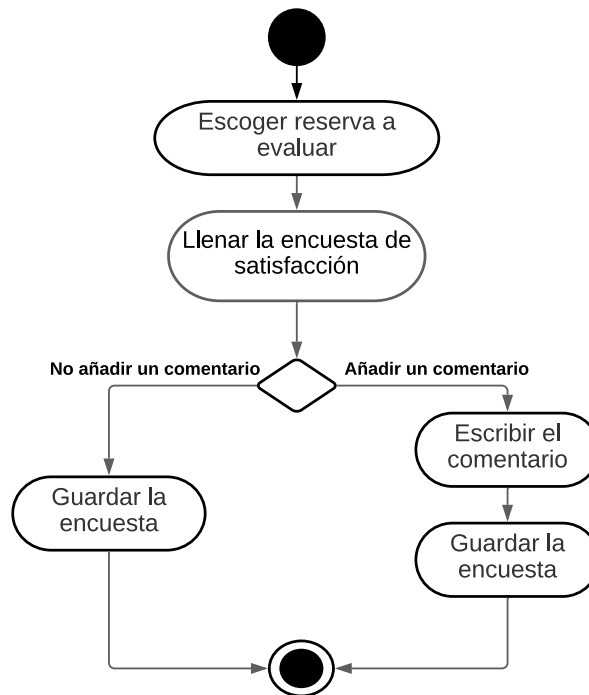


Figura 2.12 Diagrama de actividades para calificar el servicio de préstamo

Las actividades ilustradas en la figura 2.13 muestra las actividades para ingresar el registro de entrada y salida de las reservaciones confirmadas, en caso de que al usuario se le registre la asistencia este recibirá una notificación recordándole que debe llenar la encuesta de satisfacción (requerimiento RF012).

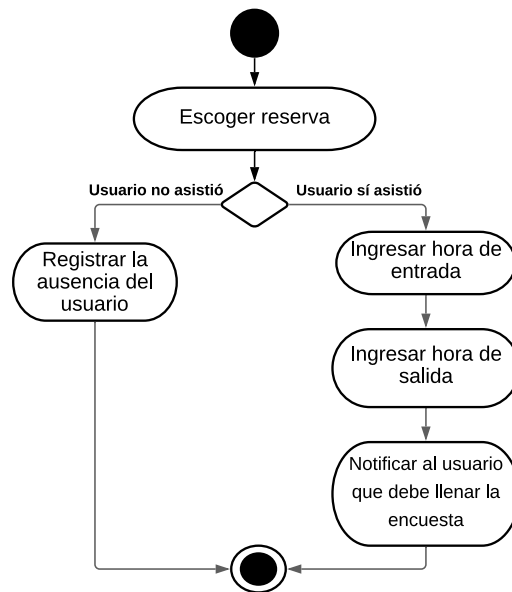


Figura 2.13 Diagrama de actividades del registro de entrada y salida

En la figura 2.14 se muestra el diagrama de actividades del módulo de administración del laboratorio, estas actividades tienen relación con los requerimientos RF009, RF010 y RF011

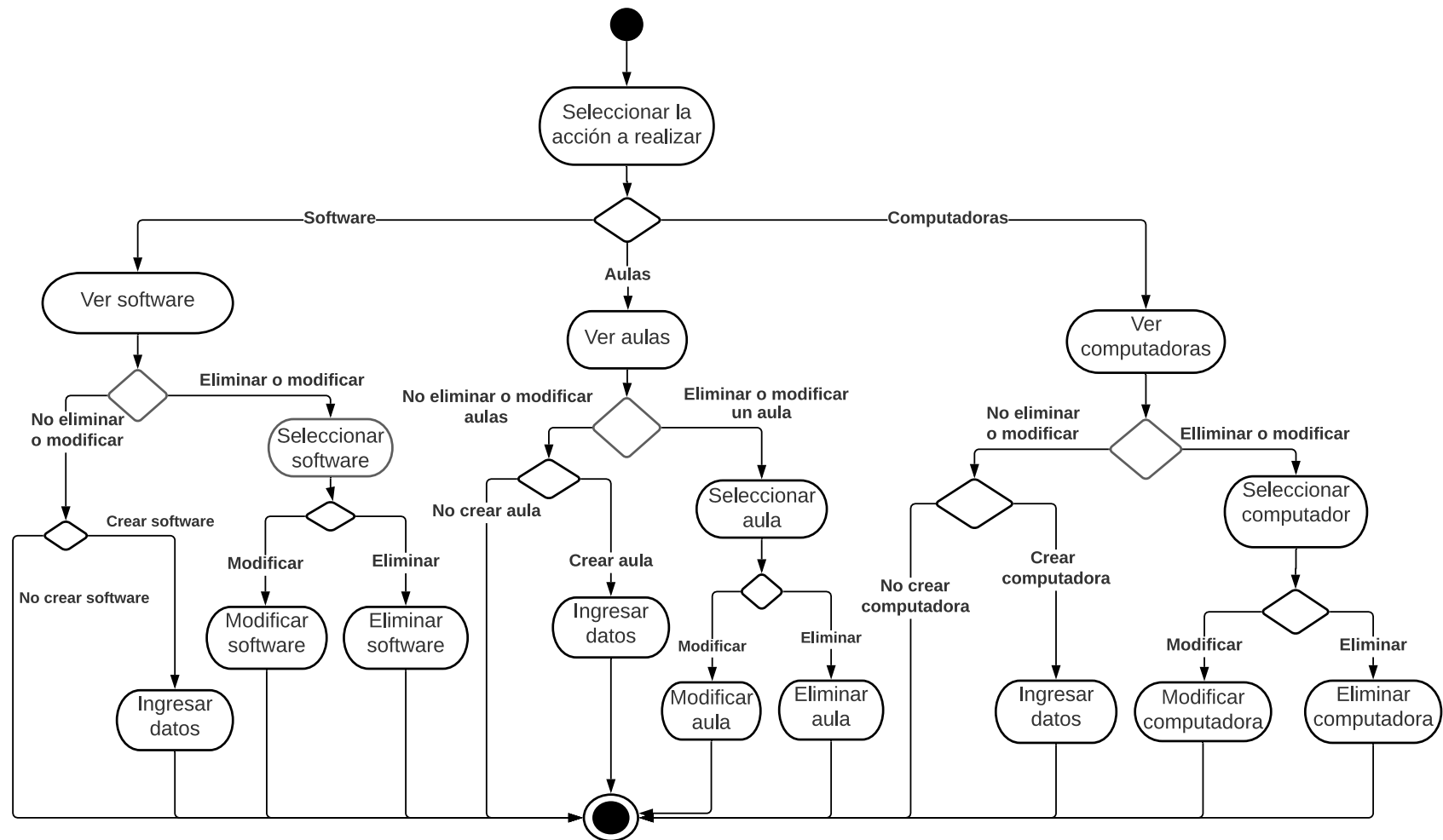


Figura 2.14 Diagrama de actividades del módulo de administración del laboratorio

El diagrama de actividades para el módulo de reportes, el cual está relacionado con los requerimientos RF013 y RF014, se muestra en la figura 2.15.

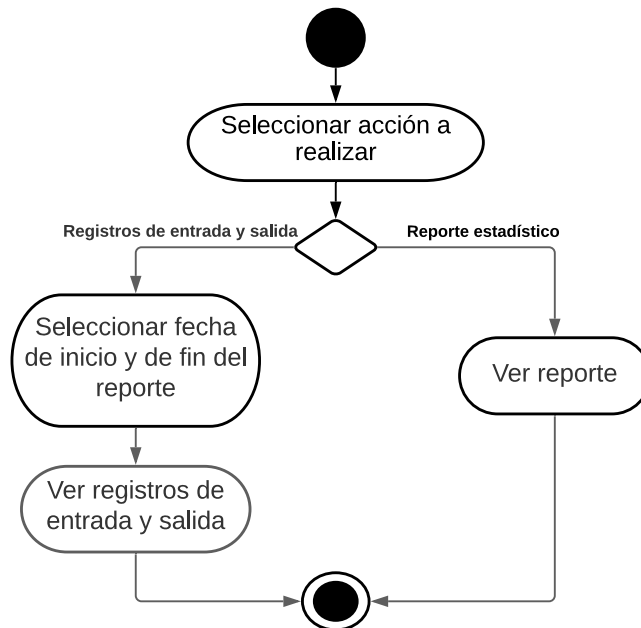


Figura 2.15 Diagrama de actividades del módulo de reportes

Para el módulo de administración de usuarios (requerimiento RF015) el diagrama de actividades se muestra en la figura 2.16.

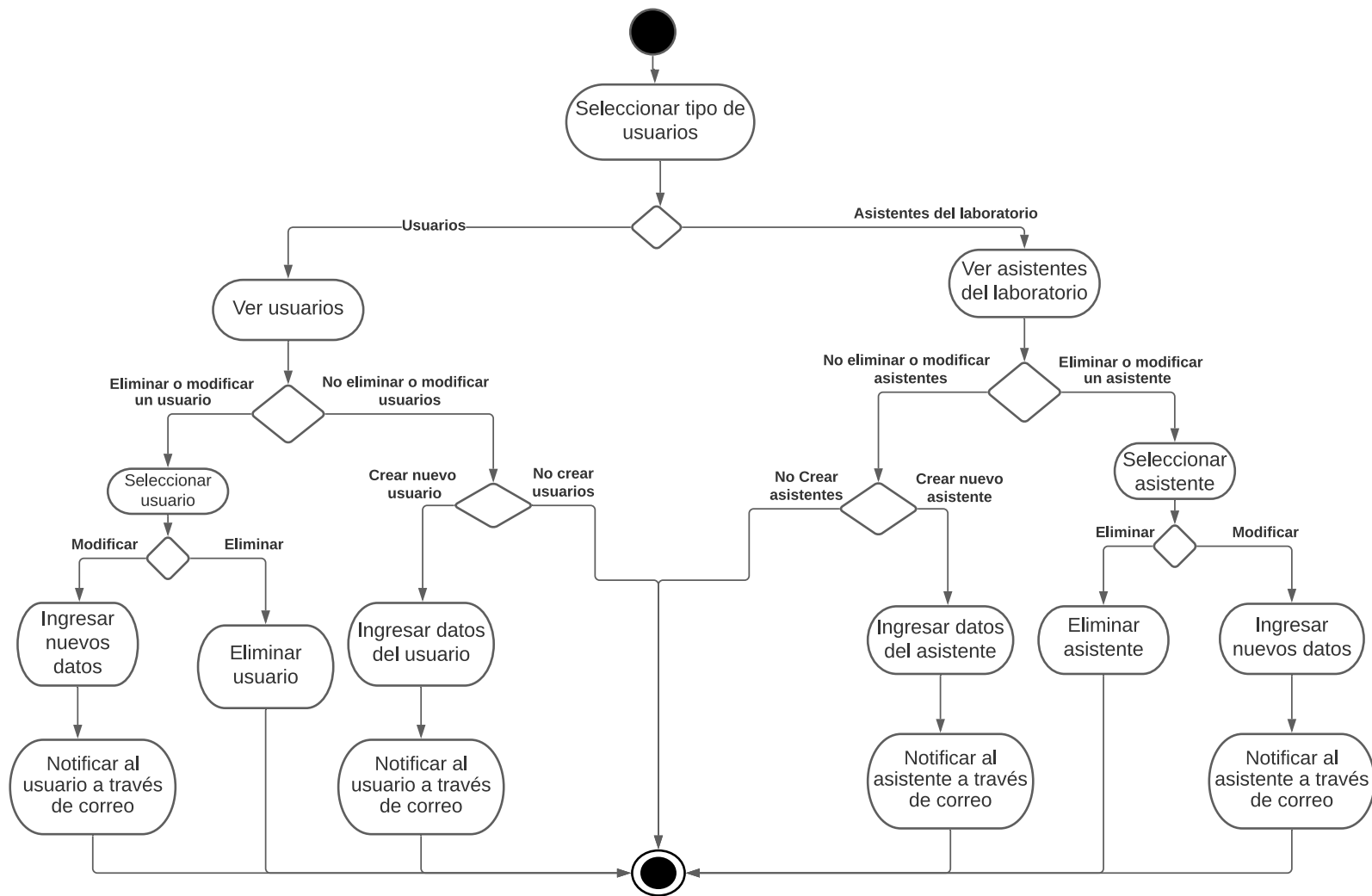


Figura 2.16 Diagrama de actividades de administración de usuarios

2.1.2.5 Diagramas de estados

Para la reserva de computadoras y aulas se presentan los diagramas de estados correspondientes para describir su comportamiento. La figura 2.17 muestra el diagrama de estados para la reserva de computadoras; el usuario deberá escoger un horario disponible, el cual es verificado por el sistema, y la reserva quedará inicialmente confirmada.

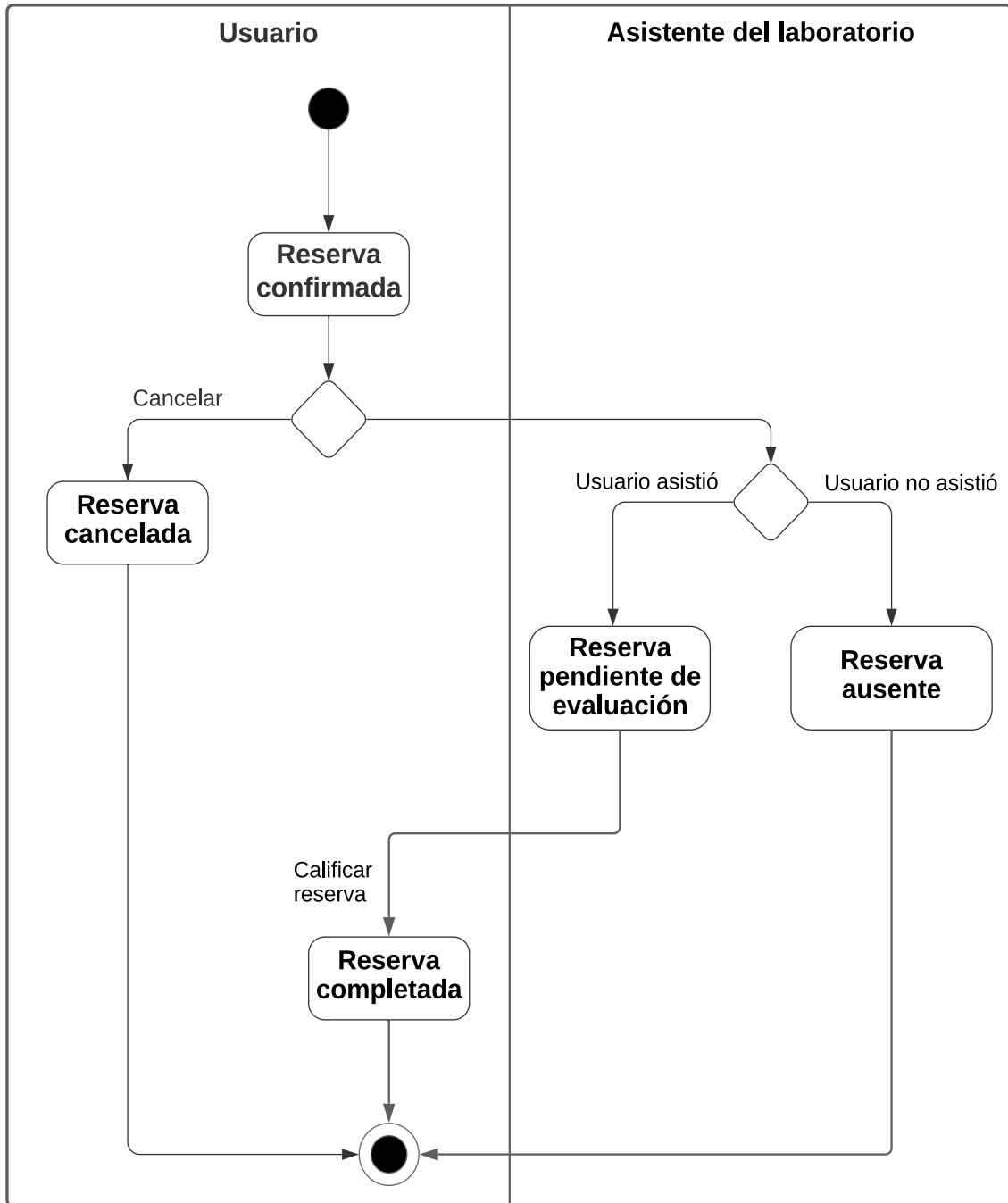


Figura 2.17 Diagrama de estados para la reserva de computadoras

Para la reserva de aulas el usuario deberá escoger un horario disponible (el cual es verificado por el sistema) y adjuntar una autorización y la reserva inicialmente quedará pendiente de ser aprobada o negada. Esta transición de estados para este tipo de reservas se muestra en la figura 2.18.

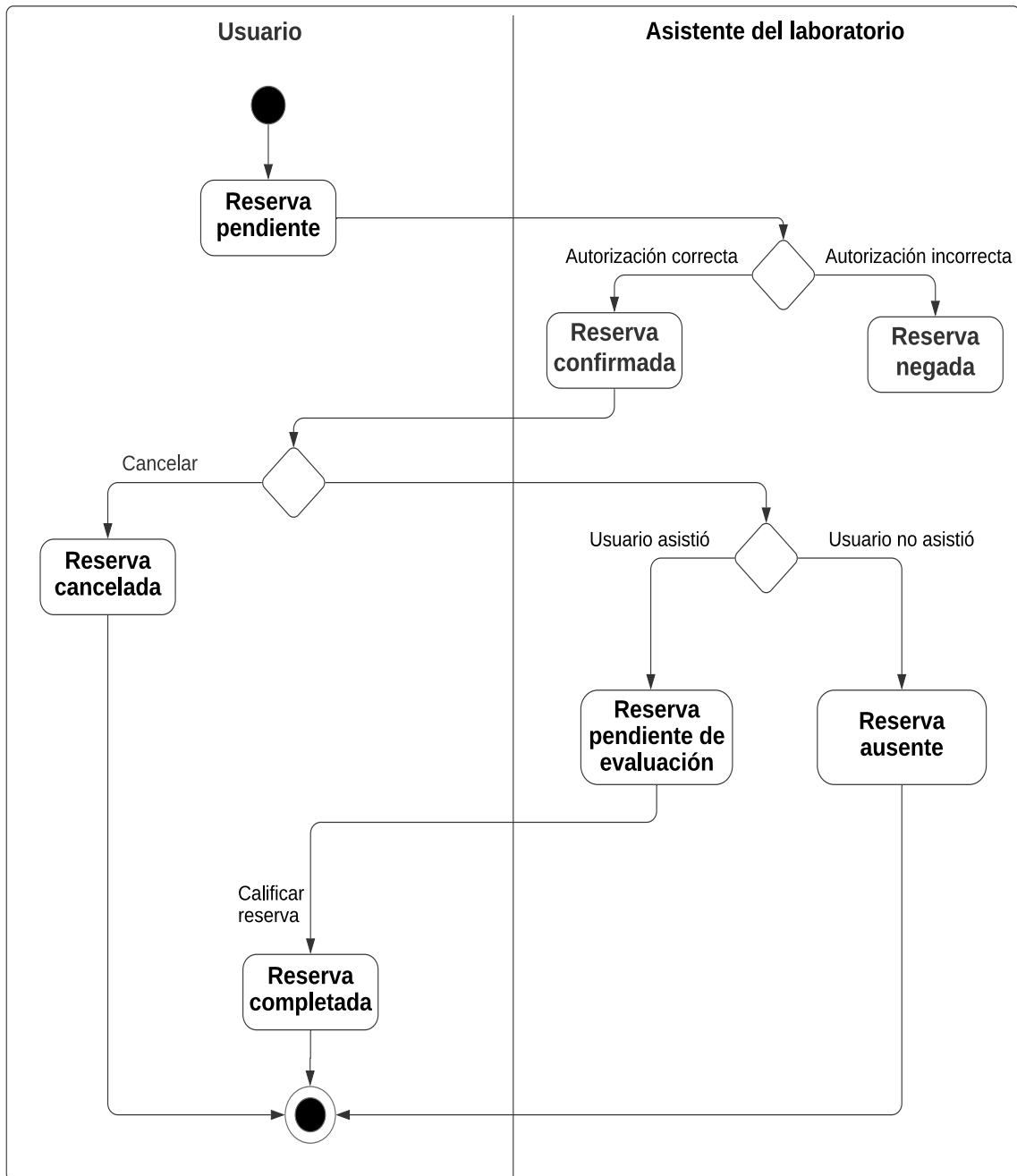


Figura 2.18 Diagrama de estados para la reserva de aulas

2.1.3 DISEÑO CAPA DE DATOS

El diseño de la capa de datos se realizó en base a la información otorgada por el personal del laboratorio de Informática; la misma incluye: las características de las computadoras, las materias que se dictan en las diferentes aulas y los programas instalados en las computadoras de las distintas aulas del laboratorio. El detalle de esta información se encuentra en el Anexo C.

Una vez obtenido los datos necesarios se procedió a estructurar el diagrama relacional de la base de datos, como se muestra en la figura 2.19.

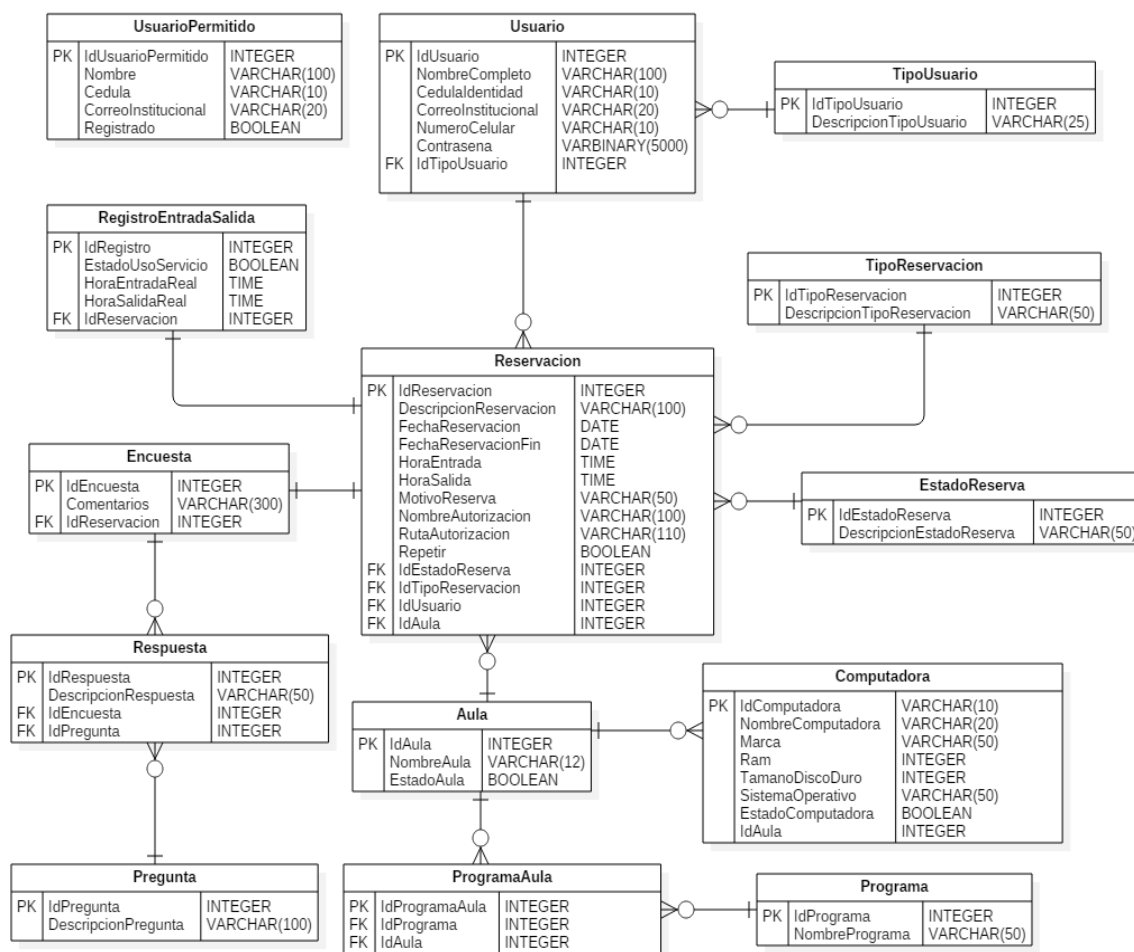


Figura 2.19 Diagrama relacional de la base de datos

2.1.4 DISEÑO CAPA LÓGICA DE NEGOCIO

La capa de lógica de negocio está representada mediante un diagrama de clases, dichas clases servirán para intercambiar información entre la capa de presentación y la capa de datos. En la figura 2.20 se muestra el diagrama de clases utilizado para desarrollar la aplicación web, en el diagrama se visualizan los métodos y atributos de las clases utilizadas.

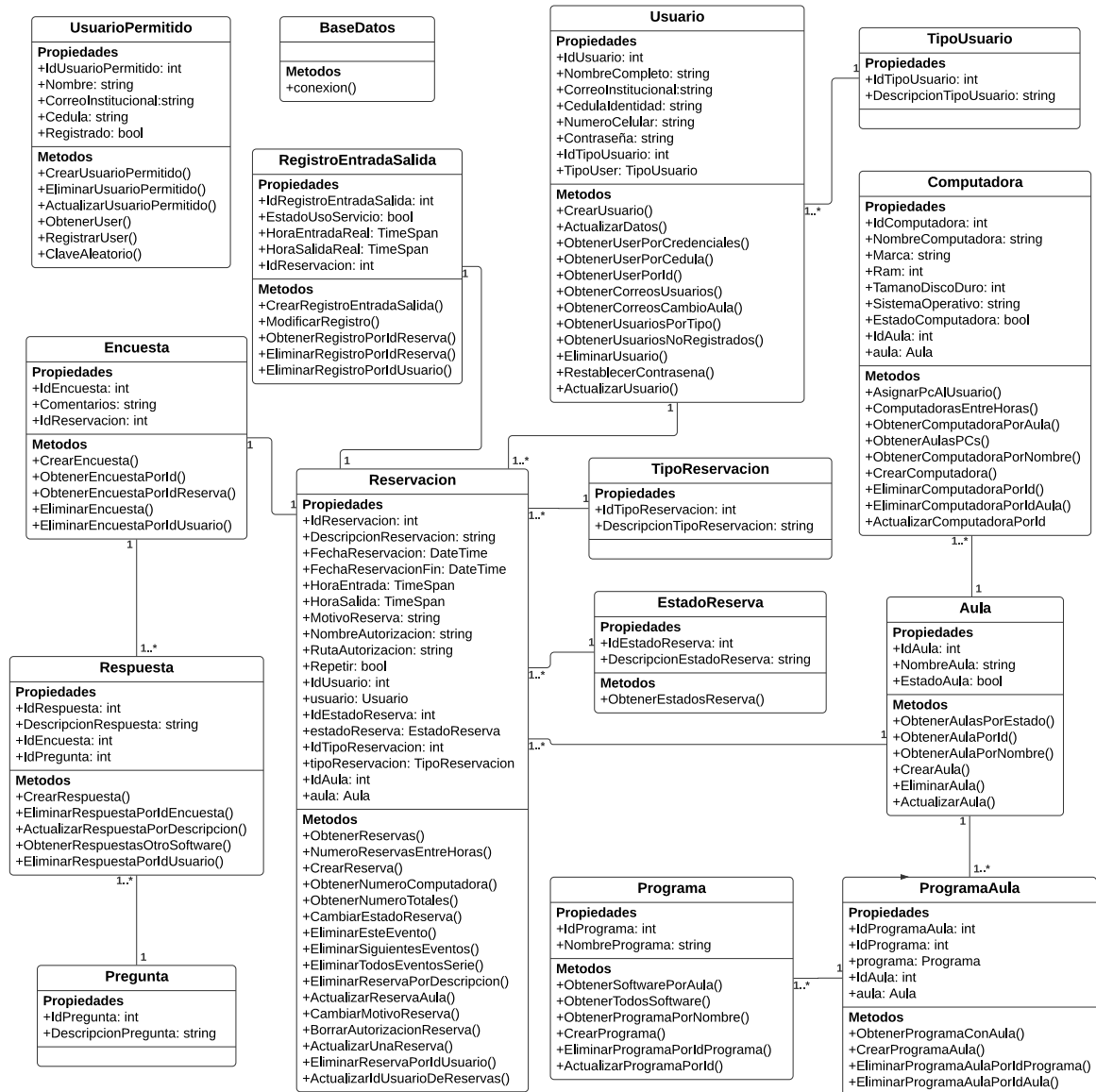


Figura 2.20 Diagrama de clases

2.1.5 DISEÑO CAPA DE PRESENTACIÓN

En este apartado se muestra los sketches que servirán como base para el diseño de las interfaces gráficas correspondiente a cada elemento de la aplicación web y los mapas de navegación para cada rol. El sketch correspondiente a la página de inicio de la aplicación se observa en la figura 2.21, en esta se muestran las opciones de registro e inicio de sesión.

RESERVAS FIEE-EPN

https://

Sistema de reserva del laboratorio de informática FIEE-EPN

USUARIO
cédula

CONTRASEÑA
Su contraseña

Ingresar

¿Va a usar este sistema por primera vez?

Regístrate

Figura 2.21 Sketch de inicio de sesión, requerimiento RF002

Los usuarios nuevos que deseen registrarse en la aplicación web deberán hacerlo en la página de registro, en esta tendrán que ingresar su cédula para cargar sus datos y registrarse, como se puede observar en la figura 2.22.

Registrarse

Datos del usuario

Número de cédula * Cargar datos

Apellidos y nombres *

Correo electrónico * Número celular

Si hay inconsistencia en los datos comuníquese con laboratorio.informatica@epn.edu.ec 2976300 ext 2331

Al registrarse usted recibirá un correo con su usuario y contraseña

Registrarse Volver al inicio

Figura 2.22 Sketch de registro de usuarios, requerimiento RF001

Una vez que el usuario haya iniciado sesión se le presentará un menú lateral con las opciones 'Reservar computador', 'Reservar aula', 'Mis reservaciones' y 'Calificar servicio'; en la parte superior se muestra el rol (etiqueta 1), el nombre del usuario que inicio sesión (etiqueta 2) y en la parte superior derecha se encuentra las opciones de perfil de usuario para actualizar datos y cerrar sesión (etiqueta 3); tal como se muestra en la figura 2.23.

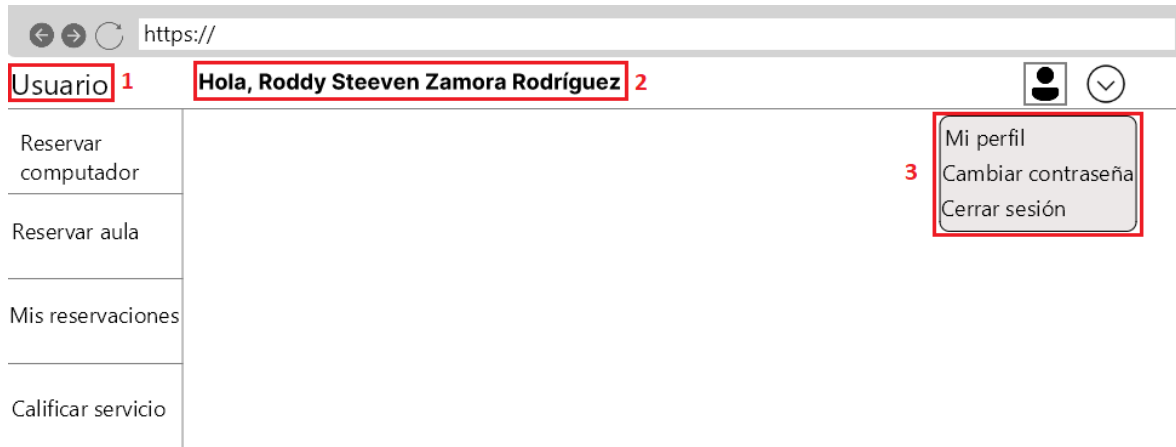


Figura 2.23 Sketch del menú de usuario, requerimiento RF003

En el sketch correspondiente a la opción de reservar computador del menú usuario (figura 2.24) se tiene un horario con las horas correspondientes al préstamo de aulas y computadoras, además de una lista desplegable para cambiar de aula (etiqueta 1), un botón para ver el software que dispone las computadoras (etiqueta 2) y un botón para realizar reservas de computadoras (etiqueta 3).

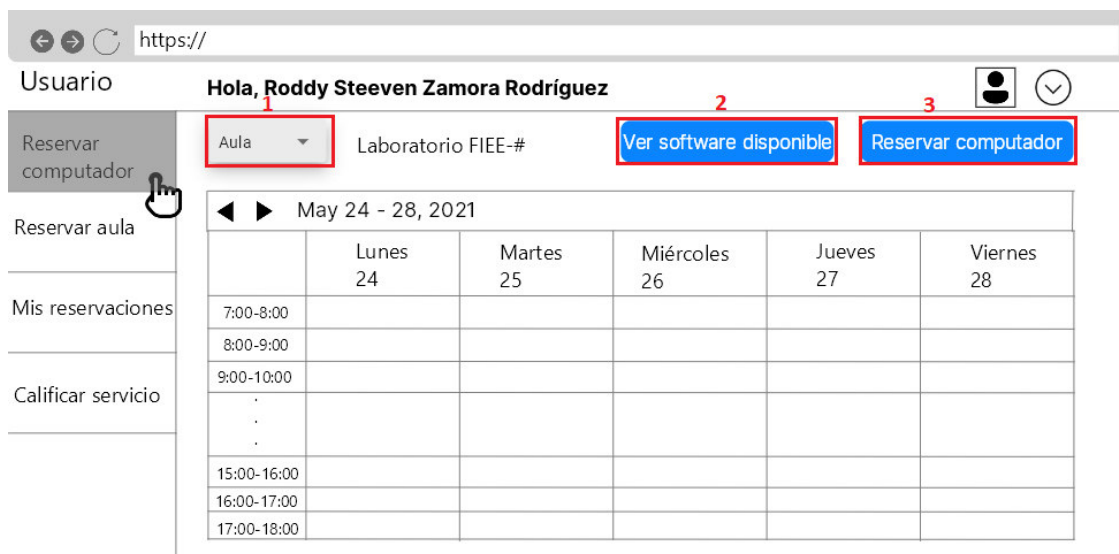


Figura 2.24 Sketch de reserva de computador del menú de usuario, requerimiento RF004

Para que el usuario reserve un aula se presentará algo muy similar a la vista correspondiente para reservar computadoras, como se puede apreciar en la figura 2.25.

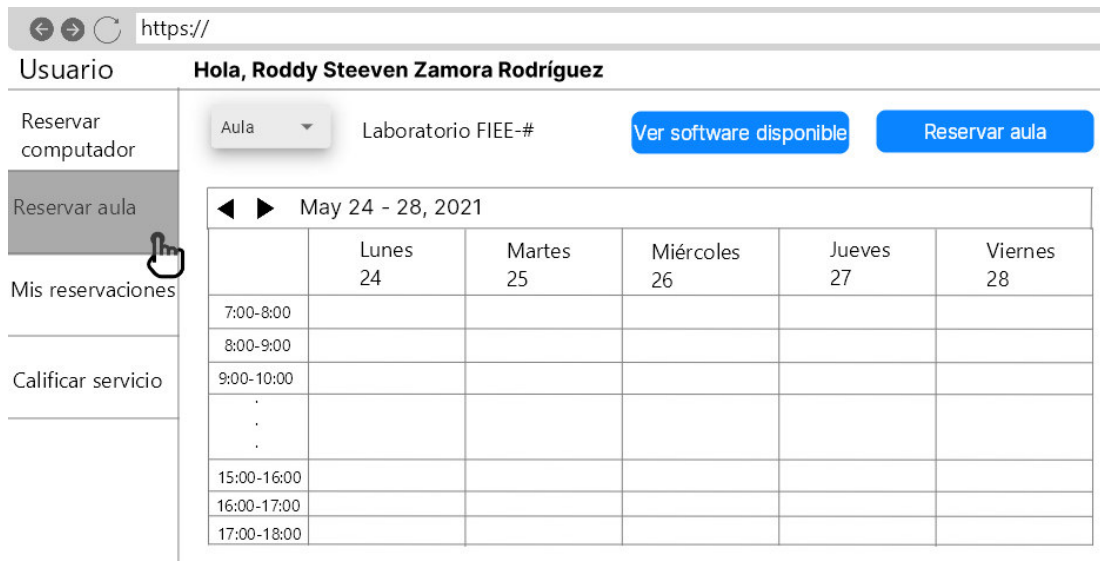


Figura 2.25 Sketch de reserva de aula del menú de usuario, requerimiento RF005

Como se puede observar en la figura 2.26 el usuario podrá ver las reservaciones de computadoras o aulas dependiendo de la opción que escoja en la lista desplegable llamada Tipo (etiqueta 2), se podrá cambiar de aula a través de la lista desplegable Aula (etiqueta 1), se podrán ver atributos de las reservas como la fecha, hora de entrada y hora de salida; y dependiendo del estado de las reservas estarán habilitados botones para actualizar o eliminar las mismas.

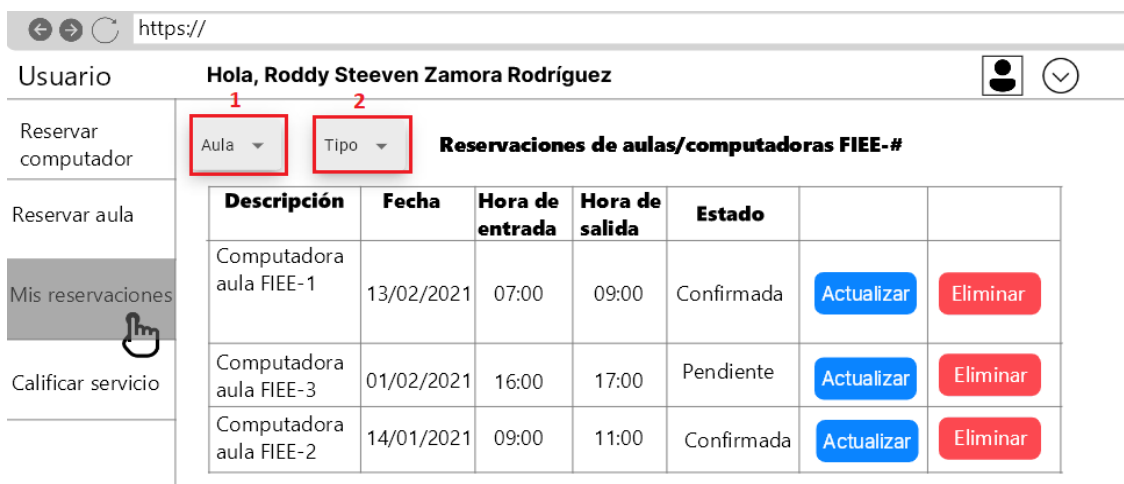


Figura 2.26 Sketch que muestra las reservaciones del usuario, requerimiento RF006

El usuario tendrá la opción de calificar el servicio de préstamo, dicha calificación se hará a través de una encuesta basada en preguntas cerradas y una sección para sugerencias, como se muestra en la figura 2.27.

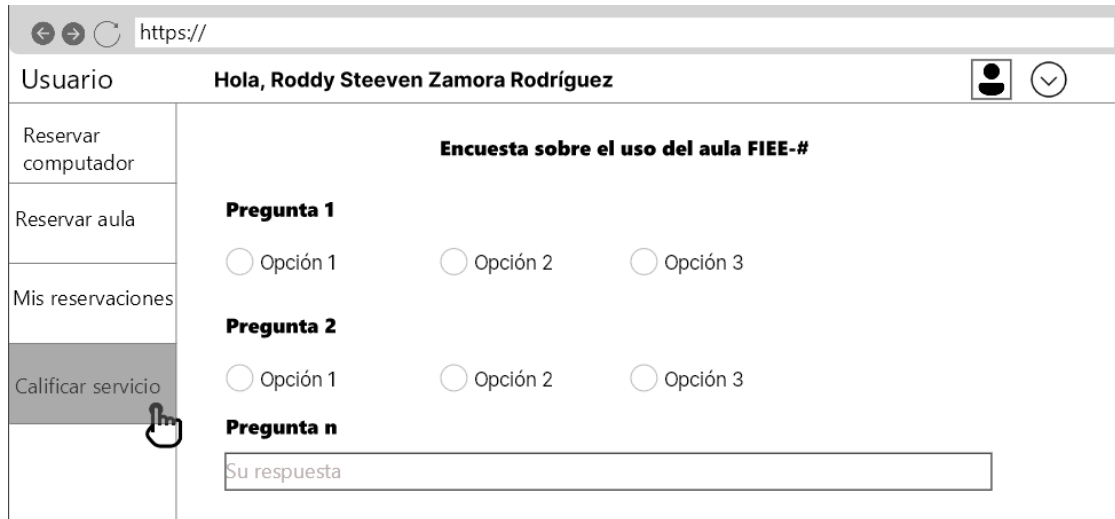


Figura 2.27 Sketch de calificar servicio del menú de usuario, requerimiento RF007

El asistente del laboratorio dispondrá de su propio menú de opciones con el que puede interactuar con la aplicación web, como se aprecia en la figura 2.28.



Figura 2.28 Sketch del menú del asistente del laboratorio

La opción “Peticiónes para reservar aulas” muestra una lista con todas las peticiónes hechas por los usuarios para reservar aulas en una fecha y hora específica, el asistente dependiendo de la documentación presentada podrá confirmar o negar las reservas de

aulas hechas por los usuarios; en la figura 2.29 se puede ver el sketch correspondiente a esta opción.

Asistente **Hola, Gabriel Alfonso Rodríguez Paladines**

Peticiónes para reservar aulas

Peticiónes para reservar aulas

Nombre de usuario	Fecha	Hora de entrada	Hora de salida	Autorización	Confirmar	Negar
María Alejandra	13/02/2021	07:00	09:00	Ver autorización	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Juan Jose	01/02/2021	16:00	17:00	Ver autorización	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Gestionar reservaciones

Gestionar laboratorio

Gestionar usuarios

Registros de entrada y salida

Reportes estadísticos

Comentarios y sugerencias

Figura 2.29 Sketch de las peticiones para reservar aulas, requerimiento RF005

El sketch correspondiente a la gestión de reservaciones se muestra en la figura 2.30, aquí se ven un horario, una lista desplegable para cambiar de aula (etiqueta 1) y un botón para que el asistente pueda hacer reservaciones (etiqueta 2).

Asistente **Hola, Gabriel Alfonso Rodríguez Paladines**

Peticiónes para reservar aulas

Reservación para clases o eventos

Aula

Laboratorio FIEE-#

Reservar

May 24 - 28, 2021

	Lunes 24	Martes 25	Miércoles 26	Jueves 27	Viernes 28
7:00-8:00					
8:00-9:00					
9:00-10:00					
.					
.					
15:00-16:00					
16:00-17:00					
17:00-18:00					

Gestionar reservaciones

Gestionar laboratorio

Gestionar usuarios

Registros de entrada y salida

Reportes estadísticos

Comentarios y sugerencias

Figura 2.30 Sketch para gestionar reservaciones, requerimientos: RF008 y RF009

El asistente tendrá la opción de gestionar los elementos del laboratorio como aulas, software y computadoras, dichas opciones de gestión y la vista de los elementos del laboratorio se muestran en la figura 2.31.



Figura 2.31 Sketch para gestionar laboratorio, requerimientos RF010 y RF011

El asistente podrá gestionar los usuarios de la aplicación, se presentará una lista de los usuarios con las opciones de actualizar y eliminar, además de una lista desplegable para cambiar los tipos de usuarios (etiqueta 1) y un botón para crear un nuevo usuario (etiqueta 2), como se puede apreciar en la figura 2.32.



Figura 2.32 Sketch para gestionar usuarios, requerimiento RF015

Para el registro de entrada (figura 2.33) el asistente podrá determinar si el usuario asistió o no a la reserva hecha por el mismo.

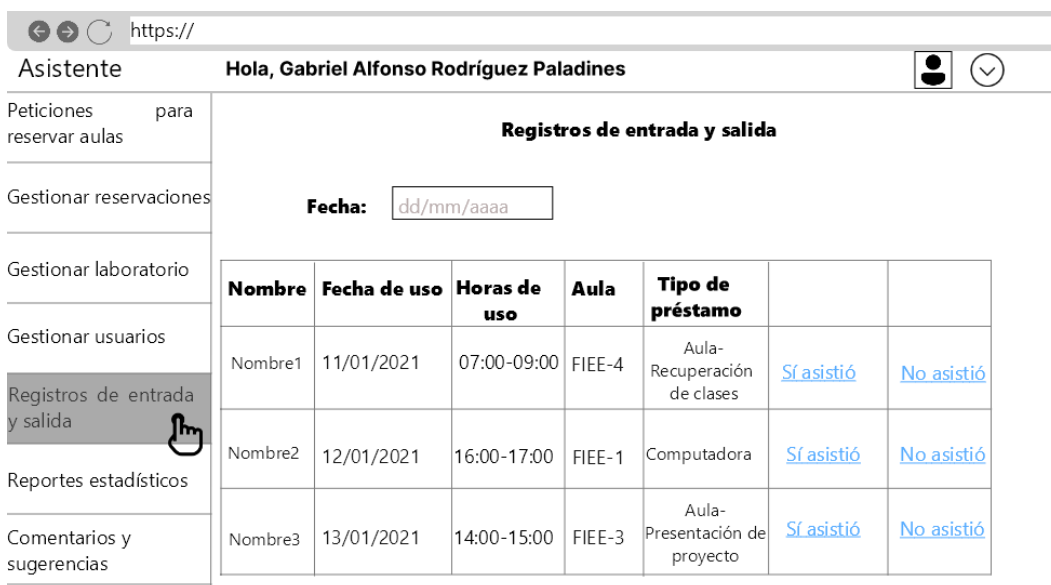


Figura 2.33 Sketch de registros de entrada y salida, requerimientos: RF0012 y RF013

El asistente podrá ver los reportes estadísticos respecto a la información de las reservas del laboratorio, el sketch correspondiente a dichos reportes se muestra en la figura 2.34.

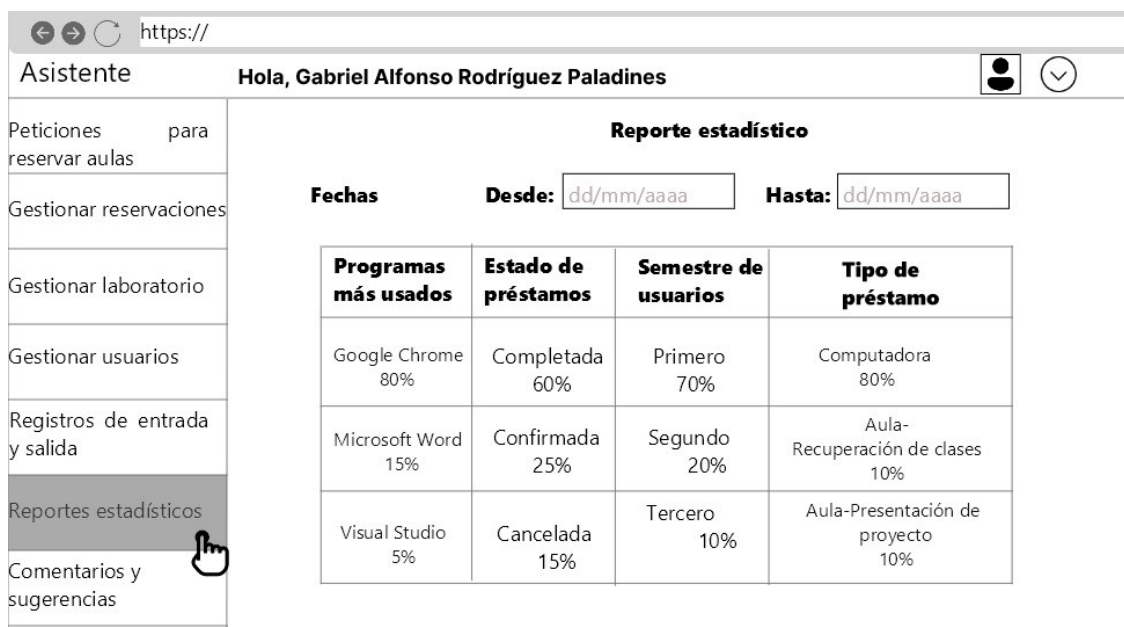


Figura 2.34 Sketch de reportes estadísticos, requerimiento RF014

El asistente también podrá visualizar los comentarios y sugerencias realizadas por los usuarios, dichos comentarios los puede filtrar por fechas, tal como se muestra en la figura 2.35.



Figura 2.35 Sketch de comentarios y sugerencias, requerimiento RF014

Los mapas de navegación correspondiente a cada rol de la aplicación se presentan a continuación. Para el rol de usuario el mapa de navegación se muestra en la figura 2.36.

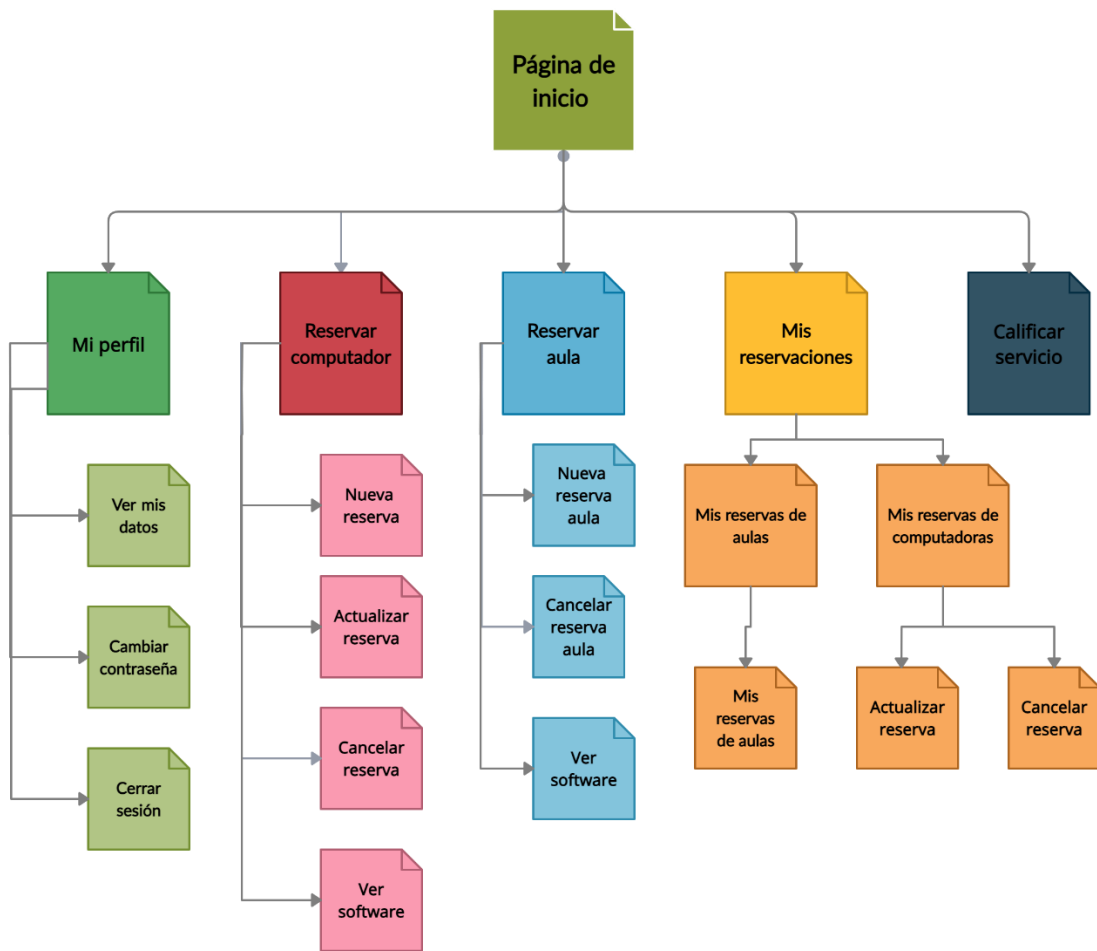


Figura 2.36 Mapa de navegación web del usuario

La figura 2.37 presenta el mapa de navegación web para los asistentes del laboratorio.

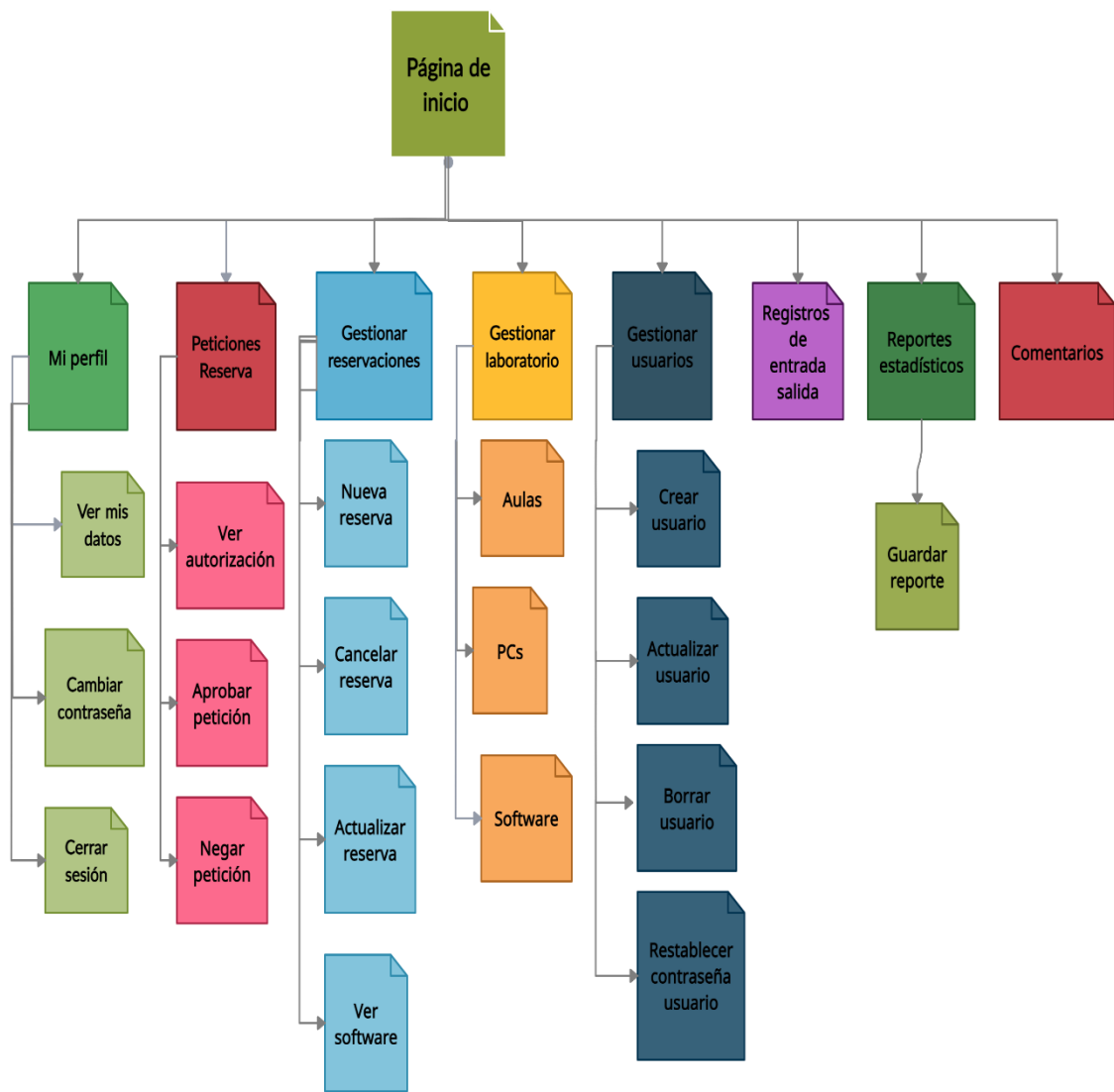


Figura 2.37 Mapa de navegación web asistencias del laboratorio

Para el administrador técnico el mapa de navegación web se visualiza en la figura 2.38.

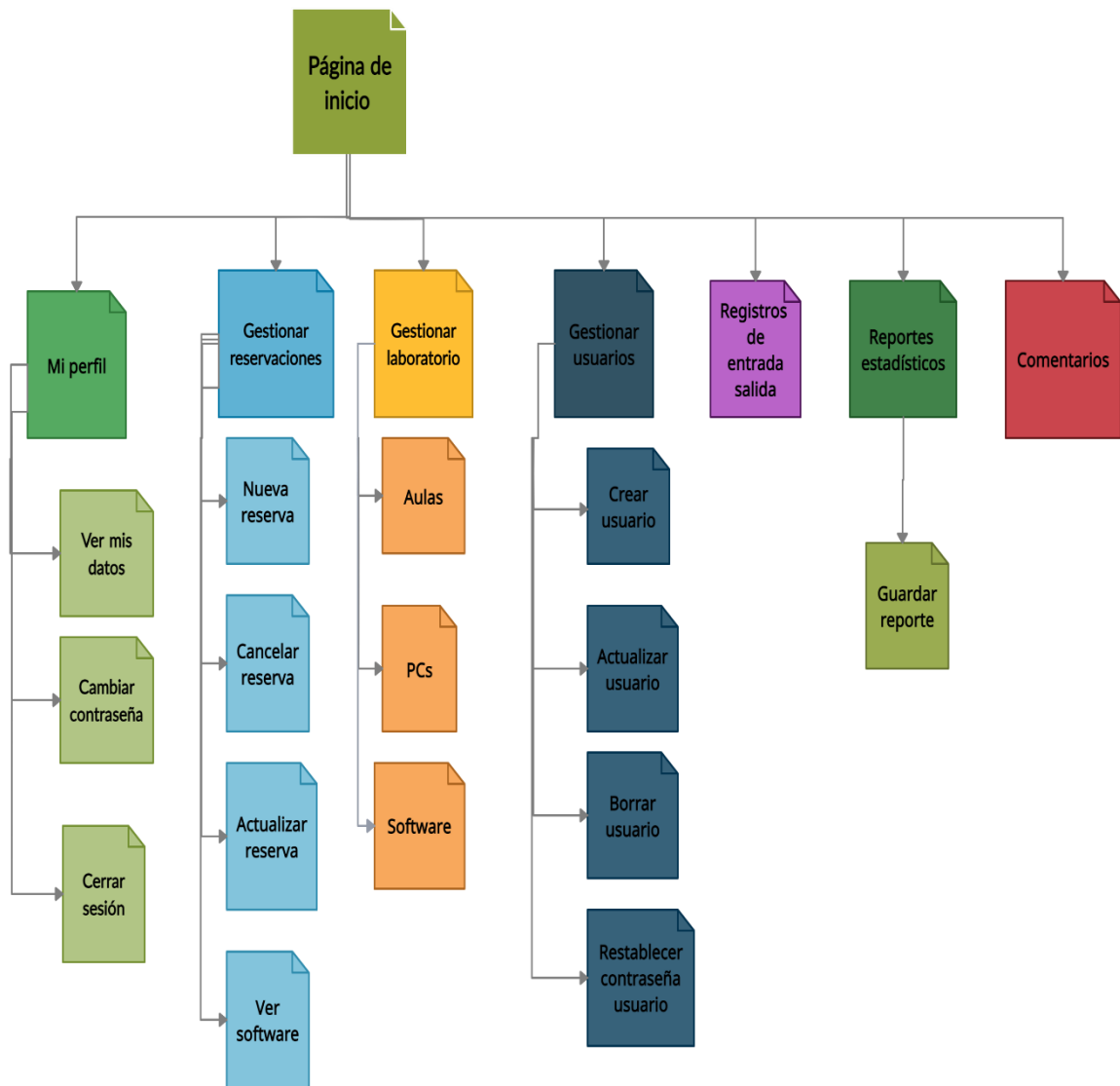


Figura 2.38 Mapa de navegación web del administrador técnico

2.2. IMPLEMENTACIÓN

En esta sección se detalla el proceso para la implementación del prototipo de aplicación web siguiendo el diseño especificado en la **sección 2.1**, que incluye: la codificación de las interfaces gráficas, la implementación de la base de datos y la lógica de funcionamiento, y el alojamiento de la aplicación web en la plataforma Microsoft Azure.

2.2.1. ACTUALIZACIÓN DEL TABLERO KANBAN

En la tabla 2.5 se muestra la actualización del tablero Kanban con las actividades por hacer, en progreso y realizadas correspondiente al inicio de la implementación de las diferentes capas del prototipo.

Tabla 2.5 Tablero Kanban con las actividades al inicio de la fase de implementación

Por hacer	En progreso	Realizadas
Codificar las vistas necesarias para la interfaz de usuario.	Instalar las herramientas tecnológicas necesarias para el desarrollo del prototipo.	Estudio de las herramientas tecnológicas necesarias para el desarrollo del prototipo.
Codificar las clases para las diferentes capas de nuestra aplicación.	Codificar de la base de datos SQL Server.	Entrevista al personal y jefe del laboratorio
Codificar la lógica de funcionamiento para los diferentes módulos del programa.		Entrevista a un grupo de estudiantes de la FIEE-EPN.
Alojar la aplicación en el servicio de computación en la nube Microsoft Azure.		Recopilación de datos necesarios para realizar el préstamo de las aulas y computadoras del laboratorio.
Realizar las pruebas de cada funcionalidad del prototipo mediante pruebas de usuario.		Elaboración del diagrama de contexto.
Realizar pruebas de validación de las funcionalidades.		Elaboración del diagrama de casos de uso.
Corregir errores detectados.		Elaboración del diagrama de actividad.
		Diseño de los sketches.
		Elaboración del diagrama de clases.
		Diseño de la base de datos.

2.2.2. INSTALACIÓN DE LAS HERRAMIENTAS DE SOFTWARE

Para desarrollar la aplicación web en el *framework* Blazor es necesario el entorno de desarrollo Visual Studio, para la instalación de Visual Studio se debe seguir las instrucciones de su sitio oficial [38].

Para verificar que la instalación se ha realizado correctamente se procede a crear la aplicación por defecto de Blazor Server; para ello se ejecuta el Visual Studio y se crea un nuevo proyecto como se muestra en la figura 2.39.

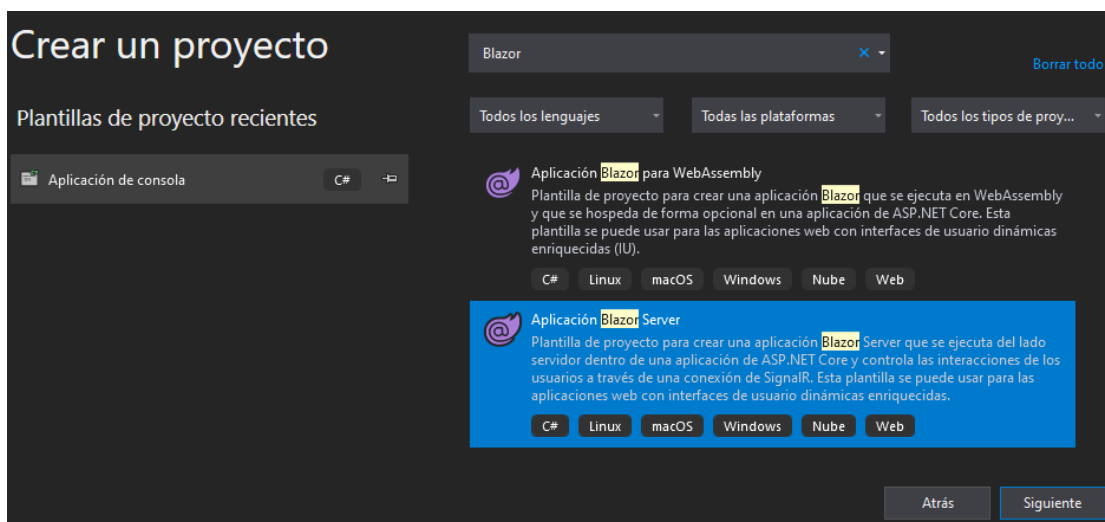


Figura 2.39 Creación de un proyecto Aplicación Blazor Server

Después de crear la aplicación se ejecuta la misma y se muestra en el navegador web que tengamos configurado por defecto. En la figura 2.40 se muestra la página de inicio de la aplicación creada con sus diferentes opciones.

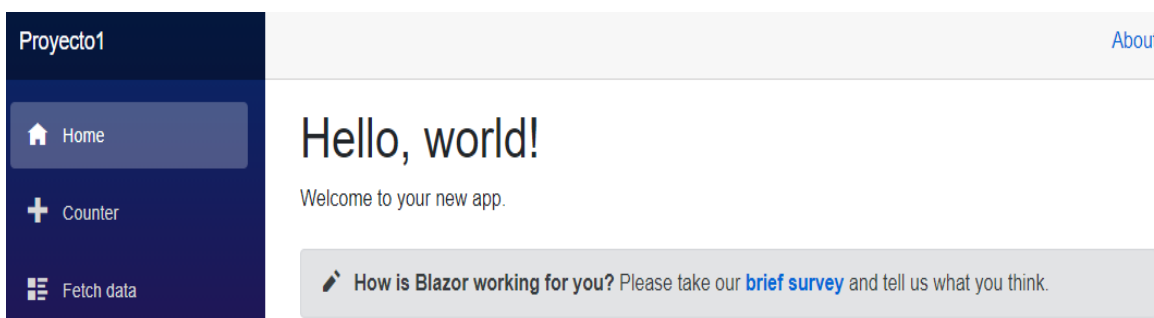


Figura 2.40 Página web de inicio del proyecto Blazor Server

Para el correcto funcionamiento del prototipo de aplicación web es necesario instalar algunos paquetes NuGets, estos paquetes facilitan tareas como, el envío de correos electrónicos, la autenticación de usuarios, la carga de archivos, la gestión de un calendario de eventos y la presentación de estadísticas a través de gráficos. Los paquetes NuGets se instalan a través del “Administrador de paquetes NuGets”; tal como se muestra en la figura 2.41.

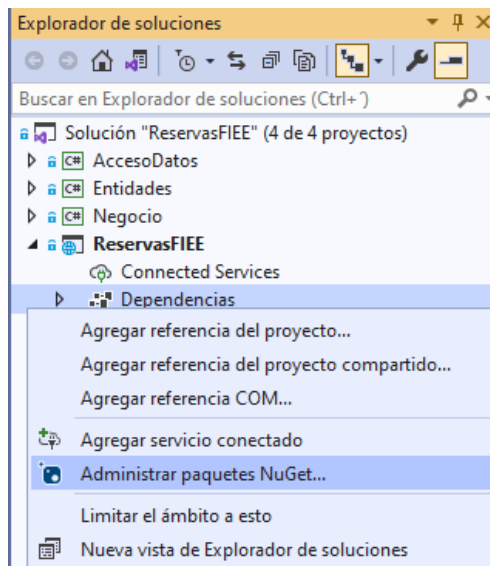


Figura 2.41 Administrador de paquetes NuGets

En la tabla 2.6 se muestran los paquetes necesarios para el correcto funcionamiento del prototipo de reservas.

Tabla 2.6 Paquetes NuGets instalados

Nombre	Funcionalidad	Versión
BlazorInputFile	Carga de archivos	v0.2.0
MailKit	Envío de correo electrónico	v2.10.1
Microsoft.AspNetCore.Authorization	Determina lo que un usuario tiene permitido hacer en la aplicación	v3.1.8
Microsoft.AspNetCore.Http	Maneja solicitudes y respuestas HTTP	v2.2.2
Microsoft.AspNetCore.Identity	Controla la autenticación, el inicio de sesión y los datos de usuario.	v2.2.0
Syncfusion.Blazor	Facilita la gestión de un calendario de eventos y la presentación de estadísticas a través de gráficos	v19.1.0.55

Algunas librerías vienen complementadas con una referencia JavaScript, dichas referencias deben ser incluidas en el archivo principal de inicio *Host.cshtml*. En el código 2.1, la línea 40 corresponde a la referencia del paquete BlazorInputFile, las líneas 43 - 47 se observan los scripts necesarios para el correcto funcionamiento de Bootstrap, finalmente en las líneas 50 - 52 se encuentra el script para los gráficos estadísticos del paquete Syncfusion.

```

39 | <!--Necesario para el inputFile-->
40 | <script src="_content/BlazorInputFile/inputfile.js"></script>
41 | <!--Bootstrap modal-->
42 | <!--JS necesario para el bootstrap-->
43 | <script src="https://code.jquery.com/jquery-3.5.1.slim.min.js"
44 |     integrity="sha384-DfXdz2htPH0lsSSs5nCTpuj/zy4C+0GpamoFVy38MVBnE+IbbVYUew+0rCXaRkfj" crossorigin="anonymous"></script>
45 |
46 | <script src="https://cdn.jsdelivr.net/npm/bootstrap@4.5.3/dist/js/bootstrap.bundle.min.js"
47 |     integrity="sha384-ho+j7jyWK8fNQe+A12Hb8AhRq26LrZ/JpcUGGOnY7RseNrtN/tE3MoK7ZeZDyx" crossorigin="anonymous"></script>
48 |
49 | <!--Necesario para los charts-->
50 | <script src="https://cdnjs.cloudflare.com/ajax/libs/lodash.js/4.17.20/lodash.min.js"
51 |     integrity="sha512-90vH1Z83AJY9DmIwa8kjkv79yF52n20xhSi2dZbIv0nC4E6m5AbH8N156kkk77ePmqD6tcZsfad1ueoaovvw=="
52 |     crossorigin="anonymous"></script>

```

Código 2.1 Referencias JavaScripts de las librerías y paquetes Nugets

2.2.3. IMPLEMENTACIÓN CAPA DE DATOS

Para el almacenamiento de la información del sistema de reserva se emplea SQL Server. En este gestor de base de datos se crearán las tablas y procedimientos almacenados necesarios para el correcto funcionamiento del prototipo; la creación de las tablas de datos se hará en base al diagrama relacional de la figura 2.19.

La creación de la base de datos se realiza mediante los comandos de SQL Server, en el código 2.2 las líneas 1-2 se muestra la creación de la base de datos llamada Laboratorio y en las líneas 4-5 se muestra los comandos necesarios para empezar a manipular la base de datos creada.

```

1  create database Laboratorio
2  go
3
4  use Laboratorio
5  go

```

Código 2.2 Crear y manipular una base de datos

A continuación, se crea las tablas siguiendo el diseño de la capa de datos expuesto en el **capítulo 2 apartado 2.1.3**. En el código 2.3 se muestra la creación de la tabla Usuario con sus respectivos atributos, en la línea 53 se muestra la sintaxis para realizar la referencia de llave foránea de la tabla TipoUsuario.

```

46 create table Usuario(
47     IdUsuario int identity(1,1) not null primary key,
48     NombreCompleto varchar(100),
49     CedulaIdentidad varchar(10),
50     CorreoInstitucional varchar(50) check (CorreoInstitucional like '%@%'),
51     NumeroCelular varchar(10),
52     Contraseña varbinary(MAX),
53     IdTipoUsuario int foreign key references TipoUsuario(IdTipoUsuario)
54 )
55 go

```

Código 2.3 Creación de la tabla usuario

Después de crear todas las tablas de la base de datos es necesario ingresar algunos datos indispensables desde el inicio del programa, uno de estos será el administrador técnico; la sintaxis para crear el administrador técnico se muestra en el código 2.4.

```

254 | ----Administrador tecnico
255 | insert into Usuario values ('Xavier Alexander Calderón Hinojosa',
256 | '1313693904', 'xavier.calderon@epn.edu.ec',
257 | '0987654321', ENCRYPTBYPASSPHRASE('XXXXXXXXXX', 'XXXXXXXXXX'), 3)
258 | go

```

Código 2.4 Ingreso en la base de datos del administrador técnico

Otro aspecto muy importante es la creación de los procedimientos almacenados lo cual permite realizar consultas desde la capa de negocio a la capa de datos de manera sencilla. En el código 2.5 se muestra la sintaxis para crear un procedimiento almacenado, el nombre del procedimiento se observa en la línea 538, los parámetros de entrada se ven en las líneas 539-541 y finalmente en las líneas 544-546 se ven la sentencia SQL del procedimiento en donde se obtienen las reservaciones de computadoras dependiendo de los parámetros de entrada.

```

538 create procedure sp_ObtenerMisReservasPC
539     @IdAula int,
540     @IdUsuario int,
541     @IdTipoReservacion int
542 as
543 begin
544     select * from Reservacion
545     where IdTipoReservacion=@IdTipoReservacion and IdAula=@IdAula and IdUsuario=@IdUsuario
546     order by FechaReservacion, HoraEntrada
547 end
548 go
549

```

Código 2.5 Creación del procedimiento almacenado sp_ObtenerMisReservasPC

El script completo de la base de datos Laboratorio se encuentra en el anexo D.

2.2.4. IMPLEMENTACIÓN DE CAPA DE NEGOCIO

Para consumir los recursos de la capa de datos se utilizó el *framework* Blazor con el lenguaje C#, esto para crear las clases y métodos expuestos en el **capítulo 2 apartado 2.1.4.**

Primeramente, se conectó la aplicación a la base de datos, para realizar este enlace se obtiene el *string* de conexión siguiendo las instrucciones ofrecidas en la documentación de Microsoft [39]. Una vez obtenido el *string* de conexión se crea un método que devuelva el *SqlConnection* correspondiente a la base de datos del laboratorio. En el código 2.6 se aprecia el método para abrir una conexión a la base de datos.

```
18 public SqlConnection conexión()
19 {
20     SqlConnection con = new
21         SqlConnection(@"Data Source=DESKTOP-QS8JAED\SQLEXPRESS;
22             Initial Catalog=Laboratorio;Integrated Security=True");
23     con.Open();
24     return con;
25 }
```

Código 2.6 Método para abrir una conexión a la base de datos

Una vez conectada la aplicación con la base de datos correspondiente se procede a realizar las clases con los respectivos métodos para consumir los recursos de la capa de datos. En el código 2.7 se presenta el método para comprobar el número de reservas de aulas que coinciden con la reserva que desea crear el usuario, esto con el fin de evitar que las reservas nuevas se solapen con reservas de aulas existentes, el método es estático (línea 817) para no tener que instanciar objetos, se observa en la línea 820 la declaración de una variable entera llamada *resultado*, la cual devolverá el número de reservas que solapan con la que el usuario desea realizar, en las líneas 821-824 se obtiene la conexión a la base de datos, en las líneas 825 - 839 se configura el comando necesario para obtener el número de reservaciones, luego se cierra la conexión en la línea 840 y finalmente en la línea 847 se devuelve la variable *resultado* con el número de reservaciones de aulas que coinciden con la reserva que se desea realizar.

```

817 public static int NumeroReservaAulasEntreHoras(TimeSpan horaEntrada, TimeSpan horaSalida,
818     DateTime fechaReservacion, int idAula)
819 {
820     int resultado=-1;
821     var db = new BaseDatos();
822     try
823     {
824         SqlConnection con = db.conexion();
825         SqlCommand comando = new SqlCommand("sp_NumeroReservaAulasEntreHoras", con);
826         comando.CommandType = CommandType.StoredProcedure;
827         comando.Parameters.AddWithValue("@HoraEntrada", horaEntrada);
828         comando.Parameters.AddWithValue("@HoraSalida", horaSalida);
829         comando.Parameters.AddWithValue("@FechaReservacion", fechaReservacion);
830         comando.Parameters.AddWithValue("@IdAula", idAula);
831         SqlDataReader reader = comando.ExecuteReader();
832
833         if (reader.HasRows)
834         {
835             while (reader.Read())
836             {
837                 resultado = reader.GetInt32(0);
838             }
839         }
840         con.Close();
841     }
842     catch (Exception e)
843     {
844         Console.WriteLine(e);
845         resultado = -1;
846     }
847     return resultado;
848 }

```

Código 2.7 Método para obtener en número de reservas de aulas

Debido a que la capa de negocio es una capa intermedia los métodos y clases pueden ser llamados desde los componentes Razor de la capa de aplicación. En la lógica de los archivos Razor se realiza el llamado de los métodos de la capa de negocio, en el código 2.8 se observa el método "MensajeError" (línea 891) en el cual el valor devuelto por la función "NumeroReservasAulasEntreHoras" se almacena en la variable "numeroCoincidencias" (línea 894), en las línea 900 si el valor de la variable "numeroCoincidencias" es igual a uno se retorna un *string* con un mensaje de advertencia, en la línea 902 si el valor es mayor a uno se retorna un mensaje de advertencia diferente al anterior, finalmente en la línea 905 se retorna un mensaje sin caracteres en caso de que no existan coincidencias.

```

890 //metodo para gestionar los mensajes
891 private string MensajesError()
892 {
893     //Aqui se compureba que la reserva que quiero realizar no coincida con ninguna de Aula
894     numeroCoincidencias = B_Reservacion.NumeroReservaAulasEntreHoras(horaEntrada.TimeOfDay,
895     horaSalida.TimeOfDay, fechaEscogida, idAula);
896     if (numeroCoincidencias > 0)
897     {
898         if (numeroCoincidencias == 1)
899         {
900             return "La reserva que se desea crear coincide con " + numeroCoincidencias + " reservación de aula";
901         }
902         return "La reserva que se desea crear coincide con " + numeroCoincidencias + " reservaciones de aulas";
903     }
904
905     return "";
906 }

```

Código 2.8 Llamado de los métodos de la capa de negocio desde un archivo Razor

El código completo con todos los métodos y clases se encuentra en el anexo E.

2.2.5. IMPLEMENTACIÓN CAPA DE PRESENTACIÓN

Para realizar las interfaces gráficas de la aplicación se sigue el diseño de los sketches mostrados en el **capítulo 2 apartado 2.1.5**. La base de la implementación de la capa de presentación es HTML, CSS y Bootstrap, estas interfaces se construyen en archivos Razor los cuales nos permiten escribir código HTML y realizar llamada a funciones en C#.

En el código 2.9 se observa que la mayoría es código HTML, además se observa la inclusión de código C#, en las líneas 31-34 se aprecia un bucle “foreach” el cual recorre una lista de aulas, para cada objeto aula se presenta la opción de invocar un evento llamado “MostrarAula” (línea 33), en las líneas 38 y 41 cada botón tiene relacionado un evento que se lanza al hacer clic sobre ellos, dichos eventos son métodos escritos en C#. El código completo de la aplicación con todas sus interfaces graficas se encuentra en el anexo E.

```

23 <!--Dropdown de las aulas-->
24 <div class="row">
25 <div class="col-1">
26 <button type="button" class="btn btn-info dropdown-toggle"
27     data-toggle="dropdown" aria-haspopup="true" aria-expanded="false">
28     Aula
29 </button>
30 <div class="dropdown-menu">
31     @foreach (var a in aulas)
32     {
33         <label @onclick="() => MostrarAula(a.IdAula, a.NombreAula)">@a.NombreAula</label>
34     }
35 </div>
36 </div>
37 <div class="col">
38 <button type="button" class="btn btn-primary" @onclick="VerSoftware">Ver software disponible</button>
39 </div>
40 <div class="col">
41 <button type="button" class="btn btn-primary" @onclick="Open">Reservar computador</button>
42 </div>
43 </div>
44

```

Código 2.9 Creación de una interfaz gráfica con un componente Razor

El resultado gráfico de la ejecución del código 2.9 se muestra en la figura 2.42.



Figura 2.42 Interfaz gráfica realizada en Blazor Server

2.2.6. ALOJAMIENTO DE LA APLICACIÓN EN MICROSOFT AZURE

El prototipo de aplicación web con su base de datos será alojado en Microsoft Azure, esto con el fin de hacer a la aplicación accesible a través de internet.

Como primer paso se realizará la migración de la base de datos alojada localmente en SQL Server Management Studio a Azure SQL Server; antes de realizar la migración se debe crear el grupo de recursos de Azure en donde se alojarán la base de datos y el prototipo de aplicación web [40]. Una vez creado el grupo de recursos con el servidor SQL se debe establecer una regla del firewall que permita a nuestra IP pública acceder a la base de datos de Azure y realizar la migración sin problemas. En las figuras 2.43 y 2.44 se muestra la configuración respectiva.

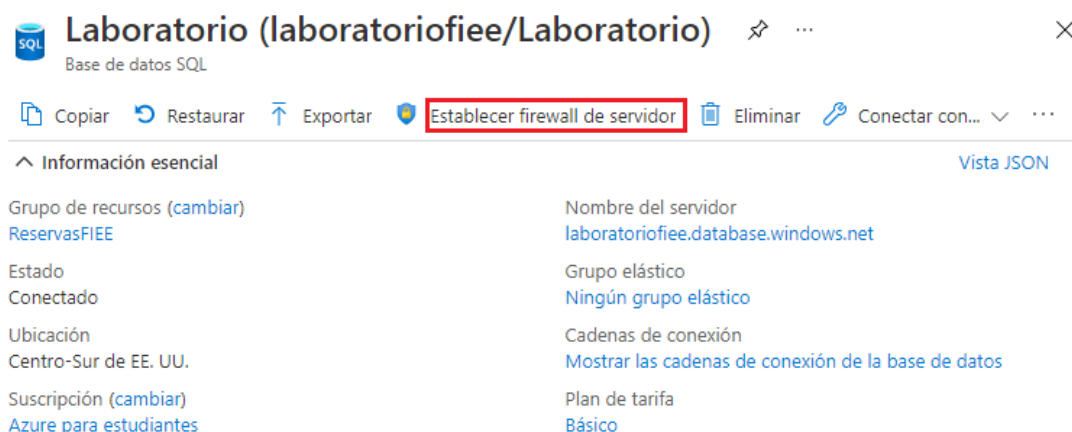


Figura 2.43 Establecer firewall del servidor Azure SQL Server

Inicio > LaboratorioFIEE (laboratoriofieee/LaboratorioFIEE) >

Configuración de firewall

laboratoriofieee (SQL server)

Guardar Descartar + Agregar IP de cliente

1 Haga clic aquí para crear un punto de conexión privado.
[Crear punto de conexión privado](#)

Versión de TLS mínima ⓘ
1.0 1.1 1.2

Directiva de conexión ⓘ
Predeterminado Proxy Redirigir

Permitir que los servicios y recursos de Azure accedan a este servidor ⓘ
Sí **No**

Dirección IP de cliente 45.70.239.228

Nombre de regla	IP inicial	IP final	
			...
MyIPPublic	45.70.239.228	45.70.239.228	...

1

Figura 2.44 Configuración del firewall Azure SQL para la IP del cliente

Después de realizar la configuración del firewall del servidor SQL de Azure se procede a realizar la migración de la base de datos [41].

Subida la base de datos de manera exitosa se procede con la aplicación web. Como paso previo al despliegue de la aplicación se debe cambiar el *string* de conexión en el código de esta; en la opción “Cadenas de conexión” de la base de datos de Azure se copia el *string* de conexión (figura 2.45) y se pega en el código de la aplicación (capítulo 2 apartado 2.1.4 código 2.6).

Laboratorio (laboratoriofieee/Laboratorio) | Cadenas de conexión

Base de datos SQL

[ADO.NET](#) [JDBC](#) [ODBC](#) [PHP](#) [Go](#)

ADO.NET (autenticación de SQL)

```
Server=tcp:laboratoriofieee.database.windows.net,1433;Initial Catalog=Laboratorio;Persist Security Info=False;User ID=roddy;Password={your_password};MultipleActiveResultSets=False;Encrypt=True;TrustServerCertificate=False;Connection Timeout=30;
```

Figura 2.45 Cadena de conexión de la base de datos Laboratorio

Después de validar correctamente la conexión desde la aplicación a la base de datos se procede a subir el sistema; la aplicación se alojará en la plataforma de computación en la nube Azure APP Service [42]. Ya desplegada la aplicación en el APP Service de Azure se habilita el protocolo Web Socket para que la autenticación de usuarios funcione correctamente [43].

Una vez alojada la aplicación se accede a ella a través de un navegador web, tal como se muestra en la figura 2.46.

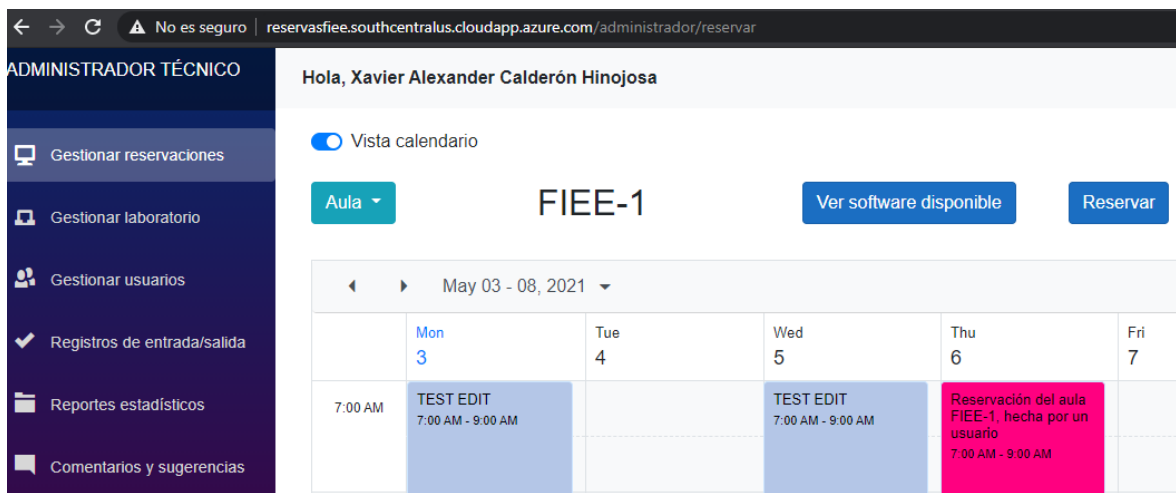


Figura 2.46 Acceso a la aplicación web alojada en Microsoft Azure

3. RESULTADOS Y DISCUSIÓN

En esta sección se presentan las pruebas realizadas a cada funcionalidad del prototipo. Aquí se verifica la validación de los requerimientos funcionales y no funcionales del prototipo definidos en el **capítulo 2 apartado 2.1.2**.

También, se muestran los resultados de la encuesta de satisfacción realizadas a los diferentes grupos de usuarios que conforman el prototipo de aplicación web.

Finalmente, se detalla los errores encontrados en las pruebas de funcionamiento con sus respectivas correcciones.

3.1 ACTUALIZACIÓN DEL TABLERO KANBAN

En la tabla 3.1 se muestra las actividades actualizadas correspondientes al inicio de la fase de resultados y discusión.

Tabla 3.1 Tablero Kanban con las actividades al inicio de la fase de resultados

Por hacer	En progreso	Realizadas
Realizar las pruebas de cada funcionalidad del prototipo mediante pruebas de usuario.	Realizar pruebas de validación de las funcionalidades.	Estudio de las herramientas tecnológicas necesarias para el desarrollo del prototipo.
Corregir errores detectados.		Entrevista al personal y jefe del laboratorio
		Entrevista a un grupo de estudiantes de la FIEE-EPN.
		Recopilación de datos necesarios para realizar el préstamo de las aulas y computadoras del laboratorio.
		Elaboración del diagrama de contexto.
		Elaboración del diagrama de casos de uso.
		Elaboración del diagrama de actividad.
		Diseño de los sketches.
		Elaboración del diagrama de clases.
		Diseño de la base de datos.
		Instalar las herramientas tecnológicas necesarias para el desarrollo del prototipo.
		Codificar de la base de datos SQL Server.

Por hacer	En progreso	Realizadas
		Codificar las vistas necesarias para la interfaz de usuario.
		Codificar las clases para las diferentes capas de nuestra aplicación.
		Codificar la lógica de funcionamiento para los diferentes módulos del programa.
		Alojar la aplicación en el servicio de computación en la nube Microsoft Azure.

3.2 PRUEBAS INICIALES DE FUNCIONAMIENTO

Previo a la realización de las pruebas de usuario se procede a validar alguna de las funcionalidades más importantes del prototipo; estas pruebas fueron realizadas de manera manual por el desarrollador.

En la página principal se colocan como usuario el número de cédula y en contraseña se coloca la contraseña correspondiente. Si las credenciales son incorrectas se mostrará un mensaje de error, tal como se muestra en la figura 3.1.

The image shows a login form with the following elements:

- Title:** Iniciar sesión
- Fields:**
 - Usuario *: 1313693903
 - Contraseña *: [Redacted with dots]
- Error Message:** Usuario o contraseña incorrecta! (displayed in red text)
- Checkbox:** Mostrar contraseña
- Button:** Ingresar (blue button)

Figura 3.1 Credenciales de usuario incorrectas

Si las credenciales son correctas se ingresará a la página principal dependiendo del rol de usuario que ingrese, en este caso se muestra en la figura 3.2 la página principal del usuario.

USUARIO

Hola, Roddy Steeven Zamora Rodriguez

Reservar computador

Reservar aula

Mis reservaciones

Calificar servicio

Tipos de reservaciones

- Reservas de computadoras
- Hora de clase
- Recuperación de clase
- Proyecto de titulación
- Presentación de proyecto
- Otros eventos

Aula FIEE-1 Ver software disponible Reservar computador

May 03 - 08, 2021 Today

	Mon 3	Tue 4	Wed 5	Thu 6	Fri 7	Sat 8
7:00 AM				Aula reservada por otro usuario 7:00 AM - 9:00 AM		Aula reservada por otro usuario 7:00 AM - 9:00 AM
8:00 AM						

Figura 3.2 Página principal para el rol usuario

3.2.1 PRUEBA DE FUNCIONAMIENTO DE RESERVAR COMPUTADOR

En la opción “Reservar computador” se procede a reservar un computador, se abrirá un modal en donde se indica la hora de inicio y fin de la reserva. Si la reserva que se intenta crear coincide con alguna reservación de aula saldrá un mensaje una advertencia informando del error (figura 3.3).

Nueva reservación

Aula * FIEE-3

Fecha * 25/05/2021

Hora entrada* 08:00

Hora salida* 10:00

La reserva que se desea crear coincide con 1 reservación de aula

Guardar Cerrar

Figura 3.3 Mensaje de horario no permitido al reservar un computador

Si la reserva que se intenta crear está dentro de un horario permitido saldrá un mensaje confirmando la reserva, tal como se ve en la figura 3.4.



The image shows a web form titled "Nueva reservación" with a close button (X) in the top right corner. The form contains four input fields, each with a red asterisk indicating it is required:

- Aula ***: A text input field containing "FIEE-3".
- Fecha ***: A date input field containing "19/04/2021" with a calendar icon on the right.
- Hora entrada***: A time input field containing "11:00" with a clock icon on the right.
- Hora salida***: A time input field containing "13:00" with a clock icon on the right.

Below the input fields, there is a green confirmation message: "Reserva guardada con éxito!". At the bottom of the form, there are two buttons: a blue "Guardar" button and a grey "Cerrar" button.

Figura 3.4 Reserva de computador exitoso

Al realizar una reserva de computador debe llegar un correo indicando los detalles de la reserva, sin embargo, dicho correo no fue enviado por el aplicativo web. Para averiguar el porqué del no envío de los detalles de la reserva se procedió a abrir el correo electrónico asociado a la aplicación, el cual es el que envía los mensajes a los diferentes usuarios; en la bandeja de entrada del correo emisor se puede apreciar un mensaje de advertencia de actividad de inicio de sesión inusual (figura 3.5), esto se da ya que la aplicación está alojada en el servicio de Azure el cual a su vez tiene sus instalaciones en Estados Unidos, por lo que el sistema de correo detecta una actividad sospechosa y bloquea la sesión de la aplicación web.

Actividad de inicio de sesión inusual de la cuenta Microsoft



Cuenta Microsoft

Comprueba tu cuenta

Hemos detectado algo inusual sobre un reciente inicio de sesión de la cuenta Microsoft pr*****@outlook.com. Por ejemplo, podrías haber iniciado sesión en un nuevo dispositivo, ubicación o aplicación.

Con el objetivo de garantizar la seguridad, hemos bloqueado el acceso a la bandeja de entrada, la lista de contactos y el calendario para ese inicio de sesión. Recomendamos revisar la actividad reciente. Proporcionaremos la ayuda necesaria para proteger la cuenta. Para volver a tener acceso, será necesario confirmar que no se trata de una actividad de una persona ajena.

[Revisar la actividad reciente](#)

Figura 3.5 Correo de inicio de sesión inusual de la cuenta Microsoft

Para solucionar este inconveniente se procedió a comprobar la actividad de inicio de sesión a través de las opciones del correo electrónico [44].


3.2.2 PRUEBA DE FUNCIONAMIENTO DE RESERVAR AULA


En esta opción para reservar un aula se debe escoger la fecha, las horas de reserva y además se debe adjuntar un archivo con la autorización del uso del aula, dicho archivo debe estar en formato PDF y no superar los 2 Megabytes.


En la figura 3.6 se muestra el modal para crear una reserva de aula, ahí se ven todos los parámetros que se necesitan para hacer este tipo de reserva.


Nueva reservación ✕

Aula *

Fecha * 

Hora entrada* 

Hora salida* 

Motivo* 

Autorización (PDF MAX: 2MB)* Ningún archivo seleccionado
Nombre: 07Gowth Hacking.pdf

Figura 3.6 Modal para crear una reserva de aula

Si la reserva se guarda correctamente el calendario se actualiza de manera automática con el evento creado, tal como se muestra en la figura 3.7; caso contrario el modal muestra un mensaje de error (figura 3.8) y el calendario no sufre alteraciones en sus eventos.

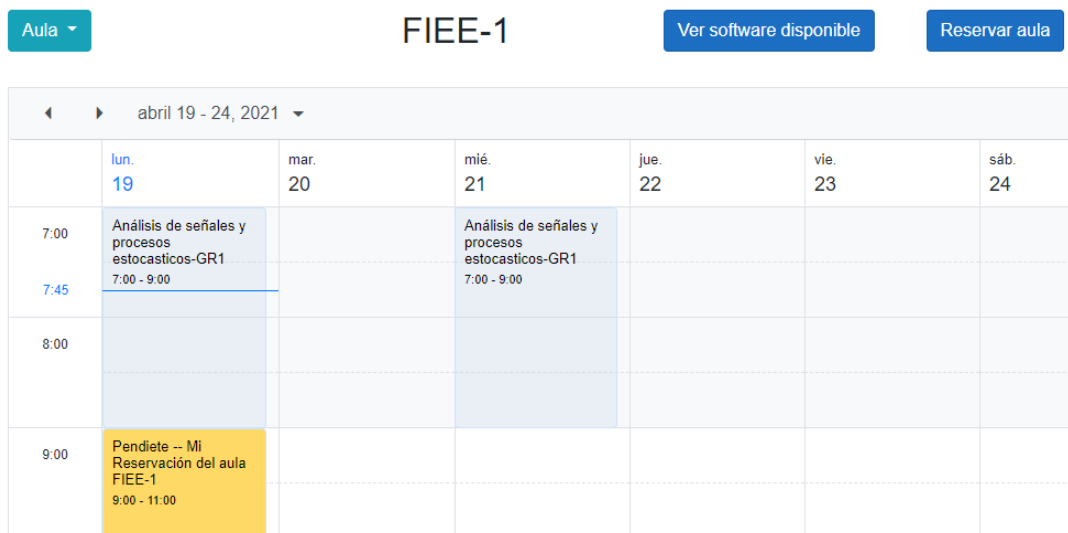


Figura 3.7 Eventos del usuario actualizados en el calendario

Nueva reservación ×

Aula *

Fecha *

Hora entrada*

Hora salida*

Motivo*

Autorización (PDF MAX: 2MB)* Ningún archivo seleccionado
Nombre: 04comunicacion roddy.pdf

La reserva que se desea crear coincide con 1 reservación de aula

Figura 3.8 Mensaje de horario no permitido al reservar un aula

3.2.3 PRUEBA DE FUNCIONAMIENTO DE MIS RESERVACIONES

En la figura 3.9 se muestra correctamente la lista de las reservaciones que posee el usuario, cada reservación se muestra con sus parámetros como la fecha, las horas y el estado.

Reservar computador

Reservar aula

Mis reservaciones

Aula ▾

Tipo ▾

Reservaciones de computadoras FIEE-4

Descripción	Fecha	Hora de entrada	Hora de salida	Estado	
Computadora FIEE-4-1 del aula FIEE-4	23/4/2021	09:00:00	11:00:00	Cancelada	<input type="button" value="Editar"/>

Figura 3.9 Lista de reservaciones del usuario

3.2.4 PRUEBA DE FUNCIONAMIENTO DE CALIFICAR SERVICIO

En esta opción se observa la lista de reservaciones a las que le falta realizar la encuesta de satisfacción del servicio de préstamo (ver figura 3.10).

Reservar aula

Mis reservaciones

Calificar servicio

Por favor escoja la reservación a evaluar

Descripción	Fecha	Hora de entrada	Hora de salida	
Computadora FIEE-3-1 del aula FIEE-3	19/4/2021	11:00:00	13:00:00	<input type="button" value="Evaluar"/>

Figura 3.10 Lista de reservaciones para evaluar

Al escoger la reservación a evaluar se despliega una lista de preguntas (ver figura 3.11) que el usuario debe contestar acorde a la atención recibida por el servicio de préstamo.

Computadora FIEE-3-1 del aula FIEE-3; lunes, 19 de abril de 2021 de 11:00 a 13:00

1: ¿Qué tipo de usuario eres? *

Alumno Profesor Personal administrativo Otro

2: ¿En qué semestre estas? *

1ero 2do 3ero 4to 5to 6to 7mo 8vo 9no 10mo

3: ¿Qué software utilizó? *

Chrome Codeblocks Dev C++ Eclipse Firefox Máquinas Virtuales de Ubuntu y CentOS
 Visual Studio Code

4: ¿Falto algún software? *

Sí No

Figura 3.11 Lista de preguntas para calificar el servicio de préstamo

3.3 PRUEBAS DE USUARIO

Una vez comprobado el correcto funcionamiento de tareas esenciales como el inicio de sesión, la creación de nuevas reservas, la visualización de eventos en el calendario, el envío y recepción de notificaciones al correo electrónico, etc. Se procedió a realizar las pruebas de usuario para cada rol de la aplicación, para el rol de usuario se pidió a tres estudiantes de la Escuela Politécnica Nacional que probaran la aplicación, y para los roles de asistente y administrador técnico se solicitó ayuda a personas con experiencia en las actividades efectuadas por el personal del laboratorio.

Para las respectivas pruebas se envió a cada usuario el manual del prototipo de reserva (Anexo F), en esta se encuentra una guía con las actividades que cada usuario puede realizar en el prototipo; al finalizar las pruebas indicadas las personas que realizaron las pruebas llenaron una encuesta indicando si se cumplieron o no los requerimientos de la aplicación web. El detalle de los resultados de las encuestas realizadas se encuentra en el Anexo G.

3.3.1 PRUEBAS DEL ROL USUARIO

En la tabla 3.2 se muestra un resumen con las respuestas de la encuesta de satisfacción de usuario obtenidas después de realizar las respectivas pruebas a los requerimientos para el rol usuario.

Tabla 3.2 Resultado de las encuestas de satisfacción del rol usuario

No.	Pregunta	Respuesta	
		Sí	No
1	¿El registro de usuario fue satisfactorio?	100%	0%
2	¿Al finalizar su registro le llego a su bandeja de correo sus credenciales de acceso?	100%	0%
3	¿El inicio de sesión fue correcto?	100%	0%
4	¿Al visualizar su perfil sus datos eran correctos?	100%	0%
5	¿Pudo cambiar su contraseña?	100%	0%
6	¿Al intentar reservar, pudo ver el software que disponían las diferentes aulas?	100%	0%
7	¿Pudo hacer una nueva reserva de computadora?	100%	0%
8	¿Al reservar una computadora le llego a su bandeja de correo la confirmación de la reserva?	100%	0%
9	¿El sistema le permite actualizar alguna de sus reservas de computadora?	100%	0%
10	¿Al actualizar alguna de sus reservas de computadora le llego a su bandeja de correo la actualización de esta?	100%	0%
11	¿Pudo cancelar alguna de sus reservaciones?	100%	0%
12	¿Pudo hacer una nueva reserva de aula?	100%	0%
13	¿Al hacer una nueva reserva de aula le llego a su bandeja de correo la aceptación o negación de la reserva?	100%	0%
14	¿En la opción 'Mis reservaciones' puede visualizar sus reservas de computadoras y aulas correctamente?	100%	0%
15	¿En la opción 'Mis reservaciones' puede cancelar o actualizar sus reservas?	100%	0%
16	¿En la opción 'Calificar servicio' puede realizar la encuesta de satisfacción?	100%	0%
17	¿Qué tan amigable e intuitiva fue la interfaz de usuario? (1 poco amigable – 5 muy amigable)	4.67 / 5.0	
18	¿Qué calificación le daría a la aplicación web? (1 muy mala – 5 muy buena)	5.0 / 5.0	

Como se puede observar todos los requerimientos del rol de usuario se cumplieron satisfactoriamente. En las pruebas para el rol de usuario se reportó un error, el detalle del error y su solución se muestra en la tabla 3.3.

Tabla 3.3 Errores encontrados en las pruebas del rol usuario

Error	Solución
Al momento de presionar el botón para realizar alguna reserva no se mostró la fecha actual	Se configuró la fecha asociada al modal de creación de reservas con la fecha del día en la que el usuario está realizando la reserva

3.3.2 PRUEBAS DEL ROL ASISTENTE DEL LABORATORIO

Las respuestas obtenidas en las encuestas de satisfacción para el rol de asistente se muestran en la tabla 3.4.

Tabla 3.4 Resultado de las encuestas de satisfacción del rol asistente

No.	Pregunta	Respuesta	
		Sí	No
1	¿El inicio de sesión fue correcto?	100%	0%
2	¿Al visualizar su perfil sus datos eran correctos?	67%	33%
3	¿Pudo cambiar su contraseña?	100%	0%
4	¿En las peticiones de reserva de aulas, pudo ver la autorización adjuntada por los usuarios?	100%	0%
5	¿Pudo aprobar o negar las peticiones de reserva de aula de los usuarios?	100%	0%
6	¿Al intentar reservar, pudo ver el software que disponían las diferentes aulas?	100%	0%
7	¿Pudo hacer una nueva reserva?	100%	0%
8	¿La aplicación le permitió actualizar reservas?	100%	0%
9	¿La aplicación le permitió eliminar reservas?	100%	0%
10	¿En la gestión del laboratorio el sistema le permitió visualizar las aulas, software y computadoras del laboratorio?	100%	0%
11	¿Pudo crear una nueva aula?	100%	0%
12	¿Pudo crear una nueva computadora?	100%	0%
13	¿Pudo crear un nuevo software?	100%	0%
14	¿El sistema le permitió actualizar aulas?	100%	0%
15	¿El sistema le permitió eliminar aulas?	100%	0%
16	¿El sistema le permitió actualizar software?	100%	0%
17	¿El sistema le permitió eliminar software?	100%	0%
18	¿El sistema le permitió actualizar computadoras?	100%	0%
19	¿El sistema le permitió eliminar computadoras?	100%	0%
20	¿Pudo visualizar los diferentes usuarios del sistema?	100%	0%
21	¿La aplicación le permitió restablecer la contraseña de algún usuario?	100%	0%
22	¿La aplicación le permitió actualizar la información de usuarios?	100%	0%
23	¿La aplicación le permitió eliminar usuarios?	100%	0%
24	¿Pudo visualizar el registro de entrada y salida de usuarios?	100%	0%
25	¿Pudo generar un nuevo registro de entrada y salida de usuario?	100%	0%
26	¿En la opción de reportes estadísticos, se visualiza correctamente la información de las reservas?	100%	0%
27	¿Pudo guardar el reporte estadístico de manera exitosa?	100%	0%
28	¿En la opción de comentarios, se visualiza correctamente los comentarios y sugerencias hecha por los usuarios?	100%	0%

No.	Pregunta	Respuesta
29	¿Qué tan amigable e intuitiva fue la interfaz de usuario? (1 poco amigable – 5 muy amigable)	4.33 / 5.0
30	¿Qué calificación le daría a la aplicación web? (1 muy mala – 5 muy buena)	4.33 / 5.0

En las pruebas para el rol de asistentes del laboratorio se presentaron errores de funcionamiento, el detalle de estos errores y su respectiva solución se muestra en la tabla 3.5.

Tabla 3.5 Errores encontrados en las pruebas del rol usuario

Error	Solución
Un asistente encontró su información personal errónea	Se actualizó los datos personales del asistente de manera correcta
Uno de los asistentes no pudo generar el registro de entrada y salida	Se configuro en la lógica de la aplicación el método de registro de entrada y salida, esto para que cambie el estado de la reserva al que se le hace el registro

Además, las personas que realizaron las pruebas de este rol hicieron algunas observaciones gráficas para mejorar la interacción de los usuarios con el prototipo de aplicación web; estas sugerencias se muestran en la tabla 3.6.

Tabla 3.6 Sugerencias obtenidas en las pruebas del rol asistente

Sugerencia	Página donde aplicar las sugerencias
Colocar la frase “Iniciar sesión” al centro y con negrilla para resaltar con los demás parámetros.	reservasfiec.com
Centrar horizontalmente la frase “Petición para reservar aulas”	reservasfiec.com/asistente/peticionesAulas
Cambiar la opción del menú de “Gestionar reservaciones” por “Gestionar reservas”	reservasfiec.com/asistente/reservar
Cambiar la lista desplegable por tres botones (aulas, computadoras, software)	reservasfiec.com/asistente/gestionarAulas

Sugerencia	Página donde aplicar las sugerencias
No mostrar los IDs de las aulas, software y computadoras	reservasfiec.com/asistente/gestionarAulas
Reemplazar el botón que tiene una lista desplegable con los diferentes roles, por tres botones que representen cada uno de estos	reservasfiec.com/asistente/gestionarUsuarios
Asegurarse que la tipografía sea la misma	reservasfiec.com/perfil

3.3.3 PRUEBAS DEL ROL ADMINISTRADOR TÉCNICO

La tabla 3.7 muestra el resumen de las respuestas obtenidas en las encuestas de satisfacción para el rol administrador técnico.

Tabla 3.7 Resultado de las encuestas de satisfacción del rol administrador técnico

No.	Pregunta	Respuesta	
		Sí	No
1	¿El inicio de sesión fue correcto?	100%	0%
2	¿Al visualizar su perfil sus datos eran correctos?	100%	0%
3	¿Pudo cambiar su contraseña?	100%	0%
4	¿Al intentar reservar, pudo ver el software que disponían las diferentes aulas?	100%	0%
5	¿Pudo hacer una nueva reserva?	100%	0%
6	¿La aplicación le permitió actualizar reservas?	100%	0%
7	¿La aplicación le permitió eliminar reservas?	100%	0%
8	¿En la gestión del laboratorio el sistema le permitió visualizar las aulas, software y computadoras del laboratorio?	100%	0%
9	¿Pudo crear una nueva aula?	100%	0%
10	¿Pudo crear una nueva computadora?	100%	0%
11	¿Pudo crear un nuevo software?	100%	0%
12	¿El sistema le permitió actualizar aulas?	100%	0%
13	¿El sistema le permitió eliminar aulas?	100%	0%
14	¿El sistema le permitió actualizar software?	100%	0%
15	¿El sistema le permitió eliminar software?	100%	0%
16	¿El sistema le permitió actualizar computadoras?	100%	0%
17	¿El sistema le permitió eliminar computadoras?	100%	0%
18	¿Pudo visualizar los diferentes usuarios del sistema?	100%	0%
19	¿La aplicación le permitió restablecer la contraseña de algún usuario?	100%	0%
20	¿La aplicación le permitió actualizar la información de usuarios?	100%	0%

No.	Pregunta	Respuesta	
		Sí	No
21	¿La aplicación le permitió eliminar usuarios?	100%	0%
22	¿Pudo visualizar el registro de entrada y salida de usuarios?	100%	0%
23	¿Pudo generar un nuevo registro de entrada y salida de usuario?	100%	0%
24	¿En la opción de reportes estadísticos, se visualiza correctamente la información de las reservas?	100%	0%
25	¿Pudo guardar el reporte estadístico de manera exitosa?	100%	0%
26	¿En la opción de comentarios, se visualiza correctamente los comentarios y sugerencias hecha por los usuarios?	100%	0%
27	¿Qué tan amigable e intuitiva fue la interfaz de usuario? (1 poco amigable – 5 muy amigable)	4.0 / 5.0	
28	¿Qué calificación le daría a la aplicación web? (1 muy mala – 5 muy buena)	5.0 / 5.0	

En las pruebas de usuario para el administrador técnico se cumplieron todas las funcionalidades y no se presentaron errores.

3.4 ACTUALIZACIÓN FINAL DEL TABLERO KANBAN

Después de realizar las correcciones respectivas y aplicar las sugerencias de los diferentes usuarios se procede a realizar la última actualización del tablero Kanban, por lo que se da por terminado todas las actividades necesarias para la realización de este trabajo de titulación. El tablero Kanban actualizado se muestra en la tabla 3.8.

Tabla 3.8 Actualización final del tablero Kanban

Por hacer	En progreso	Realizadas
		Estudio de las herramientas tecnológicas necesarias para el desarrollo del prototipo.
		Entrevista al personal y jefe del laboratorio
		Entrevista a un grupo de estudiantes de la FIEE-EPN.
		Recopilación de datos necesarios para realizar el préstamo de las aulas y computadoras del laboratorio.
		Elaboración del diagrama de contexto.
		Elaboración del diagrama de casos de uso.
		Elaboración del diagrama de actividad.
		Diseño de los sketches.

Por hacer	En progreso	Realizadas
		Elaboración del diagrama de clases.
		Diseño de la base de datos.
		Instalar las herramientas tecnológicas necesarias para el desarrollo del prototipo.
		Codificar de la base de datos SQL Server.
		Codificar las vistas necesarias para la interfaz de usuario.
		Codificar las clases para las diferentes capas de nuestra aplicación.
		Codificar la lógica de funcionamiento para los diferentes módulos del programa.
		Alojar la aplicación en el servicio de computación en la nube Microsoft Azure.
		Realizar las pruebas de cada funcionalidad del prototipo mediante pruebas de usuario.
		Realizar pruebas de validación de las funcionalidades.
		Corregir errores detectados.

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

Al concluir este proyecto de titulación se logró desarrollar un prototipo de aplicación web de reserva de computadoras y aulas del Laboratorio de Informática FIEE-EPN. Este prototipo ayuda al personal con la gestión del servicio de préstamo del laboratorio y permite a la comunidad de la Escuela Politécnica Nacional reservar computadoras y aulas del Laboratorio de Informática FIEE-EPN a través de una aplicación web.

Para el desarrollo de este trabajo de titulación se adquirieron nuevos conocimientos acerca del desarrollo web, la gestión de datos con SQL Server, el *framework* Blazor, la arquitectura por capas y los servicios de computación en la nube de Azure.

La metodología Kanban permitió organizar de manera efectiva las tareas necesarias para el desarrollo de este trabajo de titulación, al ser un método visual permite usarlo en las diferentes fases del desarrollo del prototipo de aplicación web.

Las entrevistas realizadas a las personas involucradas en el servicio de préstamo del Laboratorio de Informática FIEE-EPN permitió obtener los requerimientos funcionales y no funcionales del prototipo de aplicación web. En base a estos requerimientos se diseñaron los diferentes elementos de las respectivas capas que conforman el prototipo.

Se implementó los elementos que conforman la aplicación siguiendo el diseño de estos. En la capa de datos se creó con SSMS una base de datos SQL con todas las tablas y procedimientos almacenados necesarios para la manipulación de la información, en la capa de negocio se creó todas las clases C# con sus respectivos métodos y propiedades para obtener información de la capa de datos y enviarlos a la capa de presentación, y en la capa de presentación se creó la interfaz gráfica a través de archivos Razor, el lenguaje HTML y la herramienta Bootstrap. Finalmente, para el alojamiento de la aplicación se eligió los servicios que ofrece Microsoft Azure debido a su fácil integración con el *framework* utilizado para el desarrollo de este prototipo, así como de la base de datos que utiliza la aplicación para su correcto funcionamiento.

Las pruebas de usuario realizadas por un grupo de estudiantes y personal del Laboratorio de Informática FIEE-EPN permitieron encontrar errores de funcionamiento e identificar mejoras para la interfaz gráfica del prototipo. Estas pruebas permitieron validar el correcto funcionamiento de todos los requerimientos de la aplicación, es decir que la aplicación funciona como se esperaba y sirve como base para un sistema de reserva online para el Laboratorio de Informática FIEE-EPN lo cual beneficia a los estudiantes, profesores y

personal encargado del laboratorio; ya que permite realizar reservas a través de un aplicativo web, permite a los usuarios conocer los horarios y el software de las distintas aulas del laboratorio, y ayuda a los asistentes a gestionar de manera organizada las reservaciones así como las computadoras, software y aulas del laboratorio.

4.2 RECOMENDACIONES

El prototipo de aplicación web para las reservas de computadoras y aulas del Laboratorio de Informática FIEE-EPN cumple satisfactoriamente con los requerimientos funcionales y no funcionales, este prototipo puede mejorarse y agilizarse de muchas maneras para lo cual se presentan recomendaciones para futuros trabajos y recomendaciones técnicas para la mejora de la metodología de desarrollo.

Recomendaciones para futuros trabajos:

- Debido a que la lógica de la aplicación web está realizada con el lenguaje de programación C# y la herramienta Syncfusion también es compatible con el *framework* de desarrollo de aplicaciones móviles Xamarin se recomienda escalar este prototipo a una aplicación móvil.
- Se recomienda la expansión del prototipo a otros laboratorios de la Escuela Politécnica Nacional que ofrezcan el servicio de préstamo de sus aulas y computadoras, esto para tener un sistema de reserva online que integre a varios laboratorios y se acople a las necesidades tecnológicas actuales.
- Para trabajos futuros se recomienda añadir otras funcionalidades como la personalización de las encuestas de satisfacción que se realizan a los usuarios, el envío de los reportes estadísticos a los correos de los asistentes del laboratorio, la inclusión de un código de bien para los accesorios de las computadoras y la sincronización de los eventos de los usuarios con su calendario de Microsoft.

Recomendaciones técnicas:

- Es recomendable tener en cuenta las versiones de los paquetes utilizados en el desarrollo de este prototipo, ya que al momento de actualizarlos o cambiar de versión existe la posibilidad de que la aplicación ya no funcione.
- Para visualizar el detalle de los errores presentados al ejecutar el prototipo en el navegador se recomienda configurar la variable de entorno `ASPNETCORE_ENVIRONMENT` en desarrollo.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] R. Oakley, "The Advantages of Online Reservations", *Travel Tips - USA Today*, 2020. [Online]. Available: <https://traveltips.usatoday.com/advantages-online-reservations-63078.html>. [Accessed: 09- Nov- 2020].

[2] WP Amelia Staff, "Advantages of booking online appointments versus over the phone", *Amelia WordPress Booking Plugin*, 2021. [Online]. Available: <https://wpamelia.com/booking-online/>. [Accessed: 05- Ene- 2021].

[3] S. Luján, Programación de aplicaciones web: Historia, principios básicos y clientes web, San Vicente del Raspeig Alicante, p.48.

[4] TechTarget "What is Web Application (Web Apps) and its Benefits", *SearchSoftwareQuality*, 2021. [Online]. Available: <https://searchsoftwarequality.techtarget.com/definition/Web-application-Web-app>. [Accessed: 12- Feb- 2021].

[5] S. Luján, Programación de aplicaciones web: Historia, principios básicos y clientes web, San Vicente del Raspeig Alicante, p.49.

[6] M. Pedregosa, Diseño y programación de páginas web, p.155.

[7] J. Martínez, Aplicaciones web, Madrid España, 2014 p. 68.

[8] T. Sagara, "What Languages are Used for Back End Development?", *Medium*, 2021. [Online]. Available: <https://sagaratechnology.medium.com/what-languages-are-used-for-back-end-development-71a8a10c135c>. [Accessed: 12- Feb- 2021].

[9] Packt Editorial Staff, "What is a multi-layered software architecture? | Packt Hub", *Packt Hub*, 2021. [Online]. Available: <https://hub.packtpub.com/what-is-multi-layered-software-architecture/>. [Accessed: 12- Feb- 2021].

[10] IBM Cloud Education, "What is Three-Tier Architecture", *ibm.com*, 2020. [Online]. Available: <https://www.ibm.com/cloud/learn/three-tier-architecture>. [Accessed: 12- Feb- 2021].

[11] P. Christensson, "HTML (Hyper-Text Markup Language) Definition", *Techterms.com*, 2021. [Online]. Available: <https://techterms.com/definition/html>. [Accessed: 13- Feb- 2021].

[12] P. Christensson, "CSS (Cascading Style Sheet) Definition", *Techterms.com*, 2021. [Online]. Available: <https://techterms.com/definition/css>. [Accessed: 13- Feb- 2021].

[13] Amelia, "What is Bootstrap? An In-depth Guide of the Framework", *Amelia WordPress Booking Plugin*, 2021. [Online]. Available: <https://wpamelia.com/what-is-bootstrap/>. [Accessed: 13- Feb- 2021].

[14] Tutorialspoint, "JavaScript - Overview - Tutorialspoint", *Tutorialspoint.com*, 2021. [Online]. Available: https://www.tutorialspoint.com/javascript/javascript_overview.htm. [Accessed: 13- Feb- 2021].

[15] Oracle Corporation, "Descripción de capas lógicas (Descripción general técnica de Sun Java Enterprise System 5)", *Docs.oracle.com*, 2021. [Online]. Available: <https://docs.oracle.com/cd/E19528-01/820-0888/aaubb/index.html>. [Accessed: 13- Feb- 2021].

- [16] Microsoft, "A Tour of C# - C# Guide", *Docs.microsoft.com*, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/csharp/tour-of-csharp/>. [Accessed: 13- Feb- 2021].
- [17] Microsoft, "ADO.NET", *Docs.microsoft.com*, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/dotnet/framework/data/adonet/>. [Accessed: 13- Feb- 2021].
- [18] J. Esterkin, "Layered Architecture", *OpenClassrooms*, 2021. [Online]. Available: <https://openclassrooms.com/en/courses/6397806-design-your-software-architecture-using-industry-standard-patterns/6896176-layered-architecture>. [Accessed: 15- Feb- 2021].
- [19] Microsoft, "SQL Server Management Studio (SSMS) - SQL Server Management Studio (SSMS)", *Docs.microsoft.com*, 2021. [Online]. Available: <https://docs.microsoft.com/es-es/sql/ssms/sql-server-management-studio-ssms?view=sql-server-ver15>. [Accessed: 15- Feb- 2021].
- [20] Microsoft, "Blazor | Build client web apps with C# | .NET", *Microsoft*, 2021. [Online]. Available: <https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor>. [Accessed: 26- Feb- 2021].
- [21] D. Roth, J. Fritz, T. Southwick, *Blazor for ASP.NET Web Forms Developers*, 1.0.1 ed, Redmond, Washington 98052-6399, 2020, p. 14.
- [22] Microsoft, "Introducción a ASP.NET Core Blazor", *Docs.microsoft.com*, 2021. [Online]. Available: <https://docs.microsoft.com/es-es/aspnet/core/blazor/?view=aspnetcore-5.0>. [Accessed: 27- Feb- 2021].
- [23] D. Roth, J. Fritz, T. Southwick, *Blazor for ASP.NET Web Forms Developers*, 1.0.1 ed, Redmond, Washington 98052-6399, 2020, p. 8.
- [24] Microsoft, "ASP.NET Core Blazor hosting models", *Docs.microsoft.com*, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/core/blazor/hosting-models?view=aspnetcore-5.0>. [Accessed: 27- Feb- 2021].
- [25] E. Charbeneau, "A New Era of Productivity with Blazor", *Codemag.com*, 2021. [Online]. Available: <https://www.codemag.com/Article/1911052/A-New-Era-of-Productivity-with-Blazor>. [Accessed: 27- Feb- 2021].
- [26] Microsoft, "ASP.NET Core Blazor project structure", *Docs.microsoft.com*, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/core/blazor/project-structure?view=aspnetcore-5.0>. [Accessed: 27- Feb- 2021].
- [27] Syncfusion, "Introduction in Blazor - Syncfusion", *Blazor.syncfusion.com*, 2021. [Online]. Available: <https://blazor.syncfusion.com/documentation/introduction/#syncfusion-blazor>. [Accessed: 12- Mar- 2021].
- [28] Syncfusion, "Syncfusion Community License", *Blazor.syncfusion.com*, 2021. [Online]. Available: <https://www.syncfusion.com/products/communitylicense>. [Accessed: 12- Mar- 2021].
- [29] Syncfusion, "Blazor Components Examples & Demos | Syncfusion", *Blazor.syncfusion.com*, 2021. [Online]. Available: <https://blazor.syncfusion.com/demos/>. [Accessed: 12- Mar- 2021].

- [30] Syncfusion, "Blazor Scheduler Component - A Complete Event Calendar", *Blazor.syncfusion.com*, 2021. [Online]. Available: <https://www.syncfusion.com/blazor-components/blazor-scheduler>. [Accessed: 12- Mar- 2021].
- [31] Syncfusion, "Blazor Scheduler Local Data Example - Syncfusion Demos", *Blazor.syncfusion.com*, 2021. [Online]. Available: <https://blazor.syncfusion.com/demos/scheduler/local-data?theme=bootstrap4>. [Accessed: 12- Mar- 2021].
- [32] Syncfusion, "Blazor Charts - Beautiful & Interactive Charts", *Blazor.syncfusion.com*, 2021. [Online]. Available: <https://www.syncfusion.com/blazor-components/blazor-charts>. [Accessed: 12- Mar- 2021].
- [33] Syncfusion, "Example of Column Chart in Blazor Charts Component", *Blazor.syncfusion.com*, 2021. [Online]. Available: <https://blazor.syncfusion.com/demos/chart/column?theme=bootstrap4> [Accessed: 12- Mar- 2021].
- [34] M. Copeland, J. Soh, A. Puca, M. Manning, D. Gollob, Microsoft Azure: Planning, Deploying, and Managing your Data Center in the Cloud, New York, 2015, p. 3.
- [35] Microsoft, "Qué es Azure: Servicios en la nube de Microsoft | Microsoft Azure", *Azure.microsoft.com*, 2021. [Online]. Available: <https://azure.microsoft.com/en-us/overview/what-is-azure/>. [Accessed: 27- Mar- 2021].
- [36] Microsoft, "App Service | Microsoft Azure", *Azure.microsoft.com*, 2021. [Online]. Available: <https://azure.microsoft.com/es-es/services/app-service/>. [Accessed: 27- Mar- 2021].
- [37] Microsoft, "SQL Database: base de datos como servicio en la nube | Microsoft Azure", *Azure.microsoft.com*, 2021. [Online]. Available: <https://azure.microsoft.com/es-es/services/sql-database/>. [Accessed: 27- Mar- 2021].
- [38] Microsoft, "Install Visual Studio", *Visual Studio*, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/visualstudio/install/install-visual-studio?view=vs-2019>. [Accessed: 22- Apr- 2021].
- [39] Microsoft, "Add new connections - Visual Studio", *Docs.microsoft.com*, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/visualstudio/data-tools/add-new-connections?view=vs-2019>. [Accessed: 27- Apr- 2021].
- [40] Microsoft, "Create a single database - Azure SQL Database", *Docs.microsoft.com*, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/azure/azure-sql/database/single-database-create-quickstart?tabs=azure-portal>. [Accessed: 03- May- 2021].
- [41] D. SQL, "Deploy the database to Azure SQL - Sitefinity CMS Deploy and upgrade", *Progress.com*, 2021. [Online]. Available: <https://www.progress.com/documentation/sitefinity-cms/deploy-the-database-to-azure-sql>. [Accessed: 03- May- 2021].
- [42] Microsoft, "Publish to Azure App Service - Visual Studio", *Docs.microsoft.com*, 2021. [Online]. Available: <https://docs.microsoft.com/en-us/visualstudio/deployment/quickstart-deploy-to-azure?view=vs-2019>. [Accessed: 03- May- 2021].

[43] Microsoft, "Introduction to WebSockets on Windows Azure Web Sites | Blog y actualizaciones de Azure | Microsoft Azure", *Azure.microsoft.com*, 2021. [Online]. Available: <https://azure.microsoft.com/es-mx/blog/introduction-to-websockets-on-windows-azure-web-sites/>. [Accessed: 03- May- 2021].

[44] Microsoft, "Comprobar la actividad reciente de inicio de sesión en tu cuenta de Microsoft", *Support.microsoft.com*, 2021. [Online]. Available: <https://support.microsoft.com/es-es/account-billing/comprobar-la-actividad-reciente-de-inicio-de-sesi%C3%B3n-en-tu-cuenta-de-microsoft-5b3cfb8e-70b3-2bd6-9a56-a50177863357>. [Accessed: 06- May- 2021].

ANEXOS

ANEXO A. Entrevistas y definición de requerimientos para los usuarios.

ANEXO B. Entrevistas y definición de requerimientos para el personal del laboratorio.

ANEXO C. Información del laboratorio FIEE-EPN necesaria para el desarrollo del prototipo.

ANEXO D. Script de la base de datos.

ANEXO E. Código del prototipo.

ANEXO F. Manual de usuario.

ANEXO G. Encuestas de satisfacción de requerimientos.

ORDEN DE EMPASTADO